ESTIMATING THE MAINTENANCE COST IN ELECTRIC NETWORKS USING AN HYBRID SOFT COMPUTING METHODOLOGY

Jesús Acosta
Departamento de Instrumentación Industrial
Inst. Universitario de Tecnología "Alonso Gamero"
4101- Coro, Estado Falcón, Venezuela
E-mail: jacostav@ cantv.net

Àngela Nebot
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
08034 - Barcelona, Spain
E-mail: angela@lsi.upc.edu

Pedro Villar
Departamento de Lenguajes y Sistemas
Informáticos
Universidad de Granada
18071 - Granada, Spain
E-mail: pvillarc@ugr.es

Josep M. Fuertes
Departament d'Enginyeria de Sistemes,
Automàtica i Informàtica Industrial
Universitat Politècnica de Catalunya
08028 - Barcelona, Spain
E-mail: josep.m.fuertes@upc.edu

KEYWORDS

Electric distribution networks, genetic algorithms, fuzzy inductive reasoning, genetic fuzzy systems, machine learning.

ABSTRACT

The main goal of this research is the development of a genetic fuzzy system (GFS) to solve the problem of estimating the maintenance cost of medium voltage lines in Spanish towns. The hybrid system is composed by the Fuzzy Inductive Reasoning (FIR) methodology and a genetic algorithm (GA) that is the responsible of determining in an automatic way the fuzzification parameters involved in the fuzzy system, i.e. the number of fuzzy sets (classes) per variable and the membership functions. The results obtained are compared with some of the most popular classical statistical modeling methods, neural networks and other hybrid evolutionary data analysis techniques.

I. INTRODUCTION

Fuzzy systems have demonstrated their ability to solve different kind of problems such are control (Driankov 1993; Leondes 1999), modeling (Pedrycz 1996) or classification (Kuncheva 2000; Chi et al. 1996; Vapnik 1998), and have been successfully applied to a wide range of applications, i.e. signal and image processing (Chi et al. 1996; Sattar and Tay 1999; Suzuki et al. 2001), risk assessment (Leondes 1999), information retrieval (Chen et al. 2001; Miyamoto 1989), industrial applications (Leondes 1999; Hirota and Sugeno 1995; Dote and Ovaska 2001), etc.

In the last decade, there was an increasing interest to include learning in fuzzy systems. This has been achieved by means of the development of hybrid techniques that include fuzzy systems together with complementary techniques such are neural networks, evolutionary algorithms or probabilistic methods.

It is well known that more intelligent systems can be obtained by the hybridization of soft computing methodologies (Bonissone 1997; Cordón et al. 2001). Neural fuzzy systems (NFSs) and genetic fuzzy systems (GFSs) are the most successful approaches of hybrid systems within soft computing. NFS and GFS hybridize the approximate reasoning method of fuzzy systems with the learning capabilities of neural networks and evolutionary algorithms, respectively.

In the research presented in this paper we propose a new GFS to improve the Fuzzy Inductive Reasoning (FIR) methodology. FIR is a inductive modeling and prediction methodology that has been applied to different kinds of applications (e.g. control, biomedicine, ecology), usually obtaining good results (Nebot et al. 1996; Mugica and Cellier 1994; Nebot et al. 2001). In these studies, default values have been used to determine the number of classes and the associated membership functions. The default value for the number of classes' parameter for each system variable is three and the equal frequency partition (EFP) is used as the default method to obtain the membership functions of the classes. However, experience has shown that in some applications, i.e. biomedical and ecological, the determination of the parameters needed in the discretization step of FIR becomes significant for the identification of a good model that captures systems behavior in an accurate way. Therefore, the automatic determination of good fuzzification parameters in the FIR methodology is an interesting and useful alternative to the use of heuristics and/or default values. This is, precisely, the main contribution of this paper, i.e. the design and development of a GFS composed by the FIR methodology and a GA for the automatic determination of FIR fuzzification parameters.

The GFS developed is used for model identification of a real problem, i.e. estimating the maintenance cost of medium voltage lines in Spanish towns. The results obtained with the new method are compared with the ones obtained by other methodologies in the same application, i.e. neural networks, genetic programming, genetic fuzzy rule-base systems, lineal models, etc.

The FIR methodology is presented in Section 2. The GA proposed is described in Section 3. Section 4 presents the electrical application and the discussion of the obtained results. Finally, the conclusions of this research are given.

II. FUZZY INDUCTIVE REASONING METHODOLOGY

The conceptualization of the Fuzzy Inductive Reasoning

(FIR) methodology arises of the General System Problem Solving approach (GSPS) proposed by (Klir 1985). This methodology of modeling and qualitative simulation is based on systems behavior rather that on structural knowledge. It is able to obtain good qualitative relations between the variables that compose the system and to infer future behavior of that system. It has the ability to describe systems that cannot easily be described by classical mathematics (e.g. differential equations), i.e. systems for which the underlying physical laws are not well understood. FIR is composed of four main processes, namely: *fuzzification, qualitative model identification, fuzzy forecasting* and *defuzzification*. Fig. 1 describes the processes of the FIR methodology.

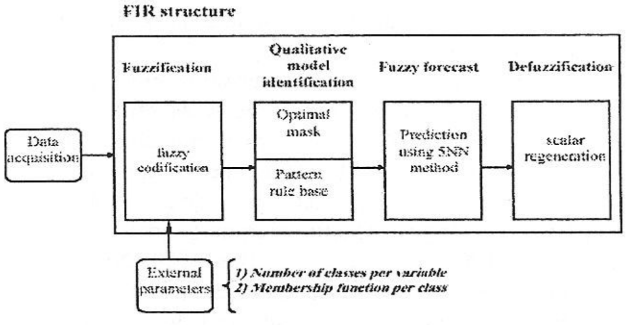

Figure 1: Fuzzy Inductive Reasoning (FIR) scheme

The *fuzzification* process converts quantitative data stemming from the system into fuzzy data, i.e. qualitative triples. The first element of the triple is the class value, the second element is the fuzzy membership value, and the third element is the side value. The side value indicates whether the qualitative value is to the left or to the right of the peak value of the associated membership function (see Fig. 2).

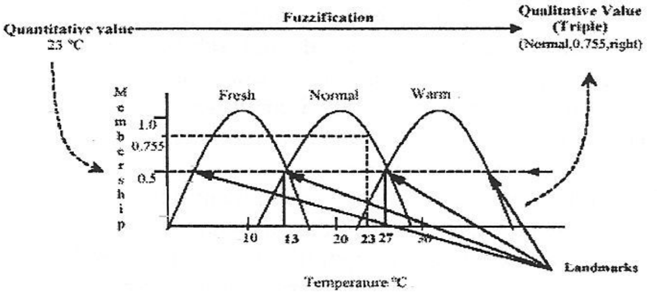

Figure 2: FIR fuzzification process of ambient temperature variable

The side value, that is not commonly used in fuzzy logic, is responsible for preserving, in the qualitative triple, the complete knowledge that had been contained in the original quantitative value.

