Studies in Computational Intelligence 312

Patricia Melin
Janusz Kacprzyk
Witold Pedrycz (Eds.)

Soft Computing for Recognition Based on Biometrics

Springer

Studies in Computational Intelligence, Volume 312

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

Vol. 289. Anne Håkansson, Ronald Hartung, and Ngoc Thanh Nguyen (Eds.)
Agent and Multi-agent Technology for Internet and Enterprise Systems, 2010
ISBN 978-3-642-13525-5

Vol. 290. Weiliang Xu and John Bronlund
Mastication Robots, 2010
ISBN 978-3-540-93902-3

Vol. 291. Shimon Whiteson
Adaptive Representations for Reinforcement Learning, 2010
ISBN 978-3-642-13931-4

Vol. 292. Fabrice Guillet, Gilbert Ritschard, Henri Briand, Djamel A. Zighed (Eds.)
Advances in Knowledge Discovery and Management, 2010
ISBN 978-3-642-00579-4

Vol. 293. Anthony Brabazon, Michael O'Neill, and Dietmar Maringer (Eds.)
Natural Computing in Computational Finance, 2010
ISBN 978-3-642-13949-9

Vol. 294. Manuel F.M. Barros, Jorge M.C. Guilherme, and Nuno C.G. Horta
Analog Circuits and Systems Optimization based on Evolutionary Computation Techniques, 2010
ISBN 978-3-642-12345-0

Vol. 295. Roger Lee (Ed.)
Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2010
ISBN 978-3-642-13264-3

Vol. 296. Roger Lee (Ed.)
Software Engineering Research, Management and Applications, 2010
ISBN 978-3-642-13272-8

Vol. 297. Tania Tronco (Ed.)
New Network Architectures, 2010
ISBN 978-3-642-13246-9

Vol. 298. Adam Wierzbicki
Trust and Fairness in Open, Distributed Systems, 2010
ISBN 978-3-642-13450-0

Vol. 299. Vassil Sgurev, Mincho Hadjiski, and Janusz Kacprzyk (Eds.)
Intelligent Systems: From Theory to Practice, 2010
ISBN 978-3-642-13427-2

Vol. 300. Baoding Liu (Ed.)
Uncertainty Theory, 2010
ISBN 978-3-642-13958-1

Vol. 301. Giuliano Armano, Marco de Gemmis, Giovanni Semeraro, and Eloisa Vargiu (Eds.)
Intelligent Information Access, 2010
ISBN 978-3-642-13999-4

Vol. 302. Bijaya Ketan Panigrahi, Ajith Abraham, and Swagatam Das (Eds.)
Computational Intelligence in Power Engineering, 2010
ISBN 978-3-642-14012-9

Vol. 303. Joachim Diederich, Cengiz Gunay, and James M. Hogan
Recruitment Learning, 2010
ISBN 978-3-642-14027-3

Vol. 304. Anthony Finn and Lakhmi C. Jain (Eds.)
Innovations in Defence Support Systems –1, 2010
ISBN 978-3-642-14083-9

Vol. 305. Stefania Montani and Lakhmi C. Jain (Eds.)
Successful Case-Based Reasoning Applications – 1, 2010
ISBN 978-3-642-14077-8

Vol. 306. Tru Hoang Cao
Conceptual Graphs and Fuzzy Logic, 2010
ISBN 978-3-642-14086-0

Vol. 307. Anupam Shukla, Ritu Tiwari, and Rahul Kala
Towards Hybrid and Adaptive Computing, 2010
ISBN 978-3-642-14343-4

Vol. 308. Roger Nkambou, Jacqueline Bourdeau, and Riichiro Mizoguchi (Eds.)
Advances in Intelligent Tutoring Systems, 2010
ISBN 978-3-642-14362-5

Vol. 309. Isabelle Bichindaritz, Lakhmi C. Jain, Sachin Vaidya, and Ashlesha Jain (Eds.)
Computational Intelligence in Healthcare 4, 2010
ISBN 978-3-642-14463-9

Vol. 310. Dipti Srinivasan and Lakhmi C. Jain (Eds.)
Innovations in Multi-Agent Systems and Applications – 1, 2010
ISBN 978-3-642-14434-9

Vol. 311. Juan D. Velásquez and Lakhmi C. Jain (Eds.)
Advanced Techniques in Web Intelligence – 1, 2010
ISBN 978-3-642-14460-8

Vol. 312. Patricia Melin, Janusz Kacprzyk, and Witold Pedrycz (Eds.)
Soft Computing for Recognition Based on Biometrics, 2010
ISBN 978-3-642-15110-1

Patricia Melin, Janusz Kacprzyk, and
Witold Pedrycz (Eds.)

Soft Computing for Recognition Based on Biometrics

Springer

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

Prof. Witold Pedrycz
Department of Electrical and
Computer Engineering
University of Alberta
Edmonton, Alberta
Canada T6J 2V4
E-mail: pedrycz@ece.ualberta.ca

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

ISBN 978-3-642-15110-1 e-ISBN 978-3-642-15111-8

DOI 10.1007/978-3-642-15111-8

Studies in Computational Intelligence ISSN 1860-949X

Library of Congress Control Number: 2010934862

Typeset & *Cover Design:* Scientific Publishing Services Pvt. Ltd., Chennai, India.

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

Preface

We describe in this book, bio-inspired models and applications of hybrid intelligent systems using soft computing techniques for image analysis and pattern recognition based on biometrics and other information sources. Soft Computing (SC) consists of several intelligent computing paradigms, including fuzzy logic, neural networks, and bio-inspired optimization algorithms, which can be used to produce powerful hybrid intelligent systems. The book is organized in five main parts, which contain a group of papers around a similar subject. The first part consists of papers with the main theme of classification methods and applications, which are basically papers that propose new models for classification to solve general problems and applications. The second part contains papers with the main theme of modular neural networks in pattern recognition, which are basically papers using bio-inspired techniques, like modular neural networks, for achieving pattern recognition based on biometric measures. The third part contains papers with the theme of bio-inspired optimization methods and applications to diverse problems. The fourth part contains papers that deal with general theory and algorithms of bio-inspired methods, like neural networks and evolutionary algorithms. The fifth part contains papers on computer vision applications of soft computing methods.

In the part of classification methods and applications there are 5 papers that describe different contributions on fuzzy logic and bio-inspired models with application in classification for medical images and other data. The first paper, by Carlos Alberto Reyes et al., deals with soft computing approaches to the problem of infant cry classification with diagnostic purposes. The second paper, by Pilar Gomez et al., deals with neural networks and SVM-based classification of leukocytes using the morphological pattern spectrum. The third paper, by Eduardo Ramirez et al., describes a hybrid system for cardiac arrhythmia classification with fuzzy K-Nearest Neighbors and neural networks combined by a fuzzy inference system. The fourth paper, by Christian Romero et al., offers a comparative study of blog comments spam filtering with machine learning techniques. The fifth paper, by Victor Sosa et al., describes a distributed implementation of an intelligent data classifier.

In the part of pattern recognition there are 6 papers that describe different contributions on achieving pattern recognition using hybrid intelligent systems based on biometric measures. The first paper, by Daniela Sanchez et al., describes a genetic algorithm for optimization of modular neural networks with fuzzy logic integration for face, ear and iris recognition. The second paper, by Denisse Hidalgo et al., deals with modular neural networks with type-2 fuzzy logic response integration for human recognition based on face, voice and fingerprint. The third paper, by Lizette Gutierrez et al., proposes an intelligent hybrid system for person

identification using the ear biometric measure and modular neural networks with fuzzy integration of responses. The fourth paper, by Luis Gaxiola et al., describes the modular neural networks with fuzzy integration for human recognition based on the iris biometric measure. The fifth paper, by Juan Carlos Vazquez et al., proposes a real time face identification using a neural network approach. The sixth paper, by Miguel Lopez et al., describes a comparative study of feature extraction methods of type-1 and type-2 fuzzy logic for pattern recognition systems based on the mean pixels.

In the part of optimization methods there are 6 papers that describe different contributions of new algorithms for optimization and their application to real world problems. The first paper by Marco Aurelio Sotelo-Figueroa et al., describes the application of the bee swarm optimization BSO to the knapsack problem. The second paper, by Jose A. Ruz-Hernandez et al., deals with an approach based on neural networks for gas lift optimization. The third paper, by Fevrier Valdez et al., describes a new evolutionary method combining particle swarm optimization and genetic algorithms using fuzzy logic. The fourth paper by Claudia Gómez Santillán et al., describes a local survival rule for steer an adaptive ant-colony algorithm in complex systems. The fifth paper by Francisco Eduardo Gosch Ingram et al., describes the use of consecutive swaps to explore the insertion neighborhood in tabu search solution of the linear ordering problem. The sixth paper by Leslie Astudillo et al., describes a new optimization method based on a paradigm inspired by nature.

In the part of theory and algorithms several contributions are described on the development of new theoretical concepts and algorithms relevant to pattern recognition and optimization. The first paper, by Jose Parra et al., describes an improvement of the backpropagation algorithm using (1+1) Evolutionary Strategies. The second paper, by Martha Cardenas et al., describes parallel genetic algorithms for architecture optimization of neural networks for pattern recognition. The third paper, by Mario Chacon et al., deals with scene recognition based on fusion of color and corner features. The fourth paper, by Hector Fraire et al., describes an improved tabu solution for the robust capacitated international sourcing problem. The fifth paper, by Martin Carpio et al., describes variable length number chains generation without repetitions. The sixth paper, by Juan Javier González-Barbosa et al., describes a comparative analysis of hybrid techniques for an ant colony system algorithm applied to solve a real-world transportation problem.

In the part of computer vision applications several contributions on applying soft computing techniques for achieving artificial vision in different areas are presented. The first paper, by Olivia Mendoza et al., describes a comparison of fuzzy edge detectors based on the image recognition rate as performance index calculated with neural networks. The second paper, by Roberto Sepulveda et al., proposes an intelligent method for contrast enhancement in digital video. The third paper, by Oscar Montiel et al., describes a method for obstacle detection and map reconfiguration in wheeled mobile robotics. The fourth paper, by Pablo Rivas et al., describes a method for automatic dust storm detection based on supervised classification of multispectral data.

In conclusion, the edited book comprises papers on diverse aspects of bio-inspired models, soft computing and hybrid intelligent systems. There are theoretical spects as well as application papers.

May 31, 2010

Patricia Melin, Tijuana Institute of Technology, Mexico
Janusz Kacprzyk, Polish Academy of Sciences, Poland
Witold Pedrycz, University of Alberta, Canada

Contents

Soft Computing Approaches to the Problem of Infant Cry Classification with Diagnostic Purposes

Carlos A. Reyes-Garcia[1], Orion F. Reyes-Galaviz[2], Sergio D. Cano-Ortiz[3], Daniel I. Escobedo-Becerro[3], Ramón Zatarain[4], and Lucia Barrón-Estrada[4]

[1] Instituto Nacional de Astrofisica Optica y Electronica (INAOE)
[2] Instituto Tecnologico de Apizaco
[3] Universidad de Oriente
[4] Instituto Tecnológico de Culiacán
kargaxxi@inaoep.mx, orionfrg@yahoo.com, scano@fie.uo.edu.cu, rzatarain@itculiacan.edu.mx

Abstract. Although the scientific field known as infant cry analysis is close to celebrate its 50 anniversary, considering the Scandinavian experience as the starting point, until now none reliable cry-based clinical routines for diagnosis has been successfully achieved. Nevertheless in support of that goal some expectations are appearing when new automatic infant cry classification approaches displaying potentialities for diagnosis purposes are added to the traditional perceptive approach and direct spectrogram observation practice. In this paper we present some of those classification approaches and analyze their potentials for newborn pathologies diagnosis as well. Here we describe some classifiers based on soft computing methodologies, among them; one following the genetic-neural approach, an experimental essay with a hybrid classifier combining the traditional approach based on threshold classification and the classification approach with ANN, one more applying type-2 fuzzy sets for pattern matching, and one using fuzzy relational products to compress the crying patterns before classification. Experiments and some results are also presented.

1 Introduction

For several decades the acoustic analysis of infant crying and their vocalizations has been led to the identification and to help diagnosis of pathologies supported by the study of the behavior and knowledge of the variations that occur in the production of the sound of infant crying. Many works have appeared reporting the linkage of age, identity and relevant information found in a number of parameters of these cries with the neurophysiological status of newborns [1-12]. In fact the diagnosis potential of infant cry analysis for various pathological conditions in the neonate has been demonstrated [1-2] [4-10] [13-16].

In this process several processing alternatives have been applied to the acoustic analysis of infant crying such as: auditory analysis, analysis tempo-frequencial of

P. Melin et al. (Eds.): Soft Comp. for Recogn. Based on Biometrics, SCI 312, pp. 3–18.
springerlink.com

the crying signal, spectrographic analysis, digital signal processing (DSP) techniques, all of them potentiated by the rise and development of computers and new information technologies. To the classical approach of infant cry analysis (ICA) to extract relevant information from crying of a diagnostic value according to the threshold behavior of acoustic parameters [4] [8] [10] [14-16], we recently added approaches like logical-combinatorial, connectionist, genetic-neural, type-2 fuzzy sets and other hybrid systems. [28-34]

2 The Infant Cry Automatic Recognition Process

The infant cry automatic classification process is, in general, a pattern recognition problem, similar to Automatic Speech Recognition (ASR). The goal is to take the wave from the infant's cry as the input pattern, and at the end obtain the kind of cry or pathology detected on the baby [32], [33]. Generally, the process of Automatic Cry Recognition is done in two steps. The first step is known as signal processing, or feature extraction, whereas the second is known as pattern classification. In the acoustical analysis phase, the cry signal is first normalized and cleaned, and then it is analyzed to extract the most important characteristics in function of time. Some of the more used techniques for the processing of the signals are those to extract: pitch, intensity, spectral analysis, linear prediction coefficients (LPC), Mel frequency cepstral coefficients (MFCC), cochleograms, etc. The set of obtained characteristics is represented by a vector, which, for the process purposes, represents a pattern. The set of all vectors is then used to train the classifier. Later on, a set of unknown feature vectors is compared with the knowledge that the computer has to measure the classification output efficiency. Figure 1 shows the different stages of the described recognition process.

In this paper we will not describe the complete acoustic analysis process, instead we recommend the interested readers to consult [26] [28] [29] y [32-34]. The rest of the paper will be devoted to the description of some models applied in the pattern recognition phase.

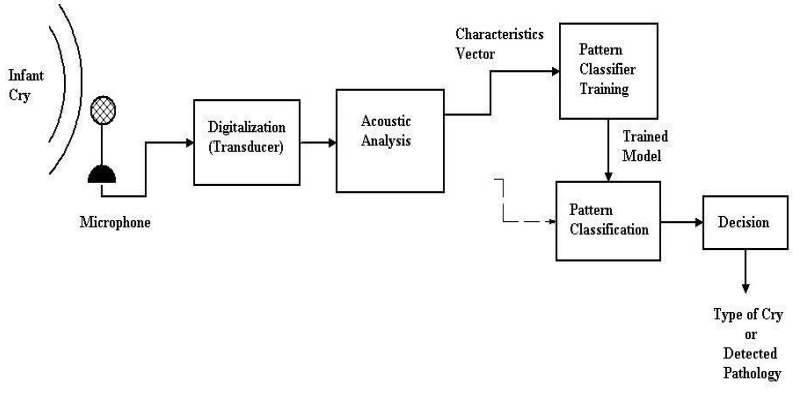


Fig. 1. Automatic Infant Cry Recognition Process

3 Logical-Combinatorial Approach

This is related to the logical-combinatorial approach of Pattern Recognition whose essential idea is to establish the analogy, in which an object may resemble another, but it might not be in its entirety, and the parts that look alike can provide information about possible regularities between objects.

This approach is an alternative to the statistical approach, regularly applied in medical investigations. It allows the appropriate treatment to the characteristics of little formalized sciences, where specialists seldom have a single explanation to their conclusions, and where in the description of objects are present both, qualitative and quantitative variables, or where often occur objects of which there is no information on some of their descriptive characteristics.

The classification by learning applies to problems where there are two or more classes of objects -- of any kind - and a group of them is known which respectively belong to these classes. The model of voting algorithms is a partial precedence algorithm, which analyzes the accumulated experience. A key feature is the opportunity to analyze and reach conclusions on the problem from different viewpoints. This model is described by the following steps:

1. Establishment of the system of support sets.
2. Similarity function.
3. Evaluation by row given a set of fixed support.
4. Evaluation by class given a set of fixed support.
5. Evaluation by class for the whole system of support sets.
6. Solution Rule

Applying the voting algorithms model implicitly entails the analysis by parts of the model being evaluated. This is a useful feature that allows weighting the analysis by different criteria in problems that can be broken down and analyzed taking different sub-descriptions and evaluation criteria. One advantage of applying this paradigm to science little formalized such as medicine, is that it lets to analyze qualitative and quantitative variables, assuming no information at all. This model of voting algorithms was used in classification of infant crying with good results [26].

4 The Connectionist Approach

These kind of methods are known as connectionist models or Artificial Neural Networks (ANN), due to the resemblance its processing has with the form of processing of the human nervous system. They are essential parts of an emerging field of knowledge known as Computational Intelligence.

The use of connectionist models has provided a solid step forward in solving some of the more complex problems in Artificial Intelligence (AI), including such areas as machine vision, pattern recognition, speech recognition and speech synthesis. The research in this field has focused on the evaluation of new neural networks for pattern recognition, training algorithms using real speech data, and

whether parallel architectures of neural networks can be designed to perform effectively the work required for complex algorithms for the recognition of crying [5]. This approach has been used in the classification of infant crying under several scenarios: use of supervised Feed Forward networks (Petroni 1995, Cano et al 2000, Reyes Garcia 2000, 2002), classifying with Kohonen's self-organizing maps (Schonweiller 1996, Cano et al 1998).

5 Genetic-Neural Approach

This approach is a recent hybrid alternative, where evolving models are applied to select the best features of the crying input vectors, which then are used to train a classification system based on neural networks [35]. To make that selection, Evolution Strategy (ES) techniques are applied. These techniques are similar to genetic algorithms (GA) but the principal difference is that GA use both crossover and mutation whereas ES uses only mutation. In addition, when an evolution strategy is used there is no need to represent the problem in a coded form, and real numbers can be used for the representation of individuals. In our application the system works as follows: We start with a p x q size array, where p is the number of acoustic characteristics that each sample has, and q is the number of samples that exist. This p x q matrix is to be reduced to an m x q matrix, where m is the number of features selected, and m < p. This reduction is carried out in the following way; there is a population of n individuals, where each individual has a length m; each of these individuals represents n arrays of m x q size, as shown in Figure 2.

Once obtained the matrices, *n* neural networks are initialized, and each one is trained with one of the matrices, at the end of each training process the efficiency of the neural network is tested by means of confusion matrices. With these data, we select the *n*/2 matrices that gave the best results as illustrated in Figure 3.

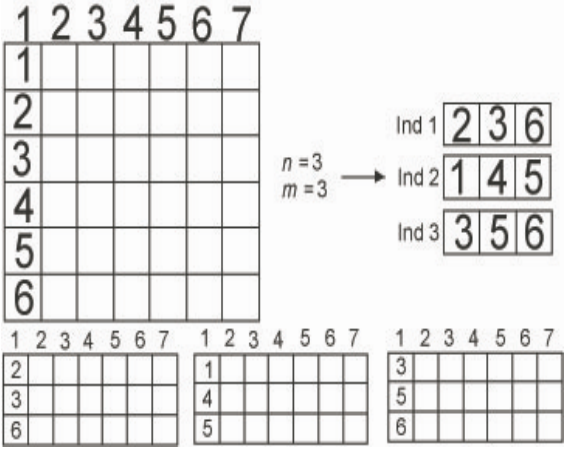


Fig. 2. Initializing Individuals

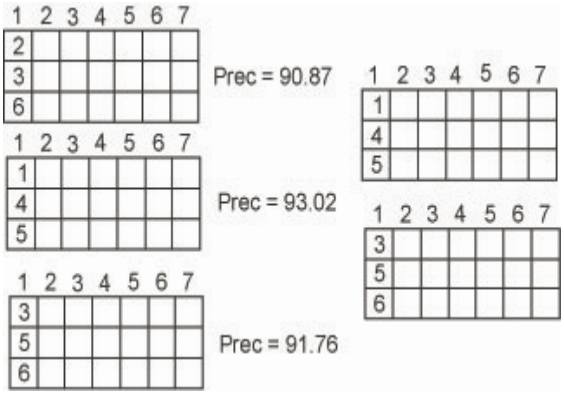


Fig. 3. Selecting the best individuals

When the best arrays are selected a tournament is applied, where l random numbers are generated ranging from 0 to the number of arrays selected (n / 2), as shown in the Figure 4, where 2 arrays of 4 were selected, then 4 random numbers from 0 to 2 are generated. It is significant to remark that the number 1 has twice the probability to be randomly generated, since when the random number is 0, it automatically becomes 1, that will be seen as a reward to the best place, along with a greater chance of being selected.

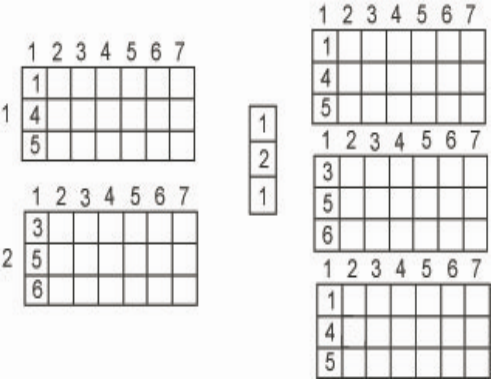


Fig. 4. Generating the new population with the best individuals

Once the new population of individuals is generated, they undergo a random mutation, for each epoch a mutation factor (*MF*) is generated, for each individual, a random number between 0 and 1 is next generated, if it is less than *MF* the individual is mutated, if it is greater or equal, passes to the next generation exactly the same. When an individual is selected to be mutated, generates a random number between 1 and m, that is to select which gene will be mutated. Once done, it

generates a random number between 1 and p, which is used to select a new feature from the original vectors.

It is worth mentioning that a feature can be selected twice in the same individual, since the algorithm does not verify if this feature already exists within the genetic information of individual. If the individual with repeated features is efficient, it means that this feature is essential and important for optimal recognition of the crying samples.

The designer has the option to choose as the stopping criterion, which, in this case, is the number of generations that will perform the system (r). At the end of r generations, we get the individual who got the best overall result, and the best individuals are shown in each of the r generations. With this we know which the best features are to be selected for robust recognition. It should be mentioned that this classification system is of the wrapping type, for which, once selected the best characteristics, through evolutionary strategies, we must train the system with them, looking for the best classification results [25].

In order to compare the behavior of our proposed hybrid system, we made a set of experiments where the original input vectors were reduced to 50 components by means of Principal Component Analysis (PCA). When we use evolutionary strategies for the acoustic features selection, we search for the best 50 features. By this way, the neural network's architecture consists of a 50 nodes input layer, a 20 nodes hidden layer (60% less nodes than the input layer) and an output layer of 3 nodes. The implemented system is interactively adaptable; no changes have to be made to the source code to experiment with any corpuses. Also, for these experiments, we have a corpus made out of 1049 samples from normal babies, 879 from hypo acoustics (deaf), and 340 with asphyxia, all this from one second segments samples. On the next step the samples are processed individually by extracting its MFCC features, this process is done with the freeware program Praat 4.2. The acoustic features are extracted as follows: for each segment we extract 16 coefficients for every 50 or 100 milliseconds, generating vectors that go from 145 to 304 features for each one second sample. The training is done up to 6000 epochs or until a 1×10^{-8} error is reached. Once the network is trained, we test it using different samples from each class separated previously for this purpose (we used from each corpus 70% for training and 30% for testing). The recognition results with the best configuration of acoustic features are shown in Table 1.

Table 1. Results using different feature extractions, comparing a simple neural network with a hybrid system

	Neural System	Hybrid System
1 sec. MFCC 16 feat 50ms	89.79%	95.40%
1 sec. MFCC 16 feat 100ms	93.33%	96.76%

6 Development of a Hybrid Classifier

In order to show the potential of a hybrid approach Specialists of the Voice Processing Group of the Universidad de Oriente in Santiago de Cuba in collaboration with the Soft Computing Group of INAOE Puebla (Mexico) implemented and tested in [27] [35] a hybrid classifier in which two approaches were combined: the traditional approach based on threshold classification and the classification approach with ANN (with Cepstral Coefficients in the scale of MEL (MFCC's) as attributes). The test took place for a primary sample taken from the BDLLanto database (32 cases: 16 from healthy neonates and 16 from pathological neonates) which were segmented into 73 units of healthy crying and 68 units of pathological crying (related to hypoxia) from which 58 crying units (by class) were selected for the training phase and 10 for classification.

The hybrid classifier corresponds to the block diagram shown in Figure 5.

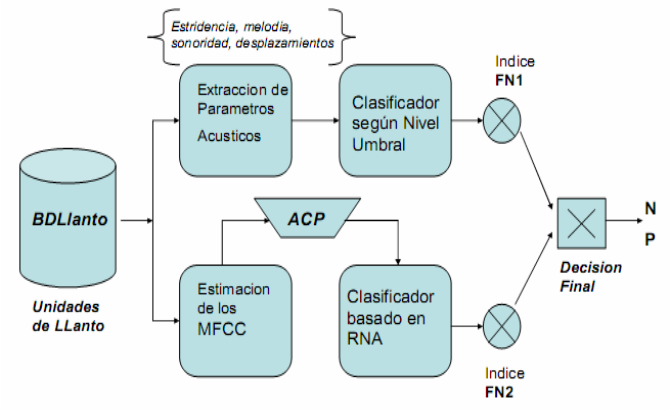


Fig. 5. Block Diagram of the infant crying hybrid classifier

The classifier calculates a normal *FN1* subscript (as the threshold criterion for the 4 acoustic attributes: loudness, stridency, displacement of the fundamental tone and melodic pattern) and a normal *FN2* subscript (according to the classification criteria of the connectionist model trained with MFCC's), obtaining finally a *D* index (average of both normal indices) that decide the membership of the 2 crying unit classes (normal and pathological) under study. The grading criteria for the *D* index was:

- Normal $D <= 0{,}5$
- Moderately pathological $D = 0{,}75$
- Pathological $D = 1{,}0$

6.1 Classification Results Analysis

Classification results are shown in tables 2, 3, 4 and 5:

Table 2. Classification results according to the threshold level

Total cases by class		**FN1 Index**				
		0	**0.25**	**0.5**	**0.75**	**1.0**
Normal	10	9	1	-	-	-
Pathologic	10	-	2	5	2	1

Table 3. Altering frequency for parameters in both classes (N/P)

	Altered Parameter			
	Stridence	**Sonority**	**Melody**	**F_0 Displacement**
Normal (10)	-	-	-	1
Pathologic (10)	6	5	5	6

Table 4. Classification results with an ANN

Type of Cry	**Samples**	**Confusion Matrix**		**Classification**
		Normal	*Pathologic*	
Normal	10	7	3	70%
Pathologic	10	2	8	80%
Total	20			75%

Table 5. Classification results with the proposed hybrid model

		Confusion Matrix		***D* Index**			**Classification**
		Normal	*Pathologic*	*x<=0.5*	*0.5<=x<=0.75*	*0.75<=x*	*%*
Normal	10	10	0	10		0	100
Pathologic	10	2	8	2	7	1	80
Total	20			12	7	1	90

For the analysis of the results we can observe that:

- The hybrid classifier performance is superior to the classification rates of similar systems reported in the literature.
- The gradation in levels of index *D* allows the physician or neonatologist the proper use of the output of the classifier to compare and evaluate the possible meanings attached to that output, in the face of neurophysiological evaluation of the multidisciplinary team that evaluates the infant (eg. how abnormal is the acoustics of that crying and its possible diagnostic value)

- The need to incorporate a greater number of relevant acoustic attributes to be considered by the infant cry classifier, reported by Shonweiller et al in [19], is widely satisfied in this experience. It is evident that the combination of attributes increases recognition rates with respect to experiments with a single attribute.
- An interesting aspect to remark is the fact that the 2 crying units bad classified erroneously as normal (see Table 4) had precisely FN1 rates of 0.75 (which implied a significant abnormality on threshold-based classifier), demonstrating the diagnostic validity that may still have each independent classifier.

7 Statistic Measures for Reducing Input Vectors

In [28] we presented a work titled "Statistical Vectors of Acoustic Features for the Automatic Classification of Infant Cry" where, in order to improve processing time, the original acoustical data vectors are reduced. An associated objective of data reduction is to preserve the most relevant information, in such a way that the resulting data are the most representative of the original ones. In this sense, statistical operations as minimum, maximum, average, standard deviation and variance, are operations that when applied on a data set the result is only one representative global value from each operation. Each operation by itself is not able to represent all the data set. Nevertheless, their combination allows obtaining a global representation of the data vectors. The reduction is carried out by means of five statistical operations, significantly reducing the size of the vectors from 304 or more MFCC or LPC attributes to only 5 statistical characteristics. Once the reduced matrices are generated, 3 groups of data are formed in the following way: 200 and 340 statistical vectors of each type of cry in a random way were selected, forming the groups A and B respectively. Another group C was formed by means of the random selection of 200 vectors of each type of cry without data reduction.

7.1 Experimental Tests and Results

The results when using several different single classifiers are presented in Table 6.

Table 6. Results using a single classifier.

Classifier	Precision data set A	Precision data set B	Precision data set C
N. Bayes	87.67%	88.43%	85.83%
SVM	89.67%	90.78%	**91.67%**
Neural N.	**91.67%**	**91.86%**	90.83%
R. Forest	90.3%	91.37%	89%
J48	89%	90.88%	83.5%

Table 7 allows us to compare the classifiers and ensembles that obtained less classification error by groups of data. It is possible to observe in the table that, almost all the ensembles include the classifier that individually obtained the best results.

Table 7. Best classifiers and ensembles by precise classification.

	A	B	C
Classifier	Neural N; **91.67%**	Neural N; **91.86%**	SMO; **91.67%**
Ensemble	Staking: Neural N, SMO, R. Forest **91.83%**	Vote: Neural N, R. Forest; **93.23%**	Staking: SMO, J48 Vote: SMO, R. Forest**91.66%**

8 The Fuzzy Approach

A work titled "Type-2 Fuzzy Sets Applied to Pattern Matching for the Classification of Cries of Infants under Neurological Risk" was presented in [29] consisting in a pattern recognition algorithm for the classification of infant cries.

Type-1 fuzzy sets are not able to directly model some kinds of uncertainties because their membership functions are totally crisp. On the other hand, type-2 fuzzy sets are able to model such uncertainties because their membership functions are themselves fuzzy. Membership functions of type-1 fuzzy sets are two-dimensional, whereas membership functions of type-2 fuzzy sets are three-dimensional. Trying to capture the uncertainties present in the infant cry signal we decided to apply pattern matching with type-2 fuzzy sets. For the experiments, we used four different acoustic features; Intensity, Cochleogram, LPC, and MFCC. For the extraction of the two last features we applied 50 ms windows (Hamming), in each of which we obtained 16 coefficients. In this way we obtained feature vectors for the corresponding characteristic with 19 values for Intensity, 304 for LPC and MFCC and of 510 for Cochleogram values for each one second segment sample.

Once the feature vectors of each class are obtained, we proceed to the infant cry recognition and classification phase. For this task we applied the Fuzzy Pattern Matching approach modified to the use of type 2 fuzzy sets (T2-FPM). The algorithm is divided in two parts, the learning one, where primary and secondary membership information on the classes is collected, the membership of each element to each class is calculated, and a decision to which class each element belongs to is taken. In the classification phase an element (an unknown feature vector) is received, from which the membership to each class is obtained.

8.1 Results

The classifier was tested using the method of 10-fold cross validation, which consists of dividing in 10 parts the testing set, and testing the classifier with each one. Nine subsets are used for training and one for testing. This process is repeated 10 times using a different test set each time. The dataset used to test the classifier

Table 8. Experiments to classify three classes: asphyxia, normal and hyperbilirubinemia

Feature(s)	Mean	
	Precision	Recall
Cochleograms	85,00%	85,73%
Intensity	61,31%	57,63%
LPC	**89.15**	**92.70**
MFCC	78,73%	85,60%
LPC-Cochleograms	**91.74%**	**92.53%**
LPC-Cochleograms-MFCC	84,67%	82,83%
LPC-MFCC-Cochleograms- Intensity	**89,85%**	**93,92%**

contains: 400 patterns for class "normal," 340 patterns for class "asphyxia" and 418 for class "hyperbilirubinemia". Each pattern contains four feature vectors: LPC (304 elements), MFCC (304 elements), Intensity (19 elements) and Cochleogram (510 elements). Different combinations of these vectors were used for testing the classifier, in order to find out the best features to discriminate among asphyxia and hyper-bilirrubinemia. For example, for the test case using the four feature vectors, the classifier gets an input vector with 1,137 attributes. Some of the most relevant results are shown at Table 8.

For three-class classification, the best results were obtained using the combination LPC-Cochleograms. Acceptable results were also obtained when all four feature vectors were used (LPC, MFCC, Cochleograms and Intensity). The fact that two feature vectors perform better than four may be explained because intensity proved to be bad discriminator when used alone (see Table 7).

9 Compressing the Cry Features

Very recently we tried to classify infant cry by compressing the original signal, instead of reducing the vectors once they were analyzed. This experience was reported in [30] with the title "Fuzzy Relational Compression Applied on Feature Vectors for Infant Cry Recognition", in which the reduction method uses Fuzzy Relational Product (FRP) to compresses the information inside a feature vector, building with this a compressed matrix that will help us recognize two kinds of pathologies in infants; Asphyxia and Deafness. This algorithm uses codebooks to build a small relational matrix that represents an original vector. Since this algorithm was firstly designed to compress and decompress images, the resulting compressed matrix, along with the codebooks should hold enough information to build a lossy representation of the original image.

The mathematical relations are another kind of fuzzy relational operations, and their properties can be applied to crisp or fuzzy matrices as follows. Let R be a relation from X to Y, and S a relation from Y to Z, furthermore let be $X = \{x_1; x_2; \ldots; x_n\}$, $Y = \{y_1; y_2; \ldots; y_n\}$, and $Z = \{z_1; z_2; \ldots; z_n\}$ finite sets, there can be many binary operations applied on them, each one resulting in a product relation from set X to set Z, operations such as: Circlet Product ($R \circ S$), where x has a relation $R \circ S$ to z, if and only if there is at least one y such that xRy and ySz:

$$x(R \circ S)z <=> \exists\, y \in Y \text{ if } (xRy \text{ and } ySz)$$

Then the circlet relation $x(R \circ S)z$ exists *if and only if* there is a path from x to z:

$$(R \circ S)xz = \max[\min(Rxy;\, Syz)] = \cap\, (Rxy \cap Syz);$$

The algorithm was firstly proposed for lossy compression and reconstruction of an image by Hirota and Pedrycz [31], where a still gray scale image is expressed as a fuzzy relation by normalizing the intensity range of each pixel from [0; 255] onto [0; 1]. In our case, a feature vector that holds the information of an infant cry sample is also normalized onto values between [0; 1], transformed into a matrix R (Figure 6), and then compressed into;

$$G \in F(I \times J)$$

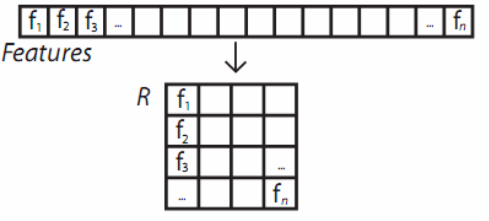


Fig. 6. Feature vector normalized and transformed into Matrix R

The whole compression process is visually described in Figure 7.

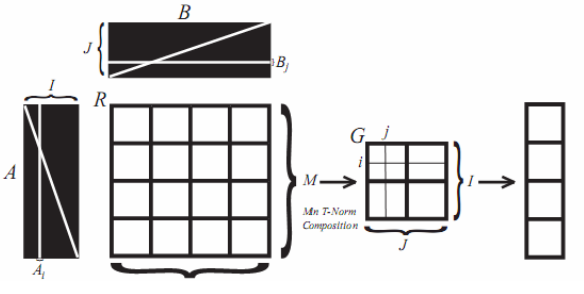


Fig. 7. Fuzzy Relational Feature compression

Where A and B are codebooks, each of which is an essential process in the matrix compression. Given a codebook, each block of the matrix can be represented by the binary address of its closest codebook vector. Such a strategy results in significant reduction of the information involved on the matrix transmission and storage. To implement the fuzzy relational compressor, the first step is to program an algorithm that multiplies two matrices of sizes $R^T = N \times M$ and $A = M \times I$. The result must be a matrix of size $Q = N \times I$, which is then transposed $Q^T = I \times N$ and

multiply it by $B = N \times J$, the resulting matrix must have a size $G = I \times J$. More details of this process can be seen in [30].

9.1 Implementation and Experiments

For the infant cry classification we used a Time Delay Neural Network (TDNN). For the reported experiments, we used 1049 samples from normal babies, 879 from hypoacoustics (deaf), and 340 with asphyxia; all samples are 1 second segments. Next, the samples are processed to extract the MFCC acoustic features. In these experiments, and since there are only 340 samples contained in the Asphyxia class, 340 samples are randomly selected from the Normal and Deaf classes respectively, for a total of 1020 vectors; as a result we have a Training matrix of size (361+1) × 714 (70% from each class) and a Testing matrix of size (361+1) × 306 (the remaining 30%).

After the training and testing matrices have been compressed and rebuilt the input vectors were reinforced by adding a vector extracted from the original uncompressed matrices; for each input vector the following statistical analysis were obtained: *maximum, minimum, standard deviation, mean,* and *median* values. These vectors were concatenated to the bottom of their corresponding previously compressed samples, giving us vectors of size (30+1) × 1. With these vectors, the final compressed training and testing matrices result in a size of (30+1) × 714 and (30+1) × 306 respectively.

In order to validate and compare the behavior of the proposed fuzzy relational compression, a set of experiments were made:

1. Original vectors without any dimensionality reduction,
2. Vectors reduced to 50 Principal Components with PCA,
3. Vectors reduced to 25 components with FRP,
4. Vectors reduced to 5 components; (max; min; std; mean; median), and
5. Reinforced vectors with 30 components (rFRP)

Each experiment was performed five times; the average results are shown on Table 9.

Table 9. Average experimental results comparing different compression methods

Experiment No.	Compression Method	Performance
1.	No Compression	96.55%
2.	PCA	97.33%
3.	FRP	64.63%
4.	Statistics	59.39%
5.	rFRP	96.44%

The feature compression using fuzzy relational product, and reinforcing it with a statistical analysis, becomes a simple implementation process and has satisfactory results, proving its effectiveness; given that the resulting matrices, which

were compressed to almost 10% of their original size, yielded good results when classifying three different infant cry classes.

10 Conclusions

This paper addresses the issue of hybrid infant crying classifiers oriented to the neonatal diagnosis which is a current trend in the infant cry analysis field.

It demonstrates the importance which has the increase in the number of acoustic attributes with diagnostic potential as well as the need to deepen into the combined effect it may have in order to improve performance rates of infant cry classifiers.

11 Recommendations

In the future we propose the incorporation of other approaches such as neuro-genetic classification and the combination of features into a single vector to improve the representation of crying. This combination is supported by the hypothesis that when the features based in predicting the signal, and those based on perception are combined, a representation with attributes complementing among themselves will be obtained, and consequently the final recognition rate should be improved.

References

1. Karelitz, S., Fisichelli, V.R.: The cry thresholds of normal infants and those with brain damage. J. Pediat. 61, 679–685 (1962)
2. Parmelee, A.H.: Infant crying and neurologic diagnosis. J. Pediat. 61, 801–802 (1962)
3. Wasz-Höckert, O., Lind, J., Vuorenkoski, V., Partanen, T., Valanne, E.: The infant cry a spectrographic and auditory analysis. Clinics in Devel. Medicine, 29 (1968)
4. Koivisto, M., Wasz-Höckert, O., Vuorenkoski, V., Partanen, T., Lind, J.: Cry studies in neonatal hyperbilirubinaemia. Acta Paediatr. Scand. Suppl. 206, 26–27 (1970)
5. Wasz-öckert, O., Koivisto, M., Vuorenkoski, V., Partanen, T., Lind, J.: Spectrographic analysis of pain cry in hyperbilirubinemia. Biol. Neonate. 17, 260–271 (1971)
6. Michelsson, K.: Sound Spectrographic cry analysyis of normal and abnormal newborns. Folia Phoniatria 28, 161–173 (1982)
7. Michelsson, K.: Cry analysis of symptomless low birth weight neonates and of asphyxiated newborn infants. Acta Paediatrica Scandinavica. Stockholm. Suppl. 216 (1971)
8. Michelsson, K., Sirviö, P., Wasz-Höckert, O.: Pain cry in full-term asphyxiated newborn infants correlated with late findings. Acta Paediatr. Scand. 66(5), 611–616 (1977)
9. Sirviö, P., Michelsson, K.: Sound-spectrographic cry analysis of normal and abnormal newborn infants. Folia phoniat. 28, 161–173 (1976)
10. Michelsson, K., Wasz-Höckert, O.: The value of cry analysis in neonatology and early infancy. In: Murry, T., Murray, J. (eds.) Infant Communication: Cry and Early Speech, pp. 152–182. College-Hill Press, Houston (1980)

11. Wasz-Höckert, O., Michelsson, K., Lind, J.: Twenty-five years of Scandinavian cry research. In: Lester, B.M., Boukydis, C.F.Z. (eds.) Infant Crying: Theoretical and Research Perspectives, pp. 83–104. Plenum Publishing Corporation, N. York (1985)
12. Golub, H.L., Corwin, M.J.: A physioacoustic model of the infant cry. In: Lester, B.M., Boukydis, C.F.Z. (eds.) Infant Crying: Theoretical and Research Perspectives, pp. 59–82. Plenum Press, N. York (1985)
13. Golub, H.L., Corwin, M.J.: Infant cry: a clue to diagnosis. Pediatrics 69(2), 197–201 (1982)
14. Lester, B.M.: A biosocial model of infant crying. In: Lipsitt, L.P. (ed.) Advances in Infancy Research, pp. 167–212. Academic Press, N. York (1984)
15. Rapisardi, G., Vohr, B., Cashore, W., Peucker, M., Lester, B.M.: Assessment of infant cry variability in high-risk infants. International Journal of Pediatric Otorhinolaryngology 17, 19–29 (1989)
16. Pinyerd, B.J.: Infant cries: physiology and assessment. Neonatal Network 13(4), 15–20 (1994)
17. Cano, S., et al.: The Spectral Analysis of Infant Cry: An Initial Approximation. In: Proceedings of EUROSPEECH 1995 (sponsored by ESCA & IEEE), Madrid, September 18-21 (1995)
18. Petroni, M., Malowany, A., Johnston, C., Stevens, B.: Identification of Pain from Infant Cry Vocalizations Using Artificial Neural Networks (ANNs). In: The International Infant Cry Research Group. Applications and Science of Artificial Neural Networks. The International Society for Optical Engineering, vol. 2492, pp. 729–738 (1995)
19. Schonweiler, R., et al.: Neural networks and self-organizing maps: new computer techniques in the acoustic evaluation of the infant cry. Int. Journal of Pediatric Otorhinolaryngology 38, 1–11 (1996)
20. Suaste, I., Reyes-Galaviz, O., Dias, A., Reyes, C.: A Fuzzy Relational Neural Network for Pattern Classification. In: Sanfeliu, A., et al. (eds.) CIARP 2004. LNCS, vol. 3287, pp. 358–365. Springer, Heidelberg (2004)
21. Alonso, L., et al.: Reconocimiento de Patrones con Redes Neuronales. In: Alonso, L. (ed.) Editorial Imprenta Catedral, pp. 37–56 (2001)
22. Orozco, J., Reyes, C.: Extracción y Análisis de Características Acústicas del Llanto de Bebés para su Reconocimiento Automático Basado en Redes Neuronales. Tesis de Maestría, INAOE, Puebla, Mex (2002)
23. Cano, D., et al.: Análisis preliminar de los resultados de una clasificación de unidades de llanto según tres arquitecturas de redes neuronales. In: Memorias de TELEC 2002 (en CD'Rom). Centro de Convenciones HEREDIA (Julio 2002)
24. Moller, S.: A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. Neural Networks 6(4), 525–533 (1993)
25. Reyes, O.F., Cano, O.S.D., Reyes, C.A.: Validation of the Cry Unit As Primary Element for Cry Analysis Using An Evolutionary-Neural Approach. In: 9no Encuentro Internacional Mexicano de Ciencias de la Computacion ENC 2008, Universidad Autonoma de Baja California, Mexicali, Mexico (October 2008)
26. Diez, R.H., Torres, M., Escobedo, B.D.I., Cano, O.S.D., Regüeiferos, P.L., Capdevila, B.L.: Una Aproximación al Diagnostico Neonatal Basado en Analisis de Llanto con Clasificacion Supervisada. In: Proceedings of the VI SICS, Editorial Oriente. Santiago de Cuba, pp. 1404–1409 (1999)

27. Cano, O.S.D., Escobedo, D.I., Suaste, I., Ekkel, T., Reyes Garcia, C.A.: A Combined Classifier of Cry Units with New Acoustic Attributes. In: Martínez-Trinidad, J.F., Carrasco Ochoa, J.A., Kittler, J. (eds.) CIARP 2006. LNCS, vol. 4225, pp. 416–425. Springer, Heidelberg (2006)
28. Amaro-Camargo, E., Reyes-García, C.A., Arch-Tirado, E., Mandujano, M.: Statistical Vectors of Acoustic Features for the Automatic Classification of Infant Cry. The International Journal of Information Acquisition 4(4), 347–355 (2007)
29. Santiago-Sánchez, K., García, C.A.R., Gómez-Gil, P.: Type-2 Fuzzy Sets Applied to Pattern Matching for the Classification of Cries of Infants under Neurological Risk. In: Huang, D.-S., et al. (eds.) ICIC 2009. LNCS, vol. 5754, pp. 201–210. Springer, Heidelberg (2009)
30. Galaviz, O.F.R., Garcia, C.A.R.: Fuzzy Relational Compression Applied on Feature Vectors for Infant Cry Recognition. In: Aguirre, A.H., Borja, R.M., Garcia, C.A.R. (eds.). LNCS (LNAI), vol. 5845, pp. 420–431. Springer, Heidelberg (2009)
31. Hirota, W.P.: Fuzzy Relational Compression. IEEE Transactions os Systems, man, and cybernetics, Part B: Cybernetics 29(3), 1–9 (1999)
32. Cano, S.D., Escobedo, D.I., Coello, E.: El Uso de los Mapas Auto-Organizados de Kohonen en la Clasificación de Unidades de Llanto Infantil. In: Grupo de Procesamiento de Voz, 1er Taller AIRENE, Universidad Catolica del Norte, Chile, pp. 24–29 (1999)
33. Ekkel, T.: Neural Network-Based Classification of Cries from Infants Suffering from Hypoxia-Related CNS Damage. Master Thesis. University of Twente, The Netherlands (2002)
34. Galaviz, O.F.R., García, C.A.R.: Infant Cry Classification to Identify Hypo Acoustics and Asphyxia comparing an Evolutionary-Neural System with a Neural Network System. In: Gelbukh, A., de Albornoz, Á., Terashima-Marín, H. (eds.) MICAI 2005. LNCS (LNAI), vol. 3789, pp. 949–958. Springer, Heidelberg (2005)
35. Reyes-García, C.A., Zatarain, R., Barron, L., Reyes-Galaviz, O.F.: A Hybrid System for Automatic Infant Cry Recognition I. In: Rabuñal, J.R., Dorado, J., Pazos, A. (eds.) Encyclopedia of Artificial Intelligence, Hershey, PA, USA. Information Science Reference (an imprint of IGI Global), pp. 860–866 (2008), ISBN 978-1-59904-849-9 (hardcover) – ISBN 978-1-59904-850-5 (ebook)

Neural Networks and SVM-Based Classification of Leukocytes Using the Morphological Pattern Spectrum

Juan Manuel Ramirez-Cortes[1], Pilar Gomez-Gil[2], Vicente Alarcon-Aquino[3], Jesus Gonzalez-Bernal[2], and Angel Garcia-Pedrero[2]

[1] Department of Electronics
National Institute of Astrophysics, Optics and Electronics.
Luis Enrique Erro No. 1 Tonantzintla, Puebla. 72840. Mexico
jmramirez@ieee.org
[2] Department of Computational Science;
National Institute of Astrophysics, Optics and Electronics.
Luis Enrique Erro No. 1 Tonantzintla, Puebla. 72840. Mexico
pgomez@acm.org, jagonzalez@inaoep.mx,
agarciapedrero@gmail.com
[3] Department of Electronics and Computer Science, University of the Americas,
Puebla, Mexico. Santa Catarina Martir, Cholula, Puebla, 72820. Mexico
vicente.alarcon@udlap.mx

Abstract. In this paper we present the morphological operator pecstrum, or pattern spectrum, as a feature extractor of discriminating characteristics in microscopic leukocytes images for classification purposes. Pecstrum provides an excellent quantitative analysis to model the morphological evolution of nuclei in blood white cells, or leukocytes. According to their maturity stage, leukocytes have been classified by medical experts in six categories, from myeloblast to polymorphonuclear corresponding to the youngest and oldest extremes, respectively. A feature vector based on the pattern spectrum, normalized area, and nucleus - cytoplasm area ratio, was tested using a multilayer perceptron neural network trained by backpropagation, and a Support Vector Machine algorithm. Results from Euclidean distance and k-nearest neighbor classifiers are also reported as reference for comparison purposes. A recognition rate of 87% was obtained in the best case, using 36 patterns for training and 18 for testing, with a three-fold validation scheme. Additional experiments exploring larger databases are currently in progress.

1 Introduction

The use of differential counting and assessment of blood white cells morphology from bone marrow is an important source of data for clinical cytology in a wide range of pathologies in the medical fields of oncology and hematology [1-3]. In order to accomplish the analysis, an important attribute to consider in leukocytes analysis is the age of the cell, which is strongly reflected in a morphological

P. Melin et al. (Eds.): Soft Comp. for Recogn. Based on Biometrics, SCI 312, pp. 19–35.
springerlink.com

evolution of its nucleus and cytoplasm. Although the maturity of the cell is a continuous variable, white blood cells are classified into discrete classes. Because the boundaries between classes are not well-defined, there are variations of counts among experts performing the same task, or even variations of counts performed by the same person on some samples in different moments.

The leukocyte differential counting consists of the quantification of the various white blood cells populations present in peripheral blood. Even though they derive from the same progenitor cell and interact with one another, each population can be considered relatively independent in terms of maturation, function, and control mechanism. The differential count has two main objectives, the search for quantitative abnormalities in morphologically normal white blood cells populations which requires high levels of precision and accuracy, and the search for morphologic abnormalities, i.e., the identification of immature or atypical cells for diagnostic or monitoring purposes, which requires a high level of clinical sensitivity. The traditional microscopic method based on the count of 100 cells has three types of error: statistical error, distributional error owing to unequal distribution of cells in the smear, and error in identifying cells related to the subjective interpretation of the examiner.

Cytology experts have identified six categories of this type of cells, according to their maturity stage, that ordered from youngest to oldest are named: Myeloblast, promyelocyte, myelocyte, metamyelocyte, band, and polymorphonuclear leukocytes [4,5]. Some typical examples of these categories are shown in figure 1, where nucleus and cytoplasm of each cell can be easily distinguished. Myeloblast is the earliest recognizable myeloid cell, and normally it has a large round to oval nucleus. Promyelocyte is slightly larger than a myeloblast. Its nucleus, although similar to a myeloblast, shows slight chromatin condensation and less prominent nucleoli with respect to the size of the cytoplasm. Myelocytes are slightly smaller than promyelocytes and have eccentric round-oval nuclei, often flattened along one side. In a typical image of a myelocyte, the proportion of the area occupied by the nucleus with respect to the cytoplasm is about 50-50 %. Metamyelocytes (10-18μ) are slightly smaller than myelocytes. They have kidney shaped indented nuclei and relatively dense chromatin, especially along the nuclear membrane. Bands are usually characterized by a U-shaped or deeply indented nucleus. Opposite sides or lobes are of roughly equal size or diameter, and there is no nuclear constriction greater than one half of the lobe diameter. Polymorphonuclear (PMN) leukocytes or segmented neutrofiles are characterized by definite lobation with thin thread-like filaments of chromatin joining the 2-5 lobes. From the previous description of typical forms of leukocytes, we can notice that definition of characteristics to discriminate among them is not straightforward, and it involves some degree of subjectivity.

Currently, there are some specialized manufacturers which offer professional equipments to perform hematologic analysis, such as Siemens ADVIA 2120, Sysmex XE-2100, or Beckman Coulter LH 750, some of them based on multicolor flow cytometry methods for the cell identification [6,7]. In this research we explore the use of mathematical morphology as a tool to create a feature extractor

able to discriminate leukocytes through image processing techniques from microscopic cell images using a personal computer.

The automatic identification of leukocytes according to their maturity stage could be very useful in the construction of automatic systems for the classification of different kinds of leukemia, differential counting of cells, and cytology applications. For this purpose, it is required to use a feature extractor able to represent shapes and at the same time, be tolerant enough to variations among patterns due to translations and rotation of the images. An additional problem associated to the creation of automatic leukocyte recognizers is the lack of public databases of leukocytes images to use for experimentation. The published research that we have found related to this kind of problem is normally carried out using private databases, which in many cases present an unbalanced number of images with respect to classes, or have been digitized under very different conditions.

Some approaches have been recently reported for automatic image analysis applied to microscopic cell imagery. In [8] the authors present a methodology to achieve a fully automated detection and classification of leukocytes based on microscope color images identifying the following classes: Basophil, Eosinophil, Lymphocyte, Monocyte and Neutrophil. The classification process is carried out using a parallel feed-forward neural network system. The feature vector is generated using the following information: Area, perimeter, convex area, solidity, major axis length, orientation, filled area, eccentricity, ratio between the cell and nucleus areas, the nucleus' "*rectangularity*" (ratio between the perimeter of the tightest bounding rectangle and the nucleus perimeter), the cell "*circularity*" (ratio between the perimeter of the tightest bounding circle and the cell perimeter), the number of lobes, and the solidity, area and mean gray-level intensity of the cytoplasm. In [9] the authors present a combination of continuous wavelet transform and moment-based features, obtained from atomic force microscope data to represent shape information of leukocyte and erythrocyte cells at multiple resolution levels. The features are then used to discriminate between anucleate versus nucleate cell types and to distinguish cells from a fibrous environment such as a tissue scaffold or stint. The authors describe the features as invariant under operations of translation, rotation, and scaling.

In [10] the authors present a classification system based on mathematical morphology to segment white blood cells. The features extracted from the segmented images include the first and second granulometric moments of nucleus, cell area, and the peak location of the morphological pattern spectrum. The classification is carried out using the fuzzy C-means algorithm. Morphological operators have been reported also for white blood cell segmentation into nucleus and cytoplasm, using scale-space filtering, the watershed transform and granulometric analysis [11].

In this work we use the morphological operator pecstrum, or pattern spectrum, as a feature extractor of discriminating characteristics in microscopic leukocytes images for classification purposes. Pecstrum is shown to provide an excellent quantitative analysis to model the morphological evolution of nuclei in blood white cells, or leukocytes according to their maturity stage. A feature vector based on the pattern spectrum, normalized area, and nucleus - cytoplasm area ratio, was tested using a multilayer perceptron neural network trained by backpropagation, and a Support Vector Machine algorithm.

2 Image Morphology

Mathematical morphology aims to quantitatively describe operations effective for the shape of objects in an image [12,13]. Operations are described by combinations of a basic set of numerical manipulations between an image A and a small object B, called a structuring element, which can be seen as a probe that scans the image and modifies it according to some specified rule. The shape and size of B, typically much smaller than the image A, together with the specific rule, define the characteristics of the performed process.

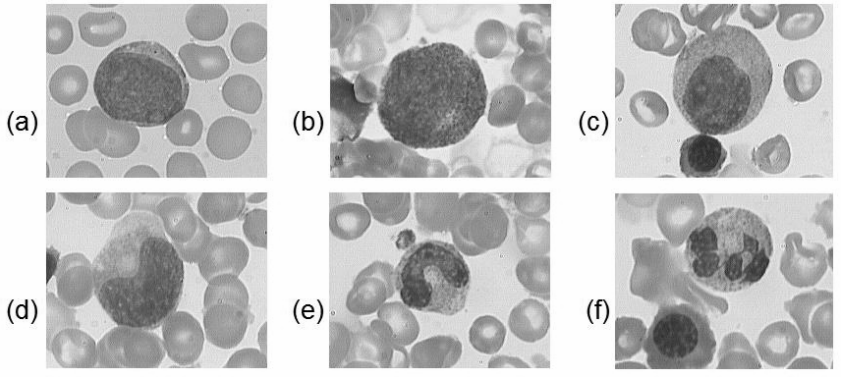


Fig. 1. Maturity stages of white blood cells. (a) Myeloblast. (b) Promyelocyte. (c) Myelocyte. (d) Metamyelocyte. (e) Band, (f) Polymorphonuclear leukocytes (PMN).

A very interesting morphological operator is the pattern spectrum or pecstrum. This operator decomposes the target image in morphological components according to the shape and size of the structuring element, providing a quantitative analysis of the morphological content of the image. Pecstrum was originally developed and reported by Maragos and Pitas [14,15]. Although it presents excellent properties as a shape extractor, with invariance to translation and rotation, pecstrum has not been extensively used, probably because it results computationally intensive in some applications, however, the available current hardware solutions easily overcome this disadvantage. The pattern spectrum has been used in the last years with several purposes: Analysis of partial discharges in high voltage systems [16], texture analysis in several applications, such as images of debris particles in polymers and composite materials [17,18], lip recognition [19], cytology of bone marrow images for the counting of white blood cells based on morphological granulometries [20], and feature extractor for hand-shape biometric applications [21]. In this work we use the pattern spectrum as a feature extractor in order to quantitatively represent the morphological evolution of the white blood cell, for cytology purposes.

2.1 Image Morphology Definitions

Binary mathematical morphology is based on two basic operators: *Dilation, and erosion*. Both are defined in terms of the interaction of the original image A to be processed, and the structuring element B. Next, both basic operators are defined. Morphological dilation is defined as the set union of the objects A obtained after the translation of the original image for each coordinate pixel *b* in the structuring element B:

$$A \oplus B = \bigcup_{b \in B} T_b(A) \tag{1}$$

Binary dilation can be interpreted as the combination of two sets by using the vector additions of set elements, called the *Minkowski Addition.* This operation is expressed as:

$$A \oplus B = \{ \mathrm{r} \mid \mathrm{r} = \mathrm{a} + \mathrm{b} \quad \forall \in \mathrm{A} \text{ and } \mathrm{b} \in \mathrm{B} \} \tag{2}$$

Morphological erosion is the morphological dual of the dilation. It is defined in terms of the Minkowski substraction as:

$$A \Theta B = \{ \mathrm{r} \mid (\mathrm{r} + \mathrm{b}) \in \mathrm{A} \; \forall \mathrm{b} \in \mathrm{B} \} \tag{3}$$

This definition can be expressed in terms of set intersections as:

$$A \Theta B = \bigcap_{b \in B} T_{-b}(A) \tag{4}$$

An important operator, which is the backbone of the pattern spectrum, is the *opening* morphological filter, defined as an erosion operation followed by a dilation using the same structuring element. The opening operator is defined as:

$$A \circ B = (A \Theta B) \oplus B \tag{5}$$

Dilation tries to undo erosion operation. However, some details closely related with the shape and size of the structuring element will vanish. Furthermore, an object disappearing as consequence of erosion cannot be recovered.

C*losing* morphological operator, which is the dual of the *opening*, is defined as dilation followed by an erosion operation using the same structuring element:

$$A \bullet B = (A \oplus B) \; \Theta B \tag{6}$$

Opening and closing filters have been used as discriminators for filtering, segmentation, edge detection, differential counting, or numerical analysis of shapes.

2.2 Morphological Pattern Spectrum

Pecstrum, or pattern spectrum of a compact binary image $A \subseteq R^2$, relative to a convex binary pattern $B \subseteq R^2$ called the *structuring element*, is defined as the differential size distribution function:

$$P_x(n,B)=\frac{dM(A\circ nB)}{dn},\quad n\geq 0 \tag{7}$$

where M represents the area measured in the intermediate operations, and nB is the n-times dilated structuring element. *Pecstrum* has the property of invariance to translation and rotation when B is an isotropic structuring element. Scale is determined by the size of the structuring element. The discrete version is given by:

$$P(n,B)=\frac{M[A\circ nB]-M[A\circ(n+1)B]}{M[A]} \tag{8}$$

Figure 2 shows the obtained pecstrum P(n) of a fractal image. It was calculated using a structuring element formed by a small circle with an initial size and incremental steps of 2 pixels.

Figure 3 shows the intermediate stages involved in the computation of the pattern spectrum between consecutive opening filters with an incremental structuring element according to the definition. The process concludes when the last image vanishes completely. It can be noticed also that the level of details filtered in each stage is closely related to the size of the structuring element.

Figure 4 shows the pattern spectra obtained from four ellipsoidal figures. It is relevant to point out that the pattern spectrum is able to represent quantitatively the geometric form when the horizontal axis of the ellipse increases. Given that the structuring element used in this experiment is a circle, there is only one spectral component when the two axes have the same length, i.e., when the input image is a circle itself.

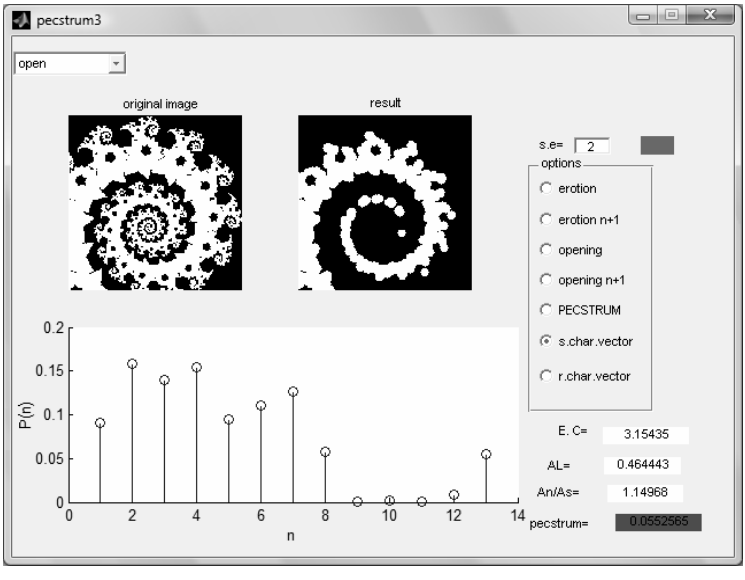


Fig. 2. Pattern spectrum of a fractal binary image.

As the horizontal axis increases some pattern spectral components arises, and the larger component, which is related to the center of the image, decreases. In any case the sum of all components always equals one, since the spectrum is normalized according to the definition.

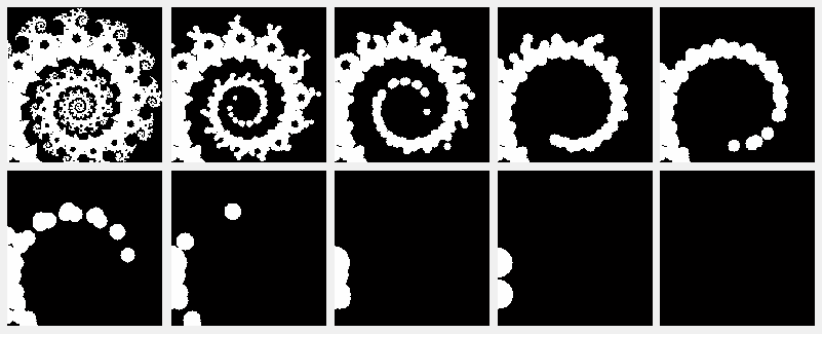


Fig. 3. Results obtained from intermediate opening filters with an incremental structuring element.

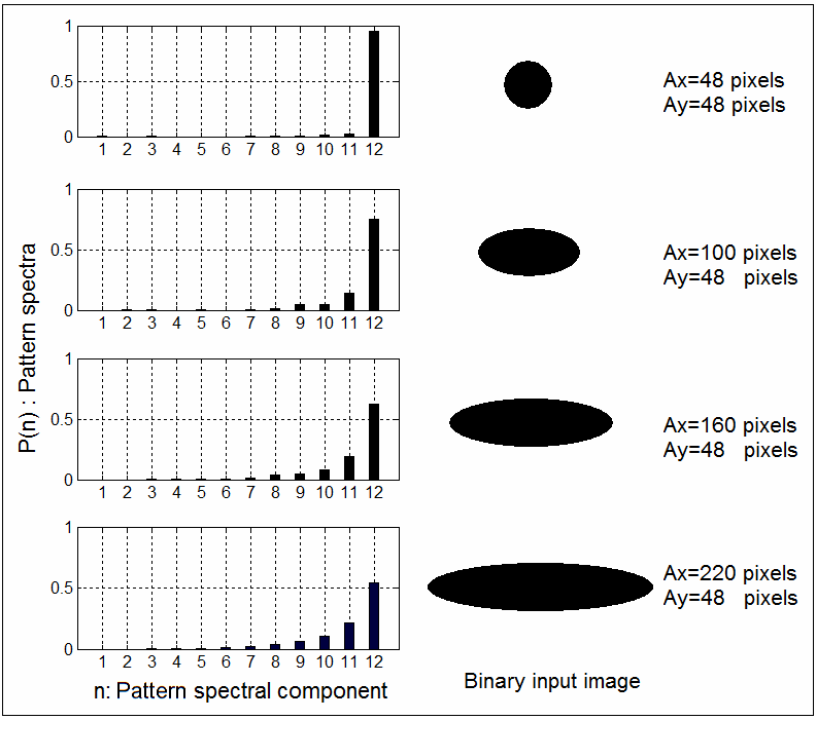


Fig. 4. Pattern spectra obtained from four ellipses increasing the major axis Ax.

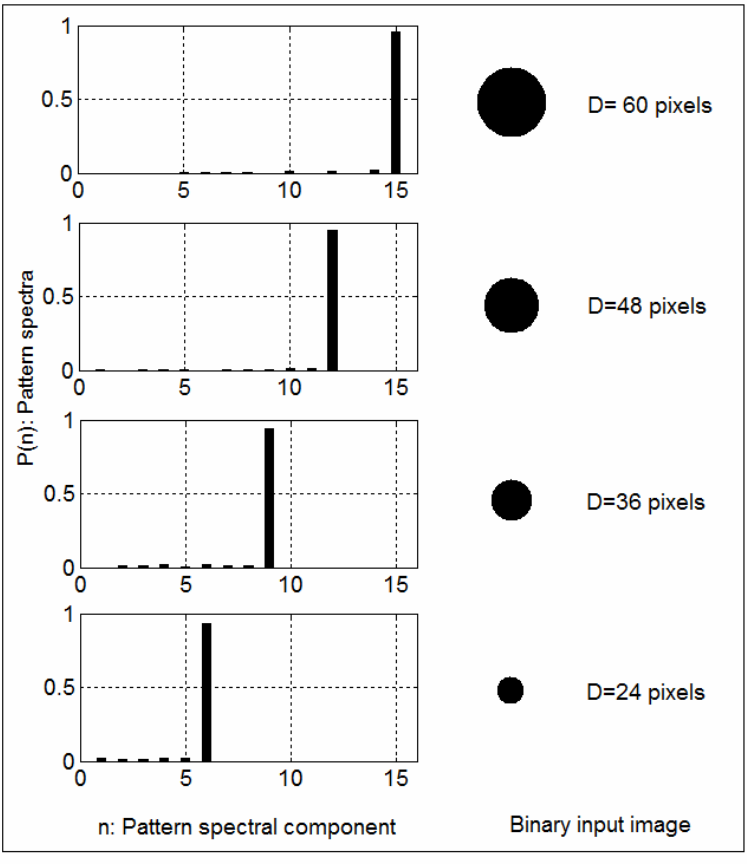


Fig. 5. Pattern spectra obtained from four circles with different diameter D.

The principal component located in the position n=12, indicates that the central part of the ellipses contains a circle of diameter D= 48 pixels. Figure 4 shows the situation in which the major axis Dx of the four ellipses consecutively increases while the minor axis remains without changes, and the result is reflected in the distribution of component in the pattern spectrum. Furthermore, figure 5 shows the case when the input image is a circle, and the diameter is increased with steps of 12 pixels. In that case there is only one component in the pattern spectrum, and the size is reflected in the position of the principal component. Some noise in the remaining pattern spectral components due to the shape and size of the structuring element can be distinguished.

These characteristics of the pattern spectrum clearly suggest the possibility of using it as a mathematical tool oriented to describe quantitatively the morphology of the white blood cells nuclei, which is the goal in the work presented in this paper.

3 Composed Feature Vector

A composed feature vector based on the cell morphological information is proposed as follows:

$$\vec{x} = \{A_L \vdots R_{nc} \vdots P(n,B)\} \tag{10}$$

where:

A_L = Leukocyte normalized area

R_{nc} = Nucleus-cytoplasm ratio

$P(n,B)$ = Pattern spectrum of the nucleus.

This vector contains information that in general human experts could consider to classify a leukocyte: Shape of the nucleus, represented by the pecstrum $P(n,B)$, relationship among the area of the cytoplasm and the area of the nucleus, represented by R_{nc}, and size of the cell, represented by A_L. Figure 6-a shows an image of a leukocyte, while figure 6-b shows the results obtained for the computation of the composed feature vector described before. The segmented image is plotted in the up-left part of figure 6-b. The obtained feature vector is plotted in the center of the figure, and the corresponding values are:

$$\begin{aligned}\vec{x} &= \{A_L \vdots R_{nc} \vdots P(n)\} \\ &= \{0.2700,\ 0.2878,\ 0.2709,\ 0.6216,\ 0.0858,\ 0.0075,\ 0.0009, 0.0, 0.0\}\end{aligned} \tag{11}$$

4 Multilayer Perceptron

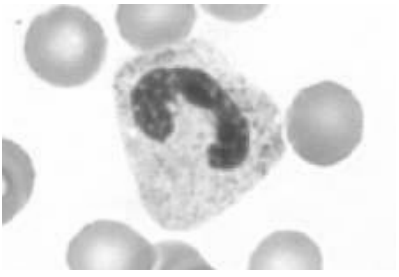
The neural network model used in this work was a feedforward multilayer perceptron (MLP) with three layers of neurons: The input layer with nine neurons corresponding to the number of elements in the feature vectors, a hidden layer, and the output layer with six neurons corresponding each to a leukocyte class. By experimentation, FFNN architectures of (9-15-6) and (9-19-6) were found to fit best the data. When a multilayer perceptron network has *n* input nodes, one hidden-layer of *m* neurons, and two output neurons, the output of the network is given by [22]:

$$y_i = f_i\left(\sum_{k=1}^{m} w_{ki}\, f_k\left(\sum_{j=1}^{n} w_{kj}\, x_j\right)\right) \tag{12}$$

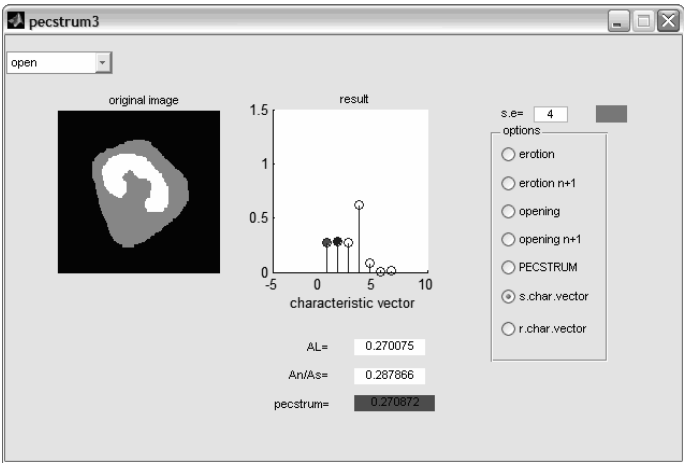
f_k, f_i: Activation functions of the hidden-layer and the output-layer neurons.

w_{ki}, w_{kj}: Weights connected to the output layer and to the hidden layer.

x_j: Input feature vector.



(a). original image



(b) Plot of composed feature vector

Fig. 6. An example of feature vector extraction

All neurons used a sigmoid as an activation function, expressed as:

$$f(u) = \frac{\alpha}{1 + e^{-bu}} \tag{13}$$

The feed forward multi-layer perceptron used in this work was trained using the Levenberg-Marquardt back propagation algorithm [23]. The backpropagation algorithm used in the training of multilayer perceptrons, is formulated as a non linear least-squares problem. Essentially, the Levenberg-Marquardt algorithm is a least-squares estimation method based on the maximum neighborhood idea. Let *E(w)* be an objective error function made up of *m* individual error terms e_i^2 *(w)* as follows:

$$E(w) = \sum_{i=1}^{m} e_i^2(w) = \left\| f(w) \right\|^2 , \tag{14}$$

where

$$e_i^2(w) = (y_{di} - y_i)^2 , \tag{15}$$

y_{di} is the desired value of output neuron i, and y_i is the actual output of that neuron. The aim of the Levenberg-Marquardt algorithm is to compute the weight vector w such as $E(w)$ is minimum. In each iteration the weight vector is updated according to eq. (6):

$$w_{k+1} = w_k + \delta w_k \ , \tag{16}$$

where

$$\delta w_k = -\left(J_k^T f(w_k)\right)\left(J_k^T J_k + \lambda I\right)^{-1} . \tag{17}$$

J_k is the Jacobian of f evaluated at w_k, λ is the Marquardt parameter, and *I* is the identity matrix. The number of epochs in the training phase differs from one example to another, however, the Levenberg-Marquardt back propagation algorithm provided consistently a fast convergence.

Figure 7 shows an example of training performance obtained in one of the experiments carried out with the data described in this work. It can be noticed that the goal was achieved in 30 epochs.

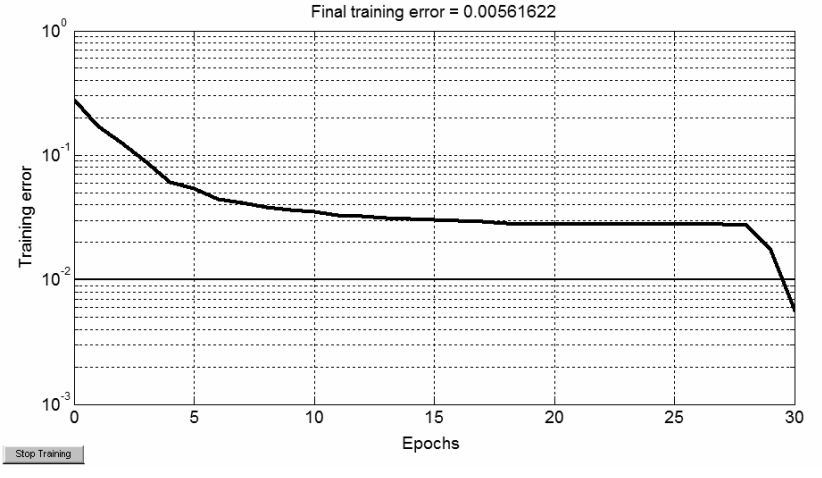


Fig. 7. An example of training performance

5 Support Vector Machines (SVM)

SVMs have been proposed for pattern recognition in a wide range of applications by its ability for learning from experimental data, and its effectiveness over some

other conventional parametric classifiers. Briefly, SVM is a statistical learning method based on a structural risk minimization procedure, which minimizes the upper bound of the generalization errors consisting of the sum of training errors and a confidence interval [24]. The original input space is transformed through a non-linear feature mapping to a high dimensional feature space, where the data is linearly separable by a hyperplane. The goal during the training process is to find the separating hyperplane with the largest margin in the obtained hyperspace. The transformation is performed using a non-linear function whose form is determined by some kernel. There are three common kernels used for the non-linear feature mapping: Polynomial, radial basis function, and sigmoid kernels. Linear hyperplane classifiers are based on the class of decision functions [25,26]:

$$y = sign\left((w^T x) + b\right) \tag{18}$$

The optimal hyperplane is defined as the one with the maximal margin of separation between the wo classes. The solution of a constrained quadratic optimization process can be expanded in terms of a subset of the training patterns called support vectors that lie on the margin:

$$w = \sum_{i=1}^{N} v_i x_i \tag{19}$$

Thus the decision rule depends only on dot products between patterns:

$$y = sign\left(\sum_{i=1}^{N} v_i (x_i \cdot x) + b \right) \tag{20}$$

The above linear algorithm is performed in the new feature space obtained through some non-linear transformation Ø by using some of the described kernels. The kernel is related to the Ø function by:

$$K(x_i, x) = \phi(x_i) \cdot \phi(x) \tag{21}$$

In this work the radial basis function (RBF) was used. RBF is defined as:

$$K(x_i, x_j) = e^{-\left\| x_i - x_j \right\|^2 / \sigma} \tag{22}$$

Classification of the test sample x is then performed by:

$$y = sign\left(\sum_{i=1}^{N} \alpha_i v_i K(x_i, x) + b \right), \tag{23}$$

where N is the number of training samples, v_i is the class label, α_i a Lagrangian multiplier, the elements x_i for which $\alpha_i > 0$ are the support vectors, and $K(x_i,x)$ is the function kernel. In this work a Gaussian radial basis function was used.

6 Experimental Description and Results

Input data was obtained from hand-segmented images of white blood cells in different maturity stages, some examples are shown in figure 8. Data set consists of 54 patterns corresponding to 7 artificial and 2 real images for each class. Artificial images were needed due to the lack of a public database to carry out the experiments. An artist generated the artificial segmented images hand-painting them in black and white pixels, using Paint. Figure 8-a shows examples of segmented real cells corresponding to each class, while figure 8-b shows examples of artificial cells. In order to test the performance of the proposed feature vector we used 4 different classifiers based on Euclidean distance, K-nearest neighbor (KNN), Feed-forward Neural Networks (FFNN) and Support Vector Machines (SVM) respectively. All classifiers were built using Matlab V 7.0. The FFNN was trained using the Matlab function ´trainlm´, which implements the Levenberg-Marquardt backpropagation training algorithm. SVM was extended to multi-class classification using the method known as DAGSVM (Directed Acyclic Graph Support Vector Machine) [25]. Matlab function ´svmtrain´ was used as a nucleus for the implementation of DAGSVM. The KNN was tested for k=1, 3 and 5. A statistical analysis of the obtained feature vectors is shown in Table 1. The table is organized by leukocyte type, showing means and variances of the feature characteristics for each class obtained from the dataset described before.

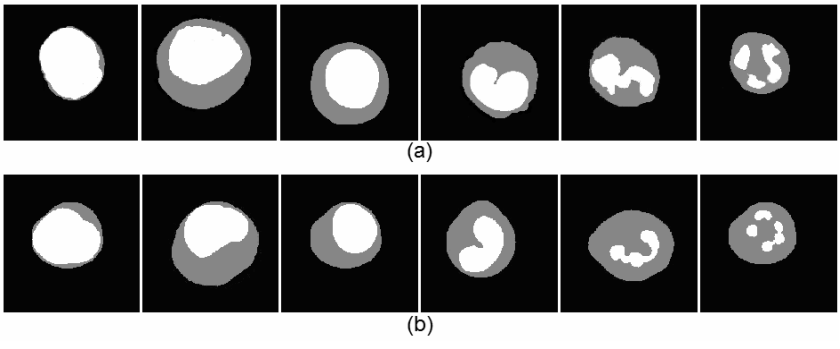


Fig. 8. Examples of patterns for each class, from left to right: Myeloblast, promyelocyte, myelocyte, metamyelocyte, band, and polymorphonuclear leukocytes (PMN). (a) Real images. (b) Artificial images.

It is interesting to analyze the plot of the mean feature vectors corresponding to each class. Figure 9 shows the feature vectors ordered according to the cell maturity stage from the youngest to the oldest. It can be noticed that the form of the feature vectors evolves by displacing the principal components to the right and decreasing their value at the same time.

Table 1. Mean and variance of composed feature vectors for the training data.

	A_L	R_n	P(1)	P(2)	P(3)	P(4)	P(5)	P(6)	P(7)
Leukocyte type:									
Myeloblast: Mean	0.1860	0.8347	0.8156	0.0726	0.0371	0.0124	0.0084	0.0074	0.0049
Variance	0.0203	0.0643	0.1005	0.0516	0.0162	0.0089	0.0062	0.0058	0.0031
Promyelocyte: Mean	0.2939	0.5075	0.841	0.0615	0.0361	0.0234	0.0114	0.0073	0.0046
Variance	0.0331	0.0029	0.0642	0.0158	0.0089	0.0183	0.0058	0.0046	0.0028
Myelocyte: Mean	0.2105	0.4984	0.9234	0.0336	0.0198	0.0091	0.0026	0.0026	0.0025
Variance	0.0321	0.0114	0.0254	0.0132	0.0049	0.0016	0.0018	0.0021	0.0003
Metamyelocyte: Mean	0.2048	0.4629	0.5170	0.3760	0.0678	0.0234	0.0070	0.0042	0.0026
Variance	0.0384	0.0136	0.0507	0.0751	0.0249	0.0035	0.0007	0.0008	0.0017
Band: Mean	0.2127	0.2028	0.5190	0.3510	0.1210	0.0088	0 .0	0.0	0.0
Variance	0.0226	0.0047	0.1691	0.1063	0.0601	0.0026	0 0	0.0	0.0
PMN: Mean	0.1796	0.1736	0.3893	0.4186	0.1573	0.0346	0.0	0.0	0.0
Variance	0.0198	0.0447	0.0179	0.0099	0.0398	0.0119	0 0	0.0	0.0

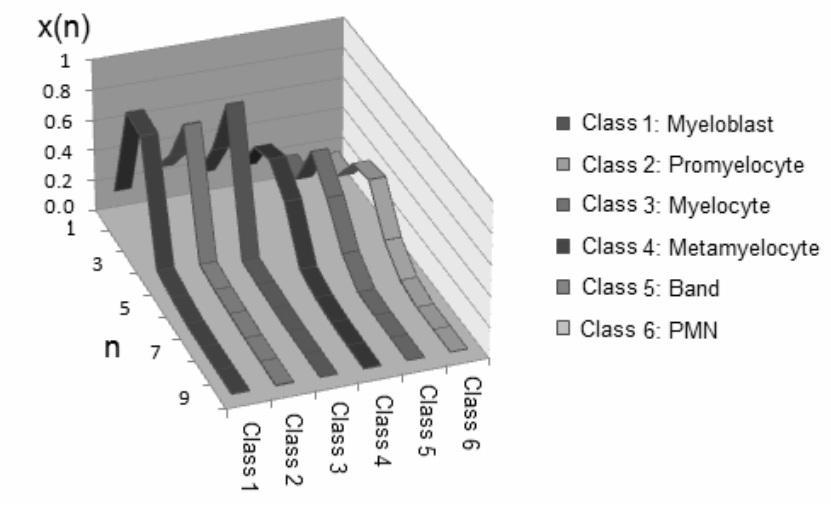


Fig. 9. Mean feature vectors corresponding to the six leukocyte classes

Table 2. Three-fold cross validation

Classifier	Classification Rate
Euclidean Distance	77.7%
K-NN with K= 1	70.4%
K-NN with K= 3	72.2%
K-NN with K= 5	70.4%
FFNN with 19 hidden nodes	**87.6%**
FFNN with 14 hidden nodes	84.9%
DAGSVM	71.6%

Three-fold cross validation was used to obtain the classification results shown in Table 2. Best results were obtained by a FFNN with 19 hidden nodes. A similar work reported a classification rate around 77%, [27] however, a fair comparison is not possible due to differences in the databases used.

7 Conclusions

A composed feature vector based on morphological information for classification of hand-segmented leukocytes images was presented. This feature vector was constructed with the morphological *pecstrum* as the central part, and some geometric considerations. The composed feature vector shows very good attributes to reflect the evolution in time of the white blood cells according to their maturity stage. The proposed feature extraction was tested using 4 different classifiers based on Euclidean distance, K-nearest neighbor (KNN), Feed-forward Neural Networks (FFNN) and Support Vector Machines (SVM) respectively. Three-fold cross validation was used to get the classification rates for each classifier. The best result was obtained by a FFNN with 19 hidden nodes, getting 87.6% of corrected classification. The reported work is part of an ongoing project on the cytology of human bone marrow images. The tools described in this paper will be used as the base for a second part in the project, oriented to the differential counting of white blood cells, and its use in the diagnosis and follow up of several pathologies, under the supervision of medical experts.

References

1. Butarello, M., Plebani, M.: Automated blood cell counts: State of the art. American Journal of Clinical Pathology 130, 104–116 (2008)
2. Araseki, K., Matsuda, A., Germing, U., Jinnai, I., Kuendgen, A., Iwanaga, M., Miyazaki, Y., Hata, T., Bessho, M., Gattermann, N., Tomonaga, M.: Differences in the distribution of subtypes according to the WHO classification 2008 between Japanese and German patients with refractory anemia according to the FAB classification in myelodysplastic syndromes. Leukemia Research 33(1), 66 (2009)
3. Bogdanovic, G., Jakimov, D., Stojiljkovic, B., Jurisic, V.: The cell growth, morphology and immunocytochemistry of novel cell line established from a bone marrow of the patient with therapy-related myelodysplastic syndrome. Medical Oncology 24(4), 419–424 (2007)
4. Germing, U., Aul, C., Niemeyer, C.M., Haas, R., Bennett, J.M.: Epidemiology, classification and prognosis of adults and children with myelodysplastic syndromes. Annals of Hematology 87(9), 691–699 (2008)
5. Jones, D.: Approaches to Classification of Lymphoma and Leukemia, book chapter on Neoplastic Hematopathology: Experimental and Clinical Approaches, pp. 3–20. Humana Press, Totowa (2010)
6. Mayumi-Ushizima, D., Costa-Rosatelli, M.: E-Learning in Medical Diagnosis. In: Proceedings of 16th Brazilian Symposium on Computer Graphics and Image Processing, Natal, Brazil (2005)

7. Kang, S.H., Kim, H.K., Ham, C.K., Lee, D.S., Cho, H.I.: Comparison of four hematology analyzers, CELL-DYN Sapphire, ADVIA 120, Coulter LH 750, and Sysmex XE-2100, in terms of clinical usefulness. International Journal of Laboratory Hematology 30(6), 480–486 (2007)
8. Piuri, V., Scotti, F.: Morphological classification of blood leukocytes by microscope images. In: Proceedings of IEEE International Conference on Computational Intelligence for Measurement Systems and Applications Boston, Boston, MD, USA, July 14-16 (2004)
9. Chen, Q., Fan, Y., Udpa, L., Ayres, V.: Cell classification by moments and continuous wavelet transform methods. International Journal of Nanomedicine 2(2), 181–189 (2007)
10. Theera-Umpon, N.: White Blood Cell Segmentation and Classification in Microscopic Bone Marrow Images. In: Wang, L., Jin, Y. (eds.) FSKD 2005. LNCS (LNAI), vol. 3614, pp. 787–796. Springer, Heidelberg (2005)
11. Dorini, L.B., Neucimar, R.M., Leite, J.: White blood cell segmentation using morphological operators and scale-space analysis. In: Proceedings of XX Brazilian Symposium on Computer Graphics and Image Processing, Belo Horizonte, Brazil, October 7-10, pp. 294–301 (2007)
12. Shih, F.Y.: Image processing and mathematical morphology: Fundamentals and applications. CRC Press, Taylor and Francis Group (2009)
13. Ledda, A., Philips, W.: Majority Ordering and the Morphological Pattern Spectrum. In: Blanc-Talon, J., Philips, W., Popescu, D.C., Scheunders, P. (eds.) ACIVS 2005. LNCS, vol. 3708, pp. 356–363. Springer, Heidelberg (2005)
14. Maragos, P.: Pattern spectrum and multiscale shape representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 11, 701–716 (1989)
15. Pitas, A., Venetsanopoulus, A.N.: Non-linear Digital Filters; Principles and Applications. Kluwer Academic Publishers, Norwell (1990)
16. Yunpeng, L., Fangcheng, L., Chengrong, L.: Pattern recognition of partial discharge based on its pattern spectrum. In: Proceedings of International Symposium on Electrical Insulating Materials, Kitakyushu, Japan (2005)
17. Ghosh, D., Tou Wei, D.C.: Material Classification Using Morphological Pattern Spectrum for Extracting Textural Features from Material Micrographs. In: Narayanan, P.J., Nayar, S.K., Shum, H.-Y. (eds.) ACCV 2006. LNCS, vol. 3852, pp. 623–632. Springer, Heidelberg (2006)
18. Ledda, A., Samyn, P., Quintelier, J., De Baets, P., Philips, W.: Polymer Analysis with Mathematical Morphology. In: Proceedings of IEEE Benelux Signal Processing Symposium, Hilvarenbeek, The Netherlands, pp. 87–92 (2004)
19. Omata, M., Hamamoto, T., Sangai, S.: Lip recognition using morphological pattern spectrum. In: Proceedings of Third International Conference on Audio- and Video-Based Biometric Person Authentication, Halmstad, Sweeden, pp. 108–114 (2001)
20. Theera-Umphon, N., Dhompongsa, S.: Morphological Granulometric Features of Nucleus in Automatic Bone Marrow White Blood Cell Classification. IEEE Transactions on Information Technology in Biomedicine 11(3), 353–359 (2007)
21. Ramirez-Cortes, J.M., Gomez-Gil, P., Sanchez-Perez, G., Prieto-Castro, C.: Shape based hand recognition approach using the pattern spectrum. Journal of Electronic Imaging 18(1), 013012, 1–6 (2009)
22. Haykin, S.: Neural networks and learning machines, 3rd edn. Pearson, Prentice Hall, New Jersey (2009)

23. Demuth, H., Beale, M., Hagan, M.: Neural Network Toolbox 6 User's guide. The MathWorks, Inc., 5-30, 5-33 (2009)
24. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2, 121–127 (1998)
25. Scholkopf, B., Smola, A.J.: Learning with kernels: support vector machines, regularization, optimization, and beyond, pp. 189–211. MIT Press, Massachussetts (2002)
26. Abe, S.: Support Vector Machines for Pattern Classification. Springer, Heidelberg (2005)
27. Theera-Umpon, N., Gader, P.D.: Counting white blood cells using morphological granulometries. Journal of Electronic Imaging 9(2), 170–177 (2000)

Hybrid System for Cardiac Arrhythmia Classification with Fuzzy K-Nearest Neighbors and Neural Networks Combined by a Fuzzy Inference System

Eduardo Ramírez, Oscar Castillo, and José Soria

Tijuana Institute of Technology, Graduate Studies, Tijuana BC. México
edurafl@hotmail.com, ocastillo@tectijuana.edu.mx, jsoria57@gmail.com

Abstract. In this paper we describe a hybrid architecture for classification of cardiac arrhythmias taking as a source the ECG records MIT-BIH Arrhythmia database. The Samples were taken from the LBBB, RBBB, PVC and Fusion Paced and Normal arrhythmias, as well as the normal heartbeats. These were segmented and transformation and 3 methods of classification were used: Fuzzy KNN, Multi Layer Perceptron with Gradient Descent and momentum Backpropagation and Multi Layer Perceptron with Scaled Conjugate Gradient Backpropagation. Finally, we used a Mamdani type fuzzy inference system to combine the outputs of each classifier, and we achieved a very high classification rate of 98%.

Keywords: Fuzzy KNN, Mamdani Fuzzy System, Neural Network, Arrhythmia Classification.

1 Introduction

An electrocardiogram or ECG represents the electrical activity of the heart, as a waveform graph. An ECG signal contains important information that can help medical diagnosis, reflecting cardiac activity of a patient, if it is normal or failing heart has certain pathologies. The ECG is the standard tool used in diagnosing heart disease [14].

The physicians get those signals easily and noninvasively adding electrodes on the patient's body. The Holter device is frequently used for ECG recording. Physicians apply the Holter device to the patient when the ECG monitor is required to find the existence of several heartbeats totally abnormal ECG in one day. A person can register about 100,000 heartbeats in one day[14].

The ECG shows each heartbeat as a series of electrical waves. The contractions that pump blood are represented by the P wave, the QRS complex and T wave. The P wave represents activity in the upper chambers of the heart. The QRS complex and T wave represents activity in the lower chambers [14] (see Fig. 1).

P. Melin et al. (Eds.): Soft Comp. for Recogn. Based on Biometrics, SCI 312, pp. 37–55.
springerlink.com

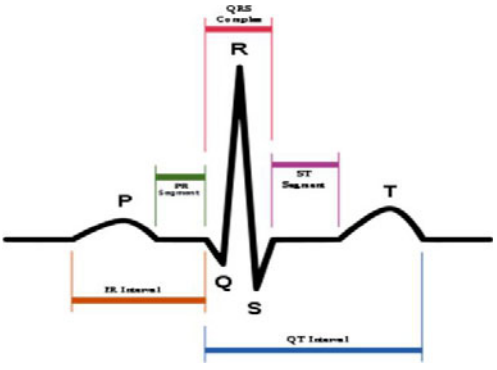


Fig. 1. Normal heartbeat with fiducial points.

By arrhythmia we mean any alteration in the activity of the heart rhythm in amplitude, duration and shape of the rhythm. The database of the MIT-BIH Arrhythmia is a set of 48 ECG records with 30 minutes duration each, and each record corresponds to a patient. In this database are different types of arrhythmias such as: L-Left Bundle Branch Block, R-Right Bundle Branch Block, A-Atrial Premature Beat, a-aberrated Premature Atrial Beat Premature junctional Nodal J-Beat, FFusion of Ventricular and Normal Beat, I-Ventricular Flutter Wave, J-junctional Nodal Escape Beat, E-Ventricular Escape Beat Supraventricular Premature Beat S-, f-Fusion of an Paced Beat Normal and normal heartbeats [24].

Many solutions have been proposed to develop automated recognition and classification of ECG. Some processing methods have been applied to the ECG signal: Statistical and Syntatic, MultiLayer Perceptron (MLP), Self-Organizing Maps (SOM), Learning Vector Quantization (LVQ), Linear Discriminant System, Fuzzy or Neuro-Fuzzy Systems, Support Vector Machines (SVM), Bayesian approach, Experts Systems, Markov Models, Hybrid system is a combination of different solutions [1, 2, 7, 10, 13, 15, 19, 20, 21, 22, 23, 25, 26, 27, 29, 30, 36, 39, 40].

In this paper, we describe a hybrid intelligent system for classifying cardiac arrhythmia classification using 3 algorithms: Fuzzy KNN, MLP Gradient Descent with momentum Backpropagation and MLP Scaled Conjugate Gradient Backpropagation, finally combine these outputs with a Mamdani fuzzy inference system where we can observe very high classification rate.

2 Problem Statement and Outline of our Proposal

The problem we address in this work is the classification of cardiac arrhythmias of the MIT-BIH arrhythmia database. The system learns a set of samples of arrhythmias: Normal Beat, Left Bundle Brunch Block, Right Bundle Brunch Block, Premature Ventricular Contraction, Fusion Paced and Normal Beat. The hybrid system is trained using several samples from each class above mentioned, and during testing it must determine the correct label for a previously unknown sample.

The hybrid system consists for 3 classification methods: Fuzzy K Nearest Neighbors, Multi Layer Perceptron with Gradient Descent with momentum Back-propagation and Multi Layer Perceptron with Scaled Conjugate Gradient Back-propagation. Each classifier provides a matrix of memberships or activitions in the case of the MLPs, this matrix named the matrix of integration is the input to a Mamdani type fuzzy inference system which combined the outputs of the 3 classifiers, where we can observe the increase accuracy in the classification rate given to the contribution of each classifier. See the proposed architecture in Figure 2.

We used the MLII electrode signal of MIT-BIH Arrhythmia Database (see Fig. 3), for this work five classes were selected: Normal beat, LBBB, RBBB, PVC, Fusion Paced and Normal Beat. We took 500 samples in total, 100 samples

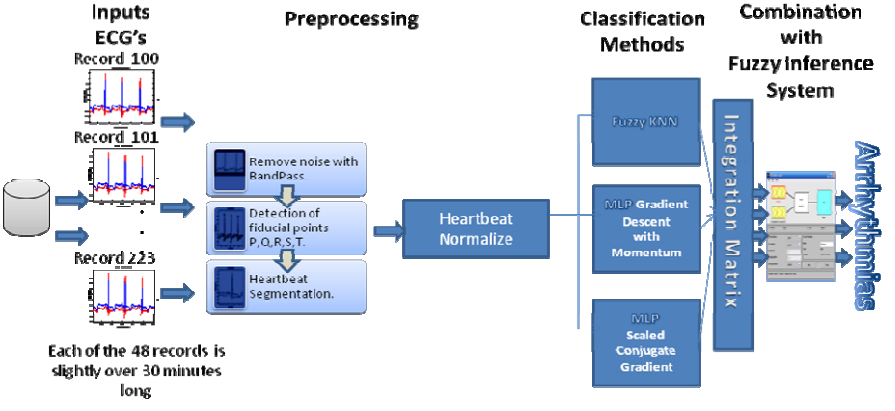


Fig. 2. Architecture of the Hybrid System.

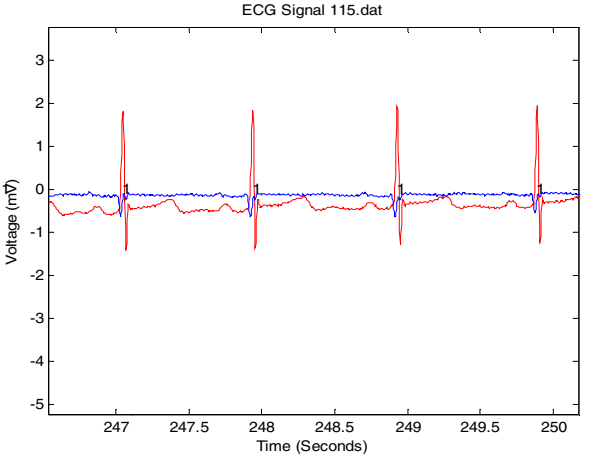
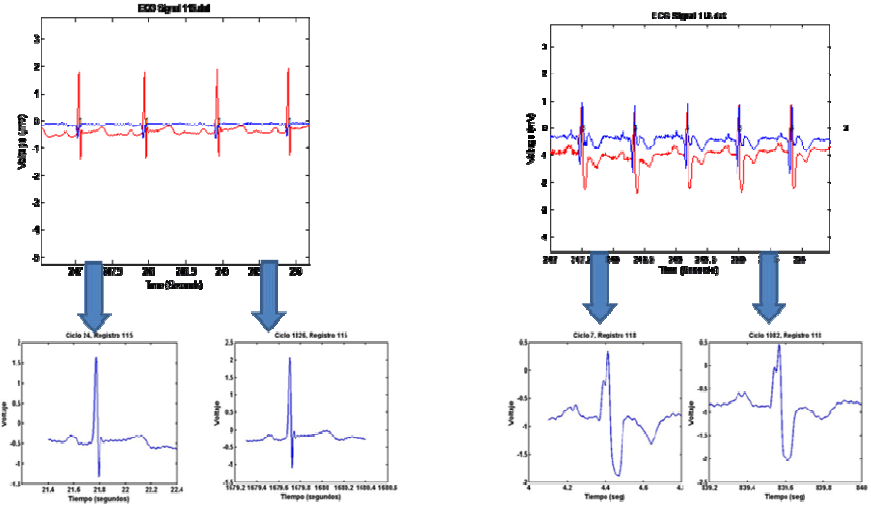


Fig. 3. The red line represents the MLII electrode signal.



(a) (b)

Fig. 4. Heartbeat Segmentation. a) sample for Normal class, b) sample for RBBB class.

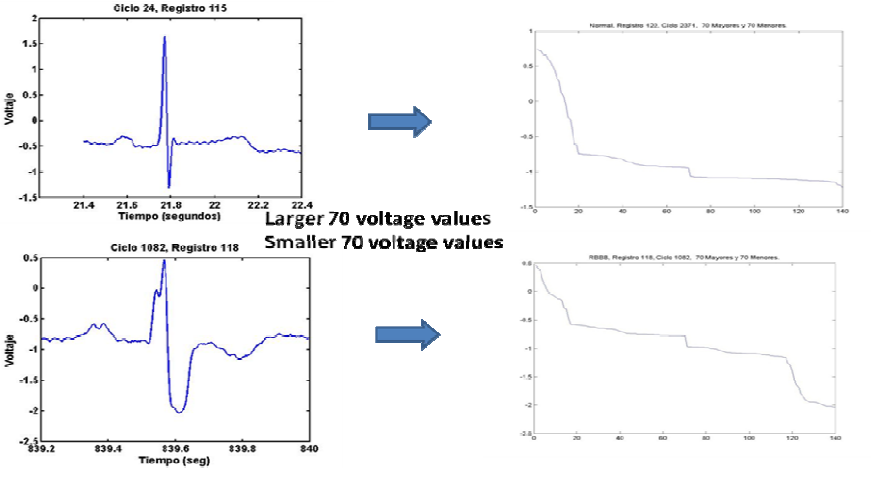


Fig. 5. Heartbeat Normalize.

for each class above mentioned. The used records for Normal was 115, 122, and 113, for LBBB 109, 111, 214, for RBBB 118, 124, 212, and for Fusion Paced and Normal class 217.

For the heartbeat segmentation we took the R points documented in the database, for this reason we determine the beginning and the end for each heartbeat, we do the segmentation manually (see Fig. 4).

Once we have the segmentation of the heartbeats, we apply a heartbeats transformation that consist in rearrange the voltage values; we just took the highest Larger 70 voltage values and the smaller 70 voltage values per each heartbeat. Having a set of vectors of 140 voltage values, where each vector corresponds a heartbeat (See Fig. 5 and 6).

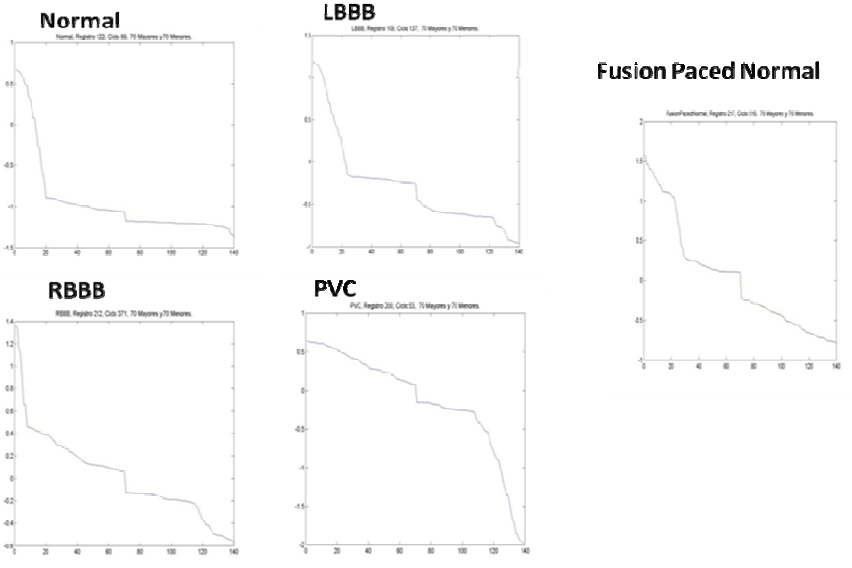


Fig. 6. Example of Heartbeats Normalized for each class.

In the following section, we present a brief review of some of the main concepts in order to understand our work.

3 Background Theory

In this section, we provide a general review of Fuzzy K-Neartest Neighbors algorithm, Multi Layer Perceptrons with Gradient Descent with momentum Backpropagation and Gradient Scaled Conjugate Backpropagation, we discuss how the outputs from the classifiers above mentioned can be combined using Mamdani fuzzy inference system, and we describe the heartbeats transformation that provide the input for each classifiers in our hibrid system.

3.1 Fuzzy K-Nearest Neighbors Algorithm

The fuzzy K-nearest neighbor algorithm assigns class membership to a sample vector rather than assigning the vector to a particular class. The advantage is that no arbitrary assignments are made by the algorithm. In addition, the vector's membership values should provide a lever of assurance to accompany the resultant

classification. For example, if a vector is assigned 0.9 membership in one class and 0.05 membership in two other classes we can be reasonably sure the class of 0.9 membership is the class to which the vector belongs. On the other hand, if a vector is assigned 0.55 membership in class one, 0.44 membership in class two, and 0.01 membership in class three, then we should be hesitant to assign the vector based on these results. However, we can feel confident that it does not belong to class three. In such a case the vector might be examined further to determine its classification, because the vector exhibits a high degree of membership in both classes one and two. Clearly the membership assignments produced by the algorithm can be useful in the classification process [17].

The basis of the algorithm is to assign membership as a function of the vector's distance from its k-nearest neighbors and those neighbors' memberships in the possible classes. The fuzzy algorithm is similar to the crisp version in the sense that it must also search the labeled sample set for the k-nearest neighbors. Beyond obtaining these k samples, the procedures differ considerably [17].

Let $W = \{x_1, x_2, ..., x_n\}$ be the set of n labeled samples. Also let $u_i(x)$ be the assigned membership of the vector x (to be computed), and u_{ij} be the membership in the ith class of the jth vector of the labeled sample set. The algorithm is as follows:

BEGIN
Input x, of unknown classification.
Set K, $1 \leq K \leq n$.
Initialize i = 1.
DO UNTIL (K-nearest neighbors to x found)
Compute distance from x to x_i
IF ($i \leq K$) **THEN**
Include xi in the set of K-nearest neighbors
ELSE IF (x_i closer to x than any previous nearest neighbor) **THEN**
Delete the farthest of the K-nearest neighbors
Include x_i in the set of the K-nearest neighbors.
END IF
END DO UNTIL
Initialize i = 1.
DO UNTIL (x assigned membership in all classes)
Compute $u_i(x)$ using (1).
Increment i.
END DO UNTIL
END

Where

$$u_i(x) = \frac{\sum_{j=1}^{k} u_{ij}(1/\left\|x - x_j\right\|^{2(m-1)})}{\sum_{j=1}^{k}(1/\left\|x - x_j\right\|^{2(m-1)})}$$

As seen by (1), the assigned memberships of x are influenced by the inverse of distances from the nearest neighbors and their class memberships. The inverse distance serves to weight a vector's membership more if it is closer and less if it is further from the vector under consideration. The labeled samples can be assigned class memberships in several ways. First, they can be given complete membership in their known class and non-membership in all other classes. Other alternatives are to assign the samples' membership based on the distance from their class mean or based on the distance from the labeled samples of their own class and those of the other class or classes and then to use the resulting memberships in the classifier [17].

3.2 Multi Layer Perceptron

An artificial neural network is a mathematical and computational model that simulates the abstract structure and functional aspects of biological neural networks. The basic computational elements in an ANN are known as *neurons*, nodes that serve as inputs, outputs or internal processing units. These neurons communicate and pass signals among themselves using what are known as *synapses*, which are modulated by weight parameters. The manners in which neurons are organized and connected define the architecture of the network. Here, we will focus on the feed-forward and fully connected architecture used by the MLP. In this network, the information moves in only one direction, forward, from the input nodes, though the hidden nodes (if any) to the output nodes, there are loops in a network of this type [5, 6, 28, 35].

3.3 Backpropagation Training for a Multilayer Perceptron

The BP algorithm can me summarized in four fundamental steps that are repeated iteratively, these are:

1. - Initialize the connection weights with random values.
2. - Compute the output of the ANN, denoted by y_p, by propagating each input pattern x_p through the network in a forward direction.
3. - Compute the error between the desired output, d, and the output produced by the ANN *y*, this is given by $E_p = \frac{1}{2}(d_{p\ln} - d_{p\ln})^2$.
4. - Adjust the connection weights according to the following rule,

$$W_{\ln[t+1]} = W_{\ln[t]} + \beta\delta_{p\ln} x_{pl} \quad (1)$$

In the above equation β represents the learning rate, a parameter that is used to modulate the amount by which the connection weights can be modified at each iteration of the algorithm. The network δ_{pln} is computed in two ways. First, if n is an output node then it is given by $(d_{p\ln} - y_{p\ln})$.

Otherwise, it is given by $f(y_{p\ln}) = \sum \delta_{p\ln} + 1w_l + 1$, where f represents the activation function of node n.

The above process is repeated until a stopping criterion is met, which can be a desired minimum error or a maximum number of iterations, also known as epochs [5, 6, 28, 35].

3.4 Gradient Descent with Momentum (GDM)

Another proposed improvement is the GDM algorithm, which takes a different approach towards overcoming some of the shortcomings of BP. It is equivalent to the traditional BP algorithm, with an added parameter called the momentum coefficient, which is used when the connection weights are updated [5, 6, 28, 35].

In GDM, the weight update takes into account previous weight changes along with the current propagated error. It can be understood as an averaging process that does not allow for weight changes that are extremely different relative to the previous ones. The update rule proposed in GDM is given by $\Delta w_m = \alpha\Delta w_{m-1} - (1-\alpha)\varepsilon d_m^w$, where α is the momentum coefficient, and Δw_{m-1} is the weight change used in the previous epoch. The ε parameter represents the learning rate. In GDM the momentum coefficient α is held constant throughout the entire training process, usually set to 0.9[5, 6, 28, 35].

3.5 Gradient Scaled Conjugate (SC)

In second-order methods, the curvature of the error surface, denoted by the second derivative of the error surface, is used to more efficiently guide the error down the error surface. These methods provide faster solutions because of the incorporation of an extra second derivative of error information and automatic internal adjustments that are made to the learning parameters. However, this comes at a substantial computational cost of the calculation of the second derivative of error, Hessian, H, and inverse of Hessian, R, especially for a network with a large number of weights. The conjugate gradients, gradient scaled conjugate, Newton and Levenberg-Maquardt are algorithms of second-order methods [5, 6, 28, 35].

4 Experiments

In this section we present the selected classes for MIT-BIH Arrhythmia Database, describe our experimental set-up, and detail the experimental results we have obtained by using Fuzzy KNN, MLP's and fuzzy inference system.

4.1 MIT-BIH Arrhythmia Database

For this work we used the MIT-BIH Arrhythmia Database. The MIT-BIH Arrhythmia Database contains 48 half-hour excerpts of two-channel ambulatory ECG recordings, obtained from 47 subjects studied by the BIH Arrhythmia Laboratory between 1975 and 1979. Approximately 60% of these recordings were obtained from inpatients. The subjects were 25 men aged 32 to 89 years, and 22 women aged 23 to 89 years. Two or more cardiologists independently annotated each record; disagreements were resolved to obtain the computer-readable reference annotations for each beat (approximately 110,000 annotations in all) included with the database. Sample images from the database are shown in Fig. 3 [24].

4.2 Experimental Setup

In the experimental setup we have a 70% for training that means 350 samples and for test 30% that represents 150 samples. With the 70% in the training we used cross validation k-fold of 10. We have 10 random subsamples that they are disjoint, we took one for validation and the rest for training and we repeat this cycle with the second subsample and so on. We apply this process of cross validation with three different parameters or architecture of respective classification method and the better result in this process determined the selections of a representative structure per each classifier. Once we have the three representative structures we training with de 70% of the samples and the rest for the test. Finally we have the tables that represent the confusion matrix for the classification.

The parameters used of the first classifier Fuzzy KNN is k=5, 4, 3 nearest neighbors. The architecture for the second classifier Multi Layer Perceptron is 140 neurons in the input layer, 10,000 epochs, learning rate= 0.3, learning method Gradient Descent with momentum 0.5 and we use 50, 100 and 150 neurons in the hidden layer, 5 neurons in the output layer. Each output represents the respective class. The architecture of the third classifier Multi Layer Perceptron with Scaled Conjugate Gradient Backpropagation is 140 neurons in the input layer, 10,000 epochs, lr= 0.001, also we used 50, 100, 150 neurons in the hidden layer, 5 neurons in the output layer.

We use a fuzzy inference system to combine the outputs for the three classifiers (see Fig. 7). The characteristics for this FIS are mamdani type with centroid defuzification, 15 inputs with three trapezoid functions. The first five inputs represent the matrix of memberships of Fuzzy KNN classifier. The second five represent the matrix of activations for the Multi Layer Perceptron with the learning method Gradient Descent with momentum Backpropagation. The last five are the matrix of activations of the third classifier Multi Layer Perceptron with Scaled Conjugate Gradient Backpropagation. We have 120 rules and 5 outputs with three trapezoid functions. The outputs represent the class Normal, LBBB, RBBB, PVC, Fusion Paced and Normal (see Fig. 8). In what follows, we present a detailed account of each of these experimental tests.

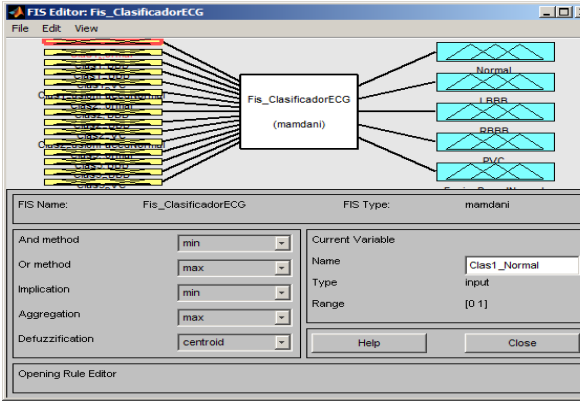
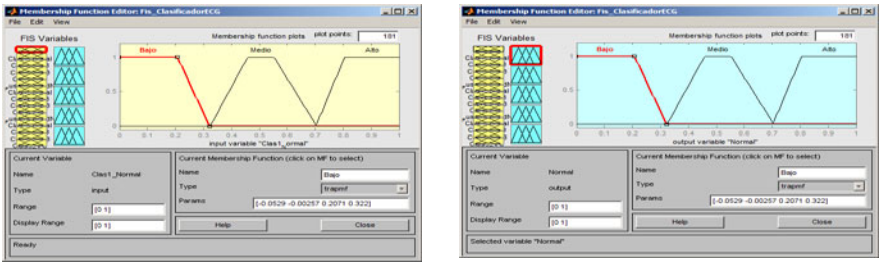


Fig. 7. Fuzzy Inference System, mamdani type.



(a) (b)

Fig. 8. Fuzzy Inference System, (a) The inputs, (b) The outputs.

Table I. Results for cross validation k-fold 10 of fuzzy KNN, K=3, first classifier.

Class	Normal	LBBB	RBBB	PVC	FPN
Normal	**68**	1	0	0	1
LBBB	0	**68**	0	2	0
RBBB	1	1	**67**	0	1
PVC	0	3	1	**64**	2
FPN	1	1	1	4	**63**

CLASSIFICATION RATE= 94.28%

4.3 Classifiers

4.3.1 First Classifier: Fuzzy K-Nearest Neighbors

In the results for the first classifier we found that the best result in cross validation k fold of 10 was with k = 3 nearest neighbors being 94.28% of classification rate are summarized in Table I. We used this parameter in the test and the classification rate was 95.33% see Table II.

Table II. Results for test of fuzzy KNN, K=3, first classifier.

Class	Normal	LBBB	RBBB	PVC	FPN
Normal	**30**	0	0	0	0
LBBB	0	**28**	0	2	0
RBBB	0	1	**29**	0	0
PVC	0	1	0	**28**	1
FPN	0	0	1	1	**28**

CLASSIFICATION RATE= 95.33%

4.3.2 Second Classifier: Multi Layer Perceptron with Gradient Descent with Momentum Backpropagation

In the results for the second classifier we found that the best result in cross validation k fold of 10 was with 150 neurons in the hidden layer being 92.85% of classification rate are summarized in Table III. We used this architecture in the test and the classification rate was 96.67% see Table IV.

Table III. Results for cross validation k-fold 10 of MLP gradient descent with momentum backpropagation, 150 neurons hidden layer, second classifier.

Class	Normal	LBBB	RBBB	PVC	FPN
Normal	**70**	0	0	0	0
LBBB	0	**67**	2	1	0
RBBB	1	2	**66**	0	1
PVC	0	7	0	**60**	3
FPN	0	2	3	3	**62**

CLASSIFICATION RATE= 92.85%

4.3.3 Third Classifier: Multi Layer Perceptron with Scaled Conjugated Gradient Backpropagation

In the results for the third classifier we found that the best result in cross validation k fold of 10 was with 100 neurons in the hidden layer being 93.62% of classification rate are summarized in Table V. We used this architecture in the test and the classification rate was 97.33% see Table VI.

Table IV. Results for test of MLP gradient descent with momentum backpropagation, 150 neurons hidden layer, second classifier.

Class	Normal	LBBB	RBBB	PVC	FPN
Normal	**30**	0	0	0	0
LBBB	0	**30**	0	0	0
RBBB	0	1	**29**	0	0
PVC	0	1	0	**28**	1
FPN	0	0	1	1	**28**

CLASSIFICATION RATE= 96.67%

Table V. Results for cross validation k-fold 10 of MLP scaled conjugated gradient backpropagation, 100 neurons hidden layer, third classifier.

Class	Normal	LBBB	RBBB	PVC	FPN
Normal	**69**	0	0	0	1
LBBB	0	**64**	2	3	1
RBBB	0	2	**67**	0	1
PVC	0	5	0	**60**	5
FPN	0	0	0	3	**67**

CLASSIFICATION RATE= 93.62

4.3.4 Fuzzy Inference System

Finally with the fuzzy inference system mentioned in previous section we combine the outputs per each classifier, and the result of classification rate was 98% see Table VII.

4.4 Statistical Analysis

Below are the results of statistical analysis to compare the error rate of the three classification methods, this analysis was done with software named statdisk[], these results are:

- Fuzzy K-Nearest Neighbor VS MLP with Gradient Descent with momentum Backpropagation.
- Fuzzy K-Nearest Neighbor VS MLP Scaled Conjugated Gradient Backpropagation.
- MLP with Gradient Descent with momentum Backpropagation VS MLP Scaled Conjugated Gradient Backpropagation.

Table VI. Results for test of MLP scaled conjugated gradient backpropagation, 100 neurons hidden layer, third classifier.

Class	Normal	LBBB	RBBB	PVC	FPN
Normal	**30**	0	0	0	0
LBBB	0	**29**	0	0	1
RBBB	0	1	**29**	0	0
PVC	0	0	0	**30**	0
FPN	0	0	0	2	**28**

CLASSIFICATION RATE= 97.33%

Table VII. Results for fuzzy inference system combining the outputs of the classifiers.

Class	Normal	LBBB	RBBB	PVC	FPN
Normal	**30**	0	0	0	0
LBBB	0	**30**	0	0	0
RBBB	0	1	**29**	0	0
PVC	0	0	0	**30**	0
FPN	0	0	1	1	**28**

CLASSIFICATION RATE= 98%

Table VIII. Results of the hypotesis test of Fuzzy K-Nearest Neighbor VS MLP with Gradient Descent with momentum Backpropagation.

Petición: $\mu = \mu(hyp)$
Tamaño muestra, n: 10
Diferencia de medias, d: -1.428571
Diferencia St Dev, sd: 4.312088
Prueba estadística, t: -1.0476
Critical t: ±2.2622
Valores-P: 0.3221
95% Intervalo de confianza:
$-4.51325 < \mu < 1.656108$
No se Rechazó la hipótesis nula
Las muestra no proveen suficiente evidencia para rechazar la petición

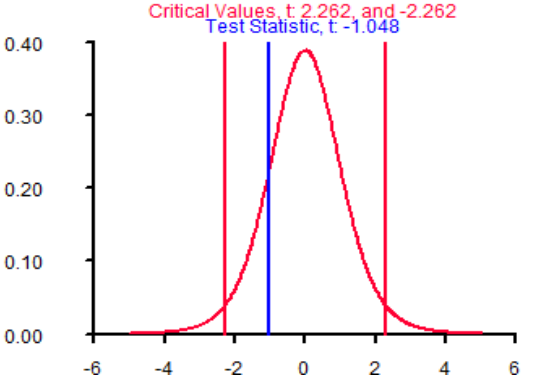


Fig. 9. Hypotesis Test of Fuzzy K-Nearest Neighbor VS MLP with Gradient Descent with momentum Backpropagation.

Table IX. Results of the hypotesis test of Fuzzy K-Nearest Neighbor VS MLP Scaled Conjugated Gradient Backpropagation.

Petición: $\mu = \mu(hyp)$
Tamaño muestra, n: 10
Diferencia de medias, d: -0.8571429
Diferencia St Dev, sd: 3.82141
Prueba estadística, t: -0.7093
Critical t: ±2.2622
Valores-P: 0.4961
95% Intervalo de confianza:
$-3.590812 < \mu < 1.876526$
No se Rechazó la hipótesis nula
Las muestra no proveen suficiente evidencia para rechazar la petición

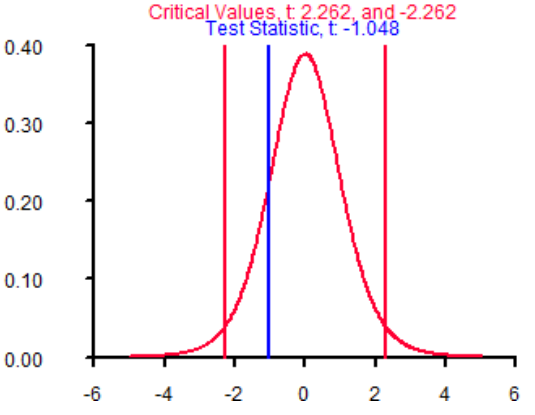


Fig. 10. Hypotesis Test of Fuzzy K-Nearest Neighbor VS MLP Scaled Conjugated Gradient Backpropagation.

