International Journal of General Systems
Vol. 36, No. 6, December 2007, 703–731

Taylor & Francis
Taylor & Francis Group

Learning fuzzy partitions in FIR methodology

JESUS ACOSTA†, ANGELA NEBOT‡*, PEDRO VILLAR¶ and JOSEP M. FUERTES§

†Departamento de Instrumentación Industrial, Instituto Universitario de Tecnología Alonso Gamero, 4101-Coro, Estado Falcón, Venezuela
‡Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya 08034, Barcelona, Spain
¶Departamento de Lenguajes y Sistemas Informáticos, Universidad de Granada 18071, Granada, Spain
§Departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial, Universitat Politècnica de Catalunya 08028, Barcelona, Spain

(Received 5 March 2007; in final form 30 April 2007)

The main goal of this research is the development of a hybrid genetic fuzzy system (GFS), composed by the fuzzy inductive reasoning (FIR) methodology and a genetic algorithm (GA) that is responsible of learning the fuzzy partitions needed in the recode process of FIR. A partition includes the number of fuzzy sets (classes) per variable and the membership function of each class. The resulting GFS is applied to two real problems, i.e. the estimation of the maintenance cost of medium voltage lines in Spanish towns and the prediction of ozone levels in Austria. The results obtained in each application are compared with some of the most popular classical statistical modeling methods, neural networks and other hybrid evolutionary data analysis techniques.

Keywords: *Genetic fuzzy systems; Fuzzy inductive reasoning; Genetic algorithms; Electric distribution networks; Ozone prediction*

1. Introduction

Fuzzy systems have demonstrated their ability to solve different kinds of problems like control (Driankov *et al.* 1993, Leondes 1999), modeling (Pedrycz 1996) or classification (Chi *et al.* 1996, Vapnik 1998, Kuncheva 2000), 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 (Miyamoto 1989, Chen *et al.* 2001), industrial applications (Hirota and Sugeno 1995, Leondes 1999, 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 like neural networks (NNs), evolutionary

*Corresponding author. Email: angela@lsi.upc.edu

International Journal of General Systems
ISSN 0308-1079 print/ISSN 1563-5104 online © 2007 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/03081070701458548

algorithms or probabilistic methods. Neural fuzzy systems (NFSs) and genetic fuzzy systems (GFSs) are the most successful approaches of hybrid systems within soft computing techniques (Bonissone 1997, Cordón *et al.* 2001). NFS and GFS hybridize the approximate reasoning method of fuzzy systems with the learning capabilities of NNs and evolutionary algorithms, respectively.

There are numerous studies on both subjects. However, neuro-fuzzy systems (Jang *et al.* 1997) have been used in a larger number of applications, in particular in the industrial area. This paper is focused on GFSs (Cordón *et al.* 2001) and, therefore, we review here some of those results that are more related to the work presented in this research.

In Cordón *et al.* (2004), the authors present an excellent overview of the research done in the last 10 years in the field of GFSs. As described in the paper, the most prominent types of GFSs are genetic fuzzy rule based systems (GFRBSs), whose genetic processes learn or tune different components of a fuzzy rule base system, i.e. scaling functions (Gudwin *et al.* 1998, Magdalena 1999, Hoffmann 2001), membership functions (Herrera *et al.* 1995, Gürocak 1999, Hoffmann 2001, Casillas *et al.* 2005), rule bases (Ishibuchi *et al.* 1999, González and Pérez 2001, Hoffmann 2001, Camargo *et al.* 2004, Carmona *et al.* 2004, Del Jesús *et al.* 2004) or knowledge bases (Heider and Drabe 1997, Hoffmann and Pfister 1997, Camargo *et al.* 2004, Pomares *et al.* 2004). In Gudwin *et al.* (1998) the use of contextual transformation functions to adjust membership functions is introduced. The fine tuning of membership functions is critical when evaluating the effectiveness of fuzzy systems in control, modeling or classification problems. Linear context adaptation is simple and fast, but the membership functions obtained are uniformly distributed. Non-linear context adaptation is more computationally expensive, but the membership functions can be stretched or expanded to best represent concepts in real environments, e.g. higher sensitivity in extreme classes or in middle classes. In that work a genetic algorithm (GA) was used to find a non-linear transformation function given the base membership functions and a set of data available from the application studied. In Magdalena (1999), the author proposed a GA to learn the rule base and the gain and sensitivity of fuzzy logic controllers by means of scaling functions. Hoffmann (2001) describes two applications of GFSs, an evolutionary strategy that tunes the scaling and membership functions of a fuzzy cart-pole balancing controller and a GA that learns the fuzzy control rules for an obstacle-avoidance behavior of a mobile robot. Casillas *et al.* (2005) presents a genetic tuning process for jointly fitting the fuzzy rule symbolic representations and the meaning of the involved membership functions. The good performance of this proposal mainly lies in the tuning approach performed at two different levels of significance. In Herrera *et al.* (1995) and Gürocak (1999), genetic-algorithm-based methods are described to alter the shapes of the fuzzy sets by shifting their peak location. In these studies, it is assumed that the rule base and the fuzzy sets are already defined. The research presented in Ishibuchi *et al.* (1999) and González and Pérez (2001) deals with the automatic generation of fuzzy if-then rules by means of genetic methods. In Ishibuchi *et al.* (1999) fuzzy if-then rules are obtained for pattern classification problems. This work uses fixed membership functions and therefore, no tuning mechanism is applied to them. In González and Pérez (2001) different search strategies (GAs, simulated annealing and hill-climbing) are analyzed to find the best fuzzy rules that describe the system under study. In Del Jesús *et al.* (2004) an Adaboost algorithm for the same task is proposed. Another strategy is identifying fuzzy modes from certainty degrees, as studied in Carmona *et al.* (2004). Hoffmann and Pfister (1997) and Heider and Drabe (1997) present two genetic perspectives