The result of the *fuzzification* process are three matrices of identical size named qualitative data matrices, one containing the class values, the second storing the membership information, and the third recording the side values. Each column represents one of the observed variables and each row denotes one time point, i.e. one recording of all variables or one recorded state. For simplicity, in FIR the class values are represented numerically, i.e. the fresh, normal and warm classes of Fig. 2 are represented with the numerical values 1, 2 and 3, respectively.

The *qualitative model identification* process is the responsible of finding causal and temporal relations between variables and therefore of obtaining the best model that represents the system. A FIR model is composed by a structure, called mask, and a pattern rule base, named behaviour matrix.

A mask denotes a dynamic relationship among qualitative variables. An example of a mask is presented in Equation (1).

x t	u_1	u_2	u_3	u_4	y_1	
t - 2δt	-1	0	0	-2	0	(1)
t - δt	0	0	0	0	-3	
t	0	-4	0	0	+1	

Each negative element in the mask is called a m-input (mask input). It denotes a causal relation with the output, i.e. it influences the output up to a certain degree. The enumeration of the m-inputs is immaterial and has no relevance. The single positive value denotes the output. The mask of Equation (1) contains four m-inputs. In position notation, it can be written as (1,4,10,12,15), enumerating the mask cells from top to bottom and from left to right. In this example, the first and second m-inputs, i_1 and i_2, correspond to the input variables u_1 and u_4 two sampling intervals back, whereas the third m-input, $i_{3,}$ refers to the output variable y_1 one sampling interval into the past, etc.

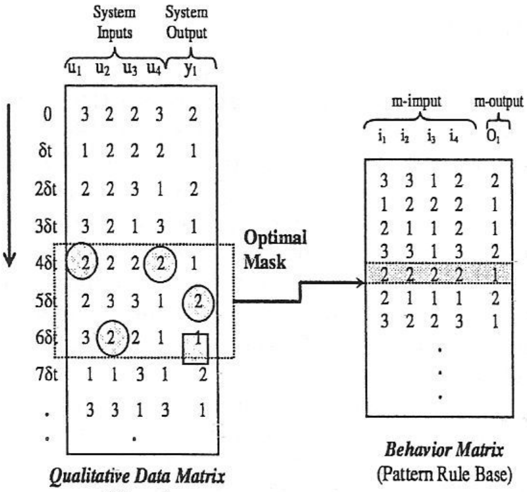

Figure 3: FIR Pattern rule base obtaining

The *qualitative model identification* process evaluates all the possible masks and concludes which one has the highest prediction power by means of the quality of the mask Q, based on an entropy reduction measure. The mask with the maximum Q value is the optimal mask. Once the best mask has been identified, it can be applied to the qualitative data obtained from the system, resulting in a particular pattern rule base.

How is the pattern rule based obtained from the mask? This process is illustrated in Fig. 3. The mask can be used to 'flatten' dynamic relationships into pseudo-static relationships. The left side of Fig. 3 shows an excerpt of the qualitative data matrix that stores the class values. It shows the numerical rather than the symbolic class values. In the example shown in Fig. 3, all the variables were discretized into three classes, except variable y_1, that has been discretized into two classes. The dashed box symbolizes the mask that is shifted downwards along the class value matrix. The round shaded 'holes' in the mask denote the positions of the m-inputs, whereas the square shaded 'hole' indicates the position of the m-output. The class values are read out from the class value matrix through the 'holes' of the mask, and are placed next to each other in the behavior matrix that is shown on the right side of Fig. 3. Here, each row represents one position of the mask along the class value matrix. It is lined up with the bottom row of the mask. Each row of the behavior matrix represents one pseudo-static qualitative state or qualitative rule (also called pattern rule). For example, the shaded rule of Fig. 3 can be read as follows: 'If all the m-inputs (i_1, i_2, i_3, i_4,) have a value of 2 (corresponding to *medium*) then the m-output, O_1, assumes a value of 1 (corresponding to *high*)'.

Once the FIR model is available, the prediction system can take place using the FIR inference engine. This process is called *fuzzy forecast*. FIR inference engine is a specialization of the k-nearest neighbor rule, commonly used in the pattern recognition field. *Defuzzification* is the inverse process of fuzzification. It allows converting the qualitative predicted output into quantitative values that can then be used as inputs to an external quantitative model. For a deeper and more detailed insight into the FIR methodology, the reader is referred to (Nebot 1994).

III. LEARNING THE FUZZIFICATION PARAMETERS OF FIR USING GENETIC ALGORITHMS

GAs are search and optimization techniques based on formalization of natural genetics (Holland 1975; Michalewicz 1996). The main aspects to be considered in the implementation of a GA are: (*A*) genetic representation, (*B*) initial gene pool, (*C*) fitness or objective function, (*D*) genetic operators and (*E*) genetic parameters. These points are highly important to achieve a good performance of the algorithm.

A. Genetic Representation

In order to define a useful chromosome codification, it is necessary to go deeply into the *fuzzification* process of the FIR methodology.

The most common shapes for the membership functions in FIR are triangular or gaussian (default). Fig. 2 illustrates the process of fuzzification by means of an example. As mentioned earlier, a quantitative value is fuzzified into a qualitative triple, i.e. the class, membership and side values.

In Fig. 2 a temperature of 23 degrees centigrade would hence be fuzzified into the class 'normal' with a side value 'right' and a fuzzy membership value of 0.755. Most fuzzy inference approaches preserve the total knowledge by associating with each quantitative data value multiple fuzzy rules consisting of tuples of class and membership values. They would thus represent the temperature of 23 degrees centigrade as being 'normal' with likelihood 0.755 and being 'warm' with likelihood 0.20. FIR accomplishes the same by associating with each quantitative data value a single fuzzy rule consisting of a quantitative triple. Then, in FIR methodology the queues of the membership functions are discarded and only the part of the membership functions in the range [0.5..1] are used. The point where two neighboring classes match with a membership value of 0.5 is named *landmark*. In the example of Fig. 2 the membership function of the class *Normal* is defined by landmarks {13,27}, being this pair the temperature values that specify the limits between the class *Normal* and its adjacent classes, *Fresh* and *Warm*, respectively.