Table X. Results of the hypotesis test of MLP with Gradient Descent with momentum Backpropagation VS MLP Scaled Conjugated Gradient Backpropagation

Petición: $\mu = \mu(hyp)$
Tamaño muestra, n: 10
Diferencia de medias, d: 0.5714286
Diferencia St Dev, sd: 2.950844
Prueba estadística, t: 0.6124
Critical t: ±2.2622
Valores-P: 0.5554
95% Intervalo de confianza:
$-1.539476 < \mu < 2.682333$
No se Rechazó la hipótesis nula
Las muestra no proveen suficiente evidencia para rechazar la petición

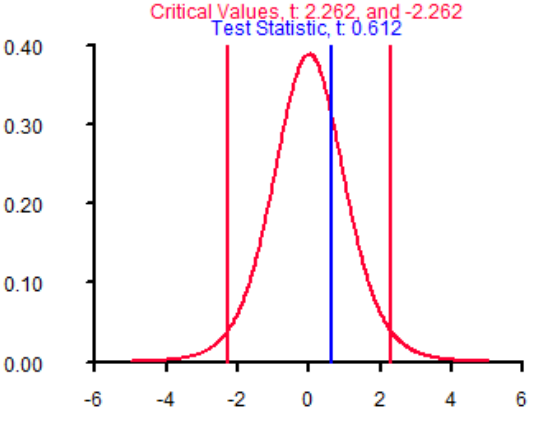


Fig. 11. Hypotesis Test of MLP with Gradient Descent with momentum Backpropagation VS MLP Scaled Conjugated Gradient Backpropagation.

4.4.1 Fuzzy K-Nearest Neighbor VS MLP with Gradient Descent with Momentum Backpropagation

The Table VIII shows the results of the statistical analysis to comparison of Fuzzy K-Nearest Neighbor VS MLP with Gradient Descent with momentum Backpropagation and the graph, Figure 9.

4.4.2 Fuzzy K-Nearest Neighbor VS MLP Scaled Conjugated Gradient Backpropagation

The Table IX shows the results of the statistical analysis to comparison of Fuzzy K-Nearest Neighbor VS MLP Scaled Conjugated Gradient Backpropagation and the graph, Figure 10.

4.4.3 MLP with Gradient Descent with Momentum Backpropagation VS MLP Scaled Conjugated Gradient Backpropagation.

The Table X shows the results of the statistical analysis to comparison of MLP with Gradient Descent with momentum Backpropagation VS MLP Scaled Conjugated Gradient Backpropagation and the graph, Figure 11.

5 Conclusions

Based on the performed experiments we noticed that the three classifiers used we achieved good results individually. But by combining their outputs using the Mamdani Fuzzy Inference System with appropriate membership functions and rules we achieved a very high classification rate of 98%. We applied the hypothesis test to compare the error rate of the three classification methods and we found that not exist enough statistical evidence to reject the nulls hypothesis in all made comparison. Therefore, for this reason we can't mention that classifier is best.

References

[1] Acharya, Kumar, Bhat: Classification of cardiac abnormalities using heart rate signals. Medical & Biological Enginnering & Computing (2004)
[2] Alzate, A., Giraldo, E.: Clasificación de Arritmias utilizando ANFIS, Redes Neuronales y Agrupamiento Substractivo (2006)
[3] Kulkarni, A.: Computer Vision and Fuzzy-Neural Systems, PH PTR
[4] Azam, F.: Biologically Inspired Modular Neural Networks, Electrical and Computer Engineering. Blacksburg, Virginia (2004)
[5] Bishop, C.M.: Neural Networks for Pattern Recognition. Clarendon Press, Oxford (1996)
[6] del Brío, B.M., Molina, A.S., Neuronales, R., Borrosos, S.: 3ª Edición, Alfaomega Ra-Ma (2007)
[7] Anuradha, B., Suresh Kumar, K., Veera Reddy, V.C.: Classification of Cardiac Signals using Time Domain Methods (2008)
[8] Anuradha, B., Veera Reddy, V.C.: ANN for Classification of Cardiac Arrhythimias (2008)

[9] Chuang, C.: Using Discrete Wavelet Transform of ECG Signals for Personal Identity Verification (2005)
[10] Patra, D., Das, M.K., Pradhan, S.: Integration of Fcm, Pca and Neural Networks for Classification of Ecg Arrhythmias (2009)
[11] Fahlman, S.: Faster Learning Variations of Backpropagation: An Empirical Study. In: Touretzky, D.S., Hinton, G.E., Sejnowski, T.J. (eds.) Proceedings of the 1988 Connectionist Models Summer School. Morgan Kaufmann Publishers, Los Altos (1988)
[12] Freeman, J.A.: Simulating Neural Networks with Mathematica. Addison-Wesley, Reading (1994)
[13] Clifford, G., Azuaje, F., McSharry, P.: Advanced Methods and Tools for ECG Data Analysis. Engineering Medicine & Biology. Artech House, Boston-London (2006)
[14] Heart Health, National Geographic, http://yourtotalhealth.ivillage.com/
[15] Nabney, I.T., Evans, D.J., Tenner, J., Gamlyn, L.: Benchmarking Beat Classification Algorithms
[16] Pan, J., Tompkins, W.J.: A real-time QRS detection algorithm. IEEE Trans. Biomed. Eng. (1985)
[17] Keller, J.M., Gray, M.R., Givens Jr., J.A.: A Fuzzy K-Nearest Neighbor Algorithm (1985)
[18] Jang, J.-S.R., Sum, C.-T., Mizutani, E.: Neuro-Fuzzy and Soft Computing. Prentice-Hall, Englewood Cliffs (1997)
[19] Khadra, Al-Fahoum, Al-Nashash: Detection of life-threatening cardiac arrhythmias using wavelets transformation. Med. Biol. Eng. Comput. (1997)
[20] Barbosa, L., Kleisinger, G.H., Valdez, A.D., Monzón, J.E.: Utilización del Modelo Kohonen y del Perceptrón Multicapa para detectar Arritmias Cardiacas (2001)
[21] Tsipouras, M., Fotiadis, D.: An Efficient System for the Detection of Arrhythmic Segments in ECG Recordings based on non-Linear Features of the RR Interval Signal (September 2003)
[22] Engin, M.: ECG Beat Classification using neuro-fuzzy network. Elsevier, Amsterdam (2004)
[23] Cepek, M., Chudácek, V., Petrik, M., Geogoulas, G., Stylios, C., Lhotská, L.: Comparison of Inductive Modeling Method to other Classification Methods for Holter ECG
[24] MIT-BIH Arrhythmia Database. PhysioBank, Physiologic Signal Archives for Biomedical Research, http://www.physionet.org/physiobank/database/mitdb/
[25] O' Dwyer, M., de Chazal, P., Reilly, R.B.: Beat Classification for Use in Arrhythmia Analysis (2000)
[26] Maglaveras, N., Stamkopoulos, T., Diamantaras, K., Pappas, C., Strintzis, M.: ECG pattern recognition and classification using non-linear transformations and neural networks (1998)
[27] Belgacem, N., Chikh, M.A., Bereksi Reguig, F.: Supervised Classification of ECG using Neural Networks
[28] Neural Network for Mathematica, Wolfram Research Inc., Chicago, IL (2003)
[29] de Chazal, P., Reilly, R.B.: Automatic Classification of ECG Beats using Wareform Shape and Heart Beat Interval Features (1998)
[30] Ceylan, R., Ozbay, Y., Karlik, B.: A novel approach for classification of ECG arrhythmias: Type-2 fuzzy clustering neural network (2009)

[31] Rogal, S., Paraiso, E., Kaestner, C., Figueredo, M., Neto, A.: Agrupamiento de Arritmias Cardiacas Utilizando ART2 (2008)
[32] Reghav, S., Amit, K.: Fractal Feature Based ECG Arrhythmia Classification (November 2008)
[33] Mehta, S.S., Lingayat, N.S.: Identification of QRS Complexes in 12-lead Electrocardiogram. Elsevier, Amsterdam (2009)
[34] Ari, S., Saha, G.: In Search of an optimization technique for Artificial Neuronal Network to Classify abnormal heart sounds (2009)
[35] Samarasinghe, S.: Neural Networks for Applied Science and Engineering. Auerback Publications (2007)
[36] Sun, Y., Chan, K.: Arrhythmia detection and recognition in ECG Signals using nonlinear techniques, Ann. Biomed. Eng. (2000)
[37] Yu, S.-N., Chen, Y.-H.: Selection of Higher Order Subband Features for ECG Beat Classification (2008)
[38] Palreddy, S.: ECG Beats Database Description, PH. D. University of Wisconsin (1996)
[39] Werbos, P.J.: The Roots of Backpropagation Forom Ordered Derivatives to Neuronal Networks and Political Forescasting. Wiley Series on Adaptive and Learning System for Signal Processing Communication and Control. Wiley, New York (1994)
[40] Ozbay, Y., Ceylan, R., Karlik, B.: A Fuzzy Clustering Neuronal Network Architecture for Classification of ECG Arrhtymias (January 2005)

A Comparative Study of Blog Comments Spam Filtering with Machine Learning Techniques

Christian Romero, Mario Garcia-Valdez, and Arnulfo Alanis

Tijuana Institute of Technology, Tijuana México
jcrmerohdz@gmail.com, mgarcia@tectijuana.edu.mx,
aalanis@tectijuana.edu.mx

Abstract. In this paper we compare four machine learning techniques for spam filtering in blog comments. The machine learning techniques are: Naïve Bayes, K-nearest neighbors, neural networks and support vector machines. In this work we used a corpus of 1021 blog comments with 67% spam, the results of the filtering using 10 fold cross-validation are presented.

1 Introduction

Spam in general is a problem that in recent years has caused a tremendous impact on the Internet. Spam can be defined as unwanted messages; spam is not only affecting E-mails, is also found in blogs, mobile phone short messages (SMS), and forum comments.

The growing popularity of Internet searching as the main access point to the Web has increased the benefits of achieving top rankings in popular search engines, especially for commercially oriented Web sites. Combined with the success of link analysis methods such as PageRank[20], this led to rapid growth in link spamming creating links which are "present for reasons other than merit"[20].

2 Background and Basic Concepts

2.1 Naïve Bayes

Suppose that we knew exactly, that the word CASINO could never occur in a legitimate message. Then when we saw a message containing this word, we could tell for sure that it was spam. This simple idea can be generalized using some probability [1]. We have two categories (classes): S (spam) and L (legitimate mail), and there is a probability distribution of messages (or, more precisely, the feature vectors we assign to messages) corresponding to each class: $P(c|x)$denotes the probability of obtaining a message with feature vector x from class c. Usually we know something about these distributions (as in example above, we

P. Melin et al. (Eds.): Soft Comp. for Recogn. Based on Biometrics, SCI 312, pp. 57–72.
springerlink.com

knew that the probability of receiving a message containing the word CASINO from the category L was zero). What we want to know is, given a message x, what category c "produced" it. That is, we want to know the probability $P(c|x)$. And this is exactly what we get if we use the Bayes' rule[1]:

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} = \frac{P(x|c)P(c)}{P(x|S)P(S)+P(x|L)P(c)} \quad (1)$$

where $P(x)$ denotes the a-priori probability of message x and $P(c)$ — the apriori probability of class c (i.e. the probability that a random message is from that class). So if we know the values $P(c)$ and $P(c|x)$ (for C 2 {S,L}), we may determine $P(c|x)$, which is already a nice achievement that allows us to use the following classification rule: If $P(S|x) > P(L|x)$ (that is, if the a-posteriori probability that x is spam is greater than the a-posteriory probability that x is non-spam), classify x as spam, otherwise classify it as legitimate comment. This is the so-called maximum a-posteriori probability (MAP) rule. Using the Bayes' formula we can transform it to the form [1]:

$$\text{If} \frac{P(x|S)}{P(x|L)} > \frac{P(L)}{P(S)} \text{classify x as spam,}$$

Otherwise classify it as legitimate comment.

Now that we have the theory of the optimal classifier, let us consider the not-so-simple practical application of the idea. In order to construct Bayesian classifier for spam detection we must somehow be able to determine the probabilities P(x | c) and P(c) for any x and c.

It is clear that we can never know them exactly, but we may estimate them from the training data. For example, P(S) may be approximated by the ratio of the number of spam messages to the number of all messages in the training data. Estimation of P(x | c) is much more complex and actually depends on how we choose the feature vector x for message m. Let us try the simplest case of a feature vector with a single binary attribute that denotes the presence of a certain word w in the message. That is, we define the message's feature vector xw to be, say, 1 if the word w is present in the message, and 0 otherwise [1][19].

• Training

1. Calculate estimates for $P(c), P(X_w = 1|c), P(X_w = 0|c)$ $(for\ c = (S, L)$ from the training data.

2. Calculate $P(c|X_w = 0), P(c|X_w = 1)$ using the Bayes' rule.

3. Calculate $\Lambda(X_w) for\ X_w = 0,1$, calculate $\lambda = \frac{P(L)}{P(S)}$. Store these 3 values.

• Classification

1. Given a message $\boldsymbol{m}$ determine, X_w retrieve the stored value for $\Lambda(X_w)$ and use the decision rule to determine the category of message $\boldsymbol{m}$.

Now this classifier will hardly be any good because it bases its decisions on the presence or absence of one word in a message. We could improve the situation if our feature vector contained more attributes. Let us fix several words w1, w2, . . . , wm and define for a message m its feature vector as x =(x1, x2, . . . , xm) where xi is equal to 1 if the word wi is present in the message, and 0 otherwise. If we followed the algorithm described above, we would have to calculate and store the values of $\Lambda(X)$ for all possible values of x (and thereare 2m of them). This is not feasible in practice, so we introduce an additional assumption: we assume that the components of the vector x are independent in each class. In other words, the presence of one of the words wi in a message does not influence the probability of presence of other words. This is a very wrong assumption, but it allows us to calculate the required probabilities without having to store large amounts of data, because due to independence

$$P(x|C) = \prod_{i=1}^{m} P(x_i|c) \qquad \Lambda(X) = \prod_{i=1}^{m} \Lambda_i(x_i) \tag{2}$$

So the algorithm presented above is easily adapted to become the Naïve Bayesian classifier. The word "naïve" in the name expresses the naïveness of the assumption used.

2.2 K Nearest Neighbors

The most basic instance-based method is the k nearest neighbors machine learning technique. It is a very simple technique to classify documents and to show very good performance on text categorization task [17]. The procedure of k-NN technique which is employed to blog comments classification is the fallows: Given a new comment, the distance between the comment and all samples in the training set is calculated. The distance used in practically all the nearest-neighbor classifiers is the Euclidian distance. With the distance calculated, the samples are ranked according to the distance. Then the k samples which are nearest to the new comment are used in assigning a classification to the case.

2.3 Neural Networks

The study of artificial neural networks (ANN) was inspired by attempts to simulate biological neural system. The human brain consists primarily of nerve cells called neurons, linked together with other neurons via strands of fiber called axons. Axons are used to transmit nerve impulse from one neuron to another whenever the neurons are stimulated.

The basic element of the brain is a natural neuron; similarly, the basic element of every neural network is an artificial neuron, or simply neuron. That is, a neuron is the basic building block for all types of neural networks [5].

A neuron is an abstract model of a natural neuron, as illustrated in Figs. 1. As we can see in these figures, we have inputs x1, x2, ..., xm coming into the neuron. These inputs are the stimulation levels of a natural neuron. Each input xi is multiplied by its Fig. 1. A neuron model that retains the image of a natural neuron [5].

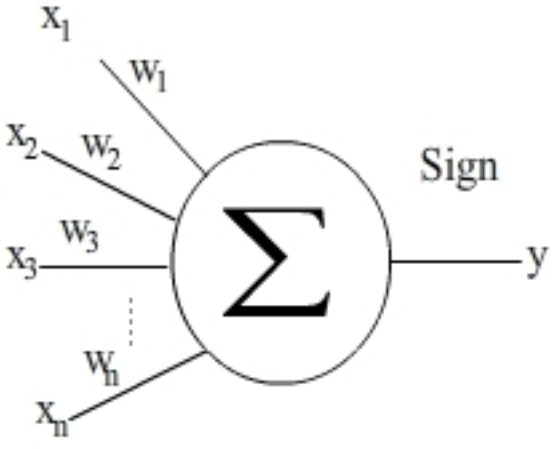


Fig. 1. Artificial neuron.

Corresponding weight w_i, then the product $x_i w_i$ is fed into the body of the neuron. The weights represent the biological synaptic strengths in a natural neuron. The neuron adds up all the products for I= 1, m. The weighted sum of the products is usually denoted as net in the neural network literature, so we will use this notation. That is, the neuron evaluates $net = x_1 w_1 + x_2 w_2 +, \ldots, + x_n w_n$. In mathematical terms, given two vectors $x = x_1 x_w, \ldots, x_n$ and $w = w_1 w_w, \ldots, w_n$, net is the dot (or scalar) product of the two vectors, $x \cdot w = x_1 + x_2 w_2 +, \ldots, + x_n w_n$. Finally, the neuron computes its output y as a certain function of net, i.e., y = f(net) [4]. This function is called the activation (or sometimes transfer) function. We can think of a neuron as a sort of black box, receiving input vector x then producing a scalar output y. The same output

Value y can be sent out through multiple edges emerging from the neuron [1][19].

2.4 Support Vector Machines

This method has its root in statistical learning theory and has shown promising empirical results in many practical applications, form handwritten digit recognition to text categorization. SVM works very well with high-dimensional data and avoids the curse of dimensionality problem [19].

Linear SVM: Separable Case.
A linear SVM is a classifier that searches for a hyperplane with the largest margin, which is why it is often known as a maximal margin classifier. To understand how SVM learns such a boundary, we begin with some preliminary discussion about the about the decision boundary and margin of a linear classifier [8].

Linear decision boundary.

Consider a binary classification problem consisting of N training examples. Each example is denoted by a tuple (x_i, y_i) $(i = 1,2, \dots, N)$ where $x_i = (x_{i1}, x_{i2}, \dots, x_{id})^T$ corresponds to the attribute set for the i^{th} example. By convention, let $y_i \in \{-1,1\}$ denote its class label. The decision boundary of a liner classifier can be written in the following form:

$$w \cdot x + b = 0 \tag{3}$$

Where w and b are parameters of the model.

A decision boundary that bisects the training set the training example into their respective classes. Any decision boundary most satisfies equation 1. For example, if x_a and x_b are two points located on the decision boundary, then

$$w \cdot x_a + b = 0 \tag{4}$$

$$w \cdot x_b + b = 0 \tag{5}$$

Subtracting the two equations will yield the following:

$$w \cdot (x_b - x_a) = 0, \tag{6}$$

Where $x_b - x_a$ is a vector parallel to the decision boundary and is directed from x_a to x_b. Since the dot product is zero, the direction for w must be perpendicular to the decision boundary.

For any square x_s located above the decision boundary, we can show what

$$w \cdot x_s + b = k \tag{7}$$

Where $k > 0$. Similarly, for any circle x_c located below the decision boundary, we can show that

$$w \cdot x_s + b = k'$$

Where $k' < 0$. If we label all the squares as class +1 and all circles as class -1, then we can predict the class label y for any test examples z in the following way:

$$y = \begin{cases} 1, & if\ w \cdot z + b > 0; \\ -1, & if\ w \cdot z + b < 0. \end{cases}$$

Margin of Linear classifier

Consider the square and circle that closest to the decision boundary. Since the square is located above the decision boundary, it must satisfy equation number for some positive value k, whereas the circle must satisfy equation number for some negative values k'. We can rescale the parameters w and b of the decision boundary so that the two parallel hyperplanes b_{i1} and b_{i2} can be express as follows:

$$b_{i1}: w \cdot x + b = 1, \tag{8}$$

$$b_{i2}: w \cdot x + b = -1. \tag{9}$$

The margin of the decision boundary is given by the distance between these two hyperplanes. To compute the margin, let x_1 be a data point located on b_{i1} and x_2 be a data point on b_{i2}. Upon substituting these points into equations number and number, the margin d can be computed by subtracting the second equation from the first equation:

$$\begin{aligned} w \cdot (x_1 - x_2) &= 2 \\ \|w\| \times d &= 2 \\ \therefore d &= \frac{2}{\|w\|}. \end{aligned} \tag{10}$$

Learning a linear SVM Model

The training phase of SVM involves estimating the parameters w and b of the decision boundary from the training data [19]. The parameters must be chosen in such a way that the following two conditions are met:

$$w \cdot x_i + b \geq 1 \; if \; y_i = 1, \tag{11}$$

$$w \cdot x_i + b \leq -1 \; if \; y_i = -1. \tag{12}$$

These conditions impose the requirements that all training instances from class $y = 1$ must be located on or above the hyperplane $w \cdot x + b = 1$, while those instances from class $y = -1$ must be located on or below the hyperplane $w \cdot x + b = -1$. Both inequalities can be summarized in a more compact form as follows:

$$y_i(w \cdot x_i + b) \geq 1, i = 1,2, \ldots, N.$$

Although the preceding conditions are also applicable to any linear classifiers, SVM imposes an additional requirement that the margin of its decision boundary must be maximal. Maximizing the margin, however, is equivalent to minimizing the following objective function:

$$f(w) = \frac{\|w\|^2}{2}. \tag{13}$$

Definition 5.1 (Linear SVM: Separable Case). The learning task in SVM can be formalized as the following constrained optimization problem:

$$\min \frac{\|w\|^2}{2} \tag{14}$$

$subject\ to\ y_i(w \cdot x_i + b) \geq 1, \ i = 1,2, \ldots, N.$ Since the objective function is quadratic and the constraints are linear in the parameters w and b, this is known as a convex optimization problem, which can be solved using the standard Lagrange multiplier method. Following is a brief sketch of the main ideas for solving the optimization problem.

First, we must rewrite the objective function in a form that takes into account the constraints imposed on its solutions. The new objective function is known as the Lagrangian for the optimization problem:

$$Lp = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \lambda_i (y_i(w \cdot x_i + b) - 1), \quad (15)$$

Where the parameters λ_i are called the Lagrange multipliers. The first term in the lagrangian is the same as the original objective function, while the second term captures the inequality constraints. To understand why the objective function must be modified, consider the original objective function given in Equation number (15). It is easy to show that the function is minimized when w = 0, a null vector whose components are all zeros. Such a solution, however, violates the constraints given in definition number because there is no feasible solution for b. The solutions for w and b are infeasible if they violate the inequality constraints; i f $y_i(w \cdot x_i + b) - 1 < 0$. The Lagrangian given in Equation number (15) incorporates this constraint by subtracting the term from its original objective function. Assuming that $\lambda_i \geq o$, it is clear that any infeasible solution may only increase the value of the Lagrangian.

To minimize the Lagrangian, we must take the derivative of Lp with respect to w and b an set them to zero:

$$\frac{\partial Lp}{\partial w} = 0 \Longrightarrow w = \sum_{i=1}^{N} \lambda_i y_i x_i, \quad (16)$$

$$\frac{\partial Lp}{\partial b} = 0 \Longrightarrow \sum_{i=1}^{N} \lambda_i y_i = 0. \quad (17)$$

Because the Lagrange multiplier are unknown, we still cannot solve for w and b. if definition number contains only equality instead of inequality constraints, then we can use the N equations from equality constraints along with equations number and number to find the feasible solution for w, b, and λ_i. Note that the Lagrange multipliers for equality constraints are free parameters that can take any values.

On way to handle the inequality constraints is to transform them into a set of equality constraints [9]. This is possible as long as the Langrange multipliers are restricted to be non-negatives. Such transformation leads to the following constraints on the Lagrange multipliers, which are known as the Karush-Kuhn-Tuker (KKT) conditions.

$$\begin{aligned} &\lambda_i \geq 0, \\ &\lambda_i[y_i(w \cdot x_i + b) - 1] = 0. \end{aligned} \quad (18)$$

At first glance, it may seem that there are as many Lagrange multipliers as there are training instances. It turns out that many of the Lagrange multipliers become zero after applying the constraint given in equation number (18). The constraint states that the Lagrange multiplier λ_i must be zero unless the training instances

x_i satisfies the equation $y_i(w \cdot x_i + b) = 1$. Such training instance, with $\lambda_i > 0$, lies along the hyperplanes b_{i1} or b_{i2} and is known as a support vector. Training instances that do not reside along these hyperplanes have $\lambda_i = 0$. Equations number (17) and number (18)also suggest that the parameters w and b, which define the decision boundary, depend only on the support vectors [8].

Solving the preceding optimization problem is still quite a daunting task because it involves a large number of parameters: w , b, and λ_i. The problem can be simplified by transforming the Lagrangian into a function of the Lagrange multipliers only (this is known as the dual problem). To do this, we first substitute Equation number (18) and number into Equation number (15). This will lead to the following dual formulation of the optimization problem:

$$L_D = \sum_{i=1}^{N} \lambda_i \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j x_i \cdot x_j. \quad (19)$$

The key differences between the dual and primary Lagrangians are as follows:

1. The dual Lagrangian involves only the Lagrange multipliers and the training data, while the primary Lagrangian involves the Lagrange multipliers as well as parameters of the decision boundary. Nevertheless, the solutions for both optimization problems are equivalent.
2. The quadratic term in Equation 19 has a negative sign, which means that the original minimization involving the primary Lagrangian, Lp, has turned into a maximization problem involving the dual Lagrangian, L_D.

For large data sets, the dual optimization problem can be solved using numerical techniques such as quadratic programming, a topic that is beyond the scope of this paper. Once the λ_i are found, we can use Equations 17 and 18 to obtain the feasible solution for w and b. the decision boundary can be expressed as follows:

$$\left(\sum_{i=1}^{N} \lambda_i y_i x_i \cdot x \right) + b = 0. \quad (20)$$

B is obtained by solving Equation 17 for the support vectors. Because the $\lambda_i's$ are calculated numerically and can have numerical errors, the value computed for b may not be unique. Instead it depends on the support vector used in Equation 17. In practice, the average value for b is chosen to be parameter on the decision boundary.

3 Architecture

In this section we describe the process of the way we work in our architecture, the figure 2 represent our architecture.

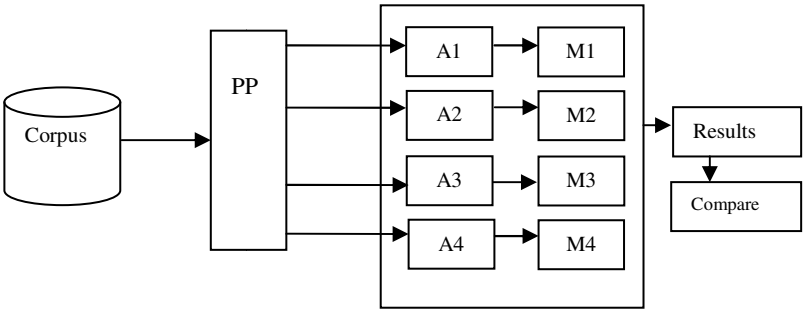


Fig. 2. Architecture

In figure 4 we see the architecture that we used to do this study for the machine learning techniques.

3.1 Corpus

In this figure we have a corpus provided by ILPS (Information an language processing Systems). This corpus is a collection of 50 pages of logs, with 1024 comments, which we know its classification and this classification is 67% of these comments are spam[11].

3.2 Preprocessing PP

For our preprocessing we use a statistical measure based in Bayes theory for our first approach. For the rest of the machine learning techniques we use the TF-IDF measure.

TF-IDF that it is a weight that is usually used in text mining. This weight is a statistical measure used to evaluate how important a word to a document in the corpus and is given by the following equations.

$$tf_{t,j} = \frac{n_{t,j}}{\sum_k n_{k,j}}$$

$n_{i,j}$ Is the number of occurrences of a term considered in the document and the denominator is the sum of occurrences of all terms.

$$idf_t = log \frac{|D|}{|\{d: t_i \in d\}|}$$

$|D|$ Is the total number of documents in the body.

$|\{d: t_t \in d\}|$ Is the number of documents where the term appears.

3.3 A1 Naïve Bayes

The A1 block is corresponding to the machine learning technique Naïve Bayes, here is when the technique receives the data from the preprocessing block, and processes this information.

3.4 A2 Support Vector Machines

The A2 is block corresponding to the machine learning technique support vector machines, here in this block the technique receive the data from the preprocessing block, and process this information.

3.5 A3 Neural Networks

The A3 block is corresponding to the machine learning technique neural networks, here in this block the technique receive the data from the preprocessing block, and process this information.

3.6 A4 K-nearest Neighbors

The A4 block is corresponding to the machine learning technique k-nearest neighbors, here in this block the technique receive the data from the preprocessing block, and process this information.

3.7 Models

Each machine learning techniques has model, this models are for M1 to M4 where each one is for the corresponding machine learning, M1 block is the model of Naïve Bayes, M2 block is the model of the support vector machine technique, M3 block is the model of the neural networks technique and finally we have the M4 model that is corresponding to the k-nearest neighbor technique.

4 Results

Our experiments where done by calculating the false positives, false negatives, true positives and true negatives. We use cross validation k-fold of 10 for each machine learning technique, then we calculate the accuracy of machine learning technique and the error rate.

4.1 Results of A1 Naïve Bayes

Table 1 shows the results obtained with the machine learning technique Naive Bayes, we represent the runs of the k-fold cross validation with their data and their respective tests:

• TP: True Positive .
• FP: False positive.
• FN: False negative.
• TN: True negative.
• Ac: Accuracy.
• ER: error rate.

Table I. Results for A1

I	TP	FP	FN	TN	Ac	ER
1	41	5	24	32	71.56%	28.43%
2	47	0	27	28	73.52%	26.47%
3	54	2	21	25	77.45%	22.54%
4	47	2	22	31	76.47%	23.52%
5	42	5	23	32	72.54%	27.49%
6	50	3	23	26	74.50%	25.49%
7	54	0	23	25	77.45%	22.54%
8	39	1	23	39	76.47%	23.52%
9	42	1	21	38	78.43%	21.56%
10	43	1	23	35	76.47%	23.52%

Below are the averages of the accuracy test, and the error rate.

• Average Accuracy: 75.49%.
• Average error rate: 24.51%.

4.2 Results of A2 Support Vector Machines

Table 2 shows the results obtained with the machine learning technique k-nearest neighbors, represent the runs of the k-fold cross validation with their data and their respective test:

• TP: True Positive.
• FP: False positive.
• FN: False negative.
• TN: True negative.
• Ac: Accuracy.
• ER: error rate.

Table II. Results for A2

I	TP	FP	FN	TN	Ac	ER
1	57	11	8	26	81.37%	18.63%
2	65	2	9	26	89.22%	10.78%
3	64	3	11	24	86.28%	13.72%
4	56	3	13	30	84.31%	15.69%
5	54	9	11	28	80.40%	19.60%
6	63	7	10	22	83.33%	16.67%
7	64	4	13	21	83.33%	16.67%
8	55	3	7	37	90.20%	9.80%
9	52	7	11	32	82.35%	17.65%
10	55	4	11	32	85.30%	14.70%

Below are the average of the accuracy test, and the error rate.

- Average Accuracy: 84.61%.
- Average error rate: 15.39%.

4.3 Results of A3 Neural Networks

Table 3 shows the results obtained with the machine learning technique neural networks represent the runs of the k-fold cross validation with their data and their respective test:

- TP: True Positive.
- FP: False positive.
- FN: False negative.
- TN: True negative.
- Ac: Accuracy.
- ER: error rate.

Table III. Results for A3

I	TP	FP	FN	TN	Ac	ER
1	57	14	8	23	78.43%	21.57%
2	63	3	11	25	86.27%	13.73%
3	64	3	11	24	86.27%	13.73%
4	63	13	6	20	81.37%	18.63%

Table III. (*continued*)

5	57	12	8	25	80.39%	19.61%
6	65	10	8	19	82.35%	17.65%
7	65	5	12	20	83.33%	16.67%
8	58	7	4	33	89.21%	10.79%
9	53	7	10	32	83.33%	16.67%
10	53	3	13	33	84.31%	15.69%

Below are the average of the accuracy test, and the error rate.

- Average Accuracy: 83.53%.
- Average error rate: 16.17%.

4.4 Results of A4 K-Nearest Neighbors

Table 4 shows the results obtained with the machine learning technique support vector machines represent the runs of the k-fold cross validation with their data and their respective test:

- TP: True Positive.
- FP: False positive.
- FN: False negative.
- TN: True negative.
- Ac: Accuracy.
- ER: error rate.

Table IV. Results For A4

I	TP	FP	FN	TN	Ac	ER
1	49	16	5	32	79.41%	20.58%
2	58	2	16	26	82.35%	17.64%
3	61	1	14	26	85.29%	14.70%
4	54	4	15	29	81.37%	18.62%
5	54	7	11	30	82.35%	17.64%
6	57	5	16	24	79.41%	20.58%
7	61	3	16	22	81.37%	18.62%
8	48	4	14	36	82.35%	17.64%
9	54	3	9	36	88.23%	11.76%
10	53	8	13	28	79.41%	20.58%

Below are the averages of the accuracy test, and the error rate.

- Average Accuracy: 82.15%.
- Average error rate: 17.84%.

4.5 Global Results

In table 5 we have the average percentage of the cross validation in each machine learning techniques that we use in this work and we represent this results graphically.

Table 5. Global results

Techniques	Accuracy	Error Rate
Naïve Bayes	75.49%.	24.51%.
k-Nearest Neighbor	82.15%.	17.84%
Neural Networks	83.53%.	16.17%.
Support Vector Machines	84.61%.	15.39%.

In Figure 3 we show a pictorial representation of the performance of the four methods that were applied in this work.

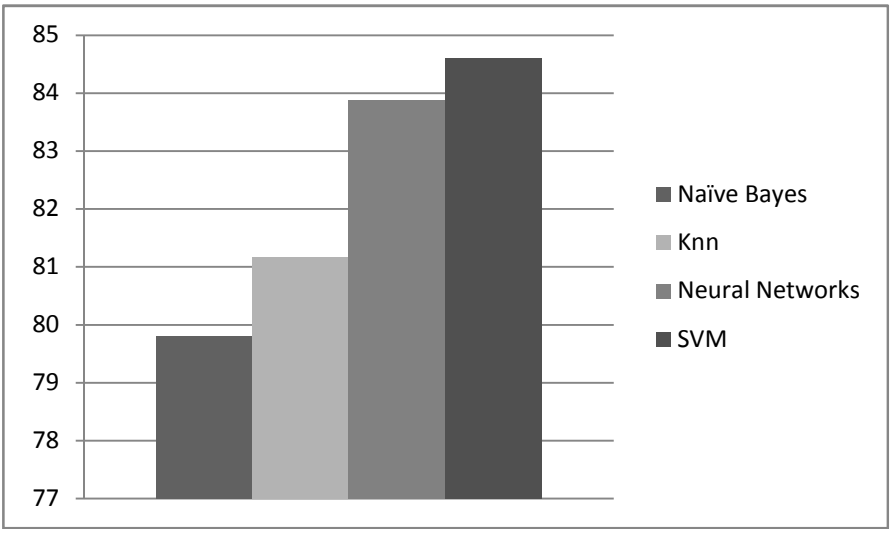


Fig. 3. Comparison of global results

But with these results we cannot say which of the methods are better, so we decided to make a statistical analysis for the neural networks and the support vector machines methods. Because in fact these methods are better in accuracy (see Figure 3).

4.6 Statistical Analysis Mean Match Pairs

In table 6 we show the results of our statistical analysis, this analysis is 10 rounds of the cross validation for each machine learning method the neural networks and the support vector machines. With that sample we get the error rate average mean of each cross validation ,in other words we have 10 error rate average mean in each method, then we proceed to do the analysis.

In figure 4 we show the representation of the t test that was performed.

Table 6. Neural Networks against SVM.

Claim:	$\mu = \mu(hyp)$
Sample size, n:	10
Mean diference, d:	1.5
St Dev, sd:	2.388104
Statistical test, t:	1.9863
Critical t:	±1.9727
P values:	0.0783
92% Interval of confidence:	$0.0102841 < \mu d < 2.989716$
Reject the null hypothesis	The sample provide evidence to reject the claim

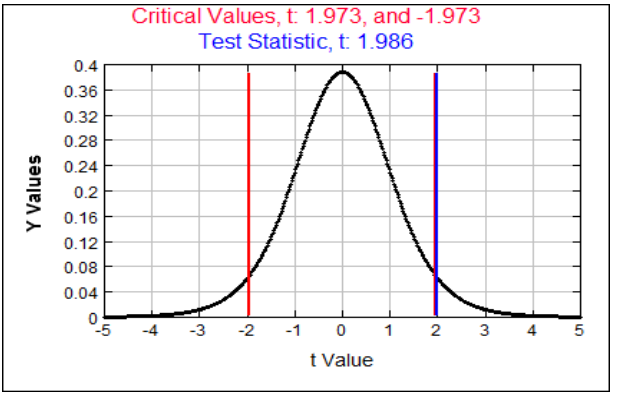


Fig. 4. Neural Networks against SVM

5 Conclusions

In conclusion we can say according to the results that the machine learning techniques that we compare in this work have a good performance for the spam filtering in blog comments particular case. We can mentioned that the support vector

machine technique have a better performance that the Naïve Bayes and the k-Nearest Neighbors techniques and a slightly better performance with respect to the neural works technique.

On the other hand we do not use a special architecture for the neural network and the support vector machines techniques and neither a feature extraction beside the text.

References

[1] Tretyakov, K.: Machine Learning Techniques in Spam Filtering. Institute of Computer Science, University of Tartu (2004)
[2] Aas, K., Eikvil, L.: Text categorization. A survey (1999), http://citeseer.ist.psu.edu/aas99text.html
[3] Cristianini, N., Shewe-Taylor, J.: An introduction to support Vector Machines and other Kernel Based Learning Methods. Cambridge University Press, Cambridge (2003)
[4] Kecman, V.: Learning and soft computing. The MIT Press, Cambridge (2001)
[5] Haykin, S.: Neural Networks: A Comprehensive Foundation. Practice Hall (1998)
[6] Androutsopoulos, I., et al.: Learning to filter Spam E-mail: A comparison of Naïve Bayesian and a Memory-Based Approach
[7] Androutsopoulos, I., et al.: An experimental comparison of Naïve Bayesian and Keywords-Based Anti-Spam filtering with Personal E-mail
[8] Cortes, C., Vapnik, V.: Support Vector Networks. Machine Learning (1995)
[9] Vladimir, N., Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
[10] Mishne, G., Carmel, D., Lempel, R.: Bocking Blog Spam with Language Model Disagreement
[11] Mishne, G.: Using Blogs Properties to Improve Retrieval
[12] Kolari, P., Finin, T., Joshi, A.: SVMs for the Blogsphere: Blog Identification and Splog Detection. In: AAAI Spring Symposium on Computational Approaches to Analysis Weblogs (2006)
[13] Cormack, G., Gomez, J.M., Puertas, E.: Spam Filterin For Shot Messages
[14] Holdens, S.: Spam Filters (2004), http://freshment.net/articles/view/964
[15] Cost, S., Salzberg, S.: A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features. Machine Learning (1992)
[16] Cover, T.M., Hart, P.E.: Nearest Neighbor Pattern Classification. Knowledge Based Systems (1995)
[17] Goldstain, M.: K-Nearest Neighbor Classification (1972)
[18] Bishop, C.M.: Neural Networks for Pattern Recognitions. Oxford University Press, U.K. (1995)
[19] Ning Tan, P., Steinbach, M., Kumar, V.: Introduction to Data Maning. Adison Wesley (2006)
[20] Arasu, A., Novak, J., Tomkins, A., Tomlin, J.: PageRank computation and the structure of the web: Experiments and algorithms. In: Proceedings of the Eleventh International World Wide Web Conference, Poster Track. Brisbane, Australia, pp. 107–117 (2002), http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.18.5264&rep=rep1&type=pdf

Distributed Implementation of an Intelligent Data Classifier

Victor J. Sosa-Sosa[1], Ivan Lopez-Arevalo[1], Omar Jasso-Luna[1], and Hector Fraire-Huacuja[2]

[1] Centro de Investigación y de Estudios Avanzados del IPN (CINVESTAV)
Ciudad Victoria, Tamaulipas, México.87130
{vjsosa,ilopez,jjasso}@tamps.cinvestav.mx
[2] Instituto Tecnológico de Ciudad Madero
1[0] de Mayo S/N, Col. Los Mangos. Ciudad Madero, Tamaulipas. México. 89440
hfraire@prodigy.net

Abstract. Industry, science and business applications need to manipulate a huge amount of data every day. Most of the time these data come from distributed sources and are analyzed trying to discover knowledge and recognize patterns using Data Mining techniques. Data classification is a technique that allows to decide if a set of data belongs to a group of information or not. Data classification requires putting all data together in a big centralized datasets. To congregate and analyze this dataset represents a very expensive task in terms of time, memory and bandwidth consuming. Nowadays, architectures for Distributed Data Mining have been developed trying to reduce computing and storage costs. This paper presents an approach to building a distributed data classifier which takes only metadata from distributed datasets avoiding the total access to the original data. Using only metadata reduces the computing time and bandwidth consumption required to build a data classifier.

Keywords: Data Mining, Global Classifier.

1 Introduction

Complex business and industrial-scientific applications require storing a huge amount of data everyday for its analysis. In most of the cases, these big organizations tend to build centralized dataset as data warehouses that store all information obtained from their geographically dispersed branches. These data are analyzed to discover patterns or tendencies that represent knowledge, which is an important asset for any organization. Many organizations are carrying out the analysis of data employing Data Mining (DM) techniques. There are some toolkits that implement different DM techniques. One of them is Weka [13], which is a widely used DM toolkit that contains a large collection of state-of-the-art machine learning algorithms written in Java. Mining huge centralized datasets using a

P. Melin et al. (Eds.): Soft Comp. for Recogn. Based on Biometrics, SCI 312, pp. 73–82.
springerlink.com

stand-alone approach requires powerful equipment with advanced computing resources. Distributed Data Mining (DDM) is considered a feasible strategy to distribute the heavy work among different remote nodes. Some DDM architectures are exposed in [2, 5, 8, 10]. Most of them propose to execute the DM tasks over remote nodes needing a complete authority for accessing private data from the participating nodes. Current architectures do not consider building data classifiers without having a global knowledge of the original data, or without accessing all the original data. This issue becomes very important when some organizations are not able to share their complete information. As an example, today it is possible to find some health care institutions wanting to detect some potential patients who could have a type of cancer based on his/her history data. These institutions sometimes do not have enough information to build a reliable data classifier that could help with this process. Even though some institutions have enough information for doing this work, they are not able to share it because of their privacy policies. This paper presents an architecture that allows this type of institutions to work together for building better data classifiers using global knowledge without accessing private data. This architecture is based on Web technologies and java components. The paper is organized as follows: Section 2 describes some of related work. Section 3 shows the architecture of our approach. Section 4 describes briefly a Distributed ID3 Classifier process. Section 5 presents our implementation. Finally, Section 6 shows an experimental evaluation and Section 7 concludes the paper with ongoing and future work.

2 Related Work

Data Mining (DM) can be defined as an infrastructure that uses a selection of different algorithms and statistical methods to find interesting and novel patterns within large datasets. The evolution in DM can be classified in three stages [8]. The first generation of tools provides users with a single DM algorithm operating on data stored in a local file. Examples include the use of classification algorithms such as C4.5 [7], clustering algorithms such as K-means [3]. Such tools were provided primarily as standalone executables, obtaining input from the command line or via a configuration file. The second generation of tools combined a collection of different algorithms for DM under a common framework, and enabled users to provide input from various data sources. Some of these tools are described below:

- Weka [13] contains tools for classification, regression, clustering, association rules, visualization, and pre-processing.
- Illimine [11] is another DM tool developed in C language. It built-ins algorithms in data cubing, association mining, sequential pattern mining, graph pattern mining, and classification.
- Rattle [12] is a DM tool based on the statistical language R [9].
- Rapid Miner [1,4] features more than 400 operators for DM in java which can be used merging some of them.

Subsequently, third generation tools started to address the limitations that are imposed by the closed world model. Some examples of third generation tools are:

- Grid Weka [2], essentially a modification to the Weka toolkit that enables the use of multiple computational resources when performing data analysis.
- WekaG [5] is an adaptation of the Weka toolkit to a Grid environment. It is based on a client/server architecture. The server side defines a set of Grid Services that implements the functionalities of the different algorithms and phases of the DM process.
- FAEHIM [8] consists of a set of DM Web services for DDM, a set of tools to interact with these services, and a workflow system used to assemble the services and tools.
- Weka4WS [10] extends Weka to support remote execution of the DM algorithms. In such way, DDM tasks can be executed on decentralized Grid nodes by exploiting data distribution and improving application performance.

For a data classification task, most of these toolkits are only focused on executing the classification process over remote nodes as a post-processing task. It means, after creating the global data classifier. To create the global classifier, it is necessary to upload the complete data from all participating processing nodes. In this case, this process assumes that there are not data privacy and bandwidth restrictions. Our approach builds a Global Data Classifier through distributed datasets paying special attention in data privacy and bandwidth consumption, since our Central Computing Node does not require the complete original data but only metadata (more information in section 5). In this way, health care institutions are motivated for sharing information, expanding the global knowledge through building better data classifier that allows to detect dangerous illness in a patient at early stages.

3 Architecture for Building a Distributed Data Classifier

This architecture attempts to build a Global Classifier through a distributed datasets with similar structures. It is based on a set of java components executed on remotes machines, which interchange metadata with a Central Computing Node. This architecture includes the following components (see figure 1): a Central Computing Node (CCN) and a set of Local Nodes.

- The Central Computing Node is the main component in this architecture and contains a Data Classifier Builder and a Global Classifiers Repository
  - The Classifier Builder is the responsible for receiving all metadata from the distributed Java Components, grouping them and to build the Global Classifier
  - The Global Classifiers Repository contains the generated classifiers

- The Local Node represents a server that contains the following components:
  - Java Components that access a private dataset and send only metadata to the CCN
  - The local dataset represents all the files like data bases or text files where the data is stored

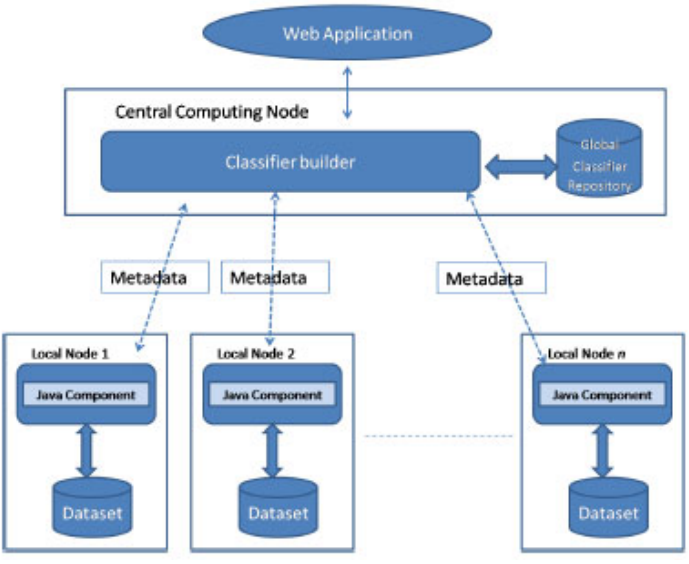


Fig. 1. Architecture

A prototype, named TeamMiner, is a Web application that implements this architecture. It allows users to be registered for collaborating in a Global Classifier building process. This process consists in defining working groups where the interested institutions work together to build a global data classifier. It is carried out by means of interchanging metadata between java nodes previously defined as a team member in our CCN. CCN gathers metadata (section 5 describes metadata) from java components without need of knowing the real data. This way to build a global classifier avoids original data intrusion and motivates organizations to participate in building a better global classifier.

3.1 Building the Global Classifier

The process of building a global data classifier begins when a new local node registered at the CCN wants to obtain a global classifier. A registered local node asks CCN to obtain a global classifier using the local node dataset and other datasets with similar structures located in all of registered local nodes. This process follows the next steps (see figure 2):

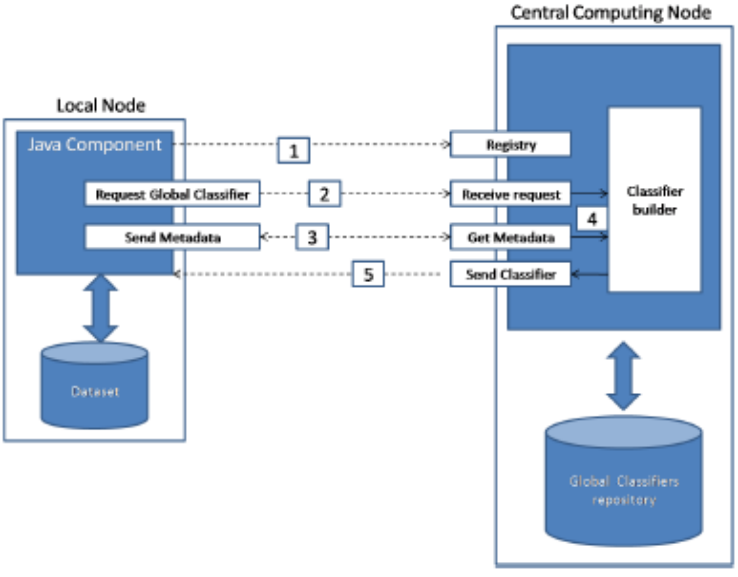


Fig. 2. Metadata interchange process

1. A java component is registered to join to the group that study or analyze datasets with similar structures. These datasets must have the same number and type of attributes.
2. The java component requests to the CCN a Global Classifier. The CCN offers two options: to get the last global classifier stored in the Global Classifiers repository (cache) or to get a new one.
3. If the java component chooses to build a new classifier then the CCN begins a number of iterations with all the registered java components asking for metadata (a summary of the data belonging to the participating node).
4. The metadata is sent to the CCN which interacts with all the local nodes asking for their metadata. This interactive process allows the CCN to build the Global Classifier.
5. Once the CCN has built a Global Classifier, it is saved and sent to the java component that request for it.

4 ID3 Global Data Classifier

The Web application TeamMiner that implements this architecture includes a classification algorithm to obtain decision trees. The original ID3 algorithm [6] has been adapted for our architecture. It is implemented using RMI and Web Services technologies. This algorithm works as follow:

- The Classifier Builder obtains metadata from each registered local node(see figure 2)
- It creates a global group of metadata
- Invokes buildTree method sending global metadata as parameter(see Algorithm 5.2)
  - The buildTree method receives the metadata
  - Calculates the gain for each attribute
  - Chooses the attribute with best gain
  - Appends the attribute to the tree
  - If maxim gain is equal to zero
    - Gets a distribution of values of attributes
    - Normalizes the distribution
    - Sets the leaf's value
  - Else
    - Asks metadata for each registered local node
    - It creates a global group of metadata
    - The method invokes itself sending the new global metadata
  - Return tree
- Finally, it returns a decision tree

5 Implementation

This section gives a brief description of our TeamMiner application, which is implemented using the architecture depicted in section 3. The first prototype of the TeamMiner application implements the ID3 algorithm described in section 4. The basic data structures used in this implementation are next described.

1. Remote interface

The java code below shows the remote interface of java components. It exposes two methods: getAttributes, it sends the attributes contained into the local sources, and getMetadata, it sends metadata (summary of data) wrapped into a Vector

```
public interface localInterface extends Remote{
        public Vector getAttributes() throws RemoteException;
        public Vector getMetadata() throws RemoteException;
        public Vector getMetadata(int attribute, String value,
        int level, boolean leaf) throws RemoteException;
}
```

This interface is implemented to interacting between the CCN and the local nodes

2. Structure of metadata

Metadata are a summary of the real data obtained from structures of nested vectors. The Metadata Vector contains one vector for each attribute of the original

source. Each attribute vector contains one vector for each value of attributes. Then, each value vector contains a vector for each value of the target class. These vectors contain the total number of related instances. An example of this structure is showed in figure 3.

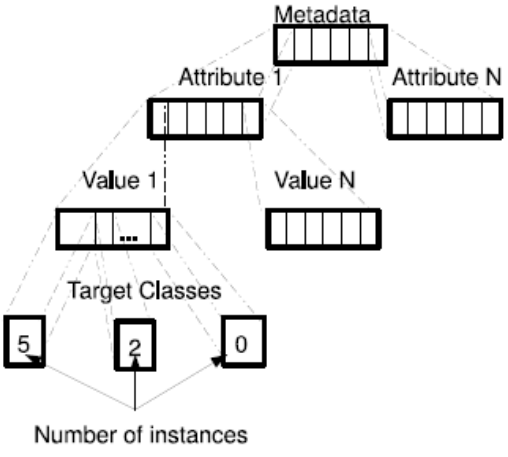


Fig. 3. Structure of Metadata

3. Central Node

CCN gathers metadata from all remote java components, grouping them into a global metadata group. The Classifier Builder starts to build the data classifier by means of getting the most significant attribute and requiring more metadata. Finally, when it generates the classifier, the CCN sends it to the requester java component. The mainly algorithms are shown in Algorithm 5.1 and Algorithm 5.2:

Algorithm 5.1: CLASSIFIER BUILDER ALGORITHM()

$$
\begin{array}{l}
Classifier\ Builder \\
\left\{
\begin{array}{l}
for\ each\ Java\ Agent\ do \\
\{getMetadata \\
end\ for \\
createGlobalMetadata \\
tree = buildTree(GlobalMetadata) \\
return\ tree
\end{array}
\right. \\
end\ Classifier\ Builder
\end{array}
$$

Algorithm 5.2: BUILDTREE ALGORITHM()

```
buildTree
{ getAttributesGain
  choose attribute with best gain
  tree = attribute chosen
  if maxGain = 0
    then { get distribution values of attributes
           get distribution normalized
           set leaf's value
    else { for each value in attribute do
           { for each Java Agent do
             {getMetadata
             createGlobalMetadata
             tree = buildTree(GlobalMetadata)
  end if
  return tree
end buildTree
```

6 Experimental Evaluations

We have tested our web application TeamMiner by means of emulating the process of registering three different hospitals as participating nodes. Each hospital owns a different dataset. The dataset are real data obtained from private hospitals. The three local nodes (one by hospital) used in this test are described as follows: 1 Linux Red Hat 4 server with dual core AMD Opteron 1.8Ghz, 4GB RAM and HD 250GB; 1 Linux Ubuntu server with Intel Pentium 4, 2.8Ghz, 1GB RAM and HD 120GB; 1 Windows 2003 server with dual core AMD Opteron 1.8Ghz, RAM 4GB and HD 250GB. The Central Computing Node (CCN) was running on a Solaris 10 server with dual core AMD Opteron 1.8Ghz, 4GB RAM and a HD of 250GB. All local nodes and the CCN were connected by a 100Mbps Ethernet LAN using a switch Foundry Networks Gigabit Ethernet 10/100/1000.

In the first three evaluations, the TeamMiner application builds a classifier using only the individual dataset taken from each hospital. We mainly evaluated the hit rate (percentage of instances that were correctly classified) of the classifier. In the second evaluation the TeamMiner builds a global classifier using information from the three registered local nodes. Results are shown in table 1.

First column in Table 1 shows the data classifier type. The size of the training set (number of instances) is shown in the second column. Following columns 3 to 5 show the percent of correctly, incorrectly and unclassified instances

Table 1. Preliminary Results

Classifier Type	Training Set	Correctly Classified (%)	Incorrectly Classified (%)	Unclassified (%)
Local	874	73.2	14.8	12
Local	1750	83.07	0	16.93
Local	4750	86.13	0.67	13.2
Global	7374	100	0	0

respectively. The four evaluations were tested using a test set of 750 instances. The table rows from one to three show results obtained by TeamMiner, using information from each hospital in an individual way. Last row shows results of TeamMiner after building the Global Classifier. We can see how the hit rate improves when using the global classifier. Due to the fact that only metadata were sent between participating nodes, the bandwidth consumption improves in 80% (this information was not included in Table 1). Using a summary of data (metadata) also motivate some institutions to share information extending the global knowledge. We can see that data classifiers perform better when global knowledge is included.

7 Conclusions

Data classifiers are tools that allow users to predict events based on historical information. Some institutions, such as hospitals, can obtain benefits from data classifiers, especially for illness prevention. However, many of them do not have enough historical information to build a good data classifier. Institutions with similar interest, like health care, would like to exchange information for building data classifiers. ID3 Classifiers can be built using only summaries taken from the real data. These summaries do not reveal the original data. This approach is attractive for institutions which are not able to share the original information (like digital health records). Global classifiers help in tasks like illness prevention and developing of better clinical guides. The data classifiers perform better when they are built from global knowledge. This situation was the motivation for developing the web application TeamMiner based on a distributed data mining architecture keeping privacy of data and taking advantage of global knowledge. As ongoing and future work, this prototype is going to extend other popular classification and clustering algorithms.

Acknowledgments

This research was partially funded by project number 51623 from “Fondo Mixto Conacyt-Gobierno del Estado de Tamaulipas”.

References

1. Artificial Intelligence Unit of University of Dortmund, Yale 4.0., http://rapid-i.com/ (last visit January 2009)
2. Khoussainov, R., Zuo, X., Kushmerick, N.: Grid-enabled Weka: A Toolkit for Machine Learning on the Grid. ERCIM 59, 47–48 (2004)
3. McQueen, J.: Some methods for classification and analysis of multivariations. In: Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967)
4. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: YALE: Rapid Prototyping for Complex Data Mining Tasks. In: 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2006)
5. Peña, J.M., Sánchez, A., Robles, V., Pérez, M.S., Herrero, P.: Adapting the Weka Data Mining Toolkit to a Grid Based Environment. In: Szczepaniak, P.S., Kacprzyk, J., Niewiadomski, A. (eds.) AWIC 2005. LNCS (LNAI), vol. 3528, pp. 492–497. Springer, Heidelberg (2005)
6. Quinlan, J.R.: Induction of Decision Trees. Machine Learning 1(1), 81–106 (1986)
7. Ross Quinlan, J.: C4.5: programs for machine learning. Morgan Kaufmann, San Francisco (1993)
8. Shaikh Ali, A., Rana, O.F., Taylor, I.J.: Web Services Composition for Distributed Data Mining. In: International Conference Workshop on Parallel Processing, pp. 11–18. IEEE, Los Alamitos (2005)
9. Statistics Department of the University of Auckland, R Project 2.6.1., http://www.r-project.org/ (last visit November 2008)
10. Talia, D., Trunfio, P., Verta, O.: Weka4WS: A WSRF-Enabled Weka Toolkit for Distributed Data Mining on Grids. In: Jorge, A.M., Torgo, L., Brazdil, P.B., Camacho, R., Gama, J. (eds.) PKDD 2005. LNCS (LNAI), vol. 3721, pp. 309–320. Springer, Heidelberg (2005)
11. University of Illinois and Data Mining Research Group and DAIS Research Laboratory, IlliMine 1.1.0., http://illimine.cs.uiuc.edu/ (last visit December 2008)
12. Williams, G.: Rattle 2.2.74, http://rattle.togaware.com (last visit May 2009)
13. Witten, H., Frank, E.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann Publishers, San Francisco (2005)

Modular Neural Network with Fuzzy Integration and Its Optimization Using Genetic Algorithms for Human Recognition Based on Iris, Ear and Voice Biometrics

Daniela Sánchez and Patricia Melin

Tijuana Institute of Technology, Tijuana México
danielasanchez.itt@hotmail.com, pmelin@tectijuana.mx

Abstract. In this paper we present the application of a Modular Neural Network (MNN) for iris, ear and voice recognition for a database of 77 persons. The proposed MNN architecture with which we are working consists of three modules; iris, ear and voice. Each module is divided in other three sub modules. Each sub module contains different information, which, the first 26 individuals are considered in module 1, the following 26 individuals in module 2 and the last 25 in module 3. We considered the integration of each biometric measure separately. Later, we proceed to integrate these modules with a fuzzy integrator. Also, we performed optimization of the modular neural networks and the fuzzy integrators using genetic algorithms, and comparisons were made between optimized results and the results without optimization.

1 Introduction

Biometrics plays an important role in public security and information security domains. Using various physiological characteristics of the human, such as face, facial thermo grams, fingerprint, iris, retina, hand geometry etc., biometrics accurately identifies each individual and distinguishes one from another [1].

The recognition of people is of great importance, since it allows us to have a greater control about when a person has access to certain information, area, or simply to identify if the person is the one who claims to be.

The achieved results indicate that biometric techniques are much more precise and accurate than the traditional techniques. Other than precision, there have always been certain problems which remain associated with the existing traditional techniques. As an example consider possession and knowledge. Both can be shared, stolen, forgotten, duplicated, misplaced or taken away. However the danger is minimized in case of biometric means [15].

2 Background

In this section we present some background of the biometric measurements that were used for this work.

P. Melin et al. (Eds.): Soft Comp. for Recogn. Based on Biometrics, SCI 312, pp. 85–102.
springerlink.com

2.1 Iris

The idea of using iris patterns for identification of persons was first proposed in 1936 by the ophthalmologist Frank Burch. However, it was not until 1987, when Leonard Flom and Aran Safir, American ophthalmologists, patented the concept of Burch.

Their interest in developing the system, brought them to connect with John G. Daugman, then professor of Harvard University so that he developed the algorithms necessary to perform the biometric recognition through the iris pattern [20]. These algorithms, patented by Daugman in 1994 and partly published in [9], are the basis of all iris recognition systems that exist today.

There are various work undertaken for iris recognition, as the work performed by Ahmad M. Sarhan [21], which uses neural networks and Discrete Cosine Transform for the identification based on iris.

2.2 Ear

The possibility of using the appearance of the ear as a means of personal identification was recognized and defended in about 1890 by the French criminologist Alphonse Bertillon [4].

The human recognition based on ear is of particular interest because is not affected by environmental factors such as mood, health, and clothing, as well as not affected by aging, making it more suitable for long-term identification compared with other measures, such as the face.

One of the best known works is that of Carreira Perpiñán in 1995 [5], where it uses artificial neural networks (ANN) for feature extraction [19]. Other work is of Ali, Javed and Basit, where they proposed a new ear recognition method using Wavelets [2].

2.3 Voice

Automatic speech recognition (ASR) has been one of the most successful technologies allowing the man-machine communications to request some information from it or to request to carry out some given task using the natural oral communication. The artificial intelligence field has contributed in a remarkable way to the development of ASR algorithms.

The automatic speech recognition is a very complex task due to the large amount of variations involved in it, such as intonation, voice level, health condition and fatigue, and so forth. Therefore, in the automatic speech recognition system, for specific or general tasks, there is an immense amount of aspects to be taken into account. This fact has contributed to increase the interest in this field, and as a consequence, several ASR algorithms have been proposed during the last 60 years [23]. Over the past two decades, voice recognition technology has developed to the point of becoming real-time, continuous speech systems that augment command, security, and content creation tasks with exceptionally high accuracy [18]. In a work more recently presented neural networks and type-2 fuzzy logic were used [14].

3 Basic Concepts

In this section we present some basic concepts needed to understand better what has been done in this research work.

3.1 Modular Neural Network

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyse. This expert can then be used to provide projections given new situations of interest and answer "what if" questions [22].

A neural network is said to be modular if the computation performed by the network can be decomposed into two or more modules (subsystems) that operate on distinct inputs without communicating with each other.

The modular neural networks are comprised of modules which can be categorized on the basis of both distinct structure and functionality which are integrated together via an integrating unit. With functional categorization, each module is a neural network which carries out a distinct identifiable subtask. Also, using this approach different types of learning algorithms can be combined in a seamless fashion [3].

3.2 Fuzzy Logic

Fuzzy logic is an area of soft computing that enables a computer system to reason with uncertainty [6]. A fuzzy inference system consists of a set of if-then rules defined over fuzzy sets. Fuzzy sets generalize the concept of a traditional set by allowing the membership degree to be any value between 0 and 1 [25]. This corresponds, in the real world, to many situations where it is difficult to decide in an unambiguous manner if something belongs or not to a specific class [13]. Fuzzy logic is a useful tool for modeling complex systems and deriving useful fuzzy relations or rules [17]. However, it is often difficult for human experts to define the fuzzy sets and fuzzy rules used by these systems [24].

The basic structure of a fuzzy inference system consists of three conceptual components: a rule base, which contains a selection of fuzzy rules, a database (or dictionary) which defines the membership functions used in the rules, and a reasoning mechanism that performs the inference procedure [11].

3.3 Genetic Algorithms

Genetic Algorithms (GAs) are nondeterministic methods that employ cross-over and mutation operators for deriving offsprings. GAs work by maintaining a constant-sized population of candidate solutions known as individuals (chromosomes). The power of a genetic algorithm lies in its ability to exploit, in

a highly efficient manner, information about a large number of individuals. The search underlying GAs is such that breadth and depth (exploration and exploitation) are balanced according to the observed performance (fitness) of the individuals evolved so far.

By allocating more reproductive occurrences to above average individual solutions, the overall effect is to increase the population's average fitness [10].

GAs have proven to be a useful method for optimizing the membership functions of the fuzzy sets used by these fuzzy systems [16].

4 Proposed Method and Results

This section describes the methodology used and results obtained in this research work.

4.1 Methodology

The methodology used in this work, consists of the following steps:

1. Obtain the databases.
2. Programming in MatLab the modular neural network for each of the biometric measurements.
3. Develop the fuzzy integrators.
4. Develop the genetic algorithms.

4.2 Databases and Pre-processing

The databases that we used and their pre-processing are described below.

4.2.1 Database of Iris

We used a database of human Iris from the Institute of Automation of the Chinese Academy of Sciences. It consists of 14 images (7 for each eye) per person, and it consists of 99 persons. The image dimensions are 320 x 280, JPEG format [8]. Only the first 77 persons were used. 8 images were used for training and 6 for testing (see Fig. 1).

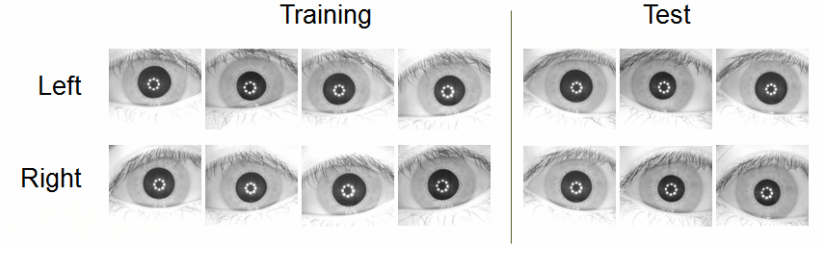


Fig. 1. Examples of the human iris images from CASIA database.

In the case of the Iris the following preprocessing steps were performed:

- The coordinates and radius of the iris and pupil using the method developed by Masek and Kovesi are obtained [12].
- A cut in the Iris is made.
- Resize of the new image to 21-21
- Convert images from vector to matrix.

4.2.2 Database of Ear

We used a database of Ear Recognition Laboratory from the University of Science & Technology Beijing (USTB). It consists of 4 images (of one ear) per person, and it is consists of 77 persons. The image dimensions are 300 x 400, BMP format [7]. 3 images were used for training and 1 for testing, and cross-validation was used (see Fig. 2).

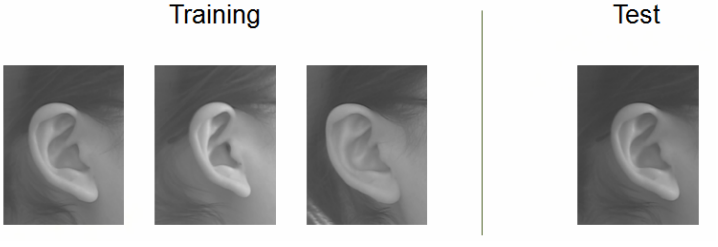


Fig. 2. Examples of Ear Recognition Laboratory from the University of Science & Technology Beijing (USTB).

In the case of the Ear for processing the following steps were performed:

- A cut of the ear is performed manually
- Resize of the new image to 132-91
- Convert images from vector to matrix

4.2.3 Database of Voice

In the case of voice, the database was made in Tijuana Institute of Technology, and it is consist of 7 voice samples (of 77 persons), WAV format. 7 voice samples were used for training and 3 for testing. The word that they said in Spanish was "ACCESAR". To preprocess the voice the Mel Frequency Cepstral Coefficients were used.

4.3 Architecture and Results of the Modular Neural Network

The general architecture of the modular neural network is shown in figure 3, we can see that consists of 3 modules, one for each biometric measure, each module is divided in other three sub modules. Each sub module contains different information, which is, the first 26 individuals are in module 1, the following 26 individuals in module 2 and the last 25 in module 3. We considered the integration of each biometric measure separately.

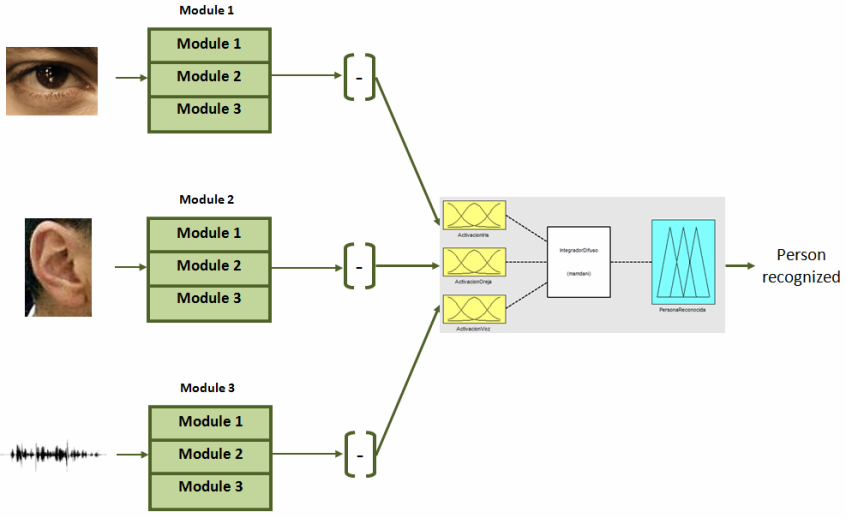


Fig. 3. The general architecture of the modular neural network

To make simulations, we did perform experiments with 3 types of learning algorithms: gradient descent with scaled conjugate gradient (SCG), gradient descent with adaptive learning and momentum (GDX) and adaptive learning (GDA), also we varied the number of neurons in the first and second hidden layer.

4.3.1 Results of the Iris

As noted, 8 images were used for training and 6 for testing (per person), in total 462 images were used for testing. To integrate is used Gating Network.

We can see in table 1, the best trainings that were obtained. Where training EI2 and training EI4 were having the best recognition rate (97.19%, 448 images from 462), talking of training EI2, in the module 1, the result was with 150 neurons in the first hidden layer and 110 in the second hidden layer, the learning algorithm was adaptive learning (GDA), with an identification rate of 96.15% (150/156), in the module 2, the result was with 150 neurons in the first hidden layer and 118 in the second hidden layer, the learning algorithm was gradient descent with scaled conjugate gradient (SCG), with an identification rate of 96.79% (151/156), and in the module 3, the result was with 150 neurons in the first hidden layer and 110 in the second hidden layer, the learning algorithm was adaptive learning (GDA), with an identification rate of 98.67% (148/150).

4.3.2 Results of the Ear

As noted, 3 images were used for training and 1 for testing (per person), in total 77 images were used for testing. Cross-validation was used. To integrate is used the winner take it all.

We can see in table 2, the best trainings that we obtained. The best average recognition was 89.29%, in the validation 1 the best result was in the training EO3

Table 1. The best results of iris

Training	Method	Neurons	Epoch	Error	Training Duration	Module Recognition	TotalRecognition
EI1	1 traingda	150, 110	2000	0.00001	00:01:59	150/156 (96.15)	448/462 96.97%
	2 traingda	150, 100			00:00:55	150/156 (96.15)	
	3 traingda	150, 110			00:00:39	148/150 (98.67)	
EI2	1 traingda	150,110	2000	0.00001	00:01:59	150/156 (96.15)	448/462 97.19%
	2 trainscg	150,118			00:00:59	151/156 (96.79)	
	3 traingda	150,110			00:00:39	148/150 (98.67)	
EI3	1 traingda	140,95	2000	0.00001	00:01:07	148/156 (95.51)	445/462 96.32%
	2 traingda	150,125			00:00:59	150/156 (96.15)	
	3 traingda	155,110			00:00:38	147/150 (98.00)	
EI4	1 traingda	150,100	2000	0.00001	00:02:22	150/156 (96.15)	448/462 97.19%
	2 traingda	150,115			00:00:59	151/156 (96.79)	
	3 traingda	155,110			00:00:39	148/150 (98.67)	
EI5	1 traingda	150,100	2000	0.00001	00:01:12	150/156 (96.15)	448/462 96.97%
	2 traingda	150,115			00:00:59	150/156 (96.15)	
	3 traingda	155,110			00:00:44	148/150 (98.67)	

Table 2. The best results of ear

# Tra.	Validation 1		Validation 2		Validation 3		Validation 4		Average recognition
EO1	(25/26) 96.15%	75/77	(18/26) 69.23%	57/77	(22/26) 84.61%	66/77	(26/26) 100%	26/77	89.28%
	(25/26) 96.15%		(20/26) 76.92%		(23/26) 88.46%		(26/26) 100%		
	(25/25) 100%	97.40%	(19/25) 76.92%	74.02%	(21/25) 84.00%	85.71%	(25/25) 100%	100%	
EO2	(25/26) 96.15%	74/77	(15/26) 57.69%	51/77	(20/26) 76.92%	64/77	(26/26) 100%	77/77	86.36%
	(24/26) 92.30%		(20/26) 76.92%		(21/26) 80.76%		(26/26) 100%		
	(25/25) 100%	96.10%	(16/25) 64.00%	66.23%	(23/25) 92.00%	83.11%	(25/25) 100%	100%	
EO3	(26/26) 100%	77/77	(18/26) 69.23%	57/77	(22/26) 84.61%	64/77	(26/26) 100%	77/77	89.28%
	(26/26) 100%		(19/26) 73.07%		(22/26) 84.61%		(26/26) 100%		
	(25/25) 100%	100%	(20/25) 80.00%	74.02%	(20/25) 80.00%	83.11%	(25/25) 100%	100%	

with 100% (77/77) of recognition, in the validation 2 the best result was in the training EO1 and the training EO3 with 74.02% (57/77) of recognition, in the validation 3 the best result was in the training EO1 with 85.71% (66/77) of recognition, and in the validation 4 the result in all the trainings was 100% (77/77) of recognition.

4.3.3 Results of the Voice

As noted, 4 voice samples were used for training and 3 for testing (per person), in total 321 voice samples were used for testing. To integrate results a gating was used.

We can see in table 3, the best trainings that we obtained. Where training EV4 had the best recognition rate (94.81%, 219 voice samples from 231), in the module 1, the result was with 190 neurons in the first hidden layer and 110 in the second hidden layer, the learning algorithm was adaptive learning (GDA), with an identification rate of 96.15% (75/78), in the module 2, the result was with 190 neurons in the first hidden layer and 120 in the second hidden layer, the learning algorithm was adaptive learning (GDA), with an identification rate of 94.87% (74/78), and

Table 3. The best results of voice

Training	Method	Neurons	Epoch	Error	Training Duration	Module Recognition	TotalRecognition
EV1	1 traingda	190,110	2000	0.000001	00:00:39	74/78 (94.87%)	214/231 92.64%
	2 traingda	190,110			00:00:42	72/78 (92.31%)	
	3 trainscg	190,110			00:00:30	68/75 (90.67%)	
EV2	1 traingda	190,110	2000	0.00001	00:00:31	75/78 (96.15%)	216/231 93.51%
	2 traingda	190,110			00:00:34	73/78 (93.59%)	
	3 trainscg	190,110			00:00:31	68/75(90.67%)	
EV3	1 traingda	190,110	2000	0.00001	00:00:30	75/78 (96.15%)	217/231 93.94%
	2 traingda	190,120			00:00:29	74/78 (94.87%)	
	3 trainscg	190,110			00:00:41	68/75 (90.67%)	
EV4	1 traingda	190,110	2000	0.00001	00:00:33	75/78 (96.15%)	219/231 94.81%
	2 traingda	190,120			00:00:29	74/78 (94.87%)	
	3 trainscg	190,110			00:00:25	70/75 (93.33%)	
EV7	1 traingda	190,115	2000	0.00001	00:00:31	73/78 (93.59%	215/231 93.07%
	2 traingda	190,120			00:00:31	73/78 (93.59%	
	3 traingda	190,120			00:00:31	69/75 (92.00%)	

in the module 3, the result was with 190 neurons in the first hidden layer and 110 in the second hidden layer, the learning algorithm was gradient descent with scaled conjugate gradient (SCG), with an identification rate of 93.33% (70/75).

4.4 Genetic Algorithm for MNN Optimization and Results

With the purpose of increasing the percentage of recognition, it was decided to use a genetic algorithm that allowed us to optimize some parameters of the modular neural networks. In this case we decided to optimize the number of neurons in two hidden layers, type of learning algorithm and the error goal.

Figure 4 shows the binary chromosome of 34 genes, which was established for optimization of neural networks. 7 genes were established for the first hidden layer and 6 genes for the second hidden layer, to set the final value of neurons, were added 60 and 50 neurons respectively, this is done to prevent that the neural network was training with a number of neurons too small, with respect to the learning algorithm, 2 genes were established, for this value was added an 1 , this for was to have 3 choices of learning algorithms (trainscg, traingdx and traingda), and the error goal was established with 19 genes, with these genes was realized an operation to set the error goal. The fitness function was (see equation 1).

$$\text{Fitness} = \frac{\text{Total errors}}{\text{Total test data.}} \tag{1}$$

4.4.1 Optimized Iris Results

We can see in table 4, the best evolution that we had for each module. In module 1, the best evolution was with 165 neurons in the first hidden layer and 78 in the second hidden layer, the learning algorithm was adaptive learning (GDA), goal error of 0.000002, and with an identification rate of 97.44% (152/156). In module 2, the best evolution was with 68 neurons in the first hidden layer and 69 in the second hidden layer, the learning algorithm was gradient descent with scaled

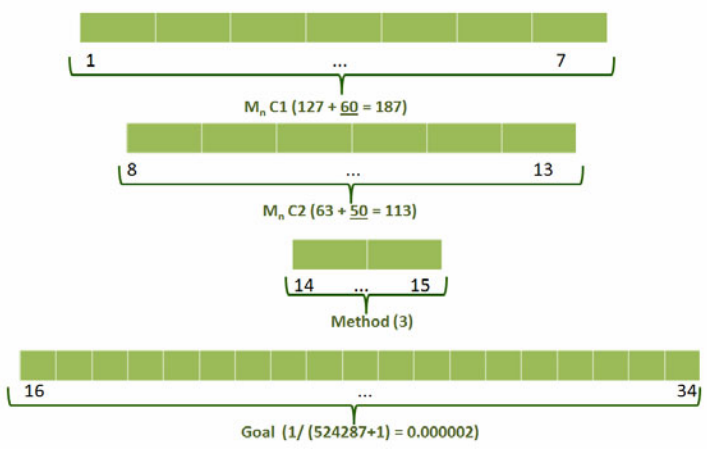


Fig. 4. The binary chromosome for the MNN

conjugate gradient (SCG), goal error of 0.000002, and with an identification rate of 98.08% (153/156). In module 3, the best evolution was with 131 neurons in the first hidden layer and 77 in the second hidden layer, the learning algorithm was adaptive learning (GDA), goal error of 0.000002, and with an identification rate of 100% (150/150). We had a percent of recognition final of 98.48%, this is greater than the percent of recognition non-optimized (97.19%)

Table 4. The best evolutions for each module for iris

Mod	GGAP	*cross-over*	Pc	Pm	Duration	Method	Error Goal	Neurons	Fitness	Module Recognition	Total Recog.
1	0.8	Xovmp	0.8	0.05	05:27:21	traingda	0.000002	165,78	0.025641	(152 /156) 97.44%	(455/462) 98.48%
2	0.8	Xovmp	0.8	0.06	01:05:56	trainscg	0.000002	68,69	0.019231	(153 /156) 98.08%	
3	0.8	xovmp	0.8	0.05	02:50:45	traingda	0.000002	131,77	0.00000	(150/150) 100%	

4.4.2 Optimized Ear Results

We can see in table 5, the best evolution that we had for each module for the validation 2. In module 1, the best evolution was with 126 neurons in the first hidden layer and 86 in the second hidden layer, the learning algorithm was adaptive learning (GDA), goal error of 0.000004, and with an identification rate of 80.76% (21/26). In module 2, the best evolution was with 58 neurons in the first hidden layer and 49 in the second hidden layer, the learning algorithm was gradient descent with gradient descent with adaptive learning and momentum (GDX), goal error of 0.000002, and with an identification rate of 84.61% (22/26). In module 3, the best evolution was with 117 neurons in the first hidden layer and 69 in the second hidden layer, the learning algorithm was gradient descent with scaled conjugate gradient (SCG), goal error of 0.000004, and with an identification rate of

Table 5. The best evolutions for each module for ear (validation 2)

Mod	GGAP	cross-over	Pc	Pm	Duration	Method	Error Goal	Neurons	Fitness	Module Recognition	Total Recog.
1	0.8	Xovmp	0.8	0.6	09:30:12	traingda	0.000004	126,86	0.192308	(21/26) 80.76%	64/77
2	0.8	Xovmp	0.8	0.5	12:03:54	traingdx	0.000002	58,49	0.153846	(22/26) 84.61%	83.11%
3	0.8	xovmp	0.8	0.6	11:02:23	trainscg	0.000004	117,69	0.160000	(22/25) 84%	

84% (22/25). We had a percent of recognition final of 83.11%, this is greater than the percent of recognition non-optimized (74.02%)

We can see in table 6, the best evolution that we had for each module for the validation 3. In module 1, the best evolution was with 140 neurons in the first hidden layer and 79 in the second hidden layer, the learning algorithm was adaptive learning (GDA), goal error of 0.000002, and with an identification rate of 92.30% (24/26). In module 2, the best evolution was with 92 neurons in the first hidden layer and 49 in the second hidden layer, the learning algorithm was gradient descent with gradient descent with gradient descent with scaled conjugate gradient (SCG), goal error of 0.000004, and with an identification rate of 92.30% (24/26). In module 3, the best evolution was with 63 neurons in the first hidden layer and 53 in the second hidden layer, the learning algorithm was gradient descent with scaled conjugate gradient (SCG), goal error of 0.000002, and with an identification rate of 92% (23/25). We had a percent of recognition final of 92.20%, this is greater than the percent of recognition non-optimized (85.71%).

Table 6. The best evolutions for each module for ear (validation 3)

No	GGAP	cross-over	Pc	Pm	Duration	Alg.	Error Goal	Neurons	Fitness	Module Recognition	Total Recog.
1	0.8	Xovmp	0.8	0.6	11:34:11	traingda	0.000002	140,79	0.076923	(24/26) 92.30%	(71/77)
2	0.8	Xovmp	0.8	0.5	15:03:34	trainscg	0.000004	92,49	0.076923	(24/26) 92.30%	92.20%
1	0.8	xovmp	0.8	0.6	14:33:54	trainscg	0.000002	63,53	0.080000	(23/25) 92%	

4.4.3 Optimized Voice Results

We can see in table 7, the best evolution that we had for each module. In the module 1, the best evolution was with 177 neurons in the first hidden layer and 86 in the second hidden layer, the learning algorithm was gradient descent with scaled conjugate gradient (SCG), goal error of 0.000010, and with an identification rate of 98.72% (77/78). In module 2, the best evolution was with 91 neurons in the first hidden layer and 113 in the second hidden layer, the learning algorithm was gradient descent with gradient descent with gradient descent with adaptive learning and momentum (GDX), goal error of 0.000003, and with an identification rate of 96.15% (75/78). In module 3, the best evolution was with 150 neurons in the

Table 7. The best evolutions for each module for voice

Mod	GGAP	cross-over	Pc	Pm	Duration	Method	Error Goal	Neurons	Fitness	Module Recognition	Total Recog.
1	0.8	Xovmp	0.8	0.05	02:04:44	trainscg	0.000010	177,86	0.012821	(77/78) 98.72 %	(223/231) 96.54%
2	0.8	Xovmp	0.8	0.05	00:58:15	traingdx	0.000003	91,113	0.038462	(75/78) 96.15 %	
3	0.8	xovmp	0.8	0.05	01:16:40	traingda	0.000012	150,87	0.053333	(71/75) 94.67 %	

first hidden layer and 87 in the second hidden layer, the learning algorithm was adaptive learning (GDA), goal error of 0.000012, and with an identification rate of 94.67% (71/75).

4.5 Fuzzy Integration

7 cases were established for combining different trainings of iris, ear and voice, optimized and non optimized. To combine the responses of the different biometric measurements, we established 2 fuzzy integrators; the first (see Fig. 5) ; of Mamdani type with 27 rules and trapezoidal membership functions, the second (see Fig. 6); of Mamdani type with 23 rules and gaussian membership functions.

The fuzzy systems have 3 input variables (one for each biometric measure) and 1 output variable, there are 3 membership functions for each variable (ABaja, Amedia and AAlta)

4.5.1 Fuzzy Integration Results Using the Fuzzy Integrator # 1

We can see in table 8 fuzzy integration results using the fuzzy integrator # 1.

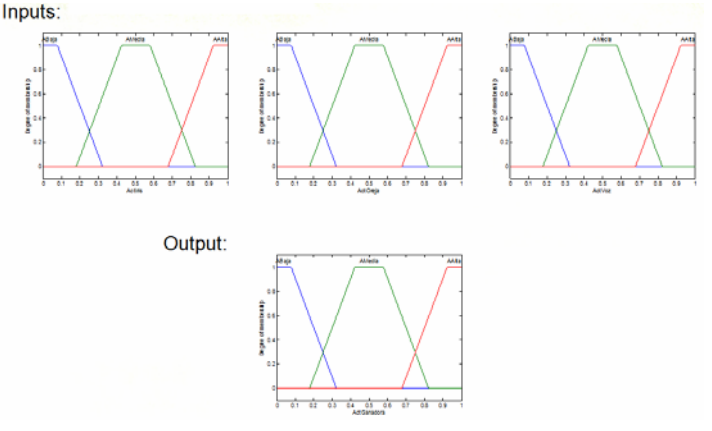


Fig. 5. Fuzzy Integrator # 1

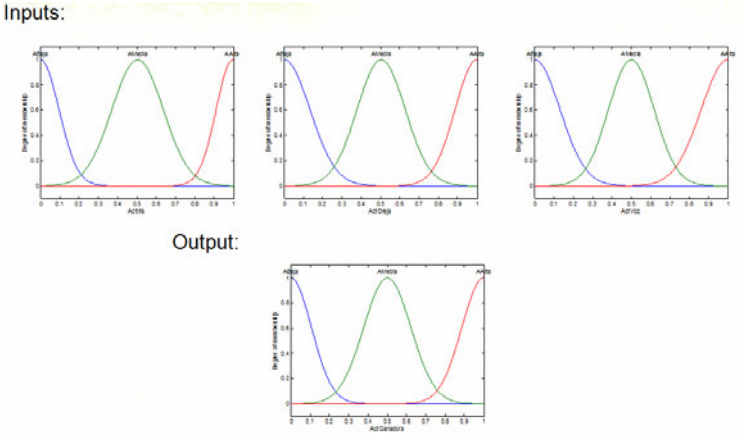


Fig. 6. Fuzzy Integrator # 2

Table 8. Fuzzy integration results using the fuzzy integrator # 1

# Case	Iris		Ear		Voice		% Recognition
1	EI3	96.32%	EO2V2	66.23%	EV1	92.64%	394/462 85.28%
2	EI3	96.32%	EO1V1	97.40%	EV3	93.94%	427/462 92.42%
3	EI5	96.97%	EO1V3	85.71%	EV5	93.07%	415/462 89.83%
4	EI2	97.19%	EO3V1	100.00%	EV4	94.81%	435/462 94.16%
5	Iris optimized	98.48%	EO3V1	100.00%	Voice optimized	96.54%	448/462 96.97 %
6	EI4	97.19%	EO3V2	74.02 %	EV4	94.81%	399/462 86.36 %
7	Iris optimized	98.48%	Ear optimized V2	83.11%	Voice optimized	96.54%	432/462 93.51%

4.5.2 Fuzzy Integration Results Using the Fuzzy Integrator # 2

We can see in table 9 fuzzy integration results using the fuzzy integrator # 2.

4.5.3 Comparison between Both Fuzzy Integrators

We can see in figure 7 the comparison between both fuzzy integrators and observed a variation in the percentage of recognition, in most cases the recognition rate was increased using the fuzzy integrator # 2.

Table 9. Fuzzy integration results using the fuzzy integrator # 2

# Case	Iris		Ear		Voice		% Recognition
1	EI3	96.32%	EO2V2	66.23%	EV1	92.64%	380/462 82.25 %
2	EI3	96.32%	EO1V1	97.40%	EV3	93.94%	454/462 98.27 %
3	EI5	96.97%	EO1V3	85.71%	EV5	93.07%	437/462 94.59 %
4	EI2	97.19%	EO3V1	100.00%	EV4	94.81%	459/462 99.35 %
5	Iris optimized	98.48%	EO3V1	100.00%	Voice optimized	96.54%	460/462 99.57 %
6	EI4	97.19%	EO3V2	74.02 %	EV4	94.81%	403/462 87.23 %
7	Iris optimized	98.48%	Ear optimized V2	83.11%	Voice optimized	96.54%	432/462 93.51%

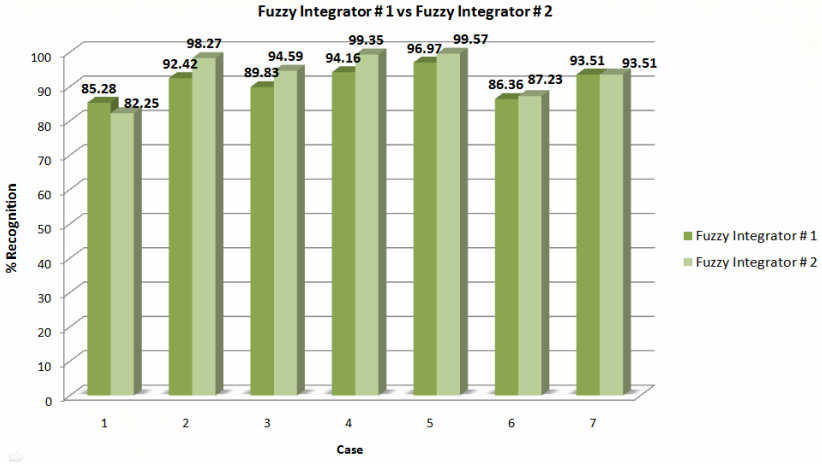


Fig. 7. Comparison between both fuzzy integrators

4.6 Genetic Algorithm for Fuzzy Integrator and Results

With the purpose to increase even more the percentage of recognition, was decided perform a genetic algorithm that allowed us to optimize the fuzzy integrator in the following aspects;

- Its type of system (mamdani or sugeno)
- Type of membership functions (trapezoidal and GBell)
- Parameters of the membership functions.

Were worked with type-1 fuzzy logic. The fitness function was:

$$\text{Fitness} = \frac{\text{Total errors}}{\text{Total test data.}} \tag{2}$$

The chromosome contains 100 genes, the first 13 are binary, genes 14 to 100 are real, its value is between 0 and 1. Is described below:

- Gene 1: type of systems (Mamdani or Sugeno)
- Gene 2: Type of membership function for Iris (ABaja)
- Gene 3: Type of membership function for Iris (AMedia)
- Gene 4: Type of membership function for Iris (AAlta)
- Gene 5: Type of membership function for Ear (ABaja)
- Gene 6: Type of membership function for Ear (AMedia)
- Gene 7: Type of membership function for Ear (AAlta)
- Gene 8: Type of membership function for Voice (ABaja)
- Gene 9: Type of membership function for Voice (AMedia)
- Gene 10: Type of membership function for Voice (AAlta)
- Gene 11: Type of membership function for Output (ABaja) (if type of system is Mamdani)
- Gene 12: Type of membership function for Output (AMedia) (if type of system is Mamdani)
- Gene 13: Type of membership function for Output (AAlta) (if type of system is Mamdani)
- Gene 14: Output Value (ABaja) (if type system is Sugeno)
- Gene 15: Output Value (AMedia) (if type system is Sugeno)
- Gene 16: Output Value (AAlta) (if type system is Sugeno)
- Gene 17 to 100: parameters of membership functions

4.6.1 Optimized Fuzzy Integrator Results

We can see in table 10 the optimized Fuzzy Integrator Results

The genetic algorithm created different fuzzy integrators, for example in Figure 8 we can see the best fuzzy integrator for case 3, in this case the type of system was Sugeno. For case 4 we can see the best fuzzy integrator created in figure 9.

4.6.2 Comparison between Fuzzy Integrators

We can see in table 11 the comparison between the different fuzzy integrators and observe that the best percentages of recognition are for the optimized fuzzy integrators in all cases.

Table 10. Optimized fuzzy integrator results

# Case	Iris		Ear		Voice		% Recognition
1	EI3	96.32%	EO2V2	66.23%	EV1	92.64%	442/462 95.67%
2	EI3	96.32%	EO1V1	97.40%	EV3	93.94%	459/462 99.35%
3	EI5	96.97%	EO1V3	85.71%	EV5	93.07%	451/462 98.05%
4	EI2	97.19%	EO3V1	100.00%	EV4	94.81%	460/462 99.57%
5	Iris optimized	98.48%	EO3V1	100.00%	Voice optimized	96.54%	461/462 99.78%
6	EI4	97.19%	EO3V2	74.02 %	EV4	94.81%	454/462 98.26%
7	Iris optimized	98.48%	Ear optimized V2	83.11%	Voice optimized	96.54%	448/462 96.96%

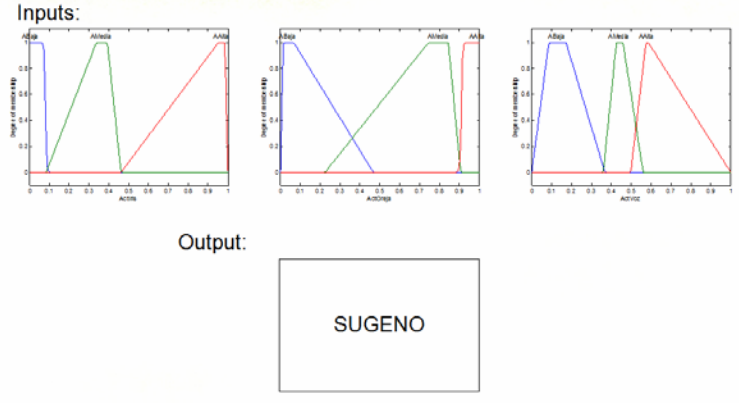


Fig. 8. The best fuzzy integrator for the case 3

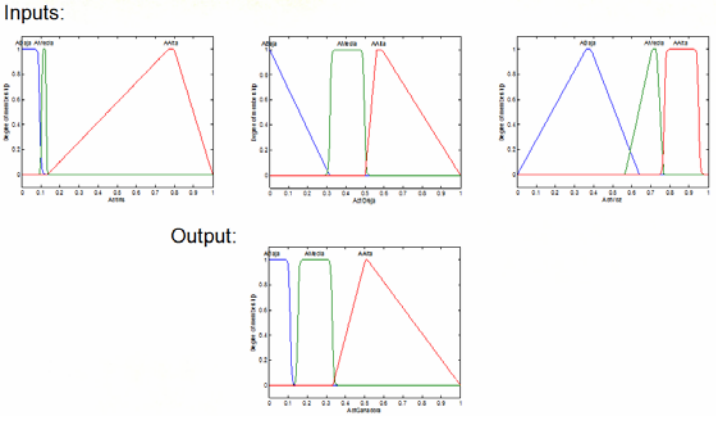


Fig. 9. The best fuzzy integrator for the case 4

Table 11. The comparison between fuzzy integrators

# Case	Fuzzy Integrator # 1	Fuzzy Integrator # 2	Optimized Fuzzy Integrator
1	394/462 85.28%	380/462 82.25 %	442/462 95.67%
2	427/462 92.42%	454/462 98.27 %	459/462 99.35%
3	415/462 89.83%	437/462 94.59 %	451/462 98.05%
4	435/462 94.16%	459/462 99.35 %	460/462 99.57%
5	448/462 96.97 %	460/462 99.57 %	461/462 99.78%
6	399/462 86.36 %	403/462 87.23 %	454/462 98.26%
7	432/462 93.51%	432/462 93.51%	448/462 96.96%

5 Conclusions

We performed the human recognition based on iris, ear and voice biometrics, based on modular neural networks, and for combining their responses we used type-1 fuzzy logic. Also, we used genetic algorithms for optimizing the modular

neural network and the fuzzy integrator (in some of their parameters), and thus the percentage of recognition was increased.

With the results obtained, we can see that the genetic algorithms are of great help to find the optimal architectures.

We will continue working on these cases of recognition, the rules will be optimized and we will work with type-2 fuzzy logic, that will surely help us to increase the percentage of recognition.

References

[1] Abiyev, R., Altunkaya, K.: Personal Iris Recognition Using Neural Network., Near East University, Department of Computer Engineering, Lefkosa, North Cy-prus, Abril (2008)

[2] Ali, M., Javed, M., Basit, A.: Ear Recognition Using Wavelets., Department of Computer Engineering, College of Electrical and Mechanical Engineering, National University of Sciences and Technology, Peshawar Road, Rawalpindi, 46000, Pakistan (2007)

[3] Azamm, F.: Biologically Inspired Modular Neural Networks. PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia. Mayo (2000)

[4] Bertillon, A.: La Photographie Judiciaire avec un Appendice sur la Classification et L'identification Anthropométriques. Gauthier-Villars, Paris (1890)

[5] Carreira, M.: Aplicación de las redes neuronales de compresión a la extracción de características para el reconocimiento a partir de imágenes de la oreja, Universidad Politécnica de Madrid, España (Septiembre 1995)

[6] Castillo, O., Melin, P.: Soft Computing for Control of Non-Linear Dynamical Systems. Springer, Heidelberg (2001)

[7] Database Ear Recognition Laboratory from the University of Science & Technology Beijing (USTB). Found on the Web page, http://www.ustb.edu.cn/resb/en/index.htm

[8] Database of Human Iris. Institute of Automation of Chinese Academy of Sciences (CASIA). Found on the Web page, http://www.cbsr.ia.ac.cn/english/IrisDatabase.asp

[9] Daugman, J.G.: High Confidence Visual Recognition of Persons by a Test of Statistical Independence. IEEE Trans. on Pattern Analysis and Machine Intelligence 15(11), 1148–1161 (1993)

[10] Huang, J., Wechsler, H.: Eye Location Using Genetic Algorithm, Department of Computer Science, George Mason University, Washington, DC (1999)

[11] Jang, J., Sun, C., Mizutani, E.: Neuro-Fuzzy and Soft Computing. Prentice Hall, New Jersey (1997)

[12] Masek, L., Kovesi, P.: MATLAB Source Code for a Biometric Identification System Based on Iris Patterns. The School of Computer Science and Software Engineering, The University of Western Australia (2003)

[13] Melin, P., Castillo, O.: Hybrid Intelligent Systems for Pattern Recognition Using Soft Computing, pp. 2–3. Springer, Heidelberg (2005)

[14] Melin, P., Urias, J., Solano, D., Soto, M., Lopez, M., Castillo, O.: Voice Recognition with Neural Networks, Type-2 Fuzzy Logic and Genetic Algorithms., Tijuana Institute of Technology, Tijuana México, Agosto (2006)

[15] Moreno, B., Sanchez, A., Velez, J.F.: On the Use of Outer Ear Images for Personal Identification in Security Applications. In: IEEE 33rd Annual International Carnahan Conference on Security Technology, pp. 469–476 (1999)

[16] Nawa, N., Takeshi, F., Hashiyama, T., Uchikawa, Y.: A study on the discovery of relevant fuzzy rules using pseudo-bacterial genetic algorithm, http://www.bioele.nuee.nagoya-u.ac.jp/~eiji/papers/transIE99 (Accessed July 29, 1999)

[17] Okamura, K.M., Yager, R., Nakanishi, S.: Character diagnosis of fuzzy system by genetic algorithm and fuzzy inference, http://www.ep.utokai.ac.jp/~masakazu/vietnam/vpaper.htm (Accessed July 30, 1999)

[18] ProQuest Information and Learning Company., History of Voice Recognition Technology, The Information Management Journal (October 07, 2009), http://findarticles.com/p/articles/mi_qa3937/is_200401/ai_n9383074/

[19] Saleh, M.: Using Ears as a Biometric for Human Recognition, Arab Academy for Science and Technology and Maritime Transport, Cairo, Egypt (Septiembre 2006)

[20] Sánchez, R.: El Iris Ocular como parámetro para la Identificación Biométrica., Universidad Politécnica de Madrid, España (Septiembre 2000)

[21] Sarhan, A.: Iris Recognition Using Discrete Cosine Transform and Artificial Neural Networks. Department of Computer Engineering, University of Jordan, Amman-11195, Jordan (2009)

[22] Stergiou, C., Siganos, D.: Neural Networks, http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html#Why%20use%20neural%20networks

[23] Verma, B., Blumenstein, M.: Pattern Recognition Technologies and Applications. In: Information Science Reference, Hershey, New York, pp. 90–91 (2008)

[24] Wang, W., Bridges, S.: Genetic Algorithm Optimization of Membership Functions for Mining Fuzzy Association Rules. Department of Computer Science Mississippi State University (March 2, 2000)

[25] Zadeh, L.A.: Fuzzy Sets. Journal of Information and Control 8, 338–353 (1965)

Comparative Study of Type-2 Fuzzy Inference System Optimization Based on the Uncertainty of Membership Functions

Denisse Hidalgo[1], Patricia Melin[2], Oscar Castillo[2], and Guillermo Licea[1]

[1] UABC University, Tijuana, México
[2] Tijuana Institute of Technology, Tijuana México
paulette1019@hotmail.com, pmelin@tectijuana.mx,
ocastillo@tectijuana.mx, glicea@uabc.mx

Abstract. A comparative study of type-2 fuzzy inference systems optimization as an integration method of Modular Neural Networks (MNNs) is presented. The optimization method for type-2 fuzzy systems is based on the footprint of uncertainty (FOU) of the membership functions. We use different benchmark problems to test the optimization method for the fuzzy systems. First, we tested the methodology by manually incrementing the percentage in the FOU, later we apply a Genetic Algorithm to find the optimal type-2 fuzzy system. We show the comparative results obtained for the benchmark problems.

1 Introduction

Biometrics are used for measuring and analyzing a person's unique characteristics. There are two types of biometrics: behavioral and physical. Behavioral biometrics are generally used for verification while physical biometrics can be used for either identification or verification. Examples of biometric measures are: fingerprint, people faces, iris patterns, voice recognition, and shape of the hand signature.

Multimodal biometric technology uses more than one biometric identifier to compare the identity of the person. Multimodal biometry is based on using different biometrics characteristics to improve the recognition rate and reliability of the final result of recognition.

For this reason, in this paper, three biometric characteristics of a person are used to achieve a good recognition rate of humans; face, fingerprint and voice [2]. In this paper we describe an optimization method for the membership functions of type-2 fuzzy systems based on the level of uncertainty, in which, the first step is to obtain the optimal type-1 Fuzzy Inference System, which allows us to find the uncertainty of their membership functions using genetic algorithms.

This paper is organized as follows: Section 2 shows an introduction to soft computing techniques, section 3 describes the development of the evolutionary

P. Melin et al. (Eds.): Soft Comp. for Recogn. Based on Biometrics, SCI 312, pp. 103–120.
springerlink.com

method, section 4 shows fuzzy systems optimization based on the level of uncertainty; section 5 shows simulation results, and section 6 the conclusions.

2 Preliminaries

The concept of soft computing, introduced by Lotfi Zadeh in 1991, serves to highlight the emergence of computing methodologies in which the accent is on exploiting the tolerance for imprecision and uncertainty to achieve tractability, robustness and low solution cost. At this juncture, the principal constituents of soft computing are fuzzy logic, neurocomputing, evolutionary computing and probabilistic computing, with the later subsuming belief networks, chaotic systems and parts of learning theory. What is particularly important about soft computing is that it facilitates the use of fuzzy logic, neurocomputing, evolutionary computing and probabilistic computing in combination, leading to the concept of hybrid intelligent systems. Such systems are rapidly growing in importance and visibility [27]. We used three different techniques of soft computing for the optimization method; modular neural networks, type-2 Fuzzy Logic and genetic algorithms.

2.1 Modular Neural Networks

A computational system can be considered to have a modular architecture if it can be split into two or more subsystems in which each individual subsystem evaluates either distinct inputs or the same inputs without communicating with other subsystems. The overall output of the modular system depends on an integration unit, which accepts outputs of the individual subsystems as its inputs and combines them in a predefined fashion to produce the overall output of the system. The modular system design approach has some obvious advantages, like simplicity and economy of design, computational efficiency, fault tolerance and better extendibility. A neural network is said to be modular if the computation performed by the network can be decomposed into two or more modules (subsystems) that operate on distinct inputs without communicating with each other. The outputs of the modules are mediated by an integrating unit that is not permitted to feed information back to the modules. In particular, the integrating unit decides:

1. How the modules are combined to form the final output of the system, and
2. Which modules should learn which training patterns [8][4].

2.2 Type-2 Fuzzy Logic

Type-2 Fuzzy Logic can handle uncertainty because it can model and reduce it to the minimum their effects. Also, if all the uncertainties disappear, type-2 fuzzy logic reduces to type-1 fuzzy logic, in the same way that, if the randomness disappears, the probability is reduced to the determinism. Fuzzy sets and fuzzy logic are the foundation of fuzzy systems, and have been developed looking to model the form as the brain manipulates inexact information. Type-2 fuzzy sets are used to model uncertainty and imprecision; originally they were proposed by Zadeh in

1975 and they are essentially "fuzzy-fuzzy" sets in which the membership degrees are type-1 fuzzy sets [3].

2.3 Genetic Algorithms

For being able to use a genetic algorithm (GA), one should represent a solution to the problem as a *genome* (or *chromosome*). The genetic algorithm then creates a population of solutions and applies genetic operators such as mutation and crossover to evolve the solutions in order to find the best one. It uses various selection criteria so that it picks the best individuals for mating (and subsequent crossover). The objective function determines the best individual where each individual must represent a complete solution to the problem that you are trying to optimize [10].

3 Optimization Method Description

Based on the theory described above, a new method for optimization of the membership functions of type-2 fuzzy systems based on the level of uncertainty, applied to the integration of response in Modular Neural Networks (MNN), is proposed.

The goal of the research is the development of a method to optimize the modular neural networks including the module of response integration. In particular, the method includes optimizing the membership functions of the type-2 fuzzy system that performs the integration of responses in the modular network, as well as the optimization of the complete architecture of the modular neural network, which consists of the number of modules, layers and neurons.

The purpose of obtaining the optimal architecture is to obtain better recognition rates and improve the efficiency of the hybrid system of pattern recognition. In figure 1 we see the specific points to the general scheme of the method.

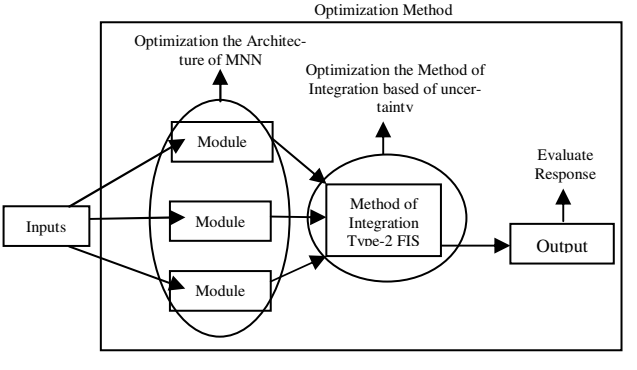


Fig. 1. General scheme of the optimization method.

First, the optimization of the complete architecture of the modular neural network, as number of modules, layers and neurons is performed. Later, the genetic algorithm for optimizing the method of integration, where this method of integration uses type-2 fuzzy inference systems. After obtaining the optimization of the architecture of the MNN with Application in Multimodal Biometry and the algorithm of the integration method, then a method of optimization is implemented and then the validation of results and conclusions for comparing statistically the method.

4 Fuzzy Systems Optimization Based on the Level of Uncertainty

For the Fuzzy Inference Systems optimization (which are the methods of response integration in the case of MNNs) based in level of uncertainty; the first step is to obtain the optimal type-1 Fuzzy Inference System, which allows us to find the uncertainty of their membership functions. In this case we used a parameter to represent (epsilon) the uncertainty. For the next step we used three cases to manage the using genetic algorithms for all situations. We show in figures 2, 3 and 4 the graphical representation of differents cases of uncertainty in the membership functions.

Case 1. Equal values of uncertainty for all membership function.

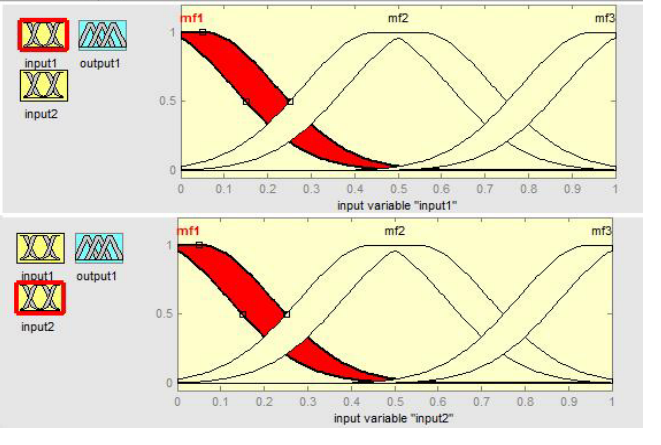


Fig. 2. Graphical representation of equal values of uncertainty for all membership functions

Case 2. Different values of uncertainty in each input.

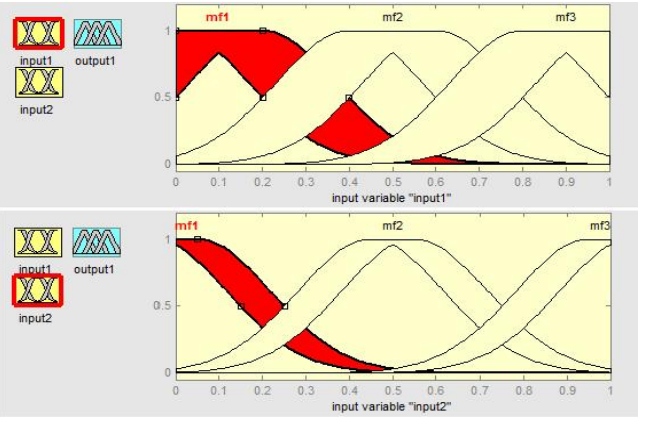


Fig. 3. Graphical representation of different values of uncertainty in each input

Case 3. Different values of uncertainty for each membership function.

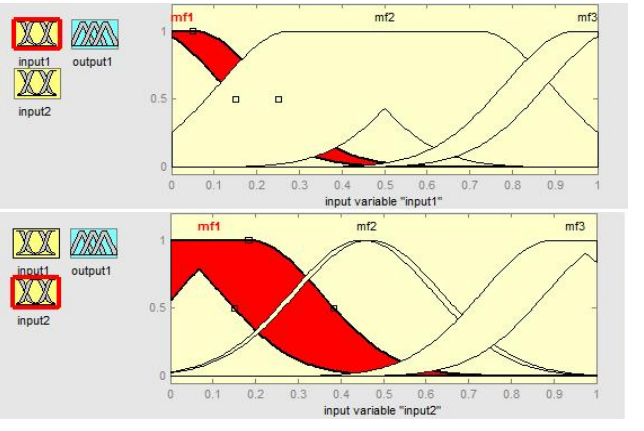


Fig. 4. Graphical representation of different value of uncertainty for each membership function

We proposed an index to evaluate the fitness of the fuzzy systems.

$$Index = \frac{N}{1+n}$$

Where N = Data inside the interval output between the total output data; n = Number of fuzzy systems and = Value of increased uncertainty of the membership functions.

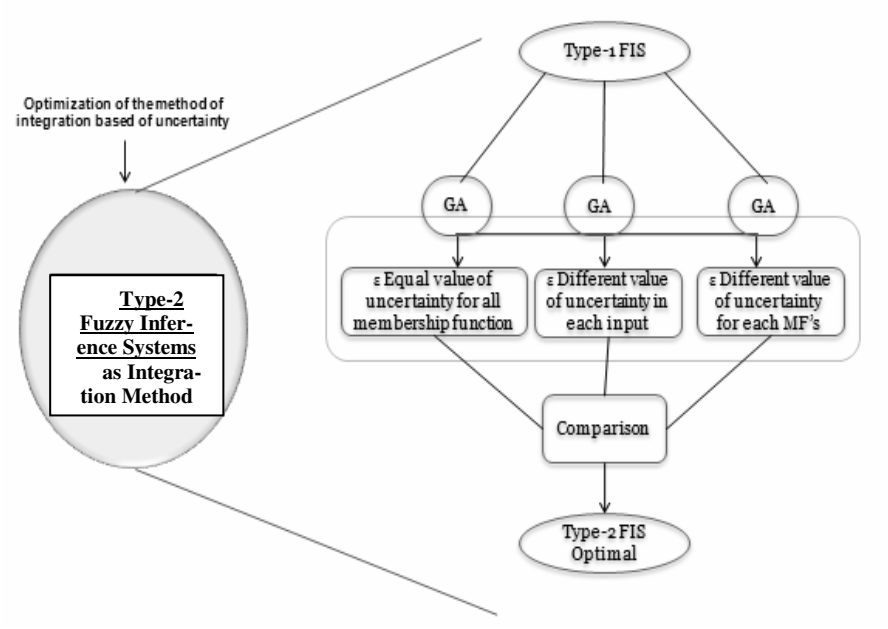


Fig. 5. Representation of Optimization to Type-2 Fuzzy Inference Systems based on level of uncertainty

We consider the optimization of type-2 fuzzy systems based on the above mentioned three cases. As a consequence our proposed method is illustrated in Fig. 5.

Figure 5 represents the proposed optimization method. First, based on a type-1 fuzzy system we use genetic algorithms to optimize the type-2 membership functions of these fuzzy systems taking into account the three cases described above. Then we can compare them to decide which is best and gives results so get an optimal type-2 fuzzy system.

5 Simulation Results

The simulation results for two benchmark problems are presented in this section.

5.1 Adaptive Noise Cancellation

The first problem is of known as the Adaptive Noise Cancellation (ANC) of a transmitted signal. The objective of ANC is to filter out an interference component by identifying a linear model between a measurable noise source and the corresponding unmeasurable interference. ANC using linear filters has been used successfully in real-world applications such as interference canceling in electrocardiograms, echo elimination on long-distance telephone transmission lines, and antenna sidelobe interference canceling. The original data base contains 601 data.

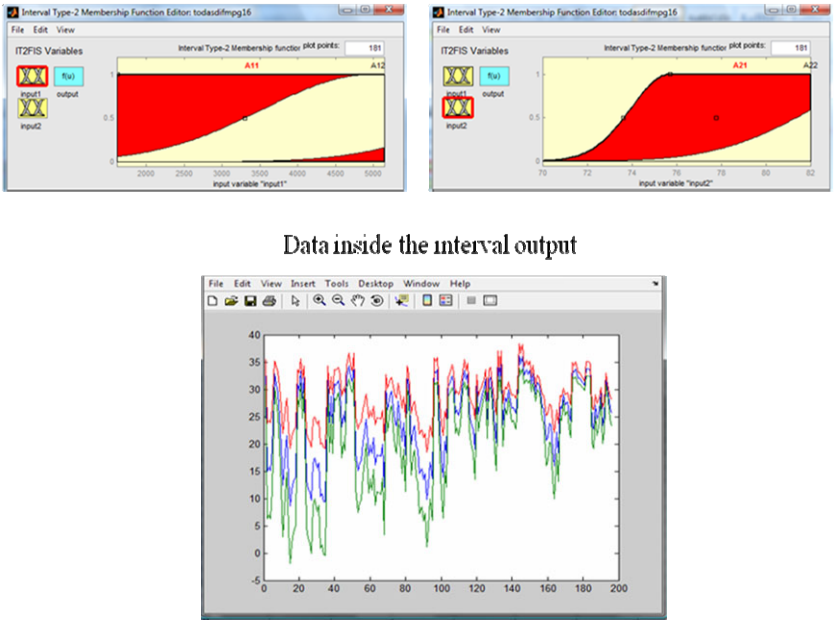


Fig. 6. Data inside the interval output, where red line indicates the upper value in the interval output, green line indicates the lower value in the interval output and blue line indicates the original output value of the fuzzy inference system.

In figure 6 we can see how the inputs in Fuzzy Inference Systems affect the output, to find the largest number of Data inside the interval output.

Jang et al. in [8] found the best type-1 fuzzy inference system with two membership function for each input. Based on those results, type-2 fuzzy inference system are obtained by manually disturbing its value with an epsilon in the membership functions of the two most significant variables and reducing the input data for training. For the first case, 20 fuzzy systems were obtained by increasing the values of equal epsilon by 10%. For case two, where the increase of epsilon is different per input, 40% increase for the first input and 20% in second input; and for the third case, where the increase of epsilon is different per membership function, 15% 30% 20% 5% increases, respectively.

Table 1 shows the simulations results with manually increase for Equal Value of Uncertainty for all Membership functions, for the benchmark problem.

Table 2, shows the simulations results with manually increase for different value of uncertainty in each input for the benchmark problem.

Table 3, shows the simulations results with manually increase for different value of uncertainty in each Membership function (15% 30% 20% 5% Increase respectively) for the benchmark problem.

Table 1. Equal value of uncertainty with manually increase (10% Increase)

No.	N	n	Epsilon ( )	Index	Data inside the interval output
Base	0.3794	20	0	0.13	228/601
1	0.5441	20	0.1	0.18	327/601
2	0.7388	20	0.2	0.25	444/601
3	0.8170	20	0.3	0.27	491/601
4	0.8353	20	0.4	0.28	502/601
5	0.8386	20	0.5	0.28	504/601
6	0.8469	20	0.6	0.28	509/601
7	0.8552	20	0.7	0.29	514/601
8	0.8686	20	0.8	0.29	522/601
9	0.8752	20	0.9	0.29	526/601
10	0.9085	20	1.0	0.30	546/601
11	0.9168	20	1.1	0.31	551/601
12	0.9218	20	1.2	0.31	554/601
13	0.9251	20	1.3	0.31	556/601
14	0.9251	20	1.4	0.31	556/601
15	0.9268	20	1.5	0.31	557/601
16	**0.9285**	**20**	**1.6**	**0.31**	**558/601**
17	0.9285	20	1.7	0.31	558/601
18	0.9285	20	1.8	0.31	558/601
19	0.9285	20	1.9	0.31	558/601
20	0.9285	20	2.0	0.31	558/601

Table 2. Different value of uncertainty in each input with manually increase (40% Increase first input and 20% in second input)

No.	N	n	Epsilon ( )	Index	Data inside the interval output
Base	0.3794	20	0	0.05	228/601
1	0.8236	20	0.3	0.12	495/601
2	0.8453	20	0.6	0.12	508/601
3	0.9185	20	0.9	0.13	552/601
4	0.9251	20	1.2	0.13	556/601
5	**0.9285**	**20**	**1.5**	**0.13**	**558/601**
6	0.9285	20	1.8	0.13	558/601
7	0.9285	20	2.1	0.13	558/601
8	0.9285	20	2.4	0.13	558/601
9	0.9285	20	2.7	0.13	558/601
10	0.9285	20	3	0.13	558/601
11	0.9285	20	3.3	0.13	558/601
12	0.9285	20	3.6	0.13	558/601
13	0.9285	20	3.9	0.13	558/601
14	0.9285	20	4.2	0.13	558/601
15	0.9285	20	4.5	0.13	558/601
16	0.9285	20	4.8	0.13	558/601
17	0.9285	20	5.1	0.13	558/601
18	0.9285	20	5.4	0.13	558/601
19	0.9285	20	5.7	0.13	558/601
20	0.9285	20	6.0	0.13	558/601

Table 3. Different value of uncertainty in each Membership function with manually increase (15% 30% 20% 5% Increase respectively)

No.	N	n	Epsilon ( )	Index	Data inside the interval output
Base	0.3794	20	0	0.05	228/601
1	0.7238	20	0.35	0.09	435/601
2	0.8186	20	0.7	0.10	492/601
3	0.8303	20	1.05	0.10	499/601
4	0.8436	20	1.4	0.11	507/601
5	0.8619	20	1.75	0.11	518/601
6	0.8852	20	2.1	0.11	532/601
7	0.9085	20	2.45	0.11	546/601
8	0.9218	20	2.8	0.12	554/601
9	0.9268	20	3.15	0.12	557/601
10	**0.9285**	**20**	**3.5**	**0.12**	**558/601**
11	0.9285	20	3.85	0.12	558/601
12	0.9285	20	4.2	0.12	558/601
13	0.9285	20	4.55	0.12	558/601
14	0.9285	20	4.9	0.12	558/601
15	0.9285	20	5.25	0.12	558/601
16	0.9285	20	5.6	0.12	558/601
17	0.9285	20	5.95	0.12	558/601
18	0.9285	20	6.3	0.12	558/601
19	0.9285	20	6.65	0.12	558/601
20	0.9285	20	7	0.12	558/601

Figure 7 shows the results graphically for benchmark plroblem ANC, in which we can see the results for the three cases. We can observe that in case two, where the increase of epsilon is different for each input, we get the best index with less increase of uncertainty in their membership functions.

Later, we used a Genetic Algorithms to optimize the epsilon value in the membership functions of a type-2 fuzzy inference system. Next we show the simulation results.