Consequently, the fuzzification parameters of the FIR methodology are the number of fuzzy sets (classes) or granularity level per variable and the membership functions that define its semantics (identified by the landmarks). These are the parameters that the GA should optimize. Therefore, each chromosome *(C)* is composed of two parts:

- *Number of classes (C_1):* The number of linguistic terms for N variables is codified using a vector of N integers in the range [2..9]. The values of the genes are forced to remain in this interval, so the genetic operators must observe this requirement.
- *Membership functions (C_2):* The genetic representation chosen takes into account the number of samples registered for each variable. A specific variable is represented by the proportion of data samples that contains each class, codified in the range [0..1]. An example of chromosome representation for a unique variable that has 4 classes could be (0.3,0.4,0.1,0.2), meaning that the membership function of the first class contains the 30% of the data samples available for this variable, and the second, third and forth membership functions contain 40%, 10% and 20% of the data records, respectively. Of course, the sum of the proportions for each variable must be 1. The minimum proportion, V_{min}, is established to 0.05 and the maximum proportion, V_{max}, is defined by $V_{max} = 1 - V_{min} * (N_{label} - 1)$, where N_{label} is the number of classes of the variable. A clear advantage of this representation is the facility to compute the landmarks from it. This is done by the following steps:

1) The observed trajectory values of each variable are sorted in ascending order.

2) The sorted vector is then split into segments (as many

segments as classes have been determined for that variable) that contain the proportion of values determined by the GA solution.

3) Finally, the landmarks are chosen anywhere between the extreme values of neighboring segments, i.e. using the arithmetic mean values of neighboring observed data points in different segments.

A full chromosome representation, C, is defined by the ensemble of the representations of the number of classes, C_1, and the membership functions, C_2, of each system variable.

$$C = C_1 C_2$$

Therefore, if we denote by E_i the number of classes for the variable i, the number of classes representation for a system of N variables, C_1, is defined by:

$$C_1 = (E_1, E_2, \ldots, E_N)$$

Also, if we denote by D_{ij} the data proportion of the variable i and class j, and C_{2i} the information of the data proportion for all the classes of the variable i, the membership representation, C_2, for a system of N variables (including inputs and outputs), is defined by:

$$C_2 = (C_{21}, C_{22}, \ldots, C_{2N})$$

where,

$$C_{2i} = (D_{i1} \ldots D_{iEi})$$

Note that each time the number of classes and/or distribution of the landmarks changes due to the action of the genetic operators, it is mandatory to re-compute the new fuzzy partition.

B. Initial Gene Pool

The initial population is composed by four groups with the same number of individuals each, except the first one. No repeated chromosomes are allowed. The generation of the initial gene pool is described next.

1) In the first group, each chromosome has the same number of classes in all its variables and the membership functions are uniformly distributed across the variable working range (EFP Method).

2) In the second group, each chromosome has different granularity per variable (different values in C_1) and the membership functions are uniformly distributed (EFP Method) as in group one.

3) In the third group, each chromosome has the same number of classes in all its variables and the membership functions are no-uniformly distributed across the variable working range (the data proportion is generated randomly).

4) In the last group, each chromosome has different number of classes per variable, as in group two, and the membership functions are established in the same way as in the third group.

Although GAs have proven to be robust and get good solutions starting from randomly generated populations (group four), a quick convergence can be obtained using the knowledge available about the problem to sample the population in a biased way, achieving at the same time an appropriate diversity.

C. Fitness or Objective Function

The evaluation of the chromosomes is done following the next steps:

1) Decode the information of the chromosome, building the associated fuzzy partition in the FIR structures.

2) Execute the *qualitative model identification* process of the FIR methodology with the training data set, using the partition built in the previous step. Therefore, the mask associated to that partition with the highest quality measure is obtained.

3) Compute the objective function. In this research two objective functions are proposed: a) the *quality of the optimal mask* or b) the *prediction error of part of the training data set*.

As has been explained earlier, in the qualitative model identification process of the FIR methodology, the optimal mask (i.e. the best model structure) is identified by means of a quality measure, Q. The quality of a mask is a value between 0 and 1, where 1 indicates the highest quality. Therefore, the first cost function proposed is 1-Q, due to the fact that the algorithm task is to minimize the cost function.

The second cost function is defined as the prediction error of a portion of the training data set. The normalized mean square error in percentage (MSE), given in Equation (2), is used for this purpose,

$$MSE = \frac{E\left[\left(y(t) - \hat{y}(t)\right)^2\right]}{VAR\left[y(t)\right]} .100\% \qquad (2)$$

where $\hat{y}$ *(t)* is the predicted output, *y(t)* the system output and VAR denotes variance. The idea is to use part of the training data set to identify the model and the rest of the data set to evaluate the prediction performance of that model. It is important to remember that the FIR model is composed of the optimal mask and the pattern rule base (behavior matrix). Therefore, both must be generated in the evaluation process of a certain fuzzy partition when this cost function is used. Moreover, the fuzzy forecasting process of the FIR methodology needs to be executed to obtain the cost of the evaluated chromosome. Thus, the computational cost of this evaluation function is considerably higher than the one obtained with the cost function that only depends on the quality of the mask. However, the prediction accuracy should be higher. The size of the portion of the training data set used for cost function evaluation purposes is defined with respect to the size of the whole training data set.

D. Genetic Operators

Due to the special nature of the chromosomes involved in the optimization process, the genetic operators become an important aspect of the GA. Since there is a strong relationship between the two chromosome parts

(C_1 and C_2), it is required that the genetic operators work cooperatively in C_1 and C_2 in order to take advantage of the representation used.

Taking into account these aspects, the following operators are considered:

1) *Selection*: The selection probability calculation follows linear ranking (Baker 1985). Chromosomes are sorted in fitness order and selection probability of each chromosome, $p_s(C_i)$, is computed according to its rank (with $rank(C_{best})$ = 1), by using the following non-increasing assignment function:

$$p_s(C_i)=\frac{1}{NC}\cdot\left(\eta_{\max}-(\eta_{\max}-\eta_{\min})\cdot\frac{rank(C_i)-1}{NC-1}\right) \quad (3)$$

where NC is the population size and η_{min} ∈ [0,1] specifies the expected number of copies for the worst chromosome (the best one has $\eta_{max} = 2 - \eta_{min}$ expected copies). In the experiments η_{min}=0.75.

Linear ranking is performed along with Stochastic Universal Sampling proposed by (Baker 1987). This procedure guarantees that the number of copies of any chromosome is bounded by the floor and by the ceiling of its expected number of copies.

Our reproduction operator includes the elitist selection.

2) *Crossover operator:* As regards the recombination process, two different operators are used according to the two parents implied in the crossing:

- *Crossover when both parents have the same granularity level per variable:* If the two parents have the same values in C_1 (each variable has the same number of classes in the two parents), the genetic search has located a promising space zone that has to be adequately exploited. This task is developed by applying the non-uniform arithmetic crossover operator in C_2 and maintaining the parent C_1 values in the offspring. This crossover operator is proposed in (Michalewicz 1996) and works in the way described subsequently.