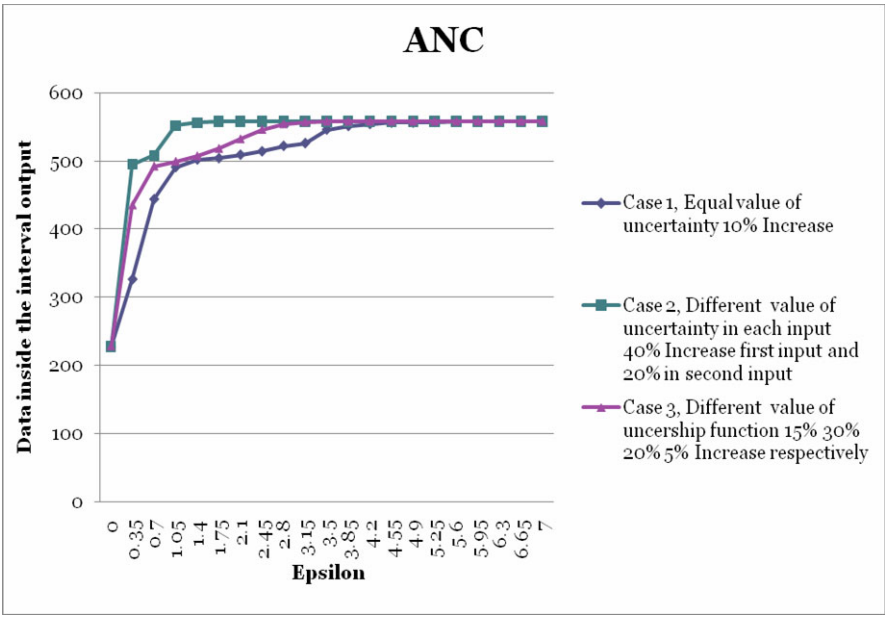


Fig. 7. Simulation results for the three cases for benchmark problem ANC

Table 4. Equal Value of Uncertainty

No.	Individuals	Generation	Mutation	Crossover	Time Execution	Epsilon	Data inside the interval output
1	80	100	0.1	0.4	1:50:28	1.00183	524/601
2	150	200	0.3	0.7	6:36:42	1	525/601
3	100	100	0.001	0.5	2:13:45	1.02672	526/601
4	40	50	0.8	0.9	0:55:31	1	529/601
5	30	30	0.5	0.6	0:12:26	1.00441	530/601
6	50	50	1	1	0:33:36	1.00355	530/601
7	60	30	0.4	0.85	0:46:34	1	538/601
8	35	30	0.95	1	0:26:21	1	552/601
9	25	30	0.8	1	0:19:02	1	556/601
10	15	100	1	1	0:20:36	1	564/601
11	10	100	0.9	1	0:13:20	1	565/601
12	10	30	0.6	0.8	0:08:18	1.502	579/601
13	5	30	0.7	0.9	0:04:04	2	597/601
14	20	30	0.5	0.7	0:16:09	2	600/601
15	**15**	**30**	**0.35**	**0.85**	**0:12:11**	**3.4934**	**601/601**

Table 5. Different value of uncertainty in each input

No.	Individuals	Generation	Mutation	Crossover	Time Execution	Epsilon First Input	Epsilon Second Input	Data inside the interval output
1	80	100	0.1	0.4	0:50:58	1.500	1.500	584/601
2	150	200	0.3	0.7	3:09:31	2.000	1.663	597/601
3	100	100	0.001	0.5	1:08:40	1.505	1.501	580/601
4	40	50	0.8	0.9	0:25:55	1.300	2.693	572/601
5	30	30	0.5	0.6	0:11:48	2.000	1.000	598/601
6	50	50	1	1	0:33:17	2.064	1.061	599/601
7	60	30	0.4	0.85	0:15:25	2.000	3.459	597/601
8	35	30	0.95	1	0:13:28	2.000	2.000	597/601
9	25	30	0.8	1	0:09:44	1.500	2.500	597/601
10	15	100	1	1	0:10:09	2.000	1.167	599/601
11	10	100	0.9	1	0:06:37	1.500	1.520	588/601
12	10	30	0.6	0.8	0:04:07	1.000	1.000	593/601
13	5	30	0.7	0.9	0:01:57	1.001	1.000	582/601
14	**20**	**30**	**0.5**	**0.7**	**0:07:56**	**2.000**	**1.534**	**601/601**
15	15	30	0.35	0.85	0:05:45	1.000	1.500	567/601

Table 6. Different value of uncertainty in each Membership function

No.	Individuals	Generation	Mutation	Cross over	Time Execution	Epsilon 1th MF 1th Input	Epsilon 2th MF 1th Input	Epsilon 1th MF 2th Input	Epsilon 2th MF 2th Input	Data inside the interval output
1	80	100	0.1	0.4	2:01:39	2.000	1.001	1.000	1.000	592/601
2	150	150	0.3	0.7	5:34:46	1.500	2.000	3.981	3.713	597/601
3	100	100	0.001	0.5	2:16:09	1.510	2.048	3.906	3.571	593/601
4	40	50	0.8	0.9	0:24:59	2.000	1.588	3.237	2.332	598/601
5	30	30	0.5	0.6	0:11:23	1.500	2.000	3.747	3.500	601/601
6	50	50	1	1	0:33:06	2.000	1.000	1.009	1.000	595/601
7	60	30	0.4	0.85	0:23:07	2.000	1.000	1.012	1.047	592/601
8	35	30	0.95	1	0:14:47	4.000	3.860	3.519	4.000	601/601
9	25	30	0.8	1	0:09:40	2.003	3.625	4.000	2.600	601/601
10	15	100	1	1	0:20:56	1.000	1.000	1.025	1.000	560/601
11	10	100	0.9	1	0:13:15	2.031	2.000	3.465	1.830	576/601
12	**10**	**30**	**0.6**	**0.8**	**0:04:00**	**1.006**	**2.000**	**3.252**	**2.725**	**601/601**
13	5	30	0.7	0.9	0:01:56	1.047	1.698	1.640	1.523	600/601
14	20	30	0.5	0.7	23:07:16	3.963	4.000	1.979	1.981	601/601
15	15	30	0.35	0.85	0:05:58	1.000	2.943	3.997	3.594	601/601

Table 4, shows the simulations results with a GA for Equal Value of Uncertainty for all Membership Functions for the benchmark problem.

Table 5, shows the simulations results with a GA for different value of uncertainty in each input for the benchmark problem.

Table 6, shows the simulations results with a GA for different value of uncertainty in each Membership function for the benchmark problem.

5.2 MPG Benchmark Problem

The second benchmark problem is of known as automobile MPG (Miles per Gallon) prediction. This problem is a typical nonlinear regression problem, in which several attributes (inputs variables) as number of cylinders, weight, model years, etc., are used to predict another continuous attribute (output variable) in this case MPG. The original data base contains 384 data, the first 192 were used for training data and the other 192 for testing. Jang et al. in [8], found the best model takes "weight" and "model year" as the inputs variables for the type-1 fuzzy inference

Table 7. Equal value of uncertainty with manually increase (10% Increase)

No.	N	n	Epsilon ( )	Index	Data inside the interval output
Base	0.3418	20	0	0.11	67/192
1	0.602	20	0.1	0.2	118/192
2	0.7296	20	0.2	0.24	143/192
3	0.7857	20	0.3	0.26	154/192
4	0.7908	20	0.4	0.26	155/192
5	0.8061	20	0.5	0.27	158/192
6	0.8265	20	0.6	0.28	162/192
7	0.8469	20	0.7	0.28	166/192
8	0.8571	20	0.8	0.29	168/192
9	0.8571	20	0.9	0.29	168/192
10	0.8622	20	1.0	0.29	169/192
11	0.8622	20	1.1	0.29	169/192
12	0.8622	20	1.2	0.29	169/192
13	0.8622	20	1.3	0.29	169/192
14	0.8622	20	1.4	0.29	169/192
15	0.8622	20	1.5	0.29	169/192
16	0.8622	20	1.6	0.29	169/192
17	0.8622	20	1.7	0.29	169/192
18	0.8622	20	1.8	0.29	169/192
19	**0.8673**	**20**	**1.9**	**0.29**	**170/192**
20	0.8673	20	2.0	0.29	170/192

Table 8. Different value of uncertainty in each input with manually increase (30% Increase first input and 10% in second input)

No.	N	n	Epsilon ( )	Index	Data inside the interval output
Base	0.3418	20	0	0.07	67/192
1	0.6531	20	0.2	0.13	128/192
2	0.7602	20	0.4	0.15	149/192
3	0.8214	20	0.6	0.16	161/192
4	0.852	20	0.8	0.17	167/192
5	0.8571	20	1.0	0.17	168/192
6	0.8622	20	1.2	0.17	169/192
7	0.8622	20	1.4	0.17	169/192
8	0.8622	20	1.6	0.17	169/192
9	0.8622	20	1.8	0.17	169/192
10	0.8673	20	2.0	0.17	170/192
11	0.8827	20	2.2	0.18	173/192
12	**0.8929**	**20**	**2.4**	**0.18**	**175/192**
13	0.8827	20	2.6	0.18	173/192
14	0.8622	20	2.8	0.17	169/192
15	0.8469	20	3.0	0.17	166/192
16	0.8469	20	3.2	0.17	166/192
17	0.8469	20	3.4	0.17	166/192
18	0.8469	20	3.6	0.17	166/192
19	0.8469	20	3.8	0.17	166/192
20	0.8469	20	4.0	0.17	166/192

system with two membership function for input. Based on those results to get a type-2 fuzzy inference system as a base to disturb its value with an epsilon in the membership functions of the two most significant variables and reducing the input data for training, the following results are obtained.

For the first case, 20 fuzzy systems were obtained by increasing the values of equal epsilon by 10%. For case two where the increase of epsilon is different per input, 30% increase for the first input and 10% in second input; and for the third

Table 9. Different value of uncertainty in each Membership function with manually increase (15% 5% 10% 20% Increase respectively)

No.	N	n	Epsilon ( )	Index	Data inside the interval output
Base	0.3418	20	0	0.1	67/192
1	0.6582	20	0.125	0.19	129/192
2	0.7602	20	0.25	0.22	149/192
3	0.7959	20	0.375	0.23	156/192
4	0.8112	20	0.5	0.23	159/192
5	0.8418	20	0.625	0.24	165/192
6	0.852	20	0.75	0.24	167/192
7	0.8571	20	0.875	0.24	168/192
8	0.8571	20	1	0.24	168/192
9	0.8571	20	1.125	0.24	168/192
10	0.8622	20	1.25	0.25	169/192
11	0.8622	20	1.375	0.25	169/192
12	0.8622	20	1.5	0.25	169/192
13	0.8622	20	1.625	0.25	169/192
14	0.8724	20	1.75	0.25	171/192
15	0.8724	20	1.875	0.25	171/192
16	**0.8878**	**20**	**2**	**0.25**	**174/192**
17	0.8673	20	2.125	0.25	170/192
18	0.8571	20	2.25	0.24	168/192
19	0.8469	20	2.375	0.24	166/192
20	0.8469	20	0.125	0.24	166/192

case where the increase of epsilon is different per membership function 15% 5% 10% 20% increase respectively.

Table 7 shows the simulations results with manually increase for Equal Value of Uncertainty for all Membership functions, for the second benchmark problem.

Table 8, shows the simulations results with manually increase for different value of uncertainty in each input for the benchmark problem.

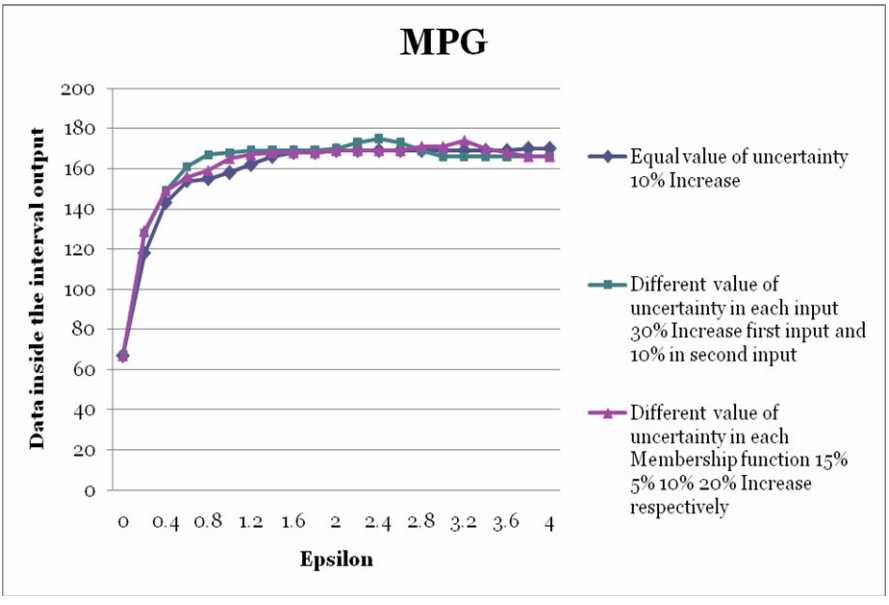


Fig. 8. Simulation results for the three cases for benchmark problem MPG

Table 9, shows the simulations results with manually increase for different value of uncertainty in each Membership function (15% 5% 10% 20% Increase respectively) for the benchmark problem.

Figure 8 shows the results graphically for benchmark problem MPG, in which we can see the results for the three cases. We can observe that in case two, where the increase of epsilon is different for each input, we get the best index with less increase of uncertainty in their membership functions.

6 Conclusions

We presented in this paper a description of an optimization method for the membership functions of type-2 fuzzy systems based on the level of uncertainty. Simulation results for Adaptive Noise Cancellation benchmark problems are presented. For this problem we can see that varying the uncertainty of the membership functions with the GA changes the fitness of the fuzzy systems.

Acknowledgment

We would like to express our gratitude to CONACYT under grant number 175883, UABC University and Tijuana Institute of Technology for the facilities and resources for the development of this research.

References

1. Alvarado-Verdugo, J.M.: Reconocimiento de la persona por medio de su rostro y huella utilizando redes neuronales modulares y la transformada wavelet., Instituto Tecnológico de Tijuana (2006)
2. Ross, A.A., Nandakumar, K., Jain, A.K.: Handbook of Multibiometrics. Springer, Heidelberg (2006)
3. Castro, J.R.: Tutorial Type-2 Fuzzy Logic: theory and applications., Universidad Autónoma de Baja California-Instituto Tecnológico de Tijuana (October 9, 2006), http://www.hafsamx.org/cis-chmexico/seminar06/tutorial.pdf
4. Chen, K., Wang, L.: Trends in Neural Computation. Studies in Computational Intelligence, vol. 35, pp. 339–341. Springer, Heidelberg
5. Cordón, O., Herrera, F., Hoffmann, F., Magdalena, L.: Genetic Fuzzy systems, Evolutionary Tuning and learning of Fuzzy Knowledge Bases. In: Advances in Fuzzy Systems-Applications and Theory, vol. 19. World Scientific, Singapore
6. Hidalgo, D., Melin, P., Castillo, O.: Type-1 and Type-2 Fuzzy Inference Systems as Integration Methods in Modular Neural Networks for Multimodal Biometry and its Optimization with Genetic Algorithms. Journal of Automation, Mobile Robotics & Intelligent Systems 2(1) (2008) ISSN 1897-8649
7. Hidalgo, D., Castillo, O., Melin, P.: Interval type-2 fuzzy inference systems as integration methods in modular neural networks for multimodal biometry and its optimization with genetic algorithms. International Journal of Biometrics 1(1), 114–128 (2008)
8. Jang, J.-S.R., Sun, C.-T., Mizutani, E.: Neuro-Fuzzy and Soft Computing, A Computational Approach to Learning and Machine Intelligence. Prentice Hall, Englewood Cliffs (1997)
9. Karnik, N., Mendel, J.M.: Operations on type-2 fuzzy sets. Signal and Image Processing Institute, Department of Electrical Engineering-Systems. University of Southern California, Los Angeles (2000)
10. Man, K.F., Tang, K.S., Kwong, S.: Genetic Algorithms, Concepts and Designs. Springer, Heidelberg (1999)
11. Melin, P., Castillo, O.: Hybrid Intelligent Systems for Pattern Recognition Using Soft Computing: An Evolutionary Approach for Neural Networks and Fuzzy Systems. Studies in Fuzziness and Soft Computing (Hardcover - April 29, 2005)
12. Melin, P., Castillo, O., Gómez, E., Kacprzyk, J., Pedrycz, W.: Analysis and Design of Intelligent Systems Using Soft Computing Techniques. In: Advances in Soft Computing, vol. 41. Springer, Heidelberg (2007)
13. Melin, P., Castillo, O., Gómez, E., Kacprzyk, J.: Analysis and Design of Intelligent Systems using Soft Computing Techniques. In: Advances in Soft Computing (Hardcover - July 11, 2007)
14. Mendel, J.M.: UNCERTAIN Rule-Based Fuzzy Logic Systems, Introduction and New Directions. Prentice Hall, Englewood Cliffs (2001)
15. Mendel, J.M.: Uncertainty: General Discussions, Article is provided courtesy of Prentice Hall, By Jerry Mendel, May 11 (2001), http://www.informit.com/articles/article.asp?p=21313
16. Mendel, J.M.: Why We Need Type-2 Fuzzy Logic Sys-tems? Article provided courtesy of Prentice Hall, By Jerry Mendel, May 11 (2001), http://www.informit.com/articles/article.asp?p=21312&rl=1

17. Mendel, J.M., Bob-John, R.I.: Type-2 Fuzzy Sets Made Simple. IEEE Transactions on Fuzzy Systems 10(2) (April 2002)
18. Mendoza, O., Melin, P., Castillo, O., Licea, P.: Type-2 Fuzzy Logic for Improving Training Data and Response Integration in Modular Neural Networks for Image Recognition. In: Melin, P., Castillo, O., Aguilar, L.T., Kacprzyk, J., Pedrycz, W. (eds.) IFSA 2007. LNCS (LNAI), vol. 4529, pp. 604–612. Springer, Heidelberg (2007)
19. Mendoza, O., Melin, P., Castillo, O., Licea, P.: Modular Neural Networks and Type-2 Fuzzy Lo- gic for Face Recognition. In: Reformat, M. (ed.) Proceedings of NAFIPS 2007, San Diego, June 2007, vol. (1), IEEE, Los Alamitos (2007) (pages CD Rom)
20. Ramos-Gaxiola, J.: Redes Neuronales Aplicadas a la Identificación de Locutor Mediante Voz Uti lizando Extracción de Características., Instituto Tecnológico de Tijuana (2006)
21. Urias, J., Hidalgo, D., Melin, P., Castillo, O.: A Method for Response Integration in Modular Neural Networks with Type-2 Fuzzy Logic for Biometric Systems. In: Melin, P., et al. (eds.) Analysis and Design of Intelligent Systems using Soft Computing Techniques, 1st edn. Studies in Fuzziness and Soft Computing, vol. 1(1), pp. 5–15. Springer, Germany (2007)
22. Urias, J., Melin, P., Castillo, O.: A Method for Response Integration in Modular Neural Networks using Interval Type-2 Fuzzy Logic. In: FUZZ-IEEE 2007, London, UK, July 2007. FUZZ, vol. (1), pp. 247–252. IEEE, Los Alamitos (2007)
23. Urias, J., Hidalgo, D., Melin, P., Castillo, O.: A New Method for Response Integration in Modular Neural Networks Using Type-2 Fuzzy Logic for Biometric Systems. In: The 2007 Inter- national Joint Conference on Neural Networks, IJCNN 2007 Conference Proceedings, Orlando, Florida, USA, August 12-17. IEEE, Los Alamitos (2007)
24. Zadeh, L.A.: Fuzzy Logic = Computing with Words. IEEE Transactions on Fuzzy Systems 4(2), 103 (1996)
25. Zadeh, L.A.: Knowledge representation in Fuzzy Logic. IEEE Transactions on knowledge data engineering 1, 89 (1989)
26. Zadeh, L.A.: Fuzzy Logic. Computer 1(4), 83–93 (1998)
27. Zadeh, L.A.: Soft Computing and Fuzzy Logic. IEEE Software 11(6), 48–56 (1994)
28. Zadeh, L.A., Bernadette, B.M., Ronald, R.Y.: Fuzzy Logic and Soft Computing. In: Advances in Fuzzy Systems-Applications and Theory, Septiembre 1995, vol. 4 (1995), ISBN:978-981-02-2345-8, 981-02-2345-5

Modular Neural Network for Human Recognition from Ear Images Using Wavelets

Lizette Gutiérrez, Patricia Melin, and Miguel López

Tijuana Institute of Technology, Tijuana México
lizette.gutierrez@hotmail.com, pmelin@tectijuana.mx,
danym23@aol.com

Abstract. This work is focused in the human recognition from ear images as biometric using modular neural networks with preprocessing ear images as network inputs. We proposed a modular neural network architecture composed of twelve modules, in order to simplify the problem making it smaller. Comparing with other biometrics, ear recognition has one of the best performances, even when it has not received much attention. To compare with other existing methods, we used the 2D Wavelet analysis with global thresholding method for compression, and Sugeno Measures and Winner-Takes-All as modular neural network integrator. Recognition results achieved was up to 97%.

1 Introduction

Is not a surprise to see that actually the human identifying methods by possessions such a cards, badges, keys or by knowledge such a passwords, userid, Personal Identification Number (PIN), are being replaced by biometrics.

We can say that every person is unique. There are not two persons with the same face, identical fingerprints or voice. This and other singularities are part of the biometrics. Biometrics is the science of identifying or verifying the identity of a person based on physiological or behavioral characteristics. In physiological also called passive characteristics, the biometric measure is taken from a subject with or without his/her consent, depends of the biometric capture system. In the other way the behavioral also called active characteristics, the subject must show his/her biometric measure doing an action in front of a captured system.

Biometrics offer much higher accuracy than the more traditional ones. Possession can be lost, forgot or replicated easily. Knowledge can be forgotten. Both possessions and knowledge can be stolen or shared with other people. In biometrics these drawbacks do exist only in small scale. [7]

Some biometrics are shown at (Figure 1). The behavioral characteristics are voice, handwritten signature, keyboard strokes, odor, and more. The physiological ones are fingerprint, iris, face, hand geometry, finger geometry, retina, vein structure, ear, and more.

P. Melin et al. (Eds.): Soft Comp. for Recogn. Based on Biometrics, SCI 312, pp. 121–135.
springerlink.com

Fig. 1. Some Biometrics.

The physiological characteristics systems are generally more reliable than the based ones on behavioral characteristics, despite the last ones can be sometimes simpler to integrate in some specific uses.

The most commonly used biometrics according to the International Biometric Group in 2006 were: fingerprint, face, voice, iris, handwritten signature and hand geometry. [10] Ear biometric is not commonly used, yet (Figure 2).

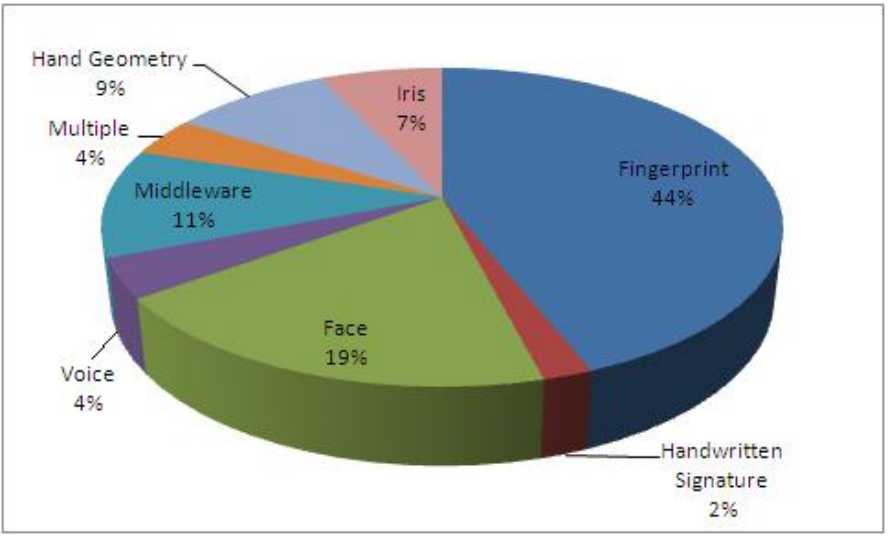


Fig. 2. Commonly used Biometrics.

There are many applications where biometrics can be used. Basically a biometric system may operate in verification mode also known as authentication, or identification mode also known as recognition. Identification mode answer the question Who are you?, the system recognizes a subject by searching the templates of all the users in the database for a match. In verification mode the question is, You are who you claim to be? and the system validates a person's identity by comparing the captured biometric data with her own biometric.

But, what biological measurements qualify to be a biometric? Any human physiological and/or behavioral characteristic can be used as a biometric characteristic as long as it satisfies the following requirements: universal, unique, permanent, measurable, acceptable, capable and reliable. These concepts are described below:

- Universality (U): each person must own the characteristic.
- Distinctiveness (D): each person must be differentiable between each subject.
- Permanence (P): the characteristic does not have to change with time.
- Collectability (Co): there must exist the capacity to characterize the characteristic measured quantitatively.
- Acceptability (A): the characteristic must have great acceptance between the societies.
- Performance (Pf): accuracy, speed, and robustness of technology used.
- Circumvention (C): reflects how easily the system can be fooled using fraudulent methods.

A brief comparison of different biometrics based on seven factors for an ideal biometric previously mentioned, is provided in (Table I). As you can see, ear, fingerprint and hand geometry have better averages than other biometrics.

Table I. Comparison for different biometrics. High, Medium, and Low are denoted by H, M, and L, respectively.

Biometric Techniques	**U**	**D**	**P**	**Co**	**A**	**Pf**	**C**
Ear	M	M	H	M	M	H	M
DNA	H	H	H	L	H	L	L
Face	H	L	M	H	L	H	H
Fingerprint	M	H	H	M	H	M	M
Keystroke	L	L	L	M	L	M	M
Hand G.	M	M	M	H	M	M	M
Iris	H	H	H	M	H	L	L
Odor	H	H	H	L	L	M	L
Retina	H	H	M	L	H	L	L
Signature	L	L	L	H	L	H	H
Voice	M	L	L	M	L	H	H

However, no single technique can outperform all the others in all operational environments. In this sense, each biometric technique is admissible and there is no optimal biometric characteristic. [5]

2 Background

2.1 Related Work

Human ear identification has been interesting in recent years. At 1906 Imhofer found that in a set of 500 ears, only 4 characteristics were needed to state the ears unique. The most famous work among ear identification was made by Alfred Iannarelli at 1989, where he compared over 10.000 ears drawn from a randomly selected sample in California; he found that all ears were different. [3]

Another study was among identical and non-identical twins using Iannarelli's measurements. The result was that ears are not identical, even identical twins had similar but not identical ears. After Iannarelli's classification there have become different and more scientific methods for ear identification:

Carreira-Perpiñan (1995). He used outer ear images are proposed for human recognition and compression neural networks to reduce the dimensionality of the images and produce a feature vector of manageable size. [9]

Moreno et al. (1999). They presented a multiple identification method, which combines the results from several neural classifiers using feature outer ear points, information obtained from ear shape and wrinkles, and macro features extracted by compression network.

Burge and Burger (1998 - 2000). They obtained automatic ear biometrics with Voronoi diagram of its curve segments.

Hurley, Nixon and Carter (2000). They used force field transformations for ear recognition. The image is treated as an array of Gaussian attractors that act as the source of the force field.

Victor, Chang, Bowyer and Sarkar (2002 - 2003). Using Principal Component Analysis (PCA) approach, they made a comparison between ears and faces.

Ear Recognition Laboratory at USTB (2003 - 2004). In 2003, they established an image database of 60 subjects (3 images for one subject). Using the method of kernel principal component analysis (KPCA), the ear identification rate was 94%. In 2004, they enlarged the image database to 77 subjects, 4 images for one subject, with pose variation and lighting variation. For ear feature extraction, they proposed a novel recognition method based on local features extraction, which means extracting the shape feature of outer ear and the structural feature of inner ear then using BP neural network for classification. In this way, the recognition rate rises to 85%. [1][11]

2.2 Ear

The ear structure is quite complex, it has a great variety of classified zones. The most important ear parts are: Helix, Concha and Lobule (Figure 3).

Fig. 3. Ear parts.

Researchers have suggested that the shape and appearance of the human ear is unique to each person, and relatively unchanging during the lifetime of an adult [6] making it better suited for long-term identification when compared to other biometrics, such as face recognition.

Ear recognition is not affected by environmental factors such as mood, health, and clothing. However, ear recognition has not received much attention like others biometrics e.g. face, fingerprint, iris, etc.

For that reasons, the ear is taken as biometric, since it presents a structural singularity that is practically impossible that two subjects own the same physiological characteristic, even for "identical" twins.

We can mention some advantages and disadvantages for ear biometric:

- Ears are smaller than other biometrics. Reduced spatial resolution.
- Ear biometric can be take with or without the subject consent.
- Biometrics like iris, retina and DNA are more permanent than ear form. At the same level are fingerprint and hand geometry. Less permanent than ear form are signature, face and voice.
- Ear biometrics are not usable if the ear is covered e.g. with a hat or hair.
- We have almost none adjectives to describe ears.
- Ear recognition systems can be fooled with methods like plastic surgery.

3 Ear Recognition Process

Ear recognition process used for this work is composed by the following steps:

- Data acquisition
- Image pre-processing
  - Regions of interest (ROI)
  - Wavelets
- Neural network structure
- Neural network training
  - Scaled Conjugate Gradient (SCG)
  - Gradient Descent with Momentum and Adaptive Learning Rate (GDX)
- Modular integration
  - Sugeno Measures
  - Winner-Takes-All (WTA)

In the next sections we describe these steps, and we show the results obtained.

3.1 Data Acquisition

The images were acquired at University of Science and Technology Beijing (USTB) [11], which contains 308 images from 77 subjects, 4 images for one subject with pose variation and lighting variation. The subjects are students and teachers from the department of Information Engineering. The database was formed between November 2003 and January 2005 with 77 subjects, four images each. Two images with angle variation and one with illumination variation. Each image is 24-bit true color image and 300*400 pixels. The first image and the fourth one are both profile image but under different lighting. The second and the

Fig. 4. USTB Database.

third one have the same illumination condition with the first while they have separately rotated +30 degree and -30 degree with the first one (Figure 4).

3.2 Image Pre-processing

The original image is true color or RGB. To get the coefficients, the image must to have a color map, it means, the image must be indexed. First of all, we convert the image to grayscale and then from RGB to indexed.

The figures (Figure 5 and Figure 6) below show how the system interprets the original RGB image and how interprets the indexed image. As we can see, the RGB image lost too much information.

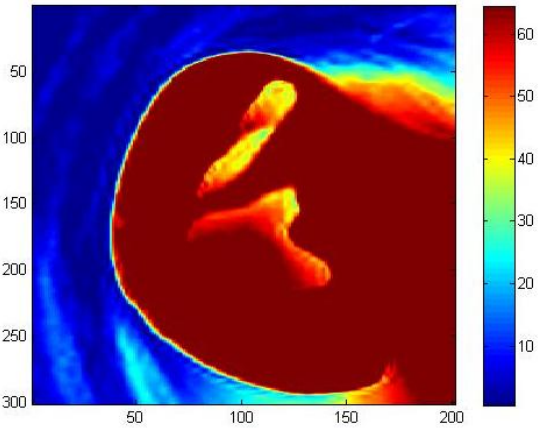


Fig. 5. RGB Image.

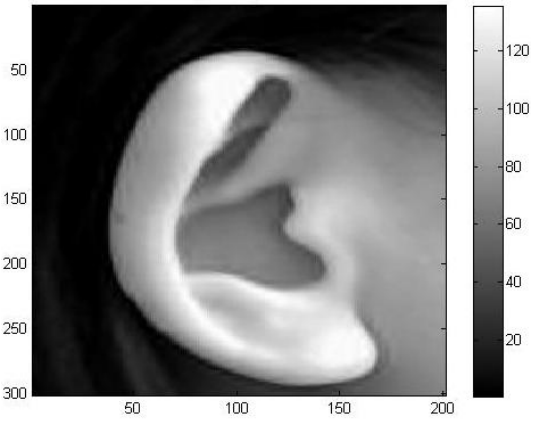
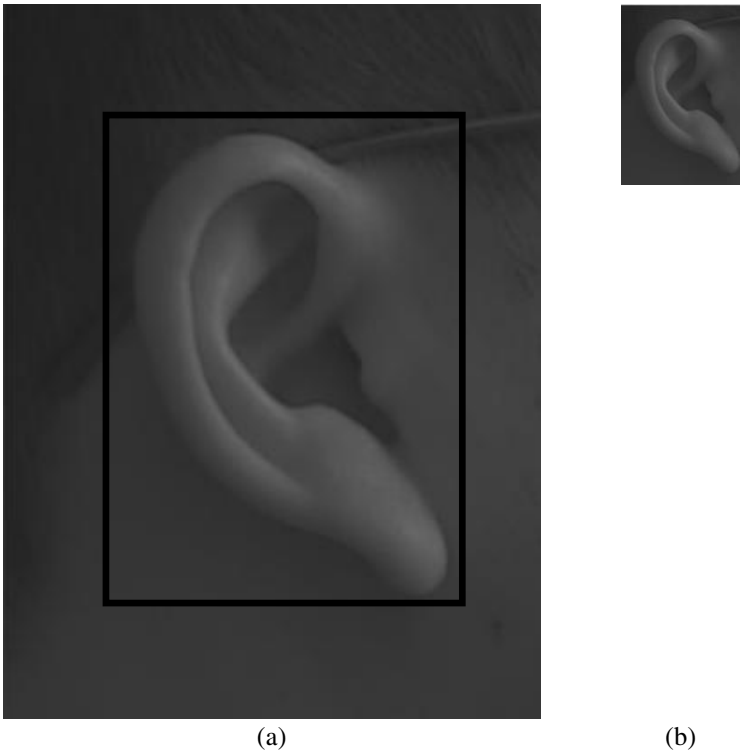


Fig. 6. Indexed Image.

The next step for image pre-processing was the image resize from 300*400 to 50*75 pixels taken the region of interest (ROI) to eliminate as much as possible noise (Figure 7).

For image compression, we used two-dimensional wavelet analysis with Global Thresholding method, with two decompose levels and near symmetric wavelet (sym8) and 20% for hard-thresholding. The approximation coefficients were stored in a row vector for training.



(a) (b)

Fig. 7. (a) 0riginal Image. (b) Resize Image.
Black square = ROI

Basically, this method consists of taking the wavelet expansion of the signal and keeping the largest absolute value coefficients. In this case, you can set a global threshold, a compression performance, or a relative square norm recovery performance. Thus, only a single parameter needs to be selected.

The following figures (Figure 8 and Figure 9) show the decomposition and compression process.

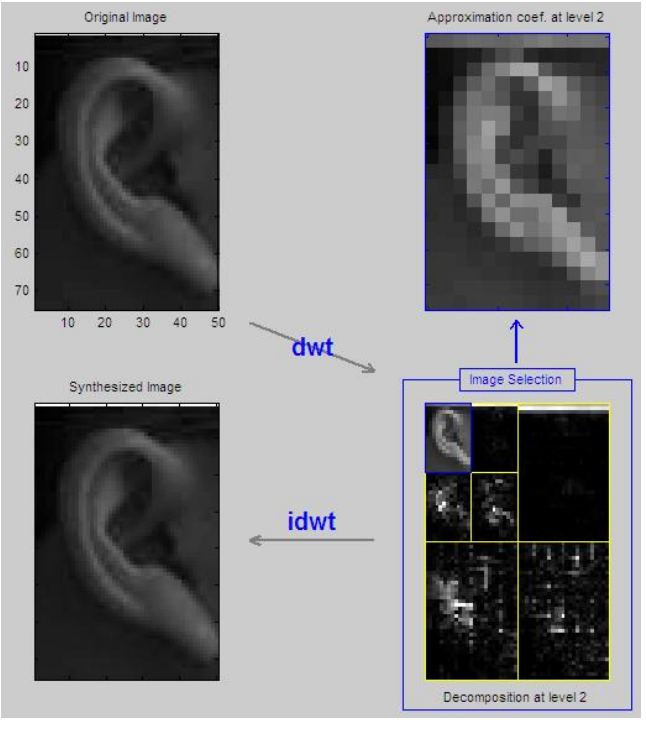


Fig. 8. Image Decomposition.

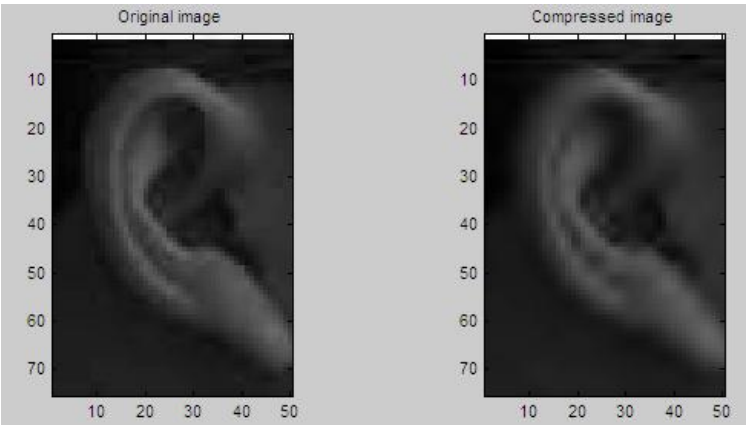


Fig. 9. Image Compression.

3.3 Neural Network Structure

The use of monolithic neural networks, such as a multilayer perceptrón, has some drawbacks: e.g. slow learning, weight coupling, and the black box effect. These can be alleviated by the use of a modular neural network (MNN). The creation of a MNN has three steps: task decomposition, module creation and decision integration. [4]

For our investigation, we decomposed our network data by this way: 308 training images, divided in 104 images for module 1 and 2, and 100 images for module 3; 77 identification images, divided in 26 images for module 1 and 2, and 25 for module 3 (Figure 10). Then, each module was divided as follows (Figure 11):

- Module 1 (Subjects 1-26)
  - Module 4: Helix
  - Module 5: Concha
  - Module 6: Lobule
- Module 2 (Subjects 27-52)
  - Module 7: Helix
  - Module 8: Concha
  - Module 9: Lobule
- Module 3 (Subjects 53-77)
  - Module 10: Helix
  - Module 11: Concha
  - Module 12: Lobule

Using the expression (2*(k+2), k+m, k) [8]:

- First hidden layer: 56 neurons module 1 and 2, 54 module 3.
- Second hidden layer: 30 neurons module 1 and 2, 29 module 3.
- Output layer: 26 neurons module 1 and 2, 25 module 3.

We used Sugeno measures and Winner-Takes-All for integrate the modules 4, 5 and 6, 7, 8 and 9, and 10, 11 and 12. For modules 1, 2 and 3 we used a Gating to take de final decision.

3.4 Neural Network Training

For the 77 subjects, we trained the 4 available images and we used every image with cross validation for testing. It means, for each module we used the first image for test at the first iteration, then the second image for test at the second iteration, and so on until the four image.

The learning algorithms used for this work were gradient descent momentum and an adaptive learning rate or traingdx, and scaled conjugate gradient or trainscg. The tables below (Table II, Table III and Table IV) show the performance for those learning algorithms, being trainscg the best with less number of iterations but not in time.

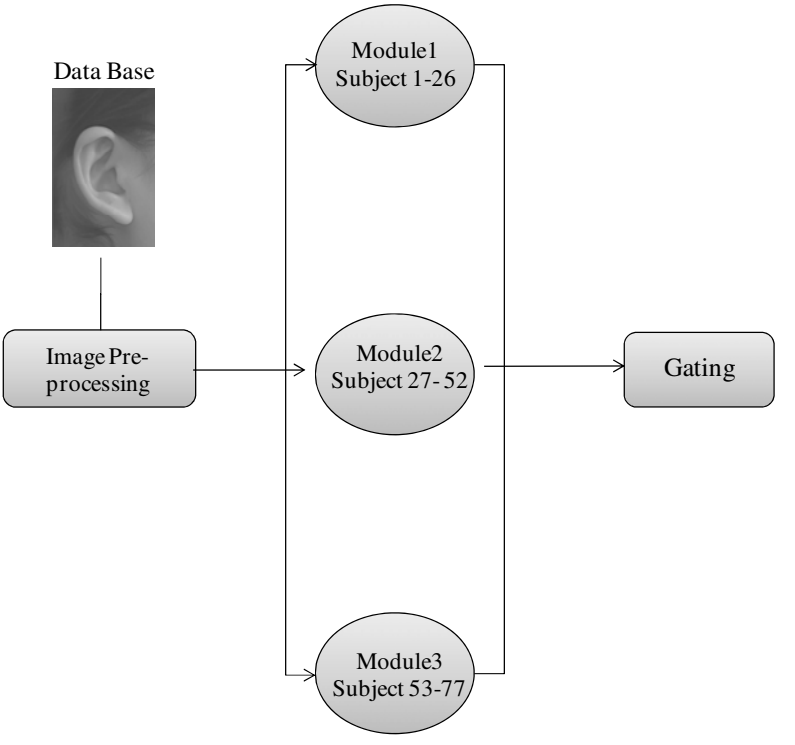


Fig. 10. Schematic representation of the neural network architecture.

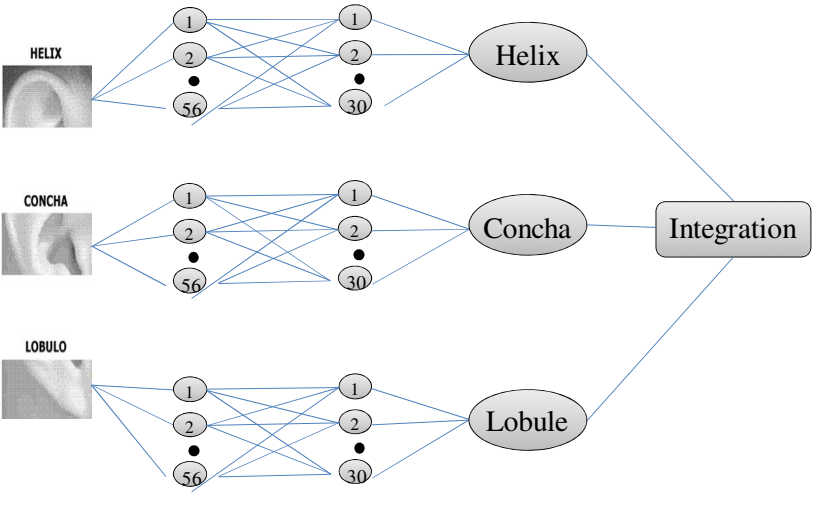


Fig. 11. Neural network architecture for each module.

Table II. Performance for trainscg algorithm.
H=Helix, C=Concha, L=Lobule.

Iterations								
Module 1			Module 2			Module 3		
H	C	L	H	C	L	H	C	L
377	509	608	584	669	948	444	411	400

Table III. Performance for traingdx algorithm.
H=Helix, C=Concha, L=Lobule.

Iterations								
Module 1			Module 2			Module 3		
H	C	L	H	C	L	H	C	L
1000			1000			1000		

Table IV. Time comparison between SCG and GDX.
SCG= trainscg, GDX= traingdx.

Train	Goal	Time
SCG	0.00001	30.37
GDX	0.00001	24.20

3.5 Modular Integration

The integrators used in this work were Sugeno Measures and WTA mechanisms. The following tables (Table V, Table VI and Table VII) show the recognition results obtained for each module using cross validation and Sugeno Measures as integrator.

The tables (Table VIII, Table IX, and Table X) show the recognition results obtained for each module using cross validation and Winner-Take-All as integrator.

Table V. Recognition results for Module 1 with Sugeno integration.
CV = Cross Validation

Train	Module 1				
	CV1	CV2	CV3	CV4	%
trainscg	25/26	26/26	24/26	26/26	97.11
traingdx	23/26	23/26	24/26	22/26	88.46

Table VI. Recognition results for Module 2 with Sugeno integration.
CV = Cross Validation

Train	Module 2				
	CV1	CV2	CV3	CV4	%
trainscg	24/26	25/26	25/26	26/26	96.15
traingdx	22/26	23/26	24/26	23/26	88.46

Table VII. Recognition results for Module 3 with Sugeno integration.
CV = Cross Validation

Train	Module 3				
	CV1	CV2	CV3	CV4	%
trainscg	24/25	25/25	25/25	25/25	99.0
traingdx	23/25	22/25	21/25	22/25	88.0

Table VIII. Recognition results for Module 1 with WTA integration.
CV = Cross Validation

Train	Module 1				
	CV1	CV2	CV3	CV4	%
trainscg	23/26	25/26	26/26	25/26	95.19
traingdx	24/26	23/26	25/26	24/26	92.30

Table IX. Recognition results for Module 2 with WTA integration.
CV = Cross Validation

Train	Module 2				
	CV1	CV2	CV3	CV4	%
trainscg	24/26	23/26	24/26	26/26	93.26
traingdx	26/26	23/26	25/26	24/26	94.23

Table X. Recognition results for Module 3 with WTA integration.
CV = Cross Validation

Train	Module 3				
	CV1	CV2	CV3	CV4	%
trainscg	24/25	23/25	25/25	23/25	95.00
traingdx	24/25	23/25	24/25	23/25	94.00

The table below (Table XI) shows the final average obtained from each module.

Table XI. Final recognition result for Sugeno and WTA integration.

Train	Sugeno Rec.%	WTA Rec.%
trainscg	97.42	94.48
traingdx	88.30	93.51

We can compare our results with the work done by other researcher in the same field using various techniques e.g. [1], [2], [3]. Our recognition rate is good because the modular neural network architecture that we used helped us to avoid slow learning and make easier the recognition process.

4 Conclusions

In this paper, is proposed a method of human recognition based on human ear images using 2D wavelet analysis for preprocessing. Ear images are resized to a fixed size followed by select regions of interest. After that near symmetric wavelet of level two is used to image compression. Ear database is trained following the proposed modular neural network architecture, which help us to improve the train process. In fact, if we want to add more subjects to the database is not necessary start over, we just need to add more modules to our architecture and that is a great advantage comparing with other similar works.

Sugeno Measures integrator gets the best recognition average being this 97.11% with SCG algorithm versus 94.48% with WTA integrator from SCG algorithm. Both integrators are good; however, in this case it was not enough to choose a winner through the neural weights as do WTA integrator.

References

[1] Xie, Z., Mu, Z.: Ear Recognition Using LLE and IDLLE Algorithm. Paper, University of Science and Technology Beijing, China (2008)
[2] Lu, L., Zhang, X., Zhao, Y., Jia, Y.: Ear Recognition Based on Statistical Shape Model. Paper, University of Science and Technology Beijing, China (2008)
[3] Ali, M., Javed, M.Y., Basit, A.: Ear Recognition Using Wavelets. Paper, National University of Sciences and Technology, Peshawar Road, Rawalpindi, Pakistan (2007)
[4] Santos, J., Alexandre, L., Marques de Sa, J.: Modular Neural Network Task Decomposition Via Entropic Clustering. In: Sixth International Conference on Intelligent Systems Design and Applications, ISDA 2006 (2006)
[5] Jain, A.K., Ross, A., Prabhakar, S.: An Introduction to Biometric Recognition. Paper, IEEE Transactions (2004)
[6] Yan, P., Bowyer, K.W.: Empirical Evaluation of Advanced Ear Biometrics. Paper, University of Notre Dame (2005)
[7] Lammi, H.K.: Ear Biometrics. Paper, Lappeenranta University of Technology, Finland (2003)
[8] Salinas, R.: Neural Network Architecture Parametric Face Recognition. Paper, University of Santiago de Chile (2000)
[9] Carreira-Perpinan, M.A.: Compression Neural Networks for Feature Extraction: Application to Human Recognition from Ear Images. Thesis, University Polytechnic of Madrid, Spain (1995)
[10] Commonly used Biometrics Graphic: International Biometric Group, `http://www.biometricgroup.com/` (Reviewed in April 2009)
[11] Data Base: Ear Recognition Laboratory at USTB, `http://www.ustb.edu.cn/resb/en/index` (Reviewed in March 2009)

Modular Neural Networks for Person Recognition Using the Contour Segmentation of the Human Iris Biometric Measurement

Fernando Gaxiola, Patricia Melin, and Miguel López

Tijuana Institute of Technology, Tijuana México
fergaor_29@hotmail.com, pmelin@tectijuana.mx,
malopezr60@hotmail.com.mx

Abstract. This paper presents three modular neural network architectures as systems for recognizing persons based on the iris biometric measurement of humans. In these systems, the human iris database is enhanced with image processing methods, and the coordinates of the center and radius of the iris are obtained to make a cut of the area of interest by removing the noise around the iris. The input to the modular neural networks are the processed iris images and the output is the number of the person identified. The integration of the modules was done with a gating network method.

1 Introduction

This paper is focused on the area of modular neural networks for pattern recognition based on biometric measures, specifically in the recognition by the human iris biometric measurement. At present, biometric measurements are being widely used for person recognition systems. A lot has been said about the use of such measures, particularly for the signature, fingerprint, face and voice. As more research was done in this area further biometric measures were discovered, among which the human iris by its peculiarity of not losing over the years its universality and authenticity.

In order to get a good identification, we proposed a modular neural network architecture divided into three modules, each module input is a part of the database of human iris, and some methods or techniques are used for pre-processing the images such as normalization, resizing, cut, edge detection, among several others. The end result in terms of modular neural network architecture and image preprocessing depends on the tests made and also the time to achieve identification.

The paper is organized as follows: Section 2 contains a brief explanation from previous research with human iris for recognition of people and basic concepts relevant to the area, section 3 defines the method proposed for this research and

P. Melin et al. (Eds.): Soft Comp. for Recogn. Based on Biometrics, SCI 312, pp. 137–153.
springerlink.com

the description of problem addressed in this paper, section 4 presents the results achieved in research and in Section 5 draws conclusions and future work.

2 Backgroun and Basic Concepts

2.1 Modular Neural Network

An artificial neural network (ANN) is a distributed computing scheme based on the structure of the nervous system of humans. The architecture of a neural network is formed by connecting multiple elementary processors, this being an adaptive system that has an algorithm to adjust their weights (free parameters) to achieve the performance requirements of the problem based on representative samples.

Therefore we note that an ANN is a distributed computing system characterized by:

- A set of elementary units, each of which has low processing capabilities.
- A dense interconnected structure using weighted links.
- Free parameters to be adjusted to meet performance requirements.
- A high degree of parallelism.

The most important property of artificial neural networks is their ability to learn from a training set of patterns, i.e. is able to find a model that fit the data [22].

The artificial neuron consists of several parts. On one side are the inputs, weights, the summation, and finally the adapter function. The input values are multiplied by a weights and added: $\sum_i x_i w_{ij}$. This function is completed with the addition of a threshold amount θ i This threshold has the same effect as an entry with value -1. It serves so that the sum can be shifted left or right of the origin. After addition, we have the function f applied to the resulting set the final value of the output, also called yi. The result of the sum before applying the function f, also often called activation value ai [23].

The modular neural networks are composed of simple networks that behave as functional blocks and these are the neural modules.

A modular neural network works similarly to a classical neural network, as it is composed of sigmoidal, linear or discrete activation neurons and are trained with common learning algorithms (gradient descent with adaptive learning algorithm, backpropagation, gradient descent scaling , etc.). What distinguishes it from other neural models, is that it is developed based on functional modules and each module runs a neural network with the same characteristics or different (input layer, hidden layers, output layer, depending activation, learning algorithm, number of neurons per layer, etc.). In this model the modules work independently and in the end a form commonly called integrator performs the function of deciding between

the different modules to determine which of them has the best solution (including network of gateways, fuzzy integrator, etc.). [24] . Figure 1 shows a modular neural network scheme:

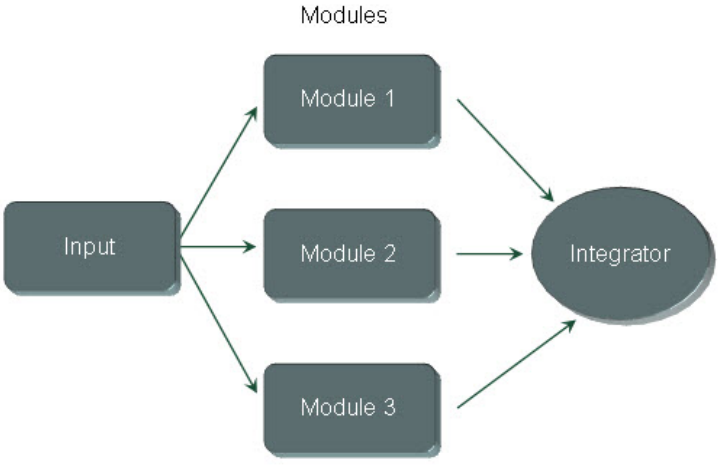


Fig. 1. Schematic of an modular artificial neural network

2.2 Historical Development

The first use of the iris was presented in Paris, where criminals were classified according to the color of their eyes following a proposal by the French ophthalmologist Bertillon (1880) [8]. Research in visual identification technology began in 1935. During that year an article appeared in the 'New York State Journal of Medicine', which suggested that "the pattern of arteries and veins of the retina could be used for unique identification of an individual" [2].

After researching and documenting the potential use of the iris as a tool to identify people, ophthalmologists Flom and Safir patented their idea in 1987 [11]; and later, in 1989, they patented algorithms developed with the mathematician Daugman. Thereafter, other authors developed similar approaches [2]. Later in 2001, Daugman also presented a new algorithm for the recognition of people using the biometric measurement of Iris [10].

The literature has well documented the uniqueness of visual identification. The iris is so unique that there are no two irises alike, even twins, in all humanity. The probability of two irises producing the same code is 1 in 10^{78}, becoming known that the earth's population is estimated at approximately 10^{10} million [1], it is almost imposible to ocurr.

Biometric identification techniques are very diverse, since any significant element of a person is potentially useful as an element of biometric identification. Even with the diversity of existing techniques, when developing a biometric identification system, this remains a totally independent of the technique. Human beings have many features in common, but also have characteristics that distinguish

them and make them unique from each other. Over the years there have been many studies and research about it and developing techniques, methods and systems that enable the use of these patterns for personal identification.

2.3 Iris Properties

The iris is an internal organ of the eye, located behind the cornea and the aqueous humor, which consists of a screening connective tissue, fibers, rings and colors that are a distinctive mark of the people to observe at close range (see Fig. 2). The iris texture is not a genetic expression and the morphogenesis is completely random [12].

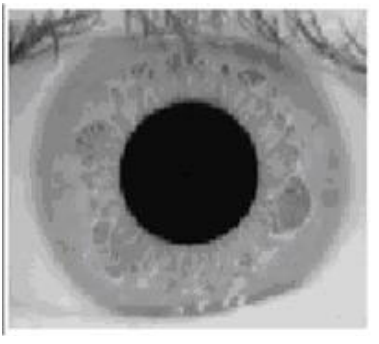


Fig. 2. Human Iris [13]

The properties of the iris that enhance its use for identification of individuals include: a) uniqueness in two individuals, b) inability to modify it without risk of vision loss, c) is a pattern with high randomness, and d) ease of record at close range. But it also presents some disadvantages such as: a) its small size makes it difficult to acquire it at certain distances, b) is a moving target, c) is located on a curved surface, moist and reflective, d), its image is often affected by eyelashes, eyelids and light reflections, and e) the deformations are not elastic when the pupil changes size [13].

3 Proposed Method and Problem Description

The proposed methodology for iris recognition can be stated as follows:

1) Search for a database of human iris.
2) Determine the database division in terms of individuals per module for the modular neural network.
3) Search for methods and / or pre-processing techniques for application to the image database and obtain an optimal identification.
4) Implementing in a programming language the different modular neural network architectures (see Fig. 3). The RNMs consist of 3 modules.
5) Find a modular integration method to provide the desired results.

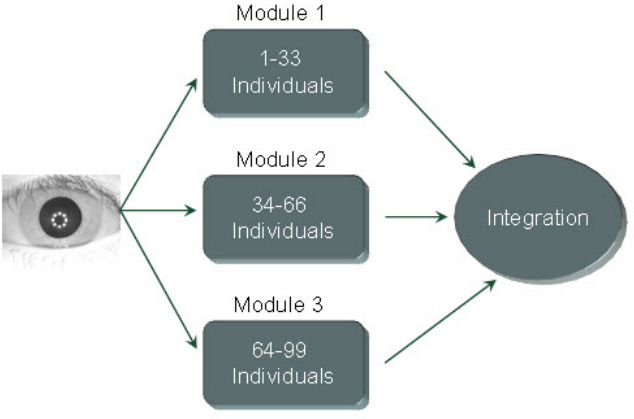


Fig. 3. Schematic representation of the division of problem and system operation

3.1 Problem Description

This work focuses primarily on the identification of individuals. This problem is well known by the scientific community, as innumerable investigations have been developed in this area, considering various measures to achieve it with biometrics (fingerprint, voice, palm of hand, signature) and various methods of identification (with particular emphasis on neural networks).

The specific problem considered in this work is: "Obtain a good percentage of person identification based on the biometric measurement of the human iris, using modular neural networks".

We used a database of human Iris from the Institute of Automation Chinese Academy of Sciences (CASIA) (see Fig. 4). It consists of 14 images (7 right eye - 7 left eye) per person, for a total of 99 individuals, giving a total of 1386 images. The image dimensions are 320 x 280, JPEG format [3].

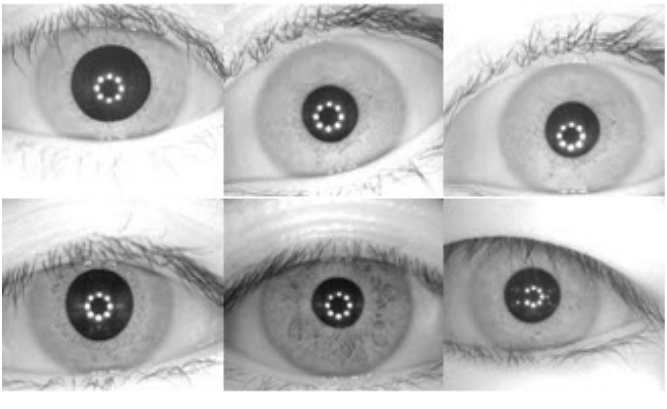


Fig. 4. Examples of the human iris images from CASIA database.

3.2 Image Pre-processing

The pre-processing that has been applied to the images before they are introduced to the neural network is as follows:

- Obtain the coordinates and radius of the iris and pupil using the method developed by Masek and Kovesi [9].
- Making the cut in the Iris.
- Resize the cut of the Iris to 21-21
- Convert images from vector to matrix
- Normalize the images.

1) Obtain coordinates of the center and radius of Iris: To get the coordinates of the center and radius of the iris and pupil of images in the CASIA database, the method developed by Masek and Kovesi was used [9].

This method involves applying a series of filters and mathematical calculations to achieve the desired gain.

First we apply edge detection with Canny's method (see Fig. 5 (a)), then the process continues using a gamma adjustment of the image (see Fig 5 (b)), to the resulting image obtained above a no maxima suppression is applied (see Fig. 5 (c)), and subsequently we applied to the image a threshold method (see Fig. 5 (d)).

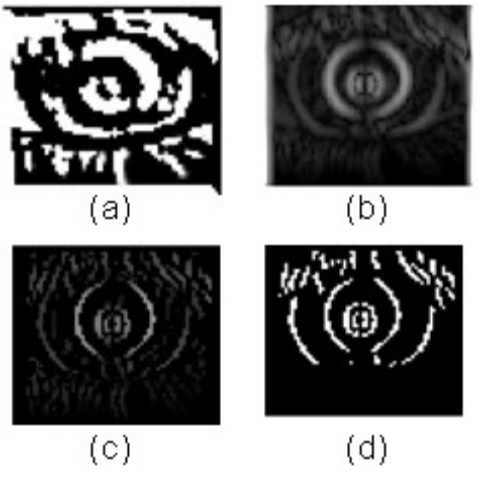


Fig. 5. (a) Edge detection with Canny's method (b) Image Adjust Gamma (c) No Maxima Suppression (d) Threshold.

Finally, we apply the Hough transform to find the maximum in the Hough space and, therefore, the circle parameters (row and column at the center of the iris and the radius).

2) Cut out the Iris: After obtaining the coordinates of the Iris, the upper right and lower left points are calculated to make the cut (see Fig. 6).

RowUpLeft = RowIris - RadiusIris;
RowLowRight = (RowIris+RadiusIris)-RowUpLeft;
ColUpLeft = ColumnIris - RadiusIris;
ColLowRight = (ColumnIris + RadiusIris) - ColUpLeft;

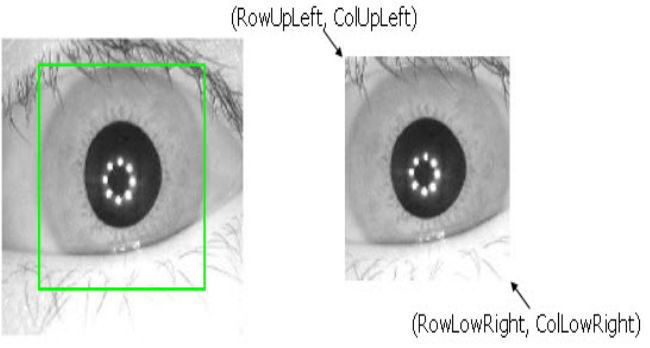


Fig. 6. Cut of iris, using the "imcrop" Matlab function.

3.3 Modular Neural Network Architecture

The work was focused on the recognition of persons using a modular neural network. We worked with 3 modules, each module input considers 33 individuals (264 images for training - 198 images for testing). We used the method of integration called Gating Network.

The architecture of the modular neural network was defined with the empirical expression of Salinas [7].

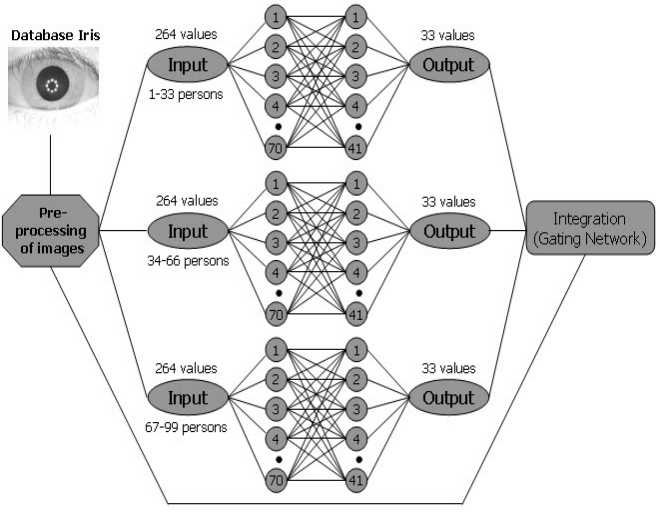


Fig. 7. Initial Modular Neural Network Architecture.

Based on the this expression we can calculate the number of nodes as follows:

- 1st hidden layer (2 _ (k + 2)) = 70.
- 2nd hidden layer (k + m) = 41.
- Output layer (k) = 33.

In determining the characteristics of the modular neural network, such as the number of modules to be used, the inputs, hidden layers, the number of neurons for each hidden layer and outputs, the initial architecture is specified in Fig. 7.

4 Simulation Results

Experiments were performed with the modular neural network architecture described in the previous section, and we perfomed experiments with 3 types of learning algorithms: gradient descent with adaptive learning (GDA), gradient descent with adaptive learning and momentum (GDX) and scaled conjugate gradient (SCG) . After obtaining the results of this architecture, two modified architectures were also tested to improve the accuracy.

4.1 Results with the Initial Modular Neural Network Architecture

The following results were achieved by each of the 3 modules in terms of percentage of identification.

Module 1
In this module the results were better with the scaled conjugate gradient algorithm (SCG), with an identification rate of 96.46% (191/198), goal error of 0.0000001, execution time of 29 seconds and 207 iterations (see Fig. 8 and Table I).

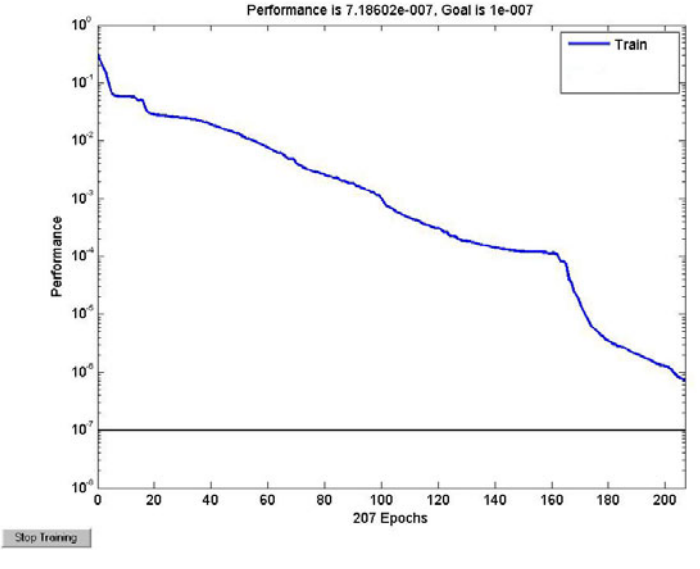


Fig. 8. Graph of error of the best training in Module 1.

Table I. Results for module 1

	Error	Time	Iterations	Rec.	Ident.	% Ident.
Traingda	0.000001	1.32 m.	1399	264/264	189/198	95.45
Traingdx	0.000001	38 s.	551	264/264	188/198	94.94
Trainscg	0.0000001	29 s.	207	264/264	191/198	96.46

Module 2

In this module the results were better with the gradient descent algorithm with adaptive learning (GDA), with an identification rate of 96.46% (191/198), goal error of 0.000001, execution time of 56 seconds and 833 iterations (see Fig. 9 and Table II) .

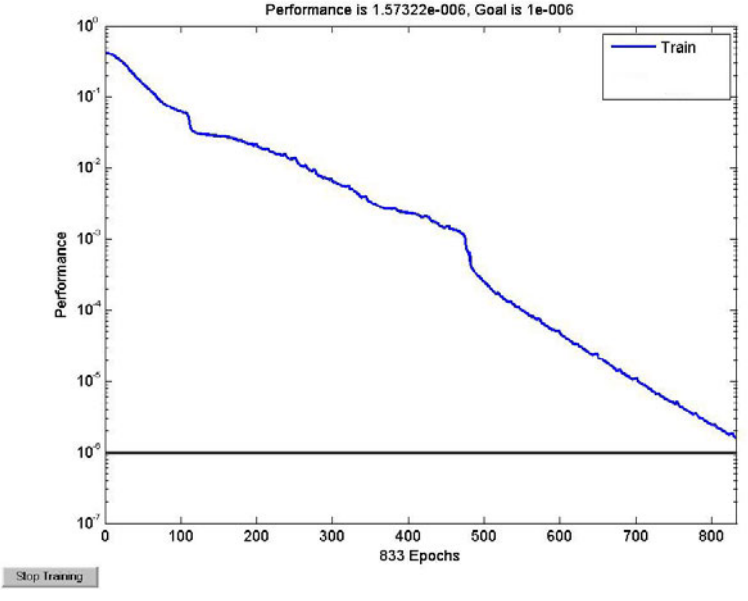


Fig. 9. Graph of error of the best training in Module 2.

Table II. Results for module 2

	Error	Time	Iterations	Rec.	Ident.	%Ident.
Traingda	0.000001	56 s.	833	264/264	191/198	96.46
Traingdx	0.000001	40 s.	586	264/264	189/198	95.45
Trainscg	0.0000001	28 s.	207	264/264	189/198	95.45

Module 3

In this module the results were better with the scaled conjugate gradient algorithm (SCG), with a 94.94% of identification rate (188/198), goal error of 0.0000001, execution time of 28 seconds and 144 iterations (see Fig. 10 and Table III).

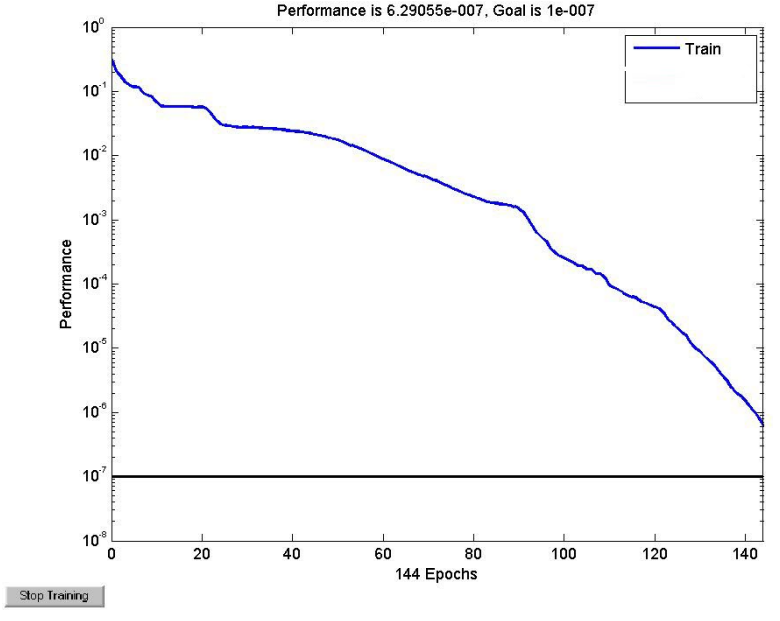


Fig. 10. Graph of error of the best training in Module 3.

Table III. Results for module 3

	Error	Time	Iterations	Rec.	Ident.	%Ident.
Traingda	0.000001	66 s.	992	264/264	187/198	94.44
Traingdx	0.000001	29 s.	416	264/264	186/198	93.93
Trainscg	0.0000001	20 s.	144	264/264	188/198	94.94

Integration

Analyzing the results obtained in the three modules, we see that in the first and third module, the learning algorithm that showed better results was the scaled conjugate gradient; and in the second module the best results were obtained with the gradient descent with learning adaptive. We decided to use the gating network integration method with the outputs of the three modules. Getting as final result a 95.95% of identification rate (570/594) (see Table IV).

Table IV. Results of the integration for the 3 modules

Integrator	MD1	MD2	MD3	Rec.	Ident.	% Ident.
Gating Network	Trainscg	Traingda	Trainscg	792/792	570/594	95.95

4.2 First Modification of the Modular Neural Network Architecture

In obtaining the results with the proposed modular neural network, we decided to modified manually the number of neurons in a hidden layer of the network, considering the second hidden layer for the modification.

There were two changes in the second hidden layer that produced good results: 70 neurons in first hidden layer and 67 neurons in the second hidden layer, and 70 neurons in first hidden layer and 100 neurons in the second hidden layer.

Module 1

The results of this module found that the best result was achieved with gradient descent algorithm with adaptive learning (GDA) with 70 hidden neurons in first layer and 67 neurons in second hidden layer with a 96.96% of identification rate (192/198) (see Table V).

Table V. Results for module 1

	70-41		70-67		70-100	
	Identif.	% Identif.	Identif.	% Identif.	Identif.	% Identif.
Traingda	189/198	95.45	192/198	96.96	190/198	95.95
Traingdx	188/198	94.94	189/198	95.45	191/198	96.46
Trainscg	191/198	96.46	189/198	95.45	190/198	95.95

Module 2

The results of this module found that the best result was achieved with gradient descent algorithm with adaptive learning (GDA) with 70 hidden neurons in first layer and 67 neurons in second hidden layer with a 97.97% of identification rate (194/198) (see Table VI).

Table VI. Results for module 2

	70-41		70-67		70-100	
	Identif.	% Identif.	Identif.	% Identif.	Identif.	% Identif.
Traingda	191/198	96.46	194/198	97.97	193/198	97.47
Traingdx	189/198	95.45	191/198	96.46	192/198	96.96
Trainscg	189/198	95.45	192/198	96.96	193/198	97.47

Module 3

The results of this module found that the best result was achieved with the scaled conjugate gradient algorithm (SCG) with 70 neurons in first hidden layer and 100 neurons in second hidden layer with a 95.45% of identification rate (189/198) (see Table VII).

Table VII. Results for module 3

	70-41		70-67		70-100	
	Identif.	% Identif.	Identif.	% Identif.	Identif.	% Identif.
Traingda	187/198	94.44	187/198	94.44	188/198	94.94
Traingdx	186/198	93.93	187/198	94.44	188/198	94.94
Trainscg	188/198	94.94	188/198	94.94	189/198	95.45

Knowing the best results for the 3 modules, it is now determined that the modular neural network architecture is modified as shown in Fig. 11:

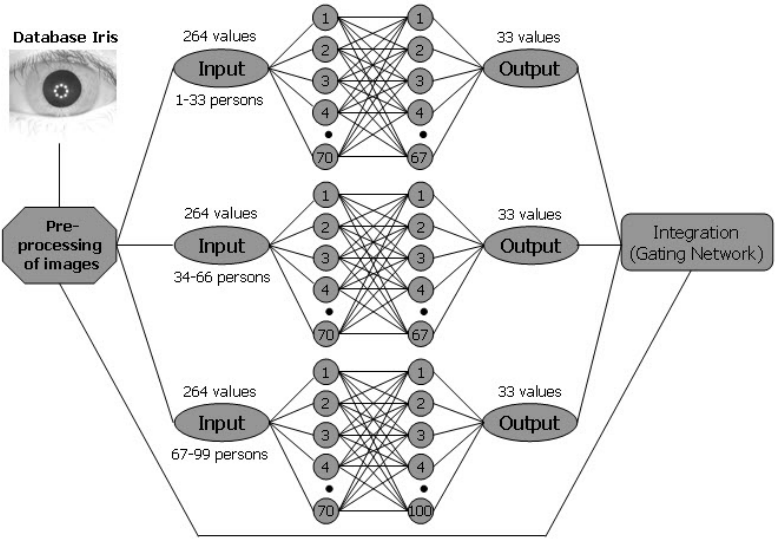


Fig. 11. Modified Modular Neural Network Architecture.

Integration

Analyzing the results obtained in the three modules, we see that in the first and second module, the learning algorithm that showed better results was the gradient descent with learning adaptive rate with 70 hidden neurons in first layer and 67 neurons in second hidden layer; and in the third module which showed best results was the scaled conjugate gradient with 7 hidden neurons in first layer and 10 neurons in second hidden layer. The gating network integration is done with the algorithms and numbers of neurons mentioned above for each module. The final result is 96.80% of identification rate (575/594) (see Table VIII).

Table VIII. Results of the integration for 3 modules with modified architecture

Integrator	MD1	MD2	MD3	Rec.	Ident.	% Ident.
Gating Network	Trainscg	Traingda	Trainscg	792/792	570/594	95.95
Gating Network Modified Architecture	Traingda	Traingda	Trainscg	792/792	575/594	96.80

Cross Validation 10 Times

After obtaining the final results of the integration we chose to carry out cross-validation of 10 different combinations of training and test images, and averaging the results of identifications (see Table IX):

Table IX. Results for cross validation for 10 times

Validator	MD1	MD2	MD3	% Ident.
Cross Validation 10 Times	91.57	92.93	90.25	91.58

4.3 Second Modification of the Modular Neural Network Architecture (Extended)

In analyzing the results obtained with the integration in the modular neural network we found that the learning algorithm (SCG) identifies the image of human iris of a particular person that other different learning algorithm (GDA) do not identified, as well as the number of neurons in the hidden layers of the neural network .

Based on the above ideas, we propose an extended architecture of the modular neural network. In this architecture each module consists of two modules with different learning algorithms or different number of neurons in the second hidden layer (the two modules were more successful results with the modified architecture), see in Fig 12.:

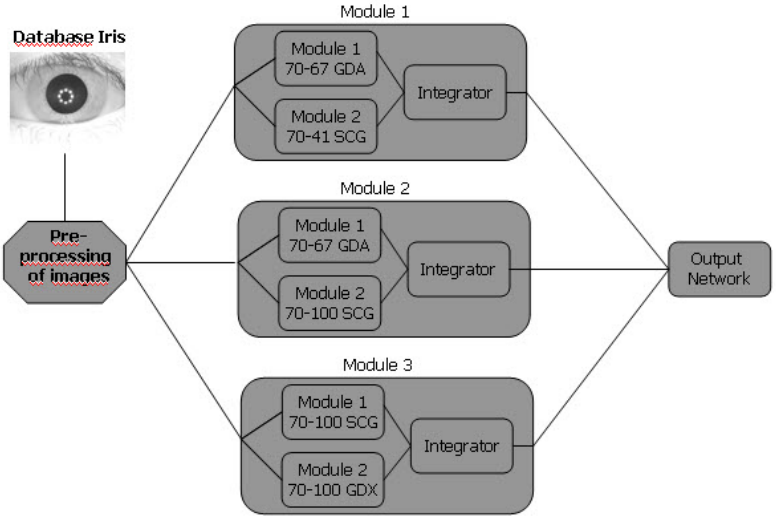


Fig. 12. Extended Modular Neural Network Architecture.

With the outputs of each module we perform the results integration. Obtaining as final result a 97.13% identification rate (577 images of 594 test images) with the winner takes all integrator and 96.96% identification rate (576 images of 594 test images) with the Mamdani type fuzzy integrator (see Table X).

Table X. Results of the integration for 3 modules with extended architecture

Integrator	MD1	MD2	MD3	Rec.	Ident.	% Ident.
Gating Network Modified Architecture	Traingda 192	Traingda 194	Trainscg 189	792/792	575/594	96.80
Integrator: Winner Takes All	194	193	190	792/792	577/594	97.13
Integrator: Fuzzy Type Mamdani	193	194	189	792/792	576/594	96.96

Cross Validation 10 Times

After obtaining the final results of the integration we chose to carry out cross-validation of 10 different combinations of training and test images, and averaging the results of identifications (see Table XI):

Table XI. Results for cross validation for 10 times with extended architecture

Cross Validation 10 Times	% Ident.
Modified Architecture	91.58
Extended Architecture with Integrator Winner Takes All	91.83
Extended Architecture with Integrator Fuzzy Type Mamdani	91.41

5 Conclusions

In this paper we presented modular neural network architectures, which have as input the database of human iris images, with the database images divided in three parts for the three modules, and each module has two hidden layers. In this work, several methods were used to make the elimination of noise that the original pictures had until the coordinates of the center and radius were obtained, and then make a cut around the iris.

With the extended architecture we achieved higher results than the initial and modified by achieving a 97.13% identification rate (577 images of 594 test images) with the integrator winner takes all and 96.96% identification rate(576 images of 594 test images) with the Mamdani type fuzzy integrator.

These results demonstrate that the use of the human iris biometric measurement worked with modular artificial neural networks and favorable results of person identification were obtained.

Future work consists in considering other different pre-processing of images or that complements the current one and using a method for optimization of the

modular artificial neural network, with which we could get beyond the rate of detection obtained in the results shown in this paper.

References

[1] Sánchez, O., González, J.: Access Control Based on Iris Recognition, Technological University Corporation of Bolívar, Faculty of Electrical Engineering, Electronics and Mechatronics, Cartagena of Indias, Monography, pp. 1–137 (November 2003)

[2] López, J., González, J.: State of the Art: Automatic Recognition of Human Iris, Politécnico Colombiano, and National University of Colombia, Scientia et Technica Año XI, No 29, December de, pp. 77–81 (2005)

[3] Database of Human Iris. Institute of Automation of Chinese Academy of Sciences (CASIA). Found on the Web page, http://www.cbsr.ia.ac.cn/english/IrisDatabase.asp

[4] Jang, J., Sun, C., Mizutani, E.: Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prentice Hall, Libro (1996)

[5] Miyazawa, K., Ito, K., Aoki, T., Kobayashi, K., Nakajima, H.: An Effective Approach for Iris Recognition Using Phase-Based Image Matching. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(10), 1741–1756 (2008)

[6] MatLab, the language used for implementation and network design. Version 7.3.0.267 (R2006b), August 03 (2006)

[7] Salinas, R.: Neural Network Architecture Parametric Face Recognition, Paper, University of Santiago of Chile, pp. 5–9 (2000), http://cabierta.uchile.cl/revista/17/articulos/paper4/index.html

[8] Tisse, C., Martin, L., Torres, L., Robert, M.: Person identification technique using human iris recognition., Universite de Montepellier (2000)

[9] Masek, L., Kovesi, P.: MATLAB Source Code for a Biometric Identification System Based on Iris Patterns. The School of Computer Science and Software Engineering, The University of Western Australia (2003)

[10] Daugman, J.: Statistical Richness of Visual Phase Information: Update on Recognizing Persons by Iris Patterns. International Journal of Computer Vision 45(1), 25–38 (2001)

[11] Khaw, P.: Iris recognition technology for improved authentication. Sala de Lectura de Seguridad de la Información, SANS Institute, pp. 1–17 (2002)

[12] Muron, A., Pospisil, J.: The human iris structure and its usages, Czech Republic, Physica, pp. 89–95 (2000)

[13] Ma, L., Wang y Tan, T.: Iris recognition based on multichannel Gabor filtering. In: ACCV2002. 5th Asian Conference on Computer Vision, Melbourne, Australia, Enero 23-25, vol. 1, pp. 279–283 (2002)

[14] Moron, Q.: Modular Systems, Experts Mixed and Hybrid Systems., Technical Report DI-2000-001, Department of Computer University of Valladolid, Spain (2000)

[15] Gómez, A.: A fast and robust approach for iris segmentation; Artículo, II Simposio Peruano de Computación Gráfica y Procesamiento de Imágenes, scgi-2008, pp. 1–10 (2008)

[16] Daugman, J.: How Iris Recognition Works. IEEE Transactions on Circuits And Systems For Video Technology 14(1), 21–30 (2004)

[17] Daugman, J., Downing, C.: Epigenetic Randomness, Complexity and Singularity of human Iris Patterns. The Royal Society Proceedings in Biological Sciences 268(1477), 1737–1740 (2001)

[18] Kasar, A., Tapamo, J.R.: Texture Detection For Segmentation of Iris Images. In: ACM Proceedings of the 2005 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries, vol. 150, pp. 236–243 (2005)

[19] Daugman, J.: The Importance of Being Random: Statistical Principles of Iris Recognition. The Journal of Pattern Recognition Society 36(2), 279–291 (2003)

[20] He, Z., Tan, T., Sun, Z., Qiu, X.: Towards Accurate and Fast Iris Segmentation for Iris Biometrics. IEEE Transactions on Pattern Analysis and Machine Intelligence (July 15, 2008)

[21] Tieniu, L., Wang, Y., Zhang, D.: Personal Identification Based on Iris Texture Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(12), 1519–1533 (2003)

[22] Salazar, P.: Biometric recognition using techniques of hand geometry and voice with computer vision for feature extraction, neural networks and fuzzy logic, Master thesis, Division of Graduate Studies and Research in Computer Science, ITT, p. 57 (2008)

[23] Wong, A., Shi, P.: Peg-Free Hand Geometry Recognition Using Hierarchical Geometry and Shape Matching. In: Artículo, Workshop on Machine Vision Applications (MAV), Nara, Japan (2002)

[24] Morcego, B.: Study of modular neural networks for modeling nonlinear dynamic systems. PhD thesis, Universitat Politecnica de Catalunya, Barcelona, Spain (2000)

Real Time Face Identification Using a Neural Network Approach

Juan Carlos Vázquez, Miguel López, and Patricia Melin

Tijuana Institute of Technology, Tijuana México
juancarlos_jcs27@hotmail.com, malopezr60@hotmail.com.mx, pmelin@tectijuana.mx

Abstract. This paper presents techniques for developing a system for identification of faces in real time based on biometric technology [22], where the identification phase is being implemented by an artificial neural network. The motivation for this research work stems from the observation that the human face provides a particularly interesting structure. Face images are obtained by a web camera and then used for the digital image preprocessing techniques. Feature extraction techniques are applied; the extracted image features are fed to the neural network for learning. Due to the fact that the effectiveness of systems for identification techniques rely primarily on Preprocessing and feature extraction, therefore, this work presents different features extraction techniques, and a comparison between methods is made, in terms of their percentages of recognition. We described the most used techniques for this task [10], i.e.: Edge extraction, Wavelet Analysis, eigenfaces.

1 Introduction

Face recognition has been studied more thoroughly in recent years. Thanks to advances in computing power, allowing the implementation of more complex algorithms [7]. This work uses a technique based on neural networks.

This allows multiple applications where today it is necessary to identify particular individuals [11]. For example: lock or unlock the computer, mobile phones, and security systems for laboratories, banks and shops.

Access control by face identification has advantages over other biometric systems: The cost of equipment is not very high, can be constructed using a simple video camera and a personal computer, no need to touch anything, say anything or bend the eye to a detector, any person can only stand in front of the camera, and the identification system performs the task [1].

This paper is organized as follows: in section 2 we describe the basic concepts of our work, section 3 describes the methods proposed for the development of our work and a brief description of the problem, section 4 describes the features extraction methods and their application to our problem, section 5 describes the

P. Melin et al. (Eds.): Soft Comp. for Recogn. Based on Biometrics, SCI 312, pp. 155–169.
springerlink.com

architecture of the modular neural network we used for our training, in section 6 we show the results we obtained by applying the techniques of feature extraction and in section 7 the conclusions of our work are presented.

2 Background and Basic Concepts

In this section we describe the basic concepts of an Artificial Neuronal Networks and the Backpropagation Algorithm also describes background information on face recognition.

2.1.1 Artificial Neural Networks (ANNs)

The ANN provides a general practical method for real-valued, discrete-valued, and vector-valued functions from examples. The back propagation algorithm which is widely used in ANNs, uses gradient descent to tune network parameters to best fit a training set of input-output pairs.

An artificial neural network (ANN) is a distributed computing scheme based on the structure of the nervous system of humans. The architecture of a neural network is formed by connecting multiple elementary processors, this being an adaptive system that has an algorithm to adjust their weights (free parameters) to achieve the performance requirements of the problem based on representative samples [18].

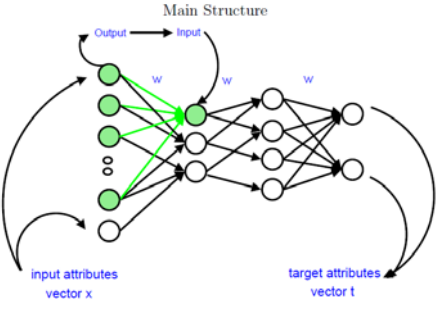


Fig. 1. Main structure

2.1.2 Backpropagation Algorithm

The learning problem faced by back propagation is to search a large problem space defined by all possible weight values for all the units in the network. The algorithm is:

backpropagation$((\sim x,\sim t), \eta, n_{in}, n_{out}, n_{hidden})$

$(\sim x,\sim t)$ for each instance from training examples, where $\sim x$ is the vector of network input values, and $\sim t$ is the vector of target network output values.

η the learning rate.

n_{in}, n_{out}, n_{hidden}

The input from unit i into unit j is denoted xji, and the weight from unit i to unit j is denoted wji.

- Create an ANN with *n*in input, *n*out output, and *n*hidden hidden units.
- Initialize all network weights to small random numbers (e.g. [−0.5, 0.5]).
- Until the termination conditions is met, Do:

  For each (~x,~t) in the training examples, Do:

  1. Input the instance ~x to the network and compute the output ou of every unit u inthe network.
  2. For each network output unit k, calculate its error terms σk

$$\sigma k = ok(1 - ok)(tk - ok) \quad (1)$$

  3. For each hidden unit k, calculate its error term σh

$$\sigma h = oh(1 - oh) \sum k \in \text{outputs of } h \ (wkh \quad \sigma k) \quad (2)$$

  4. Update each network weight wji

$$wji = wji + \quad wji$$

  where

$$wji = \eta \sigma j x ji$$

The error term for hidden unit h is calculated by summing the error terms σk for each output unit influenced by unit h, weighting each of the σk's by wkh, the weight from hidden unit h to output unit k. This weight characterizes the degree to which hidden unit h is "responsible for" the error in output unit k [23]. See figure 1.

2.2 Historical Development

Automated face recognition is a relatively new concept. Developed in the 1960s, the first semi-automated system for face recognition required the administrator to locate features (such as eyes, ears, nose, and mouth) on the photographs before it calculated distances and ratios to a common reference point, which were then compared to reference data. In the 1970s, Goldstein, Harmon, and Lesk used 21 specific subjective markers such as hair color and lip thickness to automate the recognition. The problem with both of these early solutions was that the measurements and locations were manually computed. In 1988, Kirby and Sirovich applied principle component analysis, a standard linear algebra technique, to the face recognition problem.

This was considered somewhat of a milestone as it showed that less than one hundred values were required to accurately code a suitably aligned and normalized face image. In 1991, Turk and Pentland discovered that while using the eigenfaces techniques, the residual error could be used to detect faces in images – a discovery that enabled reliable real-time automated face recognition systems. Although the approach was somewhat constrained by environmental factors, it nonetheless created significant interest in furthering development of automated face recognition technologies.

The technology first captured the public's attention from the media reaction to a trial implementation at the January 2001 Super Bowl, which captured surveillance images and compared them to a database of digital mugshots. This demonstration

initiated much-needed analysis on how to use the technology to support national needs while being considerate of the public's social and privacy concerns.

Today, face recognition technology is being used to combat passport fraud, support law enforcement, identify missing children, and minimize benefit/identity fraud [20].

3 Proposed Method and Problem Description

The proposed methodology can be stated as follows:

- Obtain the database of face images.
  - Preprocessing.
  - Feature Extraction.
- Develop the neural network for training images from the database.
- Image Acquisition webcam.
  - Preprocessing.
  - Feature Extraction.
- Simulate the trained network images were obtained, for images with identical patterns for identification

The system basically consists of two stages (Figure 2):

a) Training stage.
b) Identification stage.

3.1 Image Acquisition (Database)

The image acquisition to obtain the database was performed as follows:

- Taken by students in the Master of Science in Computer Science (25 individuals), through a web camera.

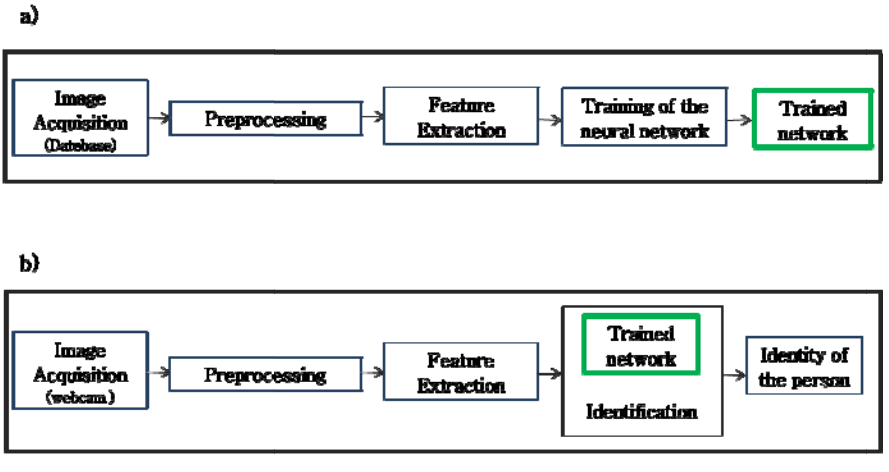


Fig. 2. System Block Diagram.

Fig. 3. Images of individuals.

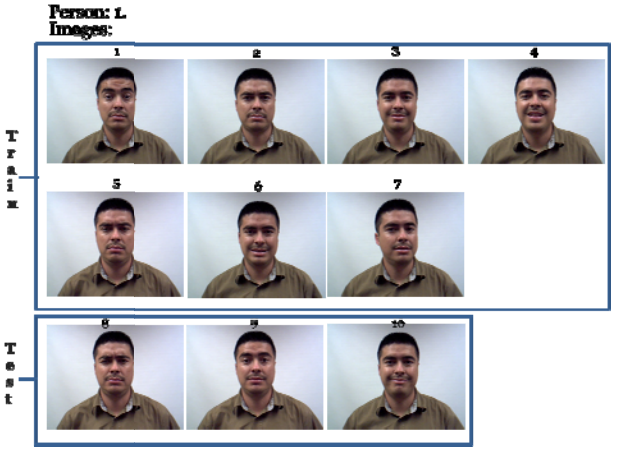


Fig. 4. Images for train and test.

- Contains a set of images of individuals at RGB, with a resolution of 640x480 pixels, JPG format, see figure 3.
- It contains different images for each person of 25, which 7 are for training and 3 to identify, see figure 4.

3.2 Preprocessing

If we use the images that were obtained, in most cases results undesirable, this is because during the acquisition process can affect the image changes [21].

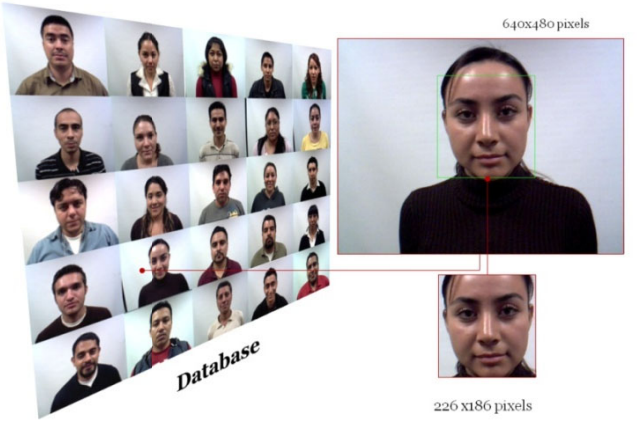


Fig. 5. face detection process.

Techniques can be applied to improve restore, segment, rotate, enlarge, and reduce extract image areas, among other [8].

The preprocessing steps are:

1. Detecting face in each of the images in the database to obtain only the *facial image* [12] [25] (figure 6).
2. In the *facial image* is obtained the threshold (figure 6-b).
3. The threshold is removed noise (figure 6-c).

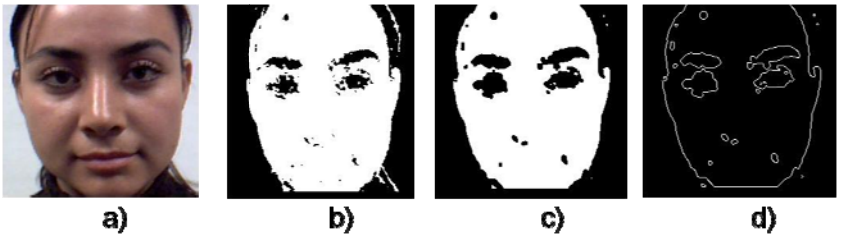


Fig. 6. a) facial image, b) threshold, c) removed noise, d) edges.

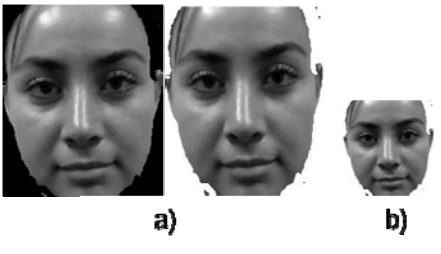


Fig. 7. a) information obtained, b) image resized.

4. Edges are obtained (figure 6-d).
5. On the edges we get the information we want (figure 7-a).
6. Finally resize the image to 50% (110x110 pixels), see figure 7-b.

4 Feature Extraction

The main objective of this stage is to extract the most discriminating components of a face, eliminating the information that is irrelevant to the recognition [3] [13]. In feature extraction we use Edges, Eigenface, and Wavelet methods.

4.1 Edge Extraction

Provide valuable information about the boundaries of objects and can be used to segment the image, recognize objects, etc [14] [24].

a) b)

Fig. 8. Edge extraction, a) train, b) test.

Once the pre-processed database, we obtain the extraction of edges using Canny is method for each individual (figure 8).

4.2 Eigenface

It is considered the first successful example of facial recognition technology [9] [6].

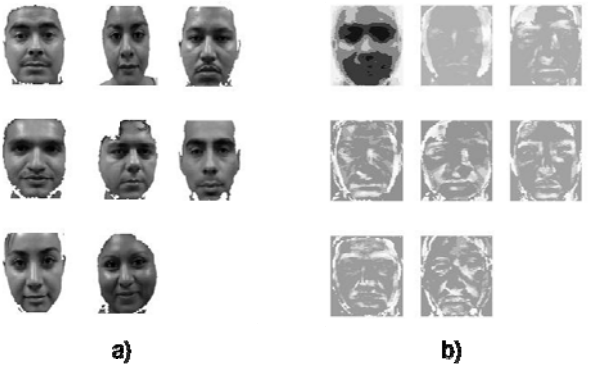


Fig. 9. a) individuals b) eigenface.

Based on the pre-processed database, we obtain the eigenface for each individual (figure 9).

4.3 Discrete Wavelet Transform (dwt)

Provides an array of 4 types of coefficients: approximation, horizontal details, vertical details and diagonal details. The *approximation* contains the most important information [4] [16].

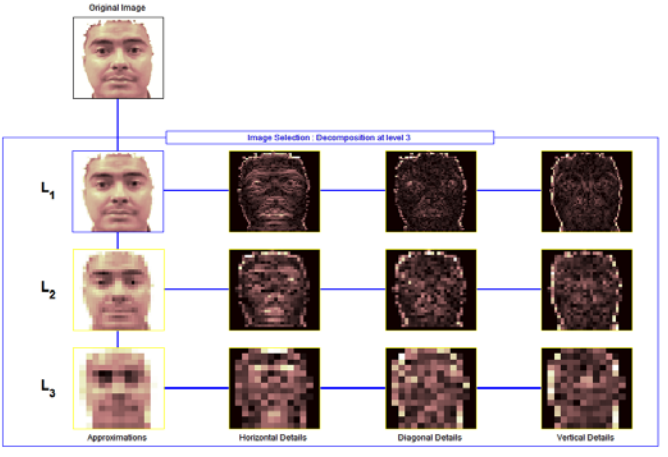


Fig. 10. L_1) Level 1, L_2) Level 2, L_3) Level 3.

Based on the pre-processed database, we obtain the approximation of wavelets at level 1, level 2 and level 3 for each individual (figure 10).

5 Modular Neural Network Architecture

The work was focused on the recognition of persons using a modular neural network [5] [2]. The image is divided into three modules (figure 11), each module input contains front, eyes and mouth respectively, which 7 images for training were used, for each of the 25 individuals, for a total of 175 images and 3 images to testing, for each of the 25 individuals, for a total of 75 images. We used the method of integration called *The winner takes it all*.

The architecture of the modules in the modular neural network was defined with the empirical expression of Salinas [19].

Based on this expression we can calculate the number of nodes as follows:

- 1st hidden layer (2 * (k + 2)) = 54.
- 2nd hidden layer (k + m) = 32.
- Output layer (k) = 25.

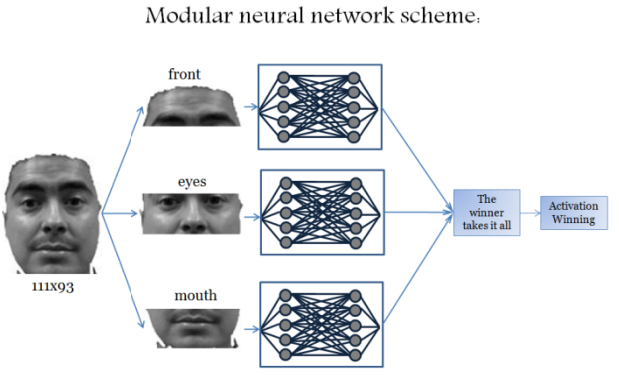


Fig. 11. Modular neural network scheme.

In determining the characteristics of the modular neural network, such as the number of modules to be used, the inputs, hidden layers, the number of neurons for each hidden layer and outputs.

6 Simulation Results

Experiments were performed with the modular neural network architecture described in the previous section, and we performed experiments with 2 types of

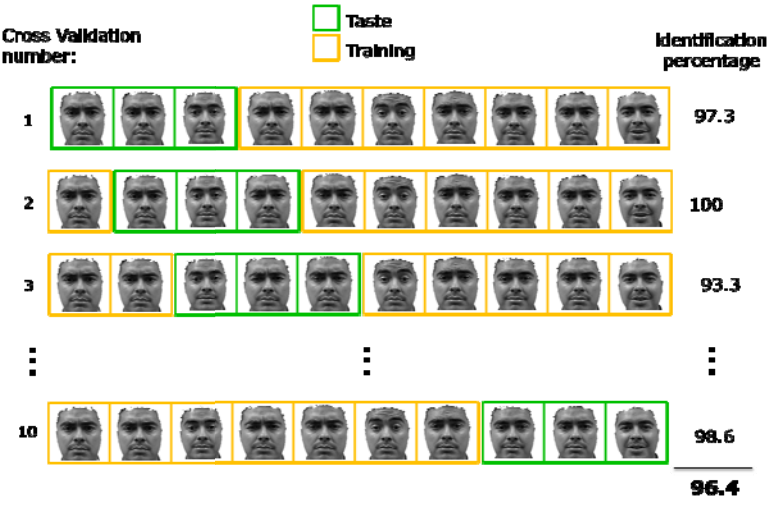
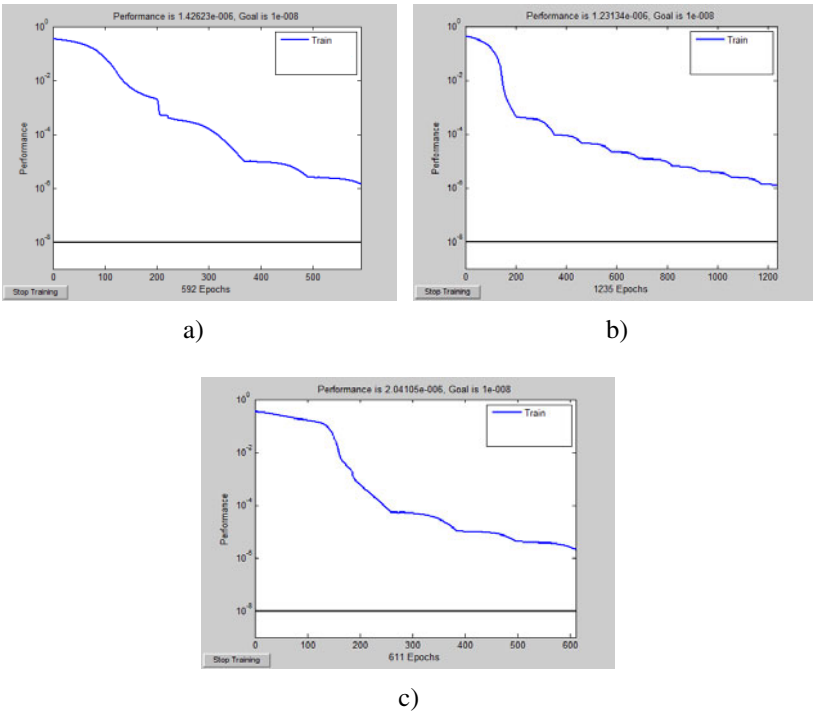


Fig. 12. Cross validation for database pre-processed with the learning algorithm SCG.

learning algorithms: gradient descent with adaptive learning and momentum (GDX) [17] and scaled conjugate gradient (SCG) [15]. All were trained with the database and cross validation (figure 12) was performed with 10 images per person, which were 7 for training and 3 for testing.

6.1 Results Using Only the Pre-processed Database

The following graphs (figure 13) show the error convergence of each module of GDX learning algorithm:

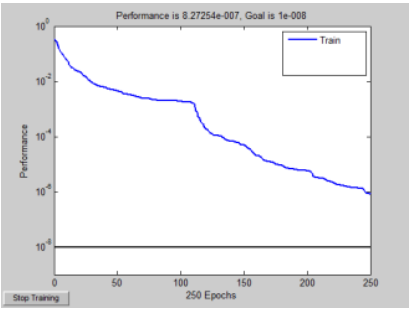


a) b)

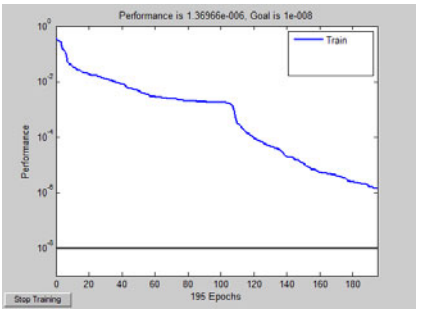
c)

Fig. 13. a) Error convergence with GDX for front, b) Error convergence with GDX for eyes, c) Error convergence with GDX for mouth.

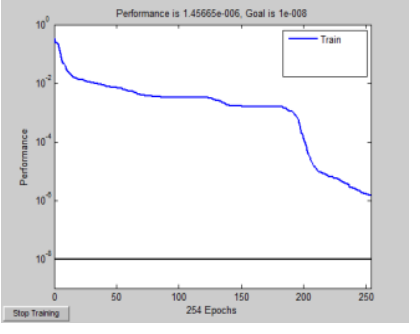
The following graphs (figure 14) show the error convergence of each module of SCG learning algorithm:



a)



b)



c)

Fig. 14. a) Error convergence with SCG for front b) Error convergence with SCG for eyes c) Error convergence with SCG for mouth.

In table 1 we show the results from tests using only the pre-processed database.

Table 1. Results of the pre-processed database.

Training	Method	Neurons	Epoch	Error	Training Duration	Total Identification
1	1 traingdx	54, 32	2000	0.00000001	01:07:45	73.3/ 75 97.7333%
	2 traingdx	54, 32				
	3 traingdx	54, 32				
2	1 trainscg	54, 32	2000	0.00000001	00:47:33	72.3/ 75 96.4%
	2 trainscg	54, 32				
	3 trainscg	54, 32				

6.2 Results Using the Pre-processed Database with Edges

In table 2 we show the results from tests using the pre-processed database with edges.

Table 2. Results of the database with edges.

Training	Method	Neurons	Epoch	Error	Training Duration	Total Identification
1	1 traingdx	54, 32	2000	0.00000001	01:16:24	59.8/ 75 79.7333%
	2 traingdx	54, 32				
	3 traingdx	54, 32				
2	1 trainscg	54, 32	2000	0.00000001	01:30:26	57.8/ 75 77.0667%
	2 trainscg	54, 32				
	3 trainscg	54, 32				

6.3 Results Using the Pre-processed Database with Eigenface

In table 3 we show the results from tests using the pre-processed database with eigenface.

Table 3. Results of the database with eigenface.

Training	Method	Neurons	Epoch	Error	Training Duration	Total Identification
1	1 traingdx	54, 32	2000	0.00000001	02:02:32	73/ 75 97.3333%
	2 traingdx	54, 32				
	3 traingdx	54, 32				
2	1 trainscg	54, 32	2000	0.00000001	00:40:12	72.2/ 75 96.2667%
	2 trainscg	54, 32				
	3 trainscg	54, 32				

6.2.1 Results with the Pre-processed Database with Wavelet

In table 4 we show the results from tests at level 1.

Table 4. Results of the database with wavelet level 1.

Training	Method	Neurons	Epoch	Error	Training Duration	Total Identification
1	1 traingdx	54, 32	2000	0.00000001	00:12:19	73.7/ 75 98.2667%
	2 traingdx	54, 32				
	3 traingdx	54, 32				
2	1 trainscg	54, 32	2000	0.00000001	00:09:37	73.1/ 75 97.4667%
	2 trainscg	54, 32				
	3 trainscg	54, 32				

6.2.2 Results with the Pre-processed Database with Wavelet

In table 5 we show the results from tests at level 2:

Table 5. Results of the database with wavelet level 2.

Training	Method	Neurons	Epoch	Error	Training Duration	Total Identification
1	1 traingdx	54, 32	2000	0.00000001	00:13:48	73.7/ 75 98.2667%
	2 traingdx	54, 32				
	3 traingdx	54, 32				
2	1 trainscg	54, 32	2000	0.00000001	00:04:09	73.3/ 75 97.7333%
	2 trainscg	54, 32				
	3 trainscg	54, 32				

6.2.3 Results with the Pre-processed Database with Wavelet

In table 6 we show the results from tests at level 3:

Table 6. Results of the database with wavelet level 3.

Training	Method	Neurons	Epoch	Error	Training Duration	Total Identification
1	1 traingdx	54, 32	2000	0.00000001	00:05:19	72.7/ 75 96.9333%
	2 traingdx	54, 32				
	3 traingdx	54, 32				
2	1 trainscg	54, 32	2000	0.00000001	00:03:25	72.8/ 75 97.0667%
	2 trainscg	54, 32				
	3 trainscg	54, 32				

7 Conclusions

In this paper we presented modular neural network architectures, which have as input the database of faces images, with the images divided in three parts for the three modules (front, eyes and mouth respectively), and each module has two hidden layers. In this work, several methods of computer vision were used to make the preprocessing of the original photographs, to obtain only the face and then apply the feature extraction.

The neural network was trained with the pre-processed databases, edges, wavelet and eigenface, with the integrator the winner takes all. The best identification was obtained by using wavelets at level 1 and level 2, with the highest percentage of *98.2667%.*

The next stage is to work with images obtained from a web camera online, and make a comparison with the tables of results.

References

[1] Abu, S., Rabiul, I., Kaushik, R.: Real Time Face recognition Using Artifial Neural Networks: A Secure Personnel Identification System, Department of Computer Science and Engineering Rajshahi University of Engineering and Technology, Rajshahi-6204, Bangladesh

[2] Azam, F.: Biologically Inspired Modular Neural Networks., Dissertation submitted to the Faculty of the Virginia Polytechnic Institute and State University (May 2000)

[3] Campadelli, P., Lanzarotti, R., Lipori, G.: Automatic Facial Feature Extraction for Face Recognition., Università degli Studi di Milano Italy

[4] Chao-Chun, L., Dao-Qing, D.: Discriminant Wavelet Packet Coordinates for Face Recognition. Center for Computer Vision and Department of Mathematics, Sun Yat-Sen (Zhongshan) University, Guangzhou, 510275 China

[5] Curran, K., Xuelong: The Use of Neural Networks in Real-time Face Detection", Department of Computer Science and Engineering Rajshahi University of Engineering and Technology, Rajshahi-6204, Bangladesh. Journal of Computer Sciences 1(1), 47–62 (2005)

[6] Eleyan, A., Demirel, H.: PCA and LDA based Neural Networks for Human Face Recognition. Eastern Mediterranean University,Northern Cyprus

[7] Gonzalez, R., Woods, R.: Digital Image Processing. Addison-Wesley, Reading (2002)

[8] Gregory, D., Kentaro, T.: X Vision: A Portable Substrate for Real-Time Vision Applications., Department of Computer Science, Yale University, P.O. Box 208285, New Haven, Connecticut 06520

[9] Heseltine, T.: Evaluation of Image Pre-processing Techniques for Eigenface Based Face Recognition. The University of York, Department of Computer Science

[10] Hyeon, B., Sungshin, K.: Real-time face detection and recognition using hybrid-information extracted from face space and facial features. School of Electrical and Computer Engineering, Pusan National University, Jangjeon-dong, Geumjeong-gu, Busan 609-735, South Korea

[11] Jain, A., Bolle, R., Pankanti, S. (eds.): Biometrics Personal Identification in Networked Society. Kluwer Academic Press, Boston (1999)

[12] Lamiaa, M., Sherif, A.: Face Detection Based on Skin Color Using Neural Networks. Information Technology Institute, Giza, Egypt
[13] Lu, X., Jain, A.: Automatic Feature Extraction for Multiview 3D Face Recognition. Dept. of Computer Science & Engineering Michigan State University East Lansing, MI 48824
[14] Mahinda, P., Özdemir, G.: Edge-end Pixel Extraction for Edge-based Image Segmentation
[15] Møller, F.: A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. Computer Science Department, University of Aarhus, Denmark (November 13, 1990)
[16] Øyvind, R.: Applications of the wavelet transform in image processing., Department of informatics, University of Oslo (November 12, 2004)
[17] Plagianakos, Sotiropoulos, Vrahatis: An Improved Backpropagation Method with Adaptive Learning Rate", University of Patras Department of Mathematics, Division of Computational Mathematics & Informatics
[18] Salinas, R.: "Neural Network Architecture Parametric Face Recognition, Paper, University of Santiago of Chile, 5–9 (2000), http://cabierta.uchile.cl/revista/17/articulos/paper4/index.html
[19] Salinas, R., Larraguibel, L.: Red Neuronal de Arquitectura paramétrica en Reconocimiento de Rostros, Departamento de Ingeniería Eléctrica, Universidad de Santiago de Chile, Facultad de Ciencias de la Ingeniería, Universidad Diego Portales
[20] Smith, K., Ross, A., Colbry, D.: Standars ICP Theam. Face recognition, National Science and Technology Council (NSTC) Committee on Technology committee on Homeland and National Security Subcommittee on Biometrics, August 7 (2006)
[21] Svoboda, T.: Image preprocessing in spatial domain Sharpening, image derivatives, Laplacian, edges. Czech Technical University, Faculty of Electrical Engineering Center for Machine Perception, Prague, Czech Republic, May 25 (2007)
[22] Wedekind, J.: Real-time Computer Vision With Ruby., O'Reilly Open Source Convention 2008, Wednesday, July 23 (2008)
[23] Weiyu, Y.: Artificial Neural Networks, October 7 (2005)
[24] Yang, Gillies: Image Processing and Edge Detection. Computer Vision Department of Computer. Imperial College, London
[25] Yu, Q., Chen, H., Chen, W., Xiaodong, Z.: Ch OpenCV for interactive open architecture computer vision, Integration Engineering Laboratory, Department of Mechanical and Aeronautical Engineering, University of California, One Shields Avenue, Davis, CA 95616, USA SoftIntegration, Inc., 216 F Street, 68 Davis, CA 95616

Comparative Study of Feature Extraction Methods of Fuzzy Logic Type 1 and Type-2 for Pattern Recognition System Based on the Mean Pixels

Miguel Lopez, Patricia Melin, and Oscar Castillo

Computer Science in the Graduate Division Tijuana, Institute of Technology Tijuana, B. C., Mexico,mlopez@tectujuana.mx, pmelin@tectijuana.mx, ocastillo@tectijuana.mx

Abstract. We describe in this paper a new approach for features extraction methods with Type-1 and Type-2 for Pattern Recognition System based on the pixels mean. In this paper we consider pattern recognition with extraction features fuzzy logic for ensemble neural networks for the case of fingerprintsn and using response integration fuzzy logic method to the test proposed method of fuzzy extraction method. An ensemble neural network of three modules is used. Each module is a local expert on person recognition based on their biometric measure (Pattern recognition for fingerprints). The fuzzy extraction features method is based on the pixels mean of the fingerprint.

1 Introduction

The emerging field of biometry technics technology is used as a measure unique physical, biological, obtained by an electronic device to establish the identification of a person. In practice the use of fingerprints as a means of identification is an indispensable aid to strengthen security measures, Every person has minutiae raised ridges of skin on the inside surfaces of their fingers, which display a number of characteristics known as minutiae, shown in the Fig. 1.

The minutiae do not change naturally during a person's life. Through the history of fingerprinting, no two fingerprints have ever been found to match exactly [1]. Fingerprint recognition systems usually include a sensor to take fingerprints and software for fingerprint analysis and recognition.

The most common techniques that are available to determine the unique identity of the person are fingerprint, voice, face and iris recognition [37]. Of these techniques, fingerprint and iris offer a very high level of certainty as to a person's identity, while the others are less accurate. The four primary methods of biometric authentication in widespread use today are face, voice, fingerprint, and iris

P. Melin et al. (Eds.): Soft Comp. for Recogn. Based on Biometrics, SCI 312, pp. 171–188.
springerlink.com

Fig. 1. Fingerprints minutiae.

recognition. In this paper, we consider pattern recognition with preprocessing fuzzy extraction features method and ensemble neural networks for the case of fingerprints.

Fingerprint Recognition. The process of authenticating people based on their fingerprints can be divided into three tasks. First, you must collect an image of a fingerprint, second, you must determine the key elements of the fingerprint for confirmation of identity, and third, the set of identified features must be compared with a previously-enrolled set for authentication. The system should be never expected to see a complete1:1 match between these two sets of data. In general, you could expect to couple any collection device with any algorithm, although in practice most vendors offer proprietary, linked solutions.

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

In the experiments performed in this research work, we used the database of the Fingerprint Verification Competition FCV2000 [2, 3].

The Database features are:

a) The fingerprints were acquired by using a low-cost optical sensor
b) The image size is 300x300.
c) The fingerprints are mainly from 20 to 30 year-old students (about 50% male).
d) Up to four fingers were collected for each volunteer (forefinger and middle finger of both the hands).
e) The images were taken from untrained people in two different sessions and no efforts were made to assure a minimum acquisition quality.
f) All the images from the same individual were acquired by interleaving the acquisition of the different fingers (e.g., first sample of left forefinger, first sample of right forefinger, first sample of left middle, first sample of right middle, second sample of the left forefinger...).

g) The presence of the fingerprint cores and deltas is not guaranteed since no attention was paid on checking the correct finger centering.
h) The sensor platens were not systematically cleaned (as usually suggested by the sensor vendors).

The acquired fingerprints were manually analyzed to assure that the maximum rotation is approximately in the range [-15°, 15°] and that each pair of impressions of the same finger have a non-null overlapping area. Sample images from the FCV200 database are shown in Figs. 2 and 3; each row shows different impressions:

Fig. 2. Images from FCV200 database; all the samples are from different fingers and all the samples are from different fingers and are ordered by quality (top-left: high quality, bottom-right: low quality).

2 Fuzzy Image Enhancement Based Pixels Brightness

In recent years, many researchers have applied the fuzzy set theory to develop new techniques for contrast improvement of the image. Following, some of the approaches of the fuzzy extraction feature method proposed to improvement the contrast of image in this case for fingerprints. Low contrast image has many pixels with similar brightness for an image of gray levels from 0 (black) to 255 (white). Without a significant change in brightness, low contrast images have an opaque appearance, while high-contrast images have a wide range of brightness, with many very light pixels and many very dark pixels. To improve the contrast is the process of changing an image of low contrast into a high contrast image and this can be done by mapping the brightness values of the original image to a new set of brightness values in the new image. The light pixels in the input image should be lighter in the output image and the dark pixels should be darker and the dark pixels in the middle range should stay more or less in the same range [10,11,14,15,17,18].

1. - Step. Calculate the Pixels Value Mean Image.
2. - Step. Obtained the fuzzy pixels value with the fuzzy extraction features.
3.- Step. Generate new gray-values with the following criterion:
For the pixel (i,j) of the Matrix Original Image
If g_{mn} mean < f(i,j), then $g_{mn}(i,j) = f(i,j)$
If g_{mn} mean >= f (i,j), then $g_{mn}(i,j) = g_{mn}$ mean
Where:
g_{mn} is the Original Matrix Image.
f_{mn} is the Fuzzy pixels value.
g_{mn} mean is the image mean pixels value.
The fuzzy rules of the extraction feature are:

1. If (input pixel is dark) then (output pixel is dark).
2. If (input pixel is gray) then (output pixel is gray).
3. If (input pixel is bright) then (output pixel is bright).

The rules of the fuzzy inference system define a mathematical function that increases the contrast of the image. The picture brightness varies from 0 (black) to 255 (white), with a one input variable and one output variable, for the fuzzy inference, with three gaussian membership functions input and three membership functions for the output, the value input pixel is modified in accordance with the rules and membership functions used. In Fig. 3 we show the input gaussian.membership functions, with Fuzzy Toolbox MATLAB [26].

In Fig. 4 we show the output gaussians membership functions.

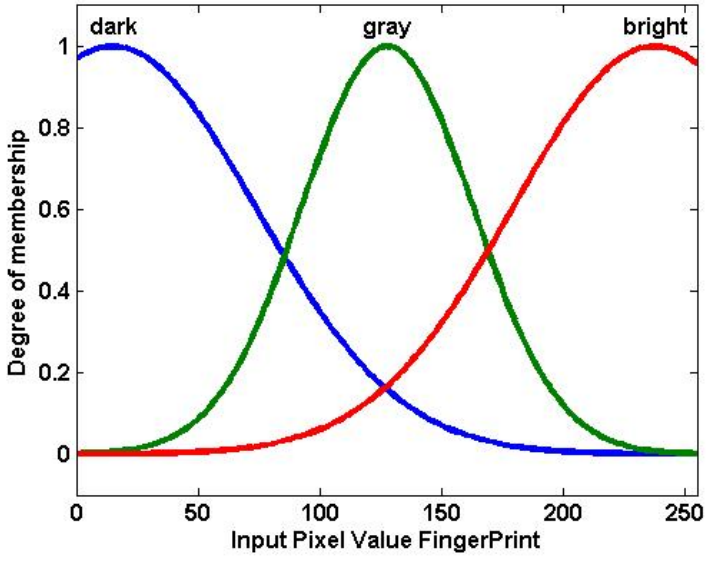


Fig. 3. Input Membership Function Fingerprint.

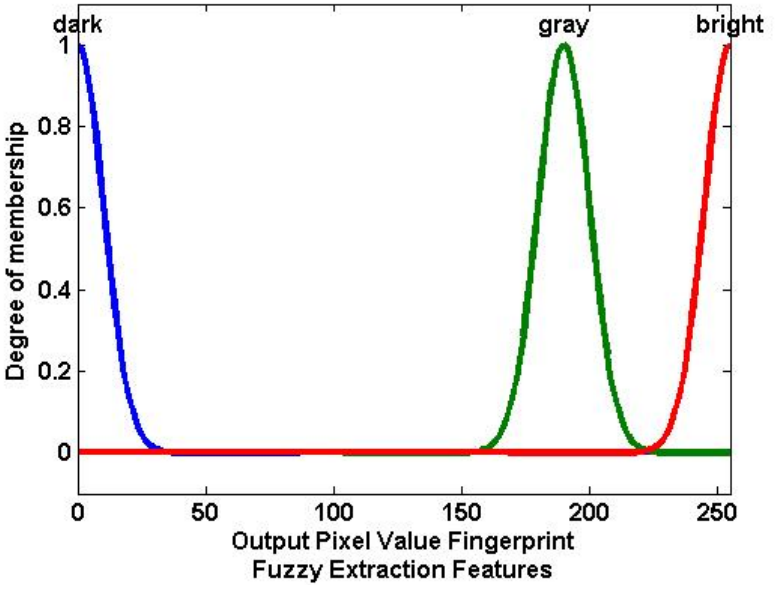


Fig. 4. Output Membership Function Fingerprint.