This operator generates two offspring as a weighted mean of the parent values. A real value, u, in the range [0..1] is selected randomly and used to compute the new offspring by means of Equation (4).

$$\begin{aligned} C_t &= u\cdot father+(1-u)\cdot mother \\ C_t' &= (1-u)\cdot father+u\cdot mother \end{aligned} \quad (4)$$

An advantage of the crossover operator selected is that assures the validity of the offsprings obtained, i.e. the sum of the data proportion for all the classes of each variable is 1.

- *Crossover when the parents encode different granularity levels:* This second case highly recommends the use of the information encoded by the parents for exploring the search space in order to discover new promising zones. Hence, when C_1 is crossed at a certain point, the values in C_2 corresponding to the crossed variables are also crossed in the two parents. In this way, a standard crossover operator is applied over the two parts of the chromosomes. This operator performs as follows: a crossover point p is randomly generated in C_1 and the two parents are crossed at the p-th variable in C_1 and C_2, producing two meaningful descendents.

Let us look at an example in order to clarify the crossover application. Let

$$C_t=(E_1,\ldots,E_p,E_{p+1},\ldots,E_N,C_{21},\ldots,C_{2p},C_{2p+1},\ldots,C_{2N})$$
$$C_t'=(E_1',\ldots,E_p',E_{p+1}',\ldots,E_N',C_{21}',\ldots,C_{2p}',C_{2p+1}',\ldots,C_{2N}')$$

be the individuals to be crossed at point p. The two resulting offspring are:

$$C_t=(E_1,\ldots,E_p,E_{p+1}',\ldots,E_N',C_{21},\ldots,C_{2p},C_{2p+1}',\ldots,C_{2N}')$$
$$C_t'=(E_1',\ldots,E_p',E_{p+1},\ldots,E_N,C_{21}',\ldots,C_{2p}',C_{2p+1},\ldots,C_{2N})$$

Hence, the complete recombination process will allow the GA to follow an adequate exploration and exploitation rate in the genetic search.

3) *Mutation operator:* Due to the nature of the values stored in the two parts of the chromosome, two different operators are considered. A brief description of them is given below.

- Mutation on C_1: The mutation operator selected for C_1 is similar to the one proposed by (Thrift 1991). In this case, if a mutation on a gene belonging to the first part of the chromosome is going to be performed, a local modification is done by changing the number of classes of the variable to the immediately upper or lower (the decision is made randomly). When the value to be changed is the minimum (2) or the maximum (9), the only possible change is done, i.e. increase or decrease by one the granularity, respectively. Once a new value is selected, a uniform fuzzy partition for this variable is stored in its corresponding zone of C_2.
- Mutation on C_2: Since both parts are based on a real coding scheme, the mutation operator selected for C_2 is also similar to the one proposed by (Thrift 1991). Here, the data proportion associated to the gene of the selected chromosome is increased or decreased (the decision is made randomly) by a factor in-between the range *[V_{min}...MAX]* set, also, randomly. Where MAX=0.5 – V_{min}·(N_{label} – 1). The other proportions of the same variable are adjusted in order to maintain the addition to 1.

When the value to be changed plus the factor get out of the limits of the range *[V_{min}...V_{max}]*, the only possible change is done, i.e. increase or decrease by the proportion factor, respectively.

E. Genetic Parameters

The values of the probabilities have been established according to (Grefenstette 1986). Table 1 shows the values of the parameters applied to this algorithm.

Table 1: Genetic parameters of the GA for the application studied.

Parameter	Value
Population size (# individuals)	50
Crossover Probability	0.6
Mutation Probability	0.1
Stop Criteria (Chromosomes evaluations)	{5000,10000,20000,40000, 80000,160000}

We have considered six stop criteria of the GA: to reach 5000, 10000, 20000, 40000, 80000 and 160000 chromosome evaluations, respectively.

IV. ELECTRICAL DISTRIBUTION NETWORK MODELS

The problem of estimating the maintenance cost of the electric network becomes difficult when we deal with medium and low voltage lines. Maintenance cost depends among other factors on the total length of electrical line each company owns, and on its kind, i.e. high, medium and low voltage (Cordón et al. 1999). To justify the distribution expenses of the companies, models of the length of the line are used. Although high voltage lines can be easily measured, this is not the case for medium and low voltage lines. These lines are contained in cities and villages, and it is very difficult and expensive to measure them, due to the fact that they have been installed incrementally, according to its own electrical needs in each moment. Therefore, it is necessary to handle the problem from the modeling perspective.

We were provided with 1059 data samples of Spanish towns (Cordón et al. 1999, Cordón et al. 1998). Four characteristics of each town correspond to the input variables, i.e. the sum of the lengths of all streets in the town (SLS) in Km, the total area of the town (TA) in Km^2, the area that is occupied by buildings (AB) in Km^2 and the energy supply to the town (ES) in MWh. The maintenance cost of the medium voltage line (MC) in Millions of pesetas is the output variable.

In the previous works (Cordón et al. 1999, Cordón et al. 1998), the available data was divided into the training set (first 847 towns) and the test set (last 212 towns), corresponding to the 80% and 20% of the whole data set, respectively. The same data distribution is used in the present study in order to compare the results obtained in an accurate way. For the same reason, the medium square error (SE) used in (Cordón et al. 1999) and described in Equation (5) is used for the computation of each model prediction error.

$$SE = \frac{1}{2 \cdot N} \sum_{i=1}^{N} (y_i(t) - \hat{y}_i(t))^2 \qquad (5)$$

where, ŷ(t) is the predicted output, y(t) the system output and *N* the number of samples.

It is interesting to notice that no temporal relation exists between two consecutive samples of the five system variables, due to the fact that each sample represents a specific town. This is the first time that FIR methodology is used to deal with a non dynamical system. However, this is solved easily by forbidding temporal relations between the system variables.

A. Previous Works

Table 2 contains the SE prediction errors achieved when classical methods and hybrid evolutionary techniques are used for the same problem (Cordón et al. 1999, Cordón et al. 1998). As regards classical methods, Cordón, Herrera and Sánchez have considered linear models fitted by linear least squares, second order polynomial models fitted by nonlinear least square and three-layer-perceptron neural network (of 4-5-1 neurons). The minimization error algorithm was the conjugate gradient. With respect to other techniques, they studied Genetic Fuzzy Rule-Based Systems (GFRBS) for the optimization of three different fuzzy models, i.e. Wang-Mendel (WM), Mamdami and Takagi-Sugeno-Kang (TSK). Finally, they use two hybrid algorithms, GA-P and Interval GA-P, that combine the traditional genetic algorithms (GA) with the genetic programming (GP) paradigm (Howeard and D'Angelo 1995). The Interval GA-P is a modified version of the GA-P method that uses interval values instead of punctual ones. All the methodologies use the same training and test data sets explained previously.