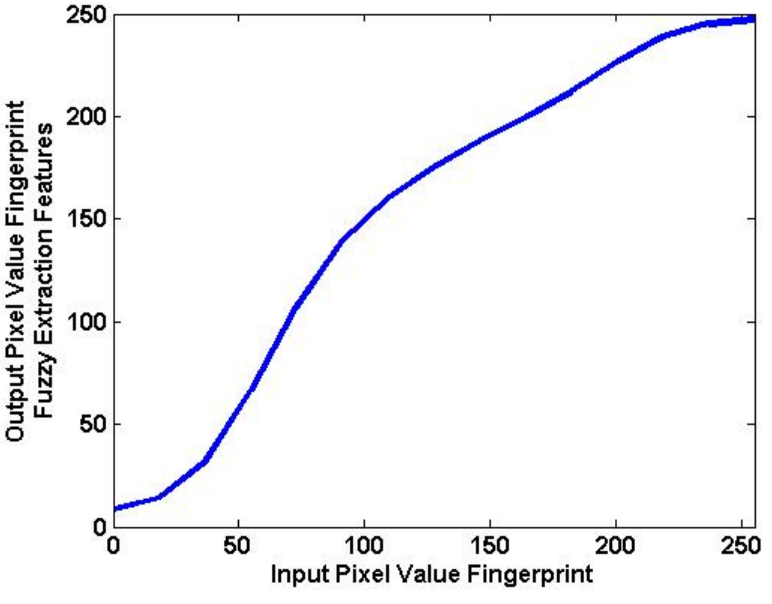


Fig. 5. Surface Solution of Fuzzy extraction features.

In Fig. 5 we show the surface solution of the fuzzy inference system, and it can be seen an almost linear behavior between the input and output pixels, observing that whenever a low contrast pixel (0 = black) remains the output pixel value, just as if the input pixel is high contrast (255 = bright) the fuzzy inference system responds with a output pixel value of the high contrast.

In Fig. 6 we show the behavior of the fuzzy extraction inference system using as input a fingerprint, showing on the right its histogram and at the bottom the output image obtained with the fuzzy extraction system and their respective histogram.

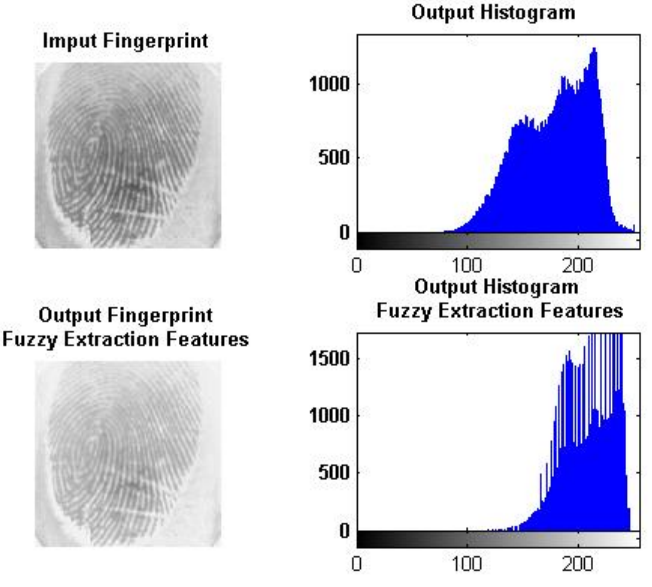


Fig. 6. Input and Output Fingerprint, and their respective Histogram.

The similar conditions of the type-1 fuzzy logic, we used for the fuzzy type-2 extraction features, the fuzzy rules, three input gaussian membership function and three outputs gaussian membership function, using the Matlab Toolbox developed for our research group [12,13], in Fig. 7 shown the input gaussian membership function.

In Fig. 8 we show the output gaussian membership function type-2.

In Fig. 9 we show the surface solution of the type-2 fuzzy extraction features, we can see that the behavior of input pixels with respect to the output pixels has linear behavior ranges pixels, obtaining a control contrast adaptive.

In Fig. 10 we show the type-2 fuzzy extraction features and their respective histogram.

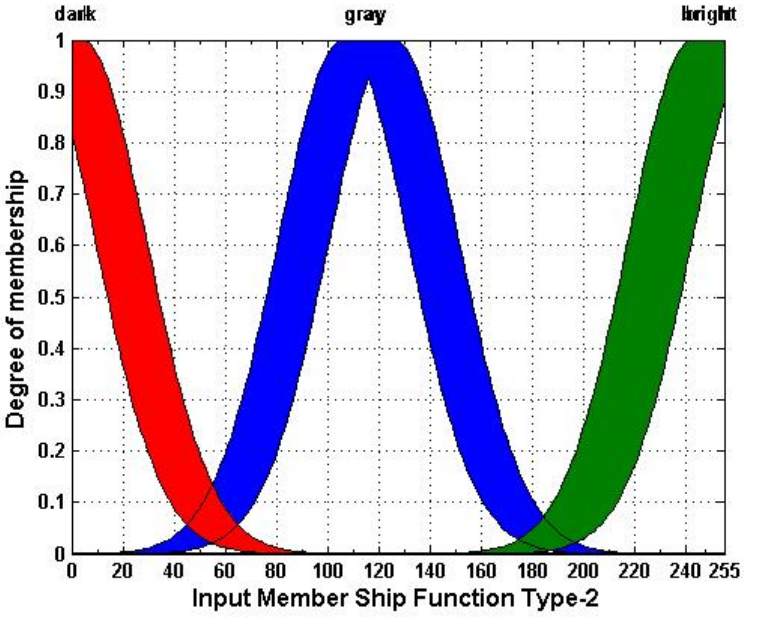


Fig. 7. Input Membership Function Fingerprint Type-2.

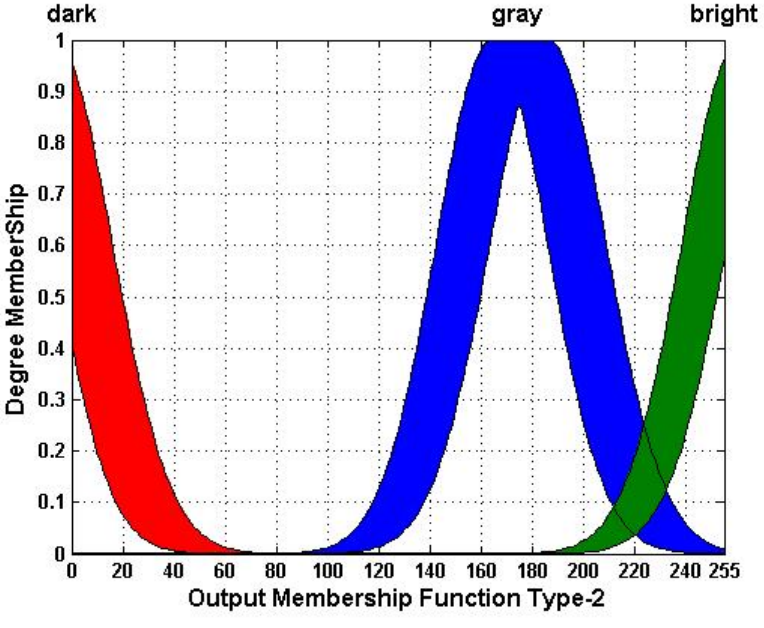


Fig. 8. Output Membership Function Fingerprint Type-2.

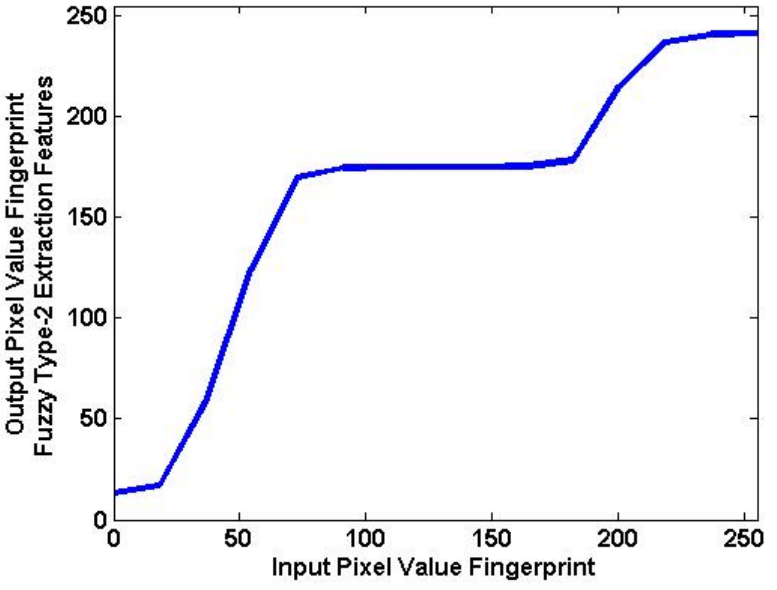


Fig. 9. Surface Solution of Type-2 Fuzzy extraction features.

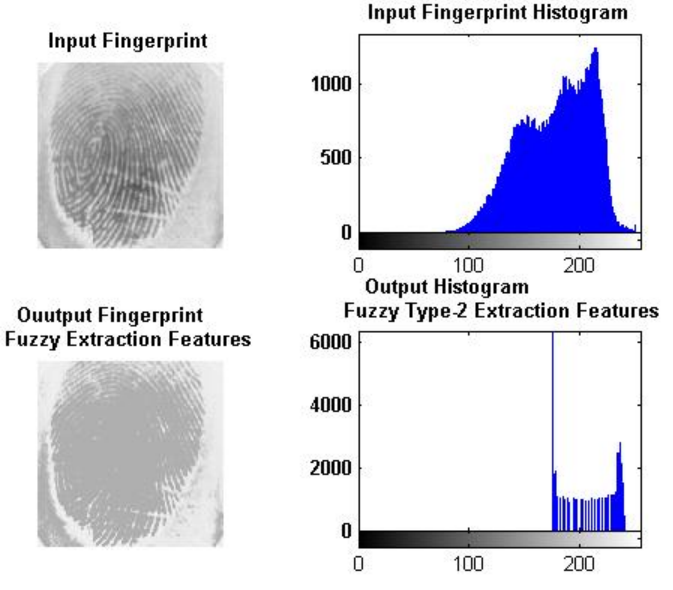


Fig. 10. Input and Output Fingerprint, and their respective Histogram, Fuzzy Type-2.

After obtaining the output pixel values from the fingerprint, using the criterion of pixels mean of the image to replace the pixel values by the pixel value mean or the fuzzy value obtained. When the pixel value is greater than or equal to the pixels mean in the image, then the pixel value is replaced by the fuzzy value obtained, and when the pixel value is less than the pixels mean is replaced by the pixel mean value of the image.

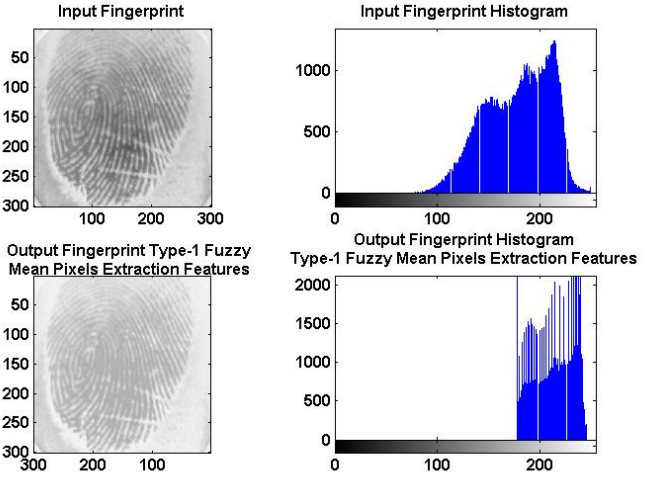


Fig. 11. Input and Output Fingerprint, and their respective Histogram, Type-1 Fuzzy Mean Pixels.

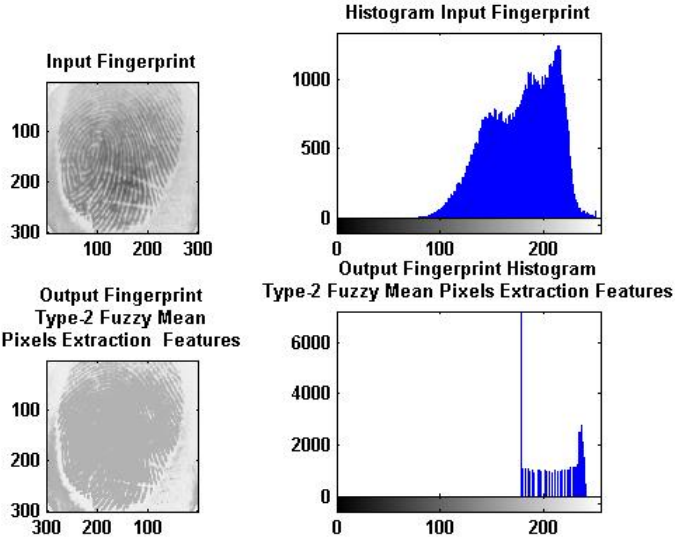


Fig. 12. Input and Output Fingerprint, and their respective Histogram, Type-2 Fuzzy Mean Pixels.

In Fig. 11 we show the input fingerprint and showing on the right its histogram and at the bottom the output image obtained with the fuzzy extraction system and their respective histogram using criterion of the pixels mean value.

In the Fig.12 we show the Input and Output Fingerprints, and their respective Histograms using Type-2 Fuzzy Extraction Mean Value.

3 Architecture Pattern Recognition with Fuzzy Extraction Features

The process of the pattern recognition system is performed by first training with a fingerprints database taken from the laboratory at the Bologna University FCV2004 [3], consisting of 10 persons with 8 samples per person, the fingerprint training is done with the architecture of a hybrid pattern recognition system [30,31,33] that consists of the preprocessing of the fingerprint image with the fuzzy extraction contrast method and then input to the Ensemble ANN´s architecture that consists of three main modules [16,23,25,27], in which each of them in turn consists of a set of neural networks trained with the same data (fingerprints image). Different parameters are used for each module of the Ensemble Neural Networks, for module 1 the used parameters learning rate =.001, Goal Error=.001, for the module 2 learning rate =.0001, Goal Error=.0001, for the module 3 the learning rate =.0001, and Goal Error=.00001, using 2 hidden layers, 36 neurons in the first layer and 17 neurons in the second layer for each module of the Ensemble Neural Network and is shown in Fig. 13.

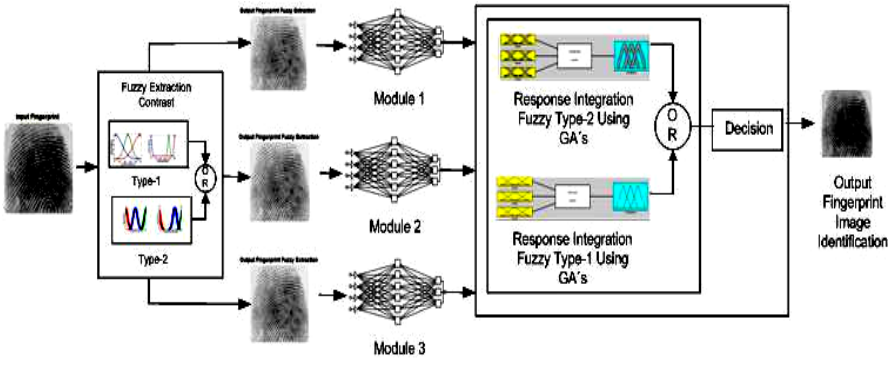


Fig. 13. Architecture of the Ensemble Neural Network for Fingerprint Recognition using Fuzzy Extraction Features.

4 Simulation Results Type-1 and Type-2 with Blur Motion

Once the Ensemble Neural Network is trained, the fuzzy inference system integrates the outputs of the modules [5,6,7,22,29]. We used the same 80 persons images to which we had applied different levels of noise with blur motion, both the type-1 and type-2 fuzzy inference systems give an output for the stage of the

final decision, and show the result if the fingerprint input was recognized. We show in the Fig.s 14, 15 and 16 the experimental results using the type-1 fuzzy inference system and in the Fig.s 17, 18 and 19 the experimental results using the type-2 fuzzy inference system, both fuzzy systems were optimized with the genetic algorithm [24, 27,32,35].

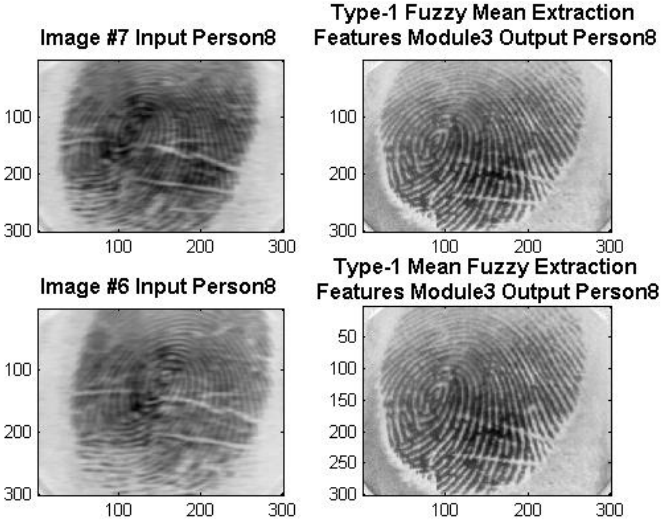


Fig. 14. Experimental results of the fingerprints using the type-1 Fuzzy Inference System (blur motion 10 displacement pixels).

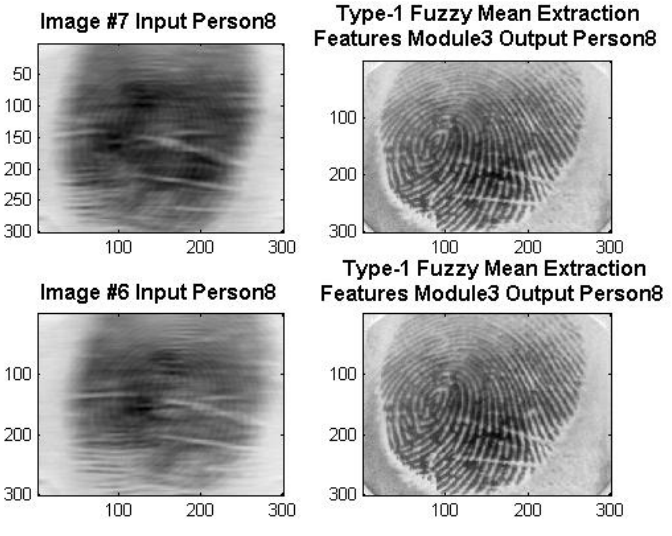


Fig. 15. Experimental results of the fingerprints using the type-1 Fuzzy Inference System (blur motion 30 displacement pixels).

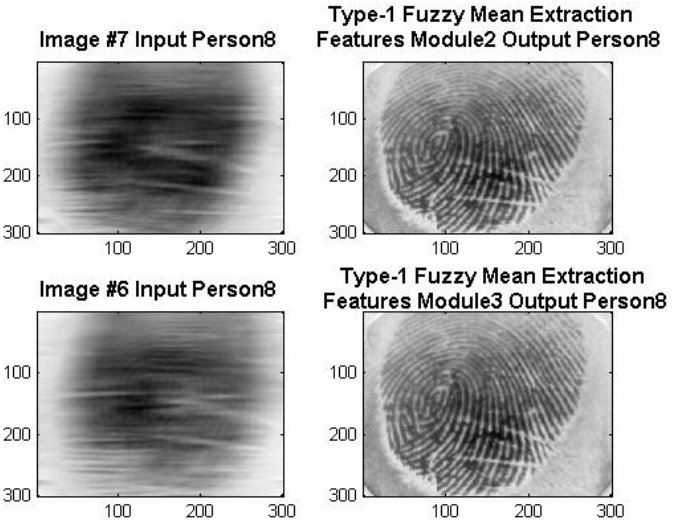


Fig. 16. Experimental results of the fingerprints using the type-2 Fuzzy Inference System (blur motion 50 displacement pixels).

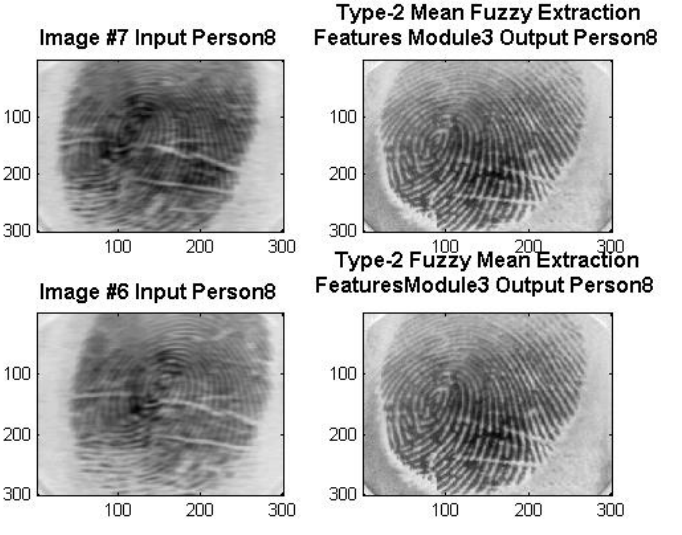


Fig. 17. Experimental results of the fingerprints using the type-2 Fuzzy Inference System (blur motion 10 displacement pixels).

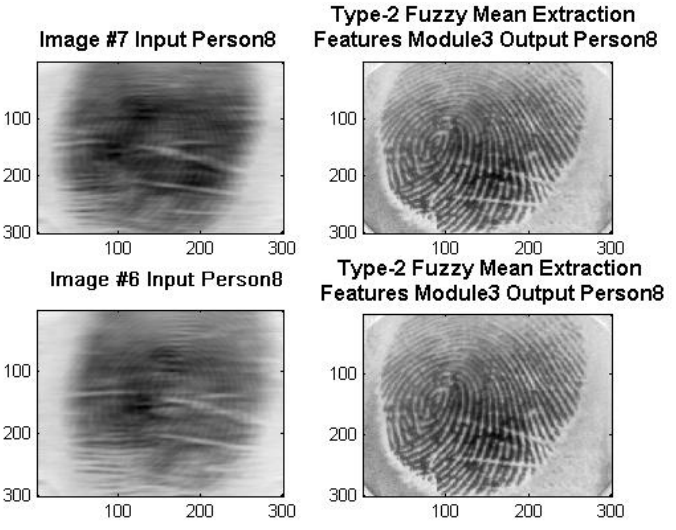


Fig. 18. Experimental results of the fingerprints using the type-2 Fuzzy Inference System (blur motion 30 displacement pixels).

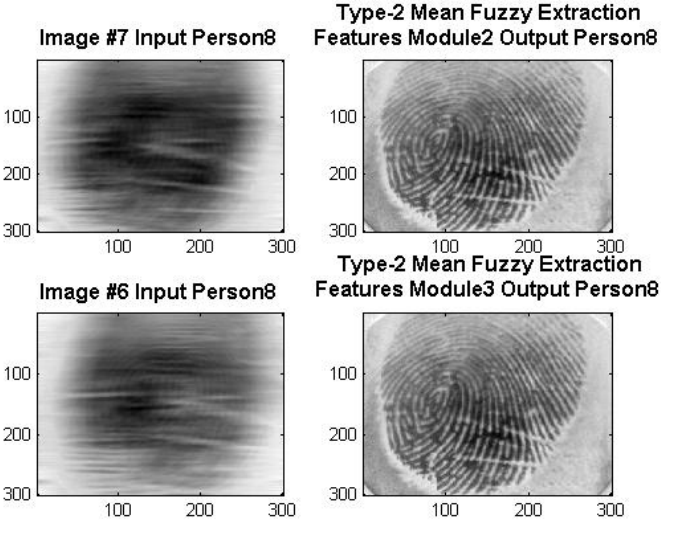


Fig. 19. Experimental results of the fingerprints using the type-2 Fuzzy Inference System (blur motion 50 displacement pixels).

5 Comparison of the Results Type-1 and Type-2 Fuzzy Logic Extraction Features

We performed 30 trials of the fingerprint recognition with an average of 100% recognition without noise for all methods. For the case in which noise was added to the fingerprint input the obtained results are as follows. The Identification Rate with blur motion of 10 displacement pixels is 98.75 % for type-1 (not optimized response integration) and also the same value for type-1 using optimized Membership functions, 100 % for type-1 GA's using type-1 Fuzzy Extraction Contrast also type-1 GA's using Type-1 Fuzzy Extraction Contrast with Mean Value Criterion.

For type-2 (not optimized response integration) is 98.75 % and also the same values for type-2 using optimized Membership function, 100 % for type-2 GA's

Table 1. Comparison Type-1 Fuzzy Extraction Features.

Response Integration Method	Recognition Rate	Identification Rate Blur Motion Noise Distance Pixels				
		10	20	30	40	50
Type-1 (not optimized)	80/80 100%	79/80 98.75%	77/80	72/80	69/80	66/80 82.5%
Type-1 using GA's	80/80 100%	79/80 98.75%	78/80	73/80	70/80	68/80 85%
Type-1 GA's using Type-1 Fuzzy Extraction Contrast	80/80 100%	80/80 100%	79/80	76/80	74/80	71/80 88.75%
Type-1 GA's using Type-1 Fuzzy Extraction Contrast with Mean Value Criterion	80/80 100%	80/80 100%	79/80	77/80	76/80	72/80 90%

Table 2. Comparison Type-2 Fuzzy Extraction Features.

Response Integration Method	Recognition Rate	Identification Rate Blur Motion Noise Distance Pixels				
		10	20	30	40	50
Type-2 (not optimized)	80/80 100%	79/80 98.75%	78/80	74/80	70/80	68/80 85%
Type-2 using GA's	80/80 100%	79/80 98.75%	78/80	75/80	72/80	70/80 87.5%
Type-2 GA's using Type-2 Fuzzy Extraction Contrast	80/80 100%	80/80 100%	79/80	76/80	74/80	72/80 90%
Type-2 GA's using Type-2 Fuzzy Extraction Contrast with Mean Value Criterion	80/80 100%	80/80 100%	79/80	77/80	76/80	73/80 91.25%

using type-1 Fuzzy Extraction Contrast also type-1 GA's using Type-2 Fuzzy Extraction Contrast with Mean Value Criterion.

When The Level Noise increase to 50 displacement pixels the best identification rate is obtained with the Type-2 Fuzzy Extraction Contrast with Mean Value Criterion 91.25 %, in table 1, and table 2, we show comparison results between fuzzy logic type-1 and type-2 with blur motion level noise.

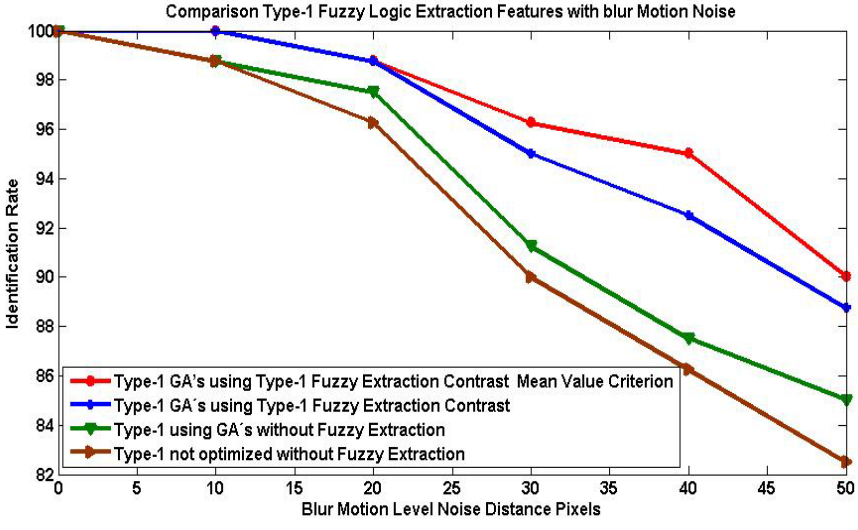


Fig. 20. Comparison Type-1 Fuzzy Logic Extraction Features (blur motion level noise 10 to 50 displacement pixels).

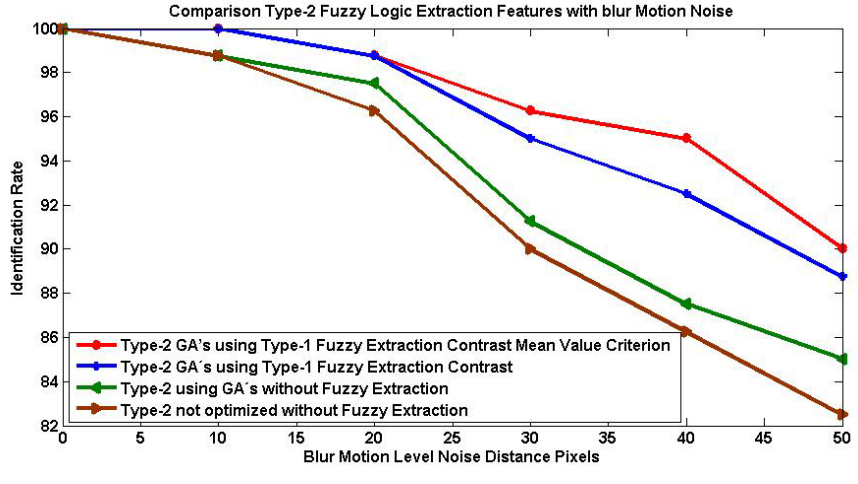


Fig. 21. Comparison Type-2 Fuzzy Logic Extraction Features (blur motion level noise 10 to 50 displacement pixels).

In Fig. 20 we show the comparison of Results with Type-1 Fuzzy Logic Extraction Features.

In Fig. 21 we show the comparison of Results with Type-2 Fuzzy Logic Extraction Features.

In the Fig. 22 we show the Comparison of Results with Type-1 and Type-2 Fuzzy Logic Extraction Features.

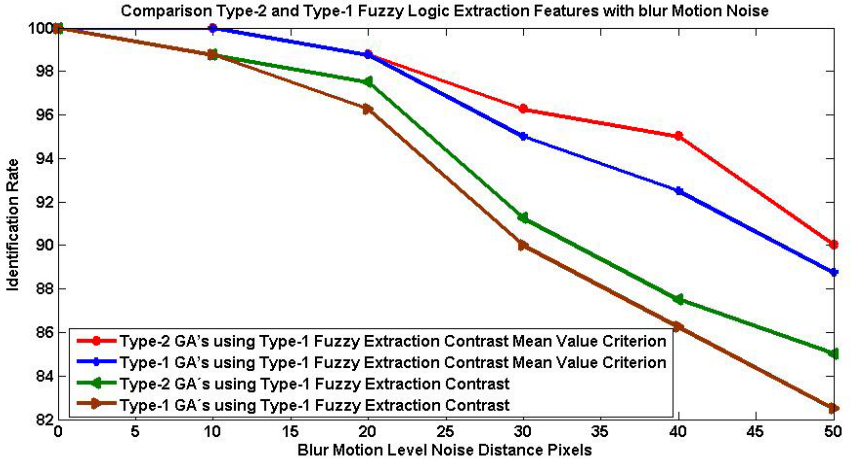


Fig. 22. Comparison between Type-1, and Type-2 Fuzzy Extraction Features (blur motion level noise 10 to 50 displacement pixels).

6 Conclusions

Based on the experimental results, we can conclude that when we used fuzzy logic fuzzy extraction features of each module of the Ensemble Neural Network improved results which had obtained when they do not used preprocessing for each module.

The difference between the type-1 fuzzy and type-2 fuzzy inference systems for extraction features of ensemble neural networks is appreciated when we use blur motion level noise. We can conclude that the behavior can improve; we think that there is an advantage in using a type-2 fuzzy inference system to manage the uncertainty of the knowledge base in pattern recognition problems.

Future work will include, using genetic algorithms to optimize membership function and fuzzy rules of type-1 and type-2 fuzzy logic extraction features, the convergence criterion genetic algorithm is to obtain the best identification rate of the pattern recognition system based on the control of contrast of the images of fingerprints

Testing with other level of noise, using edges like feature extraction, and other methods of the image compression, and using parallel processing algorithms with the goal of improving the identification rate.

Acknowledgments

We would like to express our gratitude to the CONACYT, Universidad Autonoma de Baja California and Tijuana Institute of Technology for the facilities and resources granted for development of this research project.

References

[1] Sharkey, A.C.: Modularity, combining and artificial neural nets. Connection Science 8, 299–313 (1996)
[2] Castiello, C., Castellano, G., Caponetti, L., Fanelli, A.M.: Fuzzy classification of image pixels, Intelligent Signal Processing. In: 2003 IEEE International Symposium on Digital Object Identifier, September 4-6, pp. 79–82 (2003), doi:10.1109/ISP.2003.1275817
[3] Maltoni, D., Cappelli, R., Wayman, J.L., Jain, A.K.: FVC2004: Third Fingerprint Verification Competition. In: Zhang, D., Jain, A.K. (eds.) ICBA 2004. LNCS, vol. 3072, pp. 1–7. Springer, Heidelberg (2004)
[4] Maltoni, D., Maio, D., Jain, A.K., Prabhakar, S.: The full FVC2000 and FVC2002 databases are available in the DVD included in: Handbook of Fingerprint Recognition. Springer, New York (2003)
[5] Opitz, D.W., Shavlik, J.W.: Generating accurate and diverse members of a neural network ensemble. In: Touretzky, D.S., Mozer, M., Hasselmo, M. (eds.) Advances in Neural Information Processing Systems, vol. 8, pp. 535–541. MIT Press, Cambridge (1996)
[6] Opitz, D.W., Maclin, R.: Popular Ensemble Methods: An Empirical Study. Journal of Artificial Intelligence Research 11, 169–198 (1999)
[7] Opitz, D.W.: Feature Selection for Ensembles. In: Sixteenth National Conference on Artificial/ Intelligence (AAAI), Orlando, FL, pp. 379–384 (1999)
[8] Hoffmann, F.: Evolutionary Algorithms for Fuzzy Control System Design. In: Proceedings of the IEEE, special issue on Industrial Applications of Soft Computing (to appear October 2001)
[9] Nemmour, H., Chibani, Y.: Neural Network Combination by Fuzzy Integral for Robust Change Detection in Remotely Sensed Imagery. EURASIP Journal on Applied Signal Processing 14, 2187–2195 (2005)
[10] Tizhoosh, H.R., Michaelis, B., Guericke, O.V.: Image Enhancement Based on Fuzzy Aggregation techniques. Published in Proceedings of 16th IEEE IMTC 1999, Venice, Italy, vol. 3, pp. 1813–1817 (1999)
[11] Kim, J., Peli, E.: MPEG based image enhancement for people with low vision, Soc for Information Display, Digest of Technical Papers, pp. 1156–1159 (2003)
[12] Castro, J.R., Castillo, O., Melin, P.: An Interval Type-2 Fuzzy Logic Toolbox for Control Applications. In: Proc. FUZZ-IEEE 2007 (2007)
[13] Castro, J.R., Castillo, O., Melin, P., Martinez, L.G., Escobar, S., Camacho, I.: Building Fuzzy Inference Systems with Interval Type-2 Fuzzy Logic Toolbox, 1st edn. Studies in Fuzziness and Soft Computing, vol. 6(1), pp. 53–62. Springer, Heidelberg (2007)
[14] Tang, J.: A contrast based image fusion technique in the DCT domain. Digital Signal Processing 14(3), 218–226 (2004)
[15] Tang, J., Peli, E., Actonm, S.: Image enhancement using a contrast measure in the compressed domain. IEEE Signal Processing Letters 10(10), 289–292 (2003)
[16] Urias, J., Solano, D., Soto, M., Lopez, M., Melin P.: Type-2 Fuzzy Logic as a Method of Response Integration in Modular Neural Networks. In: IC-AI 2006, pp. 584–590 (2006)

[17] Leu, J.G.: Image contrast enhancement based on the intensities of edge pixels. CVGIP: Graphical Models and Image Processing 54(6), 497–506 (1992)
[18] Sangkeun, L.: Content-based image enhancement in the compressed domain based on multi-scale alpha-rooting algorithm. PRL 10(27), 1054–1066 (2006)
[19] Zadeh, L.A.: Fuzzy Logic. Computer 1(4), 83–93 (1998)
[20] Grabisch, M.: A new algorithm for identifying fuzzy measures and its application to pattern recognition. In: Proc. of 4th IEEE Int. Conf. on Fuzzy Systems, Yokohama, Japan, pp. 145–150 (1995)
[21] Grabisch, M., Murofushi, T., Sugeno, M.: Fuzzy Measures and Integrals: Theory and Applications, pp. 348–373. Physica, NY (1989)
[22] Lopez, M., Melin, P., Castillo, O.: A Method for Creating Ensemble Neural Networks Using a Sampling Data Approach. IFSA (2), 772–780 (2007)
[23] Lopez, M., Melin, P.: Response integration in Ensemble Neural Networks using interval type-2 Fuzzy logic. In: IJCNN 2008, pp. 1503–1508 (2008)
[24] Lopez, M., Melin, P., Castillo, O.: Pattern Recognition in Blur Motion Noisy Images using Fuzzy Methods for Response Integration in Ensemble Neural Networks. In: IFSA/EUSFLAT Conf. 2009, pp. 809–814 (2009)
[25] Allah, M.M.A.: Artificial Neural Networks Fingerprints Authentication with Clusters Algorithm. Informatica 29, 303–307 (2005)
[26] MATLAB Trade Marks, ©1994-2007 by The Math Works, Inc
[27] Lopez, M., Melin, P., Castillo, O.: Optimization of Response Integration with Fuzzy Logic in Ensemble Neural Networks Using Genetic Algorithms. In: Soft Computing for Hybrid Intelligent Systems, pp. 129–150 (2008)
[28] Mendoza, O., Melin, P., Licea, G.: Modular Neural Networks and Type-2 Fuzzy Logic for Face Recognition. In: Reformat, M. (ed.) Proceedings of NAFIPS 2007, San Diego, June 2007, vol. 1 (2007) (pages CD Rom)
[29] Cunningham, P.: Overfitting and Diversity in Classification Ensembles based on Feature Selection, TCD Computer Science Technical Report, TCD-CS-2000-07
[30] Melin, P., Castillo, O.: Hybrid Intelligent Systems for Pattern Recognition using Soft Computing. Springer, Heidelberg (2005)
[31] Melin, P., Mancilla, A., Lopez, M., Mendoza, O.: Hybrid modular neural network architecture with fuzzy Sugeno integration for time series forecasting. Appl. Soft Comput. 7(4), 1217–1226 (2007)
[32] Melin, P., Mancilla, A., Lopez, M., Solano, D., Soto, M., Castillo, O.: Pattern Recognition for Industrial Security using the Fuzzy Sugeno Integral and Modular Neural Networks. In: WSC11 11th Online World Conference on Soft Computing in Industrial Applications, September 18-October 6 (2006)
[33] Melin, P., Mancilla, A., Lopez, M., Trujillo, W., Cota, J., Gonzalez, S.: Modular Neural Networks with Fuzzy Integration Applied for Time Series Forecasting. Analysis and Design of Intelligent Systems using Soft Computing Techniques, 217–225 (2007)
[34] Melin, P., González, F., Martínez, G.: Pattern Recognition Using Modular Neural Networks and Genetic Algorithms. In: IC-AI 2004, pp. 77–83 (2004)
[35] Melin, P., Urias, J., Solano, D., Soto, M., Lopez, M., Castillo, O.: Voice Recognition with Neural Networks, Type-2 Fuzzy Logic and Genetic Algorithms. Engineering Letters 13(2), 108–116 (2006)
[36] Gutta, S., Huang, J., Takacs, B., Wechsler, H.: Face Recognition Using Ensembles of Netrworks. In: 13th International Conference on Pattern Recognition (ICPR 1996), Vienna, Austria, vol. 4, p. 50 (1996)
[37] Du, Y., Ives, R.W., Etter, D.M., Welch, T.B.: Biometric Signal Processing Laboratory. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), May 2004, vol. 5, pp. 1025–1028 (2004)

Application of the Bee Swarm Optimization BSO to the Knapsack Problem

Marco Aurelio Sotelo-Figueroa, Rosario Baltazar, and Martín Carpio

Instituto Tecnológico de León, Av. Tecnológico S/N, 37290
León, Guanajuato, México
masotelof@gmail.com, charobalmx@yahoo.com.mx,
jmcarpio61@hotmail.com

Abstract. Swarm Intelligence is the part of Artificial Intelligence based on study of actions of individuals in various decentralized systems. The optimization algorithms which are inspired from intelligent behavior of honey bees are among the most recently introduced population based techniques. In this paper, a novel hybrid algorithm based in Bees Algorithm and Particle Swarm Optimization is applied to the Knapsack Problem. The Bee Algorithm is a new population-based search algorithm inspired by the natural foraging behavior of honey bees, it performs a kind of exploitative neighborhood search combined with random explorative search to scan the solution, but the results obtained with this algorithm in the Knapsack Problem are not very good. Although the combination of BA and PSO is given by BSO, Bee Swarm Optimization, this algorithm uses the velocity vector and the collective memories of PSO and the search based on the BA and the results are much better.

Keywords: Swarm Optimization, PSO, BA, BSO, Knapsack Problem.

1 Introduction

Evolutionary and meta-heuristic algorithms have been extensively used as search and optimization tools during this decade in several domains from science and engineering to the video game industry, and others.

Many demanding applications that involve the solution of optimization problems of high complexity, a lot of these belonging to a special class of problems called NP-hard have been solved by various methods [8]. Metaheuristic algorithms are now considered among the best tools must to find good solutions with a reasonable investment of resources.

As described by Eberhart and Kennedy [4] Particle Swarm Optimization or *PSO* algorithm is part of the Swarm Intelligence and is a metaheuristics that use the social-psychological metaphor; a population of individuals, referred to as particles, adapts by returning stochastically toward previously successful regions.

P. Melin et al. (Eds.): Soft Comp. for Recogn. Based on Biometrics, SCI 312, pp. 191–206.
springerlink.com

The PSO simulate a society where each individual contributes with his knowledge to the society. These metaheuristics have proved their ability to deal with very complicated optimization and search problems.

The behavior of a single ant, bee, termite or wasp often is too simple, but their collective and social behavior is of paramount significance. The collective and social behavior of living creatures motivated researchers to undertake the study of today what is known as Swarm Intelligence [5]. Two fundamental concepts, self-organization and division of labor, are necessary and sufficient properties to obtain swarm intelligent behavior.

The Bee Algorithm or *BA* [9] is also part of the *Swarm Intelligence* and this mimics the honey bees and who this search their food foraging. This algorithm is based on a random search on the neighborhood for combinatorial and functional optimization.

The *Knapsack Problem* is a classical combinatorial problem [3][14] can be described as follows: "Imagine taking a trip to which you can only carry a backpack that, logically, has a limited capacity. Given a set of items, each with a weight and a value, determine the number of each item to include in a bag so that the total weight is less than a given limit and the total value is as large as possible", this problem can be considerate as NP-easy problem but some studies show that the Knapsack Problem is an NP-Hard problem [2].

Actually the Knapsack Problem can be modeled by different ways [15] for example multi-dimensional and multiple Knapsack problem [1][13][16], quadratic Knapsack problem [6][12].

In the present paper we introduce the *Bee Swarm Optimization* or *BSO*. This algorithm is a hybrid metaheuristic between the *PSO* and the *BA*. The *BSO* use the better characteristics from both algorithms, the *Social Metaphor* from the *PSO* and the random search in the neighborhood from the *BA*, and give us a better result. The experiments were made on seven types of instances from uncorrelated, to strongly correlated. All these instances probe the algorithm varying the parameters of the profits and the weight. The algorithm was probed with an initial population of random solutions that meet the next conditions: reproducibility, without repetition inside a determinate length chain, statistically independent, fast generation process, and a minimal space inside memory. This pseudorandom number generator helps the algorithm to work better. Another important characteristic is the Confidence Interval of the results.

2 Knapsack Problem

The Knapsack problem [14] is the typical combinatorial problem that has been studied along many years and was proved that it is a NP-Hard problem [11]. The basic problem is the 0-1 Knapsack Problem or Binary Knapsack Problem and it have a search space of 2^n - possible solutions.

The Knapsack Problem can be described as follows: there are objects, each of this objects have a profit and weight, and needs to select those whose sum of their benefits is maximized subject to the sum of the weight of the same objects should not exceed an amount determined. It can be formulated mathematically by

numbering each of its objects or items from to and introducing it to a vector of binary variables $x_j (j = 1,2,3,\cdots,$, where each variable represented here will take the value 1 or 0 depending on whether it is selected or not.

The solution to the Knapsack Problem is select a subset of objects from the binary vector , solution vector, that satisfies the constraint on the equation 2 and the same time maximize the objective function on the equation 1.

$$z = \sum_{j=1}^{n} p_j x_j \tag{1}$$

$$\sum_{j=1}^{n} w_j x_j \leq c$$
$$x_j = \begin{cases} 1 & \text{If the j object is selected} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where:

- represents the j-th object.
- indicates whether the j object is part of the solution.
- is the j-th object profit.
- is the j-th object weight.
- is the volume or capacity of the knapsack.

2.1 Types of Knapsack Problem Instances

The Knapsack Problem is affected by the relationship between the profit and the weight of the objects; these types of instances are the following:

- **Uncorrelated:** the profits and Weight are distributed uniformly between one and a maximum *V* number.

$$p_j \in [1, V]; \; w_j \in [1, V] \tag{3}$$

- **Weakly correlated:** the Weight is distributed uniformly between one and a maximum *V* number and the profits are distributed uniformly around the weight and an *R* ratio.

$$w_j \in [1, V]; \; p_j \in [w_j - R, w_j + R] \tag{4}$$

- **Strongly correlated:** the Weight is uniformly distributed between one and a maximum *V* number; the profits are the Weight plus one *K* constant.

$$w_j \in [1, V]; \; p_j = w_j + K \tag{5}$$

- **Inverse strongly correlated:** the profits are distributed uniformly between one and a maximum V number and the Weight is the profits plus one K constant.

$$p_j \in [1, V]: \; w_j = p_j + K \tag{6}$$

- **Almost strongly correlated:** the Weight is distributed uniformly between one and a maximum V number and the profits are the Weight plus one random number between one and a maximum S number.

$$S \in [1, U]: \; w_j \in [1, V]: \; p_j = w_j + S \tag{7}$$

- **Subset-sum:** the profits and Weight have the same value and are distributed uniformly between one and a maximum V number.

$$w_j \in [1, V]: \; p_j = w_j \tag{8}$$

- **Uncorrelated similar Weight:** the profits are distributed uniformly between one and a maximum V_1 number and the Weight is distributed uniformly between one and a maximum V_2 number plus a K constant.

$$+K \tag{9}$$

3 Bee Algorithm (BA)

The *Bees Algorithm* [9] or *BA* is a bio-inspired metaheuristic behavior of honey bees and how they searching for plant to obtain the necessary pollen for honey production.

A colony of bees search in a large territory looking for new sources of food and begins to thrive on discovering new food sources. When these sources have much pollen are visited by large numbers of bees and when the pollen decreases the number of bees collected from these sources decreases too.

When season for recollecting pollen start, the colony sent so many bees, which are called *scouts bees* to reconnoiter randomly the territory to inform at the colony where are the best food sources. Once the harvesting season starts the colony maintains a certain percentage of their scout bees in order to detect new and better sources of food. When scout bees have returned to the colony and found a better food source than the currently is using the colony, makes the *Dance* by means of which transmits the exact position of the source food and then the colony began to send more bees to the food source.

The Algorithm of the *BA* applied to the Knapsack Problem implemented in the present paper is the following.

```
Require: ps population size, n neighborhood
Initialize population with random solutions.
Evaluate fitness of the population.
While stopping criterion not met do

   select the n sites for neighborhood search
   recruit bees for n selected sites and evaluate
their        fitness
   select the fittest bee from each site
   assign remaining bees to search randomly and eva-
luate their fitness
end while
```

4 Particle Swarm Optimization (PSO)

The *Particle Swarm Optimization* [4][7] or *PSO* is a Bio-inspired metaheuristic in flocks of birds or schools of fish. It was developed by J. Kennedy and R. Eberthart based on a concept called *social metaphor*, this metaheuristics simulates a society where all individuals contribute their knowledge to obtain a better solution, there are three factors that influence for change in status or behavior of an individual:

- The Knowledge of the environment or adaptation which speaks of the importance given to the experience of the individual.
- His Experience or local memory is the importance given to the best result found by the individual.
- The Experience of their neighbors or Global memory referred to how important it is the best result I have obtained their neighbors or other individuals.

In this metaheuristic each individual is called particle and moves through a multidimensional space that represents the social space or search space depends on the dimension of space which depends on the variables used to represent the problem.

For the update of each particle using something called *velocity vector* which tells them how fast it will move the particle in each of the dimensions, the method for updating the speed of *PSO* is given by equation (10), and it is updating by the equation (11).

$$v_i = \varphi_0 * v_i + \varphi_1(x_i - B_{Global}) + \varphi_2(x_i - B_{Local}) \tag{10}$$

$$x_i = x_i + v_i \tag{11}$$

where:

- is the velocity of the i-th particle
- is adjustment factor to the environment.
- is the memory coefficient in the neighborhood.
- is the coefficient memory.

- is the position of the i-th particle.
- B_{Global} is the best position found so far by all particles.
- B_{Lo} is the best position found by the i-th particle

The Algorithm of the PSO applied to the Knapsack Problem implemented in the present paper is the following.

```
Require: φ0 adaptation to environment coefficient, φ1
local memory coefficient, φ2 neighborhood memory
coefficient, n swarm size
Start the swarm particles.
Start the velocity vector for each particle in the
swarm.
While stopping criterion not met do
   for i = 1 to n do
     if the i-particle's fitness is better than the
local best then replace the local best with the i-
particle
     if the i-particle's fitness is better than the
global best then replace the global best with the i-
particle.
     Update the velocity vector
     Update the particle's position with the velocity
vector
   end for
end while
```

5 Bee Swarm Optimization (BSO)

The Bee Swarm Optimization or BSO is a hybrid metaheuristic population between the PSO and the BA. The main idea of BSO is based on taking the best of each metaheuristics to obtain better results than they would obtain.

The BSO use the *velocity vector* and the way to updating it, equation (10), and applies the social metaphor to get better results from the PSO and use the exploration and a search radius from the BA to indicate which is where they look for a better result.

The first thing that the BSO do is update the position of the particles through the velocity vector and then select a specified number of particles, in this case it is proposed to select the best as being the new food supply that the scout bees discovered, and conducted a search of the area enclosed by the radius search and if there are a better solution in this area than the same particle around which are looking for then the particle is replaced with the position of the best solution found in the area.

For this metaheuristic to work properly you need a way to determine what the next and earlier particle position, if we cannot determine this then it is impossible

to apply this metaheuristic because you would not know what the next and previous particle to search in that area.

We defined the next and before elements like binary operations, adding and subtracting one number to the solution vector. For example if we have a Knapsack Problem with 5 elements the solution vector will have 5 spaces and in each space can be 0 or 1, we can see the example of the next and before solution vector in the Figure 1.

Fig. 1. Example of the previous and next solution vector

The algorithm of the *BSO* applied to the Knapsack Problem implemented in the present paper is the following.

```
Require: φ0 adaptation to environment coefficient, φ1
local memory coefficient, φ2 neighborhood memory
coefficient, n swarm size, sb scout bees, r search
radio
Start the swarm particles.
Start the velocity vector for each.
While stopping criterion not met do
   for i = 1 to n do
     if the i-particle's fitness is better than the
local best then replace the local best with the i-
particle
     if the i-particle's fitness is better than the
global best then replace the global best with the i-
particle.
     Update the velocity vector
     Update the particle's position with the velocity
vector
     Choose the best sb particles
     for all best  sb  particle
       Search if there are some better particle in
the search radio and if exist it replace the particle
with the best particle in the search radio
     end for all
   end for
end while
```

6 Experiments

To Test the BSO, the Generator of Knapsack Test Instances [10] from the Dr. David Pisinger was used; it requires the number of elements and the coefficients range. Were generate the seven types of test instances described, and was use the same parameters for each metaheuristic to find which of this are better for the Knapsack Problem. Each metaheuristic was run 100 times for obtaining their *average* and *standard deviation*, and was use that data for calculating the *Confidence Interval* at the 97% of confidence.

We determinate the size of the *Knapsack* obtaining the *average size* of the instance and considerate it *average* like the size of the elements and multiply it by 20 for considerate 20 elements.

Table 1. Parameters used in the *Generator of Knapsack Test Instances*.

Parameters	Values
Elements	50
Coefficients Range	[0,100]

Table 2. Parameters used in the *BA*.

Parameters	Values
Elements	50
Iterations	100
Elements for Neighborhood	10
Search Radio	3

Table 3. Parameters used in the *PSO*.

Parameters	Values
Elements	50
Iterations	100
φ_0	1
φ_1	0.5
φ_2	0.8

Table 4. Parameters used in the *PSO*.

Parameters	Values
Elements	50
Iterations	100
Elements for Neighborhood	10
Search Radio	3
φ_0	1
φ_1	0.5
φ_2	0.8

7 Results

We show the results obtained by testing each Type Instances, the Average Comfort and Average Size, the Best Comfort and Best Size and their Standard Deviation for each metaheuristic and their Confidence Interval. We also show the metaheuristic's behavior through their graphic.

7.1 Uncorrelated Knapsack Problem

To test the Uncorrelated instance the Knapsack size was defined a 2000. We can see the results of each metaheuristic are in the Table 5 and their behavior in the Figure 2. As can be seen the BA gives the worst Average and Best Comfort, on the other hand, the PSO and the BSO give the same Best Comfort but with a smaller Standard Deviation in the case of the BSO which gives us a smaller Confidence Interval.

Table 5. Results obtained from the *Uncorrelated Knapsack Problem*.

Metaheuristic	Average Comfort	Average Size	Best Comfort	Best Size	Comfort	Confidence Interval
BA	1940.17	1902.93	2172	1933	88.46	[1913.02, 1967.32]
PSO	2374.04	1996.14	2375	2000	1.87	[2373.47, 2374.61]
BSO	2374.33	1995.34	2375	2000	1.78	[2373.78, 2374.88]

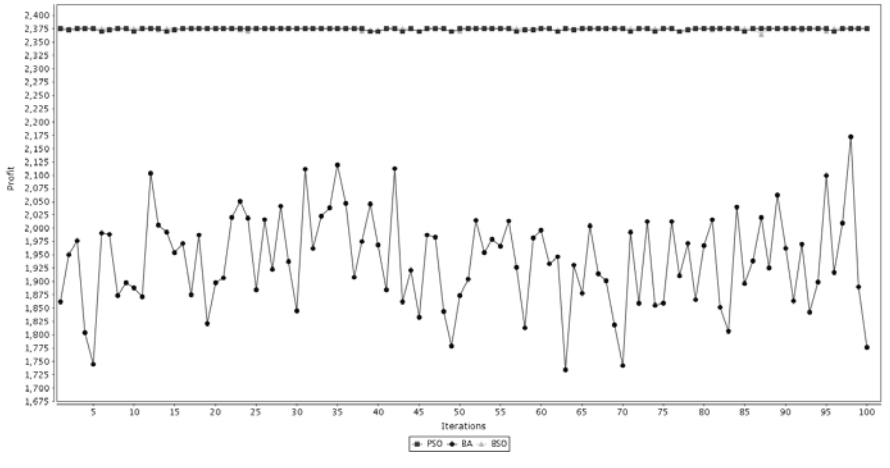


Fig. 2. Graphic behaviors of metaheuristics applied to the *Uncorrelated Knapsack Problem*, the dots represent the *BA*, the squares the *PSO* and the triangles the *BSO*.

7.2 Weakly Correlated

To test the Weakly Correlated instance, the Knapsack size was defined at 1800. We can see the results of each metaheuristic in the Table 6 and their behavior in the Figure 3. As we can see the BA gives the worst Average and Best Comfort, on the other hand, the PSO and the BSO give the same Best Comfort but with a smaller Standard Deviation in the case of the BSO which gives us a smaller Confidence Interval.

Table 6. Results obtained from the *Weakly Correlated Knapsack Problem*.

Metaheuristic	Average Comfort	Average Size	Best Comfort	Best Size	Comfort	Confidence Interval
BA	1778.79	1756.72	1833	1799	36.59	[1767.56, 1790.02]
PSO	1907.76	1798.5	1917	1799	5.1	[1906.19, 1909.33]
BSO	1911.68	1798.23	1917	1799	3.16	[1910.71, 1912.65]

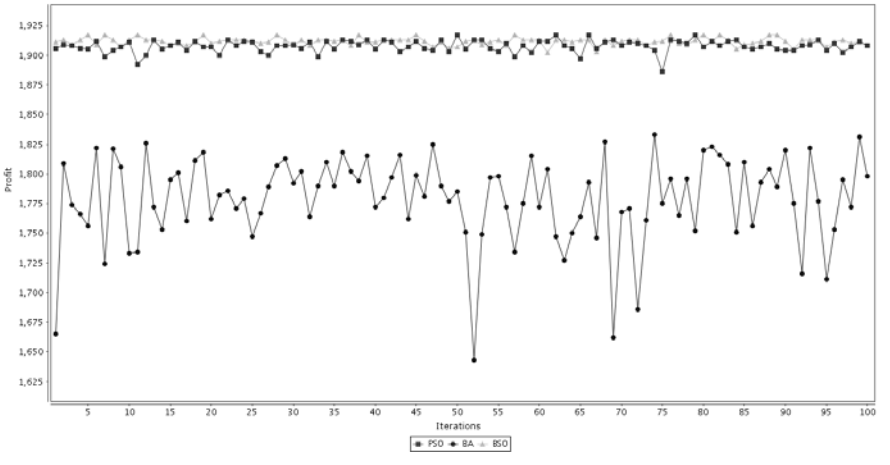


Fig. 3. Graphic behaviors of metaheuristics applied to the *Weakly Correlated Knapsack Problem*, the dots represent the *BA*, the squares the *PSO* and the triangles the *BSO*.

7.3 Strongly Correlated

To test the Strongly Correlated instance, the Knapsack size was defined at 2000. We can see the results of each metaheuristic in the Table 7 and their behavior on the Figure 4. As seen the BA gives the worst Average and Best Comfort, by other hand, the PSO gives a high Average and Best Comfort that the BSO, this give us a smaller Confidence Interval for the BSO.

Table 7. Results obtained from the *Strongly Correlated Knapsack Problem*

Metaheuristic	Average Comfort	Average Size	Best Comfort	Best Size	Comfort	Confidence Interval
BA	2323.58	1944.78	2400	2000	63.26	[2304.17, 2342.99]
PSO	2429.29	1999.59	2430	2000	2.02	[2428.67, 2429.91]
BSO	2430.08	1999.98	2439	1999	0.91	[2429.8, 2430.36]

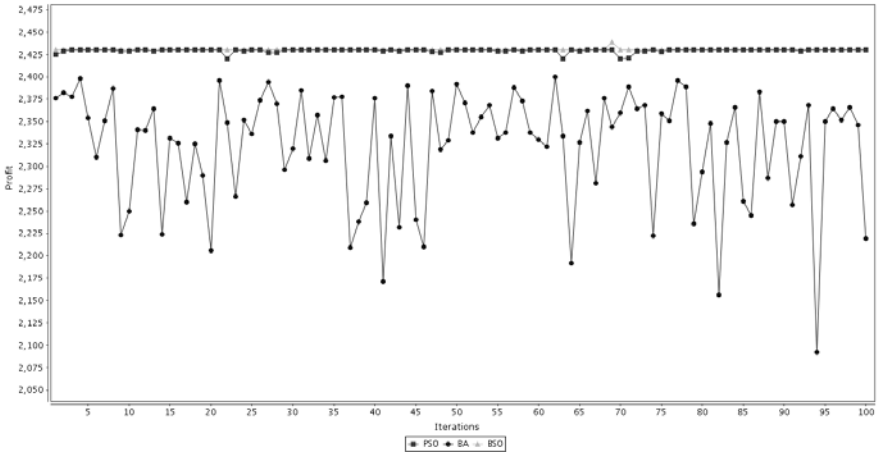


Fig. 4. Graphic behaviors of metaheuristics applied to the *Strongly Correlated Knapsack Problem*, the dots represent the *BA*, the squares the *PSO* and the triangles the *BSO*.

7.4 Inverse Strongly Correlated

To test the Inverse Strongly Correlated instance was define the Knapsack size at 2000. We can see the results of each metaheuristic in the Table 8 and their behavior in the Figure 5. As seen the BA gives the worst Average and Best Comfort, by other hand, the PSO gives a high Average and Best Comfort that the BSO, this gives us a smaller Confidence Interval for the BSO.

7.5 Almost Strongly Correlated

To test the Almost Strongly Correlated instance was define the Knapsack size at 2000. We can see the results of each metaheuristic in the Table 9 and their behavior in the Figure 6. As seen the BA gives the worst Average and Best Comfort, by other hand, the PSO and the BSO give the same Best Comfort but the Standard Deviation is smallest in the case of the BSO and it gives us a smaller Confidence Interval.

Table 8. Results obtained from the *Inverse Strongly Correlated Knapsack Problem*

Metaheuristic	Average Comfort	Average Size	Best Comfort	Best Size	Comfort	Confidence Interval
BA	1622.61	1947.01	1688	1998	43.45	[1609.28, 1635.94]
PSO	1739.92	1997.72	1750	2000	5.38	[1738.27, 1741.57]
BSO	1747.08	1998.98	1750	2000	3.81	[1745.91, 1748.25]

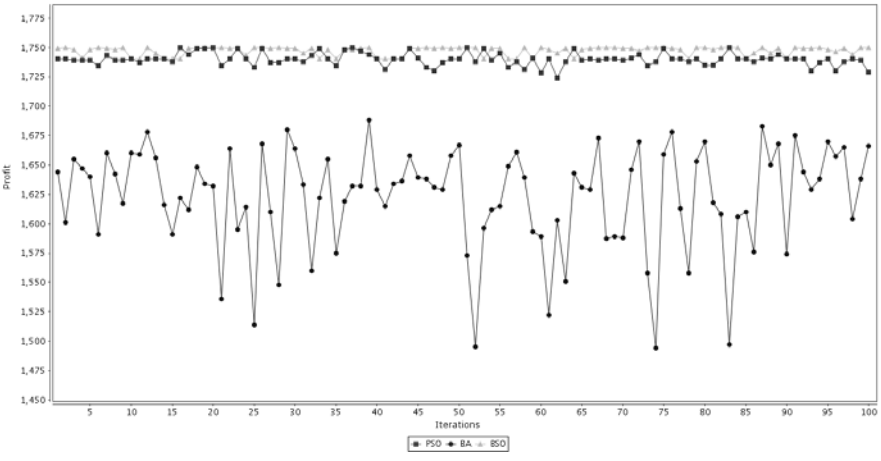


Fig. 5. Graphic behaviors of metaheuristics applied to the *Inverse Strongly Correlated Knapsack Problem*, the dots represent the *BA*, the squares the *PSO* and the triangles the *BSO*.

Table 9. Results obtained from the *Almost Strongly Correlated Knapsack Problem*

Metaheuristic	Average Comfort	Average Size	Best Comfort	Best Size	Comfort	Confidence Interval
BA	2305.62	1955.82	2389	1993	54.21	[2288.98,2322.26]
PSO	2400.27	1998.88	2407	2000	5.11	[2398.7, 2401.84]
BSO	2406.85	1999.85	2407	2000	0.72	[2406.63, 2407.07]

7.6 Subset-sum

To test the Subset-num instance was define the Knapsack size at 1500. We can see the results of each metaheuristic in the Table 10 and their behavior on the Figure 7. As seen the BA gives the worst Average and Best Comfort, on the other hand, the PSO and the BSO give the same results in the entire test.

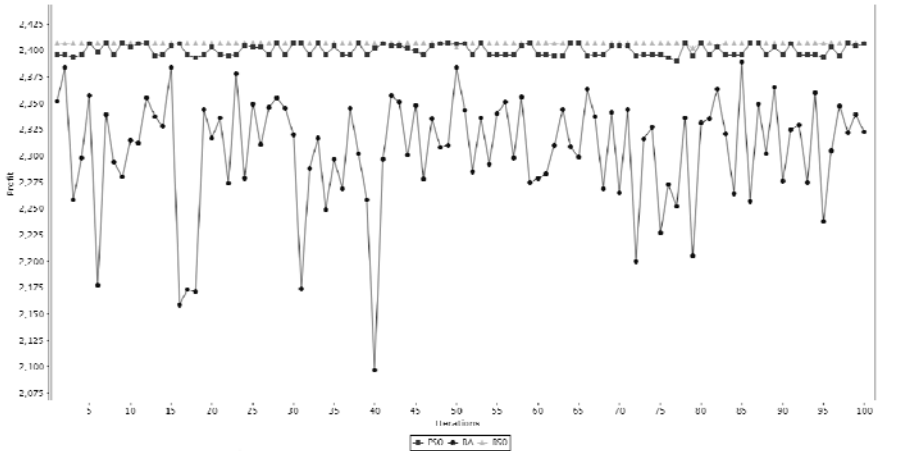


Fig. 6. Graphic behaviors of metaheuristics applied to the *Almost Strongly Correlated Knapsack Problem*, the dots represent the *BA*, the squares the *PSO* and the triangles the *BSO*.

Table 10. Results obtained from the *Subset-sum Knapsack Problem*

Metaheuristic	Average Comfort	Average Size	Best Comfort	Best Size	Comfort	Confidence Interval
BA	1462.35	1462.35	1500	1500	32.44	[1452.39, 1472.31]
PSO	1500	1500	1500	1500	0	[1500, 1500]
BSO	1500	1500	1500	1500	0	[1500, 1500]

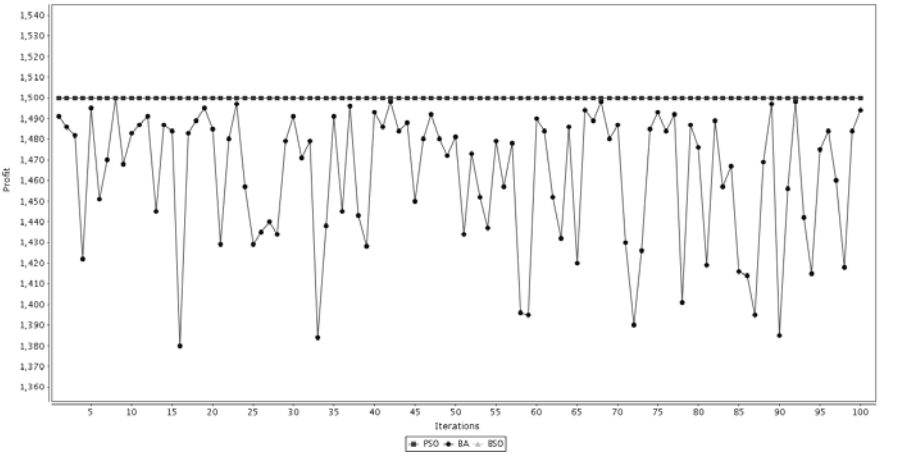


Fig. 7. Graphic behaviors of metaheuristics applied to the *Subset-sum Knapsack Problem*, the dots represent the *BA*, the squares the *PSO* and the triangles the *BSO*.

7.7 Uncorrelated Similar Weight

To test the Uncorrelated Similar Weight instance was define the Knapsack size at 200000. We can see the results of each metaheuristic in the Table 11 and their behavior in the Figure 8. As seen the BA gives the worst Average and Best Comfort, by other way the PSO and the BSO give the same Best Comfort but the Standard Deviation is smallest in the case of the BSO and it give us an smaller Confidence Interval.

Table 11. Results obtained from the *Uncorrelated Similar Weight Knapsack Problem*

Metaheuristic	Average Comfort	Average Size	Best Comfort	Best Size	Comfort	Confidence Interval
BA	1044.72	181807.73	1211	190125	63.17	[1025.33, 1064.11]
PSO	1499.48	190109.69	1500	190110	1.66	[1498.97, 1499.99]
BSO	1499.86	190109.58	1500	190110	1.03	[1499.55, 1500.17]

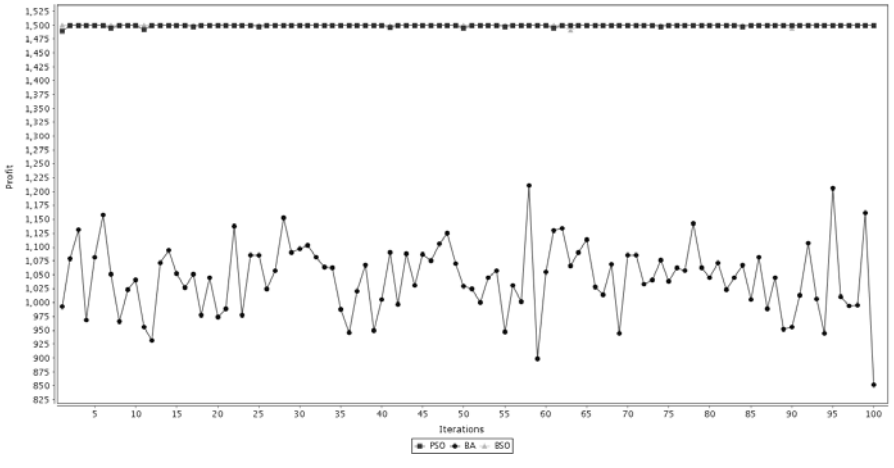


Fig. 8. Graphic behaviors of metaheuristics applied to the *Uncorrelated Similar Weight Knapsack Problem*, the dots represent the *BA*, the squares the *PSO* and the triangles the *BSO*.

8 Conclusions

There are many metaheuristics to solve the Knapsack Problem; in this work we introduce the Bee Swarm Optimization which is a hybrid metaheuristic between the Bee Algorithm and the Particle Swarm Optimization. The experiments were designed with the same parameters for the three metaheuristics to give them the same characteristics in order to be equal between them, and the initial population was generated by a pseudorandom number generator, with this generator the algorithm presented a better performance. After applying the BSO to the different tests

we can see that these metaheuristics give a small Confidence Interval and in most cases its Confidence Interval is better that the PSO's and BA's. In the worst case the Confidence Interval is the same that the PSO's. In the Uncorrelated instances, Subset-sum and Uncorrelated similar weight, the results were similar between PSO and BSO, and in the Weakly correlated instances and Strongly correlated the results were better for BSO than for PSO. Finally in the Inverse Strongly correlated and Almost Strongly Correlated the results were much better for BSO than for PSO. We can see in all the graphics the metaheuristic behavior, and we can observe that the BA is the worst metaheuristic because it always yields a highly variable result, because of this it's Standard Deviation is so high. The PSO yields more consistent results, it's graphs show that this metaheuristic in the most of the cases gives good results. We can say that the BSO is an effective metaheuristic to solve the Knapsack Problem because each time that it's runned, it gives a good solution, and this solution is better that the solutions obtained by the other metaheuristics. Overall the results present a low Standard Deviation and thus a short Confidence Interval.

References

1. Forrest, J.J.H., Kalagnanam, J., Ladanyi, L.: A Column-Generation Approach to the Multiple Knapsack Problem with Color Constraints. INFORMS Journal on Computing, 129–134, INFORMS (2006)
2. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness (1979)
3. Kellerer, H., Pferschy, U., Pisinger, D.: Knapsack Problems. Springer, Berlin (2004)
4. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. IEEE International Conference Neural Networks, vol 4, 1942–1948 (1995)
5. Kennedy, J., Eberhart, R.C.: Swarm Intelligence. Academic Press, EUA (2001)
6. Kiwiel, K.C.: Breakpoint searching algorithms for the continuous quadratic knapsack problem, Math. Program, pp. 473–491. Springer, Heidelberg (2008)
7. Maurice, C.: Particle Swarm Optimization, ISTE Ldt, USA (2006)
8. McDuff-Spears, W.: Using neural networks and genetic algorithms as Heuristics for NP-complete problems, Thesis of Master of Science in Computer Science, George Mason University, Virginia (1989)
9. Pham, D., Ghanbarzadeh, A., Koç, E., Otris, S., Rahim, S., Zaidi, M.: The bee algorithm – a novel tool for complex optimization problems, Intelligent Production Machines and Systems (2006)
10. Pisinger, D.: Core problems in knapsack algorithms. Operation Research 47, 570–575 (1999)
11. Pisinger, D.: Where are the hard knapsack problems? Computers & Operation Research 32, 2271–2282 (2005)
12. Pisinger, D., Rasmussen, A., Sandvik, R.: Solution of Large Quadratic Knapsack Problems Through Aggressive Reduction, INFORMS Journal on Computing, 280–290, INFORMS (2007)
13. Shahriar, A.Z.M., et al.: A multiprocessor based heuristic for multi-dimensional multiple-choice knapsack problem. J Supercomput, 257–280 (2008)

14. Silvano, M., Toth, P.: Knapsack Problem, Algorithms and Computer Implementations. John Wiley and Sons, New York (1990)
15. Yamada, T., Watanabe, K., Kataoka, S.: Algorithms to solve the knapsack constrained maximum spanning tree problem. International Journal of Computer Mathematics, 23–34 (2005)
16. Zemel, E.: An O(n) Algorithm for the linear multiple choice Knapsack problem and related problems. In: Information Processing Problems, pp. 123–128. North Holland, Amsterdam (1984)

An Approach Based on Neural Networks for Gas Lift Optimization

Jose A. Ruz-Hernandez[1], Ruben Salazar-Mendoza[2], Guillermo Jimenez de la C.[2,], Ramon Garcia-Hernandez[1], and Evgen Shelomov[1]

[1] Universidad Autonoma del Carmen, Av. 56 # 4 Esq. Av. Concordia, CP 24180, Cd. del Carmen, Campeche, Mexico
{jruz,rghernandez,eshelomov}@pampano.unacar.mx
[2] Instituto Mexicano del Petroleo
rsalazar@imp.mx

Abstract. Using a model-based optimization, a neural network model is used to compute the optimal values of gas injection rate and oil rate of a gas lift production system. Two cases are analyzed: a) A single well production system and b) A production system composed by two gas lifted wells. For both cases minimizing the objective function of the proposed strategy shows the ability of the neural networks to approximate the behavior of an oil production system and to solve optimization problems when a mathematical model is not available. The results obtained via our approach based on neural networks for gas lift optimization are best than other approaches based on linear programming and non linear programing.

1 Introduction

The daily operation of an oil and gas production system, have many decisions (an element of a solution) to be taken affecting the volumes produced and the cost of production. These decisions are taken at different levels in the organization, but eventually they will reach the physical production system [1]. Fig. 1 gives an overview of a physical illustrates a gas lift production system. For in such oil production systems, the decisions are related to find the lift gas rate for each well giving the maximum total oil production rate at very instance of time every time instant.

An objective function is a single-valued and well-defined mathematical function mapping the values of the decision variables into a performance measure. Examples of such performance measures are the total oil production rate, net present value (profit), or the recovery of the reservoir. In the effort toward better performance of the production system, a question to be answered is which decisions are better to maximize or minimize the objective function. In the process of making good decisions, information about the production system is used. This information may include the physical properties such as pipe diameters and lengths, or it may include measurements from the production system.

P. Melin et al. (Eds.): Soft Comp. for Recogn. Based on Biometrics, SCI 312, pp. 207–224.
springerlink.com

To support making good decisions, well models may be used to develop the production plans. Typically, well test are performed to determine the gas to oil ratio, water cut, and production rates of each individual well. Well test are performed by routing a well to a dedicated separator. This separator will separate three phases, and a rate transmitter is connected to the outlet for each phase. The well model is update using the measurements taken during a test. Fluid sampling may be used to obtain the fluid composition including the water cut.

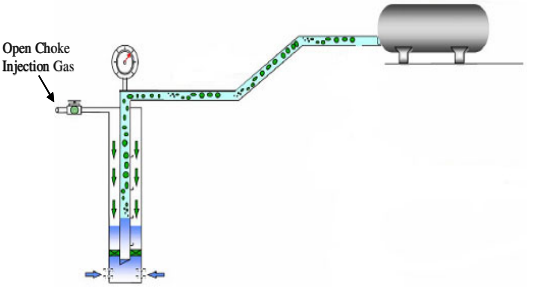


Fig. 1. Gas lift production system of a single well.

The objective of gas lift is to increase oil production or allow nonrateing wells to rate by reducing the hydrostatic head of the fluid column in the well [2]. By injecting gas into the tubing, the density of the wellbore fluid decrease; thus, the pressure-drop component resulting from gravity is reduced. However, the gas lift also gives a larger pressure-drop component resulting from friction, giving some optimal gas lift rate for the well. Usually, the available lift-gas for a group of wells is less than the sum of the individual optimum lift-gas rates for each well. The gas-lift optimization problem is to find the lift gas rates for each well giving the maximum total oil production rate subject to a gas lift processing capacity constraint and possibly other operational and processing constraints.

The objective function and the constraints define the optimization problem, or mathematical program. A solver is used to find an optimal solution, and the solver should be chosen using information about the problem structure. Many local solvers use local information about the neighborhood of a current solution to find a step that improves the objective function and, preferably, maintains the feasibility of the current solution. Examples of the local information are the derivative of the objective function and the constraints with respect to the decision variables evaluated at the current solution. Examples of solvers using derivative information are the Active Set Quadratic Programming, Sequential Quadratic Programming (SQP), and Sequential Linear Programming (SLP) methods [3]. The algorithms such as SQP and SLP would require an infinite number of iterations to find a local optimum; however, a termination criterion is used to terminate in finite time.

By another hand, the well models are used to generate the gas lift performance curves for each well. A gas lift performance curve relates the lift-gas rate injected into the well to the oil production rate from the well, and it may typically be used for finding the optimal gas lift rates for the wells [4].

The performance curves are used in the equal slope method proposed in [5]. The equal slope method established a way of finding optimum lift-gas rates. The name exists because of the characteristic of the optimum solutions where the effects of an infinitesimal increase of the lift-gas rate would be the same for all wells.

The Linear programming (LP) it is a mathematical procedure to determine the optimal allocation of limited resources. The LP is a procedure that finds its practical application in almost all the facets of the businesses, from the publicity to the planning of the production. Problems of transport, distribution, and global planning of the production are the most common objects of the LP analysis. The oil industry seems to be the most frequent user of the LP. A manager of data processing of an important oil company recently computed that from 5% to 10% of the time of computer science processing of the company she is destined to the processing of models of LP and similar [6].

Sequential Quadratic programming (SQP) is a method to solve an optimization problem with quadratic programming. A problem of quadratic programming is a special case of nonlinear programming, where the objective function is quadratic and the restrictions are linear [7].

In this paper, a model-based optimization via neural networks is developed and it is used to compute the optimal values of gas injection rate and oil rate of a gas lift production system. Two cases were analyzed: a) A single well production system and b) A production system composed by two gas lifted wells. For both cases minimizing the objective function the proposed strategy shows the ability of the neural networks to approximate the behavior of an oil production system and to solve optimization problems when a mathematical model is not available.

2 Strategy of Optimization Based on a Neural Network

In order to solve the gas-lift optimization problem, an optimization procedure based on a neural network was developed. The strategy selected is illustrated in Fig. 2. The two components are the neural network used to approximate the gas lift performance curve and the objective function to satisfy a performance index.

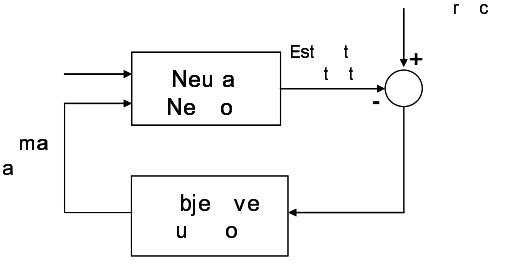


Fig. 2. Strategy of optimization based on a neural network.

The expression to define the neural network used in our strategy is given by

$$Q_{est}(k) = g[\varphi], \tag{1}$$

Where $Q_{\text{est}}(k)$ are estimates rates; and $\varphi = [Q_{\text{o}}(k\text{-}1), Q_{\text{iny}}(k\text{-}1)]$ are input vectors with $k = 1, 2, ..., N$ pairs of data.

A Multi-Layer Perceptron (MLP) for the function g is selected. This type of neural network has flexibility and excellent properties to approximate functions. The use of this kind of networks to approximate functions and carry out identification process goes back to more than one decade [8].

The following equation is used to determine the structure of a MLP with a single hidden or intermediate layer, a neuron in the output layer with function of linear activation and M hidden neurons is used

$$F(x_1,...,x_P) = \sum_{i=1}^{M} \alpha_i g(\sum_{j=1}^{p} w_{ij} x_j - \theta_i), \tag{2}$$

Most of authors and researchers on neural networks have concluded that the neuronal networks of two layers using hyperbolic tangent activation function or sigmoid activation function in the intermediate layer are universal approximations [9], [10], [11], [12] and [13].

The neural networks multilayer can be designed and trained with the aids of the Neural Networks Toolbox for use with Matlab[1]. The training of the multilayer neural networks was performed with aid of Matlab. The training data are collected from a gas lift performance curve obtained from PROSPER[2] [14]. In this case:

$$Q_{est}(k) = g[Q_o(k-1), Q_{iny}(k-1)], \tag{3}$$

where

$Q_o(k) \in \Re$ is the oil production rate (STB/day).
$Q_{iny}(k) \in \Re$ is the lift-gas rate injected into the well (MMscf/day).
$Q_{\text{est}}(k) \in \Re$ is the estimated produced oil rate (STB/day).

In this application, the approximation procedure is done using a neural network multilayer perceptron with three layers. The hidden layer has neurons using sigmoid activation function and the output layer has a unique neuron with linear activation function. Different MLPs are trained by means of Levenberg-Marquardt Algorithm, which uses the criterion of middle square error to update the neural network weights. The corresponding MLP is displayed in Fig. 3.

The expression to define the objective function used in our strategy of optimization is given by:

[1] Matlab is a registered trademark from Mathworks Inc.
[2] PROSPER is a registered trademark from Petroleum Experts.

$$J(u_k) = \alpha_k [Q_{est}(k) - ref(k)]^2 + \beta_k (\Delta u_k)^2, \quad (4)$$

where:

$Q_{est}(k)$: Neural network approximation of the produced oil rate (STB/day).

$ref(k)$: Reference of produced oil rate (STB/day).

u_k : Update injection rate (MMscf/day).

$\Delta u_k = u_k - u_{k-1}$

α_k : Weight of the quadratic error of the produced oil rate.

β_k : Weight of the control effort.

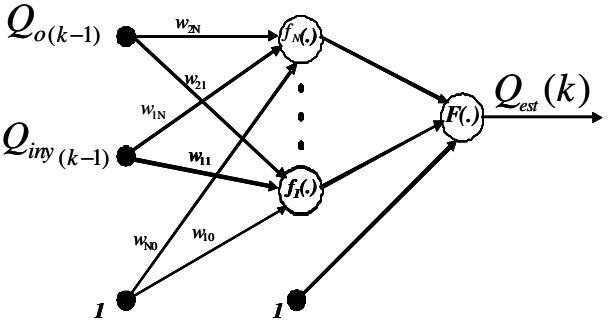


Fig. 3. Neural network MLP.

In order to minimize the objective function, the optimal gas rate for injection is computed varying the gas rate according to (5) in within the interval given in (6) and evaluating each value into the objective function in such a way that by comparison between an evaluation and another one the value of gas rate is located to minimize (4) [15]. The variable gas rate will be denoted as u_{rab}.

The number of variations that will suffer u_{rab} in order to cover the interval given in (6) with increases with size Δ it will be $nit = (u_{max}/0.01)+1$, remembering that the considered precision which it has the instrument is of 1% after the point decimal. Fig. 4, display the corresponding flow chart for the optimization procedure.

In the flow chart, the following variables and constants are used:

$u_{\max}$: Maximum optimal gas rate of injection (MMscf/day).

nit: Number of variations of injection rate.

u_n : Initial gas injection rate (MMscf/day).

min: Minimum value of the used objective function in the optimization procedure.

$J(u_{rab})$: Value that acquires the objective function when an iteration is carried out.

iter: Counter of iterations in the optimization routine.

u_{opt}: Register to store to the present value of the rate optimal gas of injection (MMscf/day).

P_{optim}: Register to store the present value of rate of oil corresponding to the optimal gas rate of injection.

Q: Produced oil rate evaluated at time u_{opt} in the considered model (STB/day).

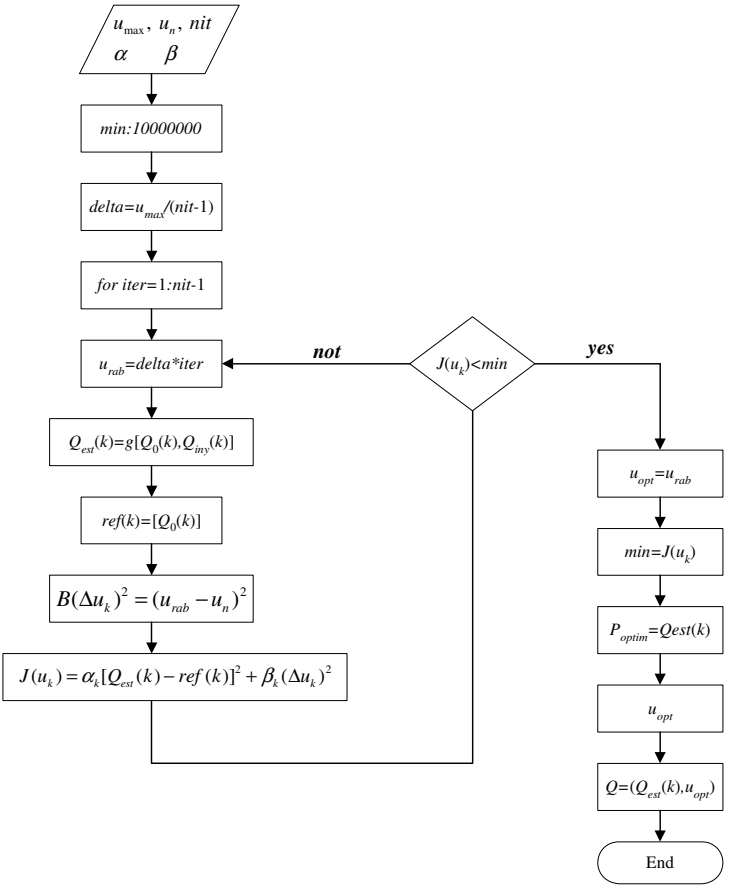


Fig. 4. Flow chart corresponding to the optimization procedure.

To solve the numerical optimization problem, the following main considerations are taken into account:

- Due to the precision of a measuring instrument used in this application, here is considered a percentage unit of nominal gas rate for injection. The gas passage of injection used in the optimization to determine the optimal gas rate of injection will be given by:

$$\Delta Q_{iny}(k) = 1\% = 0.01, \tag{5}$$

- The rate of gas injection will be a value in the following interval (MMscf/day):

$$0.0 \le Q_{iny}(k) \le u_{max}, \tag{6}$$

- There is not mathematical model of the process; however, experimental data can be obtained from a well simulator.
- A neural network model of the process is available, which can be used to approximate its behavior and to construct the objective function.
- Likewise, it is important to define the number of variations of the rate gas for injection (*nit*). This number is obtained as

$$nit = \frac{u_{max}}{\Delta} + 1 \tag{7}$$

where:

u_{max}: maximum limit of gas injection
Δ : 0.01

Starting off of the experimental data of the production curve we can define a vectorial function in the time, $\phi(k)$, whose elements are the values of gas reference of injection and oil produced at the moment *k*.

Taking into account the previous considerations, we can formulate the following optimization problem:

$$\min_{v(k)} J(v(k)) \tag{8}$$

$$J(v(k)) = \| \phi(k) - \phi_{est}(k) \| \tag{9}$$

where:

$v(k) = [Q_o(k), Q_{iny}(k)]^T \in \Re$.

Now, we can write the equation (9) as follows:

$$J(v(k)) = \left\{ \alpha_k [Q_o(k) - Q_{est}(k)]^2 + \beta_k [Q_{iny}(k) - Q_{iny}(k-1)]^2 \right\} \tag{10}$$

3 Results and Discussions

In this paper two cases are considered; in the first one, a neural network model approximates the produced oil rate for a single well and then the second one a neural network model approximates the produced oil rate for a production system of hydrocarbons constituted by two wells. The obtained results for each case are presented as follows.

3.1 First Case: Produced Oil Rate by a Single Well

The corresponding oil production system is illustrated in Fig. 1 and the following conditions are taken into account. Input vector $\varphi(k)$ is generated considering a pressure in the head of the well equal to 12 kg/cm^2, and parameters like the static pressure in the reservoir, lengths and diameters of the tubing and lines, chokes and others pipe components are also considered.

In Fig. 5a are shown the training data which are collected by using PROSPER, 600 pairs from samples experimental gas injection data and produced oil data are used and taken in a produced gas interval of $0.0 \leq Q_{iny} \leq 6.0$ MMscf/day.

In this case, neural network is trained to approximate the produced oil rate for a single well. The best trained neural network is described in Table 1.ïIn Figure 5b are shown the values for the validation of the model selected from the trained neural network for the data. The oil produced rate has units STB/day (Standard Barrels per day). In Fig. 6 are shown the obtained results using neural network as described in Table 1. The approximation achieved via neural network is compared with produced oil rate data from a single well. Fig. 7 displays this approximation.

Table 1. Neural network architecture and obtained errors, case 1.

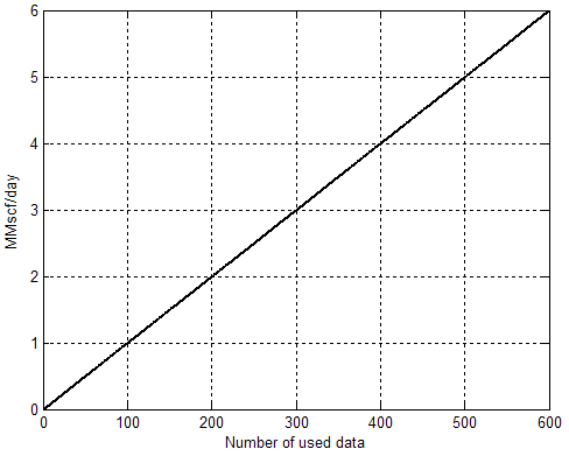
Number of neurons		error	error (%)
Intermedia layer	Output layer		
30	1	0.0119	0.0004

3.2 Second Case: Produced Oil Rate by a Production System Based on Two Wells

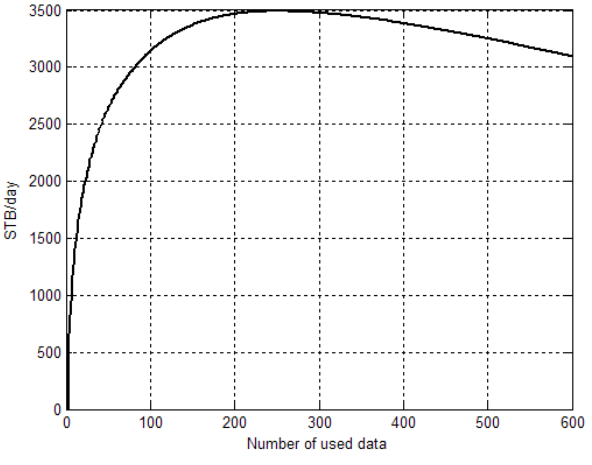
In Fig. 8 is shown an oil production system of two wells. Input vector $\varphi(k)$ is generated setting a pressure in the head of each well of 12 and 14 kg/cm^2, respectively.

In Fig. 9a are shown the training data, which consider 600 pairs from samples injection gas experimental data and produced oil data are used and taken in a produced gas interval of $0.0 \leq Q_{iny} \leq 12.0$ MMscf/day. In this case, neural network is trained to approximate the produced oil rate for a production system of two wells.

As described in previous case, after the training and validation, a neural network is selected considering its performance and complexity. The best trained neural network is described in Table 2. In Fig. 9b are shown the values for the validation the model selected from the trained neural network for the data of oil produced rate. In Fig. 10 is shown the obtained results using neural network as described in Table 2.

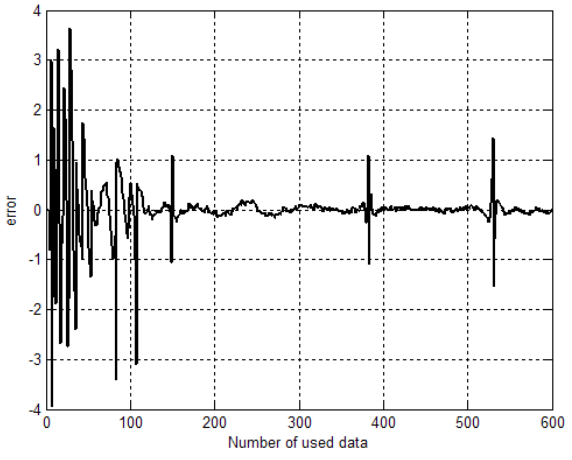


(5a)

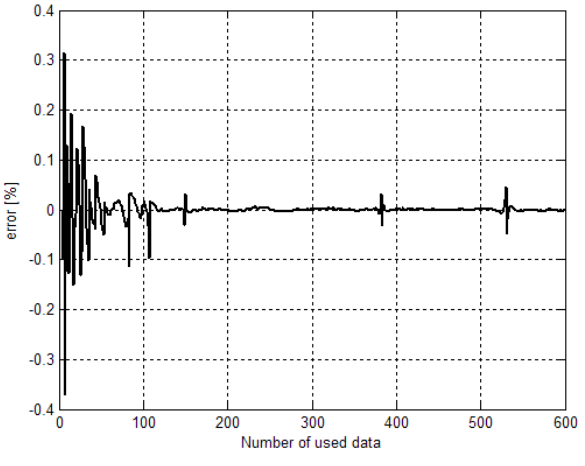


(5b)

Fig. 5. Experimental data (a) gas injection rate (b) oil production rate.



(6a)



(6b)

Fig. 6. Result for case 1, (a) error, (b) percentage of error.

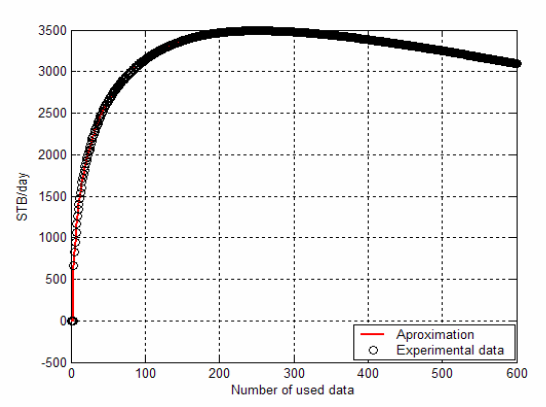


Fig. 7. Comparison between the approximations achieved using the neural network model (solid line) and the experimental data (round marks) from a single well system.

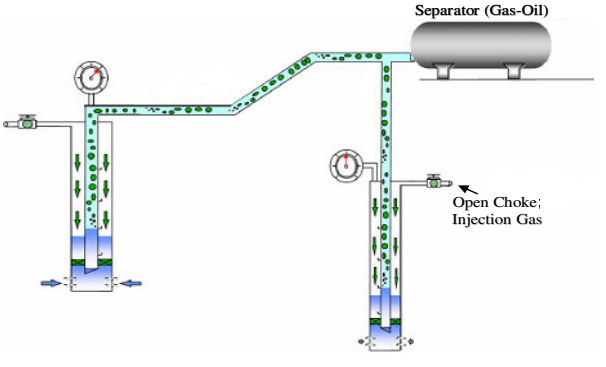
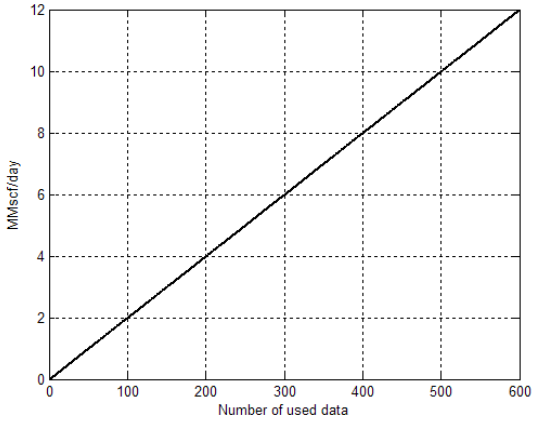


Fig. 8. Production system composed by two wells.

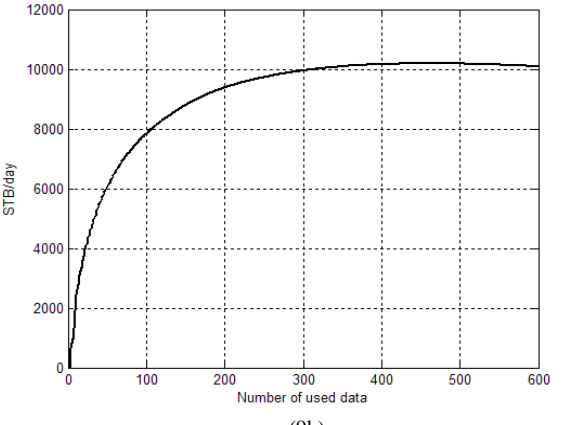
The approach of the compared neural network with the data of produced oil rate for the system composed by two wells is showed in figure 11.

Once that both systems are approximated using neural networks, the following stage establishes and solves the optimization problem.

The optimization procedure is used to determine the optimal gas injection rate for a single well (case 1). When the iteration 93, is carried out the obtained optimal gas injection rate is 0.93 MMscf/day (See Fig. 12). This rate is evaluated in



(9a)

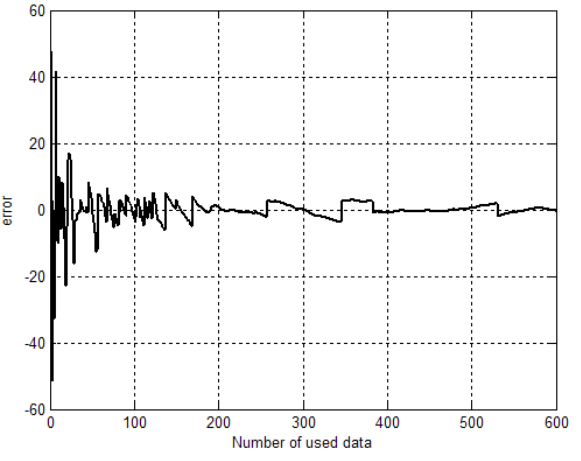


(9b)

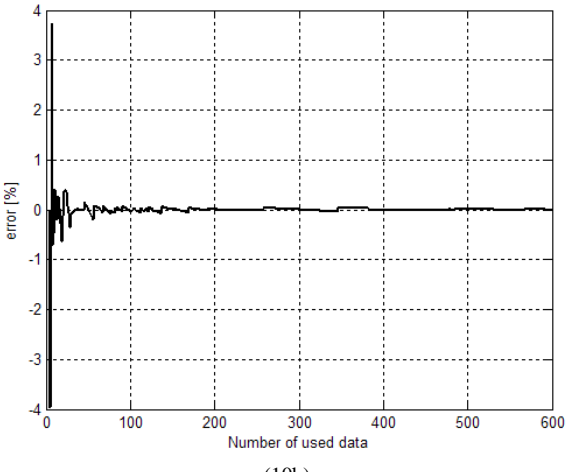
Fig. 9. Experimental data (a) gas injection rate (b) oil production rate.

Table 2. Neural network architecture and obtained errors, case 2

Number of neurons		error	error (%)
Intermedia layer	Output layer		
30	1	-0.3702	-0.0037



(10a)



(10b)

Fig. 10. Result for case 2, (g) error, (h) percentage of error.

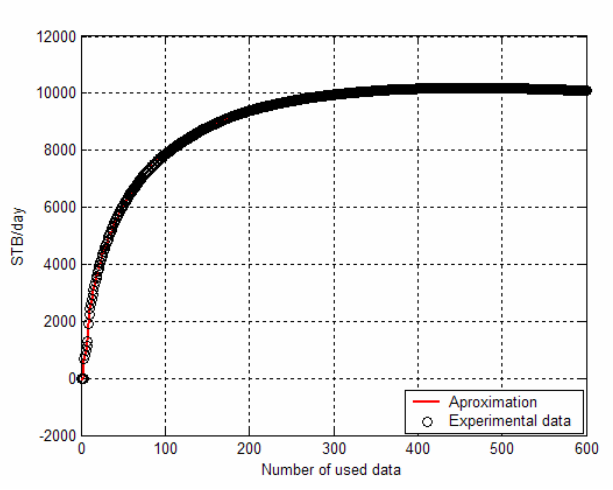


Fig. 11. Comparison between the approach obtained with the neural model (solid line) and the experimental data (round marks) for the two wells system.

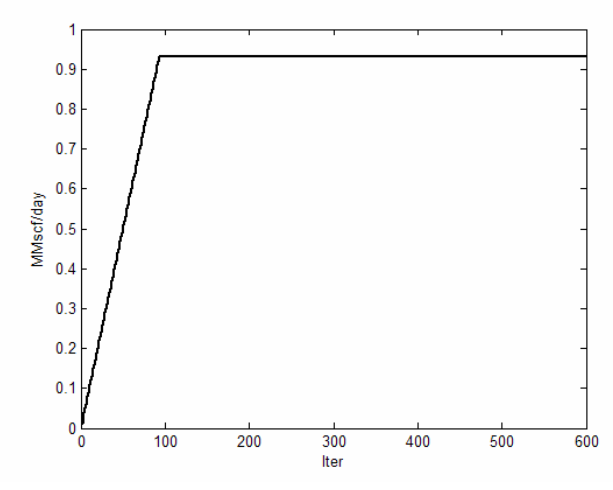


Fig. 12. Optimal gas injection case 1.

the neural network as described in table 1 to obtain an oil production of 3102.8 STB/day. The Fig. 13 shows the beahavior objective function when is minimized.

A similar optimization procedure is used to determine the optimal gas injection rate for a two wells system (case 2). When the iteration 349, is carried out the obtained optimal gas injection rate is 6.98 MMscf/day (See Fig. 14). This rate is

evaluated in our neural network as described in table 2 obtaining an oil production of 10087 STB/day. In Fig. 15 is shown the minimized objective function.

3.3 Comparative Analysis

Obtained results by using this approach are compared by using the software GAP[3] which is employed commonly to determine the optimal gas injection for a two

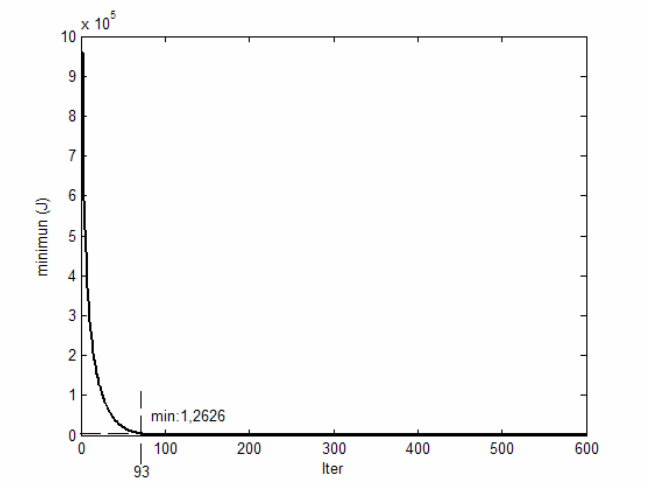


Fig. 13. Minimized objective function.

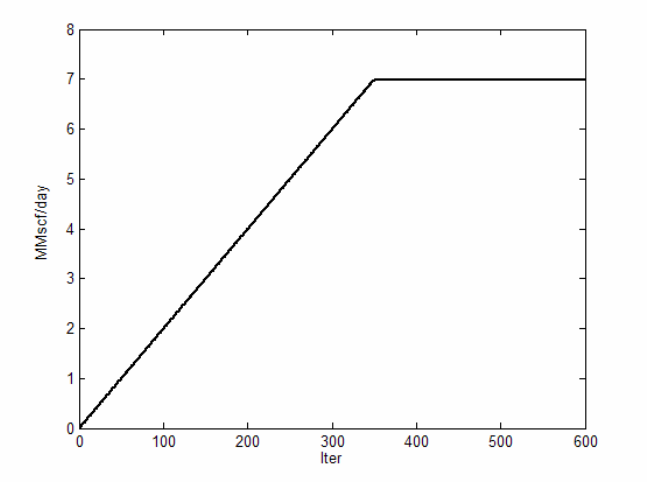


Fig. 14. Optimal gas injection case 2.

[3] GAP is a registered trademark from Petroleum Experts.

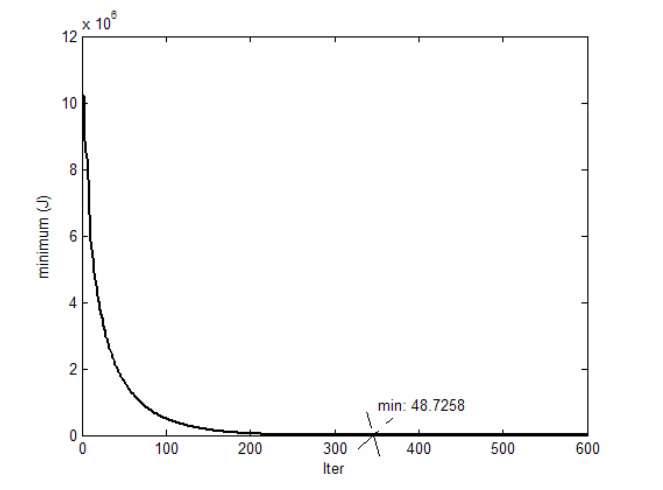


Fig. 15. Minimized objective function.

Table 3. Comparative Analysis

Optimization via Neural Networks (NN)	**System**	**Gas injection [MMpcd]**	**Oil production [BPD]**
	Well 1	0.93	3102.80
	Well 2	5.95	6990.70
	Total	**6.88**	**10093.50**
Optimization via Linear Programming (LP)	**System**	**Gas injection [MMpcd]**	**Oil production [BPD]**
	Well 1	2.78	3250.80
	Well 2	4.20	5727.18
	Total	**6.98**	**8977.99**
Optimization via Non Linear Programming (NLP)	**System**	**Gas injection [MMpcd]**	**Oil production [BPD]**
	Well 1	2.76	3478.63
	Well 2	4.18	6435.65
	Total	**6.94**	**9914.28**
Diference		**BPD**	
RN vs LP		12.43 %	
RN vs NLP		1.81 %	
LP vs NLP		10.43 %	

wells system, [16]. GAP is a multiphase oil and gas optimizer tool that models the surface gathering network of field production systems. When linked with the well models of PROSPER a full field production optimization and forecast can be achieved. Both software programs allow us to carry out the required optimization by means of Linear Programing and Non Linear Programming (via Quadratic Sequential Programming). The comparison is displayed in the Table 3. Is easy to see that obtained results by means of our proposed approach it is possible to obtain a major oil production by using minor amount of gas lift injection, [17].

4 Conclusions

In this paper a computational intelligence strategy based on neural networks is proposed to optimize the required gas rate to inject in a production system of hydrocarbons. Once that this optimal rate is determined and applied to the production system, the obtained simulation results are closed to the produced oil reference. Furthermore, the proposed strategy shows the ability of the neural networks to approximate the behavior of production systems and to solve optimization problems when a mathematical model is not available. It is just to mention that this approach can be used as an aid mechanism for production engineers.

Acknowledgment

The authors thank support of Universidad Autonoma del Carmen (UNACAR), Mexico on project PR/131/2008.

References

1. Bieker, H.P., Slupphaug, O., Johansen, T.A.: Real-Time Production Optimization of Oil and Gas Production Systems: A Technology Survey. SPE Production & Operations Journal 22(4), 382–391 (2007)
2. Vasper, A.: Auto, Natural, or In-Situ Gas-Lift Systems Explained. SPE Production & Operations Journal 23(1), 75–80 (2008)
3. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, New York (1999)
4. Fang, W.Y., Lo, K.K.: A Generalized Well-Management Scheme for Reservoir Simulation. SPE Reservoir Engineering Journal 11(2), 116–120 (1996)
5. Kanu, E.P., Mach, J., Brown, K.E.: Economic Approach to Oil Production and Gas Allocation in Continuous Gas Lift. Journal of Petroleum Technology 33(10), 1887–1892 (1981)
6. Gill, P.E., Murray, W., Wright, M.H.: Practical Optimization. Academic Press, London (1981)
7. Hossein, A.: Modelos Deterministas: Optimización Lineal, http://www.service.org/sites/home.ubalt.edu
8. Narendra, K., Parthasarathy, K.: Identification and control of dynamical systems using neural networks. IEEE Transactions on Neural Networks 1(1), 4–27 (1990)

9. Hornik, K., Stinchcombe, M., White, H.: Multilayer Feedforward Networks are Universal Aproximators. Neural Networks 2(5) (1989)
10. Norgaard, M., Ravn, O., Poulsen, N.K., Hansen, L.K.: Neural Networks for Modelling and Control of Dynamic Systems. Springer, London (2000)
11. Sanchez, E.N., Alanís, A.Y.: Redes Neuronales.: Conceptos Fundamentales y Aplicaciones al Control Automático. Prentice Hall-Pearson Education, Englewood Cliffs (2006)
12. Corchado, J.M., Díaz, F.: Redes Neuronales Artificiales. Un enfoque Practico. Universidad de Vigo., España (2000)
13. Cybenko, G.: Aproximation by superpositions of sigmoidal function. Math. Control, Sygnals and Syste. 2(4) (1989)
14. Prosper. Petroleum Experts Ltd. User Guide IPM V5.3Build #173 (June 6, 2006)
15. Ruz-Hernandez Jose, A., Suarez, D.A., Shelomov, E., Villavicencio, A.: Predictive Control Based on an Auto-Regressive Neuro-Fuzzy Model Applied to the Steam Generator Startup Process at a Fossil Power Plant. Computación y Sistemas 6(3), 204–212 (2003)
16. GAP Petroleum Experts Ltd. User Guide IPM v 6.1 (December 2006)
17. Jimenez de la, C.G., Ruz-Hernandez, J.A., Salazar-Mendoza, R.: Obtaining an Optimal Gas Injection Rate for an Oil Production System via Neural Networks. In: Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN), Hong Kong (2008)

A New Evolutionary Method with Particle Swarm Optimization and Genetic Algorithms Using Fuzzy Systems to Dynamically Parameter Adaptation

Fevrier Valdez and Patricia Melin

Tijuana Institute of Tech., Tijuana BC. México
pmelin@tectijuana.mx

Abstract. We describe in this paper a new hybrid approach for mathematical function optimization combining Particle Swarm Optimization (PSO) and Genetic Algorithms (GAs) using Fuzzy Logic for parameter adaptation and integrate the results. The new evolutionary method combines the advantages of PSO and GA to give us an improved FPSO+FGA hybrid method. Fuzzy Logic is used to combine the results of the PSO and GA in the best way possible. The new hybrid FPSO+FGA approach is compared with the PSO and GA methods with a set of benchmark mathematical functions. The new hybrid FPSO+FGA method is shown to be superior than the individual evolutionary methods on the set of benchmark functions.

Keywords: FPSO, FGA, GA.

1 Introduction

We describe in this paper a new evolutionary method combining PSO and GA, to give us an improved FPSO+FGA hybrid method. We apply the hybrid method to mathematical function optimization to validate the new approach. Also in this paper the application of a Genetic Algorithm (GA) [1] and Particle Swarm Optimization (PSO) [2] for the optimization of mathematical functions is considered. In this case, we are using a set of mathematical benchmark functions [4][5][13] to compare the optimization results between a GA, PSO and FPSO+FGA.

The main motivation of this method is to combine the characteristics of a GA and PSO. We are using several fuzzy systems to perform dynamical parameter adaptation. For decision making between the methods depending on the results that we are generating we are using one fuzzy system. The fuzzy system is used to decide and combine the outputs of both the GA and PSO, in this way obtaining the best solution to the optimization problem. Also, there are another two fuzzy systems to parameter adaptation in GA and PSO. The main goal of the fuzzy system is to evaluate the outputs of the GA and PSO in each generation and change if is

P. Melin et al. (Eds.): Soft Comp. for Recogn. Based on Biometrics, SCI 312, pp. 225–243.
springerlink.com

necessary some important parameters. The criterion for stopping the method is the maximum number of generations.

The paper is organized as follows: in section 2 a description about Genetic Algorithms for optimization problems is presented, in section 3 the Particle Swarm Optimization is presented, in section 4 the proposed method FPSO+FGA and the fuzzy systems are described, in section 5 we can appreciate the mathematical functions that were used for this research, in section 6 the simulations results are described, in section 7 we can appreciate a comparison between GA, PSO and FPSO+FGA, in section 8 the conclusions obtained after the study of the proposed evolutionary computing methods are presented.

2 Genetic Algorithm for Optimization

Holland, from the University of Michigan initiated his work on genetic algorithms at the beginning of the 1960s. His first achievement was the publication of *Adaptation in Natural and Artificial System* [7] in 1975.

He had two goals in mind: to improve the understanding of natural adaptation process, and to design artificial systems having properties similar to natural systems [8].

The basic idea is as follows: the genetic pool of a given population potentially contains the solution, or a better solution, to a given adaptive problem. This solution is not "active" because the genetic combination on which it relies is split between several subjects. Only the association of different genomes can lead to the solution.

Holland's method is especially effective because it not only considers the role of mutation, but it also uses genetic recombination, (crossover) [9]. The crossover of partial solutions greatly improves the capability of the algorithm to approach, and eventually find, the optimal solution.

The essence of the GA in both theoretical and practical domains has been well demonstrated [1]. The concept of applying a GA to solve engineering problems is feasible and sound. However, despite the distinct advantages of a GA for solving complicated, constrained and multiobjective functions where other techniques may have failed, the full power of the GA in application is yet to be exploited [12] [14].

In figure 1 we show the reproduction cycle of the Genetic Algorithm.

The Simple Genetic Algorithm can be expressed in pseudo code with the following cycle:

```
1. Generate the initial population of individuals aleatorily P(0).
2. While (number _ generations <= maximum _ numbers _ generations)
     Do:
       {
         Evaluation;
         Selection;
         Reproduction;
         Generation ++;
       }
3. Show results
   4. End
```

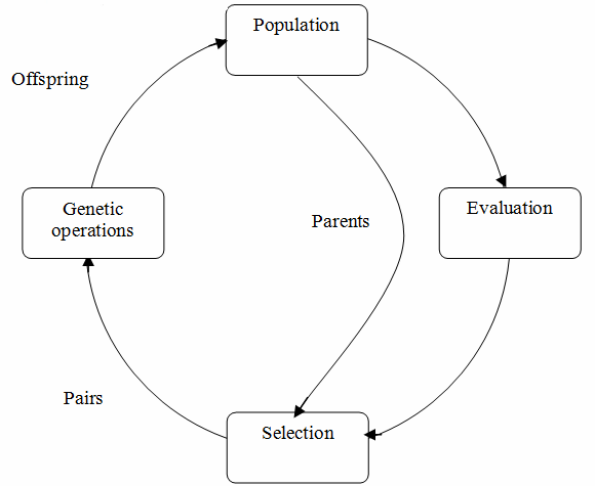


Fig. 1. The Reproduction cycle

3 Particle Swarm Optimization

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Eberhart and Kennedy in 1995, inspired by the social behavior of bird flocking or fish schooling [3].

PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA) [6]. The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike the GA, the PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles [10].

Each particle keeps track of its coordinates in the problem space, which are associated with the best solution (fitness) it has achieved so far (The fitness value is also stored). This value is called *pbest*. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the neighbors of the particle. This location is called *lbest*. When a particle takes all the population as its topological neighbors, the best value is a global best and is called *gbest*.

The particle swarm optimization concept consists of, at each time step, changing the velocity of (accelerating) each particle toward its p*best* and *lbest* locations (local version of PSO). Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward *pbest* and *lbest* locations.

In the past several years, PSO has been successfully applied in many research and application areas. It is demonstrated that PSO gets better results in a faster, cheaper way compared with other methods [11].

Another reason that PSO is attractive is that there are few parameters to adjust. One version, with slight variations, works well in a wide variety of applications.

Particle swarm optimization has been used for approaches that can be used across a wide range of applications, as well as for specific applications focused on a specific requirement.

The pseudo code of the PSO is as follows

For each particle
Initialize particle
End
Do
For each particle
Calculate fitness value
If the fitness value is better than the best fitness value (pBest) in history
set current value as the new pBest
End
Choose the particle with the best fitness value of all the particles as the gBest
For each particle
Calculate particle velocity
Update particle position
End
While maximum iterations or minimum error criteria is not attained

4 Full Model of FPSO+FGA

The general approach of the proposed method PSO+GA can be seen in figure 2. The method can be described as follows:

It receives a mathematical function to be optimized

It evaluates the role of both GA and PSO.

A main fuzzy system is responsible for receiving values resulting from step 2.

The main fuzzy system decides which method to take (GA or PSO)

After, another fuzzy system receives the Error and DError as inputs to evaluates if is necessary change the parameters in GA or PSO.

There are 3 fuzzy systems. One is for decision making (is called main fuzzy), the second one is for changing parameters of the GA (is called fuzzyga) in this case change the value of crossover (k1) and mutation (k2) and the third fuzzy system is used to change parameters of the PSO (is called fuzzypso) in this case change the value of social acceleration (c1) and cognitive acceleration (c2).

The main fuzzy system decides in the final step the optimum value for the function introduced in step 1.

Repeat the above steps until the termination criterion of the algorithm is met.

The basic idea of the FPSO+FGA scheme is to combine the advantage of the individual methods using a fuzzy system for decision making and the others two fuzzy systems to improve the parameters of the GA and PSO when is necessary.

The basic idea of the FPSO+FGA scheme is to combine the advantage of the individual methods using a fuzzy system for decision making and the others two fuzzy systems to improve the parameters of the FGA and FPSO when is necessary.

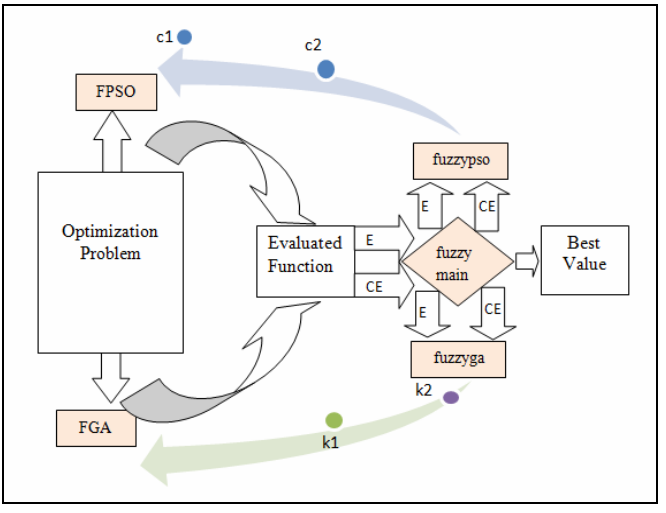


Fig. 2. The FPSO+FGA scheme

As can be seen in the proposed hybrid FPSO+FGA method, it is the internal fuzzy system structure, which has the primary function of receiving as inputs (Error and DError) the results of the FGA and FPSO outputs. The fuzzy system is responsible for integrating and decides which are the best results being generated at run time of the FPSO+FGA. It is also responsible for selecting and sending the problem to the "fuzzypso" fuzzy system when the FPSO is activated or to the "fuzzyga" fuzzy system when FGA is activated. Also activating or temporarily stopping depending on the results being generated. Figure 3 shows the membership functions of the main fuzzy system that is implemented in this method. The fuzzy system is of Mamdani type because it is more common in this type of fuzzy control and the defuzzification method is the centroid. In this case, we are using this type of defuzzification because in other papers we have achieved good results [4]. The membership functions are of triangular form in the inputs and outputs as is shown in figure 3. Also, the membership functions were chosen of triangular form based on past experiences in this type of fuzzy control. The fuzzy system consists of 9 rules. For example, one rule is if error is P and DError is P then best value is P (view figure 4). Figure 5 shows the fuzzy system rule viewer. Figure 6 shows the surface corresponding to this fuzzy system. The other two fuzzy systems are similar to the main fuzzy system.

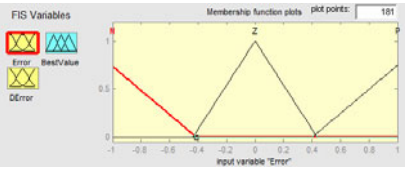


Fig. 3. Fuzzy system membership functions

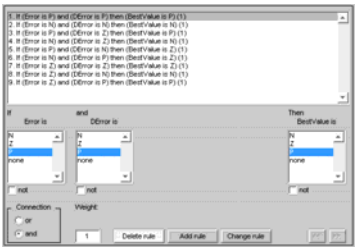


Fig. 4. Fuzzy system rules

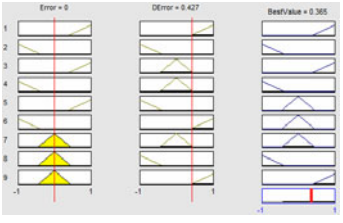


Fig. 5. Fuzzy system rules viewer

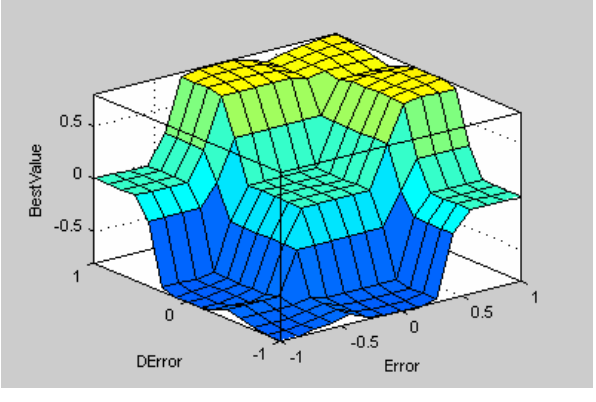


Fig. 6. Surface of the main fuzzy system

4.1 FPSO (Fuzzy Particle Swarm Optimization)

This section presents a detailed description of the FPSO model. The classical representation scheme for GAs is binary vectors of fixed length. In the case of an n_x – dimensional search space, each individual consists of n_x variables with each variable encoded as a binary string.

The swarm is typically modeled by particles in multidimensional space that have a position and a velocity. These particles fly through hyperspace (i.e., R^n) and have two essential reasoning capabilities: their memory of their own best

position and knowledge of the global or their neighborhood's best. In figure 7 we can see a simulation of F3 function. In a minimization optimization problem, "best" simply meaning the position with the smallest objective value. Members of a swarm communicate good positions to each other and adjust their own position and velocity based on these good positions. So a particle has the following information to make a suitable change in its position and velocity:

A global best that is known to all and immediately updated when a new best position is found by any particle in the swarm.

The neighborhood best that the particle obtains by communicating with a subset of the swarm.

The local best, which is the best solution that the particle has seen.

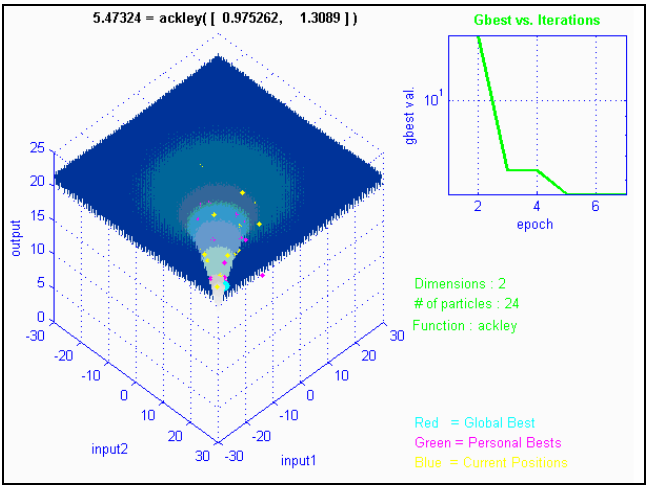


Fig. 7. Simulation of F3 function with FPSO

In this case, the social information is the best position found by the swarm, referred as $\hat{y}(t)$. For gbest FPSO, the velocity of particle i is calculated as

$$v_{ij}(t+1) = vi_j(t) + c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t)[\hat{y}_j(t) - x_{ij}(t)] \quad (1)$$

Where $vi_j(t)$ is the velocity of particle i in dimension j = 1,…,n_x at time step t, $xi_j(t)$ is the position of particle i in dimension j at time step t, C_1 and C_2 represents the cognitive and social acceleration. In this case, these values are fuzzy because they are changing dynamically when the FPSO is running, and $r_{1j}(t)$, r_{2j} ~U(0,1) are random values in the range [0,1].

4.2 FGA (Fuzzy Genetic Algorithm)

This section presents a detailed description of the FGA. Several crossover operators have been developed for GAs, depending on the format in which individuals are represented. For binary representations, uniform crossover, one point crossover and two points cross over are the most popular. In this case we are using two points crossover with fuzzy crossover rate because we are adding a fuzzy system called 'fuzzyga' that is able of change the crossover and mutation rate.

4.3 Definition of the Fuzzy Systems Used in FPSO+FGA

'fuzzypso': In this case we are using a fuzzy system called 'fuzzypso', and the structure of this fuzzy system is as follow:
Number of Inputs: 2
Number of Outputs: 2
Number of membership functions: 3
Type of the membership functions: Triangular
Number of rules: 9
Defuzzification: Centroid
The main function of the fuzzy system called 'fuzzypso' is to adjust the parameters of the PSO. In this case, we are adjusting the following parameters: 'c1' and 'c2' ; where:
'c1' = Cognitive Acceleration
'c2' = Social Acceleration
We are changing these parameters to test the proposed method. In this case, with 'fuzzypso' is possible to adjust in real time the 2 parameters that belong to the PSO.
'fuzzyga': In this case we are using a fuzzy system called **'fuzzyga'**, the structure of this fuzzy system is as follows:
Number of Inputs: 2
Number of Outputs: 2
Number of membership functions: 3
Type of membership functions: Triangular
Number of rules: 9
Defuzzification: Centroid
The main function of the fuzzy system called 'fuzzypso' is to adjust the parameters of the GA. In this case, we are adjusting the following parameters: 'mu', 'cr'; where:
'k1' = mutation
'k2' = crossover
'fuzzymain': In this case, we are using a fuzzy system called **'fuzzymain'**. The structure of this fuzzy system is as follows:
Number of Inputs: 2
Number of Outputs: 1
Number of membership functions: 3
Type of membership functions: Triangular
Number of rules: 9

Defuzzification: Centroid
The main function of the fuzzy system, called 'fuzzymain' is to decide on the best way for solving the problem, in other words if it is more reliable to use the FPSO or FGA. This fuzzy system is able to receive two inputs, called error and derror, it is to evaluate the results that are generated by FPSO and FGA in the last step of the algorithm.

5 Benchmark Mathematical Functions

To validate our method we used a set of 5 benchmark mathematical functions called F1, F2, F3, F4 and F5; all functions were evaluated with different numbers of variables, in this case, the simulation results were obtained with 8, 16, 32, 64 and 128 variables. Figure 8 shows the plot corresponding to F1 function and equation 2 shows the description to F1 function. Figure 9 shows the plot corresponding to F2 function and equation 3 shows the description to F2 function.

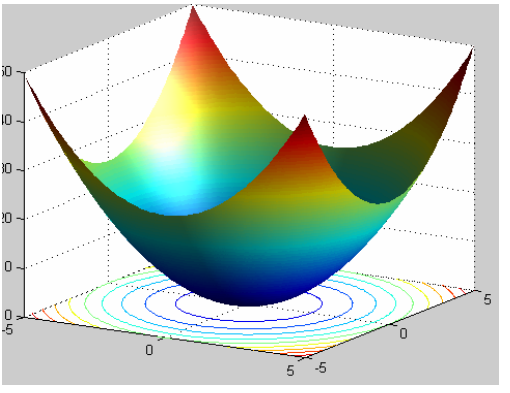


Fig. 8. F1 function

$$\sum_{n=i}^{N} x_n^2 \tag{2}$$

minimum = 0, for $-\infty \leq x \leq \infty$

$$\sum_{n=1}^{N-1} \left\{ 100 \left[x_{n+1} - x_n^2 \right]^2 + \left[1 - x_n \right]^2 \right\} \tag{3}$$

minimum = 0, for $-\infty \leq x \leq \infty$

Figure 10 shows the plot corresponding to F3 function and equation 4 shows the description to F3 function.

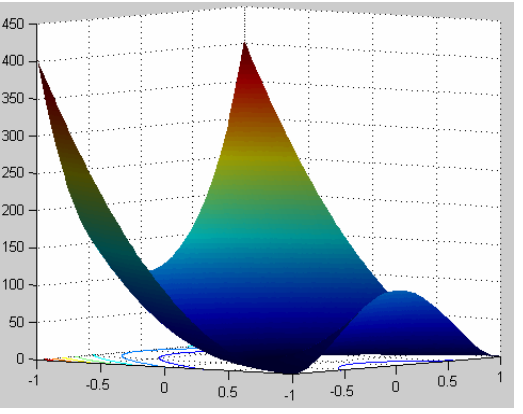


Fig. 9. F2 function

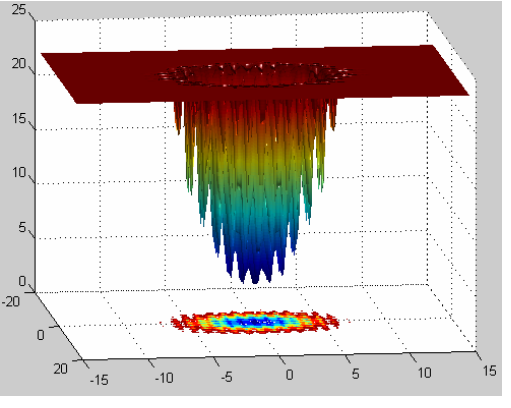


Fig. 10. F3 function

$$f(x) = 20 + e - 20e^{-1/5\sqrt{1/n\sum_{i=1}^{n} x_i^2}} - e^{1/n\sum_{i=1}^{n} \cos(2\pi x_i)} \quad (4)$$

minimum = 0, for -15≤x≤30

Figure 11 shows the plot corresponding to F4 function and equation 5 shows the description to F4 function.

$$10N + \sum_{n=1}^{N}\left[x_n^2 - 10\cos\left(2\pi x_n\right)\right] \quad (5)$$

minimum = 0, for -∞≤x≤∞

Figure 12 shows the plot corresponding to F5 function and equation 6 shows the description to F5 function.

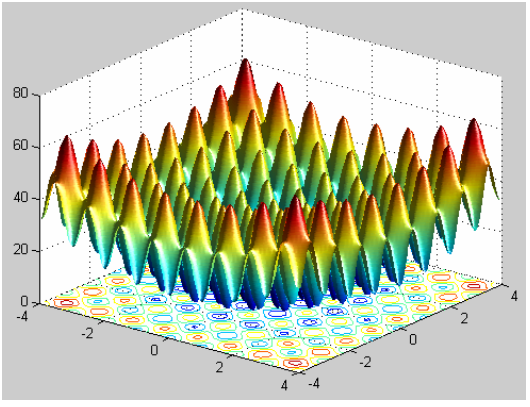


Fig. 11. F4 function

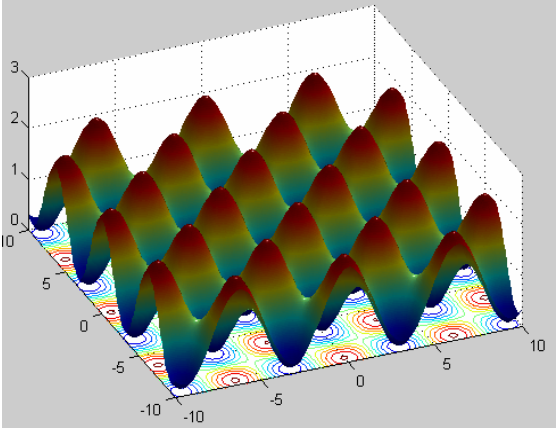


Fig. 12. F5 function

$$1+\sum_{n=1}^{N}\frac{x_n^2}{4000}\prod_{n=1}^{N}\cos\left(x_n\right) \tag{6}$$

minimum = 0, for $-\infty \leq x \leq \infty$

6 Simulations Results

Several tests of the PSO, GA and FPSO+FGA algorithms were made with an implementation in the Matlab programming language.

The implementation was developed using a computer with processor Intel Core 2 Quad of 64 bits that works to a frequency of clock of 2.5 GHz, 6 GB of RAM

Memory and Ubuntu Linux Operating System. The results obtained after applying the GA to the mathematical functions are shown in tables 1, 2, 3, 4 and 5:

The parameters used in the Tables are:

F1, F2, F3, F4 and F5 = Name of function.
VARIABLES= Number of variables used to evaluation.
BEST= The best result obtained.
AVERAGE= The average of 50 times.
WORST= The worst result obtained.

The parameters used on all tests with the GA were: population size = 100 individuals, crossover (k1) = 80%, mutation (k2) = 5%, selection= roulette. On PSO the parameters were: swarm size= 100 particles, cognitive acceleration (c1) = 1, social acceleration (c2) = 0.5, value of velocity at the beginning = 0.95, constriction factor = 1. On FPSO+FGA the parameters were different for k1, k2, c1 and c2, this four parameters were fuzzy parameters, because were obtained dynamically when the method is running, the fuzzy systems can be obtaining this four parameters the best way for achieve good results.

6.1 Simulation Results with the Genetic Algorithm (GA)

From Table 1 it can be appreciated that after executing the GA 50 times, with different number of variables, we can see the best, average and worst results for the F1 function.

Table 1. Experimental results with GA for the F1 function

VARIABLES	BEST	AVERAGE	WORST
8	8.66E-07	0.00094	0.0070
16	4.09E-06	0.00086	0.0083
32	1.14E-06	0.00094	0.0056
64	1.00E-05	0.00098	0.0119
128	1.00E-05	9.42E-04	0.0071

From Table 2 it can be appreciated that after executing the GA 50 times, with different number of variables, we can see the best, average and worst results for the F2 function.

Table 2. Experimental results with GA for the F2 function

VARIABLES	BEST	AVERAGE	WORST
8	5.29E-05	0.05823	0.30973
16	0.00071	0.05683	0.50171
32	0.00228	0.05371	0.53997
64	0.00055	0.053713	0.26777
128	0.000286	0.05105	0.26343

From Table 3 it can be appreciated that after executing the GA 50 times, with different number of variables, we can see the best, average and worst results for the F3 function.

Table 3. Experimental results with GA for the F3 function

VARIABLES	BEST	AVERAGE	WORST
8	3.006976	3.14677173	3.38354
16	3.163963	3.351902975	3.57399568
32	3.246497	3.14677173	3.86201
64	3.519591	3.86961452	4.15382873
128	3.8601773	4.209902992	4.55839099

From Table 4 it can be appreciated that after executing the GA 50 times, with different number of variables, we can see the best, average and worst results for the F4 function.

Table 4. Experimental results with GA for the F4 function

VARIABLES	BEST	AVERAGE	WORST
8	0.499336	6.7430	15.3442
16	8.160601	24.01	43.39
32	46.00850	82.35724	129.548
64	162.4343	247.0152	347.21618
128	524.78094	672.6994	890.93943

From Table 5 it can be appreciated that after executing the GA 50 times, with different number of variables, we can see the best, average and worst results for the F5 function.

Table 5. Experimental results with GA for the F5 function

VARIABLES	BEST	AVERAGE	WORST
8	0.001547453	0.0268979	0.09962
16	0.005378064	0.1215722	0.349648
32	0.141923311	0.4101969	0.917367
64	0.787436284	0.980005	1.002421
128	1.005189444	1.006888	1.008103

6.2 Simulation Results with Particle Swarm Optimization

The results obtained after applying the PSO to the mathematical functions are shown in tables 6, 7, 8, 9 and 10. From Table 6 it can be appreciated that after executing the PSO 50 times, with different number of variables, we can see the best, average and worst results for the F1 function.

Table 6. Experimental results with PSO for the F1 function

VARIABLES	BEST	AVERAGE	WORST
8	1.43E-14	4.21182E-11	9.98E-11
16	6.12E-12	5.31E-11	9.94E-11
32	3.40E-12	5.42E-11	9.86E-11
64	2.01E-12	4.89E-11	9.82E-11
128	3.323E-12	5.34E-11	9.73E-11

From Table 7 it can be appreciated that after executing the PSO 50 times, with different number of variables, we can see the best, average and worst results for the F2 function.

Table 7. Experimental results with PSO for the F2 function

VARIABLES	BEST	AVERAGE	WORST
8	2.84E-13	5.20E-11	9.98E-11
16	1.95E-12	4.97E-11	9.96E-11
32	1.93E-12	5.42E-11	9.83E-11
64	5.95E-12	6.12E-11	9.91E-11
128	2.004E-11	8.60E-11	9.55E-11

From Table 8 it can be appreciated that after executing the PSO 50 times, with different number of variables, we can see the best, average and worst results for the F3 function.

Table 8. Experimental results with PSO for the F3 function

VARIABLES	BEST	AVERAGE	WORST
8	2.9804998406	3.045430224	3.28918888
16	3.3917876285	3.17059523	3.26680512
32	3.106364703	3.217813872	3.39178762
64	3.227560368	3.379519078	3.55310978
128	3.518976467	3.66857109	3.8473198

From Table 9 it can be appreciated that after executing the PSO 50 times, with different number of variables, we can see the best, average and worst results for the F4 function.

From Table 10 it can be appreciated that after executing the PSO 50 times, with different number of variables, we can see the best, average and worst results for the F5 function.

Table 9. Experimental results with PSO for the F4 function

VARIABLES	BEST	AVERAGE	WORST
8	7.57E-11	3.661448	8.95462
16	2.984877	11.1837	24.874822
32	16.1450834	34.1697120	56.714207
64	72.364868	126.01692	198.1616
128	368.575586	467.931817	607.874952

Table 10. Experimental results with PSO for the F5 function

VARIABLES	BEST	AVERAGE	WORST
8	4.43E-11	0.00463639	0.04439
16	6.71E-11	0.007300457	0.034526
32	9.17E-06	0.011476878	0.09483
64	0.137781	0.370872190	0.667802
128	0.85604741	0.970930240	1.003151

6.3 Simulation Results with FPSO+FGA

The results obtained after applying the proposed method FPSO+FGA to the mathematical functions are shown in tables 11, 12, 13, 14 and 15.

From Table 11 it can be appreciated that after executing the FPSO+FGA 50 times, with different number of variables, we can see the best, average and worst results for the F1 function.

Table 11. Experimental results with FPSO+FGA for the F1 function

VARIABLES	BEST	AVERAGE	WORST
8	1.01E-17	4.37E-15	5.11E-12
16	2.25E-16	8.37E-14	1.32E-11
32	9.15E-16	9.41E-13	3.22E-11
64	8.25E-14	5.32E-12	4.08E-11
128	9.00E-14	6.38E-12	8.03E-11

From Table 12 it can be appreciated that after executing the PSO 50 times, with different number of variables, we can see the best, average and worst results for the F2 function.

From Table 13 it can be appreciated that after executing the PSO 50 times, with different number of variables, we can see the best, average and worst results for the F3 function.

Table 12. Experimental results with FPSO+FGA for the F2 function

VARIABLES	BEST	AVERAGE	WORST
8	3.23E-20	2.54E-17	1.25E-16
16	1.32E-18	1.25E-16	2.36E-15
32	4.20E-18	3.50E-16	4.63E-15
64	7.20E-16	8.50E-15	3.63E-14
128	8.23E-15	9.54E-14	6.28E-13

Table 13. Experimental results with FPSO+FGA for the F3 function

VARIABLES	BEST	AVERAGE	WORST
8	0.0012322412	0.01895847	0.0693545
16	0.012536546	0.3654587	0.6689584
32	1.012351254	1.2535658	1.9858445
64	2.58585258	2.9852854	2.6896595
128	2.90002123	2.9135282	2.9315822

From Table 14 it can be appreciated that after executing the PSO 50 times, with different number of variables, we can see the best, average and worst results for the F4 function.

Table 14. Experimental results with FPSO+FGA for the F4 function

VARIABLES	BEST	AVERAGE	WORST
8	8.58E-18	2.32E-17	3.25E-16
16	3.25E-16	3.75E-16	8.25E-16
32	2.54E-15	9.68E-15	3.64E-14
64	1.02E-10	9.98E-09	3.46E-08
128	7.77E-06	8.75E-04	3.68E-02

From Table 15 it can be appreciated that after executing the PSO 50 times, with different number of variables, we can see the best, average and worst results for the F5 function.

Table 15. Experimental results with FPSO+FGA for the F5 function

VARIABLES	BEST	AVERAGE	WORST
8	4.25E-15	4.44E-14	9.98E-14
16	6.50E-15	7.44E-14	9.98E-13
32	4.25E-13	2.41E-12	9.98E-10
64	6.36E-10	1.25E-09	8.16E-08
128	3.45E-05	5.94E-03	3.39E-02

7 Comparison Results between GA, PSO and FPSO+FGA

In Table 16 the comparison of the average results obtained between the GA, PSO and FPSO+FGA methods for the optimization of the 5 proposed benchmark mathematical functions is shown. It can be appreciated that the proposed method

Table 16. Comparison results between the proposed methods

Function	GA	PSO	FPSO+FGA
F1	9.42E-04	5.34E-11	6.38E-12
F2	0.05105	8.60E-11	9.54E-14
F3	4.209902	3.66857109	2.9135282
F4	672.6994	467.931817	8.75E-04
F5	1.006888	0.970930240	5.94E-03

Table 17. T Student Test

T Student Test	GA vs PSO	GA vs FPSO+FGA	PSO vs FPSO+FGA
N	25	25	25
T value	0.43	1.39	1.27
P value	0.334	0.085	0.105

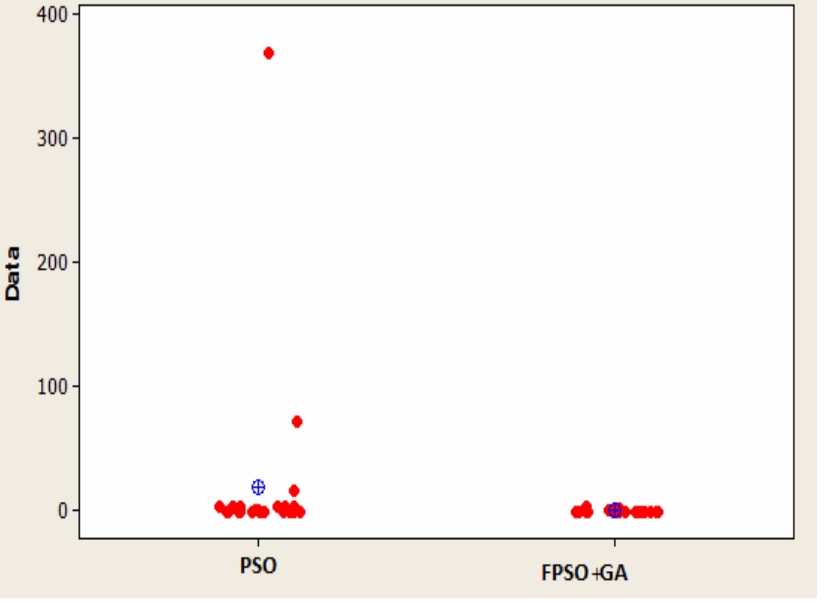


Fig. 13. T Student Plot

was better than GA and PSO used separately because when the number of variables for evaluation by each function is big, the adaptation of parameters is necessary, the results of the average for the functions with 128 variables is shown in Table 16.

From Table 17 it can be appreciated the T Student Test for all functions with all results for each number of variables used in this paper. N represents number of dates taken for the test, in this case, represents all the best values obtained with GA, PSO and FPSO+FGA, T represents the T value obtained, and P represents the P value obtained with this test. We can see in this table the T Student Test with GA vs PSO, GA vs FPSO+FGA and PSO vs FPSO+FGA. It can be seen that the difference between GA and PSO is small, while GA and PSO vs FPSO+FGA the statistical difference is good because the error is small in P.

In figure 13 we can see the plot of the T Student Test for PSO vs FPSO+FGA.

8 Conclusions

The analysis of the simulation results of the evolutionary method considered in this paper, FPSO+FGA lead us to the conclusion that for the optimization of this benchmark mathematical function with this method is a good alternative because it is easier to optimize achieve good results than to try it with PSO or GA separately. This is, because the combination PSO and GA with fuzzy rules gives a new hybrid method FPSO+FGA. It can be seen in Table 17 the T Student Test with good results for our proposed method FPSO+FGA; we are demonstrating statistically that it is reliable uses for this type of functions with n variables. Recently we are working with variables for test the effectiveness of this approach.

Acknowledgment

We would like to express our gratitude to the CONACYT, Institute of Technology for the facilities and resources granted for the development of this research.

References

[1] Man, K.F., Tang, K.S., Kwong, S.: Genetic Algorithms: Concepts and Designs. Springer, Heidelberg (1999)

[2] Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan, pp. 39–43 (1995); Lu, J.-G.: Title of paper with only the first word capitalized. J. Name Stand. Abbrev. (in press)

[3] Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, pp. 1942–1948 (1995)

[4] Holland, J.H.: Adaptation in natural and artificial system. The University of Michigan Press, Ann Arbor (1975)

[5] Valdez, F., Melin, P.: Parallel Evolutionary Computing using a cluster for Mathematical Function Optimization, Nafips, San Diego CA, USA, June 2007, pp. 598–602 (2007)
[6] Castillo, O., Melin, P.: Hybrid intelligent systems for time series prediction using neural networks, fuzzy logic, and fractal theory. IEEE Transactions on Neural Networks 13(6), 1395–1408 (2002)
[7] Fogel, D.B.: An introduction to simulated evolutionary optimization. IEEE transactions on neural networks 5(1), 3–14 (1994)
[8] Goldberg, D.: Genetic Algorithms. Addison-Wesley, Reading (1988)
[9] Emmeche, C.: Garden in the Machine. The Emerging Science of Artificial Life, p. 114. Princeton University Press, Princeton (1994)
[10] Angeline, P.J.: Using Selection to Improve Particle Swarm Optimization. In: Proceedings 1998 IEEE World Congress on Computational Intelligence, pp. 84–89. IEEE, Anchorage (1998)
[11] Angeline, P.J.: Evolutionary Optimization Versus Particle Swarm Optimization: Philosophy and Performance Differences. In: Porto, V.W., Waagen, D. (eds.) EP 1998. LNCS, vol. 1447, pp. 601–610. Springer, Heidelberg (1998)
[12] Back, T., Fogel, D.B., Michalewicz, Z. (eds.): Handbook of Evolutionary Computation. Oxford University Press, Oxford (1997)
[13] Montiel, O., Castillo, O., Melin, P., Rodriguez, A., Sepulveda, R.: Human evolutionary model: A new approach to optimization. Inf. Sci. 177(10), 2075–2098 (2007)
[14] Castillo, O., Valdez, F., Melin, P.: Hierarchical Genetic Algorithms for topology optimization in fuzzy control systems. International Journal of General Systems 36(5), 575–591 (2007)

Local Survival Rule for Steer an Adaptive Ant-Colony Algorithm in Complex Systems

Claudia Gómez Santillán[1,2], Laura Cruz Reyes[1], Elisa Schaeffer[3], Eustorgio Meza[2], and Gilberto Rivera Zarate[1]

[1] Instituto Tecnológico de Ciudad Madero (ITCM). 1ro. de Mayo y Sor Juana I. de la Cruz s/n CP. 89440, Tamaulipas, México. Phone: (52) 833 3574820 Ext. 3024
cggs71@hotmail.com, lcruzreyes@prodigy.net.mx, riveragil@gmail.com

[2] Instituto Politécnico Nacional, Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada (IPN-CICATA). Carretera Tampico-Puerto Industrial Alt., Km.14.5. Altamira, Tamps., México. Phone: 018332600124
emezac@ipn.mx

[3] CIIDIT & FIME, Universidad Autónoma de Nuevo León (UANL), Av. Universidad s/n, Cd. Universitaria, CP.66450, San Nicolás de los Garza, N.L., México
elisa@yalma.fime.uanl.mx

Abstract. The most prevalent P2P application today is file sharing, both among scientific users and the general public. A fundamental process in file sharing systems is the search mechanism. The unstructured nature of real-world large-scale complex systems poses a challenge to the search methods, becasuse global routing and directory services are impractical to implement. In this paper, a new ant-colony algorithm, Adaptive Neighboring-Ant Search (AdaNAS), for the semantic query routing problem (SQRP) in a P2P network is presented. The proposed algorithm incorporates an adaptive control parameter tuning technique for runtime estimation of the time-to-live (TTL) of the ants. AdaNAS uses three strategies that take advantage of the local environment: learning, characterization, and exploration. Two classical learning rules are used to gain experience on past performance using three new learning functions based on the distance traveled and the resources found by the ants. The experimental results show that the AdaNAS algorithm outperforms the NAS algorithm where the TTL value is not tuned at runtime.

Keywords: parameter tuning, search algorithm, peer-to-peer, adaptive algorithm, local environment, ant-colony algorithms.

1 Introduction

Although popular for other uses, the World Wide Web is impractical for user-to-user file sharing as it requires centralized infrastructure such as an HTTP server.

P. Melin et al. (Eds.): Soft Comp. for Recogn. Based on Biometrics, SCI 312, pp. 245–265.
springerlink.com

In the past decade, a new class of networks called *peer-to-peer* (P2P) systems began to spread as a solution to the increasing demand of file sharing among Internet users. In P2P networks, the users interconnect to offer their files to one another [32]. The participants, called *peers*, may connect and disconnect freely, and do so constantly, which triggers frequent changes in the network structure [35].

One of the main advantages is that peers are equal in terms of functionality and tasks which are developed. This produces high fault tolerance and auto-organization: peers form unstructured networks with an acceptable connectivity and performance. The *Semantic Query Routing Problem* (SQRP) consists in deciding, based on a set of keywords, to which neighbor to forward the query to search files related with the keywords [25, 35].

The lack of global structure caused that flooding-based search mechanisms have been mainly employed. Flooding-based mechanisms are simple, but unfortunately generate vast amounts of traffic in the network and may produce congestion on Internet. Existing approaches for SQRP in P2P networks range from simple broadcasting techniques to sophisticated methods [27, 32, 33]. The latter includes proposals based on *ant-colony systems* [14] that are specifically suited for handling routing tables in telecommunications. There exist few algorithms used for SQRP, including SemAnt [25] and Neighboring-Ant Search (NAS) [11], the latter based on the former. In this work we propose an algorithm as an extension to NAS, called the Adaptive Neighboring-Ant Search (AdaNAS). AdaNAS is hybridized with three local strategies: learning, structural characterization and exploration. These strategies are aimed to produce a greater amount of results in a lesser amount of time. The time-to-live (TTL) parameter is tuned at runtime based on the information acquired by these three local strategies.

2 Background

In order to place the research in context, this section is divided in three parts. The first part models a P2P network with graph theory, and in we continue with structural characterization. The part describes the basic ant-colony algorithms for SQRP algorithms and the last part explains parameter tuning and adaptation.

2.1 Graph Theory

A P2P network is a distributed system that can be modeled mathematically as a *graph*, $G = (V,E)$, where V is a set of *nodes* and $E \subseteq V \times V$ is a set of (symmetrical) *connections*. For more information on graph theory, we recommend the textbook by Diestel [13]. Each peer in the network is represented by a node (also called a *vertex*) of the graph. The direct communications among the peers are represented by the connections (also called the *edges*) of the graph. We denote by n the number of nodes in the system and identify the nodes by integers, $V = \{1, 2, 3, \ldots, n\}$. Two nodes that are connected are called *neighbors*; the set of all neighbors of a node i is denoted by $\Gamma(i)$. The number of neighbors of a node i is called *degree* and is denoted by k_i. Two nodes i and j are said to be connected if there exists at least one sequence of connections that begins at i, traverses from node to node

through the connections of the graph, and ends at *j*. Such sequences of connections are called *routes* or *paths* and the number of connections traversed is the length of the route.

2.2 Structural Characterization

For the purpose of analyzing the structure and behavior of complex systems modeled as graphs, numerous characterization functions have been proposed [10]. There are two main types of these functions: those based on global information that require information on the entire graph simultaneously and those based on local information that only access the information of a certain node i and its neighborhood Γ(i) at a time. In this section we review the local structural characterization functions related with this work.

2.2.1 Degree Distribution

The degree k_i of a node i is a local measure of network. $P(k)$ denotes the number of nodes that have degree k, normalized by n. The measure of $P(k)$ can be interpreted as the probability that a randomly chosen node i has degree k. The values of $P(k)$ for $k \in [0, n\text{-}1]$ (supposing that there can only be at most one connection between each pair of distinct nodes) form the *degree distribution* of the graph. Whereas the degrees themselves are local properties, obtaining the degree distribution is a global computation.

The degree distribution is widely used to classify networks according to the *generation models* that produce such distributions. Among the first and most famous generation models are the *uniform random graphs* of Erdös and Rényi [15] and Gilbert [17] that yield a binomial distribution that at the limit, approaches the Poisson distribution and most of the nodes in the graph have similar degrees [9].

In the past decade, another type of generation models became popular as various studies revealed that the degree distribution of some important real-world networks (including the WWW, the Internet, biological and social systems) was not Poisson distribution at all, but rather a power-law distribution [1, 2, 16], $P(k) \sim k^{-\gamma}$ with values of γ typically ranging between two and three. The models that produce such distributions are called *scale-free* network models. The notable structural property in networks with powerlaw distribution is the presence of a small set of extremely well-connected nodes that are called *hubs*, whereas a great majority of the nodes has a very low degree [6, 28]. This property translates into high fault tolerance under random flaws, but high vulnerability under deliberate attack [2].

2.2.2 Parameter Tuning and Adaptation

Metaheuristics offer solutions that are often close to the optimum, but with a reasonable amount of resources used when compared to an exact algorithm. Unfortunately, the metaheuristics are usually rich in parameters. The choice of the values for the parameters is nontrivial and in many cases the parameters should vary during the runtime of the algorithm [8, 19].

The process of selecting the parameter values is known as *tuning*. The goal of offline tuning is to provide a static initial parameter configuration to be used throughout the execution of the algorithm, whereas online tuning, also known as *parameter control* or *adaptation*, is the process of adjusting the parameter values at runtime. We use a discrete model for adaptation based on the proposed by Holland in 1992 [22]. We assume that the system takes actions at discrete steps $t = 1, 2, 3, \ldots$, as this assumption applies to practically all computational System. The proposed model is described in section four.

3 SQRP Search Strategies

In this section we present the problem focused in this work. First, we describe the semantic query routing problem (SQRP) as a search process. Then, strategies for solve SQRP are shown including our proposed algorithm which uses an adaptive strategy for adjusting an important parameter for the search process: TTL.

3.1 SQRP Description

SQRP is the problem of locating information in a network based on a query formed by keywords. The goal in SQRP is to determine shorter routes from a node that issues a query to those nodes of the network that can appropriately answer the query by providing the requested information. Each query traverses the network, moving from the initiating node to a neighboring node and then to a neighbor of a neighbor and so forth, until it locates the requested resource or gives up in its absence. Due to the complexity of the problem [3, 24, 25, 33, 34, 35], solutions proposed to SQRP typically limit to special cases.

The general strategies of SQRP algorithms are the following. Each node maintains a local database of documents r_i called the *repository*. The search mechanism is based on nodes sending messages to the neighboring nodes to query the contents of their repositories. The *queries* q_i are messages that contain keywords that describe for possible matches. If this examination produces results to the query, the node responds by creating another message informing the node that launched the query of the resources available in the responding node. If there are no results or there are too few results, the node that received the query forwards it to one or more of its neighbors. This process is repeated until some predefined stopping criteria is reached. An important observation is that in a P2P network the connection pattern varies among the net (*heterogeneous topology*), moreover the connections may change in time, and this may alter the routes available for messages to take.

3.2 SQRP Algorithms

The most popular technique for searching in P2P systems is flooding, where each message is assigned a positive integer parameter known as the *time-to-live* (TTL) of the message. As the message propagates from one node to another, the value of TTL is decreased by one by each forwarding node. When TTL reaches zero, the message will be discarded and no longer propagated in the system. The main

disadvantage of flooding is the rapid congestion of the communication channels [20]. Another widely used search strategy is the *random walk* [3]. A random walk in a graph is a route where the node following the initiating node is chosen uniformly at random among its neighbors.

3.2.1 AntSearch

Wu et al. [34] propose an algorithm called *AntSearch.* The main idea in the AntSearch algorithm is using pheromone values to identify the free-riders, prevent sending messages to those peers in order to reduce the amount of redundant messages the estimation of a proper TTL value for a query flooding is based on the popularity of the resources. Wu et al. use three metrics to measure the performance of the AntSearch. One is the *number of searched files* for a query with a required number of results, *R*: a good search algorithm should retrieve the number of results over but close to *R*. The second one is the *cost per result* that defines the total amount of query messages divided by the number of searched results; this metric measure how many average query messages are generated to gain a result. Finally, *search latency* is defined as the total time taken by the algorithm.

3.2.2 SemAnt

Algorithms that incorporate information on past search performance include the SemAnt algorithm [25, 26] that uses an ant-colony system to solve SQRP in a P2P environment. SemAnt seeks to optimize the response to a certain query according to the popularity of the keywords used in the query. The algorithm takes into account network parameters such as bandwidth and latency. In SemAnt, the queries are the ants that operate in parallel and place pheromone on successful search routes. This pheromone evaporates over time to gradually eliminate old or obsolete information. Also Michlmayr [25] considers parameter tuning for the SemAnt algorithm, including manual adjustment of the TTL parameter from a set of possible values {15, 20, 25, 30, 35} and concludes that 25 is the best value for the parameter. The adjustment of the TTL is made without simultaneous tuning of the other parameters.

3.2.3 Neighboring-Ant Search

NAS [11] is also an ant-colony system, but incorporates a local structural measure to guide the ants towards nodes that have better connectivity. The algorithm has three main phases: an evaluation phase that examines the local repository and incorporates the classical lookahead technique [27], a transition phase in which the query propagates in the network until its TTL is reached, and a retrieval phase in which the pheromone tables are updated.

Most relevant aspects of former works have been incorporated into the proposed NAS algorithm. The framework of AntNet algorithm is modified to correspond to the problem conditions: in AntNet the final addresses are known, while NAS algorithm does not has a priori knowledge of where the resources are located. On the other hand, differently to AntSearch, the SemAnt algorithm and NAS are focused on the same problem conditions, and both use algorithms based on AntNet algorithm. However, the difference between the SemAnt and NAS is that SemAnt only learns from past experience, whereas NAS takes advantage of

the local environment. This means that the search in NAS takes place in terms of the classic local exploration method of Lookahead [27], the local structural metric DDC[29] its measures the differences between the degree of a node and the degree of its neighbors, and three local functions of the past algorithm performance.

3.2.4 Adaptative Neighboring-Ant Search

The proposed algorithm in this work, *Adaptive Neighboring Ant Search* (AdaNAS) is largely based on the NAS algorithm, but includes the adaptation of the TTL parameter at runtime, in addition to other changes. The mechanism that may extend the TTL for an ant is called the *survival rule*. It incorporates information on past queries relying on the learning strategies included in AdaNAS, basic characteristics of SQRP and a set of parameters that are adjusted according to the results found when using the *survival rule* itself. The rule evaluates the length of the shortest known route that begins with the connection (*i, j*) from the current node *i* to a node that contains good results for the query *t*. The form in which the algorithm operates is explained in detail later in Sections 4 and 5.

4 AdaNAS Model

In this section, we present a multi-agent model in order to describe the adaptive behavior of AdaNAS. We begin by formulating it as an adaptive system in the notation presented in Section 2.2.2.

4.1 The General Model

The environment is the P2P network, in which two stimuli are observed: the occurrences of the documents being searched (I_1) and the degree k_i of the node i (I_2). The environment has the following order to send stimuli: observing I_1 has a higher priority than observing I_2. AdaNAS is an ant-colony system, where each ant is modeled as an agent. AdaNAS has four agent types:

- The *query ant* is accountable for attending the users' queries and creating the *Forward Ant*; moreover it updates the pheromone table by means of evaporation. There is a *query ant* for each node in the net and it stays there while the algorithm is running.
- The *Forward Ant* uses the learning strategies for steering the query and when it finds resources creates the *backward ant*. It finishes the routing process when its TTL is zero or the amount of found resources is enough that is denoted by R then, it creates an *update ant*.
- The *backward ant* is responsible for informing to *query ant* the amount of resources in a node found by the *Forward Ant*. In addition, it updates the values of some learning structures that are the bases of the *survival rule* which will be explaining later (Section 4.2.3).
- The *update ant* drops pheromone on the nodes of the path generated by the *Forward Ant*. The amount of pheromone deposited depends on quantity of found resources (*hits*) and number of edges traveled (*hops*) by the *Forward Ant*.

The system is subdivided into four parts: the structures A to adapt to the environment (called *agents*), the adaptation plan P, the memory M, and the operators O. Typically, A has various alternative states A_1, A_2, A_3, . . . among which one is to be chosen for the system, according to the observations made on the environment. On the other hand, P is typically a set of rules, one or more which can be applied. These rules apply the operations in the set O. An operator is either a deterministic function, denoted as $(A_i, P_j) \rightarrow Ak$, or a stochastic function to a probability distribution over a set of states for selecting A_k. The memory M permits the system to collect information on the condition of the environment and the system itself, to use it as a base for the decision making. The observations of the environment are taken as stimuli that trigger the operators.

The routing process implemented in the *Forward Ant* is required to be adaptive, thus A is defined in function of this agent. The possible states for A are five:

A_1: No route has been assigned and the *Forward Ant* is at the initial node. The ant can be only activated when the *query ant* send it a query and can only receive once time each stimulus.
A_2: A route has been assigned and TTL has not reached zero.
A_3: TTL is zero.
A_4: *Forward Ant* used the *survival rule* to extend TTL.
A_5 = Terminal state is reached by the *Forward Ant*.
X:

The Figure 1 shows the AdaNAS adaptive model. According to the stimuli -the number of documents found (dotted line, I_1) and degree of the node (solid line, I_2)- an operator is selected. The line style for state transitions follows that of the stimuli: dotted line for transitions proceeding from I_1 and solid for I_2.

The memory M is basically composed of four structures that store information about previous queries. The first of these structures is the three dimensional pheromone table τ. The element $\tau_{i;j;t}$ is the preference for moving from node i to a neighboring node j when searching by a keyword t. In this work, we assume that each query contains one keyword and the total number of keywords (or *concepts*) known to the system is denoted by C.

The pheromone table $M_1 = \tau$ is split into n bi-dimensional tables, $\tau_{j;t}$, one for each node. These tables only contain the entries $\tau_{j;t}$ for a fixed node i and hence have at most dimensions C x $|\Gamma(i)|$. The other three structures are also three-dimensional tables $M_2 = D$, $M_3 = N$ and $M_4 = H$, each splits into n local bi-dimensional tables in the same manner. The information in these structures is of the following kind: currently being at node i and searching for t, there is a route of distance $D_{i;j;t}$ starting at the neighbor j that leads to a node identified in $N_{i;j;t}$ that contains $H_{i;j;t}$ hits or matching documents.

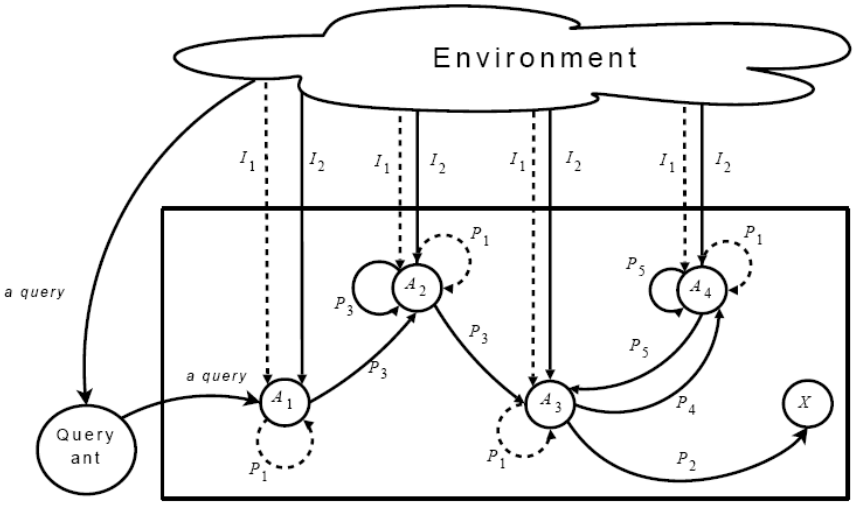


Fig. 1. AdaNAS Adaptive Model General

The adaptive plans *P* are the following:

P_1**:** **The *backward ant*.** Created when a resource is found; modifies M_2, M_3 and M_4.

P_2**:** **The *update ant*.** Modifies the pheromone table $M_1 = \tau$ when the *Forward Ant* reached 0 and *survival rule* can not to proceed.

P_3**:** **The *transition rule*.** Selects the next node applying the inherent learning stored in pheromone trails and in the memory structure $M_2 = D$.

P_4**:** **The *survival rule*.** Proceeds when the learning stored in M_2, M_3 and M_4 permits to extend TTL and determines how much TTL must be extended.

P_5**:** **The *modified transition rule*.** A variation of transition rule that eliminates the pheromone and degree effects.

The operators *O* of AdaNAS are the following:

O_1. $(A_1, I_1) \rightarrow A_1$:	Documents are found and a *backward ant* (P_1) updates the memory -no change in the state of the system-.
O_2. $(A_1, I_2) \rightarrow A_2$:	An ant selects the route according to the transition rule (P_3).
O_3. $(A_2, I_1) \rightarrow A_2$:	Similar to O_1.
O_4.$(A_2,I_2) \rightarrow \{(p_{2;2;2},A_2), (p_{2;3;2},A_3)\}$:	The transition rule (P_3) either keeps the ant in the same state with probability $p_{2;2;2}$ or moves to state A_3 with probability $p_{2;3;2}$.
O_5.$(A_3, I_1) \rightarrow A_3$	Similar to O_1.
O_6.$(A_3,I_2) \rightarrow \{(p_{3;4;2},A_4), (p_{3;X;2},X)\}$:	The survival rule (P_4) and update ant (P_2) –

with probability $p_{3;4;2}$, the ant extends its TTL using P_4, or -with probability $p_{3;X;2}$, applying P_2 the ant reach its terminal state.

O_7. $(A_4, I_1) \rightarrow A_4$: Similar to O_1.

O_8.$(A_4, I_2) \rightarrow \{(p_{3;3;2}, A_3), (p_{3;4;2}, A_4)\}$: The modified transition rule (P_5) -the ant either stays in the same state or moves to state A_3-.

The general model is illustrated in Figure 1 where can be observed the transitions among states of the *Forward Ant*.

4.2 Behavior Rule

An ant-colony algorithm has rules that determine its behavior. These rules define why the ants construct and evaluate the solution and why the pheromone is updated and used. Although the pheromone is the main learning structure, AdaNAS has three more: *D*, *N* and *H*, for know the distances toward the nodes that contain in its repository matching documents. AdaNAS own several behavior rules: the *transition rule*, *the update rules*, the *survival rule* and the *modified transition rule*.

4.2.1 Transition Rule

The transition rule P_3 considers two structures to determine the next state: τ and D. The transition rule for an ant x that is searching by keyword t and is in the node r is the following, Equation 1:

$$\ell(x,r,t) = \begin{cases} \arg\max_{i \in (\Gamma(r)/\Lambda s)} \{\psi(r,i,t)\}, si \;\; p < q \\ \mathcal{L}(\chi, r, t) \qquad\qquad otherwise; \end{cases} \quad (1)$$

where p is a pseudo-random number, q is a algorithm parameter that defines the probability of using of the exploitation technique, $\Gamma(r)$ is the set of neighbors nodes of r, Λx is the set of nodes previously visited by x, and Equation 2, defined by:

$$\psi(r,i,t) = (w_d \cdot \kappa(r,i) + w_i \cdot (D_{r,i,t})^{-1})^{\beta_1} \cdot (\tau_{r,i,t})^{\beta_2}, \quad (2)$$

where W_{deg} is the parameter that defines the degree importance, W_{dist} defines the distance importance toward the nearest node with matching documents($D_{r;i;t}$), β_1 intensifies the local metrics contribution (degree and distance), β_2 intensifies pheromone contribution ($\tau_{r;i;t}$), $\kappa(r, i)$ is a normalized degree measure expressed in Equation 3:

$$\kappa_{r,i} = \frac{k_i}{\max_{j \in \Gamma(r)} \{k_j\}}, \quad (3)$$

and $\mathcal{L}$ is the exploration technique expressed, in Equation 4:

$$\mathcal{L}(x, r, t) = f(\{p_{x,r,i,t} \mid i \in \Gamma(r)\}), \quad (4)$$

where $f(\{p_{x;r;i;t} \mid i \in \Gamma(r)\})$ is a roulette-wheel random selection function that chooses a node i depending on its probability $p_{x;r;i;t}$ which indicates the probability of the ant x for moving from r to i searching by keyword t ant it is defined in Equation 5:

$$p_{x,r,i,t} = \frac{\psi(r,i,t)}{\sum_{j \in (\Gamma(r)/\Lambda x)} \psi(r,i,t)} \quad (5)$$

The parameters in these equations are: β_1**:** local measure intensification parameter, β_2**:** pheromone intensification parameter, W_{deg}**:** weight factor that defines the importance of the degree, W_{dist}**:** weight factor that defines the importance of the distance, q**:** the relative importance of exploitation versus exploration.

The tables D and τ were described in the previous section. The exploration strategy $\mathcal{L}$ is activated when $p \geq q$ and stimulates the ants to search for new paths. In case that $p<q$, the exploitation strategy is selected: it prefers nodes that provide a greater amount of pheromone and better connectivity with smaller numbers of hops toward a resource. As is shown in the transition rule, β_2 is the intensifier of the pheromone trail, and β_1 is the intensifier of the local metrics, this means that the algorithm will be only steered by the local metrics when $\beta_2 = 0$, or by the pheromone when $\beta_1 = 0$. In this work the initial values are $\beta_1 = 2$ and $\beta_2 = 1$.

4.2.2 Update Rules

There are two basic update rules in an ant colony algorithm: the evaporation and increment of pheromone. The evaporation method of AdaNAS is based on the technique used in SemAnt [25], while the increment strategy is based on the proposed in NAS [11]. Both update rules are described below.

Pheromone Evaporation Rule, the pheromone evaporation is a strategy whose finality is avoid that the edges can take very big values of pheromone trail causing a greedy behavior on the algorithm. Each unit time the query ant makes smaller the pheromone trail of the node where the query ant is, by multiplying the trail by the evaporation rate ρ, which is a number between zero and one. To avoid very low values in the pheromone the rule incorporates a second term consisting of the product $\rho\tau_0$, where τ_0 is the initial pheromone value. The Equation 6 expresses mathematically the evaporation pheromone rule.

$$\tau_{r,s,t} \leftarrow (1-\rho) \bullet \tau_{r,s,t} + \rho \bullet \tau_0 \quad (6)$$

Pheromone Increment Rule, when a *Forward Ant* finishes, it must express its performance in terms of pheromone by means of an *update ant* whose function is to increase the quantity of pheromone depending on amount of documents found

and edges traversed by *Forward Ant*. This is done each time that an *update ant* passes on one node. The Equations 7 and 8 describe the *pheromone increment rule*.

$$\tau_{r,s,t} \leftarrow \tau_{r,s,t} + \Delta\tau_{r,s,t}(x) \quad (7)$$

where $\tau_{r;s;t}$ is the preference of going to s when the *Forward Ant* is in r and is searching by keyword t, $\Delta\tau_{r;s;t}(x)$ is the amount of pheromone dropped on $\tau_{r;s;t}$ by a *backward ant* generated by the *Forward Ant x* and can be expressed like:

$$\Delta\tau_{r,s,t}(x) \leftarrow \left[w_h \frac{hits(x,s)}{\mathrm{R}} + (1-w_h)\frac{1}{hops(x,r)} \right] \quad (8)$$

where *hits*(*x, s*) is the amount of documents found by the *Forward Ant x* from *s* to end of its path, and *hops*(*x, r*) is the length of the trajectory traversed by the *Forward Ant x* from *r* to the final node in its route passing by *s*.

4.2.3 Survival Rules

As can be seen in Figure 2, P_1 (the backward ant) updates the memory structures $M_2 = D$, $M_3 = N$, and $M_4 = H$. These structures are used in the survival rule (P_4) to increase time to live. This survival rule can be only applied when TTL is zero (see Figure 2(c)). The survival rule can be expressed mathematically in terms of the structures H, D and N as see in Equation 9:

$$\Delta TTL(x,i,t) = \begin{cases} D_{i,\omega(x,i,t),t}; si\ \Omega(x,i,t) > Z_x \\ 0; en\ otro\ caso \end{cases} \quad (9)$$

where ΔTTL(*x, i, t*) is the increment assigned to the TTL of ant *x* (that is, number of additional steps that the ant will be allowed to take) when searching for resources that match to *t*, currently being at node *i*. The number of additional steps $D_{i;\omega(x;i;t);t}$ for arriving in the node $\omega(x, i, t)$ is determined from the shortest paths generated by previous ants, and is taken when its associated efficiency Ω(*x, i, t*) is better than *Zx* which is a measure of current performance of the ant *x*. The auxiliary functions are shown in Equations10 and 11:

$$\Omega(x,i,t) = \max_{j\in(\Gamma(i)/\Lambda_x)} \left\{ \frac{H_{i,j,t}}{D_{i,j,t}} \middle| N_{i,j,t} \notin \Lambda_x \right., \quad (10)$$

$$\omega(x,i,t) = \arg\Omega(x,i,t), \quad (11)$$

where Γ(*i*) is the set of neighbors of node *i* and Λ*x* is the set of nodes previously visited by the ant *x*. The tables of hits *H*, of distances *D*, and of nodes *N* were explained in the previous section. The function $\omega(x, i, t)$ determines which node that is neighbor of the current node *i* and that has not yet been visited has previously produced the best efficiency in serving a query on *t*, where the efficiency is measured by Ω(*x, i, t*).

4.2.4 Modified Transition Rule

The *modified transition rule* is a special case of *transition rule* (see Equations 4 and 5) where $\beta_2 = 0$, $W_{deg} = 0$ and $q = 1$. This rule is greedy and provokes the replication of paths generated by previous ants. This rule takes place when TTL has been extended canceling the normal *transition rule*. Mathematically can be express in Equations 12 and 13, like:

$$\ell_m(x,r,t) = \left\{ \arg\max_{i \in (\Gamma(r)/\Lambda s)} \left\{ \psi(r,i,t) \right\} \right. \tag{12}$$

where ℓm is the *modified transition rule*, r is the current node in the path, t is the searched keyword, Λx is the set of nodes visited by the *Forward Ant x* and

$$\psi(r,i,t) = (w_i \cdot (D_{r,i,t})^{-1})^{\beta_1} \tag{13}$$

where W_{dist} is a parameter that defines the influence of $D_{r;i;t}$ that is the needed distance for arriving in the known nearest node with documents with keyword t, from r passing by i and β_1 is the distance intensifier.

5 AdaNAS Algorithm

AdaNAS is a metaheuristic algorithm, where a set of independent agents called ants cooperate indirectly and sporadically to achieve a common goal. The algorithm has two objectives: it seeks to maximize the number of resources found by the ants and to minimize the number of steps taken by the ants. AdaNAS guides the queries toward nodes that have better connectivity using the local structural metric degree (defined in Section 2.2), in addition, it uses the well known *look-ahead* technique [26], which, by means of data structures, allows to know the repository of the neighboring nodes of a specific node.

The AdaNAS algorithm performs in parallel all the queries using query ants. Each node has only a query ant, which generates a *Forward Ant* for attending only one user query, assigning the searched keyword t to the *Forward Ant*. Moreover the *query ants* realize periodically the local pheromone evaporation of the node where it is. The process done by *query ant* is represented in Algorithm 1.

In the Algorithm 2 is shown the process realized by the *Forward Ant*. As can be observed all *Forward Ants* act in parallel. In an initial phase (lines 4- 8), the ant checks the local repository, and if it founds matching documents then creates a *backward ant*. Afterwards, it realizes the search process (lines 9-25) while it has live and has not found R documents.

The search process has three sections: Evaluation of results, evaluation and application of the extension of TTL and selection of next node (lines 24-28).

```
     Algorithm 1: Query ant algorithm
1    in parallel for each query ant w located in the node r
2    while the system is running do
3      if the user queries to find R documents with keyword t then
4        create Forward Ant x(r,t,R)
5        activate x
6      End
7      apply pheromone evaporation
8    End
9    end of in parallel
```

The first section, the evaluation of results (lines 10-15) implements the classical Lookahead technique. That is, the ant *x* located in a node *r*, checks the lookahead structure, that indicates how many matching documents are in each neighbor node of *r*. This function needs three parameters: the current node (*r*), the keyword (*t*) and the set of known nodes (*known*) by the ant. The set *known* indicates what nodes the lookahead function should ignore, because their matching documents have already taken into account. If some resource is found, the *Forward Ant* creates a *backward ant* and updates the quantity of found matching documents.

The second section (lines 16-23) is evaluation and application of the extension of TTL. In this section the ant verifies if TTL reaches zero, if it is true, the ant intends to extend its life, if it can do it, it changes the normal *transition rule* modifying some parameters (line 21) in order to create the *modified transition rule*.

The third section of the search process phase is the selection of the next node. Here, the *transition rule* (normal or modified) is applied for selecting the next node and some structures are updated. The final phase occurs when the search process finishes; then, the *Forward Ant* creates an *update ant* for doing the pheromone update.

The Algorithm 3 presents the parallel behavior for each *backward ant* which inversely traverses the path given by the *Forward Ant*. In each node that it visits, it tries to update the structures *D*, *H* and *N*, which will be used for future queries (lines 7-11). The update is realized if the new values point to a nearer node (line 7). After that, it informs to *ant query* of the initial node of the path how many documents the *Forward Ant* found and which path used (line 13).

The Algorithm 4 presents the concurrent behavior for each *update ant* which inversely traverses the path given by the *Forward Ant*. In each node that it visits, it updates the pheromone trail using the Equation 6. (line 5)

```
     Algorithm 2: Forward ant algorithm
1    in parallel for each Forward Ant x(r,t,R)
2    initialization: TTL = TTLmax, hops= 0
3    initialization: path=r, Λ=r, known=r
4    Results= get_ local_ documents(r)
5    if results > 0 then
6      create backward ant y(path, results, t)
7      activate y
8    End
```

```
9   while TTL < 0 and results < R do
10    La_ results= look ahead(r,t,known)
11    if la results > 0 then
12      create backward ant y(path, la results, t)
13      activate y
14      results   results + la results
15    End
16    if TTL > 0 then
17      TTL= TTL – 1
18    Else
19      if (results < R) and ( ΔTTL(x, results, hops) > 0) then
20        TTL= TTL + ΔTTL(x, results, hops)
21        change parameters: q= 1, Wdeg =0, β2=0
22       End
23     End
24  Hops= hops + 1
25  Known= known∪[ ( r ∪ Γ(r))
25  Λ = Λ ∪ r
27  r = ℓ(x,r,t)
28  add to path(r)
29  End
30  create update ant z(x, path, t)
31  activate z
32  kill x
33  end of in parallel
```

6 Experiments

In this section, we describe the experiments we carried during the comparisons of the AdaNAS and NAS algorithms.

6.1 Generation of the Test Data

A SQRP instance is formed by three separate files: topology, repositories, and queries. We generated the experimental instances following largely those of NAS reported by Cruz et al. [11] in order to achieve comparable results. The structure of the environment in which is carried out the process described is called *topology*, and refers to the pattern of connections that form the nodes on the network. The generation of the topology (*T*) was based on the method of Barabási et al. [7] to create a scale-free network. We created topologies with 1, 024 nodes; the number of nodes was selected based on recommendations in the literature [5, 25].

```
    Algorithm 3: Backward ant algorithm
1   initialization: hops= 0
2   in parallel for each backward ant y(path, results, t)
3   for i =| path| - 1 to 1 do
4     r = path(i - 1)
```

```
5       s = path_i
6       hops= hops + 1
7       if D_{r;s;t} > hops then
8         D_{r;s;t} = hops
9         H_{r;s;t} = result
10        N_{r;s;t} = path_h
11      End
12    End
13    send (results, path) to the query ant located in path1
14    kill y
15    end of in parallel
```

```
      Algorithm 4: Update ant algorithm
1     in parallel for each update ant z(path, t, x)
2     for i= |path| - 1 to 1 do
3        r = path_(i - 1)
4        s = path_i
5        τ_{r;s;t}= τr;s;t + Δτ_{r;s;t}(x)
6     End
7     kill z
8     end of in parallel
```

The *local repository* (*R*) of each node was generated using "topics" obtained from ACM Computing Classification System taxonomy (ACMCCS). This database contains a total of 910 distinct topics. Also the content are scale-free: the nodes contain many documents in their repositories on the same topic (identified by keywords) and only few documents on other topics.

For the generation of the *queries* (*Q*), each node was assigned a list of possible topics to search. This list is limited by the total amount of topics of the ACMCCS. During each step of the experiment, each node has a probability of 0.1 to launch a query, selecting the topic uniformly at random within the list of possible topics of the node repository. The probability distribution of *Q* determines how often the query will be repeated in the network. When the distribution is uniform, each query is duplicated 100 times in average.

Table 1. Parameter configuration of the NAS algorithm.

PARAMETER	VALUE	DEFINITION
α	0.07	Global pheromone evaporation factor
ρ	0.07	Local pheromone evaporation factor
β	2	Intensifier of pheromone trail
τ_0	0.009	Pheromone table initialization
q_0	0.9	Relative importance between exploration and exploitation
R	10	Maximum number of results to retrieve
TTL_{max}	10	Initial TTL of the Forward Ants
W	0.5	Relative importance of the resources found and TTL

Table 2. Parameter configuration of the AdaNAS algorithm.

PARAMETER	VALUE	DEFINITION
ρ	0.07	Local pheromone evaporation factor
B_1	2	Intensification of local measurements (degree and distance) in transition rule.
B_2	1	Intensification of pheromone trail in the in the transition rule.
τ_0	0.009	Pheromone table initialization
q	0.9	Relative importance between exploration and Exploitation in the transition rule.
R	10	Maximum number of results to retrieve
TTL_{max}	10	Initial TTL of the Forward Ants
w_h	0.5	Relative importance of the hits and hops in the increment rule
w_{deg}	1	Degree's influence in the transition rule
w_{dist}	1	Distance's influence in the transition rule

The topology and the repositories were created static, whereas the queries were launched randomly during the simulation. Each simulation was run for 15,000 queries during 500 time units, each unit has 100ms. The average performance was studied by computing three performance measures of each 100 queries:

- **Average hops,** defined as the average amount of links traveled by a Forward Ant until its death, that is, reaching either the maximum amount of results required R or running out of TTL.
- **Average hits,** defined as the average number of resources found by each Forward Ant until its death.
- **Average efficiency,** defined as the average of resources found per traversed edge (hits/hops).

6.2 Parameters

The configuration of the algorithms used in the experimentation is shown in Tables 1 and 2. The first column is the parameter, the second column is the parameter value and the third column is a description of the parameter. These parameter values were based on recommendations of the literature [11, 14, 25, 30, 31].

6.3 Results

The goal of the experiments was to examine the effect of the strategies incorporated in the AdaNAS algorithm and determine whether there is a significant contribution to the average efficiency. The main objective of SQRP is to find a set of paths among the nodes launching the queries and the nodes containing the resources, such that the efficiency is greater, this is, the quantity of found resources is maximized and the quantity of steps given to find the resources is minimized.

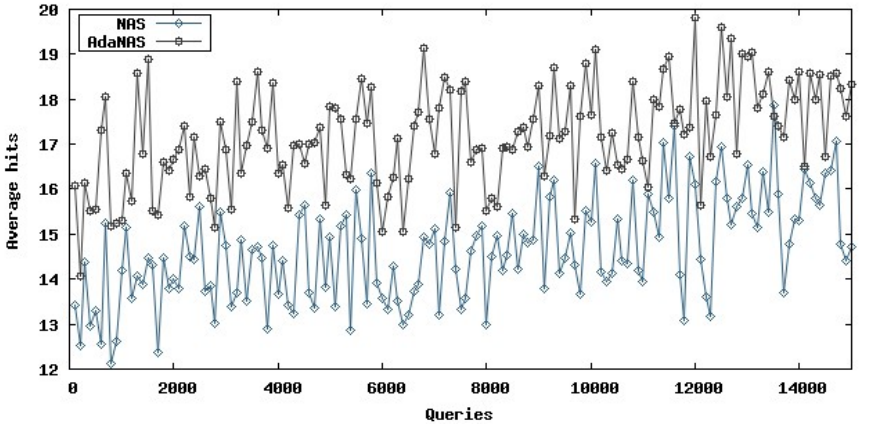


Fig. 2. Learning evolution in terms of the number of resources found for AdaNAS and NAS algorithms.

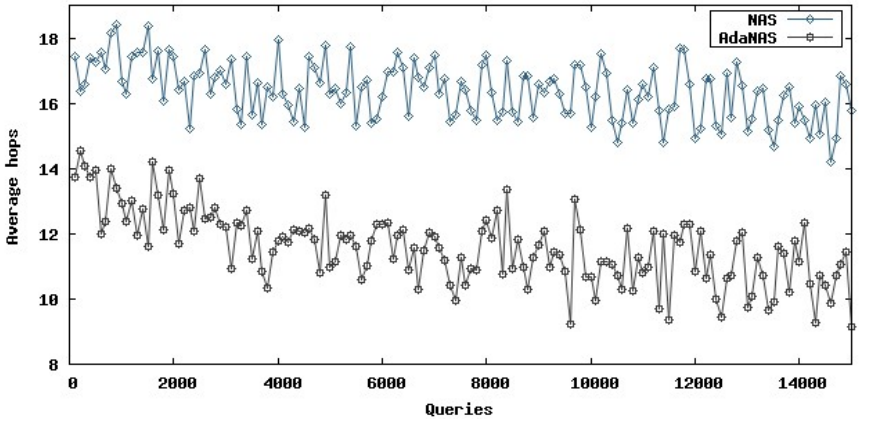


Fig. 3. Learning evolution in terms of the length of the route taken for AdaNAS and NAS algorithms.

Figure 2 shows the *average hits* performed during 15,000 queries with AdaNAS and NAS algorithms on an example instance. NAS starts off approximately at 13.4 hits per query; at the end, the average hit increases to 14.7 hits per query. For AdaNAS the average hit starts at 16 and after 15,000 queries the average hit ends at 18.3. On the other hand, Figure 3 shows the *average hops* performed during a set of queries with NAS and AdaNAS. NAS starts approximately at 17.4 hops per query; at the end, the average hops decrease to 15.7 hops per query. For AdaNAS the average hops starts at 13.7 and after 15, 000 queries the average hops ends at 9.1. Finally, Figure 4 shows the *average efficiency* performed during a set of queries. NAS starts approximately at 0.76 hits per hop; at the end, it increases to 0.93

hits per hop. For AdaNAS the average efficiency starts at 1.17 hits per hop and after 15, 000 queries the average efficiency ends at 2.

The adaptive strategies of AdaNAS show an increment of 24.5% of found documents, but the biggest contribution is a reduction of hops in 40%, giving efficiency approximately twice better on the final performance of NAS. This observation suggests that the use of degree instead of DDC was profitable. In addition, the incorporation of the survival rule permits to improve the efficiency, because it guides the Forward Ants to nodes that can satisfy the query. Moreover, in future works it will be important to study adaptive strategies for other parameters as well as the initial algorithm parameter configuration in search of further improvement in the efficiency.

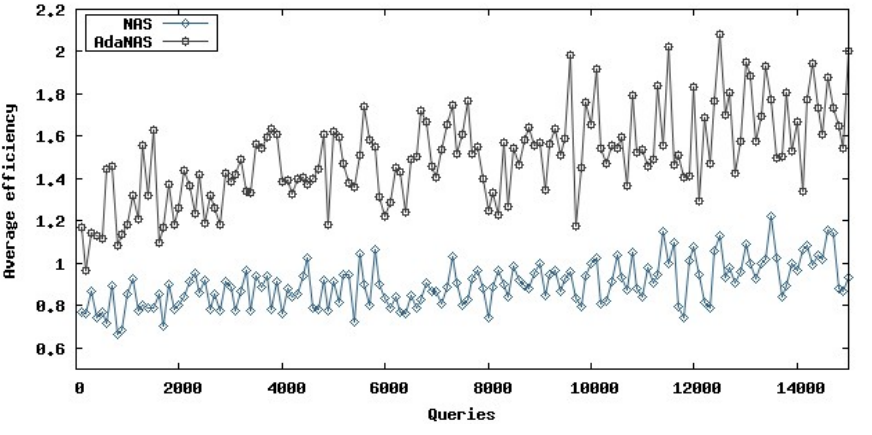


Fig. 4. Learning evolution in terms of the efficiency (hits/ hop) for AdaNAS and NAS algorithms.

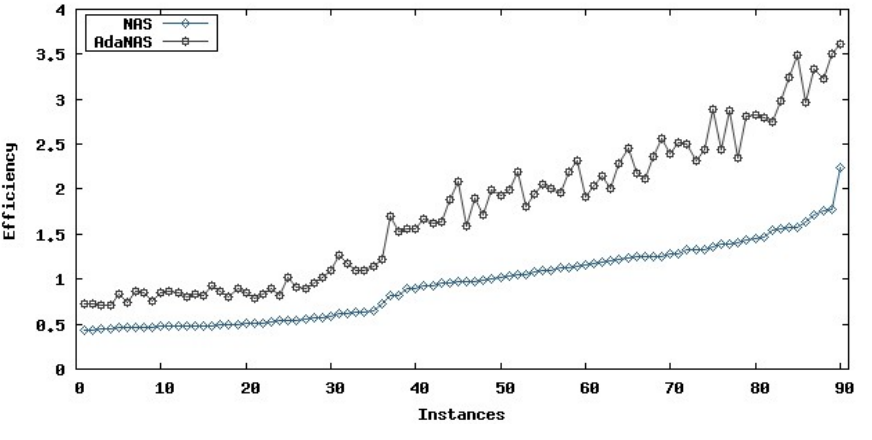


Fig. 5. Comparison between NAS and AdaNAS experimenting with 90 instances.

Figure 5 shows the results of the different experiments applied to NAS and AdaNAS on thirty runnings for each ninety different instances generated with the characteristics described in Section 6.1. It can been seen from it that on all the instances the AdaNAS algorithm outperforms NAS. On average, AdaNAS had an efficiency 81% better than NAS.

7 Conclusions

For the solution of SQRP, we proposed a novel algorithm called AdaNAS that is based on existing ant-colony algorithms. This algorithm incorporates parameters adaptive control techniques to estimate a proper TTL value for dynamic text query routing.

In addition, it incorporates local strategies that take advantage of the environment on local level, three functions were used to learn from past performance. This combination resulted in a lower hop count and an improved hit count, outperforming the NAS algorithm. Our experiments confirmed that the proposed techniques are more effective at improving search efficiency. Specifically the AdaNAS algorithm in the efficiency showed an improvement of the 81% in the performance efficiency over the NAS algorithm.

As future work, we plan to study more profoundly the relation among SQRP characteristics, the configuration of the algorithm and the local environment strategies employed in the learning curve of ant-colony algorithms, as well as their effect on the performance of hop and hit count measures.

References

[1] Adamic, L., Huberman, B.: Power-law distribution of the World Wide Web. Science 287(5461), 2115 (2000)

[2] Albert, R., Jeong, H., Barabási, A.: Error and attack tolerance of complex networks. Nature 506, 378–382 (2000)

[3] Amaral, L., Ottino, J.: Complex systems and networks: Challenges and opportunities for chemical and biological engineers. Chemical Engineering Scientist 59, 1653–1666 (2004)

[4] Androutsellis-Theotokis, S., Spinellis, D.: A survey of peer-to-peer content distribution technologies. ACM Computing Surveys 36(4), 335–371 (2004)

[5] Babaoglu, O., Meling, H., Montresor, A.: Anthill: An framework for the development of agent-based peer to peer systems. In: 22nd International Conference On Distributed Computing Systems. ACM, New York (2002)

[6] Barabási, A.: Emergence of scaling in complex networks, pp. 69–82. Wiley VHC, Chichester (2003)

[7] Barabási, A., Albert, R., Jeong, H.: Mean-field theory for scale-free random networks. Physical Review Letters 272, 173–189 (1999)

[8] Birattari, M.: The Problem of Tunning Mataheuristics as Seen From a Machine Learning Perspective. PhD thesis, Bruxelles University (2004)

[9] Bollobás, B.: Random Graphs, 2nd edn. Cambridge Studies in Advanced Mathematics, vol. 73. Cambridge University Press, Cambridge (2001)

[10] Costa, L., Rodríguez, F.A., Travieso, G., Villas, P.R.: Characterization of complex networks: A survey of measurements. Advances in Physics 56, 167–242 (2007)
[11] Cruz, L., Gómez, C., Aguirre, M., Schaeffer, S., Turrubiates, T., Ortega, R., Fraire, H.: NAS algorithm for semantic query routing systems in complex networks. In: DCAI. Advances in Soft Computing, vol. 50, pp. 284–292. Springer, Heidelberg (2008)
[12] DiCaro, G., Dorigo, M.: AntNet: Distributed stigmergy control for communications networks. Journal of Artificial Intelligence Research 9, 317–365 (1998)
[13] Diestel, R.: Graph Theory. Graduate Texts in Mathematics, vol. 173. Springer, New York (2000)
[14] Dorigo, M., Gambardella, L.M.: Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation 1(1), 53–66 (1997)
[15] Erdos, P., Rényi, A.: On the evolution of random graphs, vol. 2, pp. 482–525. Akademiai Kiad´o, Budapest, Hungary, 1976. First publication in MTA Mat. Kut. Int. Kozl. (1960)
[16] Faloutsos, M., Faloutsos, P., Faloutsos, C.: On power-law relationship on the internet topology. ACM SIGCOMM Computer Communication Review 29, 251–262 (1999)
[17] Gilbert, E.: Random graphs. Annals of Mathematical Statistics 30(4), 1141–1144 (1959)
[18] Glover, F., Kochenberger, G.: Handbook of Metaheuristics. International Series in Operations Research & Management Science, vol. 57. Springer, Heidelberg (2003)
[19] Glover, F., Laguna, M.: Tabú Search. Kluwer Academic Publishers, Dordrecht (1986)
[20] Goldberg, P., Papadimitriou, C.: Reducibility among equilibrium problems. In: Proceedings of the thirty-eighth annual ACM symposium on Theory of computing, pp. 61–70. ACM, New York (2005)
[21] Gummadi, K., Dunn, R., Saroiu, S., Gribble, S., Levy, H., Zahorjan, J.: Measurement, modeling and analisys of a peer-tp-peer file-sharing workload. In: 19th ACM Symposium on Operating Systems Principles. ACM, New York (2003)
[22] Holland, J.H.: Adaptation in natural and artificial systems. MIT Press, Cambridge (1992)
[23] Leibowitz, N., Ripeanu, M., Wierzbicki, A.: Deconstructing the kazaa network. In: 3rd IEEE Workshop on Internet Applications (2003)
[24] Liu, L., XiaoLong, J., Kwock, C.C.: Autonomy oriented computing — from problem solving to complex system modeling, pp. 27–54. Springer Science + Business Media Inc., Heidelberg (2005)
[25] Michlmayr, E.: Ant Algorithms for Self-Organization in Social Networks. PhD thesis, Vienna University of Technology (2007)
[26] Michlmayr, E., Pany, A., Kappel, G.: Using Taxonomies for Content-based Routing with Ants. In: Proceedings of the Workshop on Innovations in Web Infrastructure, 15th International World Wide Web Conference (WWW2006) (May 2006)
[27] Mihail, M., Saberi, A., Tetali, P.: Random walks with lookahead in power law random graphs. Internet Mathematics 3 (2004)
[28] Newman, M.E.J.: Power laws, pareto distributions and zipf's law. Contemporary Physics 46(5), 323–351 (2005)
[29] Ortega, R.: Estudio de las Propiedades Topológicas en Redes Complejas con Diferente Distribución del Grado y su Aplicación en la Búsqueda de Recursos Distribuidos. PhD thesis, Instituto Politécnico Nacional, México (2009)

[30] Ridge, E.: Design of Expirements for the Tuning of Optimization Algorithms. PhD thesis, University of York (2007)
[31] Ridge, E., Kudenko, D.: Tuning the Performance of the MMAS Heuristic in Engineering Stochastic Local Search Algorithms. In: Stützle, T., Birattari, M. (eds.) SLS 2007. LNCS, vol. 4638, pp. 46–60. Springer, Heidelberg (2007)
[32] Sakarayan, G.: A Content-Oriented Approach to Topology Evolution and Search in Peer-to-Peer Systems. PhD thesis, University of Rostock (2004)
[33] Tempich, C., Staab, S., Wranik, A.: REMINDIN: Semantic Query Routing in Peer-to-Peer Networks based on Social Metaphers. In: 13th World Wide Web Conference, WWW (2004)
[34] Wu, C.-J., Yang, K.-H., Ho: AntSearch: An ant search algorithm in unstructured peer-to-peer networks. In: ISCC, pp. 429–434 (2006)
[35] Wu, L.-S., Akavipat, R., Menczer, F.: Adaptive query routing in peer Web search. In: Proc. 14th International World Wide Web Conference, pp. 1074–1075 (2005)

Using Consecutive Swaps to Explore the Insertion Neighborhood in Tabu Search Solution of the Linear Ordering Problem

Francisco Eduardo Gosch Ingram, Guadalupe Castilla Valdez, and Héctor Joaquín Fraire Huacuja

Instituto Tecnológico de Ciudad Madero, México
1o. de Mayo y Sor Juana I. de la Cruz S/N C.P. 89440
Cd. Madero Tamaulipas, México
gosch8@hotmail.com, gpe_cas@yahoo.com.mx,
hfraire@prodigy.net.mx

Abstract. In this paper the linear Ordering Problem (LOP) is approached. This problem consists in to find an ordering of rows and columns of a matrix weights, such that the sum of all the values above the main diagonal is minimized. We propose in this ongoing research, increases the efficiency of exploration method in the insertion neighborhood in the state of the art Tabu search solution. The approach is evaluated on the broad set of standard instances that include the most difficult XLOLIB instances, from which the optima values are unknown. The results for instances which optimum values are known (OI), show that the proposed method has obtained reductions in execution time ranging between 21% and 97%, while, for the most difficult instances included in the set with unknown optima (BI), the reduction reaching 98%.Wilcoxon test is used to prove that the proposed method ITS, obtains similar % average error for OI instances than the reference method RTS, and a significance reduction in the average time. Now we are working in developing additional diversification strategies that take advantage of the savings in time to explore new regions of the search space.

Keywords: Linear Ordering Problem, Metaheuristic Algorithms, Tabu Search, Insertion Neighborhood Exploration.

1 Introduction

The Linear Ordering Problem (LOP) has an important application the economy field. Leontief economic model [12] represents a country's economy into sectors, and summarizes the interdependence between these sectors in the *Input-Output Tables*. The process of maximization of these tables is called the *triangulation problem*, and it provides important indicators to characterize the degree of stability of the economy of a country or region. This problem consist in simultaneously

P. Melin et al. (Eds.): Soft Comp. for Recogn. Based on Biometrics, SCI 312, pp. 267–276.
springerlink.com

permute the rows and columns of the *input-output table*, to maximize the sum of the values on the main diagonal, which is equivalent to the LOP problem. The linear ordering problem can be described as a combinatorial optimization problem as follows: "Given a matrix of weights $C\{i,j\}$, of size *n x n*, the problem is finding a permutation of columns and rows that maximizes the sum of the weights in the upper triangle of matrix" [7].

$$max\left(C_{(p)} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} C_{p_i p_j} \right) \tag{1.1}$$

Where the permutation p indicates the order in which columns and rows of C are considered, while p_i, p_j are the elements in i and j positions in the permutation, and the weight in the p_i row, p_j column of the matrix C is $C_{p_i p_j}$ The linear ordering problem is *NP-Hard* by its equivalence to the acyclic subdigraph problem and the minimum weight Feedback arc set problem [9] and [5].

2 Related Work

LOP has generated considerable research since 1958, when Chenery and Watanabe [4] described a heuristic which was developed in the sector of the economy. The sectors in input-output tables are ordered using a greedy algorithm in which sectors that have greater input into other sectors are first chosen within the permutation, where the contribution of the sectors is given by the equation:

$$a_i = \sum_{j=1}^{n} C_{ij} \tag{2.1}$$

Chenery successively builds a sort, choosing in each step the most attractive sector of the sectors that have not yet been selected.

Becker proposed another heuristic based on calculating quotas to assign a rank to each node in the graph. The proposed procedure is fast and produces reasonable results considering its simplicity. Today Becker algorithm is a good constructive algorithm for the linear ordering problem [1], based on a heuristic value q_i for each sector, this value is obtained by the following equation:

$$q_i = \frac{\sum_{j=1}^{m} C_{ij}}{\sum_{j=1}^{m} C_{ji}} \tag{2.2}$$

Chanas and Kobylanski (CK) [3] propose a method based on the symmetry property of the linear ordering problem. This property is observed in the matrix of weights and states that if the sum of the upper diagonal elements is maximized, then the sum of the elements below the diagonal will be minimized. This feature of symmetry of the problem is exploited in the CK method to escape local optimum. After several iterations of local search, when the process is trapped in a local optimum *S*, then applies an operation that reverse the solution *S*, resulting in *S**

which is the reverse permutation. Then, he applies the local search process to S^*, until the search process again becomes trapped in a local optimum. The process continues to invest the solution and repeating the local search, until it reaches a stop condition.

Although described heuristics achieve good results for the problem that is being studied, these have been superseded by methods based on metaheuristic algorithms, which implement strategies to escape from local optima. Several research works have focused on developing solutions for high-performance metaheuristics.

Prominent among them is the tabu metaheuristic with intensification and diversification developed by Laguna [1], who evaluated two procedures, one limited to structures of short-term memory and other components that incorporates long-term memory for diversification purposes, and a strategy of Path Relinking to enhance intensification. Starting with a random permutation, the short-term procedure alternates between phases of intensification and diversification. In the first phase explores elements of the permutation that are known a priori that when placed in other positions can produce solutions with better objective value. By contrast in diversification, preference is given to items that have not been explored for their little contribution in the objective function, but they help guide the search in space exploration to other regions that have not yet been reviewed.

Promising results achieved scatter search method (SS) implemented by Vicente Campos [2]. In this metaheuristic, population build solutions based on an iterative process of combining and improving operating on a set of reference solutions, and builds on concepts developed by Glover scatter search [10]. The results show that this method has a performance comparable to the methods of solution of the state of the art. Apply several concepts that can be integrated with other techniques for hybridization of metaheuristics [2].

Another outstanding solution for LOP, is the algorithm of variable neighborhood search, implemented by Garcia. This algorithm combines two neighborhoods to improve the efficiency of the process of exploring the search space. Garcia proposes a hybrid method, which binds the variable neighborhood tabu search with a short-term. In their research, Garcia and Perez-Brito show that the algorithm has a performance comparable to the best in the literature [8].

Tommaso Schiavinotto and Thomas Stützle [13] conducted a broad study of the search space for the available standard instances; the results show a fitness-distance correlation for many of the instances, directing the research towards the development of efficient metaheuristics that include adaptive reset. They applied iterated local search and memetic algorithms, and evaluated three neighborhoods for local search, one based on exchange movements, whose size is $n\ (n\text{-}1)\ /\ 2$, one based on insert movements, which has size $(n\text{-}1)^2$, and cost of exploration $O\ (n^3)$. Additionally, they propose a new strategy, to visit all elements in the insert neighborhood in $O\ (n^2)$, where the insert movements are made by applying the strategy used by Congram in Dynasearch [6], which is to implement an integral movement through a series of consecutive swap movements.

In this work we propose increases the efficiency of exploration method of the insertion neighborhood in the LOP Tabu solution, using consecutive swaps.

3 Proposed Method

The proposed method uses a Tabu search metaheuristic algorithm based on Tabu solution implemented by Manuel Laguna [8], which incorporates an improvement in the process of exploring the insertion neighborhood. In this work also is incorporated a strategy to allow a less expensive calculation of the value of the objective function. The improvement in the exploration is based on the integration through consecutive swaps proposed by Congram [6] and Schiavinotto [13].

Tabu search algorithm developed by Laguna, alternates between phases of intensification and diversification. The first phase is performed by a cycle of *best* movements that applies a *first* local search in the last iteration; this phase is reinforced by a path relinking strategy, which uses elite solutions to intensify the search into promising regions. Worth mentioning here, that a local search is applied to intermediate solutions once every four iterations. The diversification phase is done through a process of short-term memory based on frequency, in which sectors are moved toward positions that have been less employed along the search. This phase is complemented by a process of long-term memory inspired by the heuristic developed by Chanas and Kobylansky [3], which uses reverse permutation movements. This process consist in obtaining the average frequency of the positions of sectors on a set of elite solutions and then to apply reverse permutation, moving each sector to its symmetrical position to generate new solutions.

3.1 Main Idea

As described in the state of the art, the results obtained by Schiavinotto [13] show that exploration with insertion movements using consecutive swaps both in the memetic solution and in the iterated local search has a much lower computational cost that when they're performed by direct insertion. We propose that the results by incorporating this strategy in the Tabu solution method are in the same direction, because during the Tabu search process, the insertion movement is intensively applied.

3.2 Improving the Neighborhood Exploration

The insertion neighborhood is one of the most commonly used in the state of the art, and is the same as described by Laguna in his work about Tabu search for LOP [11]. This neighborhood is about moving the element located at position i to position j, as shown below:

$$Insertion(\pi, i, j) = \begin{cases} (\dots, \pi_{i-1}, \pi_{i+1}, \dots, \pi_j, \pi_i, \pi_{j+1}, \dots) \; i < j; \\ (\dots, \pi_{j-1}, \pi_i, \pi_j, \dots, \pi_{i-1}, \pi_{i+1}, \dots) \; i > j; \end{cases} \tag{3.1}$$

The computational cost for exploring this neighborhood is $O(n^3)$, however it is shown that the insertion of an element can be implemented using swap moves between consecutive elements, which leads to more efficient exploration of the search space, so that the cost is reduced to $O(n^2)$. This provides a great advantage

in their algorithm, because it uses intensively this feature. Under this lines the movement of insertion is shown, conducted through a series of swap adjacent moves:

$$Insertion(\pi, i, j) = \{\dots, \pi_{i-1}, \pi_{i+1}, \pi_{i+2}, \dots, \pi_{j-1}, \pi_j, \pi_i, \pi_{j+1}\} \tag{3.2}$$

$$= Swap(\pi, i, i+1), Swap(\pi, i, i+2), \dots, Swap(\pi, i, j-1), Swap(\pi, i, j)$$

$$= (.., \pi_{i+1}, i, \pi_{i+2}, \dots, \pi_j), (.., \pi_{i+2}, i, \dots, \pi_{j-1}, \pi_j, ..), \dots, (.., \pi_{j-1}, \pi_i, \pi_j, ..),$$

$$(.., \pi_{j-1}, \pi_j, \pi_i, ..)$$

On the other hand an operation inherent to calculate the objective function through savings is applied, avoiding recalculation of the contribution of all the elements of the permutation in the objective function value. This operation is expressed as follows:

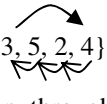
$$\Delta_I(\pi, i, j) = \left\{ \begin{array}{ll} \sum_{k=i+1}^{j} C_{\pi_k \pi_l} - C_{\pi_l \pi_k} & i < j; \\ \sum_{k=j}^{i-1} C_{\pi_l \pi_k} - C_{\pi_k \pi_l} & i > j \end{array} \right\} \tag{3.3}$$

The insertion by adjacent swaps is more efficient, because by doing an insertion movement from *i* to *j*, the intermediate movements of exchange are used to evaluate the intermediate solutions. The assessment of these intermediate solutions involves a lower cost in the calculation of the objective function considering only increases or decreases due to minor changes produced by each adjacent swap movement [13].

3.3 Example

The following example shows the insertion N_I ((1, 3, 5, 2, 4), 2, 5), by a simple insertion movement and through a sequence of swap movements.

Simple Insertion movement involves three shifts, and the insert operation.



N_I ({1, 3, 5, 2, 4}) → {1, 5, 2, 4, 3},

Insertion through swap movements of adjacent items involves three swap movements:

N_{IS} ({1, 3, 5, 2, 4}) → {1, 3, 5, 2, 4} + {1, 5, 3, 2, 4} + {1, 5, 2, 3, 4}

→ {1, 5, 2, 4, 3}

4 Experimental Results

Reference solution was coded in C++, using a Visual Studio framework 3.5, licensed with a student key. Experiments were done in a Dell Poweredge computer with an Intel Xeon dual-core processor 3.02GHz, with a 4 Gb Ram memory and 60 Gb HD. The code of the reference algorithm is executed for different sets of instances using a random initial seed based on the computer clock to generate the random numbers, in only one program execution, using 100 global iterations. The used instances were LOLIB, SGB, RandAI, RandAII, RandB, MB, XLOLIB and Spec., organized into two groups: OI, which includes instances where the optimal values are known, and BI, which include instances of which only the best solution is known. Further information on these instances is in [11] and [13].

Tables 1 and 2 show the comparative performance of the implemented tabu search algorithm (ITS) compared to the reference tabu search of the state of the art (RTS) [11]; results correspond to the instances OI. As can be seen, the performance of the reference algorithm implemented is similar to the state of the art Tabu algorithm. Based on the results of this initial assessment is concluded that the reference implementation can be relied upon to prove the exploration strategy of neighborhood integration in this reference we include the less expensive strategy to calculate the objective function value.

Tables 3 and 4, show the results of exploring the insertion neighborhood through consecutive swaps. Table 3 is a comparative of the performance for the OI instances and table 4 shows the performance of BI instances. It can be seen that RTS and ITS algorithm obtained a similar average error percentage. However, for the execution time there is a substantial improvement percentage with the proposed strategy, this is ranging between 47% for shorter instances (LOLIB), 97% for greater instances (MB), and for BI instances a reduction in time of 97% was obtained.

In addition, an experimental study based on the Wilcoxon nonparametric test is applied to compare the results of quality and efficiency of the evaluated algorithms. For this test, were performed 100 iterations of each algorithm for instances OI and BI, calculating the average error and the average time. Table 5 shows only the test parameter results for the six sets of instances included in OI: LOLIB, SGB, RandA2, RandB, MB and Spec. However, as BI only includes five sets of instances the Wilcoxon test cannot be applied.

First column in the table 5 contains the performance indicator used in the test, next three columns contain the main parameters of the Wilcoxon test and the last column contains the decision of acceptance or rejection of the null hypothesis H_0: $\mu_{RTS} = \mu_{ITS}$.

In this table, when R^+ is lower than R^-, indicate than proposed ITS is better than RTS, otherwise is worse. Although, when the lower value between R^+ and R^- is greater than the reference value in Wilcoxon table for the significance level α= 0.005 the results are not significant, and the null hypothesis is accepted, otherwise the hypothesis is rejected.

Table 1. Performance of the reference tabu solution (RTS).

Algorithm	Basic Tabu Search	TS + Path Re-linking	TS + Long Term Memory	Complete TS
Objective Function Value	22040159	22040160.9	22041257.7	22041261
Average error	0.04	0.04	0.001237	0.0007
# Optima	30	30	44	47
Time	0.33	0.54	0.67	0.93

Table 2. Performance of the implemented tabu solution (ITS).

Algorithm	Basic Tabu Search	TS + Path Re-linking	TS + Long Term Memory	Complete TS
Objective Function Value	22038469.4	22041169.6	22041253.5	22041262.6
Average error	0.0435	0.0177	0.0031	0.0003
# Optima	30	40	42	47
Time	0.263938	0.267632	0.547204	0.5477007

Table 3. Comparative summary of the performance of the reference tabu solution (RTS) and the implemented tabu solution (ITS) with swap insertion neighborhood for OI instances.

Instances	Performance indicators	RTS with Swap Insertion Neighborhood	ITS with Swap Adjacent Insertion Neighborhood	% Improvement
LOLIB	% Error	0.0000574	0.010668	-1.05
	#Optima	42	32	-
	Average Time	0.81148	0.06436	41.24
SGB	% Error	0.0000262	0.015542933	-1.52
	#Optima	11	4	-
	Average Time	3.1262	0.14684	72.20
RandA2	% Error	0.010296	0.018395	-4.59
	#Optima	9	3	-
	Average Time	3.81928	0.2356	74.36
RandB	% Error	0.020517	0.048236	-2.7
	#Optima	9	35	-
	Average Time	0.34552	0.04968	21.9
MB	% Error	0.000645	0.0079850	-0.72
	#Optima	14	1	-
	Average Time	67.897	0.8761	97.2
Spec	% Error	0.72956	0.72261	-0.4
	#Optima	21	23	-
	Average Time	1.1266	0.08844	48.78
% Average Error		0.12823	0.13724	-0.79
#Total Best Known		147	98	-
Total Average Time		12.8544	0.2435	91.02

Table 4. Comparative summary of the performance of the reference tabu solution (RTS) and the implemented tabu solution (ITS) with swap insertion neighborhood for BI instances.

Instances	Algorithm	RTS with Swap Insertion Neighborhood	ITS with Swap Adjacent Insertion Neighborhood	% Improvement
RandA1	% Error	0.38688	0.45104	-4.22
	# Best Known	1	0	-
	Average time	563.22836	7.40104	98.68
RandA2	% Error	0.01870	0.02965	-1.06
	# Best Known	1	0	-
	Average time	32.84614	0.84582	97.42
RandB	% Error	0.04786	0.17145	-10.5
	# Best Known	7	6	-
	Average Time	0.59995	0.005	99.16
XLOLIB	% Error	1.05663	1.12122	-6.25
	# Best Known	0	0	-
	Average Time	105.863103	1.82471	98.27
Spec.	% Error	0.257624	0.43753	-12.51
	# Best Known	2	2	-
	Average Time	70.8637143	2.00457	97.17
% Average Error		0.35354	0.44218	-6.14
# Total Best Known		11	8	-
Total Average Time		154.68025	2.41622	98.43

Table 5. Wilcoxon test for reference tabu solution (RTS) and the implemented tabu solution (ITS) with swap insertion neighborhood for OI instances.

RTS vs ITS	R^+	R^-	Wilcoxon Table Ref. Value	A (Accepted) R (Rejected)
% Average Error	20	1	0	A
Average Time	0	21	0	R

5 Conclusions and Future Work

In this paper, a strategy of improvement for the tabu search solution of the state of the art developed by Laguna [11] is implemented. This strategy reduces the cost of exploration in the insertion neighborhood from $O(n^3)$ to $O(n^2)$, by inserting through swaps of elements in adjacent positions, including the method that reduces the cost of calculating the value of the objective function using a matrix of differences that is calculated off-line.

These improvements are assessed for instances OI and BI. The results show that for instances OI, execution time reductions are obtained which represent improvements ranging from 21% for instances Rand B, up to 97% for MB instances, maintaining a similar performance in the quality of solutions than in the reference method. A similar situation but with more emphasis on the reduction occurs in

instances BI, for which significant reductions are obtained, representing an improvement over 90% for all instances of the group, however this entails a reduction in the quality of the solutions close to 6% compared to the results of the reference method. The Wilcoxon test obtained not significance differences in % average error for the evaluated algorithms, and significance differences in average time.

It can be seen that for both sets of instances a very substantial reduction in the execution time is achieved. The results suggest the possibility of combining these improvements with strategies focused on increasing diversification in the algorithm, investing the saved time in basic processes, in the exploration of new regions of search space.

Acknowledgments

Authors thank the support received from the Consejo Nacional de Ciencia y Tecnología (CONACYT) and the Consejo Tamulipeco de Ciencia y Tecnología (COTACYT), for the research reported in this paper. We want to thanks to Manuel Laguna and Abraham Duarte the source codes and the instances provided for this research work.

References

1. Becker, O.: Das Helmstädtersche Reihenfolge problem die Effizienz Verschiedener Näherungsverfahren in Computer uses in the Social Sciences. In: Berichteiner Working Conference, Wien (1967)
2. Campos, V., Glover, F., Laguna, M., Martí, R.: An experimental evaluation of a scatter search for the linear ordering problem. J. of Global Optimization 21, 397–414 (2001); Glover, F., Laguna, M.: Tabu Search. Kluwer Academic Publishers, Dordrecht (1997)
3. Chanas, S., Kobylanski, P.: A New Heuristic Algorithm Solving the Linear Ordering Problem, Computational Optimization and Applications (1996)
4. Chenery, H., Watanabe, T.: International Comparisons of the Structure of Production. Econometrica 26, 4 (1958)
5. Chiarini, B.: New algorithm for the triangulation of input-output tables and the linear ordering problem, University of Florida (2004)
6. Congram, R.: Polynomially Searchable Exponential Neighbourhoods for Sequencing Problems in Combinatorial Optimization, University of Southampton, Faculty of Mathematical Studies, PhD Thesis, EEUU (2000)
7. Duarte, A., Pantrigo, J., Gallego, M.: Metaheurísticas. Madrid, España, Dykinson (2007)
8. Garcia, C.G., Pérez-Brito, D., Campos, V., Martí, R.: Variable neighborhood search for the linear ordering problem. Computers and Operations Research 33, 3549–3565 (2006)
9. Garey, M.R., Johnson, D.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Co., New York (1975)

10. Glover, F.: Tabu search and adaptive memory programming-Advances, applications and challenges. In: Barr, R.S., Kennington, J.L. (eds.) Interfaces in Computer Science and Operations Research: Advances in Metaheuristics, Optimization and Stochastic Modeling Technologies, pp. 1–75. Kluwer, Boston (1997)
11. Laguna, M., Martí, R., Campos, V.: Intensification and Diversification with Elite Tabu Search Solutions for the Linear Ordering Problem. Computers and Operations Research 26, 1217–1230 (1999)
12. Leontief, W.: Input-Output Economics. Oxford University Press, New York (2004)
13. Schiavinotto, T., Stützle, T.: The linear ordering problem: Instances, search space analysis and algorithms. Journal of Mathematical Modelling and Algorithms 3(4), 367–402 (2004)

A New Optimization Method Based on a Paradigm Inspired by Nature

Leslie Astudillo, Patricia Melin, and Oscar Castillo

Tijuana Institute of Technology, Tijuana México
leslie.astudillo@suntroncorp.com, pmelin@tectijuana.mx, ocastillo@tectijuana.mx

Abstract. In this paper, we propose a new optimization method for soft computing problems, which is inspired on a nature paradigm: the reaction methods existing on chemistry, and the way the elements combine with each other to form compounds, in other words, quantum chemistry. This paper is the first approach for the proposed method, and it presents the background, main ideas, desired goals and preliminary results in optimization.

1 Introduction

Several works have proved the relevance of computing techniques to solve diverse kinds of problems, including forecasting, control and pattern recognition among others [14][15][16].

These techniques not only comply with their objective, but they also promote the creation of new ways to give solutions and improve the actual methods as well [5][6][7].

One of the main difficulties when designing the structure of a solution method is the tuning of the parameters; which are the key to the success of these applications. These parameters will vary depending on the complexity of the problem and the method used to find the solution; and in some cases, they stem from our own ability to conceptualize the problem itself, taking in account, the inputs of the system and the expected output values.

Due to these facts, several optimization strategies based on nature paradigms have arisen. From Ant Colony Optimization, to Particle Swarm Optimization among others, these strategies had emerged as an alternative way to solve problems [3][4][8][9][10][12][13].

For this work, we will be observing the process in which the different elements existing in nature are created, behave and interact with each other to form chemical compounds.

The structure of this paper is the following. Section 2 shows a brief description of the chemical method that inspired this investigation; section 3 describes the

P. Melin et al. (Eds.): Soft Comp. for Recogn. Based on Biometrics, SCI 312, pp. 277–283.
springerlink.com

proposed method and first approach; section 4 shows the preliminary experiment results; in section 5 we describe the current and future work and section 6 shows some references.

2 Chemical Paradigm

In order to have a better understanding of the process that we intend to model, we present some general definitions [1][2].

Chemistry is the study of matter and energy and the interaction between them, including the composition, the properties, the structure, the changes which it undergoes, and the laws governing those changes. A substance is a form of matter that has a defined composition and characteristic properties. There are two kinds of substances: elements and compounds.

An element is a substance that cannot be broken down into simpler substances by ordinary means. It is apparent from the wide variety of different materials in the world that there are a great many ways to combine elements.

Compounds are substances formed by two or more elements combined in definite proportions through a chemical reaction. There are millions of known compounds, and thousands of new ones are discovered or synthesized each year.

A chemical reaction is a change in which at least one substance changes its composition and its sets of properties; they are classified into 4 types.

Type 1: combination reactions: (B+C → BC).
A combination reaction is a reaction of two reactants to produce one product. The simplest combination reactions are the reactions of two elements to form a compound. After all, if two elements are treated with each other, they can either react or not.

Type 2: decomposition reactions: (BC → B+C).
The second type of simple reaction is decomposition. This reaction is also easy to recognize. Typically, only one reactant is given. A type of energy, such as heat or electricity, may also be indicated. The reactant usually decomposes to its elements, to an element and a simpler compound, or to two simpler compounds.

Binary compounds may yield two elements or an element and a simpler compound. Ternary (three-element) compounds may yield an element and a compound or two simpler compounds. These possibilities are shown in the Figure 1.
Type 3: substitution reactions: (C + AB → AC + B).
Elements have varying abilities to combine. Among the most reactive metals are the alkali metals and the alkaline earth metals. On the opposite end of the scale of reactivities, among the least active metals or the most stable metals are silver and gold, prized for their lack of reactivity. Reactive means the opposite of stable, but means the same as active.

When a free element reacts with a compound of different elements, the free element will replace one of the elements in the compound if the free element is more reactive than the element it replaces. In general, a free metal will replace the metal in the compound, or a free nonmetal will replace the nonmetal in the compound. A new compound and a new free element are produced.

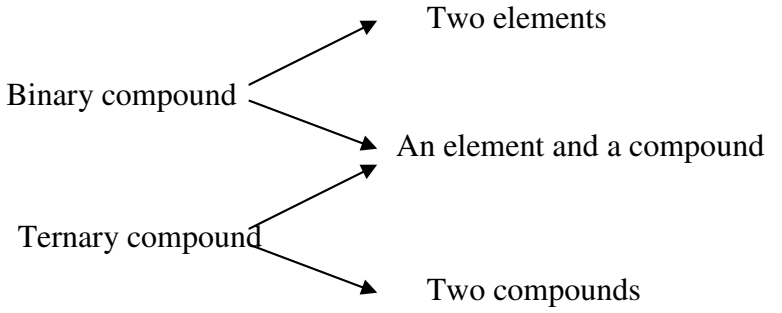


Fig. 1. Decomposition possibilities

Type 4: double-substitution reactions: (AB + CD → CB + AD).
Double-substitution or double-replacement reactions, also called double-decomposition reactions or metathesis reactions, involve two ionic compounds, most often in aqueous solution. In this type of reaction, the cations simply swap anions. The reaction proceeds if a solid or a covalent compound is formed from ions in solution. All gases at room temperature are covalent. Some reactions of ionic solids plus ions in solution also occur. Otherwise, no reaction takes place.

Just as with replacement reactions, double-replacement reactions may or may not proceed. They need a driving force. In replacement reactions the driving force is reactivity; here it is insolubility or covalence.

3 Modeling the Chemical Paradigm

Now that we have described the natural paradigm that we intent to mimic, the next step is to define the general structure of our optimization algorithm; which, initially will be developed in 5 phases: a combination algorithm, a decomposition algorithm, a substitution algorithm, a double-substitution algorithm and the final algorithm, which will be the combination of all the previous four.

The steps to consider in this optimization method will be as follows:

1. First, we need to generate an initial pool of elements/compounds.
2. Once we have the initial pool, we have to evaluate it.
3. Based on the previous evaluation, we will select some elements/compounds to "induce" a reaction.
4. Given the result of the reaction, we will evaluate the obtained elements/compounds.
5. Repeat the steps until the algorithm meets the criteria (desired result or maximum number of iterations is reached).

In order to start testing the phases of the algorithm, we will be applying these to the following (but not restricted to) functions: De Jong's and Rosenberg's functions [11][12].

4 Preliminary Experimental Results

Figure 2 shows the De Jong's first function also called *the sphere model*, which is continuous, convex, unimodal and is represented by the equation:

$$f_1(x) = \sum_{i=1}^{n} x_i^2 \,. \tag{1}$$

The domain is given by:

$$-\infty \le x_i \le \infty \,, \tag{2}$$

And has a global minimum represented by:

$$f(x) = 0; \;\; x(i) = 0; \;\; i = 1 : n. \tag{3}$$

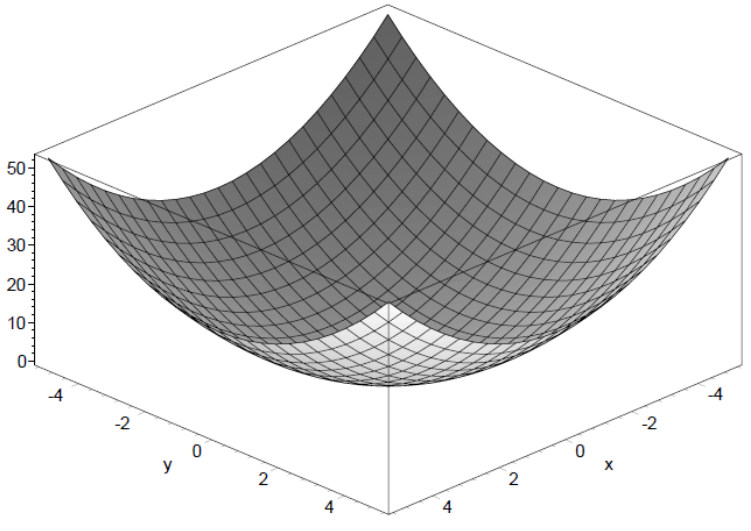


Fig. 2. De Jong's Function

The fist approach to solve this equation is given by applying a minimization algorithm based on the decomposition reactions.

The main idea of this particular algorithm is, given a random set of initial numbers, decompose each one into smaller numbers in a way that can be represented by a binary tree, where each ensuing node will be decomposed as well into smaller numbers, to lead the result into the minimum of the function.

To start from the simplest option, in these early experiments all decomposed elements are considered to have the same value, and they are given by:

$$\text{Decomposed\_Element}(n) = \text{Element}/m. \tag{4}$$

Where n is the element index and m is the number of decomposed elements generated.

Because the resulting values are the same for each decomposed element, only one will be selected to be evaluated in the function.

Let's consider an initial pool of 5 elements (randomly generated); each one will be decomposed in 3 sub elements throughout 10 iterations.

Table 1 shows the final and average values of the best and worst result reached by the algorithm throughout 30 experiments.

Table 2 shows the standard deviation calculated by iteration throughout 30 experiments.

Figure 3 shows the minimized values trough the 10 iterations of experiment number 2.

In Figure 4 we can see the behavior of the algorithm along the 30 experiments, where every experiment is represented by "Cycles" of 10 iterations each.

Table 1. Worst and best results throughout 30 experiments evaluating the first De Jong's Function.

Experiment No.	Minimum value	Average value	Comments
1	1.271e-06	50.38	Worst result
2	3.22e-10	0.013	Best result

Table 2. Standard deviation per trial in 30 experiments evaluating the first De Jong's Function.

Trial	Standard Deviation
1	154.889096154112
2	17.2098995726791
3	1.91221106363101
4	0.212467895959001
5	0.0236075439954446
6	0.00262306044393829
7	0.000291451160437587
8	3.23834622708430e-05
9	3.59816247453812e-06
10	3.99795830504235e-07

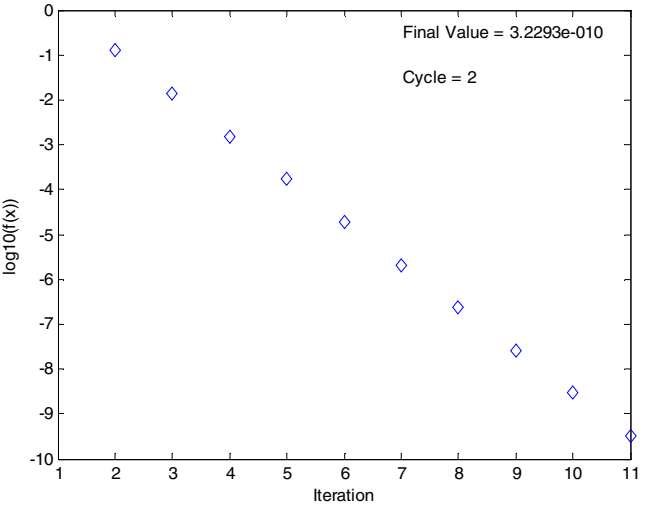


Fig. 3. Minimum value reached in experiment no. 2.

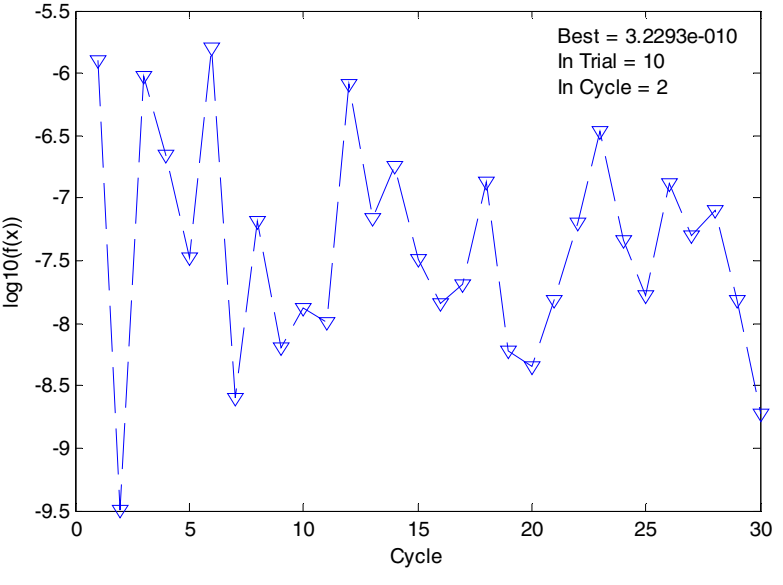


Fig. 4. Minimum values reached per experiment in 30 cycles.

5 Conclusions

In this paper, we introduced the first approach to a new optimization method that tries to mimic the way the elements combine with each other to achieve a chemical

compound. The first De Jong's function also called *the sphere model* was used to evaluate the first development phase of the optimization algorithm. At the time, more functions are being evaluated to pursue the tuning of the algorithm itself.

Acknowledgment

The authors would like to thanks CONACYT and Tijuana Institute of Technology for the facilities and resources granted for the development of this research.

References

[1] Chang, R.: General Chemistry, 5th edn. McGraw-Hill, New York (2004)
[2] Goldberg, D.: Schaum's Outline of Beginning Chemistry, 3rd edn. Schaum's Outline Series. McGraw-Hill, New York (2009)
[3] Man, K.F., Tang, K.S., Kwong, S.: Genetic Algorithms: Concepts and Designs. Springer, Heidelberg (1999)
[4] Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan, pp. 39–43 (1995)
[5] Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, pp. 1942–1948 (1995)
[6] Valdez, F., Melin, P.: Parallel Evolutionary Computing using a cluster for Mathematical Function Optimization, San Diego CA, USA, June 2007. Nafips, pp. 598–602 (2007)
[7] Fogel, D.B.: An introduction to simulated evolutionary optimization. IEEE transactions on neural networks 5(1), 3–14 (1994)
[8] Goldberg, D.: Genetic Algorithms. Addison-Wesley, Reading (1988)
[9] Angeline, P.J.: Using Selection to Improve Particle Swarm Optimization. In: Proceedings of the World Congress on Computational Intelligence, pp. 84–89. IEEE, Anchorage (1998)
[10] Montiel, O., Castillo, O., Melin, P., Rodriguez, A., Sepulveda, R.: Human evolutionary model: A new approach to optimization. Inf. Sci. 177(10), 2075–2098 (2007)
[11] GEATbx: Example Functions (single and multi-objective functions), http://www.geatbx.com/docu/fcnindex-01.html#P89_3085
[12] Haupt, R.L., Haupt, S.E.: Practical Genetic Algorithms, 2nd edn. Wiley Interscience, Hoboken (2004)
[13] Rotar, C.: A New Evolutionary Algorithm for Multiobjective Optimization Based on the Endocrine System. In: Proceedings of the International Conference on Theory and Applications of Mathematics and Informatics – ICTAMI, Alba Iulia (2003)
[14] Hidalgo, D., Melin, P., Licea, G.: Optimization of Modular Neural Networks with Interval Type-2 Fuzzy Logic Integration Using an Evolutionary Method with Application to Multimodal Biometry. In: Bio-inspired Hybrid Intelligent Systems for Image Analysis and Pattern Recognition, pp. 111–121 (2009)
[15] Astudillo, L., Castillo, O., Aguilar, L.: Hybrid Control for an Autonomous Wheeled Mobile Robot Under Perturbed Torques. In: Melin, P., Castillo, O., Aguilar, L.T., Kacprzyk, J., Pedrycz, W. (eds.) IFSA 2007. LNCS (LNAI), vol. 4529, pp. 594–603. Springer, Heidelberg (2007)
[16] Melin, P., Mancilla, A., Lopez, M., Solano, D., Soto, M., Castillo, O.: Pattern Recognition for Industrial Security Using the Fuzzy Sugeno Integral and Modular Neural Networks. In: Advances in Soft Computing, vol. 39(1), pp. 105–114 (2007)

Improvement of the Backpropagation Algorithm Using (1+1) Evolutionary Strategies

José Parra Galaviz, Patricia Melin, and Leonardo Trujillo

Instituto Tecnológico de Tijuana, Tijuana, BC, México
galavizjpg@gmail.com, pmelin@tectijuana.mx,
leonardo.trujillo.ttl@gmail.com

Abstract. Currently, the standard in supervised Artificial Neural Networks (ANNs) research is to use the backpropagation (BP) algorithm or one of its improved variants, for training. In this chapter, we present an improvement to the most widely used BP learning algorithm using (1+1) evolutionary Strategy (ES), one of the most widely used artificial evolution paradigms. The goal is to provide a method that can adaptively change the main learning parameters of the BP algorithm in an unconstrained manner. The BP/ES algorithm we propose is simple to implement and can be used in combination with various improved versions of BP. In our experimental tests we can see a substantial improvement in ANN performance, in some cases a reduction of more than 50% in error for time series prediction on a standard benchmark test. Therefore, we believe that our proposal effectively combines the learning abilities of BP with the global search of ES to provide a useful tool that improves the quality of learning for BP-based methods.

1 Introduction

Over the last two decades, Artificial Neural Networks (ANNs) have gained great acceptance as a powerful method for learning and pattern recognition. In essence, ANNs are mathematical and computational models for information processing that base their core functionality on an abstract representation of the central nervous systems of biological organisms [15]. Currently, there are many types of ANNs, but probably the most widely-known is the multilayer perceptron (MLP), which normally uses a feed-forward and fully connected architecture [15]. After choosing a desired network topology, a MLP must be assigned the correct set of connection weights in order for it to be able to perform a useful task. This is achieved by using some sort of learning method, of which the most common is the backpropagation (BP) algorithm [14].

The BP algorithm is a gradient descent method that propagates the mean square error from the output layer to the input layer taking into account all hidden layers. For years, this algorithm has been considered the standard approach towards learning in supervised ANN literature. However, it is well-known that the BP is hampered by three noteworthy shortcomings. First, the BP suffers from several

P. Melin et al. (Eds.): Soft Comp. for Recogn. Based on Biometrics, SCI 312, pp. 287–302.
springerlink.com

learning problems, such as overtraining and a slow convergence. Second, it can lead to network paralysis where the algorithm in unable to significantly modify the connection weights. And third, it often gets trapped in local minima, a common problem in gradient-based methods.

In order to overcome these shortcomings, several improvements to the basic BP algorithm have been proposed [15, 6, 9], which basically attempt to add mechanisms by which the training process can proceed in a smoother and less constrained manner. These algorithms use methods that can dynamically change certain parameters of the BP algorithm or add new ones that improve the convergence. For instance, it is possible to dynamically change the learning rate parameter using what is known as the *gradient descent with adaptive learning rate* algorithm. Another example is the g*radient descent with momentum* algorithm, which adds a new parameter that allows for a smoother convergence during learning. Moreover, it is also possible to combine these methods in order to produce a more robust learning process.

The improved versions of the BP algorithm have in fact proven to be superior to the basic BP, and are the norm in MLP training. Nevertheless, one drawback of these methods is that they tend to require several ad-hoc decisions in order to correctly parameterize them and successfully apply them in real-world problems. Furthermore, while the new parameters these methods introduce allow adaptive changes during training, these parameters themselves remain constant throughout the entire process. This constrains the manner in which the learning parameters can be adapted to fit the current error surface.

In order to overcome this, some researchers have turned to evolutionary algorithms in order to search for an optimal set of connection weights, thereby completely avoiding gradient-based learning [21]. However, we feel that such an approach does not take advantage of the local optimization capabilities of gradient-based methods. Therefore, we propose a hybrid approach that combines the learning improvements of previously proposed methods with an evolutionary search process.

The proposal we develop dynamically modifies the parameters used by the BP algorithm, in an unconstrained manner using a (1+1) evolutionary strategy. We hypothesize that by doing so the learning process will converge to a better set of connection weights, and that this will be reflected in test errors when applied to a benchmark problem. In this chapter, the above claim is supported by several experimental tests that provide a proof of concept of our proposal.

The remainder of this chapter is organized as follows. Section 2 contains a brief introduction to supervised ANNs, describes the BP algorithm and some of the most widely used improved variants. Then, in Section 3 we present the problem we address in our work and introduce the proposal we have developed using evolutionary computation. Afterwards, the experimental results are given in Section 4. Finally, Section 5 contains summary and concluding remarks.

2 Background and Basic Concepts

This section contains a brief introduction to ANNs, the MLP and the BP algorithm.

2.1 Artificial Neural Networks

An artificial neural network is a mathematical and computational model that simulates the abstract structure and functional aspects of biological neural networks. The basic computational elements in an ANN are known as *neurons*, nodes that serve as inputs, outputs or internal processing units. These neurons communicate and pass signals among themselves using what are known as *synapses*, which are modulated by weight parameters. The manner in which neurons are organized and connected define the architecture of the network. Here, we will focus on the feed-forward and fully connected architecture used by the MLP. In this network, the information moves in only one direction, forward, from the input nodes, though the hidden nodes (if any) to the output nodes, there are no loops in a network of this type. A graphical illustration of this basic architecture is depicted in Figure 1.

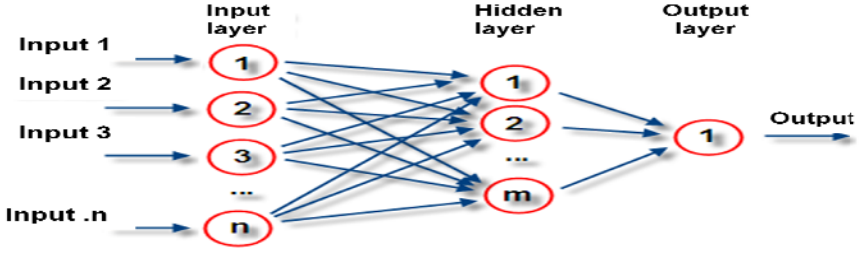


Fig. 1. The multilayer perceptron (MLP) with feed-forward architecture. It shows that it has one input layer of neurons, one output layer, and one or more hidden layers. Notice that these neurons of each layer are fully connected with the neurons of the next one.

2.2 The Backpropagation Algorithm (BP)

As mentioned above, the most common way to set the connection weights of a MLP is to use the BP algorithm. The term *backpropagation* is an abbreviation for "backwards propagation of errors", which is the manner in which the learning process is carried out. Because it is a gradient-based method it requires that the neurons use continuous and differentiable activation functions, such as a linear or sigmoid function. The algorithm associates an error surface to the network and searches for a steady state of minimal error through a descent path over the error surface. It feeds back the system error across the network and uses it to modify the weights in a manner that is proportional to the gradient of the decreasing error function [14].

2.2.1 Description of Backpropagation Algorithm

The BP algorithm can be summarized in four fundamental steps that are repeated iteratively, these are:

1. - Initialize the connection weights with random values.
2. - Compute the output of the ANN, denoted by y_p, by propagating each input pattern x_p through the network in a forward direction.

3. - Compute the error between the desired output, d, and the output produced by the ANN *y*, this is given by $E_p = \left(d_{pln} - y_{pln}\right)^2$

4. - Adjust the connection weights according to the following rule,

$$w_{ln[t+1]} = w_{ln[t]} + \beta\delta_{pln}x_{pl} \quad (1)$$

In the above equation β represents the learning rate, a parameter that is used to modulate the amount by which the connection weights can be modified at each iteration of the algorithm. The network δ_{pln} is computed in two ways. First, if n is an output node then it is given by $\left(d_{pln} - y_{pln}\right)$.

Otherwise, it is given by $f\,(y_{pln}) \bullet \delta_{pln} + 1\; w_l + 1$, where f represents the activation function of node *n*.

The above process is repeated until a stopping criterion is met, which can be a desired minimum error or a maximum number of iterations, also known as epochs.

2.3 Main Shortcomings of Backpropagation Algorithm

Despite the success of BP learning, there are some aspects which make the algorithm not guaranteed to be universally useful. Most troublesome is the long training times that it often requires, and others are also noteworthy, such as:

- Network paralysis: The BP can produce extremely large connection weights which, cannot be effectively modified by the constant update rule used during training.
- Local minimum: The error surface for an ANN normally contains many peaks and valleys. Given that BP is gradient-based, then it is common for the algorithm to fall into local optima.
- Over-training: It is known that the BP algorithm will tend to produce ANNs that are over-fitted to the training examples, and will not respond correctly to unseen patterns.

2.4 Improvements to the Backpropagation Algorithm

Given the above problems that are often encountered with BP, many researchers have proposed improvements to the canonical algorithm that are meant to overcome them [8] [5]. In what follows, we will give a brief overview of some of the more well-known proposals in ANN literature.

2.4.1 Gradient Descent with Adaptive Learning Rate (GDA)

One of the first proposed improvements to BP learning was aimed at adaptively modifying the learning rate on-line during training [6]. In the standard BP the learning rate is constant during the entire training process, it is thus imperative to choose a correct value at the beginning. For instance, if the learning rate is set too high the algorithm may oscillate and become unstable. Conversely, if the learning rate is too small, then convergence will be quite slow. It is known that setting an optimal value for β is not a trivial task. Moreover, the optimal value might change

during the training process. Therefore, if the learning rate is allowed to change during training this might help improve the quality of the learning process. The idea is to make the learning rate responsive to the complexity of the local error surface.

In order to implement this idea, the GDA algorithm modifies BP in the following ways. First, the initial network output and error are calculated. At each epoch new weights are calculated using the current learning rate. New outputs and errors are then calculated, if the new error exceeds the old error by more than a predefined ratio the new weights are discarded and the learning rate is decreased by a constant amount, we can call this parameter $\beta_{\Delta-}$. Otherwise, the new weights are kept, and if the new error is less than the old error, the learning rate is increased by a constant parameter $\beta_{\Delta+}$ .Therefore, if a larger learning rate could result in stable learning, the learning rate is increased. On the other hand, if the learning rate is too high to guarantee a decrease in error, then it is decreased until stable learning resumes [9].

2.4.2 Gradient Descent with Momentum (GDM)

Another proposed improvement is the GDM algorithm, which takes a different approach towards overcoming some of the shortcomings of BP. It is equivalent to the traditional BP algorithm, with an added parameter called the momentum coefficient, which is used when the connection weights are updated [6].

In GDM, the weight update takes into account previous weight changes along with the current propagated error. It can be understood as an averaging process that does not allow for weight changes that are extremely different relative to the previous ones. The update rule proposed in GDM is given by

$$\Delta w_m = \alpha \Delta w_{m-1} - (1-\alpha)\, \varepsilon d_m^w \,, \tag{2}$$

where α is the momentum coefficient, and Δw_{m-1} is the weight change used in the previous epoch. In GDM the momentum coefficient α is held constant throughout the entire training process, usually set to 0.9.