Table 2: Prediction errors (SE) obtained by classical methods and hybrid evolutionary techniques

Method	SE_{train}	SE_{test}
Linear	164662	36819
Second-order polynomial	103032	45332
Three-layer perceptron 4-5-1	86469	33105
GA-P	18168	21884
Interval GA-P	16263	18325
WM fuzzy model	20318	27615
Mamdani fuzzy model	19679	22591
TSK fuzzy model (α = 0)	25579	26450
TSK fuzzy model (α = 0,2)	11074	11836

The first column of Table 2 describes the method evaluated, the second and third columns show the prediction errors using the SE formula (described in Equation (5)), of the training and test data sets, respectively. As can be seen from this table the GA-P techniques and fuzzy models outperform again classical linear and non linear regression methods as well as neural networks. The TSK fuzzy model has obtained the best result. A more detailed discussion of the results presented in Table 2 can be found in (Cordón et al. 1999). These values are taken in this paper as reference errors to study the performance of the FIR methodology in the same problem.

B. Learning the Number of Classes for each System Variable and Membership Functions of the Classes

This section presents the results obtained when the GA proposed for learning the number of fuzzy sets (classes) per variable and the membership functions that define its semantics is evaluated for the problem at hand. Thirty executions are preformed for each objective function and stop criteria.

The results are presented in two sections, based on the objective function used for the evaluation of the chromosomes.

1 - Q objective function

Table 3 shows the results obtained when 1-Q was used as objective function. The table is organized as follows. The first column is divided in 2 sections. Section *A*

Table 3: Number of Classes and Membership Functions results of the electrical distribution problem using 1-Q cost function.

	# eval	Partition	Data Proportion	Opt. Mask	Q	1-Q	SE_{train}	SE_{test}

A	160000	(2,7,2,2,2)	SLS:(0.91,0.09)
			TA:(0.14,0.14,0.14,0.14,0.14,0.15,0.15)
			AB: (0.94,0.06)
			ES:(0.93,0.07)
			MC:(0.92,0.08)
				(1,3,4,5)	0,9638	0,0362	927	2728

	80000	(7,6,6,3,2)	SLS:(0.12,0.12,0.12,0.12,0.13,0.25,0.14)
			TA:(0.13,0.13,0.14,0.14,0.33,0.13)
			AB: (0.73,0.06,0.06,0.05,0.05,0.05)
			ES:(0.83,0.09,0.08)
			MC:(0.90,0.10)
				(3,4,5)	0,9777	0,0223	1080	2759

	40000	(2,5,3,2,2)	SLS:(0.91,0.09)
			TA:(0.48,0.12,0.13,0.13,0.14)
			AB: (0.88,0.06,0.06)
			ES:(0.93,0.07)
			MC:(0.92,0.08)
				(1,3,4,5)	0,9721	0,0279	927	2729

	20000	(8,3,6,2,2)	SLS:(0.11,0.11,0.11,0.11,0.12,0.18,0.13,0.13)
			TA:(0.34,0.33,0.33)
			AB: (0.74,0.06,0.05,0.05,0.05,0.05)
			ES:(0.92,0.08)
			MC:(0.90,0.10)
				(3,4,5)	0,9559	0,0441	1080	2759

	10000	(7,4,3,3,2)	SLS:(0.15,0.13,0.13,0.15,0.13,0.17,0.14)
			TA:(0.25,0.25,0.25,0.25)
			AB:(0.48,0.20,0.32)
			ES:(0.48,0.19,0.33)
			MC:(0.54,0.46)
				(3,4,5)	0,9503	0,0497	1103	5159

	5000	(2,5,5,2,2)	SLS:(0.29,0.71)
			TA:(0.20,0.20,0.20,0.20,0.20)
			AB:(0.62,0.17,0.11,0.05,0.05)
			ES:(0.84,0.16)
			MC:(0.90,0.10)
				(3,4,5)	0,9423	0,0577	1080	2759

B	160000	(9,8,6,2,2)	SLS: (0.11,0.11,0.11,0.11,0.11,0.11,0.11,0.11,0.12) TA: (0.11,0.13,0.11,0.11,0.12,0.12,0.18,0.12) AB: (0.11,0.14,0.19,0.13,0.12,0.31) ES: (0.46,0.54) MC: (0.53,0.47)	(3,4,5)	0,8934	0,1066	1109	5136

	80000	(2,8,2,2,2)	SLS:(0.53,0.47)
			TA:(0.14,0.10,0.16,0.10,0.10,0.10,0.10,0.20)
			AB: (0.49,0.51)
			ES:(0.47,0.53)
			MC:(0.54,0.46)
				(1,3,4,5)	0,8979	0,1021	949	5125

	40000	(4,5,5,2,2)	SLS:(0.08,0.69,0.14,0.09)
			TA:(0.33,0.13,0.13,0.3,0.11)
			AB: (0.15,0.19,0.21,0.17,0.28)
			ES:(0.47,0.53)
			MC:(0.48,0.52)
				(3,4,5)	0,8821	0,1179	1123	5119

	20000	(6,7,8,8,2)	SLS:(0.13,0.13,0.14,0.14,0.15,0.31)
			TA:(0.14,0.14,0.14,0.14,0.14,0.15,0.15)
			AB: (0.08,0.38,0.29,0.05,0.05,0.05,0.05,0.05)
			ES:(0.15,0.12,0.12,0.12,0.15,0.13,0.10,0.11)
			MC:(0.90,0.10)
				(3,5)	0,8848	0,1152	199450	225848

	10000	(5,6,8,7,3)	SLS:(0.25,0.23,0.06,0.23,0.23)
			TA:(0.16,0.16,0.17,0.17,0.17,0.17)
			AB:(0.07,0.14,0.16,0.33,0.08,0.08,0.07,0.07)
			ES:(0.14,0.14,0.14,0.14,0.14,0.15,0.15)
			MC:(0.87,0.05,0.08)
				(3,5)	0,8548	0,1452	206931	210251

	5000	(6,9,9,2,2)	SLS:(0.16,0.16,0.17,0.17,0.17,0.17)
			TA:(0.10,0.10,0.10,0.10,0.12,0.11,0.10,0.17,0.10)
			AB:(0.10,0.36,0.10,0.09,0.05,0.09,0.08,0.06,0.07)
			ES:(0.50,0.50)
			MC:(0.91,0.09)
				(3,5)	0,8764	0,1236	199988	216862

corresponds to the best result obtained by the GA while Section *B* corresponds to its worse result. The second column shows the number of evaluations made by the GA. The third column indicates the number of classes (granularity level) per variable. The fourth column shows the data proportion for the input variables (SLS, TA, AB, ES) and the output variable (MC). The number of elements of the data proportion corresponds to the number of classes per variable obtained in the previous column. Both, the granularity level per variable and the data proportion are the output of the GA and they constitute the parameters of the *fuzzification* process of the FIR methodology. The fifth column presents the optimal mask, in position notation, encountered by FIR when these parameters are used for the obtaining of fuzzy partitions in FIR. The sixth column corresponds to the quality associated to the optimal mask. The seventh column is the value of the 1-Q objective function. The eighth column indicates the prediction error SE obtained for training data set. The last column shows the prediction error SE obtained for test data set.

MSE_{train} objective function

Table 4 shows the results obtained when the objective function is defined as the prediction MSE of a portion of the training data set. The last 20% of the training signal is used for objective function evaluation and only the

Table 4: Number of Classes and Membership Functions results of the electrical distribution problem using MSE_{train} cost function.

	# eval	Partition	Data Proportion	Opt. Mask	Q	MSE_{train}	SE_{train}	SE_{test}
A	160000	(2,4,5,7,8)	SLS:(0.10,0.90) TA:(0.21,0.24,0.27,0.28) AB: (0.19,0.33,0.31,0.09,0.08) ES: (0.17,0.20,0.16,0.17,0.11,0.14,0.05) MC: (0.13,0.11,0.11,0.12,0.15,0.09,0.06,0.23)	(1,3,4,5)	0.4779	0.1090	932	2936

	80000	(2,7,5,5,8)	SLS:(0.10,0.90)
			TA:(0.11,0.11,0.11,0.17,0.12,0.12,0.26)
			AB: (0.12,0.4,0.24,0.10,0.14)
			ES:(0.25,0.21,0.25,0.22,0.07)
			MC:(0.13,0.11,0.11,0.12,0.19,0.11,0.13,0.10)
				(1,3,4,5)	0.4953	0.1116	943	8252

	40000	(2,8,7,9,9)	SLS:(0.08,0.92)
			TA:(0.12,0.12,0.12,0.13,0.13,0.13,0.13,0.12)
			AB: (0.29,0.12,0.12,0.12,0.12,0.11,0.12)
			ES:(0.14,0.07,0.08,0.13,0.08,0.20,0.09,0.16,0.05)
			MC:(0.08,0.11,0.05,0.06,0.07,0.08,0.14,0.18,0.23)
				(1,3,4,5)	0.5080	0.1152	938	3022

	20000	(2,7,7,8,9)	SLS:(0.10,0.90)
			TA:(0.11,0.27,0.11,0.12,0.12,0.13,0.14)
			AB: (0.12,0.12,0.30,0.11,0.11,0.12,0.12)
			ES:(0.11,0.11,0.19,0.11,0.15,0.11,0.11,0.11)
			MC:(0.09,0.09,0.05,0.12,0.15,0.10,0.10,0.19,0.11)
				(1,3,4,5)	0.5217	0.1165	934	3066

	10000	(2,3,5,8,7)	SLS:(0.57,0.43)
			TA:(0.20,0.51,0.29)
			AB:(0.18,0.18,0.18,0.27,0.19)
			ES:(0.18,0.10,0.10,0.10,0.11,0.17,0.12,0.12)
			MC:(0.08,0.08,0.08,0.11,0.28,0.21,0.16)
				(3,4,5)	0.5748	0.1258	1067	3025

	5000	(2,2,7,9,9)	SLS:(0.10,0.90)
			TA:(0.50,0.50)
			AB: (0.32,0.10,0.12,0.07,0.17,0.12,0.10)
			ES:(0.12,0.22,0.06,0.09,0.12,0.12,0.11,0.05,0.11)
			MC:(0.12,0.05,0.15,0.06,0.07,0.18,0.13,0.18,0.06)
				(1,3,4,5)	0.4999	0.1180	942	2976

B	160000	(5,7,7,6,7)	SLS:(0.17,0.22,0.24,0.17,0.20) TA:(0.14,0.14,0.14,0.14,0.14,0.15,0.15) AB: (0.10,0.08,0.11,0.12,0.33,0.13,0.13) ES: (0.09,0.27,0.09,0.21,0.27,0.07) MC: (0.06,0.07,0.30,0.33,0.06,0.10,0.08)	(3,4,5)	0.5357	0.1341	1054	2939

	80000	(4,5,4,9,8)	SLS:(0.25,0.25,0.25,0.25)
			TA:(0.20,0.20,0.20,0.20,0.20)
			AB: (0.18,0.07,0.55,0.20)
			ES:(0.09,0.16,0.09,0.09,0.09,0.09,0.09,0.18,0.12)
			MC:(0.11,0.11,0.18,0.05,0.12,0.12,0.13,0.18)
				(3,4,5)	0.4470	0.1359	1078	2963

	40000	(4,2,4,6,8)	SLS:(0.16,0.33,0.24,0.27)
			TA:(0.54,0.46)
			AB: (0.39,0.16,0.27,0.18)
			ES:(0.06,0.16,0.16,0.36,0.21,0.05)
			MC:(0.05,0.11,0.07,0.12,0.09,0.32,0.05,0.19)
				(3,4,5)	0.4646	0.1353	1061	3007

	20000	(4,3,7,9,7)	SLS:(0.25,0.25,0.25,0.25)
			TA:(0.36,0.32,0.32)
			AB: (0.14,0.14,0.14,0.14,0.14,0.15,0.15)
			ES:(0.06,0.11,0.13,0.15,0.07,0.19,0.14,0.07,0.08)
			MC:(0.18,0.05,0.09,0.11,0.32,0.13,0.12)
				(3,4,5)	0.5390	0.1361	1058	2978

	10000	(4,4,4,3,3)	SLS:(0.11,0.53,0.11,0.25)
			TA:(0.25,0.25,0.25,0.25)
			AB: (0.13,0.17,0.35,0.35)
			ES:(0.45,0.49,0.06)
			MC:(0.15,0.28,0.57)
				(3,4,5)	0.5566	0.1390	1052	3053

	5000	(8,7,2,2,2)	SLS:(0.11,0.12,0.12,0.12,0.14,0.14,0.13,0.12)
			TA:(0.11,0.11,0.12,0.12,0.28,0.13,0.13)
			AB: (0.52,0.48)
			ES:(0.50,0.50)
			MC:(0.35,0.65)
				(3,4,5)	0.6616	0.1847	1073	7896

first 80% of the signal is used to obtain the FIR models (masks and pattern rule bases). The table is organized as Table 3. The only difference is that the seventh column contains the values of the MSE_{train} instead of the 1-Q objective cost function.