2.4.3 Gradient Descent with Momentum and Adaptive Learning Rate (GDX)

Based on the previous methods, GDA and GDM, the gradient descent algorithm with momentum and adaptive learning rates was proposed (GDX), which combines the advantages of both BP improvements [6]. For some problem, the combined effect of the momentum coefficient and adaptive learning rate has achieved a substantial performance when compared with the use of each of them independently.

3 Problem Description and Proposed Method

We have outlined three of the most common improvements to the BP learning algorithm, GDA, GDM and GDX, where the last one is a combination of the other two. In GDA, the goal is to adapt the value of the learning rate parameter based on the current error surface that is encountered during training. On the other hand, in

GDM the goal is to smooth the manner in which the algorithm can change the connection weights by taking into account the previously executed weight changes. However, by addressing these issues, they raise others. For instance, in GDA the learning rate is either increased or decreased by a constant factors $\beta_{\Delta+}$, and $\beta_{\Delta-}$. One could argue that the value of these parameters should also be subject to an adaptive process during training. Furthermore, in GDM the momentum coefficient is held constant, and there is no a priori reason to assume that such a strategy is optimal.

Therefore, in this work we hypothesize that a better learning strategy would be able to adaptively modify all of the main parameters of the algorithm in unconstrained manner. This follows from the basic argument behind the GDA method, where it is assumed that because the error surface changes during training then the optimal learning rate should also change. Therefore, we argue that the same logic must hold for parameters such as the learning rate step parameters ($\beta_{\Delta+}$, and $\beta_{\Delta-}$) and the momentum coefficient. For example, in GDA the learning rate should be able to increase or decrease during training without the need of constant step sizes, and in GDM the momentum can also be adaptively modified.

We propose an improvement to BP training that can dynamically change the main parameters of the training algorithm without the need for constant step values. In order to achieve this we develop our proposal using a hybrid algorithm that combines an evolutionary search process and the basic BP algorithm. Concretely, we use evolutionary strategies as a global search method that adapts the main learning parameters during training, and allows the gradient descent algorithm to perform a local search over the ANN error surface. Our proposal can accommodate the basic BP algorithm as well as any of the previously proposed improvements (GDA, GDM and GDX) without requiring substantial modifications. In order to contextualize our contribution, in what follows we provide a brief introduction of how evolutionary computation relates to ANN research.

3.1 Evolutionary Computation and ANNs

Evolutionary computation encompasses a large variety of global search and optimization methods that are based on abstract models of Darwinian evolutionary theory. Some of the most widely known paradigms are genetic algorithms, evolutionary strategies and genetic programming, all of which are based on similar conceptual principles [2]. These methods have proven to be quite robust and flexible, applicable to a large variety of problem domains.

In the case of ANNs, many attempts have been made to use evolutionary methods in order to optimize a specific characteristic of network. In a survey on this topic, X. Yao [21] proposes the following taxonomy of works developed in this line of research, see Figure 2. He observes that evolutionary methods have been applied in three main areas : (1) as a tool used to find the connection weights of an ANN [10, 11, 17, 18, 20]; (2) in order to find the optimal ANN topology [4, 22, 23, 24]; and (3) in order to develop improvements to the learning process [1, 7 , 12, 13, 19]. In the last group, most works have focused on optimizing the BP parameters off-line, in order to set optimal values that are used throughout the entire

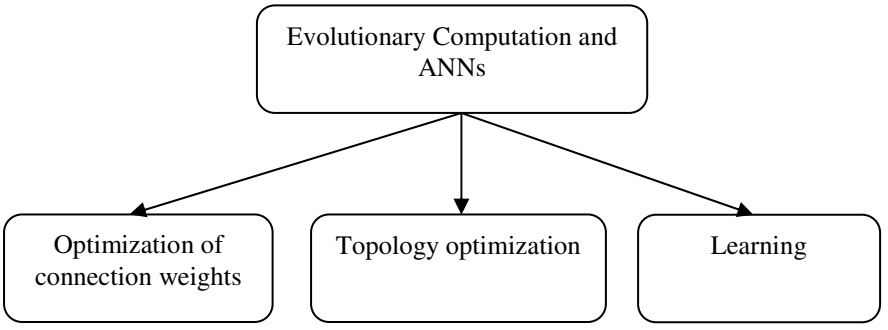


Fig. 2. Areas where evolutionary computation have been applied in the optimization of an ANN.

learning process. In our work, however, we are interested in developing an adaptive strategy similar to what is done in GDA, with an additional evolutionary search that allows for dynamic modifications of the BP parameters.

Before describing our proposal, let's briefly review the main characteristics of the evolutionary strategies search paradigm.

3.2 Evolutionary Strategies

Evolutionary strategies (ES) are an optimization technique based an abstract model of biological evolution [16]. They use a solution representation that depends on the problem domain; usually they are used for real-valued parameter optimization. In a canonical ES only one operator is used to generate offspring, a Gaussian mutation that perturbs the value of each parameter of an individual solution. There are two basic types of ESs, $\mu + \lambda$ and (μ,λ), where μ is the number of individuals in a population and λ is the number of offspring that are generated. In $(\mu + \lambda)$ ES, the individuals contained in the next generational loop are chosen from the best solutions from both the past population and the group of offspring. Conversely, in the (μ,λ) the λ offspring replace the same number of individuals in the previous population of m individuals. In other words, $(\mu + \lambda)$ is an elitist strategy while (μ,λ) is not [3].

In our work, we use a (1+1) ES as the basis for our improved BP training, which is the most basic form of this evolutionary paradigm. The main steps of this evolutionary search method can be summarized as follows:

1. - Randomly generate a search point $x \in \{0, 1\}^n$ using a uniform distribution over the search space.
2. - Create offspring y by mutating each element of x using a Gaussian operator with zero mean and σ standard deviation.

3. - The offspring ***y*** replaces its parent x if f*(y)* > *f(x)* (for a maximization problem).
4. - If the termination criterion is met then stop and return *x*, otherwise go to step 2.

3.3 Evolutionary Strategies for Backpropagation Learning

As stated above, our proposal is to combine the BP algorithm with an evolutionary search method. The goal is to provide a mechanism by which the main parameters of a BP learning process can be adaptively modified on-line during network training using variable step sizes. In order to achieve this we have chosen the (1+1) ES, for the following reasons:

- It is a well-known and understood search method.
- It is particularly well suited for real-valued parameter optimization, which is the case in BP training.
- It is very simple and easy to implement, which allows us to maintain the basic structure of BP unchanged. From this it follows that the method will not dramatically increase the computational cost of training an ANN.

Therefore, we propose a hybrid learning process, such that a (1+1) ES adaptively changes the BP parameters after a specified number of epochs, during which the BP training algorithm carries out the standard weight updates; the flow-chart of the algorithm is depicted in Figure 3.

The algorithm proceeds as follows. First, we generate a parameter vector *x* with standard initial values. In this case, the number of parameters depends on the version of BP we choose to use. For instance, if we use GD or GDA then x would contain only the learning rate. On the other hand, if we use GDM or GDX, then x would contain the learning rate and the momentum coefficient. Afterwards, we randomly generate the initial connection weights for our ANN which we call Ax, just as it would be done in traditional BP learning. This leads us to the first generation of our (1+1) ES.

In the evolutionary loop, we first create a mutated version of *x*, which we call *y*, using a Gaussian mutation with the same σ for all elements. Then, we make a copy of Ax, call this Ay. We train Ax using the BP parameters specified in *x* for a total of epochs, and the same is done for Ay with parameter values *y*. After training both networks we obtain corresponding convergence error from each, call these Ex and Ey respectively. The error values are used to determine which ANN and which parameter vector will survive for the following generation. This process is repeated until one of two conditions is met: (1) the total number of generations is reached; or (2) the goal error for the ANN is achieved.

In this process we are introducing two parameters to the learning process. One is the number of epochs per generation denoted by τ. The other is the value of the standard deviation σ sigma of the Gaussian mutation operator. In this work we have chosen these parameters based on a trial-and-error process.

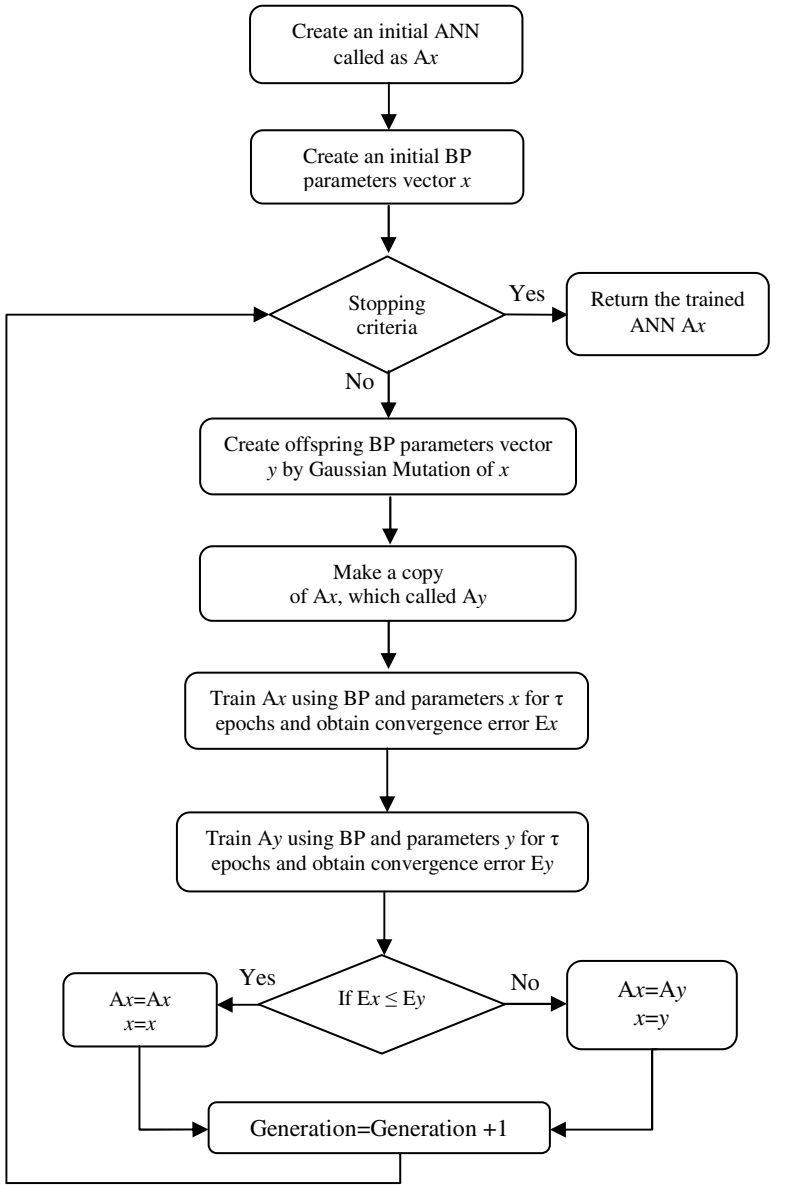


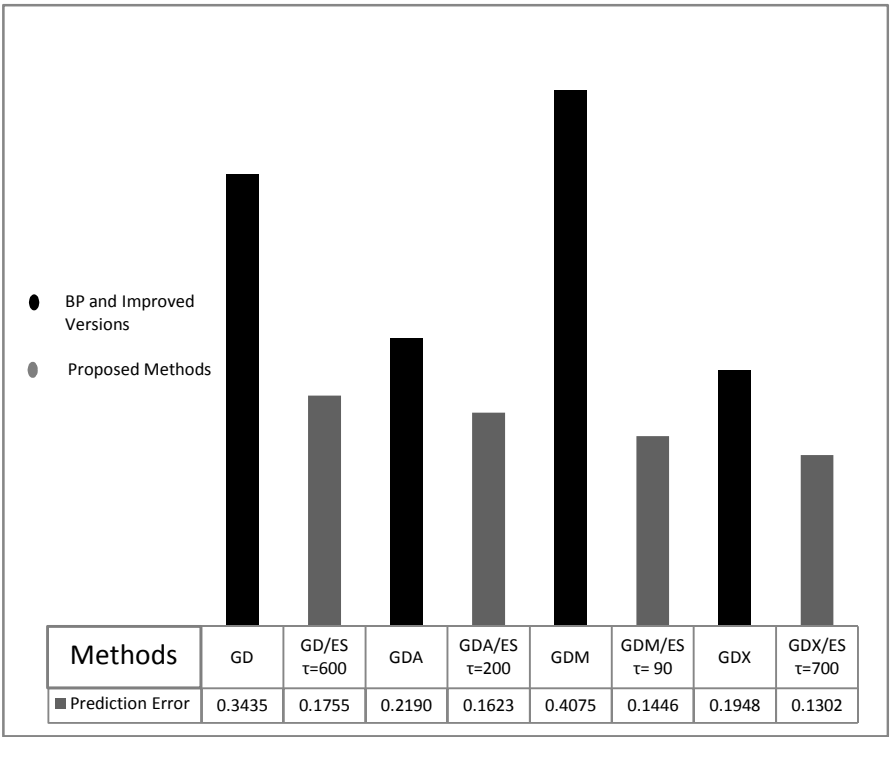
Fig. 3. Flow chart of the proposed method.

4 Experimental Results

In this work we present a simple set of experiments aimed at providing an initial validation of our proposed algorithm. The goal is to compare our BP/ES training

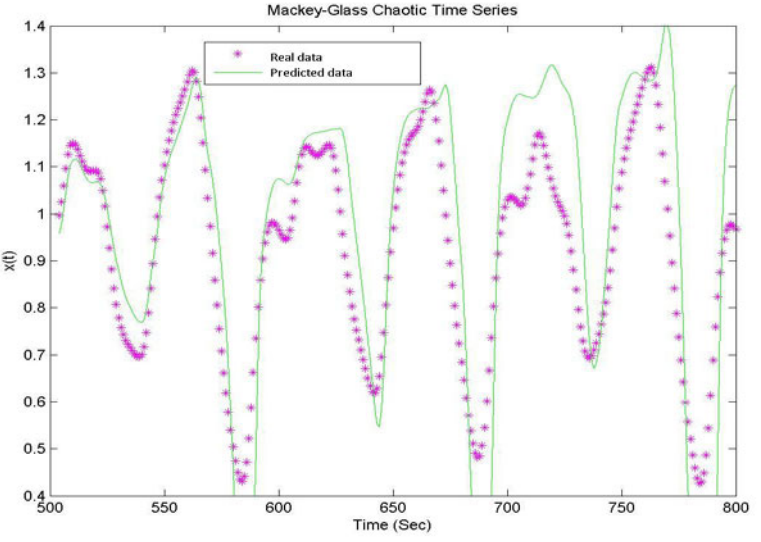
method with conventional BP and basic improvements such as GDA, GDM and GDX. For this we have chosen the Mackey Glass time series as our benchmark test. The time series has a total of 800 ground values of which we use 500 for training and 300 for testing. In all cases we use the same ANN architecture, a feed-forward MLP with one hidden layer of 25 neurons, three inputs for three delays and one output neuron. The initial learning rate for all algorithms is set to 0.01, and for GDM and GDX the initial momentum is set to 0.9. The maximum number of epochs is set to 4000, and the goal convergence error is 1e-10. For our ES/BP algorithms the total number of generations depends upon the value of parameter τ. This parameter was chosen empirically, and the best performance depends on the BP algorithm that is used. In all tests we use a Gaussian mutation to generate the single offspring with a σ=0.2.

The comparisons are made in a pair-wise fashion. For instance, we compare the prediction error of the basic BP (GD) and the enhanced BP-ES, and for the GDA we compare with GDA-ES, and so on for all of the learning methods. In order to obtain statistically significant results we execute each learning algorithm 50 times and present the mean prediction error on the Mackey Glass time series. These results are summarized in figure 4, in which we show a bar plot of the average prediction error, and for our BP/ES algorithm we specify the number of epoch per generation τ.

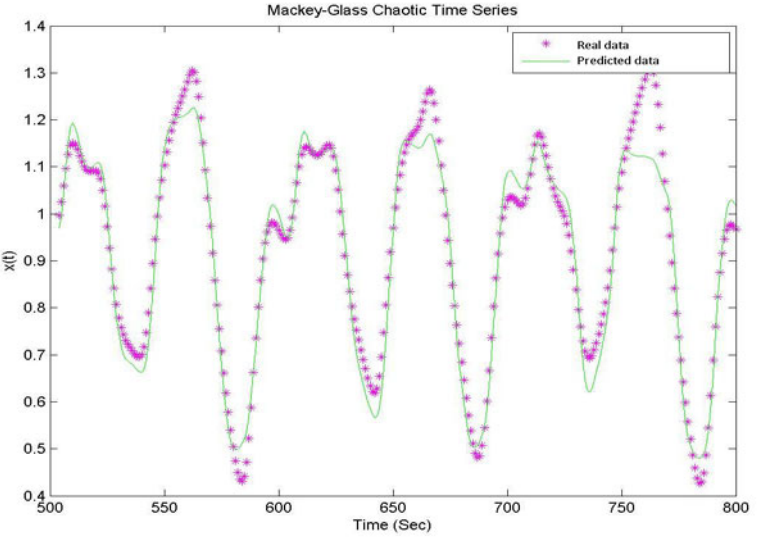


Methods	GD	GD/ES τ=600	GDA	GDA/ES τ=200	GDM	GDM/ES τ= 90	GDX	GDX/ES τ=700
Prediction Error	0.3435	0.1755	0.2190	0.1623	0.4075	0.1446	0.1948	0.1302

Fig. 4. Average of each method in the prediction of Mackey Glass series time.

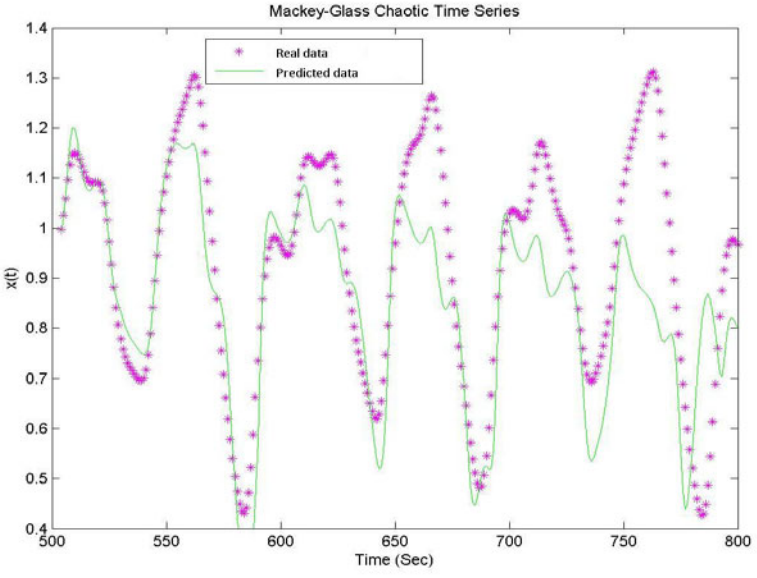


a) Time series prediction using GD algorithm.

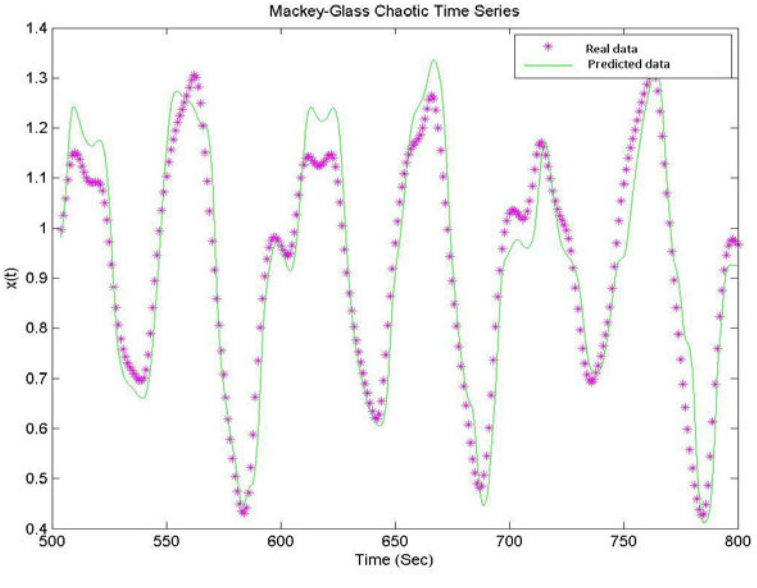


b) Time series prediction using GD/ES algorithm.

Fig. 5. Time series prediction for the Mackey Glass test data with GD and GD/ES. In Both images we see the ground truth in red and the predicted values in green.

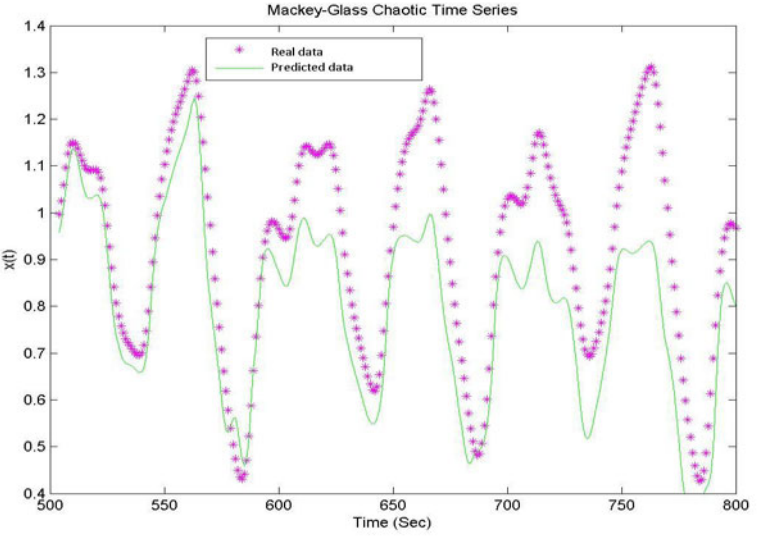


a) Time series prediction using GDA algorithm.

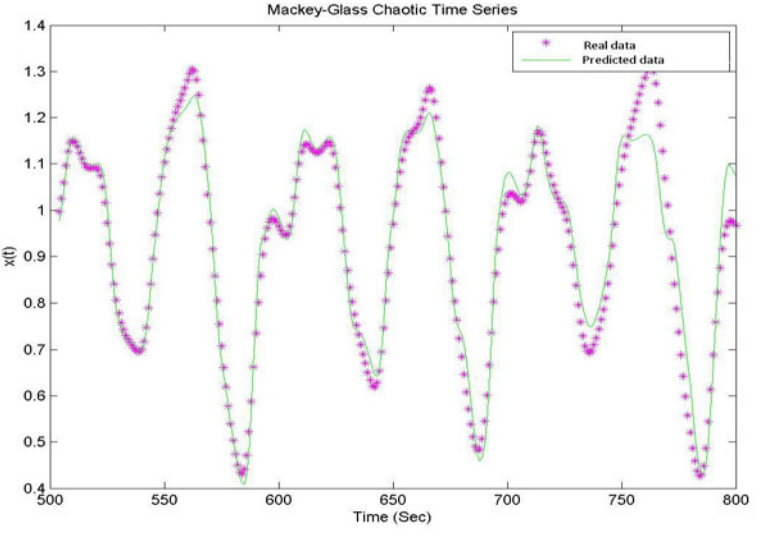


b) Time series prediction using GDA/ES algorithm.

Fig. 6. Time series prediction for the Mackey Glass test data with GDA and GDA/ES. In Both images we see the ground truth in red and the predicted values in green.

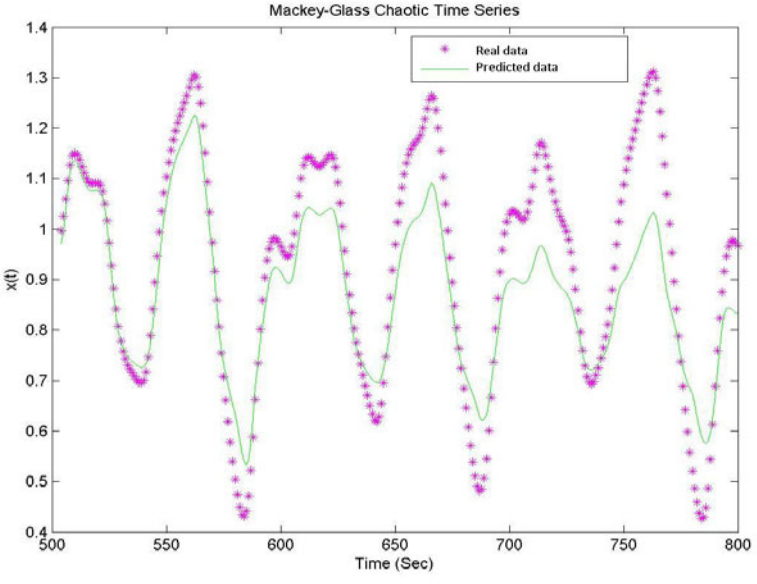


a) Time series prediction using GDM algorithm.

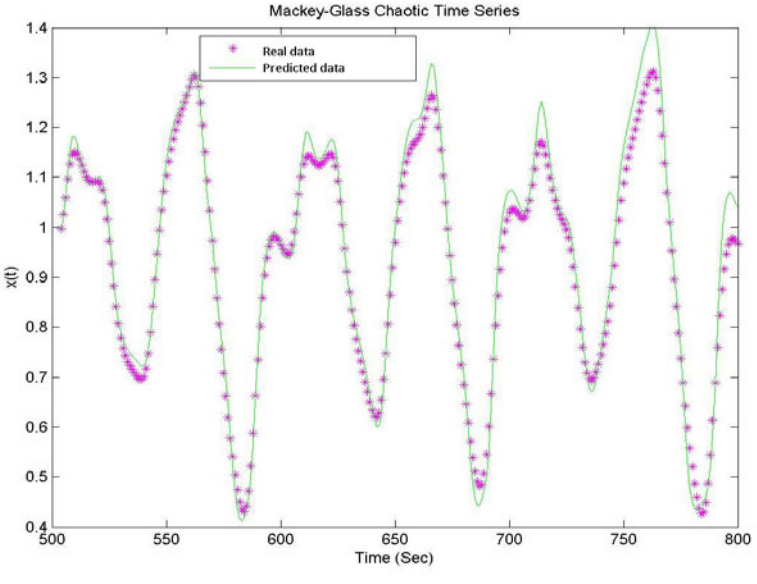


b) Time series prediction using GDM/ES algorithm.

Fig. 7. Time series prediction for the Mackey Glass test data with GDM and GDM/ES. In Both images we see the ground truth in red and the predicted values in green.



a) Time series prediction using GDX algorithm.



b) Time series prediction using GDX/ES algorithm.

Fig. 8. Time series prediction for the Mackey Glass test data with GDX and GDX/ES. In Both images we see the ground truth in red and the predicted values in green.

From these results we can affirm that in all cases the ES search induces a substantial reduction in predictive error. In order to qualitatively illustrate these results, in Figures 5, 6, 7 and 8 we present the Mackey Glass time series prediction compared with the ground truth data for one representative ANN trained with each method. In these plots we can easily see that the BP-ES training algorithms generate an overall better prediction on this well-known benchmark test.

5 Summary and Conclusions

In this chapter we have presented an improvement of the BP learning algorithm for ANNs using (1+1) ES. The goal was to allow for on-line adaptations of the main BP parameters during training, such as the learning rate and the momentum coefficient. The algorithm we have proposed is simple to implement, and can help improve the performance of some of the most widely used variants of BP, such as GDA, GDM and GDX. In order to validate our proposal we have presented preliminary results on a well-known benchmark test, which is the Mackey Glass time series. In our results the improvements provided by the combined BP learning and ES search show a substantial performance improvement with respect to the predictive error of the neural network. In some cases, the improvement was a reduction of 50% in average error over multiple runs.

Therefore, we believe that our proposal can provide a viable alternative to BP learning that allows for an adaptive modification of the main learning parameters. However, the current chapter only outlines a basic comparative study, and further work is required in order to make definite claims on the benefits of our proposal. Nevertheless, the results we have presented here show that the combined learning of BP and the search ability of a (1+1) ES can help induce substantial performance gains and enhance the learning for ANNs.

References

[1] Belew, R.K., McInerney, J., Schraudolph, N.N.: Evolving networks:Using the Genetic Algorithm with connectionist learning. In: Proc. Second Artificial Life Conference, NewYork, pp. 511–547. Addison-Wesley, Reading (1991)

[2] De Jong, K.: A unified approach to Evolutionary Computation. In: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers, GECCO 2009, Montreal, Québec, Canada, July 08-12, pp. 2811–2824. ACM, New York (2009)

[3] Eiben, A.E., Smith, J.E.: 2003 Introduction to Evolutionary Computing. Springer, Heidelberg (2003)

[4] Frean, M.: The upstart algorithm: A method for constructing and training feedforward neural networks. Neural Computation 2(2), 198–209 (1990)

[5] Gurney, K.: An Introduction to Neural Networks. Taylor & Francis, Inc., Abington (1997)

[6] Hagan, M.T., Demuth, H.B., Beale, M.: Neural Network Design. PWS Publishing Co. (1996)

[7] Harp, S.A., Samad, T., Guha, A.: Toward the genetic synthesis of neural networks. In: Schaffer, J.D. (ed.) Proc. 3rd Int. Conf. Genetic Algorithms and Their Applications, pp. 360–369. Morgan Kaufmann, San Mateo (1989)
[8] Pedro, I.V.: Redes Neuronales Artificiales: Un Enfoque Práctico. Pearson Educacion, London (2004)
[9] Jacobs, R.A.: Increased Rates of Convergence Through Learning Rate Adaptation. Technical Report. UMI Order Number: UM-CS-1987-117., University of Massachusetts (1987)
[10] Kim, H.B., Jung, S.H., Kim, T.G., Park, K.H.: Fast learning method for backpropagation neural network by evolutionary adaptation of learning rates. Neurocomput. 11(1), 101–106 (1996)
[11] Lee, S.-W.: Off-line recognition of totally unconstrained handwritten numerals using multilayer cluster neural network. IEEE Trans. Pattern Anal. Machine Intell. 18, 648–652 (1996)
[12] Merelo, J.J., Paton, M., Cañas, A., Prieto, A., Moran, F.: Optimization of a competitive learning neural network by genetic algorithms. In: Mira, J., Cabestany, J., Prieto, A.G. (eds.) IWANN 1993. LNCS, vol. 686, pp. 185–192. Springer, Heidelberg (1993)
[13] Patel, D.: Using genetic algorithms to construct a network for financial prediction. In: Proc. SPIE: Applications of Artificial Neural Networks in Image Processing, Bellingham, WA, pp. 204–213 (1996)
[14] Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. In: Rumelhart, D.E., McClelland, J.L. (eds.) Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Foundations. MIT Press Computational Models of Cognition And Perception Series, vol. 1, pp. 318–362. MIT Press, Cambridge (1986)
[15] Samarasinghe, S.: Neural Networks for Applied Sciences and Engineering. Auerbach Publications (2006)
[16] Schefel, H.-P.: Numerische Optimierung von Computer-Modellen mittels der Wvolutionsstrategie. ISR, vol. 26. Birkhaeuser, Basel (1997)
[17] Skinner, A.J., Broughton, J.Q.: Neural networks in computational materials science: Training algorithms. Modeling and Simulation in Materials Sci. Eng. 3(3), 371–390 (1995)
[18] Topchy, A.P., Lebedko, O.A.: Neural network training by means of cooperative evolutionary search. Nuclear Instrum. Methods in Phys. Res., Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 389(1-2), 240–241 (1997)
[19] Whitehead, B.A., Choate, T.D.: Evolving space-filling curves to distribute radial basis functions over an input space. IEEE Trans. Neural Networks 5, 15–23 (1994)
[20] Whitley, D., Starkweather, T., Bogart, C.: Genetic algorithms and neural networks: Optimizing connections and connectivity. Parallel Comput 14(3), 347–361 (1990)
[21] Yao, X.: Evolving artificial neural networks. Proceedings of the IEEE 87(9), 1423–1447 (1999)
[22] Liu, Y., Yao, X.: Evolutionary design of artificial neural networks with different nodes. In: Proc. 1996 IEEE Int. Conf. Evolutionary Computation (ICEC 1996), Nagoya, Japan, pp. 670–675 (1996)
[23] Hwang, M.W., Choi, J.Y., Park, J.: Evolutionary projection neural networks. In: Proc. 1997 IEEE Int. Conf. Evolutionary Computation, ICEC 1997, pp. 667–671 (1997)
[24] Sebald, A.V., Chellapilla, K.: On making problems evolutionarily friendly, part I: Evolving the most convenient representations. In: Porto, W., Saravanan, N., Waagen, D., Eiben, A.E. (eds.) EP 1998. LNCS, vol. 1447, pp. 271–280. Springer, Heidelberg (1998)

Parallel Genetic Algorithms for Architecture Optimization of Neural Networks for Pattern Recognition

Martha Cárdenas, Patricia Melin, and Laura Cruz

Tijuana Institute of Technology, Tijuana, México
mc.marthacardenas@gmail.com, pmelin@tectijuana.mx

Abstract. This Paper presents the Architecture optimization of Neural Networks using parallel Genetic Algorithms for pattern recognition based on person faces. The optimization consists in obtaining the best architecture in layers, neurons per layer, and achieving the less recognition error in a shorter training time using parallel programming techniques to exploit the resources of a machine with a multicore architecture. We show the obtained performance by comparing results of the training stage for sequential and parallel implementations.

1 Introduction

The recognition of individuals from their biometric features has been driven by the need of security applications, mainly of security such as in surveillance systems for control of employee assistance, access control security places, etc. These systems have been developed with different biometrics including face recognition, fingerprints, iris, voice, hand geometry, etc. [17]. Although there are systems based on classical methods, biometric pattern recognition was developed in the area of artificial intelligence with techniques such as fuzzy logic, data mining, neural networks, genetic algorithms, etc.

Real-World problems are complex to solve and require intelligent systems that combine knowledge, techniques and methodologies from various sources. In this case, we are talking about hybrid systems, and these can be observed in some applications already developed in [6, 15, 16]. These problems often have large computational dimensions, because of the amount of data and their required processing. The effective implementation of models for these problems, requires processor interconnection techniques to optimize the processing time [5].

Artificial Neural Networks have high potential for parallel processing. Their parallel nature makes them ideal for parallel implementation techniques not only in software but also at the hardware level, however it's difficult to find an optimal network architecture for a given application. The architecture and network optimal

P. Melin et al. (Eds.): Soft Comp. for Recogn. Based on Biometrics, SCI 312, pp. 303–315.
springerlink.com

parameter selection is the most important part of the problem and is what takes a long time to find. This paper presents the optimization of the architecture of a neural network for face recognition using parallel genetic algorithms in a multi-core platform; we focus in the Parallel Processing computation.

Parallel Computing brings together advanced software and processors that have multiple cores or engines, which when combined can handle multiple instructions and tasks simultaneously.

Parallel Genetic algorithms (PGAs) can provide considerable gains in terms of performance and scalability and can be implemented on networks of heterogeneous computers or on parallel mainframes.

Several interesting applications of parallel computing are presented by Dongarra et al. [10]. In the coming years computers are likely to have even more processors inside, and in [3] a description on multi-core processor architecture is presented. An introduction to Multi-Objective Evolutionary Algorithms can also be found in [8].

This paper is organized as follows; Section 2 describes the basic concepts of Artificial Neural Networks, Genetic Algorithms, and basic aspects of multi-core architectures. Section 3 explains the Problem statement and implementation, section 4 presents the experimental results. Finally, section 5 presents the conclusions and future work.

2 Theoretical Concepts

Soft Computing consists of several computing paradigms, including fuzzy logic, neural networks and genetic algorithms, which can be combined to create hybrid intelligent systems, these systems leverage the advantages of each of the techniques involved [17]. In this research, we use the paradigms of neural networks and genetic algorithms.

2.1 Neural Networks

A neural network is a computational structure capable of discriminating and modeling nonlinear characteristics. It consists of a set of units (usually large) of interconnected simple processing, which operate together. Neural networks have been widely used because of their versatility for solving problems of prediction, recognition, approach, etc. [6, 17, 14, 25].

These systems emulate, in a certain way, the human brain. They need to learn how to behave (Learning) and someone should be responsible for teaching (Training), based on previous knowledge of the environment problem [18, 27, 26].

2.2 Genetic Algorithms

John Holland introduced the Genetic Algorithm (GA) in 1970 inspired by the process observed in the natural evolution of living beings [27, 19]. Genetic Algorithms (GAs) are search methods based on principles of natural selection and genetics. A GA presents a group of possible solutions called a population; the

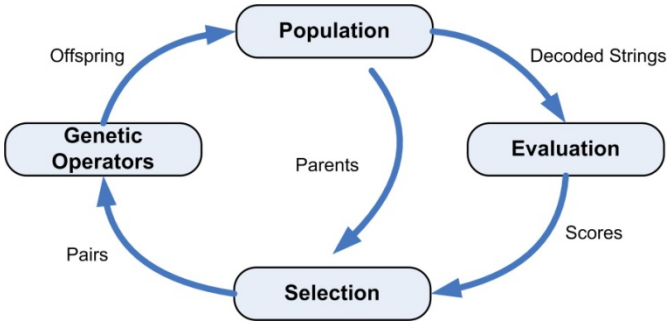


Fig. 1. Structure of a Simple GA

solutions in the population called individuals, each individual is encoded into a string usually binary called chromosome, and symbols forming the string are called genes. The Chromosomes evolve through iterations called generations, in each generation individuals are evaluated using some measure of fitness. The next generation with new individuals called offspring, are formed from the previous generation using two main operators, crossover and mutation, this representation is shown Figure 1.

These optimization techniques are used in several areas such as business, industry, engineering and computer science, also are used as a basis for industrial planning, resource allocation, scheduling, decision-making, etc. The GA is commonly used in the area of intelligent systems, some examples of optimization of fuzzy logic systems and neural networks are shown in [4]. GAs find good solutions in reasonable amounts of time, however, in some cases GAs may require hundreds or more expensive function evaluations, and depending of the cost of each evaluation, the time of execution of the GA may take hours, days or months to find an acceptable solution [4, 19].

The Genetic Algorithms have become increasingly popular to solve difficult problems that may require considerable computing power, to solve these problem developers used parallel programming techniques, the basic idea of the parallel programs is to divide a large problem into smaller tasks and solve simultaneously using multiple processors. The effort for efficient algorithms has led us to implement parallel computing, in this way it's possible to achieve the same results in less time. But making a GA faster is not the only advantage that can be expected when designing a parallel GA. A PGA has an improved power to tackle more complex problems since it can use more memory and CPU resources[1].

2.3 Parallel Genetic Algorithms

The way in which GAs can be parallelised depends of several elements, like how the fitness is evaluated and mutation is applied, if single or multiples subpopulations (demes) are used, if multiple populations are used, how individuals are exchanged, how selection is applied (globally or locally).

Existing different methods for implementing parallel GAs and can be classified in the next general classes:

- Master-Slave parallelisation (Distributed fitness evaluation)
- Static subpopulation with migration
- Static overlapping subpopulations (without migration)
- Massively parallel genetic algorithms
- Dynamic demes (dynamic overlapping subpopulations)
- Parallel Steady-state genetic algorithms
- Parallel messy genetic algorithms
- Hybrid methods

Our Implementation is based on the Master-Slave Synchronous parallelisation, and for that reason we describe only this method, other methods can be reviewed in [4, 19].

Master-slave GAs have a single population. One master node executes the operator's selection, crossover, and mutation, and the evaluation of fitness is distributed among several workers (slaves) processors. The workers evaluate the fitness of every individual that they receive from the master and return the results. A Master-Salve GA depending on whether they wait to receive the fitness values for the entire population before proceeding to the next generation can be synchronous or asynchronous.

Master-Slaves Synchronous

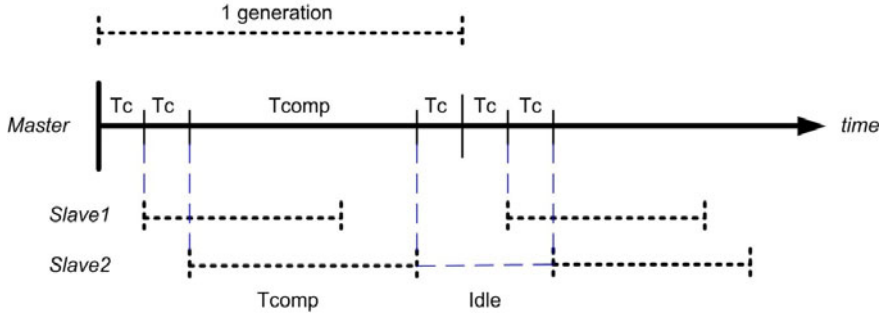


Fig. 2. Execution of Parallel Master-Slaves GA.

The improvement in actual processors is based on the development of Chips Multiprocessors (CMPs) or Multi-core processors, thus to increase the efficiency of a processor, increases the number of cores inside the processor chip. Multi-core processors technology is the implementation of two or more "execution cores" within a single processor, some of the advantages of multi-core architectures are shown in [10, 7]. These cores are essentially two or more individual processors on a single chip. Depending on the design, these processors may or may not share a large on-chip cache; the operating system perceives each of its execution cores as a discrete logical processor with all the associated execution resources [5],

however to exploit these architectures is necessary to develop parallel applications that use all the processing units simultaneously. In order to achieve parallel execution in software, hardware must provide a platform that supports the simultaneous execution of multiple threads. Software threads of execution are running in parallel, which means that the active threads are running simultaneously on different hardware resources, or processing elements. Now it is important to understand that the parallelism occurs at the hardware level too.

The improvement measure or speedup takes as reference, the time of execution of a program in a mono-processor system regarding the time of execution of the same program in a multiprocessor or multi-core system, which is represented as follows:

$$speedup = \frac{t_s}{t_p}, \tag{1}$$

Where t_s is the time it takes to run the program in a mono-processor system and t_p is the time it takes to run the same program in a system with *p* execution units.

There are many models of parallel programming, the two main choices and the most common are Shared-memory programming and Distributed memory [5], also the parallelism can be implemented in two ways, implicit parallelism, that some compilers perform automatically, these are responsible to generate the parallel code for the parts of the program that are parallel, and the explicit parallelism which is implemented using parallel languages, and the responsible of the parallelism is the programmer, that defines the threads to work, the code of each thread, the communication, etc., this last parallelism gets higher performance.

3 Problem Statement

In this section the main goal of the research is described; in this case we focus on the analysis of parallel genetic algorithms for optimizing the architecture of a neural network for recognition of persons based on the face biometry implemented in multi-core processors.

For determining the best architecture and parameters for a neural network there is no particular selection criterion, for example the number of layers and neurons per layer for a particular application is chosen based on experience and to find an optimal architecture for the network becomes a task of trial and error. In addition, there are others methods that with a empirical expression can calculate and determining the architecture of neural network for a specific problem [24].

3.1 Neural Network Structure

The data base used for this research is The ORL Database of Faces of the Cambridge University Computer Laboratory [9]. This database contains ten different images of 40 persons with different gestures for our implementation, not apply any preprocessing for this time, the examples pictures shown in figure 3.

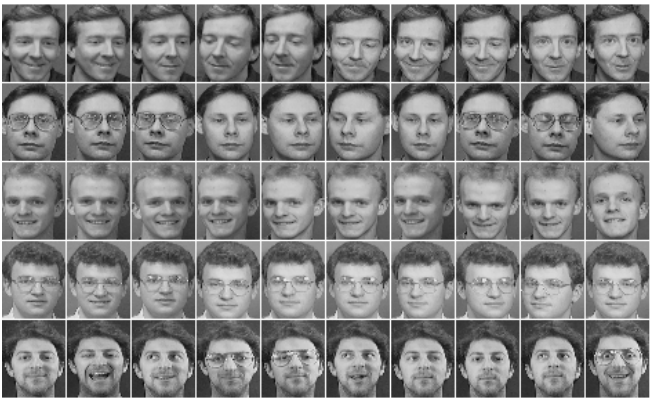


Fig. 3. Some images of the ORL database, the database is composed by 400 images, there are images of 40 diferent persons (10 images per person).

The Genetic Algorithm was tested in a Multi-core computer with following characteristics: CPU Intel Core 2 Quad 2.4 GHz, Bus 1066 MHz, 8MB of L2 cache, Memory 6 GBytes DDR2 of main memory, all the experiments were achieved in the MatLab Version R2009b using the Parallel computing toolbox.

Neural networks were applied to a database of 40 persons, we used 5 images per person for training and 5 images per person for test. First, we implemented the traditional monolithic neural network, and before we implemented a Parallel GA for optimizing layer and neurons per layer. The training method for the neural network is the Trainscg (Scaled Conjugate Gradient), with an error goal of 0.01e-006 and between 100 and 150 generations.

3.2 Parallel Genetic Algorithm for Optimization

The Master-Slave Parallel genetic Algorithm was codified with a binary chromosome of 23 bits, 2 bits for number of layers, and 7 bits for number of neurons per layer. The maximum number of layers is 3 and neurons 128, this is shown in figure 4. The proposed algorithm was implemented in a Shared Memory Multi-core machine with 4 cores, taking one core as master and the remaining cores as slaves.

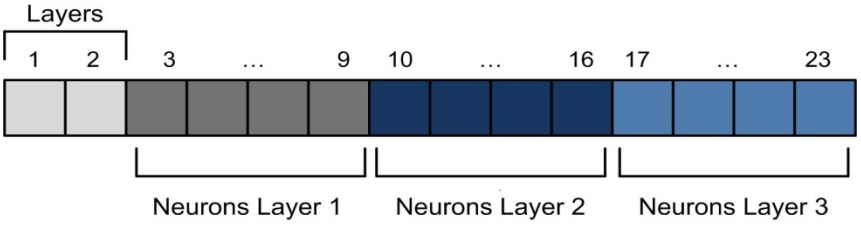


Fig. 4. Chromosome representation of the problem.

The Genetic Algorithm has the following characteristics:

- Chromosome Size: The number of genes in each individual for this application is 23 binary bits.
- Population size: Defines the number of individuals that will compose the population.
  Population Size =20
- Termination Criteria: Maximum number of generations for solving the problem.
  Max Generations=50
- Selection: We used Stochastic Universal Sampling
  Selection Prob=0.9
- Crossover: The selected individuals have a probability of mating, acting as parents to generate two new individuals that will represent them in the next generation. The crossing point is random with a probability of 0.7.
- Mutation: Represents the probability that an arbitrary bit in the individual sequence will be changed from its original stat. Mutation Probability 0.8

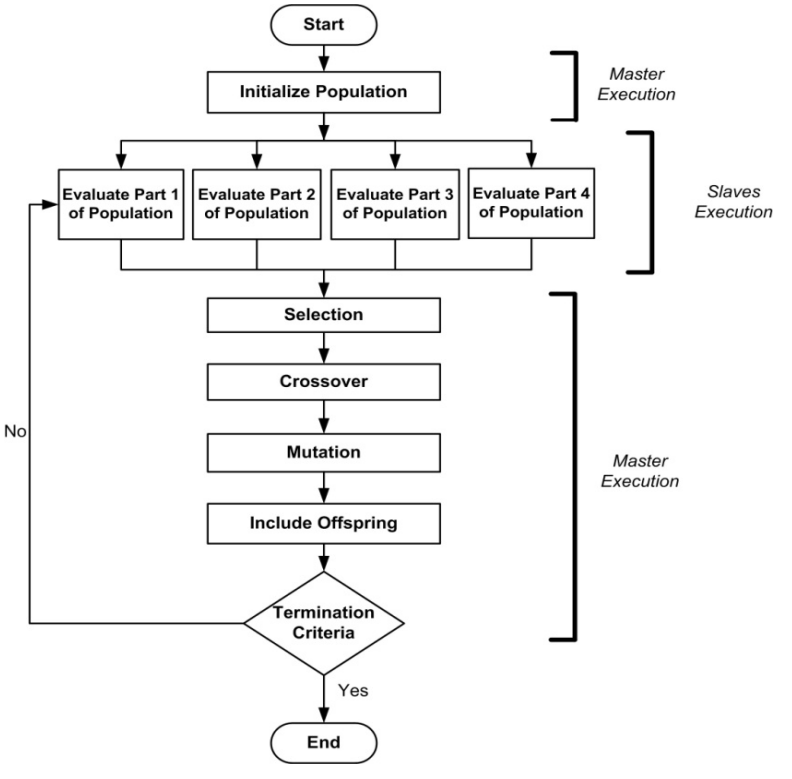


Fig. 5. Parallel GA Implementation.

The flow chart of figure 5 shows the parallel GA implementation. Other methods for solving a Parallel GA can be seen in [21, 12, 4].

First we create a random initial population, the master divides the population and send it to the slaves (in this case the cores of processor), and take a part of the population to evaluate.
In the Slaves, for each individual
Load the architecture of the network
Read the images and put as input in the network
Propagate this image in the network
Calculate the error of training
When finish the slaves send the results of evaluation to the master.
Master waits to all slaves finish to collect the entire population and make selection based on the fitness of all the individuals.
The Master performs the operators of crossover, mutation and generates the new population to be divided and evaluated until the maximum number of generations is reached.

The experimental results achieved with this GA implementation are presented in the following section.

4 Experimental Results

Different experiments were developed to observe the performance of parallel genetic algorithms; the results are presented in this section, the results in time represent the average time execution of each neural network.

We train the neural network without optimization, in sequential form. After manually changing the architecture for several times, we defined the architecture of the neural network with the expression of Salinas [24] as follows:

- First hidden layer (2 * (k + 2)) = 84.
- Second hidden layer (k + m) = 45.
- Output layer (k) = 40.

where k corresponds to the number of individuals and m to the number of images of each of them.

Table 1 shows the average of 20 trainings in sequential form of the network in a dual-core and quad-core machines, in this experiment we enabled only one of the cores available and one thread of execution in the processor for simulating sequential execution. Figure 6 shows the usage of a dual-core machine in the training of the network.

In the experiment of training with implicit parallelism without optimization all the cores and threads available are enabled. Matlab R2009b uses as a default the implicit parallelism for the applications run on it and take all the cores for execution automatically generating a thread of execution per processor. The results obtained for Matlab in a dual-core and quad-core machines shown in the table 2.

Table 1. Average of 20 trainings in sequential form of the network for dual-core and quad-core machines.

No. Cores	Epochs	Error Goal	Error	Total Time
4	150	1.00E-06	0.00178	**1:18 min**
2	150	1.00E-06	0.00468	**1:14 min**

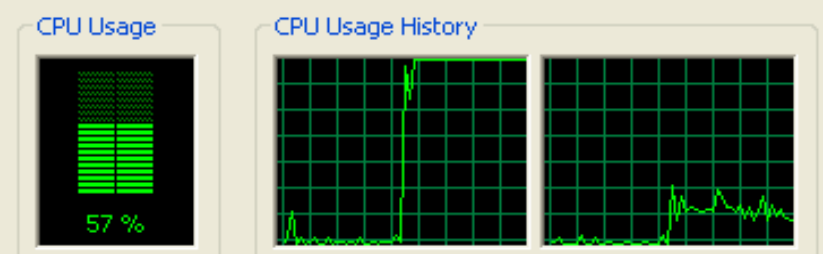


Fig. 6. Cores Usage in the sequential training of the network.

Table 2. Average of 20 trainings of implicit parallelism form of the network for dual-core and quad-core machines.

No. Cores	Epoch	Error Goal	Error	Total Time
4	150	1.00E-06	0.00113	**1:58 min**
2	150	1.00E-06	0.00384	**1:41 min**

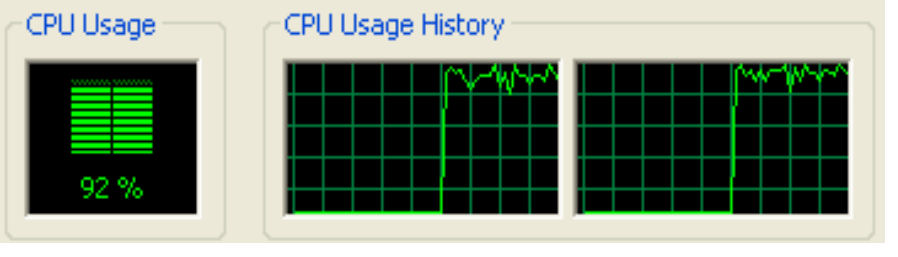


Fig. 7. Cores Usage in the implicit parallelism in training of the network.

The results show that the execution of serial training of the network are more efficient that the implicit paralellism of matlab, because when a single core is working (figure 6) all the cache memory is available for them.

Implicit Parallelism with GA optimization

We optimize the monolithic Neural Network with a simple GA in the form of implicit parallelism. Table 3 shows the average of 20 training test for 2 and 4 cores. Figures 8 and 9 show the usage of processor in dual-core and quad-core machines.

Table 3. Average of 20 training of implicit parallelism of Simple GA for optimization of the network with for dual-core and quad-core machines.

N. Cores	Ind	Gen	Cross	Mut	Error	Time/ nework	Average time
2	20	30	0.7	0.8	3.0121e-004	1:51min	33 min
4	20	30	0.7	0.8	9.7361e-005	4:10 min	83 min

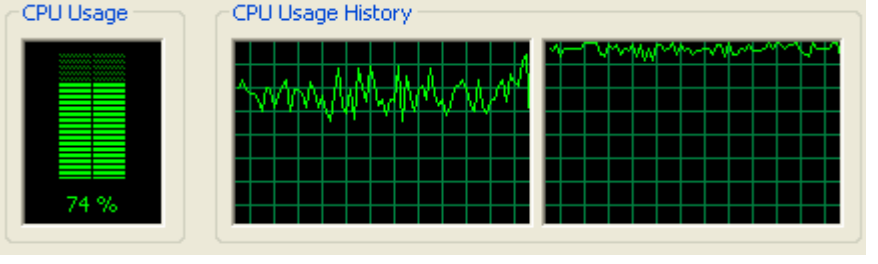


Fig. 8. CPU Performance Implicit Parallelism with 2 cores.

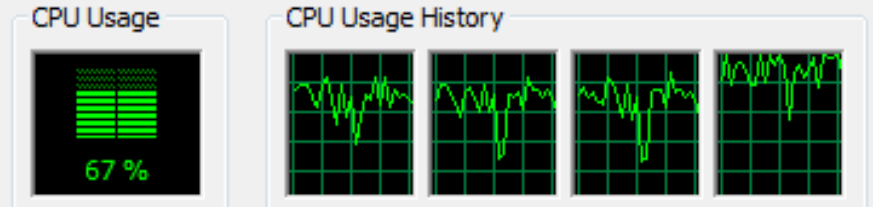


Fig. 9. CPU Performance Implicit Parallelism with 4 cores.

Explicit Parallelism with Parallel GA optimization

In this experiment, we utilize the matlabpool that enables the parallel language features within the MATLAB language by starting a parallel job, which connects this MATLAB client with a number of labs. The average results for 20 executions are shown in table 4. Figures 10 and 11 show the usage of the processor for training with explicit parallelism in a dual-core and quad-core machines.

Table 4. Average of 20 training of explicit parallelism of Simple GA for optimization of the network with for dual-core and quad-core machines.

N. Cores	Ind	Gen	Cross	Mut	Error	Time/ nework	Ave- rage time
2	20	30	0.7	0.8	3.3121e-004	1:03min	21 min
4	20	30	0.7	0.8	4.5354e-004	:35 seg	12 min

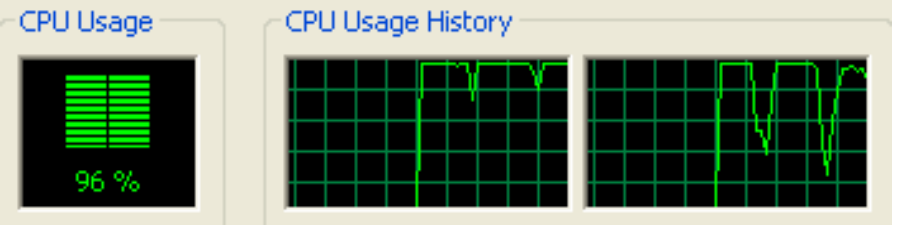


Fig. 10. CPU Performance Explicit Parallelism with 2 cores.

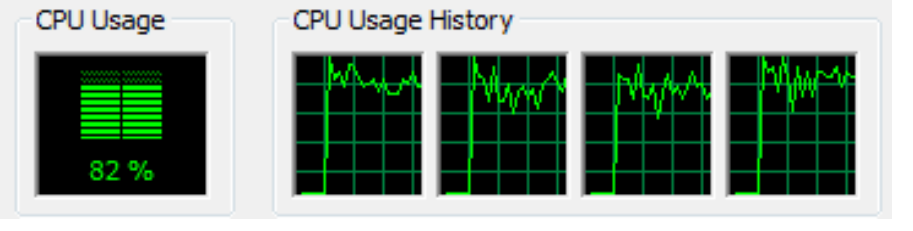


Fig. 11. CPU Performance Explicit Parallelism with 4 cores.

Table 5 shows a comparison between all the training experiments, and observed that

Table 5. Table of Results of experiments Sequential and Parallel.

	RNA		GA-RNA			
	Sequential		GA. Implícit Parallelism		GA. Explicit Parallelism	
No. Cores	**2**	**4**	**2**	**4**	**2**	**4**
Average time/ network	**1:18 min**	**1:14 min**	**1:58 min**	**1:41 min**	**1:03 min**	**0:35 seg**

5 Conclusions

We present the experiments with training of the monolithic neural network for database of face, we used different implementations of parallelism to show that the parallel GA used multi-core processor offers best results in the search for optimal neural network architectures in less time.
The genetic Algorithms take considerable time to successfully complete convergence depending of application, but always achieve satisfactory optimal solutions. Genetic Algorithm can be parallelized to speedup its execution; and if we use Explicit Parallelization we can achieve much better speedup than when using implicit Parallelization, anyway it's necessary to make more test.

The future work consists in considering a larger size data bases and implementing a modular neural network in a multi-core cluster applying different techniques of parallel processing.

References

[1] Alba, E., Nebro, A., Troya, J.: Heterogeneous Computing and Parallel Genetic Algorithms. Journal of Parallel and Distributed Computing 62, 1362–1385 (2002)
[2] Bertona, L.: Neural Network training based on Evolutionary Algorithms, Engineering Thesis, Universidad de Buenos Aires (2005)
[3] Burger, T.W.: Intel Multi-Core Processors: Quick Reference Guide, http://cachewww.intel.com/cd/00/00/20/57/205707_205707.pdf
[4] Cantu-Paz, E.: Efficient and Accurate Parallel Genetic Algorithms. Kluwer Academic Publishers, Dordrecht (2001)
[5] Cárdenas, M., Tapia, J., Montiel, O., Sepúlveda, R.: Neurofuzzy system implementation in Multicore Processors, IV Regional Academic Encounter, CITEDI-IPN (2008)
[6] Castillo, O., Melin, P.: Hybrid intelligent systems for time series prediction using neural networks, fuzzy logic and fractal theory. IEEE Transactions on Neural Networks 13(6) (2002)
[7] Chai, L., Gao, Q., Panda, D.K.: Understanding the Impact of Multi-Core Architecture in Cluster Computing: A Case Study with Intel Dual-Core System. In: The 7th IEEE International Symposium on Cluster Computing and the Grid, CCGrid 2007 (2007)
[8] Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solvin Multi-Objective Problem. Springer, Heidelberg (2004)
[9] The Database of Faces, Cambridge University Computer Laboratory, http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html
[10] Dongarra, J., Foster, I., Fox, G., Gropp, W., Kennedy, K., Torczon, L., White, A.: Sourcebook of Parallel Computing. Morgan Kaufmann Publishers, San Francisco (2003)
[11] Domeika, M., Kane, L.: Optimization Techniques for Intel Multi-Core Processors, http://softwarecommunity.intel.com/articles/eng/2674.htm
[12] González, S.: Optimization of Artificial Neural Network Architectures for time series prediction using Parallel Genetic Algorithms. Ms. Thesis (2007)

[13] Haupt, R.L., Haupt, S.E.: Practical Genetic Algorithms. Wiley-Interscience, Chichester (2004)
[14] Hornik, K.: Some new results on neural network approximation. Neural Networks 6, 1069–1072 (1993)
[15] Jeffrey, A., Oklobdzija, V.: The computer Engineering Handbook, Digital Systems and Aplications, segunda edicion. CRC press, Boca Raton (1993)
[16] Kouchakpour, P., Zaknich, A., Bräunl, T.: Population Variation in Genetic Programming. Elsevier Science Inc., Amsterdam (2007)
[17] Melin, P., Castillo, O.: Hybrid Intelligent Systems for Pattern Recognition using Soft Computing: An Evolutionary Approach for Neural Networks and Fuzzy Systems. Springer, Heidelberg (2005)
[18] Melin, P., Castillo, O.: Hybrid Intelligent Systems for Pattern Recognition Using Soft Computing. Springer, Heidelberg (2005)
[19] Mitchell, M.: An introduction to genetic algorithms. MIT Press, Cambridge, ISBN-10: 0-262-63185-7
[20] Nowostawski, M., Poli, R.: Parallel Genetic Taxonomy. In: KES 1999, May 13 (1999) (submitted for publication)
[21] Ross, A., Nandakumar, K., Jainet, A.K.: Handbook of Multibiometrics. Springer, Heidelberg (2006)
[22] Sahab, M.G., Toropov, V., Ashour, A.F.: A Hybrid Genetic Algorithm For Structural Optimization Problems. Asian journal of civil Engineering, 121–143 (2004)
[23] Saldivar, P.: Control of a Mechanical Arm Through Soft Computing Techniques. Ms. Thesis, CITEDI-IPN (2006)
[24] Salinas, R.: Neural Network Architecture Parametric Face Recognition, Paper, University of Santiago of Chile, pp. 5–9 (2000), http://cabierta.uchile.cl/revista/17/articulos/paper4/index.html
[25] Serrano, R.: Multicore computing applied to Genetic Algorithms. Ms. Thesis, CITEDI-IPN (2008)
[26] Shorlemmer, M.: Basic Tutorial of Neural Networks. In: Artificial Intelligence Research Institute, Barcelona, Spain (1995)
[27] Soto Castro, M.: Face and Voice Recognition in real time using Artificial Neural Networks, Tijuana Institute of Technology, Ms. Thesis (2006)

Scene Recognition Based on Fusion of Color and Corner Features

Mario I. Chacon-Murguia[1], Cynthia P. Guerrero-Saucedo[2], and Rafael Sandoval-Rodriguez[2]

[1] DSP & Vision Laboratory, Chihuahua Institute of Technology, Chihuahua, Mexico
[2] Robotic Laboratory, Chihuahua Institute of Technology, Chihuahua, Mexico
{mchacon,cpguerrero,rsandoval}@itchihuahua.edu.mx

Abstract. The advance of science and technology has motivated to face new and more complex engineering applications. These new challenges must involve not only the design of adaptive and dynamic systems but also the use of correct information. Everyday, it is more evident that good multicriteria decision making systems require different types of information; therefore data fusion is becoming a paramount point in the design of multicriteria decision making systems. This chapter presents a scenery recognition system for robot navigation using a neural network hierarchical approach. The system is based on information fusion in indoor scenarios. The neural systems consist on two levels. The first level is built with one neural network and the second level with two. The system extracts relevant information with respect to color and landmarks. Color information is related mainly to localization of doors. Landmarks are related to corner detection. The hierarchical neural system, based on feedforward architectures, presents 90% of correct recognition in the first level in training, and 95% in validation. The first ANN in the second level shows 90.90% of correct recognition during training, and 87.5% in validation. The second ANN has a performance of 93.75% and 91.66% during training and validation, respectively. The total performance of the systems was 86.6% during training, and 90% in validation.

Keywords: Scene recognition, robotics, corner detection.

1 Introduction

The advance of science and technology has motivated new and more complex engineering applications. These new challenges must involve not only the design of adaptive and dynamic systems but also the use of correct information. It is everyday more evident that good multicriteria decision making systems requires the fusion of data from multiple sources. A research area where data fusion has become a fundamental issue is autonomous robot navigation. Making a robot to navigate and perceive its environment requires similar information as the used by a human [1, 2, 3]. This information usually comes from range detection sensors such as ultrasonic, laser, or infrared, and also from image acquisition sensors, such as CCD

P. Melin et al. (Eds.): Soft Comp. for Recogn. Based on Biometrics, SCI 312, pp. 317–331.
springerlink.com

or CMOS cameras [4]. The information of each sensor must be processed adequately in order to extract useful information for the navigation system of the robot. One paramount issue in autonomous navigation of robots is related to scenery recognition. Recognition of sceneries consists on the identification of a scenario perceived through measurements provided by a sensor. The sensor may be any of the previously mentioned. However, vision sensors are the most frequently used in this task [5, 6, 7]. The advantage of a vision sensor is that it provides compound information that may be separated into useful properties like color, edges, texture, shape, spatial relation, etc. Therefore, it is possible to achieve data fusion with the information of a vision sensor.

This chapter presents the design of a hierarchical neural system for scene recognition using information fusion from indoor scenarios provided by a camera. The problem to solve is constrained to the recognition of 10 indoor scenarios shown in Figure 1. The features used in the design of the hierarchical neural networks are related to door position, and corner detection.

2 Corner Detection Method

The methods for the detection of corners can be divided in two groups: those which can accomplish the detection from the image in gray scale, and those which first detect edges and then detect corners. Among the methods of the first group, the most mentioned in the literature are the method of SUSAN [8] and the method of Harris [9].

Among the second group of corner detectors, which use any method of edge detectors, we can mention the one of X.C. He and N.H.C. Yung [10]. They use

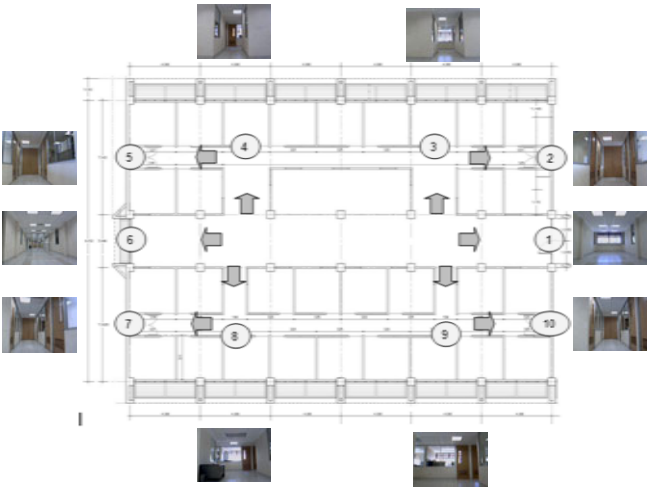


Fig. 1. Scenarios to be recognized.

the method of Canny and indicate the steps to follow for the detection of corners calculating the curvature for each edge.

Other authors use windows for corner detection from edge images, such as K. Rangarajan et al., [11]. In a similar way, G. Aguilar et al., [12], compare images of fingerprints for the identification of persons using 3x3 windows. On those, they propose different bifurcations to be found, which we could call 'corners'. W. F. Leung et al., [13], use 23 windows of different bifurcations, and 28 different windows of other type of corners for their detection in the finger print image using neural networks. The method described in this work is based on the second group of corner detectors. Those which first apply edge detection and then detect corners using windows over the edge image.

2.1 Edge Detection

The corner definition adopted in this work is the one provided by Rangarajan, it is necessary to find the main line intersections of the scene under analysis. These lines are detected through an edge detection procedure. Among the edge detector operators tested in this work were Sobel, Prewitt, Robert, Canny, and Laplacian. It was decided to use the Canny edge detection method [13], because it generated the best edges in the experiments achieved in this research. It is also one of the most mentioned and used edge detector methods in the literature.

2.2 Corner Detection Windows

The papers from G. Aguilar [12], and W.F. Leung et al. [13], coincide in that there are different types of bifurcations or corners that we call them, Y´s, V´s, T´s, L´s, and X´s, accordingly to the form they take, as shown in Figure 2. Based on the similitude of these corners with fingerprint marks, it was decided to investigate the possibility of using a unified theory between fingerprint recognition and scene recognition. Thus, from the fingerprint recognition works, some windows were chosen to detect corners.

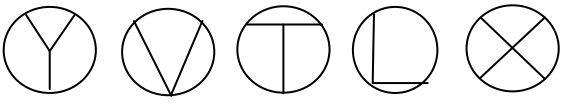


Fig. 2. Type of corners.

These selected windows plus other proposed in this work make a set of 30 windows. Each corner detection window, w_c, is a 3X3 mask and their structures are illustrated in Figures 3 and 4.

The set w_c of windows is composed as follows. Windows w_1, w_2, w_3, and w_4, are four windows modified from the work of Leung et al. [13]. The modification consists on the aggregation of one pixel because they try to find terminal points, and in our case we look for crossing lines. The extra pixel is darkened in these windows. Windows w_5 to w_{20} were also taken from Leung. The windows w_{17} to w_{20}

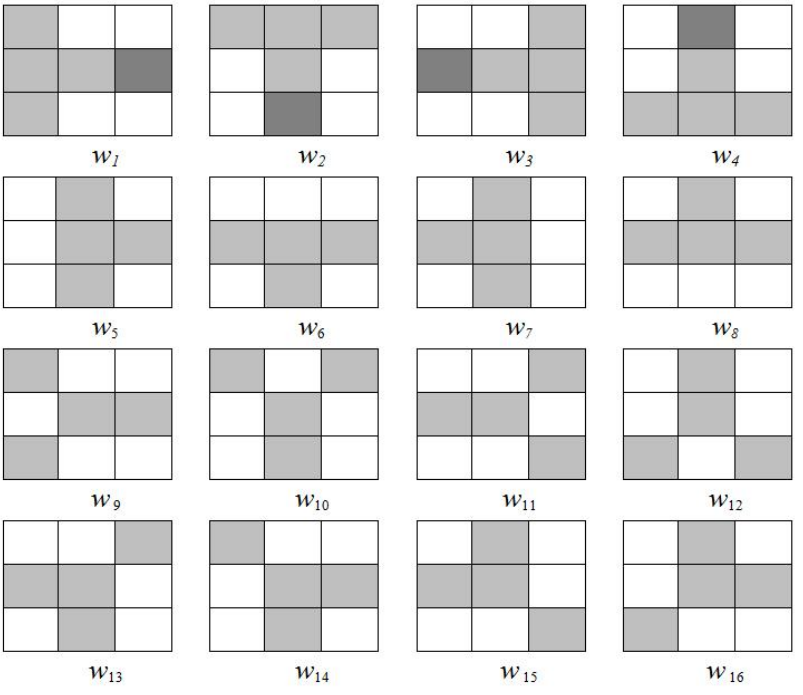


Fig. 3. Windows w_1 to w_{16} for corner detection.

appear in Aguilar et al. [12]. The subset w_{21} to w_{30} are windows proposed in this paper. The proposed windows were defined by analysis of the corners usually found in the set of images considered in this work.

2.3 Corner Detection

Corner detection is achieved through a windows matching process. The process starts by generating a generic binary weight matrix T_n defined as

$$\begin{bmatrix} 256 & 32 & 4 \\ 128 & 16 & 2 \\ 64 & 8 & 1 \end{bmatrix} \tag{1}$$

Each corner detection window is then associated with an index window B_i

$$T_n : w_c \Rightarrow B_i \qquad for \quad c, i = 1,..,30 \tag{2}$$

obtained by

$$B_i = w_c \times T_n \tag{3}$$

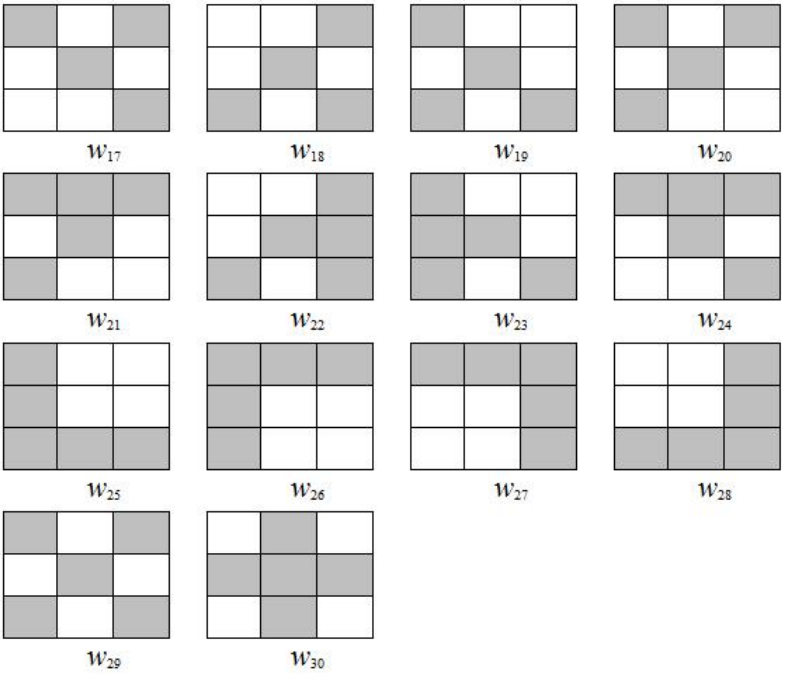


Fig. 4. Windows w_{17} to w_{30} for corner detection.

where the multiplication is element by element and not a matrix multiplication. In this way, each w_c window is related to an index window B_i. In the same way, each index window B_i can be associated to a total weighting factor α_i obtained by

$$\alpha_i = 1 + \sum_{b_i \in B_i} b_i \tag{4}$$

where the b_i corresponds to the weighting factor in B_i.

Corner detection of a scene is accomplished by the next steps. First convolve the binary Canny result image $I_b(x,y)$ with the index matrix B_i

$$I_{ci}(x, y) = I_b(x, y) * B_i + 1 \tag{5}$$

This step yields the possible corners related to each corner window w_c. The next step is to decide which of the possible candidate pixels in each $I_{ci}(x,y)$ is a corner that corresponds to w_c. This process is realized scanning the $I_{ci}(x,y)$ and assigning a pixel value according to

$$p_{ei}(x, y) = \begin{cases} 1 & p_{ci}(x, y) = \alpha_i \\ o & \text{otherwise} \end{cases} \tag{6}$$

to produce a new set of images $I_{ei}(x,y)$ where $p_{ci}(x, y) \in I_{ci}(x, y)$ and $p_{ei}(x, y) \in I_{ei}(x, y)$. The value 1 indicates that the pixel $p_{ci}(x,y)$ is a corner of the

type w_c. This process ends up with 30 binary images that indicate the position of the different type of corners. The final step consists on the union of the $I_{ei}(x,y)$ images to produce the final corners

$$I_{FC}(x,y) = \bigcup_{i=1}^{30} I_{ei}(x,y) \tag{7}$$

The proposed method was tested with semi-artificial as well as with real scenarios. The artificial scenarios were used to obtain a quantitative performance. Examples of semi-artificial scenarios are shown in Figure 5. A summary showing the performances of both, the proposed and the Harris methods, are shown in Tables 1 and 2. The detection of real corners is very alike in the two methods, 98.41and 97.36%, respectively. There is a more noticeable difference in the false positives, where the proposed method has 7.66%, while the Harris has a 33.33%. A comparison with the SUSAN algorithm is not possible because it requires multi-gray level information.

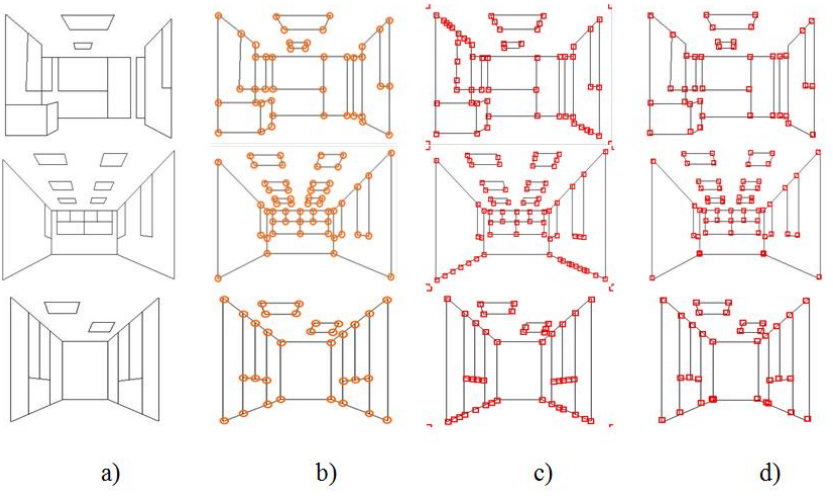


Fig. 5. a) Semi-artificial scenarios, 3, 6, 10, 15, 16. b) Corners to detect, c) Harris detection, d) detection with the proposed method.

Table 1. Performance of the proposed method.

Scenario	Real Corners	Corner detected	Hits	False positives	False negatives
3	42	40	40 / 95.23%	0 / 0%	2 / 5%
6	55	61	55 / 100%	6 / 11%	0 / 0%
10	32	36	32 / 100%	4 / 12%	0 / 0%
Total	129	137	127 / 98.41%	10 / 7.66%	2 / 1.6%

Table 2. Performance of the Harris method.

Scenario	Real corners	Corner detected	Hits	False positives	False negatives
3	42	55	41/ 97.6%	14 / 33%	1 / 2.4%
6	55	70	52 / 94.5%	18 / 33%	3 / 5.5 %
10	32	43	32 / 100%	11 / 34%	0 / 0%
Total	129	168	125/ 97.36%	43 / 33.33%	4 /3.95%

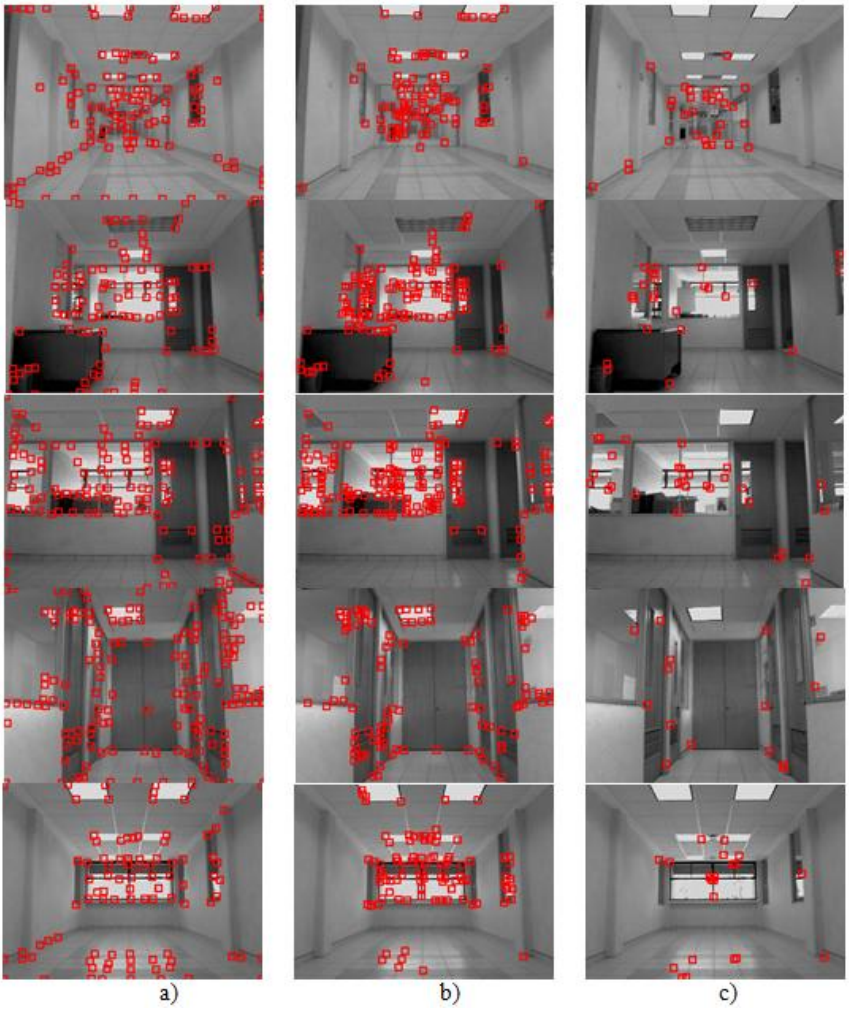


Fig. 6. Corner detection by a) Harris b) SUSAN and c) Proposed method.

In the case of the Harris method, it was assumed two gray level images. A qualitative comparison will be given over the original images later on.

Results of Harris, SUSAN, and the proposed mehtod, on scenarios of interest are shown in Figure 6. It can be observed that, in general, Harris and SUSAN tend to detect more corners than the proposed method. However, the false positive rate is presumed to be very high, as proved with the semi-artificial images using the Harris method. Considering that corner information is used for robot navigation, high rate on false positives may lead to complicate more the scene recognition than the lack of some corners.

3 Scene Segmentation

This section describes the segmentation process to obtain features related to the doors found in the scenarios under analysis. The process is shown in Figure 7. The RGB image is transformed to the HSV color space to be more tolerant to illumination changes [14]. Then detection of the doors is achieved by color analysis.

Fig. 7. A block diagram of the process for door detection.

3.1 RGB to HSV Color Space

The color space transformation from RGB to HSV is obtained by the following equations

$$V = \max(R,G,B) \tag{8}$$

$$S = \begin{cases} 0 & \text{if } \max(R,G,B) = 0 \\ 1 - \dfrac{\min(R,G,B)}{\max(R,G,B)} & \text{otherwise} \end{cases} \tag{9}$$

$$H = \begin{cases} \dfrac{1}{6}\dfrac{G-B}{\max(R,G,B)-\min(R,G,B)} & \text{if } \max(R,G,B) = R \quad \text{and} \quad G \geq B \\ \dfrac{1}{6}\dfrac{B-R}{\max(R,G,B)-\min(R,G,B)} + 1 & \text{if } \max(R,G,B) = R \quad \text{and} \quad G < B \\ \dfrac{1}{6}\dfrac{B-R}{\max(R,G,B)-\min(R,G,B)} + \dfrac{1}{3} & \text{if } \max(R,G,B) = G \\ \dfrac{1}{6}\dfrac{R-G}{\max(R,G,B)-\min(R,G,B)} + \dfrac{2}{3} & \text{if } \max(R,G,B) = B \end{cases} \tag{10}$$

3.2 HSV Component Analysis

Statistics of the HSV values where determined by a sampling process. The sampling consisted on a set of 11samples from each door in the scenarios, Figure 8. Each sample corresponds to a window of 5X5 pixels. The process involved the computation of the mean, and variance, of the mean distribution of the windows samples over the HSV values. Table 3 shows the mean distribution over the scenarios, while Table 4 the statistics values of the means.

Fig. 8. Color door sampling in two scenarios.

Table 3. Mean distribution over the scenarios.

Scenario	H **Mean**	S **Mean**	V ***Mean***
2	0.078	0.361	0.357
3	0.059	0.471	0.205
4	0.084	0.605	0.252
5	0.075	0.393	0.360
7	0.099	0.367	0.361
9	0.075	0.576	0.243
10	0.078	0.308	0.372

Table 4. Statistics values of the means

Component	**Mean**	**Standard deviation**
H	$\overline{H} = 0.078$	$H_\sigma = 0.018$
S	$\overline{S} = 0.440$	$S_\sigma = 0.132$
V	$\overline{V} = 0.307$	$V_\sigma = 0.077$

3.3 Door Segmentation

Door detection in autonomous robot navigation is an important issue because they appear in many interior environments [15]. Thus, doors are important landmarks

that can be used for scene recognition. Door detection is achieved in this work by analysis of the HSV components implemented in the following condition

$$if \quad \left|\overline{H} - H_p\right| < T_h and \left|\overline{S} - S_p\right| < T_s and \left|\overline{V} - V_p\right| < T_v \quad Then\ p(x,y) \text{ is a door pixel}$$

where H_p, S_p , and V_p are the HSV components of the pixel p at coordinates (x,y). The thresholds T_h, T_s , and T_v are

$$T_h = \left|\overline{H} - H_\sigma\right|, \quad T_s = \left|\overline{S} - S_\sigma\right|, \quad T_v = \left|\overline{V} - V_\sigma\right| \tag{11}$$

After the classification of the color pixels, blobs of less than 300 pixels are eliminated since they are not considered doors. Figure 9 illustrates some examples of doors detected by the previous method.

4 Recognition of Scenarios

The recognition of the scenarios is achieved with a hierarchical neural network, HNN. This type of architecture was selected due to the similarity of some of the scenarios. The HNN is composed of two levels, Figure 10. The first level is composed by one neural network and the second by two.

The idea of this HNN is to separate the scenarios into 4 classes, and then use the second level to resolve more specific cases.

Fig. 9. Door segmentation.

The first neural network is a feedforward – backpropagation network with 32 inputs, 32 neurons in the hidden layer, and 4 output neurons, sigmoid tangent activation functions in the hidden layer and sigmoid logarithmic functions in the output layer. This first neural network is trained to classify the four classes, see Figure 11, with the next feature vector

$$\mathbf{X}_1 = \begin{bmatrix} C_{Pxi} \\ C_{Pyi} \\ h_i \\ a_i \end{bmatrix} \tag{12}$$

here C_{pxi} and C_{pyi} are the centroids of the blobs that corresponds to the doors, while h_i and a_i are the normalized height and width, respectively, of those blobs. Figure 12 presents examples of the door blobs with their respective centroids.

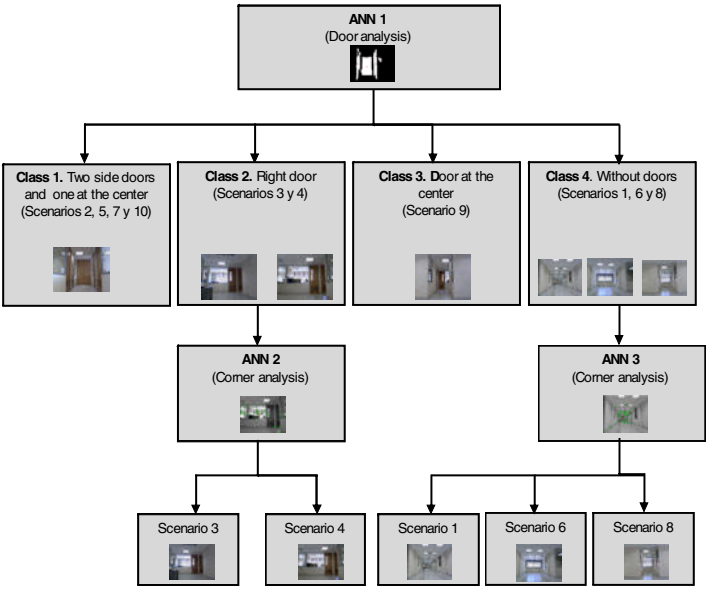


Fig. 10. Hierarchical neural network scheme.

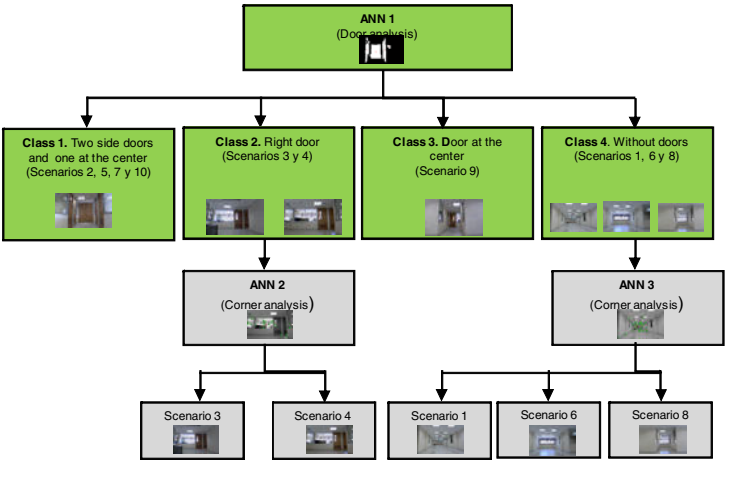


Fig. 11. First level of classification.

The neural network of the second level is trained to classify classes 2 and 4 into their corresponding scenarios, as shown in Figure 13, using the next feature vector

$$\mathbf{X}_{2,3} = \begin{bmatrix} C_{Ex} \\ C_{Ey} \\ N \end{bmatrix} \tag{13}$$

where C_{Ex} and C_{Ey} are the centroids of the corner coordinates and N is the number of corners found in the scenario. Figure 14 presents examples of the corners with their respective centroids.

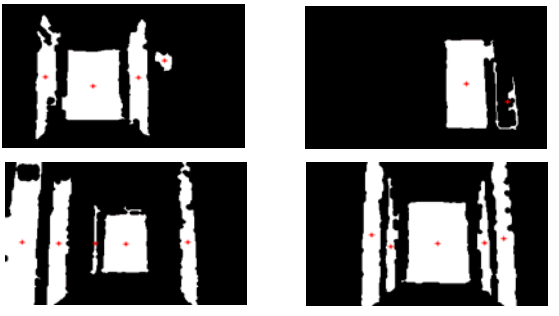


Fig. 12. Examples of door centroids.

The neural network of the second level is a feedforward -backpropagation, with 3 inputs, 26 neurons in the hidden layer, and 2 output neurons, sigmoid tangent activation functions in the hidden layer, and sigmoid logarithmic functions in the output layer.

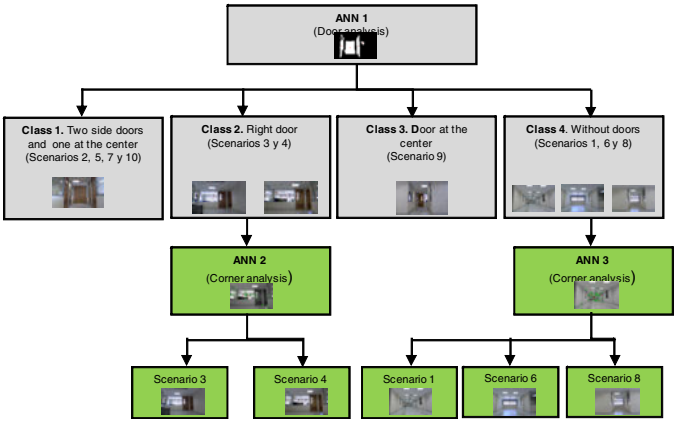


Fig. 13. Second level of classification.

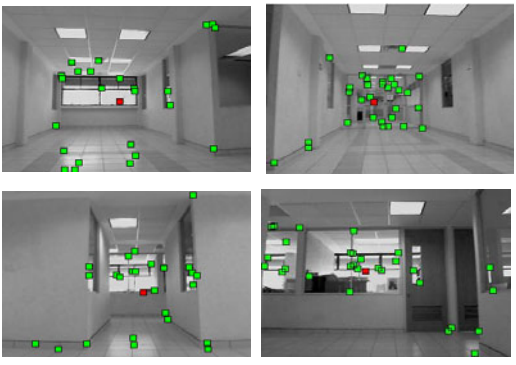


Fig. 14. Examples of corner centroids.

5 Results

Two important results are derived from this work. The first one is related to the proposed corner detector method and the second to the recognition for scenarios

In regards the corner detector method we can mention that the proposed method has similar performance in semi-artificial scenarios as the Harris detector, 98.41% versus 97.36 respectively. However, the propose method outperforms Harris in false positives, where the proposed method has 7.66%, while the Harris has a 33.33%.

With respect to real sceanrios the performance of Harris and SUSAN tend to detect more corners than the proposed method. However, the false positive rate is presumed to be very high, as proved with the semi-artificial images using the Harris method.

Results of the scenario recognition are commented next. The performance by levels of the HNN over the 10 scenarios considering the two levels is illustrated in Figure 14. The first classification was achieved by door detection using the centroids, height and area of the door blobs. The ANN of the first level was trained to classify the 10 scenarios into 4 classes. This ANN had 90% of correct classification during training, and 95% in validation. Class 1 that contains the scenarios 2, 5, 7, and 10 was considered as one type of scenario because of the high degree of similarity among the four scenarios. This similarity turns to be hard to resolve even for human observers. Class 3 was not reclassified because it only contains images of scenario 9. Regarding the classification of scenarios in the classes 2 and 4, in the second level, it was performed by using corner detection information as well as the number of corners. ANNs 2 and 3 were trained with this information. The ANN 2 separated class 2 into scenario 3 and 4wih a performance of 90.90% in training and 87.5% during validation. The neural network 3 that classifies class 4 into the scenarios 1, 6, and 8 has a performance of 93.75% in training, and 91.66% in validation.

The total performance of the systems considering all the scenarios was 86.66% for training, and 90% in validation.

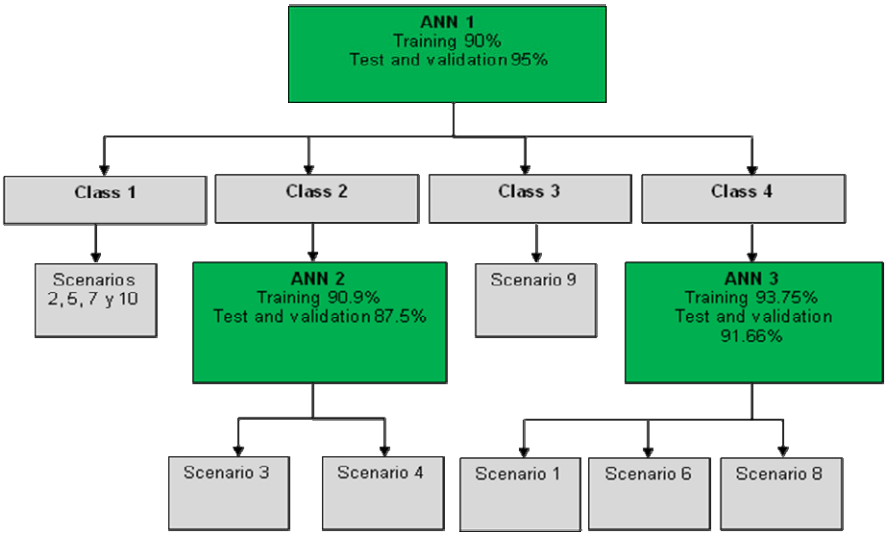


Fig. 15. System performance for levels.

6 Conclusions

In conclusion, it can be said that the proposed corner detector method shows good corner detection in semi-artificial as well as in real scenarios as it was validated in the corner detection experiments performed in this research. Besides, the corner detection method provides correct information that is validated with the performance achieved in the ANNs 2 and 3.

Another important conclusion is that the proposed solution to the scene recognition problem based on fusion of color and corner features proved to be effective based on the experimental results obtained in this work.

Results shown in this research confirm that complex problems like scene recognition for robot navigation are well faced with information fusion where different type of information complements each other.

Acknowledgments

This work was supported by SEP-DGEST under Grants 2173.09-P and 2172.09-P. The authors also thanks to Fondo Mixto de Fomento a la Investigación Científica y Tecnológica CONACYT- Gobierno del Estado de Chihuahua, by the support of this research under grant CHIH-2009-C02-125358.

References

[1] Kemp, C., Edsinger, A., Torres-Jara, D.: Challenges for Robot Manipulation in Human Environments. IEEE Robotics & Automation Magazine, 20–29 (March 2007)

[2] Durrant-Whyte, H., Bailey, T.: Simultaneous Localization and Mapping (SLAM): Part I. In: IEEE Robotics & Automation Magazine, June 2006, pp. 99–108 (2006)
[3] Bailey, T., Durrant-Whyte, H.: Simultaneous Localization and Mapping (SLAM): Part II. IEEE Robotics & Automation Magazine, 108–117 (September 2006)
[4] Addison, J., Choong, K.: Image Recognition For Mobile Applications. In: International Conferences on Image Processing ICIP 2007, pp. VI177–VI180 (2007)
[5] DeSouza, G., Kak, A.: Vision for Mobile Robot Navigation: A Survey. IEEE Transactions On Pattern Analysis And Machine Intelligence 24(2), 237–267 (2002)
[6] Kelly, A., Nagy, B., Stager, D., Unnikrishnan, R.: An Infrastructure-Free Automated Guided Vehicle Based on Computer Vision. IEEE Robotics & Automation Magazine, 24–34 (September 2007)
[7] Srinivasan, M.V., Thurrowgood, S., Soccol, D.: Competent Vision and Navigation Systems. IEEE Robotics & Automation Magazine, 59–71 (September 2009)
[8] Smith, S.M., Brady, J.M.: SUSAN - A New Approach to Low Level Image Processing. Int. Journal of Computer Vision 23(1), 45–78 (1997)
[9] Harris, C.G., Stephens, M.: A combined corner and edge detector. In: Proceedings of the Alvey Vision Conference, Manchester, pp. 189–192 (1988)
[10] He, X.C., Yung, N.H.C.: Curvature Scale Space Corner Detector with Adaptive Threshold and Dynamic Region of Support. In: Proceedings of the 17th International Conference on Pattern Recognition, August 2004, vol. 2, pp. 791–794 (2004)
[11] Rangarajan, K., Shah, M., van Brackle, D.: Optimal Corner Detector. In: Second International Conference on Computer Vision, pp. 90–94 (December 1988)
[12] Aguilar, G., Sanchez, G., Toscano, K., Salinas, M., Nakano, M., Perez, H.: Fingerprint Recognition. In: Second International Conference on Internet Monitoring and Protection, ICIMP 2007, July 2007, p. 32 (2007)
[13] Leung, W.F., Leung, S.H., Lau, W.H., Luk, A.: Fingerprint Recognition Using Neural Network, Neural Networks for Signal Processing [1991]. In: Proceedings of the 1991 IEEE Workshop, 30 September-1 October, pp. 226–235 (1991)
[14] Chen, Z., Birchfield, S.T.: Visual detection of lintel-occluded doors from a single image. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 1–8 (2008)
[15] Cariñena, P., Regueiro, C., Otero, A., Bugarín, A., Barro, S.: Landmark Detection in Mobile Robotics Using Fuzzy Temporal Rules. IEEE Transactions on Fuzzy Systems 12(4) (August 2004)

Improved Tabu Solution for the Robust Capacitated International Sourcing Problem (RoCIS)

Héctor Fraire Huacuja[1], José Luis González-Velarde[2], and Guadalupe Castilla Valdez[1]

[1] Instituto Tecnológico de Ciudad Madero, México.1o. de Mayo y Sor Juana I. de la Cruz S/N C.P. 89440, Cd. Madero Tamaulipas, México
hfraire@prodigy.net.mx, gpe_cas@yahoo.com.mx, haldape@hotmail.com

[2] Centro de Calidad y Manufactura, Tecnológico de Monterrey, Monterrey Nuevo León, México
gonzalez.velarde@itesm.mx

Abstract. In this paper the robust capacitated international sourcing problem (RoCIS) is approached. It consists of selecting a subset of suppliers with finite capacity, from an available set of potential suppliers internationally located. This problem was introduced by González-Velarde and Laguna in [1], where they propose a deterministic solution based on tabu search memory strategies. The process consists of three stages: build an initial solution, create a neighborhood of promising solutions and perform a local search in the neighborhood. In this work we propose improving the construction of the initial solution, the construction of the neighborhood and the local search. Experimental evidence shows that the improved solution outperforms the best solutions reported for six of the considered instances, increases by 13.6% the number of best solutions found and reduces by 34% the deviation of the best solution found, respect to the best algorithm solution reported.

Keywords: tabu search, robust optimization, robust capacitated international sourcing problem.

1 Introduction

The international sourcing problem consists of selecting a subset of suppliers, with a finite production capacity, from an available set of potential suppliers located internationally. In this paper we analyze the variant proposed in [1], which considers only a product in a single period and uncertainty on the demand and the exchange rate are modeled via a set of scenarios. In the formulation of this problem it is assumed that the costs depend on the economic conditions in the countries where the suppliers and the plants are located and that the production capacity of suppliers is finite. The robust formulation considers that a solution is feasible if and only if it is feasible in all the scenarios. The objective function minimizes the expected

P. Melin et al. (Eds.): Soft Comp. for Recogn. Based on Biometrics, SCI 312, pp. 333–347.
springerlink.com

value of the costs and penalizes the solutions whose optimal cost in some scenario surpasses the expected value of the optimal costs in all the scenarios. Through this mechanism the associated risk is incorporated.

The rest of our paper is organized as follows: related work, problem formulation, improved tabu solution and experimental results.

2 Related Work

Now we summarize the most relevant works from the literature about the plant location problem, because it is closely related to the international sourcing problem. Jucker and Carlson solve a single product, single period problem, with price and demand uncertainty [2]. Hodder and Jucker present a deterministic single period, single product model [3]. Hodder and Jucker optimally solve a single period, single product model, setting the plants quantity [4]. Haug approaches the deterministic problem with a single product and multiple periods with discount factors [5]. Louveaux and Peters solve a scenario-based problem in which the capacity is a first stage decision [6]. Gutierrez and Kouvelis explore the generation of scenarios to model price uncertainty and solve a simple plant location problem [7]. Kouvelis and You propose an un-capacitated version robustness approach based on a minimax regret criterion [8].

Now we describe the most relevant work about the international capacitated sourcing problem. The robust formulation of the international capacitated sourcing problem was proposed by Gonzalez-Velarde and Laguna [1]. In this work they propose a solution method based on the Benders paradigm, incorporating tabu search (TS) mechanisms. The process consists of building an initial solution, creating a neighborhood of promising solutions and performing a local search on the neighborhood. As the choice of the initial solution determines the efficiency of the process, this solution is constructed by applying a heuristic that gives preference to suppliers with lower fixed costs and greater production capacity.

González-Velarde and Martí propose a non-deterministic solution method based on GRASP, without incorporating the adaptive element, so the algorithm is classified as adaptive memory programming (AMP) type and path relinking is used to post processing the built solutions [9]. In the heuristic used to build a set of initial solutions, the shipping cost of each supplier to all plants is considered. The authors suggest that this way of incorporating the shipping cost seems too pessimistic.

In this work we propose to modify the TS based solution by improving the construction of the initial solution, the construction of the neighborhood and the local search.

3 Problem Formulation

The robust capacitated international sourcing problem (RoCIS) consists of selecting a set of suppliers to satisfy the demand for products at several plants located in different countries. The model deals with a single item in a single period. The

uncertainty in the demand and the exchange rates are modeled via a set of scenarios. The model uses the following definitions:

Parameters

N:	international plants set{1, 2,..., n}.
M:	international suppliers set {1, 2,..., m}.
S :	scenarios set
f_i:	fixed cost associated with supplier i.
c_{ij}:	total unit cost for delivering items from supplier i to plant j.
b_i:	capacity of supplier i.
d_{js}:	demand at plant j under scenario s.
e_{is}:	exchange rate at supplier's i location under scenario s.
p_s:	occurrence probability of scenario s.

Variables

x_{ijs}:	product shipment from supplier i to plant j under scenario s.
y_i:	1 if supplier i is contracted and 0 otherwise.

Given a supplier selection $y=[y_i]$ i=1,2,....m, then the problem becomes separable, and the following transportation problem must be solved for each scenario s:

Minimize

$$z_S = \sum_{i \in M} \sum_{j \in N} e_{is} c_{ij} x_{ijs} \tag{1}$$

subject to:

$$\sum_{i \in M} x_{ijs} \geq d_{js}, \qquad \forall j \in N \tag{2}$$

$$\sum_{i \in N} x_{ijs} \leq b_i y_i, \qquad \forall i \in M \tag{3}$$

$$x_{ijs} \geq 0, \qquad \forall i \in M \quad \forall j \in N \tag{4}$$

Then, the problem consists of minimizing:

$$F(y) = \sum_{s \in S} p_s \left(\sum_{i \in M} e_{is} f_i y_i + z_s \right) + w \sqrt{\frac{\sum_{s \in S^+} p_s (z_s - E(z))^2}{\sum_{s \in S^+} p_s}} \tag{5}$$

where $S^+ = \{s : z_s - E(z) \geq 0\}$ **and** $E(z) = \sum_{s \in S} p_s z_s$

4 Improved Tabu Solution

The solution method reported in [1] is a heuristic search based on Benders decomposition paradigm. An initial solution is constructed giving priority to the suppliers of smaller fixed cost and larger production capacity. For each supplier selection, the problem is decomposed into transportation subproblems, one for each scenario. The optimal dual solution for each sub problem is used to find a

promissory neighborhood and a local search in the neighborhood is carried out. The method uses several short term tabu memories, to monitor the suppliers used in the visited solutions [10]. As the search goes, the best found solution is updated and continues until finishing the neighborhood exploration. When the search stops, a new search begins in the best found solution neighborhood, and continues during 50 iterations.

4.1 Improving the Initial Solution Construction

The reported solutions to RoCIS problem considers two strategies to select the suppliers that must be incorporated into an initial solution [1, 9]. The first one gives priority to the smaller fixed cost suppliers and greater production capacity the second one incorporates the expected value of the products shipment cost from the selected supplier to *all* plants. The main limitation of the first strategy is that it does not consider the shipment cost, and even though this factor is considered the second one, the mechanism used is too pessimistic. In this work we propose to modify the incorporation mechanism of the shipment cost, *to include only the plants towards which the products shipment from the site of the supplier is less expensive.*

To describe this proposal, let $C_i=\{c_{ij} \mid j=1,2,...n\}$ the shipment costs set from supplier i to all the plants and BC_i a threshold cost defined on C_i. Then, the set of plants toward which the products shipment from the site of the supplier i is less expensive, can be defined as:

$$P_i^+ = \left\{ j \in N \,\middle|\, c_{ij} < BC_i \right\}$$

Now if c_{min} and c_{max} are the minimum and maximum of the shipment costs in C_i, then BC_i can be modeled as:

$$BC_i = c_{\min} + \alpha(c_{\max} - c_{\min}) \qquad \forall i \in M$$
$$where \qquad 0 < \alpha < 1$$

To include in G_i only the plants in P_i^+, must be defined as:

$$G_i = \frac{f_i + \left(\sum_{s \in S} p_s \left(\sum_{j \in P_i^+} c_{ij} e_{is} \right) \right)}{b_i}, \qquad \forall i \in M$$

The α value locates the initial solution on different regions of the search space, and it can be used as a diversification mechanism by dynamically changing its values.

4.2 Improving the Neighborhood Construction

To improve the quality of generated neighbors the mechanism used to determine the relative cost of the three types of movements that are applied to generate the

neighborhood (insert, delete and suppliers exchange) is modified. With the current r_i definition, *the movements are selected based on their impact on the growth rate of the objective function*, which could return in some cases an inappropriate choice. Currently r_i is defined as:

$$r_i = \begin{cases} \frac{E(\pi_i)}{f_i} & si \quad E(\pi_i) < 0 \\ f_i & si \quad E(\pi_i) = 0 \end{cases}$$

where f_i is the fixed cost of supplier *i*, $E(\pi_i) = \sum_{s \in S} p_s \pi_{is}$ is the expected dual price of supplier *i*, p_s is the occurrence probability of scenario *s* and π_{is} is the supplier *i* dual price in scenario *s*.

In opposite would be more appropriate to select the movements based on the net increase of the objective function generated when the movements are applied. Then r_i is redefined as:

$$r_i = \begin{cases} \frac{E(\pi_i) \cdot b_i}{f_i} & si \quad E(\pi_i) < 0 \\ f_i & si \quad E(\pi_i) = 0 \end{cases}$$

where b_i is the production capacity of supplier *i*.

4.3 Improving the Neighborhood Local Search

To improve the local search process is proposed to apply path re-linking on the two best global solutions founded when the iterations ends, after the second

Data structures used:

y´	prior best global solution found.
y´´	actual best global solution found.

For each pair of solutions *y´* and *y´´*:

Step 1: Determine the $y_\cap$ and $y_\cup$ solutions considering:

a. $y_{\cap i} = 1$ if $y'_i = 1$ and $y''_i = 1$, otherwise $y_{\cap i} = 0$,

b. $y_{\cup i} = 1$ if $y'_i = 1$ or $y''_i = 1$, otherwise $y_{\cup i} = 0$.

Step 2: Determine the *S'* set of selected suppliers in *y'* but not selected in *y''*

Step 3: Determine the *S''* set of not selected suppliers in *y'* and selected in *y''*

Step 4: Add to the $y_\cap$ solution the suppliers in set S' in appropriated order to reach y'

Step 5: Alternate between delete of *y'* a supplier of *S'* and add a supplier of *S''* until reach the *y''* solution

Step 6: Append to *y''*, one by one the suppliers of set *S'* in the appropriate order to reach $y_\cup$.

Fig. 1. Path re-linking algorithm.

Data structures used:

Supplier selection: $y = [y_1, y_2, ..., y_m]$

Hashing solution representation: $H(y) = \sum_{i \in M} y_i 2^i$

List of evaluated solutions: *coded_sol[H[y]]*

Tabu list of suppliers: *insertion, delete and swap*

Function InitialSolution(α)

Step 1: Calculate $D = \max_{s \in S}(\sum_{j \in N} d_{js})$

Step 2: Build a suppliers list in ascending sorted by

$$G_i = \frac{f_i + \left(\sum_{s \in S} p_s \left(\sum_{j \in P_i^+} c_{ij} e_{is} \right) \right)}{b_i}, \qquad \forall i \in M$$

where $P_i^+ = \left\{ j \in N \middle| c_{ij} < CF_i \right\}$ and

$$CF_i = c_{\min} + \alpha(c_{\max} - c_{\min}) \qquad \forall i \in M$$

$$where \qquad 0 < \alpha < 1$$

Step 3: Build the solution *y*, selecting suppliers in the sorted list until the sum of the Capacities of the selected suppliers is greater than D.

Step 4: Determinate *F*[*y*], solving the transportation sub problems generated in all scenarios.

Step 5: Record the *y* solution and its objective value *F*[*y*] in the list of evaluated Solution *coded_sol[H[y]]*

Step 6: Return the actual solution *y*.

Main algorithm

Step 1: *y* = InitialSolution(α)

Step 2: Repeat until 50 iterations

2.1 Generate a promising solution neighborhood of *y*

2.1.1 Determine the expected value of shadow prices (π_{is}) linked to the constraint corresponding to each supplier, in the solution of the 27 sub problems (for all the scenarios).

$$E(\pi_i) = \sum_{s \in S} p_s \pi_{is}$$

Fig. 2. Improved tabu solution (ITS) algorithm.

2.1.2 Calculate the suppliers relative cost (r_i.)

$$r_i = \begin{cases} \frac{E(\pi_i)\cdot b_i}{f_i} & si \quad E(\pi_i) < 0 \\ f_i & si \quad E(\pi_i) = 0 \end{cases}$$

2.1.3 For each of the possible insertions, deletions and swaps of suppliers that can be made from the solution, validate the feasibility of configuration with respect to the maximum demand *D*.

2.1.4 Build lists of candidate movement for insertion, deletion and swap, with *the movements identified in the previous step that are not stored in the tabu list for each type of movement*.

- The list of candidates for insertion contains the suppliers with the 3 lowest values of r_i.
- The list of candidates for deletion contains the suppliers with the 3 highest values of r_i.
- The list of candidates for swap contains the suppliers with the $\frac{(m^2-m)}{8}$ lowest values of r_j - r_i corresponding to the swap between supplier *i* by supplier *j* in configuration *y*

2.2 Local search process

2.2.1 For each configuration *y*' generated from the movements of the candidate lists of insertion, deletion and swapping:

2.2.1.1 The suppliers involved in the movement used to generate the configuration *y*' are: appended to the insertion tabu list (if the movement was for deletion), or removed (if the movement was to insertion). This two tabu lists are used too when a swap movement is applied, cnsidering that a swap movement requires a deletion and an insertion.. The number of iterations during which a supplier involved in a movement is considered tabu are: $\frac{m}{3}$ for insertions and eliminations and $\frac{m(m-1)}{16}$ for swaps.

2.2.1.2 If the solution *y*' is already saved in the list of evaluated solutions, its objective value *F*[*y*'] is retrieved, otherwise *F*[*y*'] is calculated and appended to the list.

2.2.1.3 Update the best solution found y_{best}.

2.2.2 Path relinking process

2.2.2.1 From the second iteration, update the two best solutions found.

2.2.2.2 Apply path relinking to the two best solutions found and update y_{best}

Fig. 2. (*continued*)

iteration. The path re-linking strategy used is basically the described in [9]. The algorithm used to perform the process is shown in Figure 1.

Figure 2 shows the detailed improved tabu solution (ITS) algorithm, which incorporates the construction of the initial solutions, the construction of neighborhoods and the local search proposed.

5 Experimental Results

The experiments were done in a computer Dell Optiplex 160L with a Pentium IV processor to 2.4 GHZ and 1 GB ram. The source code was compiled using Visual C 6.0 and the operating system Windows XP. For the solution of the transportation sub problems LINDO API 2.0 was used. To evaluate the performance of the algorithms, the larger instances with 20 plants, 40 suppliers and 27 scenarios reported in [9] were used. As the optimal solutions for these instances are not known, the results obtained in this work are compared with the best solutions reported in [9]. To evaluate the impact of the improvements on the TS algorithm performance, two experiments were carried out.

In the first experiment the proposed improvements for the initial construction, the construction of the neighborhood and local search were evaluated. Table 1 shows the results obtained with the improved tabu solution (ITS α), solving the instances with different α values. In the table the better solutions are emphatized. The table shows that the improved TS found better solutions than those reported for instances 16, 18, 20, 21, 27 and 30. As we can observe the best global solutions found by AMP are also found by ITS using one or more of the α values. However, there is not a single α value which allows to ITS find the best global solution for all instances. Experimental evidence confirms that the α value operates as a diversification mechanism on the search process.

Table 2 shows a summary of the experimental results, including: the average cost of the found solutions, the number of overall best solutions found, the error rate over the average cost of best solution and the average time used to solve each instance (in CPU seconds). As we can observe, ITS achieves its best performance for α=0.2 increases 8.3% the number of best solutions found and reduces 50% the deviation from the best solution found, respect to AMP algorithm. However, the average execution time required to solve each instance increases 75%.

In the second experiment, the number of iterations was reduced to 30 to reduce the resources consumption of ITS. Table 3 shows the results obtained, and we can observe that ITS (for α=0.2) outperfoms to AMP. ITS increases 13.6% the number of best solutions found, reduces 34% the deviation from the best solution found, respect to AMP algorithm. But now, the average execution time required to solve each instance increases only 8%.

Moreover, in Table 1 we can observe that 4 improved tabu solution (ITS) algorithms obtains the best known solutions for 19 instances, and similarly 3 ITS algorithms for 3 instances, 2 ITS algorithms for 4 instances and 1 ITS algorithm for 4 instances. Now we consider the division of the instances in four groups: I_1, I_2, I_3 and I_4. Where the group I_n contains all the instances for which n algorithms

Table 1. Performance of the improved tabu solution ITS (with α= 0.2, 0.4, 0.6, 0.8 and 50 iterations)

	AMP	ITS 0.2		ITS 0.4		ITS 0.6		ITS 0.8	
1	**33178.634**	**33178.634**	12	**33178.634**	10	**33178.634**	9	**33178.634**	8
2	**44181.482**	**44181.482**	12	**44181.482**	10	**44181.482**	9	**44181.482**	10
3	**39558.824**	**39558.824**	12	**39558.824**	9	**39558.824**	7	**39558.824**	6
4	**47120.476**	**47120.476**	10	**47120.476**	7	**47120.476**	9	**47120.476**	16
5	**41515.933**	**41515.933**	12	**41515.933**	8	**41515.933**	10	**41515.933**	13
6	**41285.573**	**41285.573**	20	**41285.573**	17	**41285.573**	5	**41285.573**	13
7	**42015.045**	**42015.045**	7	**42015.045**	12	**42015.045**	7	**42015.045**	11
8	**55627.074**	**55627.074**	10	**55627.074**	9	**55627.074**	11	**55627.074**	7
9	**46055.986**	**46055.986**	6	**46055.986**	5	**46055.986**	10	**46055.986**	9
10	**57188.416**	**57188.416**	32	**57188.416**	28	**57188.416**	2	**57188.416**	3
11	**60692.588**	**60692.588**	6	**60692.588**	11	**60692.588**	9	**60692.588**	20
12	**55603.798**	**55603.798**	11	55617.163	9	**55603.798**	10	55617.163	23
13	**67389.803**	**67389.803**	14	68158.761	13	68158.761	9	68158.761	7
14	**65420.806**	65595.875	9	65427.008	24	**65420.806**	10	**65420.806**	10
15	**78184.024**	**78184.024**	6	**78184.024**	7	**78184.024**	21	**78184.024**	36
16	38094.866	**37809.009**	22	**37809.009**	12	**37809.009**	11	37820.650	11
17	**34109.310**	**34109.310**	10	**34109.310**	12	**34109.310**	9	**34109.310**	9
18	34127.480	**33814.099**	10	**33814.099**	9	**33814.099**	9	**33814.099**	12
19	**40558.798**	**40558.798**	9	**40558.798**	11	**40558.798**	11	40570.844	30
20	32210.967	**31496.848**	11	**31496.848**	12	**31496.848**	9	**31496.848**	12
21	41551.650	41741.155	11	**41527.770**	7	41741.155	7	41741.155	10
22	**38833.676**	**38833.676**	8	**38833.676**	25	**38833.676**	11	**38833.676**	10
23	**44391.636**	**44391.636**	7	**44391.636**	12	**44391.636**	13	**44391.636**	11
24	**41831.945**	**41831.945**	18	**41831.945**	6	**41831.945**	6	**41831.945**	28
25	**53709.188**	**53709.188**	7	**53709.188**	8	54180.966	43	**53709.188**	10
26	**61377.260**	**61377.260**	11	**61377.260**	10	**61377.260**	8	**61377.260**	19
27	69464.057	69541.176	27	69496.302	9	**69464.046**	26	**69464.046**	45
28	**75482.597**	**75482.597**	5	**75482.597**	20	75952.113	4	**75482.597**	20
29	**61818.891**	62170.326	28	**61818.891**	5	61963.374	44	61851.609	30
30	68193.731	**68073.378**	11	68193.720	24	68193.720	18	68292.717	17

obtains the best known solutions. We can consider that for $n>m$, the instances in I_n are easiest than the instances in I_m. Then the question is ¿ a structural and landscape analysis of the instances can help us to explain the relative hardness observed? To search an answer to this question two additional experiments were realized to analyze the instances estructure and the ruggdness of the landscape.

Table 2. Comparative summary of the performance of the improved tabu solution ITS with 50 iterations

	AMP	**TS**	**ITS 0.2**	**ITS 0.4**	**ITS 0.6**	**ITS 0.8**
Val.	50359.151	51730.250	50337.797	50341.934	50383.512	50352.945
# Bests	24	1	26	25	24	23
Desv.	0.10%	2.93%	0.05%	0.06%	0.14%	0.08%
CPU secs	218.23	381.25	374.22	392.92	371.75	368.97

Table 3. Comparative summary of the performance of the improved tabu solution ITS with α=0.2 and 30 iterations

	AMP	**ITS 0.2**
Val.	50,359.151	50,344.086
#Bests	22	25
Dev.	0.10%	0.066%
CPU secs	218.23	235.76

Table 4. Structural information of the instances groups I_1, I_2, I_3 and I_4, respect to the variation coefficient.

Shipment cost						Demand				
	I_1	I_2	I_3	I_4			I_1	I_2	I_3	I_4
Min	0.45	0.46	0.46	0.45		Min	0.15	0.15	0.16	0.15
Median	0.48	0.48	0.47	0.47		Median	0.16	0.16	0.16	0.16
Max	0.48	0.49	0.49	0.50		Max	0.16	0.16	0.16	0.17
Mean	0.47	0.48	0.47	0.47		Mean	0.16	0.16	0.16	0.16
Supplier capacity						Exchange rate				
Min	0.10	0.11	0.11	0.10		Min	0.10	0.10	0.10	0.10
Median	0.12	0.11	0.12	0.12		Median	0.10	0.10	0.10	0.10
Max	0.12	0.12	0.12	0.13		Max	0.10	0.10	0.10	0.10
Mean	0.12	0.11	0.12	0.11		Mean	0.10	0.10	0.10	0.10
Fixed cost										
Min	0.10	0.11	0.11	0.10						
Median	0.12	0.11	0.12	0.12						
Max	0.12	0.12	0.12	0.13						
Mean	0.12	0.11	0.12	0.11						

Table 5. Structural information of the instances groups I_1, I_2, I_3 and I_4, respect to the skewness.

Shipment cost						Plants demand				
	I_1	I_2	I_3	I_4			I_1	I_2	I_3	I_4
Min	0.08	0.13	0.17	-0.03		Min	-0.05	-0.10	-0.06	-0.12
Median	0.17	0.17	0.25	0.19		Median	-0.04	-0.01	0.02	-0.03
Max	0.23	0.25	0.38	0.35		Max	0.03	0.12	0.05	0.10
Mean	0.16	0.18	0.27	0.17		Mean	-0.02	0.00	0.00	-0.01
Supplier capacity						Exchange rate				
Min	-0.10	0.00	-0.30	-0.56		Min	-0.03	-0.08	0.05	-0.06
Median	-0.03	0.13	0.00	-0.05		Median	0.00	-0.01	0.06	0.02
Max	0.50	0.49	0.22	0.64		Max	0.09	0.05	0.07	0.06
Mean	0.09	0.19	-0.03	-0.01		Mean	0.01	-0.01	0.06	0.01
Fixed cost										
Min	-0.10	0.00	-0.30	-0.56						
Median	-0.03	0.13	0.00	0.08						
Max	0.50	0.49	0.22	0.64						
Mean	0.09	0.19	-0.03	0.09						

Table 6. Average of the *l* values obtained in five random walks with two lengths: 1000 and 50000 steps

Walk	*l* average (Length=1000)	*l* average (Length=50000)
1	0.25	0.19
2	0.26	0.18
3	0.28	0.20
4	0.28	0.19
5	0.25	0.20

In the third experiment the structural analysis of the instances was done. For all instances the sparsity, the variation coefficient and the skewness of the instance parameters (shipment cost matrix, fixed cost vector, capacity supplier vector, plants demand matrix and currency rate matrix) were calculated. The sparsity measures the percentage of parameter structure elements that are equal to zero; the

main interest in this measure is that according to Mitchell and Borchers, it has a strong influence on algorithm behavior [11]. The *variation coefficient* (VC) is defined as σ / X where σ is the standard deviation and X the mean of the structure elements. VC gives an estimate of the variability of the structure elements, independent of their size. The *skewness* is the third moment of the mean normalized by the standard deviation; it gives an indication of the degree of asymmetry of the structure elements. In all the experiments the instances were considered grouped in I_1, I_2, I_3 and I_4. Then the sparsity, the variation coefficient and skewness were calculated for the five components of each instance: shipment cost matrix, fixed cost vector, capacity supplier vector, plants demand matrix and currency rate matrix. For all the instances and components the observed sparsity percentage was zero. Table 4 contains the obtained results for the variation coefficient and Table 5 the results for the skewness. As we can observe the four instances groups shows a similar structure, because the differences between the values of the variation coefficient and of the skeeness are minimal.