As expected, the CPU time needed by the GA when the MSE_{train} objective function is used is clearly grater than the time needed when the 1-Q is used. For example, the computational time needed to perform 30 executions for 160000 evaluations when the 1-Q and MSE_{train} objective functions are studied are 61:16 and 132:17 hours, respectively, in a Pentium III computer (0.6 GHz).

The errors obtained by FIR methodology in this application are significantly lower than the ones obtained by the methodologies of Table 2. The best result of 11836 SE obtained by the TSK fuzzy model is much bigger than the 2728 SE obtained by FIR methodology enhanced by the GA. From Tables 3 and 4 it can be also seen that the results obtained by both cost functions are equivalent. In this case the performance of FIR models when the MSE_{train} cost function is used is not superior to the performance of 1-Q objective function. Therefore, the 1-Q objective function is preferable because of its lower computational cost.

Fig. 4 presents the best predictions obtained by AG3 for the two cost functions studied. The top plot shows real and predicted test signals of the partition (2,7,2,2,2) when the 1-Q objective function is used, whereas the bottom plot shows the prediction error signal when using the partition (2,4,5,7,8) with the cost function MSE_{train}.

The SE errors for the top and bottom plots are 2728 and 2936, respectively. The error values are high due to the range [0..10000] of the maintenance cost variable and that the SE formula do not normalize by the variance. However, as can be observed in the top plot of Fig. 4, the prediction signals obtained by FIR models are able to follow the real maintenance cost signal very accurately.

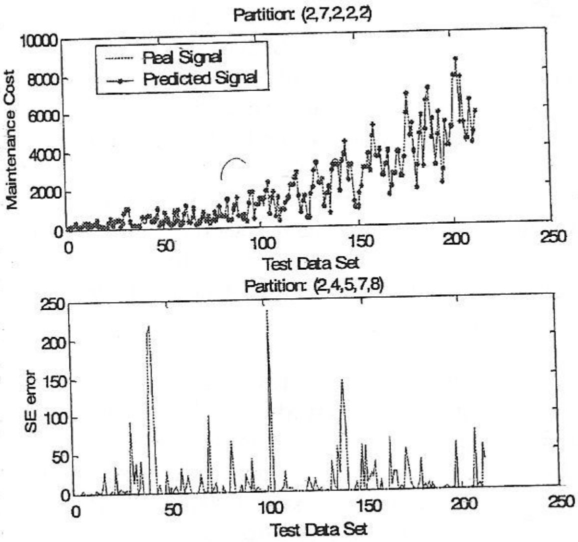

Figure 4: Predictions of the test data set when the best FIR models with 1-Q (top) and MSE_{train} (bottom) cost functions are used.

V. CONCLUSION

A Fuzzy Inductive Reasoning (FIR) model is a qualitative, non-parametric, shallow model based on fuzzy logic. Therefore, variations on fuzzy partitions have a direct effect to the performance of the model identification and prediction processes of FIR methodology. In this paper a new genetic algorithm is developed in the context of FIR in order to optimize the fuzzification parameters. Two objective functions have been evaluated and compared from the perspective of their performance and computational time.

The performance of FIR models were superior when compared with the performance of other methodologies presented in previous works such are linear models, second order polynomial models, neural networks, hybrid genetic programming and different fuzzy models, for the same problem.

REFERENCES

Baker J. E. 1985. "Adaptive selection methods for genetic algorithms". In *Proc. First Int. Conf. on Genetic Algorithms*, J.J. Grefenstette (Ed.). Hillsdale, 101-111.

Baker J. E. 1987. "Reducing bias and inefficiency in the selection algorithm". In *Proc. Second Int. Conf. on Genetic Algorithms (ICGA'87)*, J.J. Grefenstette (Ed.). Hillsdale, 14-21.

Bonissone P.P. 1997. "Soft computing: the convergence of emerging reasoning technologies." *Soft Comput.* 1, no. 1 (Apr.), 6-18.

Chen S.-M.; Y.-J. Horng; and C.-H. Lee. 2001. "Document retrieval using fuzzy-valued concept networks." *IEEE Trans. Syst., Man, Cybern. B* 31, no. 1 (Feb.), 111-118.

Chi Z.; H. Yan; and T. Pham. 1996. "Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition". World Scientific.

Cordón O.; F. Herrera; and L. Sánchez. 1998. "Computing the Spanish Medium Electrical Line Maintenance Costs by Means of Evolution-Based Learning Processes". In *Proc. 11th Int. Conf. on Industrial and Engineering Applications of AI and ES (IEA-98-AIE)* (Castellón, Spain). 478-482.

Cordón O.; F. Herrera; and L. Sánchez. 1999. "Solving Electrical Distribution Problems Using Hybrid Evolutionary Data Analysis Techniques," *Applied Intelligence* 10, 5-24.

Cordón O.; F. Herrera; F. Hoffmann; and L. Magdalena. 2001. "Genetic Fuzzy Systems. Evolutionary tuning and learning of fuzzy knowledge bases". World Scientific.

Dote Y. and S.J. Ovaska. 2001. "Industrial applications of soft computing: a review." *Proc. IEEE* 89, no. 9 (Sept. 2001), 1243-1265.

Driankov D.; H. Hellendoorn; and M. Reinfrank. 1993. *An Introduction to Fuzzy Control.* Springer-Verlag, New York.

Grefenstette J. J. 1986. "Optimization of Control Parameters for Genetic Algorithms." *IEEE Trans. Syst., Man, Cybern.* 16, no. 1 (Jan./Feb), 122-128.

Hirota K. and M. Sugeno. 1995. "Industrial Applications of Fuzzy Technology in The World". World Scientific.

Holland J. 1975. *Adaptation in Natural and Artificial Systems.* The University of Michigan Press, Ann Arbor.

Howeard L. and D. D'Angelo. 1995. "The GA-P: A genetic algorithm and genetic programming hybrid." *IEEE Expert*, 11-15.

Klir G. 1985. *Architecture of Systems Problem Solving.* Plenum Press, N.J.

Kuncheva L. I. 2000. *Fuzzy Classifier Design.* Physica-Verlag, Heidelberg.

Leondes C. T. 1999. *Fuzzy Theory Systems: Techniques and Applications, vols. 1-4.* Academic Press, San Diego, CA.

Michalewicz Z. 1996. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer-Verlag, New York.

Miyamoto S. 1989. "Two approaches for information retrieval through fuzzy associations." *IEEE Trans. Syst., Man, Cybern.* 19, no. 1 (Jan./Feb.), 123-130.

Mugica F. and F. Cellier. 1994. "Automated synthesis of a fuzzy controller for cargo ship steering by means of qualitative simulation". In *Proc. ESM'94, European Simulation MultiConference* (Barcelona, Jun.1-3). 523-528.