In the fourth experiment a ruggedness analysis of the landscape was done. The central idea of the landscape analysis in combinatorial optimization is to represent the space searched by an algorithm as a landscape formed by all feasible solutions and the objective value assigned to each solution [12]. The information generated with the landscape analysis is used to gain knowledge about: the search space characteristics and their relation with the behavior of local search or metaheuristics algorithms [13, 14], the possibility to predict problem or problem instance hardness [15, 16], or indications on useful parameterizations of local search algorithms [17]. A search landscape is considered rugged if there is a low correlation between neighboring points. To measure this correlation a *random walk* of length *m* is performed in the search landscape to interpret the resulting series of *m* points $\{f(x_t), t=1,\ldots, m\}$ as a time series. The autocorrelation $r(s)$ of the points in the series that are separated by *s* steps is defined as:

$$r(s) = \frac{1}{\sigma^2(f)(m-s)} \sum_{t=1}^{m-s} (f(x_t) - \bar{f})(f(x_{t+s}) - \bar{f})$$

where $\sigma^2(f)$ and $\bar{f}$ are the variance and the mean of the points in the series. Now the *search landscape correlation length* is defined as

$$l = -\frac{1}{\ln(|r(1)|)}$$

where $|r(1)| \neq 0$. Then the lower is the *l* value, the more rugged is the landscape [18].

Previously to the determination of the search landscape correlation (*l*) values for the instances in the considered groups, we must determine the length of the random walk to be applied. For this purpose were calculated five times the average of the *l* values of all the instances with a random walk length given. The obtained results, with random walk lengths of 1000 and 50000 steps, are showed

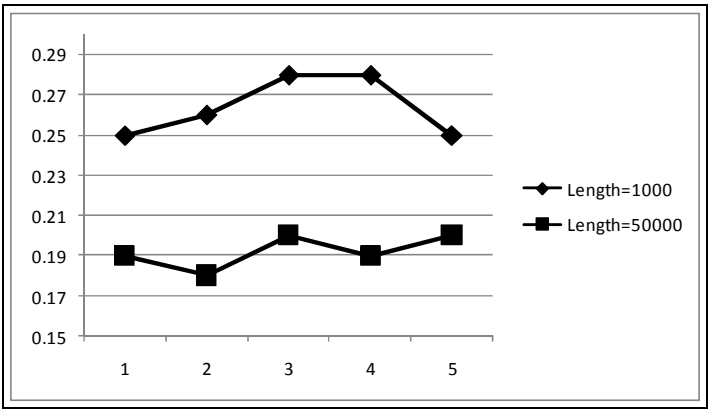


Fig. 3. Average of the *l* values obtained in five random walks with two lengths: 1000 and 50000 steps

Table 7. Average of the *l* values obtained with a random walk of 50000 steps, for the groups I_1, I_2, I_3 and I_4

	Search landscape correlation length (*l* value)			
	I_1	I_2	I_3	I_4
minimum	0.17	0.16	0.13	0.14
median	0.20	0.18	0.16	0.19
maximum	0.22	0.21	0.17	0.23
mean	0.19	0.18	0.15	0.19

in Table 6 and Figure 3. As we can observe, with 1000 steps the average of the *l* values varies from 0.25 to 0.28 and for 50000 steps varies from 0.18 to 0.20. Given the high resource consumption required to solve the ROCIS instances, we consider that with 50000 steps the *l* values shows an appropriated precision level and stability. Now we calculate the *l* values for the instances in each group using a random walk with 50000 steps. Table 7 shows the minimum, median, maximum and the mean of the average of the *l* values for each group (I_1, I_2, I_3, and I_4). We can observe that do not exist a significant difference respect to the landscape ruggedness generated for the random walk with the instances in the groups. All the average *l* values are very similar and closer to zero. The random walk algorithm seems perceive that the instances have a high hardness level regardless of the group that they belong. Given the high ruggedness of the landscape, the different levels of instances hardness perceived for the ITS algorithm can be explained by the high intensification used in the search process. It seems more appropriate to use algorithms that apply more diversification to solve the considered instances.

6 Conclusions and Future Work

This paper approaches the robust capacitated international sourcing problem (RoCIS) which consists of selecting a subset of suppliers with finite capacity, from an available set of potential suppliers internationally located. The tabu solution proposed in [1] consists of three phases: build an initial solution, create a neighborhood of promising solutions and perform an extensive search in the neighborhood. In this work the construction of the initial solution, the construction of the neighborhood, and the local search were improved. Experimental evidence shows that the improved tabu solution outperforms the best solution reported for six of the instances considered, increases 13.6% the number of best solutions found and reduces 34% the deviation from the best solution found, respect to the best algorithm solution reported.

The structural analysis shows that the considered instances groups have a similar structure, because the differences between the variation coefficient and the skewness are minimal. Moreover, the landscape analysis shows that do not exist a significant difference respect to the landscape ruggedness generated for the random walk with the instances of the groups. The random walk algorithm seems perceive that all the instances have a high hardness level regardless of the group that they belong. Given the high ruggedness of the landscape, the different levels of instances hardness perceived for the ITS algorithm can be explained by the high intensification applied in the search process. It seems more appropriate to use algorithms that apply more diversification to solve the considered instances.

As in the improved tabu solution the α value locates the initial solution on different regions of the search space, it can be used as a diversification mechanism in the local search by dynamically changing its values. We are now incorporating this diversification mechanism in the ITS algorithm.

Acknowledgments

Authors thank the support received from Tecnológico de Monterrey, Consejo Nacional de Ciencia y Tecnología (CONACYT) and Consejo Tamulipeco de Ciencia y Tecnología (COTACYT) through projects CAT128, CONACYT-67032 and TAMPS-2007-C15-106096, for the research reported in this paper.

References

1. González-Velarde, J.L., Laguna, M.: A Benders-based heuristic for the robust capacitated international sourcing problem. IIE Transactions 36, 1125–1133 (2004)
2. Jucker, J.V., Carlson, R.C.: The Simple Plant-Location Problem under Uncertainty. Operations Research 24(6), 1045–1055 (1976)
3. Hodder, J.E., Jucker, J.V.: Plant Location Modeling for the Multinational Firm. In: Proceedings of the Academy of International Business Conference on the Asia-Pacific Dimension of International Business, Honolulu, December 1982, pp. 248–258 (1982)

4. Hodder, J.E., Jucker, J.V.: A Simple Plant-Location Model for Quantity-Setting Firms subject to Price Uncertainty. European Journal of Operational Research 21 (1985)
5. Haug, P.A.: Multiple-Period, Mixed-Integer-Programming Model for Multinational Facility Location. Journal of Management 11(3), 83–96 (1985)
6. Louveaux, F.V., Peters, D.: A dual-based procedure for stochastic facility location. Operations Research 40(3), 564–573 (1992)
7. Gutiérrez, G.J., Kouvelis, P.: A Robustness Approach to International Sourcing. Annals of Operations Research 59, 165–193 (1995)
8. Kouvelis, P., Yu, G.: Robust Discrete Optimization and its Applications. Kluwer Academic Publishers, Dordrecht (1997)
9. González-Velarde, J.L., Martí, R.: Adaptive Memory Programming for the Robust Capacitated International Sourcing Problem. Computers and Operations Research 35(3), 797–806 (2008)
10. Glover, F., Laguna, M.: Tabu Search. Kluwer Academic Publishers, Dordrecht (1997)
11. Mitchell, J.E., Borchers, B.: Solving linear ordering problems with a combined interior point/simplex cutting plane algorithm. In: Frenk, H.L., et al. (eds.) High Performance Optimization, pp. 349–366. Kluwer Academic Publishers, Dordrecht (2000)
12. Merz, P., Freisleben, B.: Fitness landscapes and memetic algorithm design. In: Corne, D., Dorigo, M., Glover, F. (eds.) New Ideas in Optimization, pp. 245–260. McGraw-Hill, London (1999)
13. Boese, K.D.: Models for Iterative Global Optimization. PhD thesis, University of California, Computer Science Department, Los Angeles, CA, USA (1996)
14. Stutzle, T., Hoos, H.H.: MAX-MIN Ant System. Future Generation Computer Systems 16(8), 889–914 (2000)
15. Angel, E., Zissimopoulos, V.: On the classification of NP-complete problems in terms of their correlation coefficient. Discrete AppliedMathematics 99, 261–277 (2000)
16. Stadler, P.F., Schnabl, W.: The landscape of the travelling salesman problem. Physics Letters A 161, 337–344 (1992)
17. Angel, E., Zissimopoulos, V.: Autocorrelation coefficient for the graph bipartitioning problem. Theoretical Computer Science 191, 229–243 (1998)
18. Weinberger, E.D.: Correlated and uncorrelated fitness landscapes and how to tell the difference. Biological Cybernetics 63, 325–336 (1990)

Variable Length Number Chains Generation without Repetitions

Carpio Martín, Soria-Alcaraz Jorge A., Puga Héctor J., Baltazar Rosario, Ornelas Manuel, and Mancilla Luís Ernesto

Instituto Tecnológico de León, Maestría en ciencias en ciencias de la computación, León Guanajuato México
jmcarpio61@hotmail.com, soajorgea@gmail.com, pugahector@yahoo.com, charobalmx@hotmail.com, mornelas67@yahoo.com.mx, lmancilla01@hotmail.com

Abstract. Pseudorandom and random numbers generators, plays an important role in solving many real or simulated problems, in different domains such as Scientific Computing, Physics, Chemistry, Computer Science, Artificial Intelligence, Chaos, Games theory, Statistics, Economics, etc. that directly or indirectly include a probabilistic element. These generators can be found in calculators, compilers, spreadsheets, electronics files or library tables, However, the progressive use of increasingly sophisticated models will demand a fast pseudorandom number generation process, which can generate strings of arbitrary sizes, and ensure it's reproducibility, uniformity and statistical independence, hence it constitutes an active research field area. This paper presents a novel method for obtaining these numbers relevant to various branches of computational optimization.

1 Introduction

Random numbers are a staple in the simulation of most discrete systems. One of the biggest problems facing people who use heuristics in solving problems is the generation of an initial population of random solutions that meet the following conditions:

A) Reproducibility
B) No repetition inside a determinate length chain.
C) Statistical independence
D) Fast generation process
E) A minimal space inside memory.

The solution to this problem can be found with pseudorandom number generator. The main idea is to generate statistically random numbers with repeatable characteristics.

P. Melin et al. (Eds.): Soft Comp. for Recogn. Based on Biometrics, SCI 312, pp. 349–364.
springerlink.com

2 Linear Congruential Method

Most pseudorandom number generators are based in the general congruential equation, [13], [10], [15], [12], [2], [11], [1] that can be expressed as following:

$$n_{i+1} \equiv (an_i + c) \,(\mathrm{mod}\, m)\,, \tag{1}$$

For $i = 0, 1, 2, \ldots$,and n_i, a, c y m are non-negative numbers. n_0 is the seed or initial value. If $c \neq 0$ Equation (1) is called *mixed method of congruencies*. When $c = 0$, our equation (1), is known as *multiplicative method of congruencies*. Whit an initial value n_0, a constant factor a and an additive constant c, equation (1) give us a congruence relation (module m) for all i, generating a remainder succession $\{n_1, n_2, \ldots, n_i, \ldots\}$ module m. The previous implies that $n_i < m$. Starting with the numbers from the $\{n_i\}$ succession, we can obtain rational numbers inside an interval (0,1).

Example: Use the general congruencies equation (1), to generate a succession of numbers with $n_0 = 27$, $a = 17$, $c = 43$ y $m = 100$.

$n_0 = 27$,

$$n_1 \equiv [(17(27) + 43] \bmod 100 = 502 \bmod 100 = 2, \qquad r_1 = \frac{2}{100} = 0.02,$$

$$n_2 \equiv [(17(2) + 43] \bmod 100 = 77 \bmod 100 = 77, \qquad r_2 = \frac{77}{100} = 0.77,$$

$$n_3 \equiv [(17(77) + 43] \bmod 100 = 1352 \bmod 100 = 52 \qquad r_3 = \frac{52}{100} = 0.52,$$

$$n_4 \equiv [(17(52) + 43] \bmod 100 = 927 \bmod 100 = 27, \qquad r_3 = \frac{27}{100} = 0.27,$$

$$\vdots \qquad\qquad \vdots$$

We can see that $n_4 = n_0 = 27$, so from $i=4$ on, the succession values repeats. to the number of values without repetition is known as *Maximum Period* , and is denoted with the letter h , In the previous example $h=4$.

2.1 Calculation of n_0, a, c and m Parameters

Now we perform an interrelation analysis that can be achieved for n_0, a, c y m, in order to satisfy the maximum period property and statistical independence of the numbers generated by equation (1)

Using equation (1) for $i = 0, 1, 2, \ldots$,we have:

$$n_1 \equiv (an_0 + c) \,(\mathrm{mod}\, m)$$

$$n_2 \equiv (an_1 + c) \,(\mathrm{mod}\, m) = [a(an_0 + c) + c](\mathrm{mod}\, m) = [a^2 n_0 + (a+1)c](\mathrm{mod}\, m)$$

$$n_3 \equiv (an_2 + c) \pmod m = [a(a^2 + (a+1)c) + c] \pmod m$$
$$\equiv [a^3 n_0 + (a^2 + a + 1)c] \pmod m$$
$$\vdots$$
$$n_i \equiv [a^i + (a^{i-1} + a + 1)c] \pmod m . \quad (2)$$

Using the known equation for geometric progression:

$1 + a + a^2 + ... + a^s = \frac{a^{s+1} - 1}{a - 1}$, *to* $a \neq 1$, then the equation (2) takes the form:

$$n_i \equiv \left[a^i n_0 + c\left(\frac{a^i - 1}{a - 1} \right) \right] \pmod m . \quad (3)$$

As can be seen in equation (3), we can easily deduce that, with n_0, a, c y m given, the values n_1 and n_2 are totally determinate. Several questions arise at this point, Does exists a minimum positive value for i,($i=h$), when $n_h = n_0$ where h is the period of the $\{n_i\}$succession ?, If h does exists, What values need to be assigned to n_0, a, c y m in order to achieve the longest succession period possible?. This question is important because if $n_i = n_0$, for some $i = h$, then $n_{h+1} = n_1$, $n_{h+2} = n_2$,etc. So this succession is going to be repeated after a h period. Another difficult question is What values we can be assigned to n_0, a, c y m, in order to have a statistical independent $\{n_0, n_1, \ldots, n_h\}$,succession and for all practical randomness considerations?

2.1.1 Answers

The first question has an affirmative answer, Number Theory has demostrated this period exists and its value depends on m [Naylor, 1977, Cáp. 3], [L'Ecuyer, 2006]. This is equivalent to say that there is not possible to obtain a succession without repetitions using congruencies methods. However in practice the period of a succession can be fixed in a very high value only when a big enough module is chosen.

The third question have an affirmative answer also, whenever the parameters a, c y m achieve the minimum theorical conditions of minimum correlation established by [3], [5], [6], [8] According to these references " the magnitude of a seriated correlation $\rho(n_i, n_{i+1})$ is founded between the values

$$\frac{1}{a} - \left(\frac{6c}{am} \right) \left(1 - \frac{c}{m} \right) \pm \frac{a}{m} . \quad (4)$$

The Idea is to find values for the parameters a, c y m that minimize the equation (4) and ideally make it zero."

The answer of the second questions can be founded in [7], [8], [9], Most of the programming and simulation languages uses a module $m = p^e$, where p is a prime number or a product of prime numbers and e denotes the number of digits in a word according to processor. Since computers uses decimal or binary system, we will focus only on cases when $p = 2$ o $p = 10$.

1. For m a potency of 2, fo example $m = 2^b$ and $c \neq 0$, the longest possible period generated is $h = m = 2^b$, this require that c be a prime relative number with m, i.e *g.c.d(c,m)*=1, and $a \equiv 1 mod(p)$, if p is a prime factor of m, i.e $a=1+pk$ para $k = 0, 1, 2, \ldots$ o $a \equiv 1 mod(4)$ if 4 is a factor of m, i.e , $a=1+4k$ para $k = 0, 1, 2, \ldots$ applying Coveyou and Greenberger conditions we found an approximated condition for a y c, $a = 2^{\left[\frac{b}{2}\right]} + 1$, parameters where $\left[\frac{b}{2}\right] = \begin{cases} \frac{b}{2} & if \quad b = par \quad number \\ \frac{b-1}{2} & if \quad b = non-par \quad number \end{cases}$, c need to have non-par positive values, ther are non conditions about n_0, which can be take any positive value less to m, We need to said that this is a necessary condition and do not guarantee a minimum seriated correlation, only we can assure the Coveyou and Greenberger condition applying several statistical tests of randomness to the generated numbers by the combinations of parameters, leaving the determination of conditions for a future work.
2. For m a potency of 2, for example $m = 2^b$, y $c = 0$, the longest possible period is generate $h = m = 2^{b-2}$, which require a non-even seed and a multiplicative factor a who satisfies the relation $a \equiv \pm 3 mod(8)$, then the solution is $a = 3 + 8k$ o $a = 5 + 8k$, for $k = 0, 1, 2, \ldots$. Then applying the Coveyou and Greenberger solution like $c = 0$, we find a direct way a necessary and sufficient condition for the a, c, n_o and m, parameters that ensures the generation of pseudorandom and statistical independent numbers, this is $a \approx \sqrt{m}$. Therefor order to establish an optimal a value, we need to assign $h = 2^{b-2}$, $m = 2^b$ then , $a \approx \sqrt{m} = 2^{\frac{b}{2}}$. Now take the value $a=3+8k$ *or* $a=5+8k$, *for* $k = 0, 1, 2, \ldots$, that is closer to, $a \approx \sqrt{m} = 2^{\frac{b}{2}}$. Finally a non-even positive seed is established.

   Example: If $h = 2^{7-2} = 32$, $m = 2^7 = 128$, then $a \approx \sqrt{128} = 11.31$ Take $k = 1$, to obtain $a = 11$ o $a = 13$, then we take $a = 11$, that is the closer a value to 11.31. So our final equation that generate pseudorandom numbers is $n_{i+1} \equiv (11 n_i) \pmod{128}$, $n_0 = 1, 3, 5, \ldots, m-1$, $i = 0, 1, 2, \ldots, h-1$.
3. For a m prime number and $c=0$, the longest possible period to generate is $h=m-1$. It is necessary to select several values for a who that have the property that the smallest value of k, $a^k \equiv 1 \mathrm{mod}(m)$ be $k=m-1$. Therefore the parameter a be a primitive root of m *and* n_0 and a non-par number, [Naylor, 1978]. **Example** If $m = 17$, $n_0 = 1$, $a = 3$, satisfies the established condition . applying Coveyou and Greenberger condition , as $c = 0$, we find a direct way to have a necessary and sufficient condition with a and c, n_o

and m, parameters to generate pseudorandom and statistical independent numbers. This condition is , $a \approx \sqrt{m}$. However the optimal value for a becomes harder to find that the last example.

4. For m potency of 10, like $m = 10^d$, y $c \neq 0$, the longest possible period generated is $h = m = 10^d$, so we need that c be a relative prime number to m, like $(c, 2) = 1$, and $(c, 5) = 1$, and $a \equiv 1 \bmod(20)$. Applying Coveyou y Greenberger, conditions we found an approximated condition to a and c, $a = 2^{\left[\frac{d}{2}\right]} + 1$, where $\left[\frac{d}{2}\right] = \begin{cases} \frac{d}{2} & if \quad d = par \quad number \\ \frac{d-1}{2} & if \quad d = non-par \quad number \end{cases}$, c needs to take a non-even positive values and relative prime numbers to 5, i.e.$(c, 5) = 1$, there are not conditions for n_0, which can take any smaller value than m. We must say that this is a necessary condition but do not guarantee minimum seriated correlation, we can only assure that the Coveyou and Greenberger condition applying several statistical tests of randomness to the generated numbers by the combinations of parameters, leaving for determinate the enough conditions for a future work.
5. For m a potency of 10, as $m = 10^d = 2^d 5^d$, y $c = 0$, the longest possible period generated is $h = m = 5\text{x}10^{d-2}$ [IBMC, 1959], [Naylor, 1978], which require a seed n_0 non-even and a relative prime to 5, as $(n_0, 5) = 1$; and a multiplicative factor that satisfies the relation $a \equiv \pm 3 \bmod(8)$, the solution is either $a = 3 + 8k$ or $a = 5 + 8k$, and $(a, 5) = 1$, for $k = 0, 1, 2, \ldots$. The values of a multiplier with period $h = m = 5\text{x}10^{d-2}$ can be assign into 36 different residual classes module 200, given by:

$$a \equiv \pm\{3,11,13,19,21,27,29,37,53,59,61,67,69,77,83,91,93,99\} (\text{mod } 200) \quad (5)$$

In consequence, a can be expressed by:

$$a = 200k \pm p; \; with \; k = 0,1,2,3,4,5,\cdots \quad (6)$$

where p is one of 36 numbers given by equation (5). Applying Coveyou y Greenberger conditions, as c = 0 , we find a direct way to have a necessary and sufficient condition with a and c, n_o and m, parameters to generate pseudorandom and statistical independent numbers. This condition is $a \approx \sqrt{m}$. Therefore we can establish an optimal a value:

Establish $h = 5\text{x}10^{d-2}$, m = 10^d, calculate $a \approx \sqrt{m} = 2^{d/2}$. Take a value for $a = 3 + 8k$ o $a = 5 + 8k$, then $(a, 5) = 1$, for k = 0, 1, 2,….., who is closer to $a \approx \sqrt{m} = 2^{d/2}$. Assign a positive non-par seed to $(n_0, 5) = 1$.

Example: if $h = 5\text{x}10^{4-2} = 500$, $m = 10^4 = 10000$, calculate $a \approx \sqrt{10000} = 100$ take the value $k = 12$, to obtain $a = 99$ o $a = 101$, then we have two possible values for the multiplicative factor $a_1 = 99$ y $a_2 = 101$who are two possible values for a that are closer to 100. So the equations that generate pseudorandom numbers are : $n_{i+1} \equiv (99n_i) \; (\text{mod } 10000)$, and

$n_{i+1} \equiv (101 n_i) \pmod{10000}$, for $i = 0, 1, 2, \ldots, h\text{-}1$, con $n_0 = 1, 5, \ldots, m\text{-}1$, and $(n_0, 5) = 1$.

2.1.2 Resume

After analyzing the properties and interrelations that can be achieved by the parameters n_0, a, c and m in order to satisfy the maximum period and statistical independence properties of the generated numbers by the equation (1) we summarize the properties of the 4 methods discussed on Table 1.

General congruencies formula

$$n_{i+1} \equiv (an_i + c) \pmod{m}$$

For *i=0,1,2,….,m-1 where* a, n_0, c, and m are positive natural numbers, $c<m$, and n_0 *is* called initial seed. h=Maximum generated period.

Table 1. Characteristics for the a, n_0, c, and m, parameters that optimize multiplicative, mixed, binary and decimal methods.

	Multiplicative method c = 0	Mixed method c ≠ 0
Bin	$h = 2^{b-2}; b > 2$ $m = 2^b$ $a \approx \sqrt{m}$ a non - par $a = 3 + 8k$ o $a = 5 + 8k;$ $k = 0,1,2,3,\ldots$ y $(a, m) = 1$ n_0 non - par positive multiplicative binary congruencies method	$h = m = 2^b; b > 2$ $a \approx 2^{b/2} + 1$ $a = 2^s + 1; s \geq 2$ c positive non - par y n_0 positive natural *if* $b = 3$ *then* $a = 5$ *for* $b \geq 4$ $a = 2^{\left[\frac{b}{2}\right]} + 1; \quad \left[\frac{b}{2}\right] = \begin{cases} b/2 \text{ if } b \text{ is par} \\ \frac{b-1}{2} \text{ if } b \text{ is non - par} \end{cases}$ Mixed binary congruencies method
Dec	$h = 5 \times 10^{d-2}; d > 3$ $m = 10^d$ $a \approx \sqrt{m}$ a non - par $a = 3 + 8k$ o $a = 5 + 8k$ y $(a, 5) = 1$ $a = 200k \pm p$; *with* $k = 0,1,2,3,4,5,\cdots$ y $p = \pm\{3,11,13,19,21,27,29,37,53,59,61,$ $67,69,77,83,91,93,99\} \pmod{200}$ n_0 positive non - par, y $(n_0, 5) = 1$ Multiplicative decimal congruencies method	$h = m = 10^d; d \geq 3$ $a \approx 10^{d/2} + 1; \quad a = 10^s + 1; s > 1$ *Particular ycases* : *if* $d = 3$, *then* $a = 101$ *for* $d \geq 4$ $a = 10^{\left[\frac{d}{2}\right]} + 1$ *where* $\left[\frac{d}{2}\right] = \begin{cases} d/2 \text{ if } b \text{ is par} \\ \frac{d-1}{2} \text{ if } b \text{ is non - par} \end{cases}$ c positive non - par y $(c, 5) = 1$ n_0 positive natural.. Mixed decimal congruencies method

In the following tables, we show the optimal values that needs to be taken by a, n_0, c, and m, in order to achieve the conditions of generation of pseudorandom numbers for each one of previously shown methods in table 1 based on the general congruencies equation(1).

Table 2. Some optimal values for a, n_0, c, and m, for multiplicative binary congruencies method. In every case, $c = 0$, and $n_0 \in \{1,3,5,\cdots m-1\}$.

b	$h = 2^{b-2}$	$m = 2^b$	$a \approx \sqrt{m} = 2^{b/2}$	a_1	a_2
3	2	8	2.828427125	3	3
4	4	16	4	3	5
5	8	32	5.656854249	5	5
6	16	64	8	11	5
7	32	128	11.3137085	11	11
8	64	256	16	19	13
9	128	512	22.627417	21	21
10	256	1024	32	35	29
11	512	2048	45.254834	45	45
12	1024	4096	64	67	61
13	2048	8192	90.50966799	91	91
14	4096	16384	128	131	125
15	8192	32768	181.019336	181	181
16	16384	65536	256	259	253
17	32768	131072	362.038672	363	363
18	65536	262144	512	515	509
19	131072	524288	724.0773439	725	725
20	262144	1048576	1024	1027	1021
21	524288	2097152	1448.154688	1451	1451
22	1048576	4194304	2048	2051	2045
23	2097152	8388608	2896.309376	2899	2899
24	4194304	16777216	4096	4099	4093
25	8388608	33554432	5792.618751	5795	5795

Table 3. Some optimal values for a, n_0, c, and m, for multiplicative decimal congruencies method el in every case, $c = 0$, y $n_0 \in \{1,3,\cdots m-1\}$, and $(n_0, 5) = 1$.

d	$h = 5x10^{d-2}$	$m = 10^d$	$a \approx \sqrt{m} = 2^{d/2}$	a_1 **(optimal)**	a_2 **(optimal)**
4	100	10,000	100	99	101
5	1,000	100,000	316.227766	317	317
6	10,000	1,000,000	1000	1003	997
7	100,000	10,000,000	3162.27766	3163	3163
8	1,000,000	100,000,000	10000	10003	9997
9	10,000,000	1,000,000,000	31622.7766	31621	31621
10	100,000,000	10,000,000,000	100000	100003	99997

Table 4. Some optimal values for a, n_0, c, and m, for mixed binary congruencies method. In every case $c \in \{1,3,5,\cdots m-1\}$, y $n_0 \in \{0,1,2,3,\cdots m-1\}$.

b	$h = m = 2^b$	a
3	8	5
4	16	5
5	32	5
6	64	9
7	128	9
8	256	17
9	512	17
10	1024	33
11	2048	33
12	4096	65
13	8192	65
14	16384	129
15	32768	129
16	65536	257
17	131072	257
18	262144	513
19	524288	513
20	1048576	1025
21	2097152	1025
22	4194304	2049
23	8388608	2049
24	16777216	4097
25	33554432	4097

Table 5. Some optimal values for a, n_0, c, and m, for mixed decimal congruencies method. In every case $[c \in \{1,3,\cdots m-1\}, y\,(c,5)=1]$, y $[n_0 \in \{1,3,\cdots m-1\}, y\,(n_0,5) = 1]$.

d	$h = m = 10^d$	a
1000	101	1000
10000	101	10000
100000	101	100000
1000000	1,001	1000000
10000000	1,001	10000000
100000000	10,001	100000000
1000000000	10,001	100000000
10000000000	100,001	100000000

3 Variable Length Integer Number Chain Generation without Repetitions

Let be N the quantity of numbers without repetitions that is needed, the form to generate this variable length chains of numbers without repetition is described as follows:

1. Select one of the methods shown in Table1.
2. Look on tables 2 and 5 and select a period h of numbers as well as a, n_0, c, and m parameters, with $N \le h$.
3. Generate with the chosen method, the first N pseudorandom integer numbers $\{n_0,n_1,n_2,\ldots,n_{N-1}\}$ using the general congruencies equation (1).
4. Obtain the hierarchy J_i* of each ones of the numbers generated in the second step for $i=0, 1, 2, 3, \ldots, N-1$whit respect of all N pseudorandom generated numbers. By its definition $\{J_0, J_1, J_2, \cdots, J_{N-1}\} \in \{1, 2, 3, \cdots, N\}$ but in a pseudorandom form and $J_0 \neq J_1 \neq J_2 \neq \cdots \neq J_{N-1}$. Moreover because the order of these factors does not modifies its result, then exists an order of hierarchies such as $J_0 = 1, J_1 = 2, J_i = i+1, \cdots, J_{N-1} = N$. In this case we can establish the relation:

$$J_i = i+1, \quad for \quad i = 0, 1, 2, \cdots, N-1. \tag{7}$$

Whit:

$$M = \sum_{i=0}^{N-1} J_i = \sum_{i=1}^{N} i = \frac{N(N+1)}{2}. \tag{8}$$

5. The hierarchies obtained in step 4, are precisely the N numbers without repetition searched.

Thiese hierarchies have interesting properties, We start defining one discrete function $f(J)$ like:

$$f(J) = \frac{1}{N}, \quad with \quad J = J_0, J_1, J_2, \cdots, J_{N-1}. \tag{9}$$

$f(J)$ is a discrete probability distribution, and the expected value of 1 *E(1) is:*

$$E(1) = \sum_{J=J_0}^{J_{N-1}} f(J) = \sum_{i=0}^{N-1} f(J_i) = \sum_{i=0}^{N-1} \left(\frac{1}{N}\right) = \frac{\sum_{i=0}^{N-1}(1)}{N} = \frac{N}{N} = 1. \tag{10}$$

Then the expected value $\mu_J = E(J)$, and the variance $\sigma_J^2 = Var(J) = E[(J-\mu_J)^2]$ of $f(J)$.

$$\mu_J = E(J) = \sum_{J=J_0}^{J_{N-1}} J\, f(J) = \sum_{i=0}^{N-1} J_i f(J_i)$$
$$= \frac{1}{N}\sum_{i=0}^{N-1} J_i = \frac{M}{N} = \frac{\frac{N(N+1)}{2}}{N} = \frac{N+1}{2} \quad , \tag{11}$$

It is easy to see that $\mu_J = \overline{J}$, because

$$\overline{J} = \frac{\sum_{J=J_0}^{J_{N-1}} J}{N} = \frac{\sum_{i=0}^{N-1} J_i}{N} = \frac{M}{N} = \frac{N+1}{2}. \tag{12}$$

Again the order of the factors does not modify the result, then we can use the order given by equation (7) to calculate:

$$E(J^2) = \sum_{J=J_0}^{J_{N-1}} J^2 f(J) = \sum_{i=0}^{N-1} J_i^2 f(J_i) = \frac{1}{N}\sum_{i=0}^{N-1} J_i^2 = \frac{1}{N}\sum_{i=0}^{N-1} (i+1)^2$$
$$= \frac{1}{N}\sum_{i=1}^{N} i^2 = \frac{1}{N}\frac{N(N+1)(2N+1)}{6} = \frac{(N+1)(2N+1)}{6} \tag{13}$$

So $\sigma_J^2 = Var(J) = E[(J-\mu_J)^2] = E(J^2) - \mu_J^2$, Then from equations (11) y (13), we follows that:

$$\sigma_J^2 = Var(J) = E(J^2) - \mu_J^2 = \frac{(N+1)(2N+1)}{6} - \left(\frac{N+1}{2}\right)^2$$
$$= \frac{(N+1)[2(2N+1) - 3(N+1)]}{12} = \frac{(N+1)(N-1)}{12} \tag{14}$$

As we can see, they were precisely the same values of a uniform discrete distribution.

4 Variable Length Number Chains without Repetitions in the Interval (0,1), (0,1] and [0,1]

Method 1. Calculate:

$$x_i = \frac{J_i}{N}, \text{ with } i = 0, 1, 2, 3, \ldots, N\text{-}1, \tag{15}$$

Where J_i are hierarchies , and $x_i \in (0,1]$. Now, using the results from equation (8) we have:

$$\sum_{i=0}^{N-1} x_i = \sum_{i=0}^{N-1} \frac{J_i}{N} = \frac{\sum_{i=0}^{N-1} J_i}{N} = \frac{M}{N} = \frac{\left(\frac{N(N+1)}{2}\right)}{N} = \frac{N+1}{2}, \tag{16}$$

Using equations (11) , we have:

$$\mu_1 = E(x) = E\left(\frac{J}{N}\right) = \frac{1}{N}E(J) = \frac{1}{N}\left(\frac{N+1}{2}\right) = \frac{1}{2}\left(1+\frac{1}{N}\right), \tag{17}$$

And it is easy to prove that $\bar{x} = \mu_x$, because

$$\bar{x} = \frac{1}{N}\sum_{i=0}^{N-1} x_i = \frac{1}{N}\left(\frac{N+1}{2}\right) = \frac{1}{2}\left(1+\frac{1}{N}\right), \tag{18}$$

For big *N*, we have:

$$\mu_x = \bar{x} \to \frac{1}{2} + o\left(\frac{1}{N}\right) = \frac{1}{2}. \tag{19}$$

Simply we use the equation $Var(ay+b) = a^2 Var(y)$ and the result of the equation (14) is:

$$\begin{aligned} \sigma_1^2 &= Var(x) = Var\left(\frac{J}{N}\right) = \frac{1}{N^2}Var(J) = \frac{1}{N^2}\frac{(N+1)(N-1)}{12} \\ &= \frac{(N^2-1)}{12N^2} = \frac{1}{12}\left(1-\frac{1}{N^2}\right) \end{aligned}. \tag{20}$$

For big *N*, we have:

$$\sigma_1^2 \to \frac{1}{12} - o\left(\frac{1}{N^2}\right) = \frac{1}{12}. \tag{21}$$

Method 2. Calculate**:**

$$x_i = \frac{J_i}{N+1}, \text{ for } i = 0, 1, 2, 3, \ldots, N\text{-}1, \tag{22}$$

Where J_i are the hierarchies and $x_i \in (0,1)$. now, Using the results from equations (8), we have:

$$\sum_{i=0}^{N-1} x_i = \sum_{i=0}^{N-1}\left(\frac{J_i}{N+1}\right) = \frac{\sum_{i=0}^{N-1} J_i}{N+1} = \frac{M}{N+1} = \frac{\left(\frac{N(N+1)}{2}\right)}{N+1} = \frac{N}{2}, \tag{23}$$

using results from (11), we have:

$$\mu_2 = E(x) = E\left(\frac{J}{N+1}\right) = \frac{1}{(N+1)}E(J) = \frac{1}{(N+1)}\left(\frac{N+1}{2}\right) = \frac{1}{2}, \tag{24}$$

It is easy to prove that: $\bar{x} = \mu_x$, because

$$\bar{x} = \frac{1}{(N+1)}\sum_{i=0}^{N-1} x_i = \frac{1}{N}\left(\frac{N}{2}\right) = \frac{1}{2}, \tag{25}$$

For all $N > 1$, we have that:

$$\mu_x = \bar{x} \to \frac{1}{2} + o\left(\frac{1}{N}\right) = \frac{1}{2}. \tag{26}$$

Similar if we use $Var(ay+b) = a^2 Var(y)$, end the result from (14), we have that:

$$\begin{aligned} \sigma_2^2 &= Var(x) = Var\left(\frac{J}{N+1}\right) = \frac{1}{(N+1)^2}Var(J) \\ &= \frac{1}{(N+1)^2}\frac{(N+1)(N-1)}{12} \\ &= \frac{(N-1)}{12(N+1)} = \frac{(N+1-2)}{12(N+1)} = \frac{1}{12}\left(1 - \frac{2}{N+1}\right) \end{aligned} \tag{27}$$

For big N we have that:

$$\sigma_1^2 \to \frac{1}{12} - o\left(\frac{1}{N^2}\right) = \frac{1}{12}. \tag{28}$$

Method 3. Calculate:

$$x_i = \frac{J_i}{M}, \text{ for } i = 0, 1, 2, 3,\ldots, N\text{-}1, \tag{29}$$

Where J_i are the hierarchies, $M = \sum_{i=0}^{N-1} J_i$, y $x_i \in (0,1)$. Now, using the result from (8), we have that:

$$\sum_{i=0}^{N-1} x_i = \sum_{i=0}^{N-1} \frac{J_i}{M} = \frac{\sum_{i=0}^{N-1} J_i}{M} = \frac{M}{M} = 1, \tag{30}$$

Using the result from (11), we obtain:

$$\mu_3 = E(x) = E\left(\frac{J}{M}\right) = \frac{1}{M} E(J) = \frac{1}{\left(\frac{N(N+1)}{2}\right)}\left(\frac{N+1}{2}\right) = \frac{1}{N}. \tag{31}$$

We can prove easily that: $\bar{x} = \mu_x$, because

$$\bar{x} = \frac{1}{N}\sum_{i=0}^{N-1} x_i = \frac{1}{N}(1) = \frac{1}{N}. \tag{32}$$

For big N, we have that:

$$\mu_3 = \bar{x} \to o\left(\frac{1}{N}\right) = 0. \tag{33}$$

Using $Var(ay+b) = a^2 Var(y)$, and equation (14), we have that:

$$\begin{aligned} \sigma_3^2 &= Var\left(\frac{J}{M}\right) = \frac{1}{M^2} Var(J) \\ &= \frac{1}{\left(\frac{N(N+1)}{2}\right)^2}\left[\frac{(N+1)(N-1)}{12}\right] = \frac{(N-1)}{3N^2(N+1)}. \end{aligned} \tag{34}$$

For big N ,we have that:

$$\sigma_3^2 = \frac{(N-1)}{3N^2(N+1)} < \frac{1}{3N^2}\left(\frac{N}{N}\right) = \frac{1}{3N^2} \to o\left(\frac{1}{N^2}\right) = 0. \tag{35}$$

Method 4. Calculate:

$$x_i = \frac{J_i - J_{mín}}{J_{máx} - J_{mín}}, \text{ for } i = 0, 1, 2, 3, \ldots, N\text{-}1, \tag{36}$$

Since the hierarchy structure we have that:

$$\begin{aligned} J_{mín} &= 1, \quad and \\ J_{máx} &= N \end{aligned} \tag{37}$$

where J_i are hierarchies y $x_i \in [0, 1]$. Now using results from (8), we have that:

$$\begin{aligned} \sum_{i=0}^{N-1} x_i &= \frac{1}{(N-1)} \sum_{i=0}^{N-1} (J_i - 1) = \frac{1}{(N-1)} (M - N) \\ &= \frac{1}{(N-1)} \left(\frac{N(N+1)}{2} - N \right) = \frac{N(N+1-2)}{2(N-1)} = \frac{N(N-1)}{2(N-1)} = \frac{N}{2} \end{aligned} . \tag{38}$$

Using the results from equation (11), we have that:

$$\begin{aligned} \mu_4 = E(x) &= E\left(\frac{J-1}{N-1} \right) = \frac{1}{(N-1)} E(J-1) = \frac{1}{(N-1)} \left[E(J) - E(1) \right] \\ &= \frac{1}{(N-1)} \left(\frac{N+1}{2} - 1 \right) = \frac{1}{(N-1)} \left(\frac{N+1-2}{2} \right) = \frac{(N-1)}{2(N-1)} = \frac{1}{2} \end{aligned} . \tag{39}$$

Is easy to prove that: $\bar{x} = \mu_x$, because

$$\begin{aligned} \bar{x} &= \frac{1}{N} \sum_{i=0}^{N-1} x_i = \frac{1}{N} \sum_{i=0}^{N-1} \left(\frac{J_i - 1}{N-1} \right) = \frac{1}{N(N-1)} \left(\sum_{i=0}^{N-1} J_i - \sum_{i=0}^{N-1} (1) \right) = \\ &= \frac{1}{N(N-1)} \left(\frac{N(N+1)}{2} - N \right) = \frac{N(N+1-2)}{2N(N-1)} = \frac{N(N-1)}{2N(N-1)} = \frac{1}{2} \end{aligned} . \tag{40}$$

for $N > 1$, we have that:

$$\mu_4 = \bar{x} = \frac{1}{2} . \tag{41}$$

Using $Var(ay+b)=a^2Var(y)$, and the result from (14), we have that:

$$\begin{aligned}\sigma_4^2 &= Var\left[\frac{1}{(N-1)}J-\frac{1}{(N-1)}\right]=\frac{1}{(N-1)^2}Var(J)\\ &=\frac{1}{(N-1)^2}\frac{(N+1)(N-1)}{12}\\ &=\frac{N+1}{12(N-1)}=\frac{1}{12}\left(1+\frac{2}{N-1}\right)\end{aligned} \quad . \tag{42}$$

For big *N* we have that:

$$\sigma_4^2=\frac{1}{12}\left(1+\frac{2}{N-1}\right)\to\frac{1}{12}+o\left(\frac{1}{N}\right)=\frac{1}{12}. \tag{43}$$

If we analyze the behaviors of μ y σ^2 for the 4 methods to obtain values x_i a from hierarchies J_i, we can determinate:

a) The worst statistical behavior, belongs to method 3, equation (22), because for big *N* $\mu_1,\sigma_1^2\to 0$, this destroy the pseudorandom behavior for x_i. values. The only advantage is that x_i, sum values is always 1. For not so big *N* values the discrete uniform distribution behavior is preserved.

b) Methods 1, 2 and 4, equations (15), (22), and (36) do not have any substantial differences , because for each one $\mu\to\frac{1}{2}$, y $\sigma^2\to\frac{1}{12}$, these values are precisely the same for uniform continuous distribution.

Example of variable length integer number chains generation without repetitions: suppose that we need to generate a certain amount of numbers $N = 20$.

1. First we select one of the congruencies methods from table 1, for this example we take the multiplicative binary congruencies method.
2. Because *N*= 20, we take $h \geq 20$,for our example $h = 2^6 = 64$. Looking into table 2, the parameters values for equation 1, are: $m = 2^8 = 256$, a_1 =13 o a_2 =19, $c = 0$, y $n_0\in\{1,3,5,\cdots m-1\}$.
3. From equation .(1), and and the parameters values *a*, n_0, *c*, and *m*, obtained by the second step, we have 2 possible equations to generate pseudorandom numbers, these are:

$$n_{1,i+1}\equiv 13n_{i,1}(\text{mod } 256), \text{ and} \tag{44}$$

$$n_{2,i+1} \equiv 19 n_{2,i} \,(\text{mod } 256) \tag{45}$$

Where $i = 0, 1,2,\ldots, N\text{-}1$.

4. Obtain hierarchies from n_i, numbers generated by equations (44) y (45).
5. Using one of the methods translate this hierarchies values to intervals (0,1) or (0, 1], or(0, 1) or [0, 1], from the equations (15), (22), (29), y (36).

Acknowledgments

The authors thank the participation of Victor Zamudio from I.T.L for the revision and correction of this article.

References

1. Jerry, B., Carson II, J.S., Nelson, B.L.: Dicrete-Event Simulation, 2nd edn. Prentice-Hall, Upper Saddle River (1996)
2. Boni, C.P.: Simulation of Information and Decision Systems in the Firm. Prentice-Hall, Englewood Cliffs (1963)
3. Coveyou, R.R.: Serial Correlation in the Genaration of Pseudo-Random Numbers. Journal of the Association for Computing Machinery VII, 72 (1960)
4. Dimacs, Discrete Mathematics and Theoretical Computer Science (1999), `http://dimacs.rutgers.edu/Volumes/Vol35.html`
5. Greenberger, M.: And a Priori Determination of Serial Correlation in Computer Genarated Random Números. Mathematics of Computation XV, 383–389 (1961)
6. Hull, T.E., Dobell, A.R.: Random Numbers Generation. SIAM Review IV(3), 230–254 (1962)
7. (IBMC) International Business Machines Corporation, Random Number Generation and Testing, Reference Manual (C20-8011), Nueva York (1959)
8. L'Ecuyer, P.: Random Number Generation. In: Henderson, S.G., Nelson, B.L. (eds.) Elsevier Handbooks in Operations Research and Management Science: Simulation, ch. 3, pp. 55–81. Elsevier Science, Amsterdam (2006)
9. Law, A.M., Kelton, W.D.: Simulation modeling and analysis, 2nd edn. McGraw-Hill, New York (1991)
10. Naylor, B., Burdick, Chu, K.: Tecnicas De Simulación En Computadoras, Limusa, México, Cáp. 3 (1977)
11. Pooch Udo, W., Wall, J.A.: Discrete Event Simulation, A Practical Approach. CRC Press, Inc., Boca Raton (1993)
12. Ross Sheldom, M.: A Course in Simulation. Macmillan, New York (1990)
13. Ross Sheldom, M.: Simulacion. Prentice Hall, México (1997)
14. Ross Sheldom, M.: Simulación. Prentice Hall, México (1999)
15. Schmith, J.W., Taylor, E.: Análisis y simulación de sistemas industriales, Trillas, México (1979)

Comparative Analysis of Hybrid Techniques for an Ant Colony System Algorithm Applied to Solve a Real-World Transportation Problem

Juan Javier González-Barbosa, José Francisco Delgado-Orta, Laura Cruz-Reyes, Héctor Joaquín Fraire-Huacuja, and Apolinar Ramirez-Saldivar

Instituto Tecnológico Ciudad Madero. 1o. de mayo s/n, 89440. Col. Los Mangos, Cd. Madero, México
{francisco.delgado.orta, jjgonzalezbarbosa}@gmail.com,
{lcruzreyes, hfraire@}prodigy.net.mx,
apolinar.ramirez.s@gmail.com

Abstract. This work presents a comparison of hybrid techniques used to improve the Ant Colony System algorithm (ACS), which is applied to solve the well-known Vehicle Routing Problem (VRP). The Ant Colony System algorithm uses several techniques to get feasible solutions as learning, clustering and search strategies. They were tested with the dataset of Solomon to prove the performance of the Ant Colony System, solving the Vehicle Routing Problem with Time Windows and reaching an efficiency of 97% in traveled distance and 92% in used vehicles. It is presented a new focus to improve the performance of the basic ACS: learning for levels, which permits the improvement of the application of ACS solving a Routing-Scheduling-Loading Problem (RoSLoP) in a company case study. ACS was applied to optimize the delivery process of bottled products, which production and sale is the main activity of the company. RoSLoP was formulated through the well-known Vehicle Routing Problem (VRP) as a rich VRP variant, which uses a reduction method for the solution space to obtain the optimal solution. It permits the use in efficient way of computational resources, which, applied in heuristic algorithms reach an efficiency of 100% in the measurement of traveled distance and 83% in vehicles used solving real-world instances with learning for levels. This demonstrates the advantages of heuristic methods and intelligent techniques for solving optimization problems.

1 Introduction

Planning and scheduling are related to many producer activities for several companies. Sometimes they are associated with the transportation of goods, the main activity for many companies. A survey of transportation applications [34] establishes that delivery of goods using the minimum quantity of resources reduces operation costs, which increases the utilities for the companies from 5 to 20 percent.

Transportation problems, based on study cases, have been an object of study for many researchers who have approached their solution in two main areas: 1) the

P. Melin et al. (Eds.): Soft Comp. for Recogn. Based on Biometrics, SCI 312, pp. 365–385.
springerlink.com

development of rich formulations of solution and 2) the construction of efficient algorithms to solve them. Theoretical applications of transportation have been developed to prove the performance of the solution algorithms, as the dataset of Solomon, a set of instances developed to measure the implementations for the Vehicle Routing Problem with Time Windows, a well-known NP-Hard problem related to the transportation of goods. RoSLoP formulation, defined in [10], presents a heuristic approach as an alternative for solving 11 VRP variants in an integrated rich VRP problem. This formulation requires a minimum of 2^9 integer variables to solve 30 restrictions to solve the basic formulation. However, due as a necessity of metrics of performance for the developed algorithms, it was developed a new formulation in [13]. It is based on a linear transformation function and its formulation contains 2^2 integer variables and 15 restrictions to solve 12 VRP variants. This formulation allowed the computation of an optimal solution for RoSLoP. This work, focuses on the heuristic solution presented in [10], based on an ant colony system algorithm and the addition of techniques used to improve the quality of the obtained solutions for VRPTW and its application solving RoSLoP given the computed optimal solution. VRP variants are defined in section 2. Ant algorithms and ACS are described in section 3. Section 4 presents the RoSLoP definition and the reduction method, needed to obtain the optimal solution. Sections 5 and 6 present the heuristic and exact solution strategies. Section 7 show the experimentation and results solving Solomon's and RoSLoP instances and Section 8 presents the conclusions and future applications for this work.

2 Vehicle Routing Problem (VRP) Related Works

VRP, defined by Dantzig in [12], is a classic problem of combinatorial optimization. It consists in one or various depots, a fleet of m available vehicles and a set of n customers to be visited, joined through a graph $G(V,E)$, where $V=\{v_0, v_1, v_2, \ldots,v_n\}$ is the set of vertex v_i, such that v_0 the depot and the rest of the vertex represent the customers. Each customer has a demand q_i of products to be satisfied by the depot. $E=\{(v_i, v_j) \mid v_i,v_j \in V, i \neq j\}$ is the set of edges. Each edge has an associated value c_{ij} that represents the transportation cost from v_i to v_j. The VRP problem consists of obtaining a set R of routes with a total minimum cost such that: each route starts and ends at the depot, each vertex is visited only once by either route, or the length of each route must be less than or equal to L.

2.1 VRP Variants

The most known variants of VRP add several constraints to the basic VRP such as capacity of the vehicles (Capacitated VRP, CVRP) [29], independent service schedules at the customers facilities (VRP with Time Windows, VRPTW-VRPMTW) [8], multiple depots to satisfy the demands (Multiple Depot VRP, MDVRP) [24], customers to be satisfied by different vehicles (Split Delivery VRP, SDVRP) [1], a set of available vehicles to satisfy the orders (site dependent VRP, sdVRP) [33], customers that can ask and return goods to the depot (VRP with Pick up and Delivery, VRPPD) [16], dynamic facilities (Dynamic VRP,

DVRP) [2], line-haul and back-haul orders (VRP with Backhauls, VRPB) [22], stochastic demands and schedules (Stochastic VRP, SVRP) [1], multiple use of the vehicles (Multiple use of vehicles VRP, MVRP) [17], a heterogeneous fleet to deliver the orders (Heterogeneous VRP, HVRP) [31], orders to be satisfied in several days (Periodic VRP, PVRP) [8], constrained capacities of the customers for docking and loading the vehicles (CCVRP) [10], transit restrictions on the roads (road dependant VRP, rdVRP) [10] and depots that can ask for goods to another depots (Depot Demand VRP, DDVRP) [10].

A rich VRP variant, based on the Dantzig's formulation, is defined in [11] as an *Extended Vehicle Routing Problem*, which is applied to real transportation problems. However, the extended VRP requires the addition of some restrictions that represent transportation situations, this increase its complexity making more difficult the computation of an optimum solution through exact algorithms.

Recent works have approached the solution of rich VRP problems like the DOMinant Project [20], which solves five variants of VRP in a transportation problem of goods among industrial facilities located in Norway. Goel [19] solves four VRP variants in a problem of sending packages for several companies. Pisinger [25] and Cano [5] solve transportation problems with five VRP variants. RoSLoP was formulated initially in [28] with six VRP variants. However, due new requirements of the company it was necessary to formulate a VRP with 11 variants in [21]. This new formulation allows the solution of instances of 12 VRP variants. Table 1 details the variants solved by various authors.

Table 1. Related works about known rich VRP variants

Solved variants / Author	CVRP	VRPTW	VRPMTW	OVRP	MDVRP	SDVRP	sdVRP	VRPM	HVRP	CCVRP	DDVRP	rdVRP
Hasle [20]	✓	✓	✓					✓	✓			
Goel [19]	✓	✓						✓	✓			
Pisinger [25]	✓	✓		✓	✓		✓					
Cano [5]	✓	✓				✓		✓	✓			
Rangel [28]	✓	✓			✓	✓		✓	✓			
Herrera [21]	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓

3 The Ant Colony System Algorithm (ACS)

ACS is a bioinspired algorithm based on the behavior of ants which searches the best routes toward their anthill to carry food. ACS uses two main characteristics: the heuristic information η_{rs}, used to measure the predilection to travel between a pair of vertex (*r*,*s*); and the trails of artificial pheromone τ_{rs}. They are used to compute the learned reference of traveling in a determined arc (*r*,*s*). The

heuristic information is used to determine the next node to visit, so is computed with Equation 1.

$$\eta_{rs} = \left(\Delta t_{rs} \cdot (ws_s + st_s) \cdot tc_{rs}\right)^{-1} \tag{1}$$

Where Δt_{rs} is the difference between the current time and the arrival time to node *s*, ws_s represents the remaining size of the time window in *s*, st_s is the time of service in *s* and tc_{rs} is the travel cost from node *r* to node *s*. This calculation gives preference to those customers where: *a*) the needed time to arrive to the facilities starting from the actual position is the smallest, *b*) the time of attention since the current hour plus the time of service is the smallest, and *c*) the traveling time is minimum. However, due to the existence of the variant HVRP, the next more appropriate vehicle must be chosen, for this reason Equation 2 defines the computation of the heuristic information to select the vehicles.

$$\eta_v = \left(nv_v \cdot (\overline{TM_v} + \overline{TR_v}) \cdot \frac{tr_v}{tt_v} \cdot idpref_v \right)^{-1} \tag{2}$$

Where η_v is the value of the heuristic information for the mobile unit *v*, nv_v is a bound of the quantity of travels required for the vehicle *v* to satisfy all the demands of $N_k(r)$, $\overline{TM_v}$ is the average of the service time in $N_k(r)$, $\overline{TR_v}$ is the time trip average of the vehicle to $N_k(r)$, tr_v is the available time for the vehicle *v*, tm_v is the time of attention for the vehicle *v*; tr_v / tm_v is a factor of use/availability. Equation 2 forces the selection of those vehicles whose times of trip, times of service, remaining period of service and preference level are the smallest. The election of a node is done through a pseudo-randomly rule. It establishes that given an ant *k*, located in a node *r* with q_0 a balancing value between the constructive focuses and *q* a random value, these parameters are used to choose the next node to be visited through the Equation 3.

$$p_{rs}^{x} = \begin{cases} \dfrac{\tau_{rs}\eta_{rs}^{\beta}}{\sum_{s \in \mathrm{N}_k(r)} \tau_{rs}\eta_{rs}^{\beta}} & \text{if } s \in \mathrm{N}_k(r) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Where β represents the relative importance of the pheromone versus the distance. If $q < q_0$ then, exploitation of the learned knowledge is done, applying a nearest neighborhood heuristic. Otherwise, a controlled exploration is applied to the ant with the best solution using Equation 3. ACS uses the evaporation of pheromone trails to reduce the solution space through global and local updates. Both update strategies use an evaporation rate ρ. Local update is done to generate different solutions the already obtained. Equations 4 and 5 represent the local and global update respectively. The global update process and the pseudo-randomly rule directs toward a more direct search.

$$\tau_{rs} \leftarrow (1-\rho)\tau_{rs} + \rho\Delta\tau_{rs} \tag{4}$$

$$\tau_{rs} \leftarrow (1-\rho)\tau_{rs} + \rho\tau_0 \tag{5}$$

The global update $\Delta\tau_{rs}$ is computed like the inverse of the length of the shortest global solution generated by the ants; the trail of pheromone τ_0 used in the local update, is the inverse of the product of the length of the shortest global solution generated and the number of visited nodes, establishing dependence with the size

ACS_Algorithm(β , ρ , q_0)

1. $\psi^{gb} \leftarrow InitialSolution$
2. $t \leftarrow \#active\_vehicles(\psi^{gb}) - 1$
3. repeat
4. repeat
5. $\psi^{lb} \leftarrow InitialSolution$
6. For each ant $k \in K$
7. $\psi^k \leftarrow$ new_ant_solution(k, t, IN)
8. $\forall j \notin \psi^k : IN_j \leftarrow IN_j + 1$
9. If $visitedClients(\psi^k) > visitedClients(\psi^{lb})$
10. $\psi^{lb} \leftarrow \psi^k$
11. $\forall j : IN_j \leftarrow 0$
12. End repeat
13. End If
14. Execute *Local_update* (τ_0 , τ_{rs})
15. End For
16. Execute *Global_update* (Δ_{rs} , τ_{rs})
17. Until stop criteria are reached
18. If ψ^{lb} is feasible and better than ψ^{gb} then
19. $\psi^{gb} \leftarrow \psi^{lb}$
20. $t \leftarrow \#active\_vehicles(\psi^{gb}) - 1$
21. End If
22. Until stop criteria are reached

Fig. 1. The ACS algorithm

of the instance. In each iteration, ants examine in the neighborhood of the best ant while local update changes the desirable use of the roads. Fig. 1 shows the scheme of the basic ACS, which is regulated by three parameters: ρ, β and q_0. An initial feasible solution is created using a Nearest Neighborhood algorithm (Line 1).

When an ant colony is created (Lines 4-18), an improved global solution is searched through the construction of a fixed number of ants per colony and with local and global updates of the pheromone trails for vehicles and customers (Lines 13 and 15). If the local solution has been improved (Line 9), the current colony ends (Line 12), the local solution is compared with regard to the best global solution (Line 18) and the updates are done (Lines 19-20).

The stop criteria (Lines 17 and 22) are specified with respect to the number of iterations and the execution time. ACS uses the methods *#active_vehicles* and *#visited_customers* to determine the number of vehicles used and the number of customers visited in the solution.

The function *#depot* is used to get the owner of a specific vehicle to a depot. The integer vector IN_r stores the number of times a customer *r* has not been inserted in a solution. *IN* is used by the construction procedure to favor the customers that are less frequently included in a solution. The constructive procedure *new_ant_solution* builds the routes of the new solution with the ant *k*, using *t* vehicles to satisfy customer demands. The basic ACS is improved through some techniques as restricted lists, learning and complexity reduction techniques.

3.1 State of Art

ACO was introduced initially in [15] through the creation of Ant System (AS), which was formed by three algorithms: Ant-density, Ant-quantity and Ant-Cycle. Ant-density and Ant-quantity uses the update of pheromone trails in every step of the ants, while Ant-Cycle makes updates after a complete cycle of the ant. These algorithms were tested with the Traveling Salesman Problem (TSP), with good results and the assurance that this method can be applied in several types of problems. A study of the correct configuration of AS for solving TSP was done in [7], which concludes that the main parameter is β and establishes that the optimal number of ants is equivalent to the number of nodes of the problem. The properties of the Ant-cycle algorithm was done in [6] based on uniform and randomly distributions of the ants at the nodes whole results shown a few differences; however, randomly distribution obtains the best results.

A learning technique, based on *Q*-learning applied to AS was applied in [18], it was named Ant-Q. It was applied for solving the TSP and Asymmetric TSP (ATSP) through the use of a table of values *Q*, equivalent to the table of values of *Q*-learning, which is used to indicate how good a determined movement towards a node *s* is since a node *r*. It applies a rule to choose the next node to be visited and reinforcement learning applied to the *Q* values using the best tour of the ants.

An improved AS named Ant Colony System (ACS) [14] presents three main differences with regard AS: *a*) the transition-state rule is modified to establish a balance between the exploration of new arcs and a apriority exploitation of the problem; *b*) the global updating rule is applied only to the arcs of the tour of the

best ant; *c*) a local updating of the pheromone is applied while ants build a solution. ACS was applied to TSP and ATSP with the addition of a local search based on a 3-opt scheme. A new ant algorithm named Max-Min Ant System (MMAS) was proposed in [30], establishing three differences with regard to AS: *a*) the updating rule was modified to choose the best tour of the ants and the best found solution during the execution of the algorithm, increasing with this the exploration; *b*) a upper limit for the pheromone trails was established, which permits that a subset of nodes not being chosen in recurrent form; and *c*) the pheromone trails were initialized with the permitted upper bound to choose only the best arcs, increasing their pheromone trails and ants rarely choose bad arcs to build a solution.

Other variant of AS, named ASrank, was developed in [4]. It consist of a sorting of the ants (once that ants has built their solution) following the length of travel and to weight the contribution of the ant to the trail-pheromone updating level according a rate μ; it permits that only the best ω ants are considered, omitting the pheromone trails of some ants that uses sub-optimal roads. This algorithm was tested with TSP instances. This works focuses on some techniques applied to the Ant Colony System (ACS) for solving the vehicle routing problem; a generalized TSP with transportation constrains that defines the Vehicle Routing Problem with Time Windows (VRPTW), a known NP-hard problem object of study for many researchers. The techniques used to improve the performance for the basic ACS and an application for ACS in a real environment is presented.

The clustering technique (CT). A Clustering Technique (CT), proposed in [14], is elaborated through feasibility conditions and location of the customers in the graph. These techniques are applied to limit the global population into subsets with certain conditions. The developed CT is created in five steps:

Step 1. A Minimum Spanning Tree (MST) is generated including all the customers and the depot of the instance.

Step 2. The mean μ and standard deviation σ are obtained, the minimum and maximum costs associated to the roads included in the MST.

Step 3. The percentage of visibility θ of the associated costs to each road belonging to the MST is computed through the Equation 6.

$$\theta = \sigma \Big/ 2\left(\arg\max\left\{tc_{rs}\right\} - \arg\min\left\{tc_{rs}\right\}\right) \qquad (r,s) \in MST \tag{6}$$

If $\theta < 0.1$, the location of the customers in the instance follows an uniform distribution. Otherwise, it is possible the existence of regions in the space with more density.

Step 4. The clustering rule to define the regions is applied: if $\theta \geq 0.1$, the conglomerates are formed following a hierarchical clustering. Otherwise, all the customers form a single conglomerate. The threshold of acceptance ω of the Equation 7 is an autoadaptative characteristic of the method.

$$\omega = 2 \cdot \arg\max_{arc_{rs} \in MST} \left\{tc_{rs}\right\} \tag{7}$$

Step 5. When the ownership of each customer to a conglomerate is defined, heuristic information is modified with the ownership rule for the customers c_r and c_s, which belong to different groups h_i and h_j.

$$\textbf{If } h_i \neq h_j \mid c_r \in h_i \wedge c_s \in h_j \textbf{ then } \eta_{rs} = \eta_{rs} \cdot \frac{|H|}{|C|} \quad c_r, c_s \in C ; h_i, h_j \in H \tag{8}$$

Equation 8 permits the inhibition in proportional form to prefer a determined customer with regard to others; it is based in the ownership to the different conglomerates and the number of these. Search Techniques (ST). The developed ACS, uses an initial search based on the nearest neighborhood technique of the Equation 9, which gets a deterministic solution of reference that is improved through local search techniques.

$$\eta_{rs} = \left(\Delta t_{rs} \cdot ws_s + st_s\right)^{-1} \tag{9}$$

The local search includes schemes of exchange of axes as 3-opt [3], which includes the 2-opt[32] and Cross-Exchange [9], which operate on two routes, it includes other simple operators, which permits the use of empty segments using movements type 2-opt* [26], Relocation, and Exchange [27]. These schemes are applied to cover the solution space widely.

Distributed Q-Learning (DQL). A new ant algorithm named DQL is presented in [23]. It is similar to the Ant-Q algorithm [14], but establishes three main differences: a) DQL does not use heuristics dependent of domain, reason why it does not require additional parameters, b) DQL updates the Q values only once with the best solution obtained for all the agents and c) DQL permits more exploration and a best exploitation. DQL establishes that all the agents have an access to a temporal copy Qc of the evaluation functions (Q values) for each pair state-action. Each

```
DQL ( )
Initialize Q(s,a)
Repeat (for n episodes)
   Initialize s, Qc(s,a)← Q(s,a)
   Repeat (For each step of the episode)
      Repeat (for m agents)
         Execute action a, observe r, s'
         Qc(s,a)←Qc(s,a)+ α[γmax_a' Qc(s',a')-Qc(s,a)]
         s ← s';
     Until s is terminal
   Evaluate the m proposed solutions
  Assign a reward to the best found solution
    and update the Q-values:
     Q(s,a) ← Q(s,a)+ α[r + γ max_a' Q(s',a')-Q(s,a)]
```

Fig. 2. The DQL Algorithm

time that an agent has to choose an action, it observes the copy of the Q values to execute the next action, once that the next action is chosen the table Qc is updated. This process is similar to Ant-Q; however, all the agents of DQL update and share their common values Qc. When all the agents have found a solution, this copy is deleted and the original Q values are rewarded, using the best obtained solution by the agents. The DQL algorithm is shown in Fig. 2.

The *Qc* values are used as a guide, which permits that the agents can observe more promissory states. So, DQL permits a bigger exploration and a better exploitation since only the best actions are rewarded. DQL was compared in [23] with Ant-*Q* and *Q*-learning solving TSP instances and it obtains a faster convergence with regard the other algorithms without needing the configuration of extra parameters.

Learning Levels (LL). The main similarity of DQL with ACS is the fact that both use a table of values in which the learned experience is storaged: in DQL is the table of *Q*-values while in ACS is the pheromone table τ_{rs}. This relation permits the implementation of DQL over the ACS algorithm and it permits the development of a new technique, named learning for levels. The scheme of ACS with learning for levels is shown in Fig. 3.

```
ACS_LL( )
Initialize Data Structures
Do
   For each ant: start a solution
    Tau_rs_copy ← Tau_rs
     Do
            For each ant
              Apply the pseudo-randomly rule to
                build a solution
              local pheromone update (Tau_rs_copy)
     until ∀ ants have complete their solution
    Global Pheromone Update (Tau_rs)
Until Stop criteria is reached
```

Fig. 3. ACS with Learning for Levels

Learning levels defines two levels of knowledge: the first level is equal to the values of the original pheromone table, which only contains the information of the best obtained solution for the ants and it is modified only in the global updating process; while the second level equals to the copy of the pheromone table, which contains the local values of the pheromone and it is used for the ants as a guide in the search of better solutions. This level is updated locally for each ant in the local update process.

4 RoSLoP: A Real-World Industrial Application

RoSLoP, immersed in the logistics activity of the company study case, involves three main tasks: routing, scheduling and loading. The mathematical model of

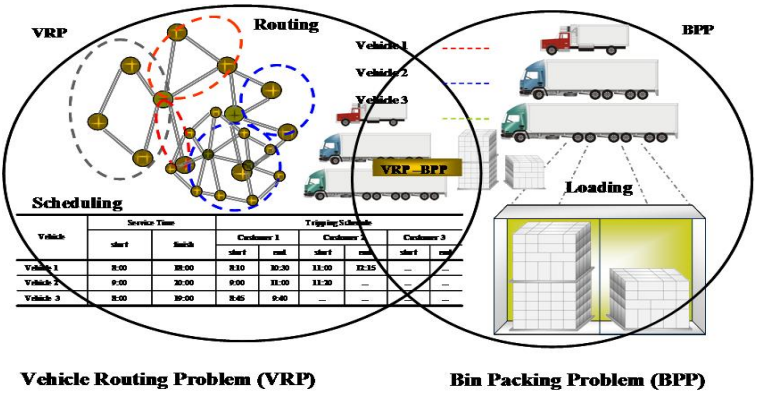


Fig. 4. Definition of RoSLoP

RoSLoP was formulated with two classical problems: routing and scheduling through VRP and the loading through the Bin Packing Problem (BPP). Fig. 4 shows RoSLoP and its relation with VRP-BPP.

4.1 Routing-Scheduling Constrains

The Routing-Scheduling problem is formed by the next elements:

- A set of facilities with finite capacity for the attention of the vehicles, formed by customers *C* and depots *D* with independent service schedules at a facility $j\left[st_j, et_j\right]$, where st_j and et_j represent the time when a facility *j* starts and ends its operation. The travel time between a pair of facilities *i* and *j* is represented by t_{ij}. An integer variable x_{ijk} is used to assign an arc (*i*,*j*) to a route *k*. Depots have the possibility of request goods to other depots. A set of routes K_d that starts and ends at the depot must be formed. This description is related to VRPTW, VRPMTW, CCVRP, SDVRP, MDVRP and DDVRP.
- A fleet of vehicles with heterogeneous capacity V_d, with a service time $stime_v$ and a time for attention tm_{vj}, which depends on the capacity of the vehicle C_{vj} that visit a customer *j* and the available people for docking and loading the containers $Pallets_v$ of the vehicle *v* where the goods are transported (HVRP, CVRP, MVRP, sdVRP). An integer variable y_{vk} is used to assign a vehicle *v* to a route *k*.

4.2 Loading Constrains

The Loading problem consists of the next elements:

- A set of orders Q to be satisfied at the facilities of the customers, formed by units of products per each customer $I_j \in Q$. Each box (product package) has different attributes such as weight (*w*), high (*h*), product type (*pt*), supported weight (*sw*), packing type (*pkt*) and beverage type (*bt*).
- The load must have different accommodation: This process is done in two stages of balancing: *a*) with the accommodation of the most weighted and *b*) the order of customers to be visited. These processes are solved through the DiPro algorithm [11], which places the load into the vehicles according to different restrictions of balancing of load.
- A set of roads represented by the edges of the graph. Each road has an assigned cost c_{ij}, each one with a threshold of allowed weight $MAXLoad_{vj}$ for a determined vehicle *v* that travels towards a facility *j*, and a travel time t_{ij} from facility *i* to *j*. (rdVRP)

The objective of RoSLoP is to get a configuration that allows the satisfaction of the set of *ORDERS* at the set of the customer facilities, minimizing the number of vehicles used and the distance traveled. This new formulation includes a model with 12 variants of VRP: CVRP, VRPTW, VRPMTW, MDVRP, SDVRP, sdVRP, VRPM, HVRP, CCVRP, DDVRP, rdVRP and OVRP, described in section 2.1. A study of complexity factors of the base case of RoSLoP sets that Rangel's mathematical formulation [28] requires 2^6 integer variables to solve 30 restrictions, it solves five VRP variants. Herrera's approach [21] needs 2^9 integer variables to solve 30 restrictions of the formulation for solving 11 VRP variants. The proposed formulation contains 2^2 integer variables and 15 restrictions to solve 12 VRP variants.

4.3 A Reduction Technique for the Loading Elements

The loading dataset of RoSLoP was defined in [11] as a set of *n*-variant units, in which a unit is defined *unit*=(*w*,*h*,*sw*,*idp*,*kp*,*pkt*,*bt*). A preprocessing must be done to reduce the loading dataset into a linear set of object in the domain of multiple variants. The loading dataset is characterized in a linear dataset through a reduction technique, developed through the learned reference of this case study: "bigger objects are the most weightened". The reduction technique is illustrated in Fig. 5, in which, an order of a customer *j* is transformed. It consists of two steps: 1) the construction of loading units and 2) the transformation of these units in a representative set of real numbers.

The construction of loading units, done through the DiPro, is invoked. As a result, two kinds of units are created: homogeneous and heterogeneous platforms. Homogeneous platforms are constituted by products of the same type, while heterogeneous platform are constituted with different types of products with similar characteristics. Both, homogeneous and heterogeneous platforms are defined as a set $ITEMS_j = \{\forall(w_i, h_i)\}$, where w_i and h_i represent the weight and height of each

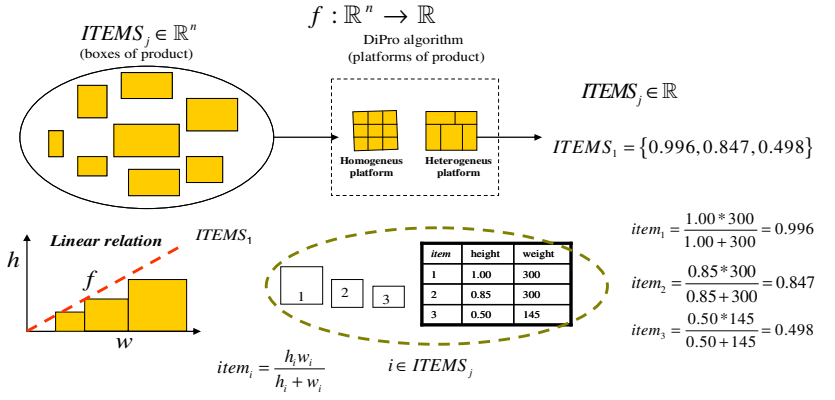


Fig. 5. Transformation of the orders dataset of the case study

unit. Then, each pair (w_i, h_i) is transformed into a number $item_i$ using the following Equation 10. This equation represents the relationship between the dimensions of the objects. A detailed review of DiPro is presented in [21].

$$item_i = \frac{h_i w_i}{h_i + w_i} \qquad i \in ITEMS_j \qquad (10)$$

The capacity C_{vj} of a vehicle v to visit node j is transformed likewise. Each container that belongs to a trailer has two attributes: a high $hpallet_{ij}$ and weight $wpallet_{ij}$ of the assigned load to visit customer j. The width of the load is determined by a categorization of products, asked the company to group the products. This is necessary for adjusting the load to the containers. The transformation of the vehicles dimensions is shown in Fig. 6.

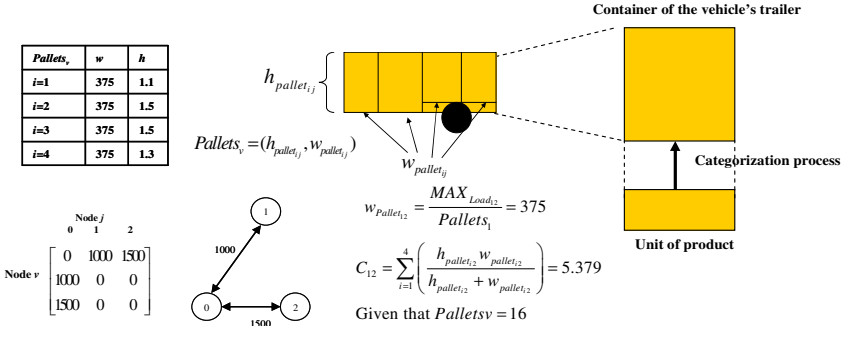


Fig. 6. Transformation of the vehicles dimensions

It is established a uniform distribution of weight for the load in each container. Equation 11 is used to obtain the capacity of the vehicle. It ensures that the dimensions of the load objects and the vehicles are equivalent.

$$C_{vj} = \sum_{i=1}^{Pallets_v} \frac{h_{pallet_{ij}} w_{pallet_{ij}}}{h_{pallet_{ij}} + w_{pallet_{ij}}} \qquad j \in C, v \in V_d \tag{11}$$

Once defined the input parameters and the instance is preprocessed; these elements are used to formulate the integer programming model. The combination of these elements generates the exact solution for the related rich VRP.

5 The Exact Approach of RoSLoP

The objective of RoSLoP is to minimize the assigned vehicles and the distance traveled, visiting all the customer facilities and satisfying the demands. Each route k is constituted by a subset of facilities to be visited and a length φ_k. Equation 12 is used to get the maximum covering set established by the use of variant HVRP. Equations 12-14 permit obtaining the length and the travel time on a route k.

$$|K_d| = |V_d| \left\lceil \frac{\arg max(I_j)}{\arg min(C_{vj})} \right\rceil \qquad j \in C, K_d \in K, v \in V_d \tag{12}$$

$$\varphi_k = \sum_{j \in C \cup D} \sum_{i \in C \cup D} c_{ij} x_{ijk} \qquad k \in K, v \in V_d \tag{13}$$

$$t_k = \sum_{j \in C \cup D} \sum_{i \in C \cup D} t_{ij} x_{ijk} + \sum_{i \in C \cup D} \sum_{j \in C \cup D} \sum_{v \in V_d} tm_{vj} x_{ijk} y_{vk} \qquad k \in K_d \tag{14}$$

The objective function of the problem, defined by Equation 15, minimizes the number of assigned vehicles and the length of all the routes generated, formulated according VRP. Equations 16-18 are used to generate feasible routes and solve the related TSP problem. Equation 16 restricts each edge (i, j) on a route k to be traversed only once. Equation 17 ensures that route k is continuous. Equation 18 is used to optimize the covering set related with the objective function. These equations solve variants DDVRP and MDVRP. Equations 19-22 compute the time used by a vehicle assigned to route k. Equation 22 ensures that the use of a vehicle does not exceed the attention time at facility j. These equations permit solving the variants VRPTW, VRPMTW and VRPM. The variants CCVRP and SDVRP are solved using the Equations 23-24, which ensures that two routes k and k' do not intersect each other at a facility j.

$$\min \; z = \sum_{k \in K_d} \sum_{v \in V_d} \varphi_k y_{vk} \tag{15}$$

$$\sum_{i \in C \cup D} x_{ijk} = 1 \qquad k \in K_d, j \in C \cup D \tag{16}$$

$$\sum_{i \in C \cup D} x_{ijk} - \sum_{i \in C \cup D} x_{jik} = 0 \qquad k \in K_d, j \in C \cup D \tag{17}$$

$$\sum_{i \in C \cup D} \sum_{j \in C \cup D} x_{ijk} \geq 1 \qquad k \in K_d \tag{18}$$

$$t_k y_{vk} \leq stime_v \qquad k \in K_d; v \in V_d \tag{19}$$

$$l_{jk} \geq a_{jk} \sum_{i \in C \cup D} t_{ij} x_{ijk} + \sum_{v \in V_d} tm_{vj} y_{vk} \qquad j \in C \cup D, k \in K_d \tag{20}$$

$$a_{jk} \sum_{i \in C \cup D} x_{ijk} = \sum_{i \in C \cup D} t_{ij} x_{ijk} \qquad j \in C \cup D, k \in K_d \tag{21}$$

$$a_{jk} \leq l_{jk} \leq a_{jk'} \qquad k < k', \forall k, \forall k' \in K_d \tag{22}$$

Equations 23-25, combined with the linear transformation function, define the restrictions for variants CVRP, sdVRP, rdVRP and HVRP. Equation 23 establishes that a vehicle is assigned to a route *k*. Equation 24 ensures that vehicle capacities are not exceeded. Equation 25 establishes that all goods must be delivered and all demands are satisfied. The relaxation of the model that permits the solution of the variant OVRP consists of the reformulation of Equation 17 through 26.

$$\sum_{v \in V_d} y_{vk} \leq 1 \qquad k \in K_d \tag{23}$$

$$\sum_{i \in C \cup D} \sum_{r \in I_j} item_r x_{ijk} \leq C_{vj} y_{vk} \qquad j \in C \cup D, k \in K_d, v \in V_d \tag{24}$$

$$I_j - \sum_{i \in C \cup D} \sum_{r \in I_j} item_r x_{ijk} = 0 \qquad j \in C \cup D, I_j \in Q \tag{25}$$

$$\sum_{i \in C \cup D} x_{ijk} - \sum_{i \in C \cup D} x_{jik} \leq 1 \qquad k \in K_d, j \in C \cup D \tag{26}$$

The right side of Equation 26 is 0 when a route starts and ends at a depot; otherwise, when the right side has the value 1 means that route starts at a depot and finishes at a different facility.

6 The Heuristic Solution of RoSLoP

The methodology of solution, shown in Fig. 7, creates a feasible solution for RoSLoP. The Routing-Scheduling is solved by the basic ACS algorithm and three

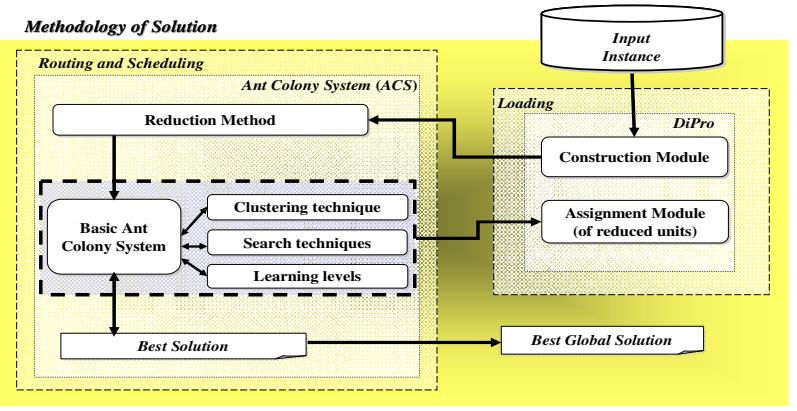


Fig. 7. Methodology of Solution

modules of improvement: reduction complexity, search and learning techniques. The DiPro algorithm, created in [11] was modified to compute the solution for the reduced elements and its structure was reduced to the Construction and Assignment modules for this heuristic solver.

The scheme of solution is shown in Fig. 7. When a solution for Routing-Scheduling is built, the Assignment module is invoked to place the transformed units into the mobile units. This scheme is able to generate feasible solutions for RoSLoP, using the developed ant colony system algorithm and the improvement techniques described in section 3.

7 Experimentation and Results

7.1 The Dataset of Solomon

The performance of the basic ACS, described in section 3.1 and proposed in [14] was compared versus the ACS plus Search Techniques (ACS+ST), ACS plus Clustering Techniques (ACS+CT) and ACS plus Learning for Levels (ACS+LL). All the algorithms were applied to solve the Vehicle Routing Problem with Time Windows of Solomon's dataset, which is formed of 56 instances grouped in six sets named: C1, C2, R1, R2, RC1, RC2; everyone with 9,8,12,11,8 and 8 cases with n = 100 nodes respectively.

The set of instances is named according the characteristics of the cases that form them: the group C is formed by customers in one area. Instances of type R have sets of customers uniformly distributed over a squared area; the group RC has a combination of customers uniformly distributed and grouped in areas. The instances of type 1 have reduced time Windows and small capacity for the vehicles while instances of type 2 have wider time windows and bigger capacity for the vehicles. Therefore, solutions of instances type 2 have few routes and more customers visited by a route. Best known solutions are referenced to VRPWEB [16]. The configuration of ACS was: q_0 = 0.65, β = 6 and ρ = 0.1, with 15

generations per colony and 100 ants per generation, according to [14], where the optimal number of ants is equivalent to the number of nodes of the instance. The ACS developed was coded in C# and it was executed during 1800 seconds (30 minutes). Results are shown in table 2.

Table 2. Comparative of ACS plus Improvement Techniques

Instances dataset	**Distance Traveled**				
	ACS	ACS+ ST	ACS+ CT	ACS+ LL	Best known solution
C1	883.30	829.77	1894.17	887.17	828.38
C2	634.48	595.01	1643.15	633.67	589.86
R1	1455.35	1271.85	1902.76	1437.93	1208.50
R2	1221.16	980.20	1992.72	1245.66	961.72
RC1	1627.36	1441.39	2251.40	1599.03	1377.79
RC2	1452.36	1178.50	1972.77	1456.16	1119.79
Accumulated	68559.44	59231.57	101876.14	68452.79	57231.0

Table 2. Continuation

Instances dataset	**Vehicles Used**				
	ACS	ACS+ ST	ACS+ CT	ACS+ LL	Best known solution
C1	10.00	10.00	17.78	10.00	10.00
C2	3.00	3.00	10.00	3.00	3.00
R1	13.58	12.75	23.00	13.42	11.92
R2	3.09	3.09	9.27	3.00	2.73
RC1	13.00	12.50	21.75	13.00	11.50
RC2	3.38	3.38	10.88	3.38	3.25
Accumulated	442	422	879	439	405

Table 2 shows that the search techniques (ACS+ST) reaches the best solutions for the test dataset in Distance Traveled, obtaining an approach of 96%. The ACS+ST reaches the most of the best solutions in vehicles used, reaching an efficience of 92% with regard to the best known solution reported for the Solomon's dataset.

However, for the set R2, ACS+LL improves the performance of ACS+ST, which indicates that ACS+LL could be efficient for some kind of problems while for VRPTW search tequniques are better, which approaches the no-free lunch theorem [35] for optimization problems. Experimentation with real-world instances is observed next insection. The ACS+CT does not improve the solution of the test dataset, therefore other clustering technique is necessary to improve its performance.

7.2 Experimentation with RoSLoP Instances

A set of test instances were provided by the bottled company; they were used to evaluate the performance of the improvement techniques for ACS. The ACS was

Table 3. Experimentation with real-world instances, provided by the bottling company

Instance	Distance Traveled				
	ACS	ACS+ ST	ACS+ CT	ACS+ LL	Optimal Solution
06/12/2005	1444	1444	1444	1444	1444
09/12/2005	1580	1580	1580	1580	1580
12/12/2005	2500	2500	2500	2500	2500
01/01/2006	2560	2560	2560	2560	2560
03/01/2006	1340	1340	1340	1340	1340
07/02/2006	2660	2660	2660	2660	2660
13/02/2006	1980	1980	1980	1980	1980
06/03/2006	1960	1960	1960	1960	1960
09/03/2006	2570	2570	2570	2570	2570
22/04/2006	3358	3358	3358	3358	3358
14/06/2006	2350	2350	2350	2350	2350
04/07/2006	2640	2640	2640	2640	2640
Average	2245	2245	2245	2245	2245

Table 3. Continuation

Instance	Vehicles Used				
	ACS	ACS+ ST	ACS+ CT	ACS+ LL	Optimal solution
06/12/2005	6	4	4	4	3
09/12/2005	7	5	5	5	5
12/12/2005	8	6	6	6	6
01/01/2006	7	7	7	7	6
03/01/2006	5	5	5	4	4
07/02/2006	8	6	6	6	6
13/02/2006	7	5	5	5	5
06/03/2006	5	5	5	5	5
09/03/2006	8	6	6	6	6
22/04/2006	7	7	7	7	7
14/06/2006	8	6	6	6	6
04/07/2006	7	6	6	6	6
Average	7	5.66	5.66	5.58	5.33

developed assisting the necessities of the company, which were formulated in section 3 and it was coded in C#.

A set of 12 instances of test were selected of the database of the company, which contains 312 instances classified by the date of the orders. The database contains also 1257 orders and 356 products in its catalogues. Both algorithms were configured the parameters of the ACS: $q_0 = 0.9$; $\beta = 1$ and $\rho = 0.1$. The number of ants is defined dynamically according to [14], which is equal to the number of customers of each instance. It was disposed eight available vehicles and the ACS was executed two minutes using the different techniques.

Results are shown in tables 2 and 3. The ACS was tested using the techniques of improvement: ACS plus Search Techniques (ACS+ST), ACS plus Clustering Technique (ACS+CT) and ACS plus Learning for Levels (ACS+LL). The experiment consisted in the measurement of the time in which ACS obtains the best solution. The results are shown in tables 3 and 4.

Table 3 shows that all the techniques reach the 100% in distance traveled. The basic ACS reaches 24% of efficiency in vehicles used while ACS+ST and ACS+CT reaches the 75% and ACS+LL reaches 83%. In contrast with VRPTW, where local search is the best technique, ACS+LL reach an improve for the other techniques of ACS, it reveals that learning for levels is efficient for solving RoSLoP, improving in 8% the solution of ACS+ST and ACS+CT with regard to the found optimal solutions reached; and 20% with regard the average performance of the basic ACS. The measurement of time and the reduced set of loading objects are observed in table 4.

Table 4 shows that ACS-CT and ACS-ST reduce 73% the computation time of the Basic ACS while ACS-LL reduces the computation time in 75% with regard to the basic ACS. It is observed that the loading dataset is reduced 97%, which is

Table 4. Execution time of ACS solving RoSLoP instances

Instance	***n***	***Q***		**ACS***	**ACS+ ST**	**ACS+ CT**	**ACS+ LL**
		boxes*	***units***				
06/12/2005	4	6928	158	51.33	12.03	12.03	10.51
09/12/2005	5	7600	171	73.66	15.64	15.64	17.79
12/12/2005	7	11541	250	58.35	19.58	19.58	16.93
01/01/2006	6	9634	286	75.64	15.31	15.31	14.02
03/01/2006	4	5454	116	63.02	13.96	13.96	10.71
07/02/2006	6	12842	288	83.74	20.03	20.03	18.72
13/02/2006	5	9403	208	65.52	15.93	15.93	15.71
06/03/2006	6	9687	224	32.61	16.49	16.49	14.09
09/03/2006	6	12319	269	56.18	18.73	18.73	16.53
22/04/2006	8	16903	381	77.36	18.56	18.56	16.61
14/06/2006	7	9857	245	40.84	16.09	16.09	13.93
04/07/2006	7	11331	270	42.48	16.92	16.92	15.12
Average	6	10291	238	60.05	16.18	16.18	14.98

proven in experimentation. It was demonstrated also that, ACS-LL searches high-quality solutions faster than other techniques, which permit it to reach the 8% more of efficiency with regard to ACS-CT and ACS-ST. It reveals that learning for levels improve the performance of the developed techniques solving RoSLoP. ACS-ST and ACS-CT needs the same computation time as a consequence of the small size and the characteristics of the instances. Therefore, the clustering technique optimizes in this case the performance for the basic ACS.

8 Conclusions and Future Contributions

This paper presented some techniques mostly used in the development of ant algorithms, which were applied to the solution of an ant colony system algorithm for solving VRPTW and a real-world problem (RoSLoP). These techniques demonstrated experimentally that are efficient in a real environment.

It was demonstrated that the learning for levels is a technique fast and efficient for some kind of problems, it was used to obtain high-quality solutions in a smaller time than other techniques. Search techniques and learning for levels are better for VRPTW, while learning for levels demonstrated a better performance solving RoSLoP with regard to the exact solution of reference for RoSLoP, reported in [13].

Even though, not all the optimal solutions were reached. For this reason, for future work is proposed the combined use of these techniques or a parameters adjusting for the elements of ACS to reach better-quality solutions and the solution of large-scale instances of RoSLoP.

References

1. Archetti, C.: The Vehicle Routing Problem with capacity 2 and 3, General Distances and Multiple Customer Visits. Operational Research in Land and Resources Manangement, 102 (2001)
2. Bianchi, L.: Notes on Dynamic Vehicle Routing. Technical Report IDSIA - Istituto Dalle Molle di Studi sull'Intelligenza Artificiale, Switzerland (2000)
3. Bock, F.: An algorithm for solving traveling salesman and related network optimization problems. In: Fourteenth National Meeting of the Operational Research Society of America, St. Louis, MO, USA (1958)
4. Bullnheimer, B., Hartl, R.F., Strauss, C.: A New Rank Based Version of the Ant System: A Computational Study. Technical report, Institute of Management Science, University of Vienna, Austria (1997)
5. Cano, I., Litvinchev, I., Palacios, R., Naranjo, G.: Modeling Vehicle Routing in a Star-Case Transporation Network. In: XVI International Congress of Computation, CIC-IPN (2005)
6. Colorni, A., Dorigo, M., Maniezzo, V.: An Investigation of Some Properties of an Ant Algorithm. In: Manner, R., Manderick, B. (eds.) Proceedings of PPSN-II, Second International Conference on Parallel Problem Solving from Nature, pp. 509–520. Elsevier, Amsterdam (1992)

7. Colorni, A., Dorigo, M., Matienzo, V.: Distributed Optimization by An Colonies. In: Varela, F.J., Bourgine, P. (eds.) Proc. First European Conference on Artificial Life, pp. 134–142. MIT Press, Cambridge (1992)
8. Cordeau, F., et al.: The VRP with time windows. Technical Report Cahiers du GERAD G-99-13, Ecole des Hautes ´Etudes Commerciales de Montreal (1999)
9. Croes, G.: A method for solving traveling salesman problems. Proc. Operations Research 5, 791–812 (1958)
10. Cruz, L., et al.: An Ant Colony System to solve Routing Problems applied to the delivery of bottled products. In: An, A. (ed.) Foundations of Intelligent Systems. LNCS (LNAI), vol. 4994, pp. 68–77. Springer, Heidelberg (2008)
11. Cruz, L., et al.: DiPro: An Algorithm for the Packing in Product Transportation Problems with Multiple Loading and Routing Variants. In: Gelbukh, A., Kuri Morales, Á.F. (eds.) MICAI 2007. LNCS (LNAI), vol. 4827, pp. 1078–1088. Springer, Heidelberg (2007)
12. Dantzig, G.B., Ramser, J.H.: The Truck Dispatching Problem. Management Science 6(1), 80–91 (1959)
13. Delgado, J., et al.: Construction of an optimal solution for a Real-World Routing-Scheduling-Loading Problem. Journal Research in Computing Science 35, 136–145 (2008)
14. Dorigo, M., Gambardella, L.: Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem, Technical Report TR/IRIDIA/1996-5, IRIDIA, Université Libre de Bruxelles (1996)
15. Dorigo, M.: Positive Feedback as a Search Strategy. Technical Report. No. 91-016. Politecnico Di Milano, Italy (1991)
16. Dorronsoro, B.: The VRP Web. AUREN. Language and Computation Sciences of the University of Malaga (2005), http://neo.lcc.uma.es/radi-aeb/WebVRP
17. Fleischmann, B.: The Vehicle routing problem with multiple use of vehicles. Working paper, Fachbereigh Wirtschafts wissens chaften, Universitt Hamburg (1990)
18. Gambardella, L., Dorigo, M.: Ant-Q: A Reinforcement Learning Approach to the Traveling Salesman Problem. In: Prieditis, A., Russell, S. (eds.) Proceedings of ML-95, Twelfth International Conference on Machine Learning, Tahoe City, CA, pp. 252–260. Morgan Kaufmann, San Francisco (1995)
19. Goel, A., Gruhn, V.: Solving a Dynamic Real-Life Vehicle Routing Problem. In: Haasis, H.-D., et al. (eds.) Operations Research Proceedings 2005, Bremen, Deutschland (2005)
20. Hasle, G., Kloster, O., Nilssen, E.J., Riise, A., Flatberg, T.: Dynamic and Stochastic Vehicle Routing in Practice. Operations Research/Computer Science Interfaces Series, vol. 38, pp. 45–68. Springer, Heidelberg (2007)
21. Herrera, J.: Development of a methodology based on heuristics for the integral solution of routing, scheduling and loading problems on distribution and delivery processes of products. Master's Thesis. Posgrado en Ciencias de la Computación. Instituto Tecnológico de Ciudad Madero, México (2006)
22. Jacobs, B., Goetshalckx, M.: The Vehicle Routing Problem with Backhauls: Properties and Solution Algorithms. Techincal report MHRC-TR-88-13, Georgia Institute of Technology (1993)
23. Mariano, C.E., Morales, E.F.: DQL: A New Updating Strategy for Reinforcement Learning Based on Q-Learning. In: Mariano, C., Morales, E. (eds.) ECML 2001. LNCS (LNAI), vol. 2167, pp. 324–335. Springer, Heidelberg (2001)

24. Mingozzi, A.: An exact Algorithm for Period and Multi-Depot Vehicle Routing Problems. Department of Mathematics, University of Bologna, Bologna, Italy (2003)
25. Pisinger, D., Ropke, S.: A General Heuristic for Vehicle Routing Problems. Tech. report, Dept. of Computer Science, Univ. Copenhagen (2005)
26. Potvin, J., Rousseau, J.: An Exchange Heuristic for Routeing Problems with Time Windows. Proc. Journal of the Operational Research Society 46, 1433–1446 (1995)
27. Prosser, P., Shaw, P.: Study of Greedy Search with Multiple Improvement Heuristics for Vehicle Routing Problems. Tech. report, University of Strathclyde, Glasgow, Scotland (1996)
28. Rangel, N.: Analysis of the routing, scheduling and loading problems in a Products Distributor. Master Thesis. Posgrado en Ciencias de la Computación. Instituto Tecnológico de Ciudad Madero, México (2005)
29. Shaw, P.: Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems. In: Maher, M. (ed.) CP 1998. LNCS, vol. 1520, pp. 417–431. Springer, Heidelberg (1998)
30. Stützle, T., Hoos, H.: Improving the Ant System: A detailed report on the MAX-MIN Ant System. Technical report AIDA-96-12, FG Intellektik, FB Informatik, TU Darmstadt (1996)
31. Taillard, E.: A Heuristic Column Generation Method For the Heterogeneous Fleet VRP. Istituto Dalle Moli di Studi sull Inteligenza Artificiale, Switzerland. CRI-96-03 (1996)
32. Taillard, E., Badeau, P., Gendreu, M., Guertin, F., Potvin, J.Y.: A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows. Transportation Science 31, 170–186 (1997)
33. Thangiah, S.: A Site Dependent Vehicle Routing Problem with Complex Road Constraints. Artificial Intelligence and Robotics Laboratory, Slippery Rock University, U.S.A. (2003)
34. Toth, P., Vigo, D.: The vehicle routing problem, Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics (2001)
35. Wolpert, D.H., et al.: No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation (1997)

Comparison of Fuzzy Edge Detectors Based on the Image Recognition Rate as Performance Index Calculated with Neural Networks

Olivia Mendoza[1], Patricia Melin[2], Oscar Castillo[2], and Juan Ramon Castro[1]

[1] Universidad Autonoma de Baja California
[2] Instituto Tecnologico de Tijuana
omendoza@uabc.edu.mx, pmelin@tectijuana.mx,
ocastillo@tectijuana.mx, jrcastror@uabc.edu.mx

Abstract. Edge detection is a process usually applied to image sets before the training phase in recognition systems. This preprocessing step helps to extract the most important shapes in an image, ignoring the homogeneous regions and remarking the real objective to classify or recognize. Many traditional and fuzzy edge detectors can be used, but it's very difficult to demonstrate which one is better before the recognition results. In this work we present an experiment where several edge detectors were used to preprocess the same image sets. Each resultant image set was used as training data for neural network recognition system, and the recognition rates were compared. The goal of this experiment is to find the better edge detector that can be used as training data on a neural network for image recognition.

1 Introduction

In previous works we proposed some extensions for traditional edge detectors to improve them using fuzzy systems [14][16][19]. In all the experiments we show the resulting images, demonstrating that the images obtained with fuzzy systems were visually better than the obtained with the traditional methods.

Now the next research consists on developing formal validations for our fuzzy edge detectors using different methods. In the literature we find comparison of edge detectors based on human observations [5][8][9][11][12], and some others that found the better values for parametric edge detectors [23].

Edge detectors can be used in systems for different purposes, but in our research group we are particularly interested in knowing, which is the better edge detector for a neural recognition system. In the present work we present some experiments that show fuzzy edge detectors are a good method to improve the performance of neural recognition systems, for this reason we can use the recognition rate with neural networks as edge detection performance index.

P. Melin et al. (Eds.): Soft Comp. for Recogn. Based on Biometrics, SCI 312, pp. 389–399.
springerlink.com

2 Overview of the Tested Edge Detectors

2.1 Sobel Edge Detector Improved with Fuzzy Systems

In Sobel edge detector we used the Sobel operators $Sobel_x$ and $Sobel_y$ as in the traditional method, and then we substitute the Euclidean distance equation (1) using instead a fuzzy system, as we show in Fig. 1 [19].

$$Sobel\_edges = \sqrt{Sobel_x^{\,2} + Sobel_y^{\,2}} \tag{1}$$

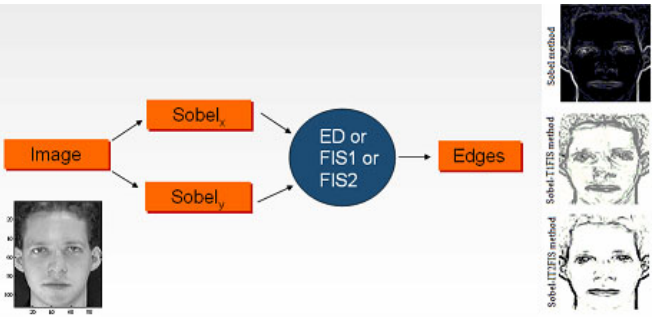


Fig. 1. Scheme of the Sobel edge detector improved with fuzzy systems.

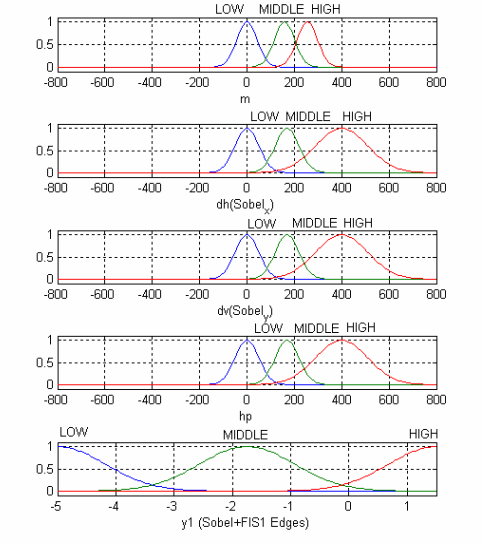


Fig. 2. Variables for the Sobel+FIS1 Edge Detector

Sobel operators are the main inputs for the type-1 fuzzy inference system (FIS1) and type-2 fuzzy inference system (FIS2), and we also made experiments adding two more inputs, that are filters that improve the final edge image. The fuzzy variables used in the Sobel+FIS1 and Sobel+ FIS2 edges detectors are shown in Fig. 2 and Fig. 3 respectively.

The use of the FIS2 [6][7] provided images with best defined edges than the FIS1, which is a very important result in providing better inputs to the neural networks that will perform the recognition task.

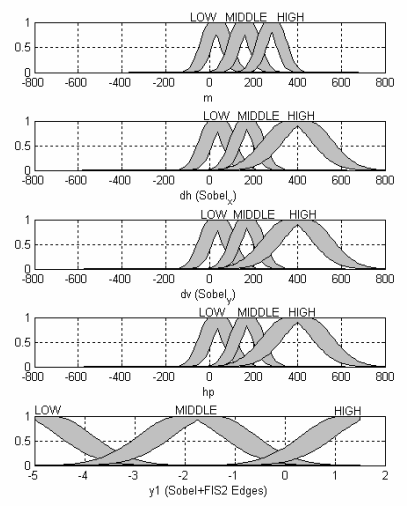


Fig. 3. Variables for the Sobel+FIS2 Edge Detector

The rules for the FIS1 and FIS2 are the same and are shown below:

1. If (dh is LOW) and (dv is LOW) then (y1 is HIGH)
2. If (dh is MIDDLE) and (dv is MIDDLE) then (y1 is LOW)
3. If (dh is HIGH) and (dv is HIGH) then (y1 is LOW)
4. If (dh is MIDDLE) and (hp is LOW) then (y1 is LOW)
5. If (dv is MIDDLE) and (hp is LOW) then (y1 is LOW)
6. If (m is LOW) and (dv is MIDDLE) then (y1 is HIGH)
7. If (m is LOW) and (dh is MIDDLE) then (y1 is HIGH)

The rules set shown above infers the gray tone of each pixel for the edge image with the follow reasoning: When the horizontal gradient dh and vertical gradient dv are LOW means that do not difference exists between the gray tones in it's neighbors pixels, then the output pixel must belong of an homogeneous or not edges region, then the output pixel is HIGH or near WHITE. In opposite case, when dh and dv are HIGH means that a difference exists between the gray tones in its neighborhood, then the output pixel is an EDGE.

3 Morphological Gradient Detector Improved with Fuzzy Systems

In the morphological gradient, we calculated the four gradients as in the traditional method [1][4], and substitute the sum of gradients equation (2), using instead a fuzzy inference system, as we show in Fig. 4.

$$MG\_edges = D_1 + D_2 + D_3 + D_4 \tag{2}$$

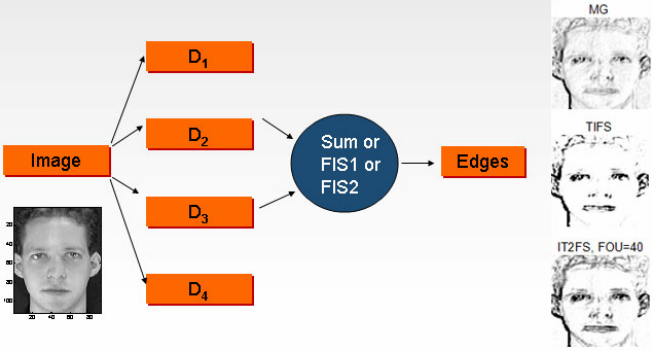


Fig. 4. Scheme of the Morphological gradient edge detector improved with fuzzy systems.

The fuzzy variables used in the MG+FIS1 and MG+FIS2 edges detectors are shown in Fig. 5 and Fig. 6 respectively.

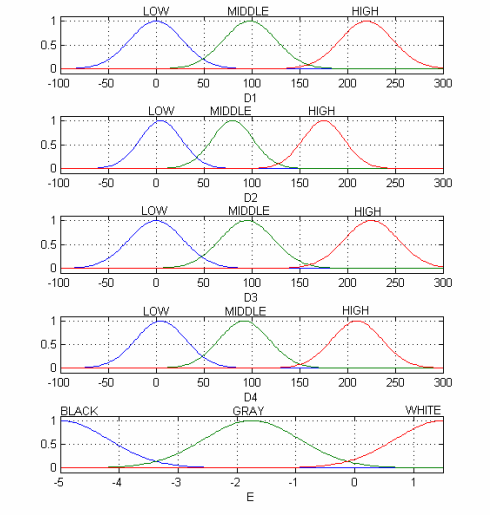


Fig. 5. Variables for the MG+FIS1 Edge Detector

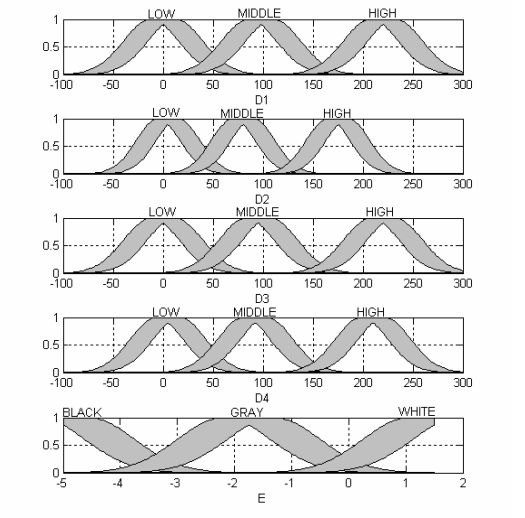


Fig. 6. Variables for the MG+FIS2 Edge Detector

The rules for the FIS1 and FIS2 are the same shown below:

1. If (D1 is HIGH) or (D2 is HIGH) or (D3 is HIGH) or (D4 is HIGH) then (E is BLACK)
2. If (D1 is MIDDLE) or (D2 is MIDDLE) or (D3 is MIDDLE) or (D4 is MIDDLE) then (E is GRAY)
3. If (D1 is LOW) and (D2 is LOW) and (D3 is LOW) and (D4 is LOW) then (E is WHITE)

After many experiments we found that an edge exists when any gradient Di is HIGH, that means, a difference of gray tones in any direction of the image must produce a pixel with BLACK value or EDGE. The same behavior occurs when any gradient Di is MIDDLE, that means even when the difference in the gray tones does not are maximum, the pixel is an EDGE, then the only rule that found a non edge pixel is the number 3, where only when all the gradients are LOW, the output pixel is WITHE, that means a pixel belonging to an homogeneous region.

4 Design of the Experiment

The experiment consists on applying a neural recognition system using each of the follow edge detectors: Sobel, Sobel+FIS1, Sobel+FIS2, Morphological Gradient (MG), Morphological Gradient+FIS1 and Morphological Gradient+FIS2.

4.1 General Algorithm Used for the Experiment

1. Define the database folder.
2. Define the edge detector.
3. Detect the edges of each image as a vector and store it as a column in a matrix.

4. Calculate the recognition rate using the k-fold cross validation method.
    a. Calculate the indices for training and test k folds.
    b. Train the neural network k-1 times, one for each training fold calculated previously.
    c. Test the neural network k times, one for each fold test set calculated previously.
5. Calculate the mean of rate for all the k-folds.

4.2 Parameters Depend on the Database of Images

The experiment described can be performed with databases of images used for identification purposes. That is the case of the faces recognition application, then we use three of the most popular sets of images, the ORL database of faces [3], the Cropped Yale database of faces [2][10] and the FERET database of faces [22].

For the three databases we defined the variable ***p*** as people number and ***s*** as samples number for each person. The test made with k-fold cross validation method, with *k=5* for both databases. We can generalize the calculation of folds size ***m*** or number of samples in each fold, dividing the total number of samples for each person ***s*** by the folds number, and then multiplying the result by the people number ***p*** (3), then the train data set size ***i*** (4) can be calculated as the number of samples in ***k-1*** folds ***m***, and test data set size t (5) are the number of samples in only one fold.

$$m = (s / k) * p \tag{3}$$

$$i = m(k - 1) \tag{4}$$

$$t = m \tag{5}$$

The total number of samples used for each people were 10 for the ORL and YALE databases; then if the size m of each 5-fold is 2, the number of samples for training for each people is 8 and for test 2. For the experiments with the FERET database of faces we use only the samples of 74 people who have 4 frontal sample images. The particular information for each database is show in the Table I.

Table I. Particular Information for the Tested Database of Faces

Database	People number (p)	Samples number (s)	Fold size (m)	Training set size (i)	Test set size (t)
ORL	40	10	80	320	80
Cropped Yale	38	10	76	304	76
FERET	74	4	74	222	74

4.3 The Monolithic Neural Network

In previous experiments with neural networks for image recognition, we have found a general structure with acceptable performance, even as it was not optimized. We used the same structure for multi-net modular neural networks, in order to establish a comparison standard for all our experiments [13][15][17][18][19][20][21]. The general structure for the monolithic neural network is shown in Fig. 7:

- Two hidden layers with 200 neurons.
- Learning Algorithm: Gradient descent with momentum and adaptive learning rate back-propagation.
- Error goal 1e-4.

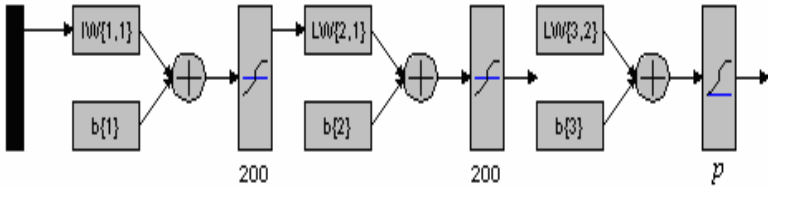


Fig. 7. General structure for the monolithic neural network.

5 Results

In this section we show the numerical results of the experiment. Table II and contains the results for the ORL database of faces, table III contains the results for the Cropped Yale database and table IV contains the results for the FERET database of faces.

Table II. Recognition Rates for the ORL database of faces.

Training set pre-processing method	*Mean time (sec.)*	*Mean rate (%)*	*Standard deviation*	*Max rate (%)*
MG+FIS1	1.2694	89.25	4.47	95.00
MG+FIS2	1.2694	90.25	5.48	97.50
Sobel+FIS1	1.2694	87.25	3.69	91.25
Sobel+FIS2	1.2694	90.75	4.29	95.00

For a better appreciation of the results we made plots for the values presented in the tables above. Even if this work does not pretend to make a comparison using the training times as performance index for the edge detectors, it is interesting that the necessary time to reach the error goal is established for the experiment.

Table III. Recognition Rates for the Cropped Yale database of faces.

Training set pre-processing method	*Mean time (sec.)*	*Mean rate (%)*	*Standard deviation*	*Max rate (%)*
MG+FIS1	1.76	68.42	29.11	100
MG+FIS2	1.07	88.16	21.09	100
Sobel+FIS1	1.17	79.47	26.33	100
Sobel+FIS2	1.1321	90	22.36	100

Table IV. Recognition Rates for the FERET database of faces.

Training set pre-processing method	*Mean time (sec.)*	*Mean rate (%)*	*Standard deviation*	*Max rate (%)*
MG+FIS1	1.17	75.34	5.45	79.73
MG+FIS2	1.17	72.30	6.85	82.43
Sobel+FIS1	1.17	82.77	00.68	83.78
Sobel+FIS2	1.17	84.46	03.22	87.84

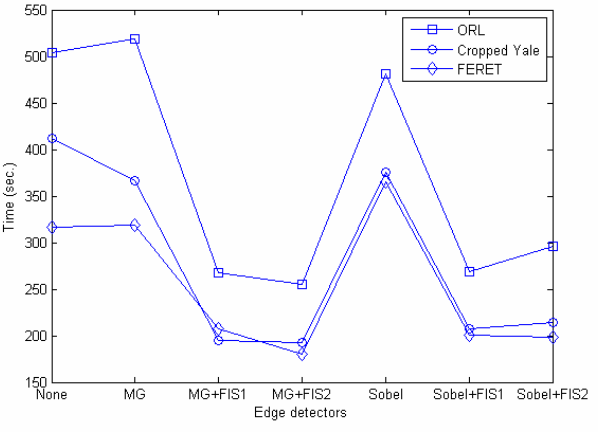


Fig. 8. Training time for the compared edge detectors tested with ORL, Cropped Yale and FERET database of faces.

As we can see in Fig. 8 the lowest training times are for the Morphological Gradient+FIS2 edge detector and Sobel+FIS2 edge detector. That is because both edge detectors improved with interval type-2 fuzzy systems produce images with more homogeneous areas; which means, a high frequency of pixels near the WHITE linguistic values.

But the main advantage of the interval type-2 edges detectors are the recognition rates plotted in Fig. 9, where we can note the best mean performance of the

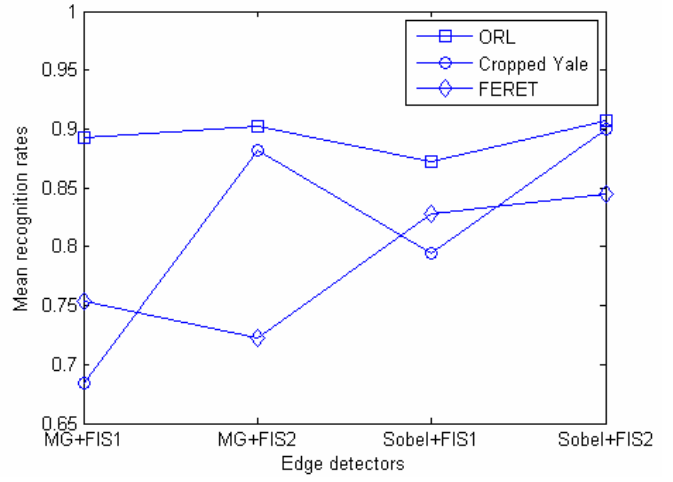


Fig. 9. Mean recognition rates for the compared edge detectors with ORL, Cropped Yale and FERET database of faces.

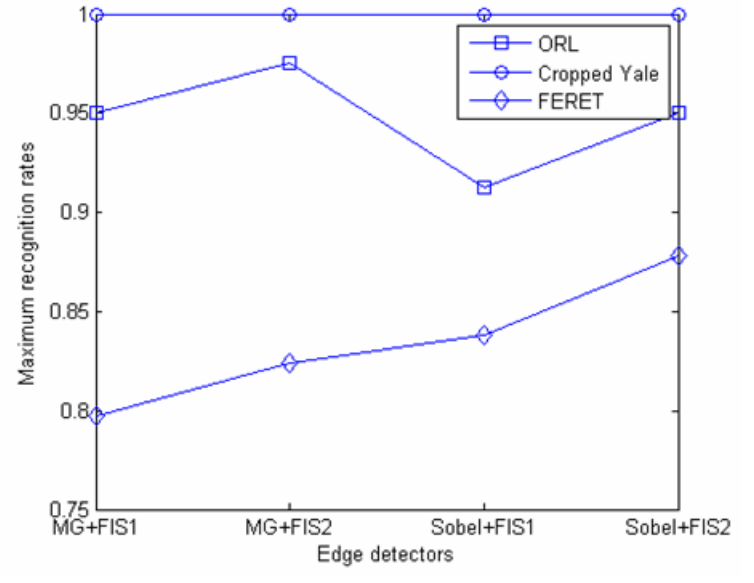


Fig. 10. Maximum recognition rates for the compared edge detectors with ORL, Cropped Yale and FERET database of faces.

neural network when it was trained with the data sets obtained with the MG+FIS2 and Sobel+FIS2 edge detectors.

Fig. 10 shows that the recognition rates are also better for the edge detectors improved with interval type-2 fuzzy systems. The maximum recognition rates

could not be the better parameter to compare the performance of the neural networks depending on the training set; but is interesting to see the maximum recognition rate of 97.5% reached when the neural network was trained with the ORL data set preprocessed with the MG+FIS2. That is important because in a real world system we can use this best configuration for images recognition, expecting good results.

6 Conclusion

This is the first effort for develop a formal comparison method for edge detectors as a function of their performance in different types of systems. In this work we demonstrate that Sobel and Morphological Gradient edge detectors improved with type-2 fuzzy logic have a better performance than the traditional methods in an image recognition neural network system.

References

[1] Evans, A.N., Liu, X.U.: Morphological gradient approach for color edges detection. IEEE Transactions on Image Processing 15(6), 1454–1463 (2006)

[2] Georghiades, A.S., Belhumeur, P.N., Kriegman, D.J.: From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose. IEEE Trans. Pattern Anal. Mach. Intelligence 23(6), 643–660 (2001)

[3] AT&T Laboratories Cambridge, The ORL database of faces, `http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html`

[4] Russo, F., Ramponi, G.: Edge extraction by FIRE operators Fuzzy Systems. In: IEEE World Congress on Computational Intelligence, pp. 249–253 (1994)

[5] Bustince, H., Berrenechea, E., Pagola, M., Fernandez, J.: Interval-valued fuzzy sets constructed from matrices: Application to edge detection. In: Fuzzy Sets and Systems. Elsevier, Amsterdam (2008), `http://www.sciencedirect.com`

[6] Mendel, J.: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice-Hall, U.S.A. (2001)

[7] Castro, J.R., Castillo, O., Melin, P., Rodriguez-Diaz, A.: Building fuzzy inference systems with a new interval type-2 fuzzy logic toolbox. In: Gavrilova, M.L., Tan, C.J.K. (eds.) Transactions on Computational Science I. LNCS, vol. 4750, pp. 104–114. Springer, Heidelberg (2008)

[8] Revathy, K., Lekshmi, S., Prabhakaran Nayar, S.R.: Fractal-Based Fuzzy Technique for Detection of Active Regions From Solar. Journal of Solar Physics 228, 43–53 (2005)

[9] Suzuki, K., Horiba, I., Sugie, N., Nanki, M.: Contour extraction of left ventricular cavity from digital subtraction angiograms using a neural edge detector. In: Systems and Computers Japan, pp. 55–69. Wiley, Japan (2003)

[10] Lee, K.C., Ho, J., Kriegman, D.: Acquiring Linear Subspaces for Face Recognition under Variable Lighting. EEE Trans. Pattern Anal. Mach. Intelligence 27(5), 684–698 (2005)

[11] Hua, L., Cheng, H.D., Zhanga, M.: A High Performance Edge Detector Based on Fuzzy Inference Rules. Information Sciences: An International Journal 177(21), 4768–4784 (2007)

[12] Heath, M., Sarkar, S., Sanocki, T., Bowyer, K.W.: A Robust Visual Method for Assessing the Relative Performance of Edge-Detection Algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(12), 1338–1359 (1997)

[13] Mendoza, O., Melin, P.: The Fuzzy Sugeno Integral As A Decision Operator in The Recognition of Images with Modular Neural Networks. In: Hybrid Intelligent Systems, pp. 299–310. Springer, Germany (2007)

[14] Mendoza, O., Melin, P., Licea, G.: A New Method for Edge Detection in Image Processing Using Interval Type-2 Fuzzy Logic. In: IEEE International Conference on Granular Computing (GRC 2007), Silicon Valley, Ca, U.S.A. (2007)

[15] Mendoza, O., Melin, P., Licea, G.: A Hybrid Approach for Image Recognition Combining Type-2 Fuzzy Logic. In: Modular Neural Networks and the Sugeno Integral, Information Sciences, vol. 179(13), pp. 2078–2101. Elsevier, U.S.A. (2007)

[16] Mendoza, O., Melin, P., Licea, G.: Fuzzy Inference Systems Type-1 And Type-2 for Digital Images Edges Detection, Engineering Letters. In: International Association of Engineers, E.U.A., vol. 15(1) (2007), http://www.engineeringletters.com/issues_v15/issue_1/EL_15_1_7.pdf

[17] Mendoza, O., Melin, P., Licea, G.: Interval Type-2 Fuzzy Logic for Module Relevance Estimation in Sugeno Integration of Modular Neural Networks. In: Soft Computing for Hybrid Intelligent Systems, pp. 115–127. Springer, Germany (2008)

[18] Mendoza, O., Melin, P., Licea, G.: A hybrid approach for image recognition combining type-2 fuzzy logic, modular neural networks and the Sugeno integral. In: Information Sciences, vol. 179(3), pp. 2078–2101. Elsevier, Amsterdam (2008)

[19] Mendoza, O., Melin, P., Licea, G.: Interval type-2 fuzzy logic for edges detection in digital images. International Journal of Intelligent Systems 24(11), 1115–1134 (2009)

[20] Mendoza, O., Melin, P., Licea, G.: Interval type-2 fuzzy logic and modular neural networks for face recognition applications. Applied soft computing journal 9(4), 1377–1387 (2009)

[21] Mendoza, O., Melin, P., Castillo, O., Licea, G.: Type-2 Fuzzy Logic For Improving Training Data And Response Integration In Modular Neural Networks For Image Recognition. In: Foundations of Fuzzy Logic And Soft Computing (LNCC), pp. 604–612. Springer, Germany (2007)

[22] Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J.: The FERET Evaluation Methodology for Face-Recognition Algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(10), 1090–1104 (2000)

[23] Yitzhaky, Y., Peli, E.: A Method for Objective Edge Detection Evaluation and Detector Parameter Selection. IEEE Trans. Pattern Anal. Mach. Intell. 25(8), 1027–1033 (2003)

Intelligent Method for Contrast Enhancement in Digital Video

Roberto Sepúlveda, Oscar Montiel, Alfredo González, and Patricia Melin

[1] Centro de Investigación y Desarrollo de Tecnología Digital (CITEDI- IPN),
Av. del Parque No.1310, Mesa de Otay, 22510, Tijuana, B.C., México
r.sepulveda@ieee.org, o.montiel@ieee.org,
agonzalez@citedi.mx
[2] Division of Graduate Studies and Research,
Calzada Tecnológico S/N, Tijuana, B.C., México
epmelin@hafsamx.org

Abstract. In this paper, an intelligent method for image contrast enhancement in real time digital video applications is proposed. This new technique is based on the generation of adaptive transfer functions by using Neural Network. The method is designed in order to be implemented in digital television systems, based on Thin Film Transistor Liquid Cristal Displays (TFT-LCD), that has become the dominant technology in consumer electronics. The method provides the required amount of contrast enhancement for every image according to real time analysis of brightness and contrast in the video frame information, then a different contrast enhancement transfer curve for each image is generated. The statistical information is extracted by histogram analysis from current or previous image frames. Technological factors are considered where it is applied, as well as aspects of the visual perception within a non-controlled environment, which is the case of the consumer environment. The method is based on the design of a general transfer function model that compensates the decrement in gray scale dynamic range in images representation which is a consequence of the TFT-LCD technology limits; in particular the light leakage from the panel backlight sources of the Cold Cathode Fluorescent Light (CCFL) or the LED type, as well as an unsuitable dynamic control of backlight intensity. Using subjective methods for image quality, the contrast is enhanced in a set of representative images for different degrees of brightness and contrast, a unique transfer function is obtained for each representative image. This set of transfer functions, as well as the statistic of the pixel values distribution from each image frame are used as an input - output pattern during the training of a neural network. On the basis of the experimentation, an evaluation and comparative analysis takes place between the functions of transformation obtained by subjective method of image quality and the curves obtained through the trained neural network.

P. Melin et al. (Eds.): Soft Comp. for Recogn. Based on Biometrics, SCI 312, pp. 401–422.
springerlink.com

1 Introduction

In the last years, the image display technology based on TFT-LCD has become the dominant technology in consumer electronics applications, over passing systems based on PDP (Plasma-Display-Panel), Projection Systems and in many aspects the CRT technology, mainly in the consumer preference as a solution in applications for High-Definition-TV (HDTV). The space for the innovation and improvements within this area of images display technology is still wide open, new techniques are continually introduced for the improvement of the displayed images, specially in image display quality aspects where the CRT technology is still superior, like the static contrast reproduction in the displayed images. The main factor of the low contrast reproduction in televisions with TFT-LCD technology is the light leakage coming from the LCD light source, known as Backlight which can be CCFL or LED type. These lamps provide a constant light source, the liquid crystal (LC) cells of the TFT-LCD panel act like switches controlling the amount of illumination towards the corresponding pixels. When an LC switch is completely closed, the light cannot completely be obstructed, causing a light leakage that avoids that the corresponding pixels be perceived completely dark, deteriorating the static contrast ratio and the displayed image quality specially in dark images. The light leakage degrades the color saturation of displayed images reducing their variation range, and also it contributes to the decrease of the dynamic range in the representation of the gray scale in an image. This problem affects the correct representation of gray levels in dark zones within the image and therefore a decrease in the image contrast ratio.

In this paper a method of contrast enhancement for consumer electronics digital video applications is presented, the method is based on adaptive transfer functions using a neural network. The proposed method takes in consideration technical factors of LCD display technology that degrades the correct reproduction of contrast ratio, besides other aspects like the visual perception and preference of the human being, in relation to the perception of television images in a non controlled environment in terms of ambient light. The proposed method is comprised next:

- Design of a general transfer function model, that compensates the decrease in dynamic range representation of gray scale in displayed images, due to light leakage from backlight in TFT-LCD display technology.
- General transfer functions are obtained using subjective method of contrast enhancement in a set of representative images in brightness and contrast. The input-output patterns from this general transfer functions are used in the training of a Neural Network.
- Transfer functions are obtained using the trained Neural Network.
- The method is evaluated using simulation, and subjective evaluation techniques of image quality in non controlled light environments, such is the case of consumer application environments.

This paper is organized as follow: Section 2 presents the fundamental theory as a basis to the accomplishment of this paper work, image quality concepts in TFT-LCD consumer electronics applications are described; in Section 3 the development

of this method is presented, explaining the contrast enhancement method design which is based on adaptive transfer functions; Section 4 describes the evaluation of the proposed method and give us the corresponding analysis of the obtained results; finally Section 5 discusses some conclusions of this work.

2 General Concepts for an Image Quality in TFT-LCD Consumer Applications

The intention of the proposed method in this work is to improve the visual appearance of an image, in relation to the preference and visual perception of the human being. In the preferential aspect of the human being like consumer and spectator, it is demonstrated that an image with enhancement is subjectively pleasanter than a perfectly reproduced image in its original condition. At present a general theory unifying the concept of quality of an image does not exist, since there is not a standard for the quality of an image that could serve as support criteria for design during the implementation of the stages of the processing of video in a digital television. Generally, we could say that the quality of video image depends on the type of technology, appropriate application of processing techniques and enhancement according to the visual preference of the human being as consumer and spectator.

2.1 Analysis of Brightness and Contrasts in an Image

The basic tool for the analysis of the content of brightness and contrasts in an image is histogram H_f of the image f, which provides information referring to the pixel values that are perceived in a displayed image as levels of gray in the image.

Histogram represents a reduction of the relative dimension with respect to original image f, most part of the information is lost. Image f cannot be deduced of histogram H_f except in trivial cases, for example, a totally white image, i.e., images of constant value. Histogram H_f does not contain spatial information about f, this one describes the frequency of the pixel values. It is possible to know the distribution of the gray levels in an image, the distribution range of these levels of gray, as well as the average optical density, which is the basic measurement of the global average of the gray levels or brightness of displayed image. This value can be calculated directly from the image.

Considering an image of unique value $f(n)$ which is defined over a coordinate system for a bi-dimensional discrete space $n = (n_1, n_2)$, where a finite range is assumed with a dominion [0,N-1]x[0, M-1], contained in a matrix or dimensionally arranged N x M (lines, columns). This discrete space is originated by means of the sampling of a continuous image $f(x, y)$, image $f(n)$ is assumed to be quantified to L levels [0,...,L-1], where a pixel value takes one these integer values. By simplicity, reference to those values is done as levels of gray reflecting the form in which contributes the content of the information of luminance (y) in a displayed image [1, 6, 7, 8, 10]. The calculation of the average optical density from the image is given by:

$$AOD(f) = \frac{1}{NM} \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{M-1} f(n_1, n_2) \tag{1}$$

This can also be calculated from the image histogram:

$$AOD(f) = \frac{1}{NM} \sum_{r=0}^{L-1} rH_f(r) \tag{2}$$

The AOD is a metric to estimate the center of pixel's gray scale distribution within an image, it will be denoted as μ. With this parameter and the standard deviation σ, both obtained from the image histogram, two new parameters are derived, which are denoted as α and β from which we will determine the new distribution of gray levels in the enhanced image. In Figure 1 there are three images and their respective histograms. The content of these images corresponds to a Gaussian noise pattern used for illustrative purposes. In each one of the images, the center of the distribution of gray levels or μ is different, which is an indicator of three images with a relative difference in the brightness level; Figure 2 shows another example, in this case two images with similar brightness and different contrast.

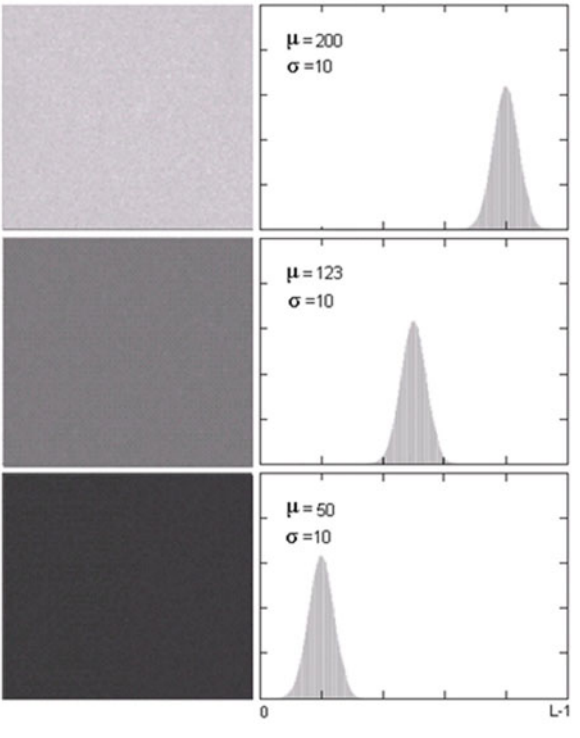


Fig. 1. The relative comparison between the three histograms shows that the images have the same contrast but different brightness level.

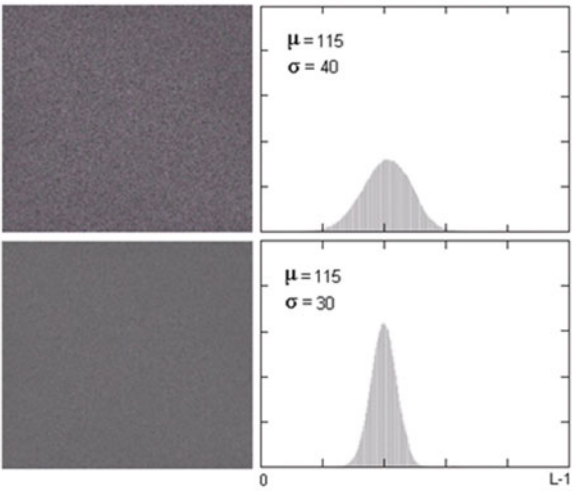


Fig. 2. Two images with similar brightness level but different in contrast.

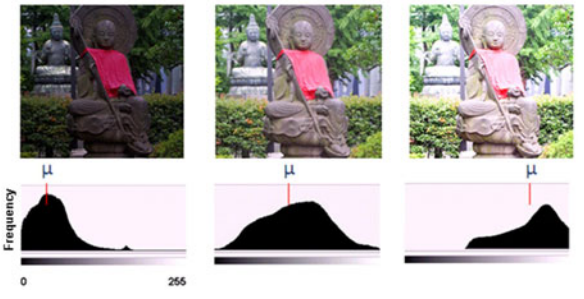


Fig. 3. Three images with different degree of brightness and conrast, as it is shown in the images and their respective histograms.

By making a relative comparison between both histograms from Figure 2, we can say that the images have similar brightness level and there is a higher contrast in the top image. Above concepts can be appreciated in Figure 3, where the images are different in brightness and contrast.

In the image of the center, it is observed that the arithmetic mean of the pixel values μ, is in the average zone of the histogram, allowing a dispersion or expansion of pixel values in both ends, and that corresponds to the zones of low and high illumination. Opposite case of the two images in the edges, where a low contrast is a consequence of having a pixel arithmetic mean value μ near an end and therefore it exists what it is known commonly as ”trims”, in that end of the gray scale.

2.2 Linear Point Operations in Images

An effective way to improve the contrast in an image is by point operations that expand the distribution of gray levels in the image. A point operation of a digital image $f(n)$ is a function g of a single variable applied identically for each pixel in the image, creating therefore a new transformed image $g(n)$, therefore for each coordinate n,

$$g(n) = T[f(n)] \tag{3}$$

The form of the T function is determined by the type of transformation in the image. Since *g(n)* is a function solely for a single value of pixel, the effects that can be obtained by a point operation are limited in a certain way. Specifically in the equation (3), spatial information is not used, for that reason any change in the spatial relationship between the pixels in the transformed image does not take place, and the point operations do not affect the spatial position of objects nor its forms. Each pixel value or gray level can be modified according to the relation in the equation (3), so a point operation T modifies the distribution of gray levels or histogram of an image, and also the global appearance of the image.

2.3 Transformation Function

The method for contrast enhancement described in this paper is based on transformation function, it is a temporal method to obtain the statistic of actual image frame which can be of previous frames. It is also an spatial method referring to the direct manipulation of the pixels within an image. The processing in the space dominion will be denoted by the following expression:

$$g(x,y) = T[f(x,y)] \tag{4}$$

where *f(x, y)* is the input image, *g(x, y)* is the processed image, also named as transformed image, and T is an operator over f defined on one location *(x, y)*.

The simplest form of T is when it is of size 1 x 1, or is a single pixel, in this case g only depends on the value of f in *(x, y)*, T is the function of transformation of gray level (also known as intensity), and it is of the following form:

$$s = T(r) \qquad\qquad 0 \leqslant r \geqslant L-1 \tag{5}$$

where, s and r are the variables that denote the levels of gray for *f(x, y)* and *g(x, y)* respectively, for any point *(x, y)*. If *T(r)* has the form of Figure 4, then the effect of this transformation is an image with a greater contrast than the original image. The levels of gray smaller than m reduced its value, and the levels of pixel over m increased its value in the original image. An expansion of contrast takes place, the r values smaller than m are compressed by the transformation function to a closer rank of s, towards a level of smaller illumination, nearer the black. There is an opposite effect for the values of r greater than m. We can say that we have a

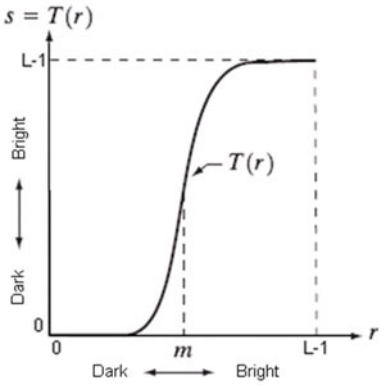


Fig. 4. Transformation function monotonically increasing, of the form *s=T(r)* that produces an *s*-value for each pixel of *r*-value in the original image.

method of processing by point, because the enhancement in any point in an image only depends on the level of gray in that point. There is an example of this type of transformation function in Figure 4. The function of transformation *T(r)* must satisfy the following conditions:

(a) *T(r)* is monotonically increasing in $0 \leqslant r \geqslant L-1$

(b) $0 \leqslant T(r) \geqslant L-1$ for $0 \leqslant r \geqslant L-1$.

The condition established in (a) is necessary to preserve an order in the increase of pixel values from black to white in the output image. A transformation function that does not monotonically increases, can have an output with inverted values of gray levels in at least one section of the image, this effect can be convenient for some applications, nevertheless for applications of video it is not acceptable. The condition (b) guarantees that the levels of gray of the output image fall in the same interval contained in the input image. *L* depends on the quantification used in the stage of pos-processing of video within the architecture of a digital television, where the method is applied. Normally, a digital television is designed to receive and to process different formats and resolutions of video signals; within the video formats, generally these can be composite, component, High Definition Media Interface (HDMI). In resolutions they can be Standard Definition TV (SDTV) and High Definition TV (HDTV), in progressive mode and interlace. Independently of the format and resolution of the video signal at the input of the television, incoming signal is processed and quantified in such way that in the pos-processing stage always is processed with a same quantification, independently of the format and resolution at the input of the television. Currently, a quantification of 8 or 10 bits is used, later, the format and resolution of the video information are again processed to the own resolution of panel LCD-TFT. The final format for display is selectable by menu and depends generally on the preference of the customer as spectator.

3 Design of the Intelligent Method for Contrast Enhancement in Digital Video

This work presents a new adaptive technique of dynamic contrast enhancement in digital video for consumer applications with TFT-LCD display technology , the method is based on the generation of adaptive transformation functions. A different curve for each image on the basis of the statistic is generated from the distribution of pixel values contained in an image frame. The statistical information is extracted by histogram of the present image frame or previous image frames, according to the capacity of the main video processor in the stage of post-processing in the architecture of digital television system where this method is applied. The transformation functions are obtained by a neural network previously trained using statistical data of representative images considering different levels of brightness and pixel values distribution. For each image, the method decides the increase in the dynamic range of the contrast, fitting in an adaptive way the pixel values in the image areas and the required magnitudes. The average level of brightness in the image is preserved thus avoiding deterioration in the quality of image, as it is the case of many of the traditional methods for contrast enhancement, where the brightness of the original image is not preserved, causing steep changes of brightness and saturation of levels of gray in the brighter areas of the image. Such kind of symptoms that deteriorates the quality of images, are not apt for applications in digital television for consumer applications. This method is designed to be used in systems of digital television based on technology of TFT-LCD. It considers some factors, from the point of view of the technology where it is applied as well as the aspects of the visual perception within a non controlled environment, so it is the case of the consumer environment. In this type of environment the conditions of illumination generally are variable and therefore the perception is affected by the light incidence on the spectator and in the screen of the television. A controlled environment like in the cinema, near the 100% of the perceived light comes from the images reproduced in the screen, for that reason the human eye at any moment is adapted to the visual content without interference of the surrounding environment. The proposed method improves the visual appearance of an image in relation to the visual perception of the human being. In the preferential aspect of the human being like consumer and spectator, it is demonstrated that an image with enhancement is subjectively pleasanter than a perfectly reproduced image in its original condition. At present does not exist a general theory unifying the concept of quality in video image since does not exist a general standard for the quality of image that could serve as a criteria for designing during the implementation of the stages of the video processing in a television system. Generally, we could say that the quality of video image depends on the type of technology, appropriate application of processing techniques and enhancement according to the visual preference of the human being.

Although the technology of TFT-LCD is currently placed as the dominant technology in consumer applications, the representation of the contrast of images is an aspect where technological improvement is still being applied with new techniques for the improvement in the displayed images. Specifically for the appropriate

dynamic control of the panel light source, the backlight which are CCFL or LED. One of the major problems is the light leakage trough the panel that degrades the images in the color saturation reducing the variation range, and also it contributes to the decrease of the contrast ratio in the representation of the gray scale in the displayed images. This problem affects the correct Gray representation of levels in dark zones and therefore a decrease in the contrast ratio. Combined to this problem, the majority of the content of video information is adapted to be displayed in CRT, which has a contrast ratio greater than LCD, so it is desirable to enhance the video content when this one is displayed in LCD screens. Existing literature [9, 11, 12, 13], generally handles two methods for contrast enhancement:

- Methods using predefined mathematical functions with semi-fixed parameters, for example sigmoid and exponential functions.
- Global methods of equalization by histogram.

In their great majority, these methods are not apt for applications intended to the consumer due to their lack of flexibility for their implementation and effectiveness. The advantages of the method described in this work, is their wide flexibility to create an infinite diversity of transformation functions with inflexion point, concavities and variable convexities. The inflexion point corresponds at the average level of the pixel values in the image and it is used as a reference in the compensation of the pixel values in the image areas with lower or greater value with respect to this point of average brightness. Resulting in a better expansion of the contrast in the image. In the design of the method, main factors were taken into account since the technology where it is applied, as well as the aspects of the visual perception within an environment with non controlled illumination, as the case of the environment of the consumer.

The Intelligent Method for Contrast Enhancement in Digital Video, was designed to be applied on the basis of transformation curves, being this technique computationally apt for its implementation in real time. The general form of the transformation curves and generally the method for contrast enhancement, was designed on the basis of the following factors:

o **Relationship between the decrease in contrast ratio due to light leakage from the backlight in TFT-LCD**. Light leakage diminish image quality, specially in the visual effect in dark images because the black color can not be reached as a consequence of light leakage through the LCD panel. In order to avoid this problem, the pixel values in the dark zones of the image never are increased and they are compensated increasing the pixel values that are above the arithmetic mean.

o **Perception, adaptation of light and simultaneous contrast**. The human visual system firstly adapts at a threshold level and later it perceives slight variations with respect to this one, therefore the level average of the pixel values of the original image must remain unchanged. Maintaining the average level of the pixel values of the original image, avoids causing abrupt changes in the light intensity between image frame sequences that could be perceivable, especially the variations of light between dark images, symptom known as flicker.

o **Preserve the brightness level of the original image**. The intention to preserve the brightness of the original image is to avoid a possible saturation of gray levels, especially in brighter areas of the images. Preserving the original brightness is a basic concept in contrast enhancement.

3.1 Contrast Ratio and Light Leakage in TFT-LCD Devices

TFT-LCD are transmissive type, they require an internal light source for the radiation of the luminance towards the panel outside. The CCFL or LED light sources are lamps that provide a constant source light, of such form that LC cells of the panel TFT-LCD act like switches controlling the illumination towards the corresponding pixels. When the pixel value is maximum, corresponding LC cell must act like a closed switch, letting pass a maximum of light, corresponding at a maximum level of gray that is the white. Opposite case, when the value of pixel is minimum, switch LC is completely open, nevertheless, the light cannot be obstructed completely, this leakage of light, avoids that the corresponding pixel not perceived with a level of minimum gray that corresponds to the black. The leakage of light especially degrade the images in the color saturation reducing the variation range, and also it contributes to the decrease of the contrast ratio in the representation of the gray scale in an image. This problem mainly affects the correct gray level representation of levels in dark zones of the image and therefore a decrease in the contrast relation. This symptom can be appreciated in the comparative example of Figure 5, where two images corresponding to a signal of video with a level of 0% IRE that corresponds at the minimum level of illumination in an image.

3.2 Light Perception and Simultaneous Contrasts

A gray object is perceived more brilliant if it is seen contrasted in dark surroundings, and darker when it is contrasted in brighter surroundings. This effect is known as simultaneous contrast. It is one of the so many effects that are attributed to the visual process commonly known as lateral inhibition that happens in the retina [15], whereby cells in a region inhibit cells in adjacent regions. It has been demonstrated that a change in the perceptual interpretation can have determining effects on a judgment of the brilliance [3, 5, 16], the theories on human visual perception, as far as the light phenomenon and simultaneous contrast, indicates that it is possible

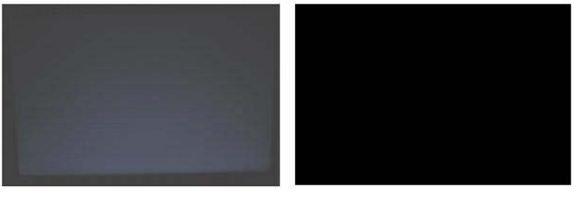


Fig. 5. Left image shows the light leakage sympton in LCD at 0% IRE.

to cause that the levels of gray in an image are perceived with a smaller level if we increased the levels of greater value [8]. This can be appreciated in the following classic example of the phenomenon of simultaneous contrast shown in Figure 6.

Many of the methods for image enhancement, generally for dark images only tends to increase the levels of pixel in the brighter areas of an image, bringing with it a saturation of gray levels in these areas, which can be not absolutely acceptable in applications for TFT-LCD consumer applications. Another example are the global methods like those based on equalization by histogram, that generally in dark images these levels of gray are increased excessively. To increase the pixel values in dark zones can have a side effect, by the aspects mentioned in previous sections, related to the light leakage in TFT-LCD that affects the contrast ratio, and consequently the correct reproduction of dark levels in images.

We can have an effective way for using these concepts, on perception and simultaneous contrasts, to obtain a better perception of the contrast, if for example, in the case of dark images, we increased the levels of pixel in the brighter areas, but simultaneously reducing the levels of pixel in the areas of lower brightness. The result is an image with a real increase of contrast, without the necessity to apply an excessive increase of pixel values in a global way over all image, or only increasing the zone of greater brilliance, causing a contrast saturation, so is the case of the traditional methods. An image to be displayed in a television requires a contrast enhancement. This is one of the noticeable differences between television images with respect to the cinema, where the environment is dark and controlled. For applications in television, the consumers are generally in environments with high level of environmental light and it is not a controlled environment, therefore we assure a better perception of contrast if we reduce the pixel values in the zones of lower value and compensating in the zones of greater pixel values. The average level of pixel values is preserved.

Fig. 6. The horizontal bar is perceived with different gray level at both ends because they are inmerse in a contrasted surrounding.

3.3 Brightness Perception

Sudies on brightness and its perception, tell us that the brilliance is the magnitude of the subjective sensation that is produced by visible light. Being the radiation easily measurable, the brilliance cannot exactly be quantified. Brilliance commonly is approximated as the logarithm of the luminance, or the luminance elevated to the power of 1/2 to 1/3 depending on the context. The relation luminance-brilliance depends on the level of adaptation to the environmental light, Stevens [14] tell us that as the adaptation luminance *La* falls, the human sensitivity to the contrast also falls. The Human visual system presents more sensitivity to the luminance changes in the darkest regions of an image. Two images with different luminance levels can have the same values of brightness, and can appear to the human visual system like being identical. In its analysis, Stevens describes the brightness according to the following relation:

$$B = \lambda(\frac{L}{La})^{\sigma} \tag{6}$$

$\sigma = 0.4 \quad log_{10}(La) + 2.92$
$\lambda = 10^{2.0208}(La)^{0.336}$

In equation (7), B describes the brightness in Brils units, L is the value of the original luminance (Lamberts = 3.183 cd/m2), *La* describes the adaptation of the eye to the luminance.

According to [14], a Bril is equal to the brightness sensation that is induced in a completely adapted eye to the dark by a brief exhibition to a luminance of 1 micro-Lambert. In relation to the visual perception of the human being on the absolute value of luminance, which we can judge is the relation of luminance, the brightness. We are more sensitive at the low levels of illumination, i.e. where the human visual system presents higher sensitivity to the changes of luminance, in the darkest regions of an image. It is one of the main reasons why is so important a good contrast ratio and therefore, the correct representation of low levels of gray in an image.

3.4 To Preserve Level of Original Brightness

We can find a large number of traditional methods for contrast enhancement, many of them based on equalization by histogram or the manipulation of transformation functions based on mathematical. We can find examples of these methods in [9, 11, 12, 13], these methods are of the global type and they do not provide a local gray scale compensation, consequently the result is brightness over-enhancement, causing distortions due to the saturation in gray levels, especially in the most brighter areas of the image. This type of methods could be used in applications with static images, like computer graphics or in portable devices with small screens of LCD, where the quality details in images are not absolutely perceivable because of the small displays and the nature in the content of the images. The method of contrast enhancement must assure and maintain the objective quality of the images. It is understood that the objective quality of an image is the first step in the development of the design of the diverse stages of a television system, and has the intention

to reproduce an image with the fidelity as nearest possible as the original image, i.e. an image without enhancement and without distortions caused by the diverse stages and systems of processing of the television. Examples of this metric are the signal to noise ratio (S/N), frequency response, correct luminance representation, etc. Any method of contrast enhancement is not apt to be applied in television if it is not able firstly to maintain the objective quality in images. Once objective quality is achieved, the following step is the subjective quality of images. At this moment the type and degree of enhancement is determined, where the main factor is the intention of the video designer based on aspects like technological advantages or obstacles, consumer preferences , market tendencies or exigencies, etc. The metric used in the subjective quality of the image is based on trained experts to judge the quality in the reproduction of the video images. Generally the methods used for quality of image in the manufacture process of televisions, are methods more rigorous than those established in standards like the recommended in norms like ITU-T BT.500. The main idea is based on sequences of video especially designed for such intention, where the opinion of the group of experts is recorded during the evaluation of each of these special video sequences. An important point is that at any moment a simultaneous comparison is made of the images with more than a reference, therefore is possible at any moment to make evaluation under subjective comparison with references of other models or competitors televisions brands already in the market. In the method of contrast enhancement presented in this work, special consideration is taken about preserving of original brightness in the image, assuring in this way, always a correct representation of gray scale in the screen.

3.5 Description of the Proposed Method

The method initiates with the statistical information by histogram of pixel values of present image frame, although this information can be based on previous frame, according to the capacity of the video processing system. Considering that it uses the present information, the method is described in the following steps:

1. A set of representative images in brightness and contrast will be selected.
2. For each image the probability density function *Pr (rk)* of the gray levels will be obtained.
3. Using *Pr (rk)*, it is obtained the average μ, and standard deviation σ.
4. The point in the curve corresponding to the average (μ) is fixed, setting the inflexion point in the curve.
5. Using subjective methods for evaluation of image quality, the transformation function of each image is obtained and the required compensation is done emphasizing the contrast enhancement in the lower and higher pixel values respect to the average μ, corresponding to the darkest and brighter areas in the image, obtaining in this way the values of gain α and β. This can be appreciated in Figure 6 .
6. Using the resulting transformation function, the data set of training for a neural network will be obtained.

7. The architecture of a neural network is proposed and trained.
8. Test with different images is performed and the results are evaluated to determine if they are adapted.

In Figure 7, notice that the values of smaller pixels than the average will have a transformation similar to a concave curve, where α corresponds to a gain whose magnitude is determined by the statistic of the image frame. The pixel values higher than the average will have a transformation similar to a convex curve with a magnitude β. In the proposed method the transformation function is a multi-layer neural network. At the inputs are the average μ, the standard deviation σ, the parameters α, β, and the set of values for the transformation curve.

$$s_i = T_{NN}(r_i) \tag{7}$$

where r_i are the input pixel values in the interval $0 \leq r_i \leq 255$, s_i is the corresponding level of transformed pixel. The average and the standard deviation are indicators of the level of brightness in the image of concentration of gray levels, that are the resistance; image and the degree of gray levels distribution, the contrast.

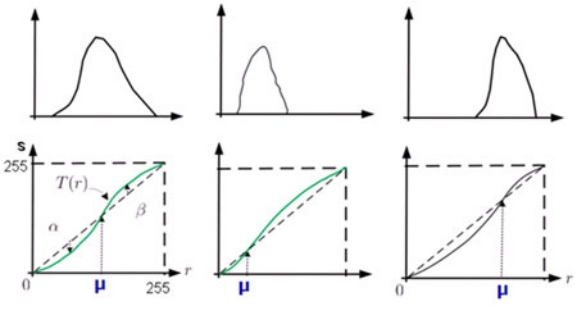


Fig. 7. Monotonically increasing transformation functions that produce an s value for each pixel of r value in the original image.

3.6 Proposed Neural Network and Its Training

The design of the network consists of the selection of the model of the network, the variables to incorporate and the preprocessing of the information for the training set. The proposed neural network has three layers, monolithic with architecture 5-10-1. At the input pattern, five data appear in the following order: $\alpha, \beta, \sigma, \mu, r_i$, connected to the five input neurons, and in the output we have the data s_i, see Figure 8.

In all the neurons, the implemented activation functions were logarithmic-sigmoid. A training was done using the method of optimization of Levenberg-Marquard, which is a standard technique for nonlinear minimums square problems, being obtained the training error goal of 10^{-6}. The training time is variable, being 20 seconds typical time, the structure of the neural network is parallel, optimal for real-time applications. Following the method, it was trained the neural network with the proposed architecture. It was necessary to use at least 20 images of typical

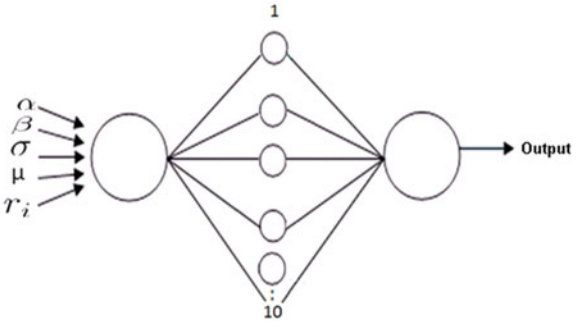


Fig. 8. Proposed Neural Network.

scenes, to obtain an error of training in the order of 10^{-6} and one capacity of generalization adapted for different images. Different methods from optimization were tried on, finally we chose the one of gradients conjugated for all the layers of the network. Previously it was mentioned that the training consists of providing to the input, statistics of typical images, and the values of gain of each typical image. The Neural Network (NN) is trained to associate input-output patterns. When an input pattern occurs that does not have output associated with this, then the network gives an output that corresponds to the closest pattern of inputs, and corresponding to the closest training patterns given. The specifications of the input-output patterns forms a set of examples. The learning process consists of finding the weights that codify the knowledge. A learning rule makes vary the value of the weights of a network until these adopt a constant value. When this happens, the network "already has learned". Combined to it, for an implementation in industrial process, the transformation of contrast enhancement is defined with a wider sample in the process of pre-manufacture, according to the opinion of a group of experts in video quality, by means of a subjective judgment with own standards, generally more rigorous than the established ones in international norms like norm ITU500. Images and functions with representative parameters as input in the process of training of the NN will be used. To observe that the transformation function where upon the process begins is linear, considering the average value of the histogram, the inflexion point μ is obtained. From it, for training aims, a visual analysis of the image is realized and the straight line is deformed obtaining a nonlinear function of transformation $T\ (r)$, from where the gain parameters α, β are obtained which corresponds to the maximum curve deformation referring to its concavity and convexity. The training is part of the process of pre-manufactures, once obtained the parameters of the NN. The method is apt for applications in the industry in real time, the hardware implementation of the neural network is considering parallel implementations in highly dedicated systems, based on FPGA technology, with clock cycles above 600 MHz with independent modules for processing of video, sufficient capacity for a high resolution image 1080P, which requires 60 million pixel operations per second.

4 Description of the Evaluation Method and Analysis of Results

In this section the analysis of comparative results are presented applying the concepts of the method described in this work. Representative images in their original condition are presented. Each image is statistically analyzed by histogram, with this information and using subjective techniques for image quality, the contrast in every image is modified obtaining in this way the processed image and its corresponding curve of transformation. Each of these curves of transformation, has its input values r_i and its corresponding output s_i, as well as the parameters α, β, μ, σ which are used as training patterns during the training of the neural network. Finally the curves of transformation from the trained neural network are obtained and they are compared with the curves obtained by subjective method of quality of image. The subjective curves of transformation are obtained with the aid of an application program which allows the direct manipulation of the image starting for a linear straight line, then it is manipulated subjectively modifying the pixel values in the magnitudes and in the required zones. The neural network was implemented in a simulation environment using Matlab.

4.1 Evaluation Method Using Test Pattern Image

As a first step, the effectiveness of the method was evaluated using a special test pattern image made of 9 levels of Gray scale, as shown in Figure 9 top, as input. By histogram the statistic of this initial image is obtained, $\mu = 127$, $\sigma = 66$, which indicates an image with an average level of brightness. Using the subjective method of image evaluation, the contrast of the input image is modified, where the levels of pixel below the average μ, were decremented in value with a maximum gain of $\alpha = 22$. The pixel values above of the average were increased with a maximum gain $\beta = 22$. The average level in the transformed image is preserved as can be confirmed comparing both images in the center zone and in the initial and final pixel average value (μ), which remains constant at $\mu = 127$. In Figure 10 is shown the curve of transformation obtained by subjective method of image quality evaluation. The analysis by histogram of the transformed image at the bottom of Figure 9 (output image), indicates an increase in the distribution of pixel values as it is demonstrated by the comparison with the statistic of the transformed image, $\sigma = 85$. The standard deviation σ increases from 66 to 85. By a visual comparison of the input and output images in Figure 9, we can confirm the effect of the method intention, preserving the brightness while pixel values below the inflexion point μ are decreased, and the opposite for pixel values above this inflexion point.

The intention of this experiment is to confirm the effectiveness of this method in a test pattern image, where the transformation effects are easily measured and perceivable by visual comparison between original and transformed image.

Table 1 corresponds to the transformation in Figure 10.

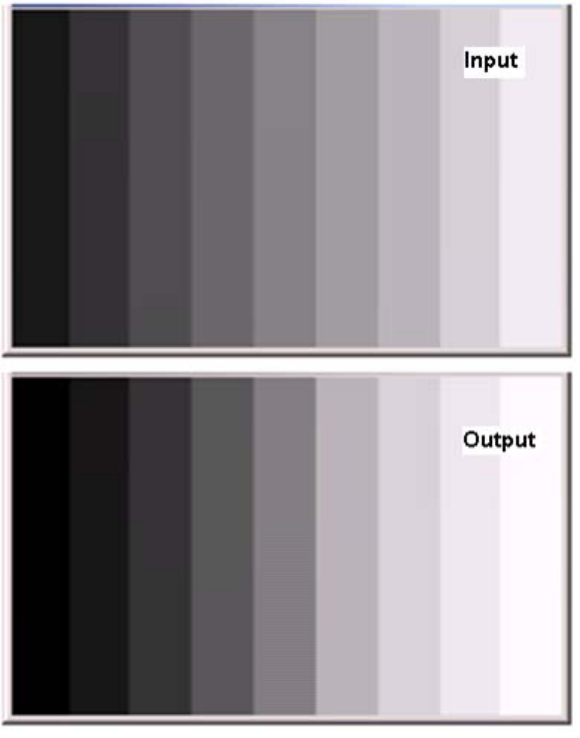


Fig. 9. Top image is original and the corresponding processed image at bottom. The difference between both is observed in the areas of lower and higher illumination, preserving the same level of gray in the average zone.

Table 1. Statistical values and parameters of α and β from the presented images.

Image, and its transformation func.	**Mean** (μ)	**Standard Dev.**	**Alfa** (α)	**Beta** (β)
Figure 9 input	127	66		
Figure 9 output	127	85	22	22

4.2 Evaluation Method Using Real Images

In this section we describe the evaluation and results of the experiment achieved using real images.

In Figure 11, there is an image in its original condition, while Figure 12 shows the processed image using the concepts described by this method.

Making a comparison, it can be appreciated that the lower illumination zones in the image were decremented in pixel values as it is appreciated in the table at front, the floor, and in general the dark areas at the background. The higher illumination areas were incremented in pixel values as observed in the white dress, the top of

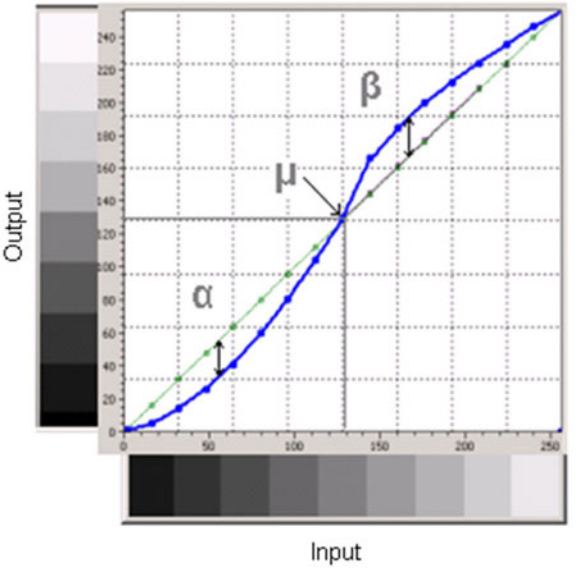


Fig. 10. Shows the curve of transformation obtained for the processed image, as well as the parameters α, β and μ.

Fig. 11. Image in their original condition

the table, the window frames in doors and windows. In general, processed image has more clarity and depth specially in the background details. Figure 13 shows the transformation curve obtained by subjective image quality in the contrast enhancement in order to obtain the image in Figure 12.

In Figure 14 is the histogram of the original image from Figure 11, the histogram of Figure 15 corresponds to the processed image. Comparing both histograms, in Figure 14, it is observed a wider distribution of pixel values, as it is appreciated in

Fig. 12. Image after processing, contrast expansion is observed in the low an high ilumination areas, pixel values in average level areas are preserved.

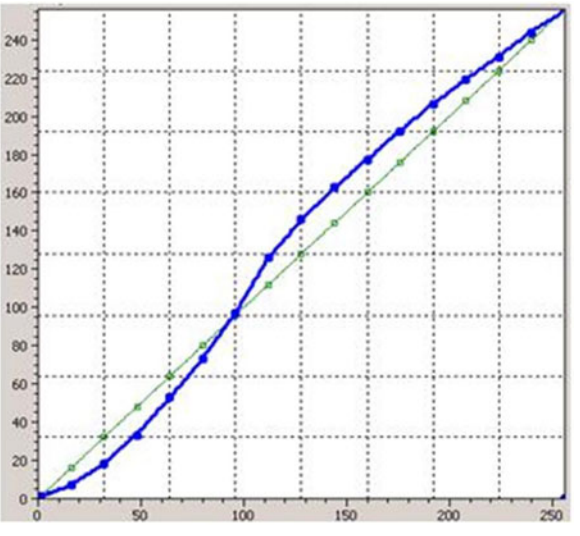


Fig. 13. Transformation curve corresponding to obtain the image in Figure 12.

the zones near pixel values below 20 and above 200. In general, the result is a wider distribution of pixel values.

Table 2 shows the statistical values before and after image transformation of Figures 11 and 12.

As it is appreciated from the images and the results in Table 2, the transformation function in the output image has the effect of contrast expansion, a wider gray scales values are obtained as shown in the increment in standard deviation value from initial value $\sigma = 56$ to final value of $\sigma = 70$.

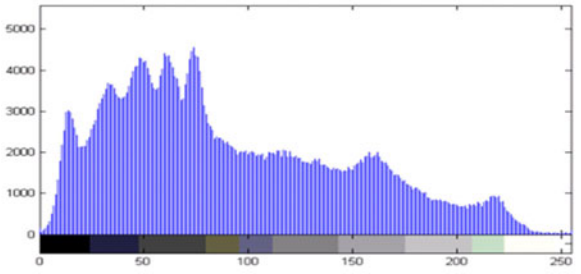


Fig. 14. Histogram corresponding to the original image in Figure 11.

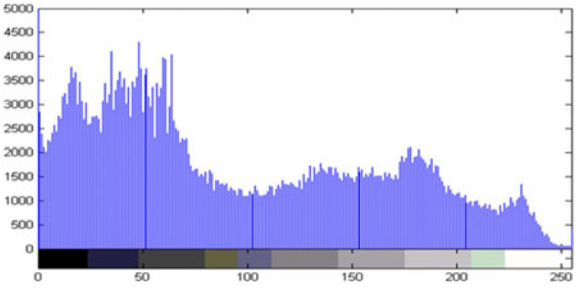


Fig. 15. Histogram corresponding to the image processed shown in Figure 12.

Table 2. Statistical values and parameters α and β of the presented images.

Image, and its transformation func.	Mean (μ)	Standard Dev.	Alfa (α)	Beta (β)
Figure 11	92	56		
Figure 12	92	70	15	17

The average pixels value remains in its initial value $\mu = 92$ and it corresponds to the inflexion point in the transformation curve as appreciated in Figure 13. The pixel values in the dark areas were decremented in value with a gain $\beta = 17$, obtaining in this way an increase in contrast and in general an overall improvement of the image. The results from previous experiments, show that the concepts from the method are effective in order to achieve the contrast enhancement in the image.

5 Conclusions

An intelligent method for contrast enhancement in digital video for applications in TFT-LCD was presented. The intention in the design of the method, is to obtain a general model of transfer function that compensates the decrease of the dynamic range in the representation of the gray scale in images, as a consequence of the light

leakage from the backlight in LCD-TFT technology. The method is based on the generation of transfer functions by the training of a neural network using representative images for several degrees in brightness and contrast. An evaluation of the method was done using simulation and techniques of subjective evaluation of quality images in a non controlled light environment, as it is the case of a typical user of consumer electronics. As a result of the evaluation an analysis of the experiment was performed, confirming the effectiveness of the method.

The method is adaptive and compatible with the dynamic control system of the LCD-TFT light source, the backlight. A flexible method that can easily be adapted for the handling of different video modes (Lived, Cinema, Standard, etc.).

Acknowledgement. The authors thank Comisión de Operación y Fomento de Actividades Académicas del I.P.N., Instituto Tecnológico de Tijuana for supporting our research activities, and CONACYT.

References

1. ANSI IT7.215-1992: Data Projection Equipment and Large Screen Data Displays – Test Methods and Performance Characteristics
2. Poynton, C.A.: Gamma and its Disguises: The Nonlinear Mappings of Intensity in Perception. CRTs, Film and Video SMPTE Journal 102(12), 1099–1108 (1993)
3. Land, E.H., McCann, J.J.,, J.: Lightness and retinex theory. Opt. Soc. Am. In: B.K. (ed.) Comp.Graph.Image Proc., vol. 61(1) (1971)
4. Gonzalez, R.C.: Digital Image Processing / Richard E. Woods. Interscience, NY (2001)
5. Blackwell, H.R.: Contrast thresholds of human eye. Journal of Optical Society of America 11(36) (November 1946)
6. ITU-R.500 Methodology for the subjective assessment of the quality of television
7. ITU-R BT.601
8. ITU-T Rec J.247. Objective perceptual multimedia video quality measurement in the presence of a full reference
9. Kim, J.-Y., Kim, L.-S., Hwang, S.-H.: An advanced contrast enhancement using partially overlappedsub-block histogram equalization Circuits and Systems for video T. IEEE Transactions on Technology 11(4), 475–484 (2001)
10. Chen, M.-T., Lin, C.-C.: Comparison of TFT-LCD and CRT on visual recognition and subjective preference. International Journal of Industrial Ergonomics 34(3), 167–174 (2004)
11. Wang, Q., Ward, R.K.: Fast Image/Video Contrast Enhancement Based on Weighted Thresholded Histogram Equalization. IEEE Transactions on Consumer Electronics 53(2) (May 2007)
12. Kim, S.-Y., Han, D., Choi, S.-J., Park, J.-S., DTV Res. Labs., LG Electron. Inc., Seoul: Image contrast enhancement based on the piecewise-linear approximation of CDF. IEEE, Los Alamitos (2002) ISSN: 0098-3063
13. Chen, S.-D., Ramli, A.R.: Contrast enhancement using recursive mean-separate histogram equalization for scalable brightness preservation. IEEE Transactions on Consumer Electronics 49, 1301–1309 (2003)

14. Stevens, S.S., Stevens, J.C.: Brightness Function: Effects on Adaptation. Journal of Optical Society of America 53(3), 375–385 (1963)
15. Cornsweet, T.N.: Visual Perception. Academic Press, New York (1970)
16. Wiegand, T.E., Hood, D.C., Graham, N.: Testing a computational model of light-adaptation dynamics. Vision Research 35(21), 3037–3051 (1995)
17. Yu, Z.C.B.: A fast and adaptive method for image contrast enhancement. In: Image Processing, ICIP 2004. 2004 International Conference Publication, vol. 2, pp. 1001–1004 (2004)

Method for Obstacle Detection and Map Reconfiguration in Wheeled Mobile Robotics

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

[1] Centro de Investigación y Desarrollo de Tecnología Digital (CITEDI- IPN), Av. del Parque No.1310, Mesa de Otay, 22510, Tijuana, B.C., México
o.montiel@ieee.org, r.sepulveda@ieee.org, gonzalez@citedi.mx

[2] Division of Graduate Studies and Research, Calzada Tecnológico S/N, Tijuana, B.C., México
epmelin@hafsamx.org

Abstract. This paper describes a method for obstacle detection, map building and map reconfiguration for autonomous mobile robotic application, this proposal is based on a stereoscopic vision system and epipolar geometry to achieve obstacle detection, map building and map reconfiguration is presented. A tridimensional projection of every scene in front of the mobile robot is estimated based on geometrical relationship between the environment world coordinates (x,y,z) and their projections through the 2D images captured by the on board stereoscopic vision system of the mobile robot. Depth measure for obstacle distance estimation is calculated by epipolar geometry. The proposal for implementation stereoscopic photometric correspondence problem, is solved in an efficient way with the implementation of the adaptive window for candidate matching (AWCM) sensor, motion control and algorithms are proposed to be implemented in FPGA technology to obtain high performance overall processing.

1 Introduction

In general, global planning methods complemented with local methods are used for indoor missions since the environments are known or partially known; whereas, for outdoor missions local planning methods are more suitable, becoming global planning methods, a complement because of the scant information of the environment.

In previous work, we developed a path planning method for wheeled MR navigation using a novel proposal of ant colony optimization named SACOdm (Simple Ant Colony Optimization with distance (d) optimization, and memory (m) capability), considering obstacle avoidance into a dynamic environment [8]. In order to evaluate the algorithm we used virtual obstacle generation, being indispensable for real world application to have a way of sensing the environment.

This work presents a proposal to achieve stereoscopic vision for MR application, and its future development and implementation into VLSI technology to obtain high performance computation to improve local and global planning, obtaining a faster

P. Melin et al. (Eds.): Soft Comp. for Recogn. Based on Biometrics, SCI 312, pp. 423–441.
springerlink.com

navigation by means of reducing idle times due to slow computations. Navigation using ant colony environment is based on map building and map reconfiguration; in this model, every ant is a virtual mobile robot (MR). The MR system, composed by the MR and the global planner in the main computer, has the task to construct the map based on a representation of the environment scene, avoiding the use of landmarks to make the system more versatile. The MR stereo vision transforms the visual information of two 2D images of the same scene environment into depth measure data. Hence, the MR sends this data via RF to the global planner in the main computer; this data is a 3D representation of the MR scene environment and its local position and orientation. By this way, the optimal path in the environment is constantly updated by the global planner.

The MR stereo vision has the advantage with respect to other navigation techniques, that depth can be inferred with no prior knowledge of the observed scene, in particular the scene may contain unknown moving objects and not only motionless background elements.

The proposed method has some advantages over existing methods, for example it does not need camera calibration for depth (distance) estimation measurements; an improved efficiency in the stereoscopic correspondence for block matching; adaptive candidate matching window concept is introduced for stereoscopic correspondence for block matching resulting in improved efficiency by reducing calculation time, also improves matching accuracy assuring corresponding process only in the areas where there are vertical or corners arranged pixels corresponding to the obstacles selected features. The calculation process is reduced in average 40% corresponding to the surface ground image content which is previously extracted from every image. The areas between edges in the obstacles itself are also excluded from matching process. This is an additional increase in the method efficiency by reducing calculation for matching process. This feature provides an efficient implementation of a vision module for obstacle detection, for map building and dynamic map reconfiguration as an extension research of the ant colony environment model described in a previous work [8].

This work is organized as follows: In Section 2 the general system is described. Section 3 is dedicated to give a description of the process to extract the surface ground and obstacle edge detection using luminance components, and the Hue to obtain the ground surface; moreover, some comments about implementation of the vision module into an FPGA. In Section 4 important concepts about stereoscopic vision are given. In Section 5 is explained how the modification of the road map is achieved. Finally, in Section 6 are the conclusions.

2 General System Description

The two main components of the system architecture are the computer, and the MR. The computer contains the global planner based on the SACOdm algorithm, and the communication system. The MR is a three wheels system with frontal differential tracking, it has five main subsystems: The stereoscopic vision which includes parallel arrange dedicated purpose video decoders controlled via IIC by the FPGA; the

Spartan-3 FPGA controller board that contains an embedded Microblaze microcontroller, as well as the motors and tracking controllers that were coded in VHDL hardware description language software; the power module consists of a high capacity group of rechargeable batteries, two H-bridges motor drivers, and the two Pittman DC geared-motor model GM9236S025-R1; the communication system based on the XbeePro RF, integrated WiFi communication module [17]; A high accuracy GPS module with 1 cm of resolution, 0.05% or accuracy, such as the VBOX 3i from Racelogic [15], or similar; an electromagnetic custom made compass IIC bus compatible, based on the LIS3LV02DL integrated circuit from STMicroelectronics.

The communication between the MR and the computer is achieved using the XBeePro RF Modules that meet the IEEE 802.15.4 standards, the modules operate within the ISM (Industrial Scientific and Medical) 2.4 GHz frequency band.

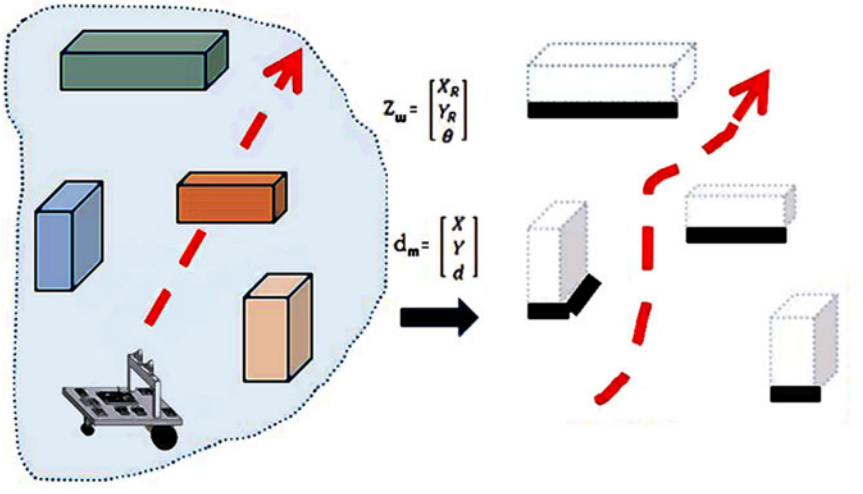


Fig. 1. Basic idea of the method for obstacle detection and map reconfiguration based on a 3D representation of every scene.

Figure 1 shows the basic idea of the method, the mobile robot vision system captures and process stereoscopic images, it obtains a three dimensional representation of every scene and sends this information consisting of a disparity map, its position and orientation to the global planner, which is a computer outside the environment that receives the data for the representation of the map building and map reconfiguration.

3 Stereoscopic Vision and Obstacles Detection

For the environment map construction and reconfiguration, the MR makes an inference of the three dimensional structure of a scene from its two dimensional 2D

projections. The 3D description of the scene is obtained from different viewpoints. With this 3D description we are able to recreate the environment map for use in robot navigation, see Figure 1.

In general, in any stereoscopic vision system after the initial camera calibration, correspondence is found among a set of points in the multiple images by using a feature based approach. Disparity computation for the matched points is then performed. Establishing correspondences between point locations in images acquired from multiple views (matching) is one of the key tasks in the scene reconstruction based on stereoscopic image analysis. This feature based approach involves detecting the feature points and tracking their positions in multiple views of the scene. In [4], it is presented a review of the problem where the developments in establishing stereoscopic correspondence for the extraction of the 3D structure is discussed, also a few well-known algorithms representing widely different approaches were presented, the focus of the review was stereoscopic matching.

For map construction or reconfiguration of the MR obstacles environment there is not necessary to reconstruct an exact scene of the environment. There are other works in the same line, in [3] is presented an approach that integrates appearance models and stereoscopic vision for decision people tracking in domestic environments. In [9] the author used a vision system for obstacle detection and avoidance, it was proposed a method to integrate the decisions by using potential field theory [11] with fuzzy logic variables. It was used Hue, Saturation, and Intensity (HSI) hue color since it is invariant to the light. In [18] was presented an omnidirectional vision camera system that produces spherical field of view of an environment, the continuation of this work was presented in [19] where the authors explained several important issues to consider for using fisheye lens in omnidirectional vision, some of them are the lens camera calibration, rectification of the lens distortion, the use of a particle filter for tracking, as well as the algorithms and the hardware configuration that they implemented.

The navigation task is achieveed using the relative depth representation of the obstacles based on stereoscopic vision and the epipolar geometry. The map represents the status at the time of drawing the map, not necessarily consistent with the actual status of the environment at the time of using the map. Mapping is the problem of integrating the information gathered in this case by the MR sensors into a complex model and depicting with a given representation. Stereo images obtained from the environment are supplied to the MR, by applying a disparity algorithm on stereo image pairs, depth map for the current view is obtained. A cognitive map of the environment is updated gradually with the depth information extracted while the MR is exploring the environment. The MR explores its environment using the current views, if an obstacle in its path is observed, the information of the target obstacles in the path will be send to the global planner in the main computer. After each movement of the MR in the environment, stereo images are obtained and processed in order to extract depth information. For this purpose, obstacle's feature points, which are obstacle edges are extracted from the images. Corresponding pairs are found by matching the edge points, i.e., pixel's features which have similar vertical orientation. After performing the stereo epipolar geometry calculation, depth

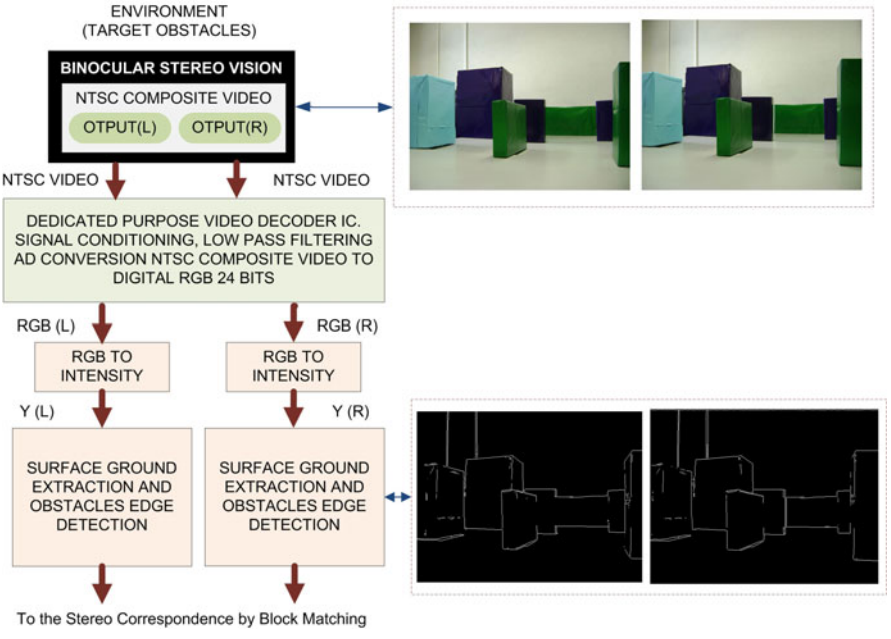


Fig. 2. Process for surface ground extraction, and obstacles edge detection using luminance component.

for the current view is extracted. By knowing the camera parameters, position, and orientation, the map can be updated with the current depth information.

3.1 Surface Ground Extraction and Obstacle Detection Using Luminance and Hue

The vision based obstacle detection module classifies each individual image pixel as belonging either to an obstacle or the ground. An appearance based method is used for surface ground classification and extraction from the MR vision module captured images, see Figure 2. Any pixel that differs in appearance from the ground is classified as an obstacle. After the surface ground extraction the remaining image content are only obstacles. A combination of pixel appearance and feature base method is used for individual obstacle detection and edge extraction. Obstacles edges are more suitable for stereo correspondence block matching in order to determine the disparity between left and right images. For the ground surface extraction purpose two assumptions were established, that are reasonable for a variety of indoor and outdoor environments:

1. The ground is relatively flat.
2. Obstacles differ in color appearance from the ground. This difference is reasonable and can be subjectively measured as Just Noticeably Difference (JND), which is reasonable for a real environment.

These assumptions allow us to distinguish obstacles from the ground and to estimate the distances between detected obstacles from the vision based system. The classification of a pixel as representing an obstacle or the surface ground can be based on local visual attributes: Intensity, hue, edges, and corners. Selected attributes must provide information so that the system performs reliably in a variety of environments. Selected attributes should also require low computation time so that real time system performance can be achieved. The less computationally expensive the attribute, the higher the obstacle detection update rate, and the faster an MR can travel safely.

For appearance classification we used Hue or luminance as a primary attribute for ground surface detection and extraction. Hue provides more stable information than color or luminance based in pixel gray level. Color saturation and luminance perceived from an object is affected by changes in incident and reflected lightness. Also compared to texture, Hue is more local attribute and faster to calculate. In general, Hue is one of the main properties of a color, defined as the degree of perceived stimulus described as Red, Green, Blue. When a pixel is classified as an obstacle, its distance from the MR stereo vision cameras system is estimated using the following steps:

1. Color image from each video camera is converted from NTSC composite video to RGB 24 bits color space.
2. A typical ground area in front of the MR is used as a reference. The Hue attributes from the pixels inside this area are histogrammed in order to determine its Hue attribute statistics.
3. Surface ground is extracted from the scene captured by the MR stereo vision by means of a comparison against the reference from step 2, and based on Hue attribute. Hue limits are based in JND units.
4. Remaining content in images are only obstacles. Edges are extracted from individual obstacles based on feature and appearance pixel's attributes.
5. Correspondence for block matching is established in pixels from the obstacle vertical edges.
6. Disparity map is obtained from the sum of absolute differences (SAD) correlation method.

In this case we use Luminance Canny based method for obstacles edge detection and surface ground extraction, because of the square shape of the obstacles and the pixel appearance which is this case, in a histogram follows a normal distribution. By using this method, surface ground extraction and obstacles edge detection is achieved in the same process.

3.2 Stereoscopic Vision System Module and FPGA Implementation

When a robot has to react immediately to real-world events detected by a vision system, high speed processing is required. Vision is part of the MR control loop during navigation. Sensors and processing system should ideally respond within one robot control cycle in order to not limit their MR dynamic [13]. An MR vision system

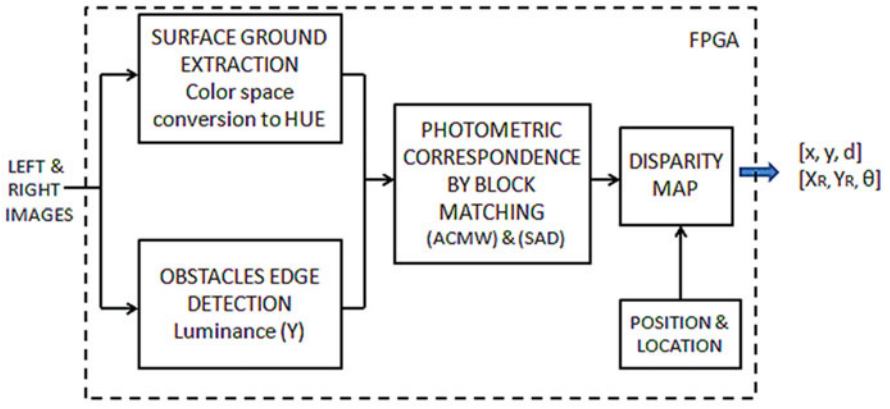


Fig. 3. Block diagram of the stereoscopic processing in FPGA

equipped, requires high computational power and data throughput which computation time often exceed their abilities to properly react. In the ant colony environment model, every ant is a virtual mobile robot (MR) full equipped, trying to find the optimal route, eventually, weather there exist, it will be obtained. Of course, the ACO based planner will give the best route found, and the real ant, the MR, which is equipped with the vision system on board, will update the global map in the planner. There are many tasks to do at the same time, however, a good feature of using FPGAs is that they allow concurrently implementation of the different tasks, this is a desirable quality for processing high speed vision. High parallelism is comprised with high use of the FPGA resources; so a balance between parallelization of task, and serial execution of some of them will depend on the specific necessities.

The vision system proposal consists of stereoscopic vision module implemented in VHDL and C codes operating in a Xilinx based FPGA, hence a balanced used of resources was used, Figure 3 shows a block diagram of the stereoscopic processing in FPGA. Video information is processed in a stereo vision system and video interface. The NTSC composite video signals from each camera after properly low pass filtering and level conditioning needs to be converted to RGB 24 bits color space by a state of the art video interface system HDTV capable. The rest of the video stage can be programmed in C for the Microblaze system embedded into the FPGA. Other subsistems, such as the motion control block needs to work concurently to the video system.

4 Design of the Stereoscopic Vision Module

The two stereo cameras parallel aligned, capture images of the same obstacle from different positions. The 2D images on the plane of projection represent the object from camera view. These two images contain the encrypted depth distance

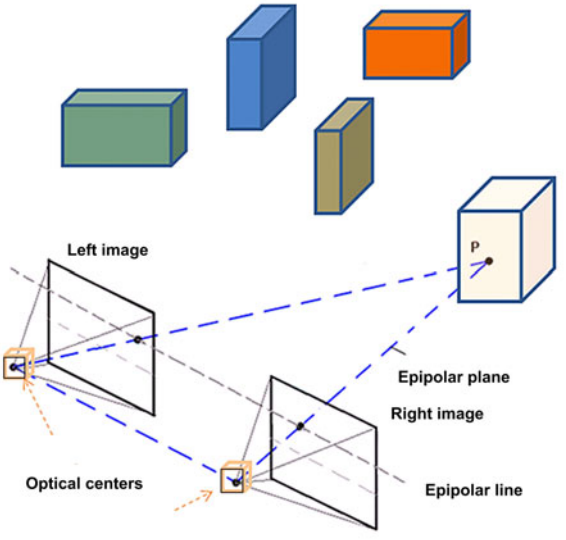


Fig. 4. Obstacles and stereoscopic vision. Projection of one point into left and right images from parallel arrange video cameras.

information. This depth distance information can be used for a 3D representation in order to build a map.

The MR using his side by side left and right cameras see the scene environment from different positions in a similar way as human eyes see Figure 4. The FPGA based processing system finds corresponding points in the two images and compares them in a correspondence matching process. Images are compared by shifting a small pixels block "window". The result is a comparison of the two images together over top of each other to find the pixels of the obstacle that best match. The shifted amount between the same pixel in the two images is called disparity, which is related to the obstacle depth distance. The higher disparity means that the obstacle containing that pixel is closer to the cameras. The less disparity means the object is far from the cameras, if the object is very far away, the disparity is zero that means the object on the left image is the same pixel location on the right image.

4.1 Depth Measure from Stereo Image

In order to estimate the depth measure of the obstacles in the scene first step is to determine the points of interest for correspondence matching between the two images. This corresponding points are selected based on the obstacle edge feature. Then calculate the depth distance based on the shifting "disparity". The disparity is calculated based on the amount of pixel's shifting in a particular corresponding point. There are stereo image constraints to be assume for solving the correspondence problem:

1. Uniqueness. Each point has at most one match in the other image.
2. Similarity. Each intensity color area matches a similar intensity color area in the other image.
3. Ordering. The order of points in two images is usually the same.
4. Continuity. Disparity changes vary slowly across a surface, except at depth edges.
5. Epipolar constraint. Given a point in the image, the matching point in the other image must lie along a single line.

4.1.1 Photometric Correspondence by Block Matching

The map for representation the ant colony environment, is estimated based on geometrical relationship between the world points and their projections on the 2D images captured by the MR stereo visual system equipped. Obstacles deep measure is calculated by epipolar geometry. To do this the problem of estimating corresponding points in one image respect to the other image is the central problem [6], it is a stereo correspondence problem and usually solved assuming that the cameras are fixed and their relative positions are known [16]. The stereo images has to be rectified in order to align the epipolar lines to the line joining the optical centers of the stereo cameras. Before solving the correspondence problem [10]. In order to avoid image rectification and to simplify camera calibration and stereo correspondence matching process, the stereo cameras are arranged in a parallel configuration [2]. The correspondence problem concerns the matching of pixels in two images such that the matched points are the projections of the same point on the same horizontal line in the scene. The stereo corresponding searching process is confined to one dimension along the horizontal axis. Stereo matching is used to generate the disparity or depth maps, generally, depth information is obtained from two broad categories of stereo matching algorithms [14] global and local. Global based algorithms, for example, those described in [5] produce accurate disparity maps but involve high computational costs. On the other hand, local algorithms [1] [7], are computationally efficient but do not produce results as accurately as global algorithms. We used a local or area based window correlation matching algorithm, due to their computational efficiency as compared to global algorithms. The construction of stereo images allows for a disparity in only the horizontal direction, there is no disparity in the Y image coordinates. This is a property that can also be achieved by precise alignment of the stereo cameras by means of epipolar geometry.

The disparity is usually computed as a shift to the left of an image feature when viewed in the right image. It is calculated by determining a measure of absolute differences or SAD used to calculate disparities at each pixel in the right image. For every point of interest captured in the environment scene, a matching reference window is defined by 8×8 pixels and also a searching area of 80×80 pixels around the reference matching window, see Figure 5. This reference window from left image is matched into a corresponding same size window in the right image. This 8×8 reference window from left image is shifted over the searching area of 80×80 pixels in the right image. At every shift the absolute difference of comparing pixel attributes is computed and saved. After this SAD "match strength" has been computed for all

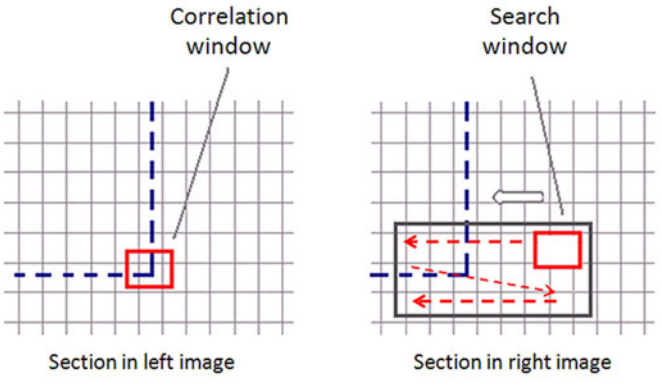


Fig. 5. Reference and searching windows in the stereo correspondence by block matching process using SAD.

valid disparities, the disparity that yields the lowest SAD is determined to be the disparity at that location in the right image.

$$SAD(x,y,d) = \sum_{i,j=n}^{n} |L(x+j,y+i) - R(x+j+d,y+i)| \tag{1}$$

Where (x,y,d) is the value for the disparity at point $L(x,y)$ of the left image and the corresponding point $R(x+d,y)$ of the right image. $L(x,y)$ and $R(x,y)$ are the pixel attribute at left and right image respectably. The disparity at certain point is the value of d that minimizes SAD.

The computational time can be reduced by reducing the number of valid disparity search at every shift, limiting the disparity to a maximum value, where d is the disparity based on a maximum value for disparity $dmax$. Another way to reduce computation time is by reducing the searching area. The size of the optimal pixel searching area can be estimated based on the image noise, quality of pixel attributes, edge detection and estimated disparity in the points of interest in scene.

4.1.2 Constraints for Matching

Because of factors such as noise and occlusions, the appearances of the corresponding points will differ in the two images. For a particular feature in one image, there are usually several matching candidates in the other image. It is convenient to use additional constraints in order to obtain the correct match. Commonly used constraints are:

1. The matching points must lie on the corresponding epipolar lines of the two images.
2. Each point has at most, one match in the other image.

3. Each similarity intensity or color area matches a similar intensity or color area in the other image.
4. The disparity gradient should be within a certain limit. Due to occlusions in certain 3D surfaces, the appearances of the corresponding points will differ in the two images.

4.1.3 Improved Efficiency in the Corresponding Matching Process

The complexity may increase in the correspondence matching process in cases when the images are captured by the on board stereo vision system of the MR. The epipolar lines may vary from the horizontal axis because of vibrations due to variations in ground surface or by mechanical gears. In order to compensate this condition, the epipolar lines and selected matching points may provide some indication during the matching process concerning the spatial position of the stereo cameras.

In our application the epipolar lines are selected for robustness and efficiency by processing for matching only the image information represented by the edges of obstacles.

- Stereo correspondence matching is done only in the portions of the images where there are obstacles. The extracted portion from ground surface will be omitted from matching process. This reduces the calculation time. In every captured image, in average at least 40% of the image content is composed by ground surface.
- The points of interest used as candidates for matching are limited only to the vertical edges. After edge detection in both images, the remaining image data corresponds only to the obstacles edge intensities information.
- Adaptive Candidate Matching Window (ACMW). In order to detect the candidate points for matching and assure these points belong only to the vertical edges, a candidate matching window 4×4 all zeroes (0's) is defined. This non overlapping window is shifted horizontally in the left image, which is the reference image. At every shift this window is logical AND compared with the information in the image in that position. If the result contains at least one vertical line containing logical ones (1's). then a new window is selected, this is the matching window containing the image information in that location and it will be compared with the right image for SAD matching process as described previously in the process photometric correspondence for block matching.

The candidate matching window assures SAD matching processing only in the image containing the obstacle's vertical edges. By doing this, all image portions previously filled with logical 0's will be excluded from the correspondence matching process. These 0 filled image areas are the surface ground and obstacles portions inside its edges. SAD matching process computationally is more expensive than simple logical AND operations. Matching accuracy and calculation time optimization are the major advantages due to the novel concept of candidate matching window.

An Adaptive Window for Candidate Matching (AWCM) can be implemented in case of more dense environments and wider diversity in the shapes of the target obstacles. More features of interest can be added like corners, round, horizontal.

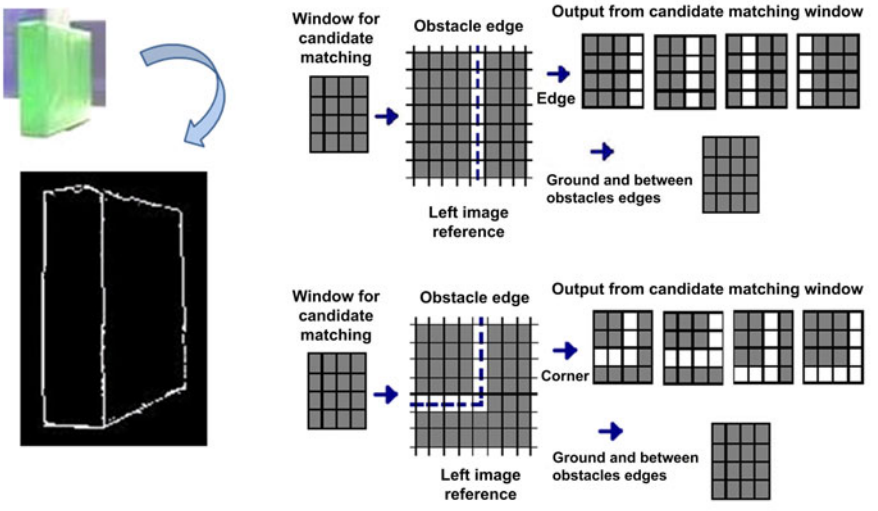


Fig. 6. Process of candidate matching window using edges and corners.

The method can be easily adapted combining edges and corners, Figure 6 shows an example of the process for Candidate Matching using corners as a feature of interest. In this case, overlapping window is more conveniently. The figure shows four outputs for right bottom corner, we can have 16 possibilities at the output in order to cover the four corner shapes.

We used a feature based correspondence matching method based on the obstacles detected edges. The metric used for estimation implemented is based on the algorithm sum absolute difference (SAD). Additional considerations were applied for efficiency and robustness like candidate matching window.

4.1.4 Epipolar Geometry

The epipolar geometry of the stereoscopic system is described in Figure 7. The stereoscopic system model shows two different perspective views of a point in the obstacle (P) from two identical cameras centers, which are separated only in X axis by a baseline distance. The points P_L and P_R in the image plane are the perspective projections of P in left and right view. The plane passing through the camera centers and the object point in the scene is called the epipolar plane. The intersection of the epipolar plane with the image plane is called epipolar line, the correspondences at point P_L and P_R must lie on the epipolar line.

The epipolar geometry is determined by the point correspondences. Selection and matching of the point features in the two views are the standard procedure for the depth recovery. The depth information can be evaluated by using the triangle similarity algorithms. The epipolar geometry is illustrated in Figure 8. In the illustration, the baseline distance (T) and the focus length (f) of both cameras are known. The perspective projection P_L and P_R on epipolar line is shift from their center by distance X_L and X_R respectively.

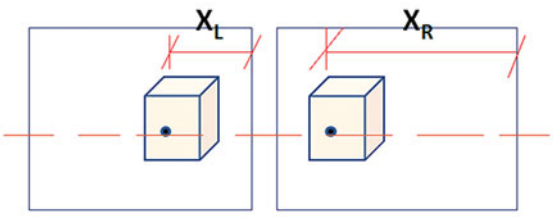


Fig. 7. Horizontal disparity as a result of the perspective projections of one point P in left and right views.

4.1.5 Triangulation for Depth Distance Estimation

One point in 3D space and its projections in two images always compose a triangle, the reconstruction problem, which is to estimate the position of a point in 3D space. The 3D position (X,Y,Z) of a point P, can be reconstructed from the perspective projection of P on the image planes of the cameras, once the relative position and orientation of the two cameras are known. We choose the 3D world reference system to be the left camera reference system. For non parallel coupled stereo cameras, the right camera is translated and rotated with respect from the left one, therefore six parameters describe this transformation. The simplest case arise when the optical axes of two cameras are parallel, and the translation of the right camera is only along the X axis.

From Figure 8 using the two triangles from left camera and right camera we can obtain the expression:

$$Z = f \cdot \frac{T}{d} \tag{2}$$

The reference system for the left camera reference system according to a 3D point $P = (X,Y,Z)$ is mapped by central projection into $P_L = (X_L, Y_L)$

and into $P_R = (X_R, Y_R)$; so, there is a translation in X axis due to baseline T

$$X = \frac{(X_R - T) \cdot Z}{f} \tag{3}$$

In case of origin at mid point between camera centers (cyclopean coordinates). Then, for our one camera model, the new reference system is translated between both camera centers,

$$X = \frac{(X_L - (T/2)) \cdot Z}{f} \tag{4}$$

From the epipolar geometry, the projection point can be computed for each view and matched image points; then, it can be back projected to give 3D structure.

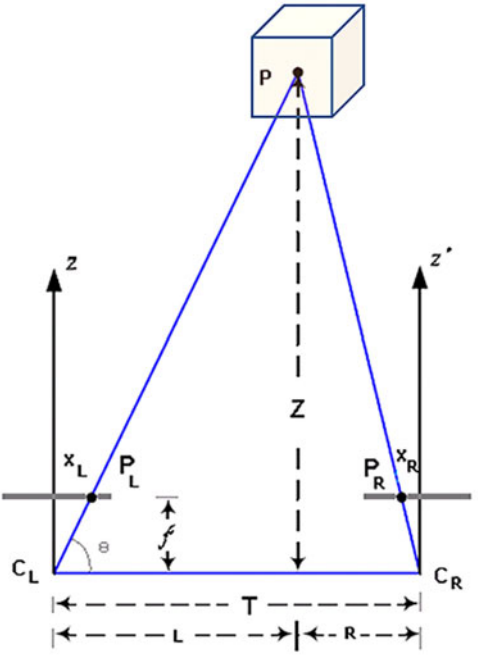


Fig. 8. Distance Estimation.

4.1.6 Stereo Image Model: Coordinate Systems and Scaling

Assuming that the cameras can be modeled by pinhole cameras, this model is shown on left side of Figure 9. The 2D image consider four coordinate systems: The (XW,YW,ZW) coordinate system; the 3D coordinate of the object point P in the 3D world coordinate system, (X,Y,Z); the 3D coordinate of the object point P in 3D camera coordinate system.

The Z axis coincide with the optical axis. Distance between image plane and the optical center O. (x,y) The image coordinate system with the origin at point $(O,O,)$ with respect to the 3D camera coordinate system and the image plane is the same as the plane $Z=(u,v)$ the coordinate system used in the computer or in respective digitized image.

$$\frac{f}{Z}=\frac{x}{X}=\frac{y}{Y} \tag{5}$$

ku, kv are the image scale factors, then:

$$x=\frac{u}{k_u}; \qquad y=\frac{v}{k_v} \tag{6}$$

By coordinate translation the scaled image coordinate (x,y) is transform in the respective computer coordinate:

$$u \rightarrow u - u_o; \qquad v \rightarrow v - vo \tag{7}$$

In this case (u_o, v_o) coincides with the image center O. From above equation we obtain:

$$x = \frac{u - u_0}{k_u}; \qquad y = \frac{v - v_0}{k_v} \tag{8}$$

From above equations,

$$u = \frac{\alpha_u X}{Z} + u_0; \qquad \alpha_v = f \cdot k_v \tag{9}$$

In our case, the parameters u_o, v_o, k_u, k_v, f are the camera intrinsic parameters. Lens distortion coefficients are not considered in such consideration [12].

4.1.7 Representing Robot Position

In order to specify the position of the robot on the plane we establish a relationship between the global reference frame of the plane and the local reference frame of the robot, as in Figure 9. The axes X_w and Y_w define an arbitrary inertial basis on the plane as the global reference frame from some origin $O : Xw, Yy$. Normally, to specify the position of the robot, a point Q on the robot chassis is selected as its position reference point, see Figure 10. The basis X_R, Y_R defines two axes relative to Q on the robot chassis and is thus the robot's local reference frame. The position of Q in the global reference frame is specified by coordinates x and y, and the angular difference between the global and local reference frames is given by θ.

In our case, point Q is located at the middle of the baseline T between both camera centers, as it was determined in (4).

We can describe the pose of the robot as a vector with these three elements. Note the use of the subscript w to clarify the basis of this pose as the global reference frame:

$$\xi_w = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{10}$$

To describe robot motion in terms of component motions, it will be necessary to map motion along the axes of the global reference frame to motion along the axes of the robot's local reference frame. Of course, the mapping is a function of the current pose of the robot. This mapping is accomplished using the orthogonal rotation matrix:

$$R(\theta) = \begin{bmatrix} cos(\theta) & sin(\theta) & 0 \\ -sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{11}$$

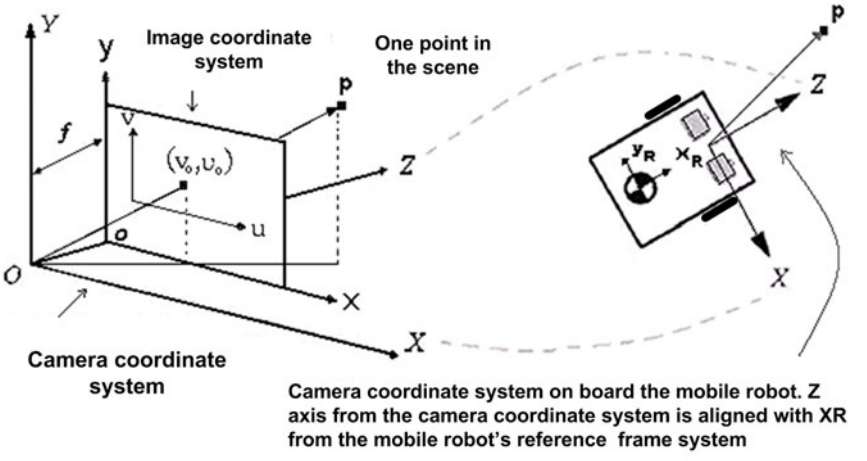


Fig. 9. Coordinates system using one camera model simplification. Z axis align with X_R MR local frame.

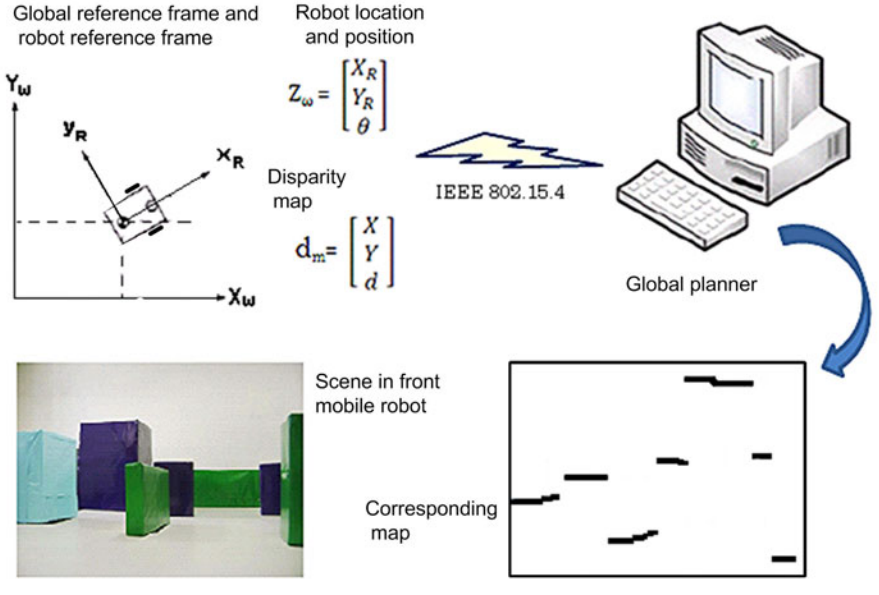


Fig. 10. Global planning based on robot location position and disparity map.

5 Map Building and Map Reconfiguration

The modification of the Road Maps is achieved using the information of disparity in pixels, where the distance of the MR from the obstacle is estimated using disparity measures, the less disparity measure means that the obstacle is far from the visual system of the MR. Moreover, the MR uses a high accuracy GPS and a digital compass for a better position and orientation in the environment, and by means of epipolar geometry and disparity map calculation is able to do the obstacle deep measure calculation.

The information coming from the MR has all the necessary (x, y, d) coordinates and corresponding disparities which in reality are a 3D representation of the 2D obstacles images captured from the stereoscopic visual system. After pixel's scaling and coordinates translation, the global planner is able to update the environment. The perceived obstacles are sent to the global planner, as it is seen in Figure 11, which includes the visual shape and geographical coordinates. Once, the global planner in the main computer has been modified using the new information about new obstacles and current position of the MR, the global planner performs calculations using ACO to obtain an updated optimized path, which is sent to the MR

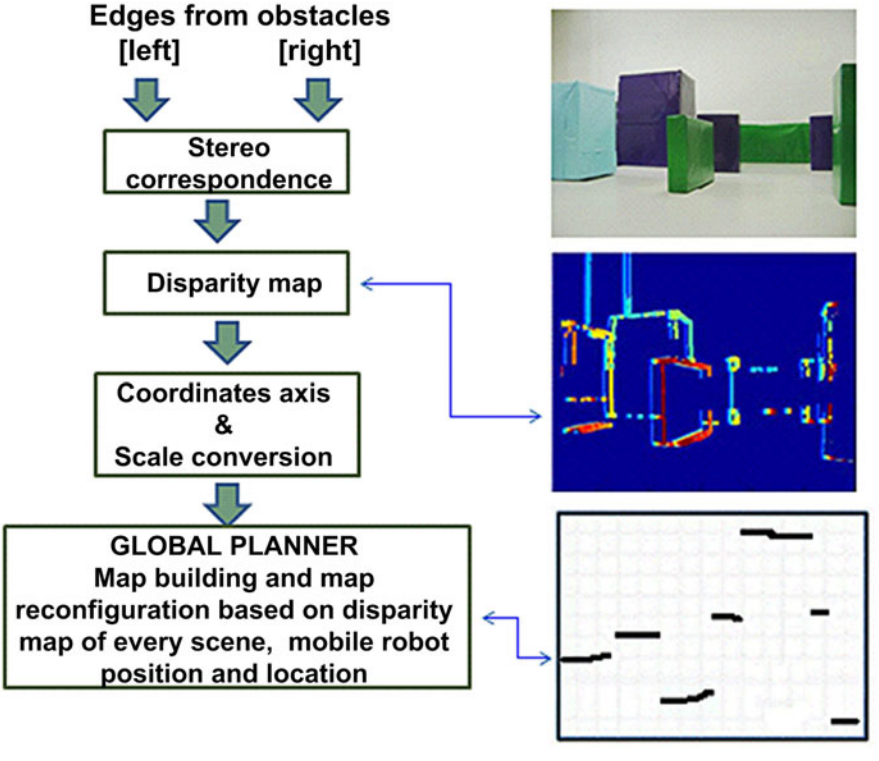


Fig. 11. Process for map building and map reconfiguration.

to achieve the navigation. The MR has the ability to send new information every 100*ms* via RF from every scene captured; however, times in the global planner are bigger since it is based on a natural optimization method, and it depends on the actual position of MR with respect to the goal. Hence, most of times a new path can be obtained every 3 seconds.

6 Conclusion

In this work a method for obstacle detection and map reconfiguration is presented, it is based on an stereoscopic vision module for a wheeled mobile robot, suitable to be implemented into an FPGA.

The main purpose of this proposal is to provide the necessary elements for perception, obstacles detection, map building and map reconfiguration in a tough environment where there are no landmarks or references.

There are only obstacles and new ones can be added. The stereoscopic vision system captures left and right images from the same MR scene, the system is capable of using both appearance based pixel descriptors for surface ground extraction, luminance and Hue depending of the environment particular characteristics. In an environment with constant lightness, minimum reflections and proper setting in the edge detector threshold level, luminance can be suitable because surface ground and obstacles edge detection can be performed at the same time. For environment with variable light conditions or uncertain, Hue is the primary attribute for pixel appearance descriptor in the surface ground extraction process due to it's invariance to changes in luminance and color saturation. After surface ground extraction and obstacles edge detection, stereoscopic corresponding by block matching is performed, the correspondence is found among a set of points in the left and right images by using a feature based approach. Disparity computation for the matched points is then performed. Establishing correspondences between point locations in images acquired from multiple views (matching) is one of the key tasks in the reconstruction based on stereo image analysis. This feature based approach involves detecting the feature points and tracking their positions in multiple views of the environment. In order to solve the stereo correspondence problem, an improved method is introduced for calculation efficiency and accuracy "adaptive candidate matching window" (ACMW). We used the sum of absolute differences metric (SAD) for stereoscopic correspondence matching. Stereoscopic camera calibration is not required due to the improvements in matching process. Disparity maps which are the depth measure of the obstacles position in the environment are obtained after the stereo correspondence process. The MR sends this data, including its position and orientation via RF to the global planner located in the main computer outside the environment. With this information the global planner is able to constantly update the environment map.

Acknowledgements. The authors thank Comisión de Operación y Fomento de Actividades Académicas del I.P.N., Instituto Tecnológico de Tijuana for supporting our research activities, and CONACYT.

References

1. Fusiello, A., Roberto, V., Trucco, E.: Symmetric stereo with multiple windowing. Int. J. Pattern Recognit. Artif. Intell. 14(8), 1053–1066 (2000)
2. Woods, A.J., Docherty, T.M., Koch, R.: Image distortions in stereoscopic video systems. In: Proceedings of the SPIE, San Jose, Ca, USA, vol. 1925 (1993)
3. Calisi, D., Iocci, L., Leone, G.R.: Person Following through Appearence Models and Stereo Vision using a Mobile Robot. In: Proceedings of International Workshop on Robot Vision, pp. 46–56 (2007)
4. Aggarwal, J.K., Zhao, H., Mandal, C., Vemuri, B.C.: 3D Shape Reconstruction from Multiple Views. In: Bovik, A.C. (ed.) Handbook of Image and Video Processing, pp. 243–257. Academic Press, London (2000)
5. Kolmogorov, V., Zabih, R.: Computing visual correspondence with occlusions using graph-cuts. In: Proceedings Int. Conf. Comp. Vis., pp. 508–515 (2001)
6. Okutomi, M., Kande, T.: A multiple baseline stereo. IEEE Transactions on pattern analysis and machine intelligence 15(4), 353–363 (1993)
7. Okutomi, M., Katayama, Y., Oka, S.: A simple stereo algorithm to recover precise object boundaries and smooth surfaces. International Journal of Computer Vision 47(1-3), 261–273 (2002)
8. García, M.A.P., Montiel, O., Castillo, O., Sepúlveda, R., Melin, P.: Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation. Applied Soft Computing 9(3), 1102–1110 (2009)
9. Abellatif, M.: Behavior Fusion for Visually-Guided Service Robots. In: In-Teh. Computer Vision, Croatia, pp. 1–12 (2008)
10. Faugeras, O.: Three dimensional computer vision. MIT Press, Cambridge (1993)
11. Khatib, O.: Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. In: Procedings of IEEE International conference on Robotics and Automation, pp. 500–505 (1985)
12. Tsai, R.Y.: An efficient and accurate camera calibration technique for 3D machine vision. In: IEEE Conference on Computer Vision and Pattern recognition, pp. 364–374 (1986)
13. Siegwart, R., Nourbakhsh, I.R.: Introduction to Autonomous Mobile Robots, A Bradford Book. The MIT Press, Cambridge (2004)
14. Adhyapak, S.A., Kehtarnav, N., Nadin, M.: Stereo matching via selective multiple windows. Journal of Electronic Imaging 16(1) (2007)
15. VBOX product, `http://www.racelogic.co.uk/?show=VBOX`
16. Grimson, W.E.L.: Computational experiments with feature based stereo algorithm. IEEE Transactions on pattern analysis and machine intelligence 7(1), 17–34 (1985)
17. Xbee XBee-Pro OEM RF Modules, Product Manual v1.xAx - 802.15.4 Protocol, MaxStream, Inc. (2007)
18. Cao, Z., Hu, J., Cao, J., Hall, E.L.: Ommi-vision based Autonomous Mobile Robotic Platform. In: Proceedings of SPIE Intelligent Robots and Computer Vision XX: Algorithms, Techniques, and Active Vision, Newton USA, vol. 4572, pp. 51–60 (2001)
19. Cao, Z., Meng, X., Liu, S.: Dynamic Omnidirectional Vision Localization Using a Beacon Tracker Based on Particle Filter, Computer Vision. In: Zhihui, X. (ed.) In-Teh, pp. 13–28 (2008)

Automatic Dust Storm Detection Based on Supervised Classification of Multispectral Data

Pablo Rivas-Perea[1], Jose G. Rosiles[1],
Mario I. Chacon Murguia[2], and James J. Tilton[3]

[1] The University of Texas El Paso,
Department of Electrical and Computer Engineering, El Paso TX 79968, USA
[2] Chihuahua Institute of Technology,
Graduate Studies Department, Chihuahua Chih, Mexico
[3] NASA Goddard Space Flight Center,
Computational and Information Sciences and Technology Office,
Greenbelt MD 20771, USA

Abstract. This paper address the detection of dust storms based on a probabilistic analysis of multispectral images. We develop a feature set based on the analysis of spectral bands reported in the literature. These studies have focused on the visual identification of the image channels that reflect the presence of dust storms through correlation with meteorological reports. Using this feature set we develop a Maximum Likelihood classifier and a Probabilistic Neural Network (PNN) to automate the dust storm detection process. The data sets are MODIS multispectral bands from NASA Terra satellite. Findings indicate that the PNN provides improved classification performance with reference to the ML classifier. Furthermore, the proposed schemes allow real-time processing of satellite data at 1 km resolutions which is an improvement compared to the 10 km resolution currently provided by other detection methods.

1 Introduction

Dust aerosols are a major cause of health, environmental, and economical hazards, and can adversely impact urban areas [1]. From a scientific perspective, understanding dust storm genesis, formation, propagation and composition is important to reduce their impact or predict their effect (*e.g.*, increase of asthma cases). Multispectral instruments allow space imaging of atmospheric and earth materials based on their spectral signature. More specifically, they allow the detection of dust air-borne particles (aerosols) propagated through the atmosphere in the form of dust storms.

Current methods for dust aerosol are based on the Moderate Resolution Spectroradiometer (MODIS) Aerosol Optical Thickness (AOT) product [2, 3] which is provided by the NASA Terra satellite.

P. Melin et al. (Eds.): Soft Comp. for Recogn. Based on Biometrics, SCI 312, pp. 443–454.
springerlink.com

The AOT product allows tracking of pollutant aerosols by observing the aerosol optical thickness. However, AOT products require a considerable amount of processing time (*i.e.*, two days after satellite pass) before useful information on aerosol events is extracted. The use of simple band arithmetic (e.g., subtraction) has been reported as a scheme to visualize the presence of dust storms [1]. This method is highly subjective making interpretation dependent on the experience of the analyst.

Given the large amount of data produced by MODIS, it is also desirable to have automated systems that assists scientist on finding or classifying different earth phenomena with minimal human intervention. For example, Aksoy, *et al.* [4], developed a visual grammar scheme that integrates low-level features to provide a high level spatial scene description on land cover and land usage. Similar automated schemes for dust detection are highly desirable.

In this paper we present two methods for the detection of dust storms from multispectral imagery using statistical pattern classifiers. Based on reported data, we present a feature set that allows accurate and real-time detection of dust aerosol. The proposed feature set is extracted from MODIS spectral bands and evaluated with a maximum likelihood (ML) classifier and a probabilistic neural network (PNN). We will show that the PNN approach provides a better detection and representation of dust storm events.

This paper is organized as follows. Section 2 of the paper introduces the dust aerosol multispectral analysis. The ML and PNN models are explained in Section 3 and 4. Section 5 presents experimental results leading to different levels of segmentation between dust storms and other materials. Finally, conclusions are drawn in Section 6.

2 An Overview of MODIS Data

Remote sensing is the research area that studies how to gather and analyze information about the Earth from a distance. Uses include the mapping of fires, weather monitoring, cloud evolution, and land cover analysis. The information gathered can be used to produce images of erupting volcanoes, monitor for dust storms, view the sequential growth of a city, and track deforestation over time [6, 18].

In this paper we collected thermal information about the land, stratosphere, and atmosphere using special instruments aboard a satellite orbiting the Earth surface. This instrument is called "Moderate-Resolution Imaging Spectroradiometer" (MODIS). These remotely sensed data is collected as digital files, containing data captured at different spectral waves in the optical range (*i.e.* multispectral data). These digital files are known as "granules" and can be downloaded from the web at the NASA WIST tool.

The MODIS instrument is built in NASA Terra and Aqua satellites. MODIS multispectral data is currently used in the analysis of different

phenomena like sea temperature and surface reflectivity. MODIS provides information in 36 spectral bands between wavelengths 405nm and 14.385μm.

MODIS multispectral data is available in different levels. These levels depend on the level of data processing. Level 0 is raw telemetry data (*i.e.* satellite unorganized data). Level 1A is raw data organized by spectral bands. Level 1B consists of corrected multispectral data (*i.e.* bad sensor information is pointed out). Subsequent levels are processed for particular analysis that include aerosol, water vapor, and cloud. In this paper we use the multispectral bands available in MODIS Level 1B.

3 Selection and Analysis of Spectral Bands for Feature Extraction

In this section we described the proposed feature extraction process based on the analysis of spectral bands reported in the literature. These studies have focused on the visual identification of the image channels that reflect the presence of dust storms through correlation with meteorological reports. Visual assessment of dust storms can be achieved using MODIS bands $B1$, $B3$, and $B4$ which are within human visual range [5]. An RGB-like composite image can be produced by the mapping red to $B1$, green to $B4$, and blue to $B3$. Hao *et al.*[6] demonstrated that bands $B20$,$B29$,$B31$ and $B32$ can also be utilized for dust aerosol visualization. Ackerman *et al.*[7] demonstrated that band subtraction $B32 - B31$ improves dust storm visualization contrast. Based on these findings, we will form feature vectors using pixels values from the recovered bands $B20$, $B29$, $B31$, and $B32$.

A "recovered" radiance is a 16 bit MODIS band mapped to its original units ($W/m^2/\mu m/sr$). The recovery process is given by

$$L = \kappa(\iota - \eta), \tag{1}$$

where L denotes the recovered radiance, κ is the radiance scale, η denotes the radiance offset, and ι is the scaled intensity (raw data). For each pixel location (n, m), a feature vector $F \in \Re^4$ is formed by

$$F_{nm} = \left[L_{nm}^{B20}, L_{nm}^{B29}, L_{nm}^{B31}, L_{nm}^{B32}\right]^T. \tag{2}$$

corresponding to the recovered radiances of the dust sensitive wavelengths.

4 Dust Storm Detection Using the Maximum Likelihood Classifier

The Maximum Likelihood Classifier (ML) has been extensively studied in remotely sensed data classification and analysis [9, 4]. Here we present a straightforward adaptation of the ML classifier to dust storm detection

using the feature set described in the previous section. Let $f_{X|k}(x) = (X = x|C = k)$ be the conditional probability density function of feature vector X having a value x, given the probability that the k-th class occurs. This might be referred as the "data likelihood" function. Assuming normally distributed features (*i.e.*, pixel values), we can define a discriminant function

$$\psi_k(x) = -\det(\mathbf{\Sigma}_k) - (x - \mu_k)^T \mathbf{\Sigma}_k^{-1} (x - \mu_k) \tag{3}$$

for each class k, where $\mathbf{\Sigma}_k$ the covariance matrix, μ_k denotes the mean feature vector, and $\det(\cdot)$ is the determinant function. Then, the decision rule can be simply stated as

$$x \in C = j \quad \text{if} \quad \psi_j(x) > \psi_i(x) \quad \forall j \neq i. \tag{4}$$

The parameters $\mathbf{\Sigma}_k$ and μ_k were obtained from the training data described in the previous section using the maximum likelihood estimators (*e.g.*, sample mean and sample covariance matrix).

5 Neuro-Probabilistic Modeling: The Probabilistic Neural Network

Specht's Probabilistic Neural Network (PNN) is a semi-supervised neural network [10]. It is widely used in pattern recognition applications [11]. The PNN is inspired in Bayesian classification and does not require training. It estimate the PDF of each feature assuming they are normally distributed. The PNN has a four-layered architecture as shown in Figure 1. The first layer is an input layer receiving the feature vectors F_{nm}. The second layer consists of a set of neurons which are fully connected to the input nodes. The output of this layer is given by

$$\varphi_{jk}(F) = \frac{1}{(2\pi)^{\frac{d}{2}} \sigma^d} e^{-\frac{1}{2\sigma^2}\left(F - \nu_{jk}^F\right)^T \left(F - \nu_{jk}^F\right)}. \tag{5}$$

where j is an index labeling each design vector and k is its the corresponding class. The pattern units ν_{jk}^F correspond to the mean feature vector for each class. The parameter σ is estimated with the method developed by Srinivasan *et al.* [12].

The third layer contains summation units to complete the probability estimation. There are as many summation units as classes. The $j-th$ summation unit denoted as $\Omega_j(\cdot)$, receives input only from those pattern units belonging to the $j-th$ class. This layer computes the likelihood of F being classified as C, averaging and summarizing the output of neurons belonging to the same class. This can be expressed as

$$\Omega_j\left(\varphi_{jk}\left(F\right)\right) = \frac{1}{\left(2\pi\right)^{\frac{d}{2}}\sigma^d}\frac{1}{N_j} \times \ldots$$
$$\sum_{i=1}^{N_j} e^{-\frac{1}{2\sigma^2}\left(\varphi_{ik}(F)-\varpi_i\right)^T\left(\varphi_{ik}(F)-\varpi_i\right)}. \tag{6}$$

The last layer classifies feature input vector F_{nm} according to the Bayesian decision rule given by

$$F \in C_j \text{ if, } \ldots$$
$$C_j\left(\Omega_j\left(\varphi_{jk}\left(F\right)\right)\right) = \arg\max_{1\leq i\leq j} \Omega_i\left(\varphi_{ik}\left(F\right)\right). \tag{7}$$

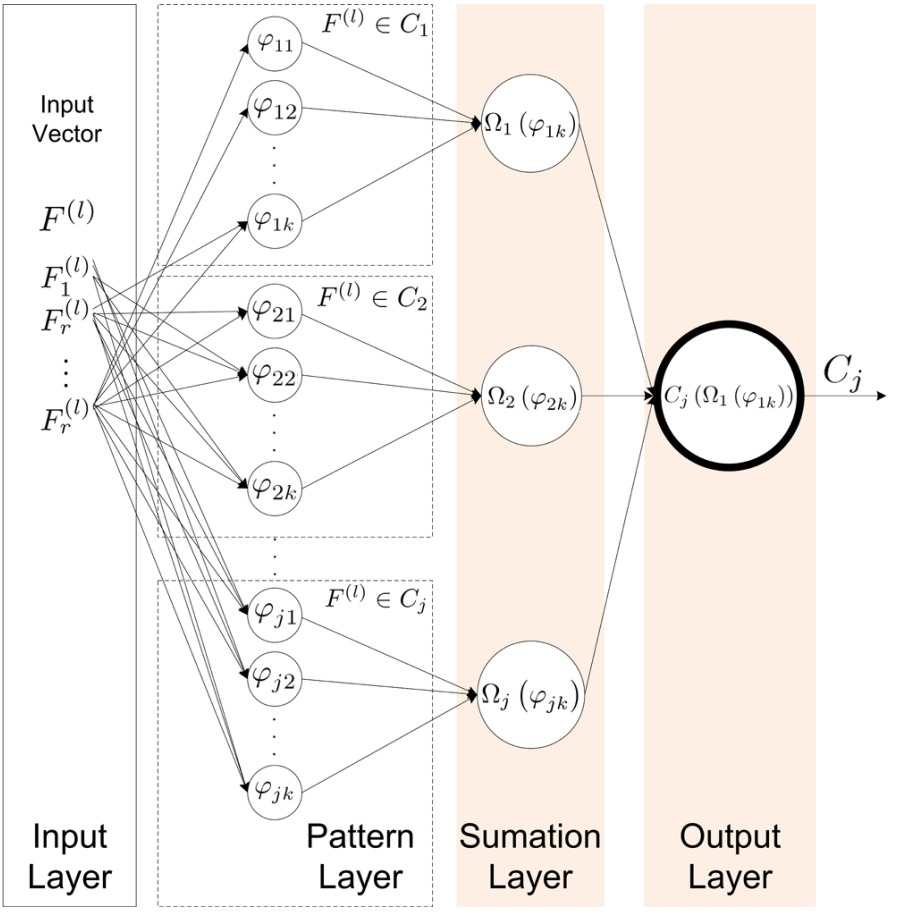


Fig. 1. The hybrid architecture of the Probabilistic Neural Network. Note the probabilistic nature embedded in a neural architecture.

5.1 The PNN Large Sample Size Problem

To avoid the overwhelming processing of millions training samples, we limited the training samples number. We based our reduction method on Kanellopoulos criteria [13] which establishes that the number of training samples must be at least three times the number of feature bands. Therefore, in our PNN design we used six times the feature vector size (*e.g.*, four) requiring 24 training samples per class. In order to select the testing vectors (24 per class), principal component analysis (PCA) was applied to a training set consisting of millions of feature vectors. Then the test feature vectors associated to the 24 largest eigenvalues were selected as the PNN training set.

6 Results and Discussion

In our experiments we selected 31 different events corresponding to the southwestern US, and north-western Mexico area. The 31 events are known dust storm cases reported in [8]. From these events, 23 were selected to train and test the classifiers. Each event contains multispectral images of size 2030×1053 pixels. We manually segmented the images using the MODIS visual range into four classes $C = \{dust\ storm, blowing\ dust, smoke, background\}$. The selection of modeling (training) and testing feature vectors was performed by PCA as explained in the last section. The complete data set provides approximately 75 million feature vectors from which 97.5% correspond to the background class. The feature vectors are sliced into 0.005% for training and the remaining are for testing.

In order to evaluate the performance of the classifiers, we need to select a figure of merit. Typically accuracy, received operating characteristic (ROC) or area under the ROC curve (AUC) have been used individually. However, as reported in [16] these measures can only be used interchangeably when the positive and negative test sets are large and balanced. Hence, it is now recognized that using more than one figure of merit is necessary to have a good assessment of a classifier. We evaluate our results using accuracy defined as

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN}, \tag{8}$$

where TP is the number of true positives, FP is the number of false positives, TN is the number of false negatives and FN is the number of false negatives. Hence accuracy corresponds to the correct classification rate over all classes. A related measure is precision or positive predictive value (PPV) given by

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{9}$$

In this case, precision represents the fraction of true positives from all the vectors classified as a positive. Finally we use AUC which is has been recognized in the machine learning community to provide a better metric than

Table 1. Classifiers Performance.

	Precision	Std. Dev.	Accuracy	Std. Dev.	AUC	Std. Dev.	P. Time	Std. Dev.
ML	0.5255	0.2610	0.6779	0.1282	0.4884	0.0036	0.1484	0.0021
PNN	0.7664	0.1616	0.8412	0.1612	0.6293	0.0654	2.5198	0.0018

accuracy [14]. In summary, higher precision and accuracy reflect that a system produces more true positives results while reducing the number of false negatives. Similarly, a higher AUC reflects how a classifier is able to correctly classify feature vectors and at the same time minimize the misclassification errors.

Since in our case we have four classes, generalizing precision and accuracy is obtained by considering a 4×4 confusion matrix where the main diagonal entries represent the true positives for each class. We can drop the idea of a negative set and use TP_i to identify the true positives for class class i. The idea of false negatives is now represented by the off-diagonal elements of the confusion matrix. For instance, the false negatives for class *dust storm* consists of those vectors misclassified as *blowing dust*, *smoke* or *background*. Similarly, false positives consists of all those vectors classified as *dust storm* that belong to any of the other three classes. Based on these considerations, expressions for precision and accuracy are straightforward to derive. The case of multi-class ROCs and therefore AUCs is an open problem. Some multi-class AUCs are described in [17]. In this paper we resorted to a simpler method where we create a binary classifier by grouping both types of dust as a single (i.e., positive) class, and lumping *smoke* and *background* as the negative.

We present metric results on Table 1. These results were obtained from the whole set of 26 events by averaging each event results. Overall the PNN approach provides better classification than ML. In particular, the AUC indicates that the ML classifier should not be used in the dust storm detection problem. On the other hand, the other metrics show a modest level of performance. Hence, using multiple metrics provides a better understanding on the capabilities of each classifier.

Beyond classifier performance, it is important to integrate the results of the classification with actual images. Ultimately, the output of the classifiers should be used as a tool to help scientists develop insights about dust storms. We present two typical dust storm events in Figure 2. These color images were obtained by mapping three MODIS bands to red, blue and green respectively. The classification results can be visualized as the segmentations shown in Figure 3 for the ML and PNN classifiers. Pixels classified as *dust storm* are labeled red, *blowing dust* to green, *smoke* to blue, and *background* to black. Both classifiers detect the presence of the storms, albeit the PNN detects larger regions. This can be directly explained by the higher PNN metric values on Table 1. From a detection perspective, both classifiers are successful. The ML classifier would be attractive as a detector given its lower computational

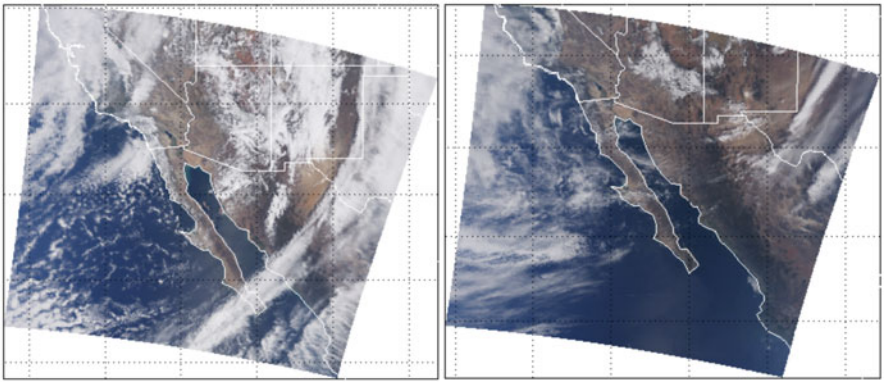


Fig. 2. Left, dust storm event on April 6th 2001. True color image R=B1, G=B4, and B=B3. Right, dust storm event on December 15, 2003. True color image R=B1, G=B4, and B=B3.

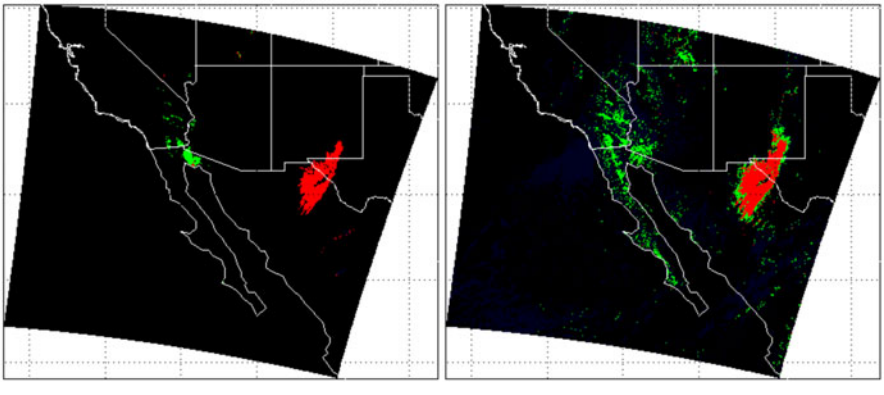


Fig. 3. Dust storm event on April 6th 2001. Left, segmentation using ML. Right segmentation using PNN.

requirements. However, if a better understanding on the spatial distribution of the storm is needed, then the PNN should be the selected classifier.

Processing time is an important measure when modeling real-time processing systems. In the case of the MODIS instrument, image swaths of $10 \times 1053 \times 36$ pixels known as "scans" are produced every 2.96 seconds (*i.e.* 20.3 scans per minute). Thus, a real-time system must perform a classification in less than or equal to this time. The fourth column on Table 1 shows the processing time per scan in seconds. The time shown is computed by taking the time average over all scans for all he events. The times were measured with a MATLAB implementation running on a 2 GHz PC. The time was measured using the *tic(), toc()* functions that give the true CPU

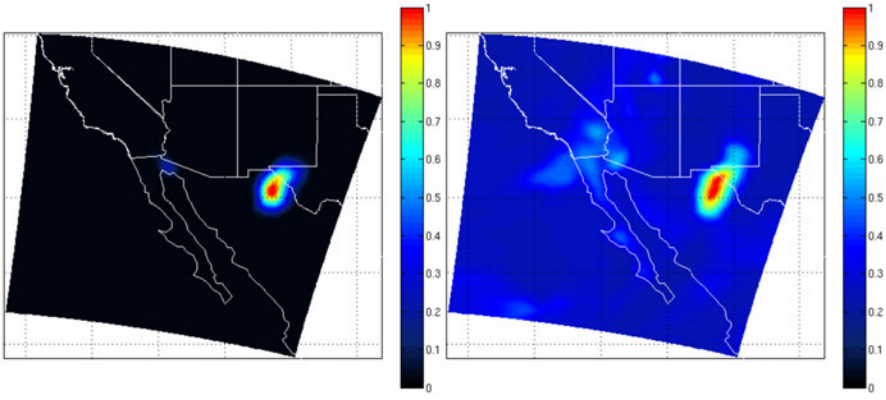


Fig. 4. Dust storm event on April 6th 2001. Left, dust likelihood probability ML. Right, dust likelihood probability PNN.

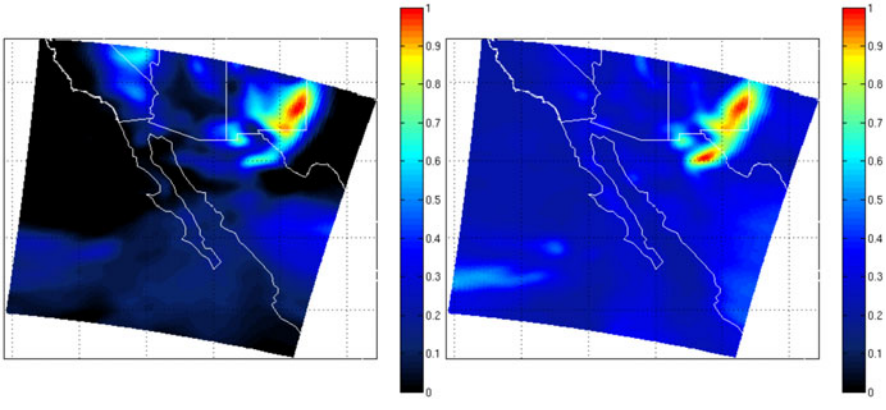


Fig. 5. Dust storm event on December 15, 2003. Left, dust likelihood probability ML. Right, dust likelihood probability PNN.

processing time. The ML approach takes less than one second to classify the complete scan, and the PNN approach takes about 2.5 seconds to produce the classification result. In conclusion, both can be considered suitable for real time detections at 1km resolution. In contrast, the MODIS AOT product takes two days to be produced and released at a 10km resolution [15].

Finally, as a byproduct of the classification stage, it becomes possible to extract more information about a dust storm by visualizing the dust likelihood over the whole image. Both classifiers produce a parametrization of the likelihood probability density function of dust $f(\mathbf{x}|\text{dust storm})$ under a multivariate Gaussian assumption. Namely, a new image is formed by assigning a value of $f(\mathbf{x}|\text{dust storm})$ for each feature vector F_{nm}. This visualization

over an image provides unique information about the spatial distribution of dust at the moment the image was acquired. This can be utilized to track dust aerosols with a particular degree of confidence. The degree of confidence is proportional to the probability of a pixel being classified as dust storm. With this kind of visualization we can show only those pixels classified as dust storm with a high degree of confidence (*e.g.* above 90% of confidence), that resemble a conservative detection with a high degree of exigence. On the other hand, we can use a low confidence interval (*e.g.* above 5% of confidence) to study how the dust storm spreads across land. This analysis is known as "dust transport," and is relevant on establishing the origin and extensions of a dust storm. Since the dust aerosol concentration is reduced as the storm advances, dust transport can be studied by analyzing the pixels classified as dust storm but with lower probability.

The dust likelihood visualization of the April 6th, 2001 event is shown in Figure 4 One particularly interesting case is shown in Figure 5, where the visual composite of the satellite image (Figure 2) shows one dust cloud; however, when we observe the dust likelihood visualization we can notice that there where two different dust storm outbreaks at different sources. This information is difficult to see using only the visual composite of MODIS, neither is possible using the AOT product because of the lack of spatial resolution.

7 Conclusion

The dust aerosol detection problem has been addressed in this paper. We have modeled probabilistic approaches for dust storm detection and classification. These models are specialized on measuring the dust aerosol probability given MODIS Level 1B data. Machine learning techniques were utilized to model a dust aerosol detection neural architecture. To the best of the authors knowledge, the presented work is first in its kind. We compared the Maximum Likelihood classification (ML) model, and the Probabilistic Neural Network (PNN). The PNN showed a strong ability classifying dust, and discriminating other classes, such as clouds, smoke, and background. Moreover, the proposed probabilistic models are suitable for near real-time applications, such as direct broadcast, rapid response analysis, emergency alerts, etc. The reported work has relevancy in dust aerosol analysis, since the algorithms can show the dust presence to a resolution of 1km. This represents an improvement over Aerosol Optical Thickness index (AOT) methods which lack resolution and have a two day generation delay.

Acknowledgments

The author P.R.P performed the work while at NASA Goddard Space Flight Center under the Graduate Student Summer Program (GSSP). This work

was partially supported by the National Council for Science and Technology (CONACyT), Mexico, under grant 193324/303732, as well as by the University of Texas El Paso Graduate School Cotton Memorial Funding. The author M.I.C.M. wants to thank DGEST for the support in the dust storm project.

References

1. Rivera Rivera, N.I., Gebhart, K.A., Gill, T.E., Hand, J.L., Novlan, D.J., Fitzgerald, R.M.: Analysis of air transport patterns bringing dust storms to El Paso, Texas. In: Symposium on Urban High Impact Weather, AMS Annual Meeting, Phoenix, January 2009, vol. JP2.6 (2009)
2. Khazenie, N., Lee, T.F.: Identification Of Aerosol Features Such As Smoke And Dust, In NOAA-AVHRR Data Using Spatial Textures. Geosc. and Rem. Sens. Symp. (May 1992)
3. Levy, R.C., Remer, L.A., Kaufman, Y.J., Tanr, D., Mattoo, S., Vermote, E., Dubovik, O.: Revised Algorithm Theoretical Basis Document: MODIS Aerosol Products MOD/MYD04 (2006)
4. Aksoy, S., Koperski, K., Tusk, C., Marchisio, G., Tilton, J.C.: Learning bayesian classifiers for scene classification with a visual grammar. IEEE Trans. on Geosc. and Rem. Sens. 43(3), 581–589 (2005)
5. Gumley, L., Descloitres, J., Schmaltz, J.: Creating Reprojected True Color MODIS Images: A Tutorial. Technical Report. Version: 1.0.2, January 14, (2010)
6. Hao, X., Qu, J.J.: Saharan dust storm detection using MODIS thermal infrared bands. Journal of Applied Remote Sensing 1, 013510 (2007)
7. Ackerman, S.A., Strabala, K.I., Menzel, W.P., Frey, R., Moeller, C., Gumley, L., Baum, B.A., Seeman, S.W., Zhang, H.: Discriminating clear-sky from cloud with MODIS algorithm theoretical basis document (MOD35). ATBD-MOD-06, version 4.0, NASA Goddard Space Flight Center
8. Novlan, D.J., Hardiman, M., Gill, T.E.: A synoptic climatology of blowing dust events in El Paso, Texas. In: 16th Conference on Applied Climatology, AMS, vol. J3.12, p. 13
9. Richards, J.A., Jia, X.: Remote Sensing Digital Image Analysis. Springer, Heidelberg (2006)
10. Specht, D.F.: Probabilistic neural networks for classification, mapping, or associative memory. In: IEEE international conference on neural networks, vol. 1(7), pp. 525–532 (1988)
11. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, pp. 1–654. John Wiley and Sons, New York (2001)
12. Ramakrishnan, S., Selvan, S.: Image texture classification using wavelet based curve fitting and probabilistic neural network. Int. J. Imaging Syst. Technol., 266–275 (2007)
13. Kanellopoulos, G., Wilkinson, G., Austin, J.: Neurocomputation in Remote Sensing Data Analysis. Springer, Heidelberg (1997)
14. Hanley, J.A., McNeil, B.J.: The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology 143(1), 29 (1982)

15. Crosson, W., Rickman, D., Limaye, A.: Optimal temporal scale for the correlation of AOD and ground measurements of PM2.5 in a real-time air quality estimation system. Jnl. Atmospheric Environment 43(28), 4303–4310 (2009)
16. Mazurowski, M.A., Tourassi, G.D.: Evaluating classifiers: Relation between area under the receiver operator characteristic curve and overall accuracy. In: IEEE - INNS - ENNS International Joint Conference on Neural Networks, 2009 International Joint Conference on Neural Networks, pp. 2045–2049 (2009)
17. Fawcett, T.: ROC graphs: Notes and practical considerations for researchers. Machine Learning 31 (2004)
18. Tao, H., Yaohui, L., Hui, H., Yongzhong, Z., Yujie, W.: Automatic Detection of Dust Storm in the Northwest of China Using Decision Tree Classifier Based on MODIS Visible Bands Data. In: Proceedings of IEEE International Geoscience and Remote Sensing Symposium IGARSS 2005, Korea (July 2005)

Author Index