Nebot A. 1994. "Qualitative Modeling and Simulation of Biomedical Systems Using Fuzzy Inductive Reasoning". Ph.D. thesis. Dept. Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Barcelona, Spain.

Nebot A.; F. Cellier; and D. Linkens. 1996. "Synthesis of an anaesthetic agent administration system using fuzzy inductive reasoning." *Artificial Intelligence in Medicine* 8, no. 3, 147-166.

Nebot A.; F. Mugica; and P. Gómez. 2001. "Long term prediction of maximum ozone concentration using fuzzy inductive reasoning". In *Proc. EUNITE'01: European Symposium on Intelligent Technologies, Hybrid Systems and their implementation on Smart Adaptive Systems* (Tenerife, Dec. 13-15). 91-101.

Pedrycz W. 1996. *Fuzzy Modelling: Paradigms and Practice.* Kluwer Academic Press.

Sattar F. and D. B. H. Tay. 1999. "Enhancement of document images using multiresolution and fuzzy logic techniques." *IEEE Signal Processing Lett.* 6, no. 10 (Oct.), 249-252.

Suzuki Y.; K. Itakura; S. Saga; and J. Maeda. 2001. "Signal processing and pattern recognition with soft computing." *Proc. IEEE* 89, no. 9 (Sept.), 1297-1317.

Thrift P. 1991. "Fuzzy logic synthesis with genetic algorithms". In *Proc. Fourth Int. Conf. on Genetic Algorithms (ICGA'91)* (San Diego, CA). 509-513.

Vapnik V. 1998. *Statistical Learning Theory.* Wiley & Sons, New York.

JESÚS ACOSTA was born in Coro-Estado Falcón, Venezuela. He received the Licenciatura degree in information engineering (cum laude) in 1995 from the Universidad Tecnológica del Centro (UNITEC), Guacara, and the M.S. degree in process engineering (cum laude) in 2000 from the Universidad Nacional Experimental Politécnica "Antonio José de Sucre" (UNEXPO), Barquisimeto. At the moment, he is pursuing the Ph.D. degree from Technical University of Catalonia (UPC), Barcelona (Spain), in control engineering. He joined the Departament of *Instrumentación Industrial*, IUTAG, in 1996 as an Assistant Professor and since April 1999 he has been Permanent Professor. He performed research in optimization of fuzzy partitions for inductive reasoning at the Soft Computing Group, UPC. His research interests are in areas including control, fuzzy logic, genetic algorithms, pattern recognition and machine learning. Mr. Acosta received a Ph.D. fellowship in 2001 to 2003 from the Agencia Española de Cooperación Internacional (Spain) and is currently a research fellow at the UPC. He is a member of IEEE Student since 2004. His e-mail address is: `jacostav@cantv.net`.

ÀNGELA NEBOT received her B.S. and Ph.D. degrees in computer science from the Polytechnical University of Catalonia (UPC), Barcelona, Spain, in 1988 and 1994, respectively. From 1988 to 1992 she was a research student at the *Institut de Cibernètica* (Spanish Consejo Superior de Investigaciones Científicas), Barcelona, holding a pre-doctoral research grant from the Government of Catalonia and finishing the doctoral courses on software and artificial intelligence. She joined the Department of *Llenguatges i Sistemes Informàtics*, UPC, in 1994 as an Assistant Professor, and since March 1998 she has been Associate Professor in the same department. She is currently the head of the Soft Computing group of the UPC. Her current research interests include fuzzy systems, neuro-fuzzy systems, genetic algorithms, simulation and e_learning. Dr. Nebot is an associate editor of the journal "Simulation: Transactions of the Society for Modeling and Simulation International" and collaborates as a reviewer with other international journals such are "IEEE Transactions on Fuzzy Systems", "International Journal of General Systems", "Neurocomputing", "Artificial Intelligence in Environmental Engineering" and "Artificial Intelligence Communication". Her e-mail address is: `angela@lsi.upc.edu`.

PEDRO VILLAR was born in Granada, Spain, on August 15, 1968. He received his M. S. degree in Computer Science in 1991 from the University of Granada, Spain and his Ph. D. in Computer Science in 2000 from the University of Vigo, Spain. He joined the Department of *Lenguajes y Sistemas Informáticos* at the University of Vigo in October 1991, as an Assistant Professor, and since May 2002 until October 2004 he was an Associate Professor in the same department. Now, he is an Assistant Professor in the Department of Software Engineering at the University of Granada. His current main research interests are genetic fuzzy systems, evolutionary algorithms and multi-objective genetic algorithms. His e-mail address is: `pvillarc@ugr.es`.

JOSEP M. FUERTES was born in Barcelona, received his Industrial Engineer degree in 1976, his PhD degree in 1986, and became Permanent Professor at the Technical University of Catalonia (UPC) in 1987. He was *Researcher* (1975-1986) at the Institut de Cibernètica (Spanish Consejo Superior de Investigaciones Científicas). In 1987 he got a position for a year at the Lawrence Berkeley Laboratory working as *Visiting Scientific Fellow* in the design of the Active Control System of the W.M.Keck 10 meter segmented telescope (Hawaii). From 1992 to 1999 he was the director of the University Research Line in Advanced Control Systems. From 2001 to 2005 he was the director of the Department of *Enginyeria de Sistemes, Automàtica i Informàtica Industrial*, also at UPC. Dr. Fuertes has collaborated as organizer (IEEE-WFCS'97, IEEE-ETFA'99), chairman, session organizer, or member of program committee in several international conferences and has participated with many papers in the areas of Intelligent Control Systems and of Distributed Control Systems and Applications. From 1989 he has coordinated projects at national and international level, related with the above areas of expertise. He acted (1996-2001) as Spanish *Representative* at the Council of the European Union Control Association (EUCA). He is member of the ADCOM at the IEEE IES (Industrial Electronics Society); He is also member of CEA-IFAC (Comité Español de Automática, International Federation of Automatic Control) and other scientific and technical societies. His e-mail address is: `josep.m.fuertes@upc.edu`.

International Mediterranean Modelling Multiconference

I3M

2006

October, 4-6, 2006
Barcelona, Spain

EDITED BY
Agostino G. Bruzzone
Antoni Guasch
Miquel Angel Piera
Jerzy Rozenblit

LOGISIM

International Mediterranean & Latin American
Council of Simulation

Liophant Simulation

McLeod Institute of Simulation Science
MISS Spanish & Genoa Centers

CEA
comité español de automática
Comité Español de Automática

DIP
Genoa
University

Universitat
Autònoma
de Barcelona

UPC
Technical
University of
Catalonia

ISBN 84-690-0726-2