for the learning of fuzzy knowledge bases. In Heider and Drabe (1997) a cascaded GA is introduced with the idea of splitting the fuzzy system design process into optimization of the structure and of the parameters. This algorithm is tested on a fuzzy controller design task. In Pomares *et al.* (2004) a novel approach to achieve global learning in fuzzy controllers is proposed.

Although the largest number of research efforts have been reported on GFRBSs, other kinds of GFSs, like genetic fuzzy NNs (Russo 1998, Chung *et al.* 2000, Alpaydin *et al.* 2002) and genetic fuzzy clustering algorithms (Hall *et al.* 1994, Van Le 1995, Yuan *et al.* 1995), have also been developed with successful results.

In the research presented in this paper we propose a new GFS to improve the fuzzy inductive reasoning (FIR) methodology. FIR is an inductive modeling and prediction methodology that has been applied to different kinds of problems (e.g. control, biomedicine, ecology), usually obtaining good results (Mugica and Cellier 1994, Nebot *et al.* 1996, 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. The EFP method consists in distributing the system data into a predefined number of classes maintaining the same number of occurrences in each class. However, experience has shown that in some applications, i.e. mainly 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 a good partition 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 that determines FIR recode (or fuzzification) parameters.

The GFS developed is used for model identification of two real problems, i.e. estimating the maintenance cost of medium voltage lines in Spanish towns and prediction of ozone levels in Austria. The results obtained with the new method are compared with the ones obtained by other methodologies in the same applications, i.e. NNs, genetic programming, genetic fuzzy rule base systems, linear models, etc. The material covered in this paper can be found in more depth in the PhD dissertation (Acosta 2006).

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

2. Fuzzy inductive reasoning methodology

The conceptualization of the FIR methodology arises of the general system problem solving approach (GSPS) proposed by Klir and Elias (2002). This methodology of modeling and 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: *recode, optimal mask, prediction* and *regeneration*. Figure 1 describes the processes of the FIR methodology. FIR methodology is currently implemented as a Matlab toolbox, called Visual-FIR.

The *recode* 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 element indicates whether the qualitative value is to the left or to the right of the peak value of the associated membership function (figure 2).

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 *recode* 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.

The *optimal mask* process is responsible for finding causal and temporal relations between variables and therefore for obtaining the best model that represents the system. A FIR model is composed by a structure, called mask and a pattern rule base, named behavior matrix. A mask denotes a dynamic relationship among qualitative variables. An example of a mask is presented in equation (1).

$$\begin{array}{cccccc} & u_1 & u_2 & u_3 & u_4 & y_1 \\ t-2\delta t & -1 & 0 & 0 & -2 & 0 \\ t-\delta t & 0 & 0 & 0 & 0 & -3 \\ t & 0 & -4 & 0 & 0 & +1 \end{array} \tag{1}$$

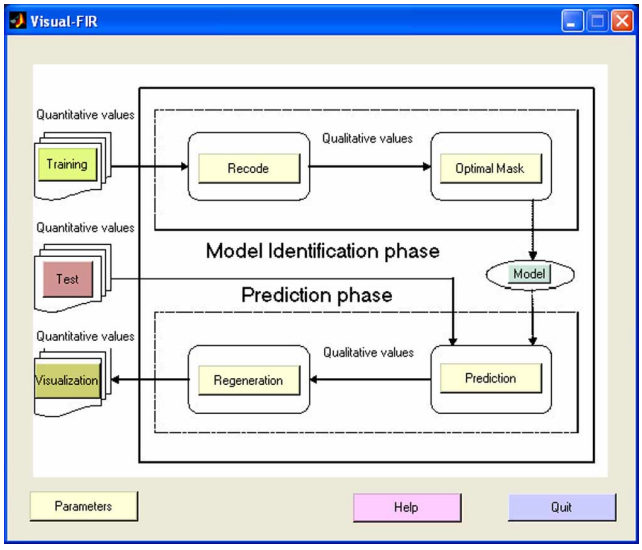

Figure 1. Fuzzy inductive reasoning (FIR) scheme.

Each negative element in the mask is called an *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 left to right and from top to bottom. 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.

The *optimal mask* process evaluates all possible masks and concludes which one has the highest prediction power by means of the quality of the mask Q, based on an entropy 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.

Once the FIR model is available, the prediction system can take place using the FIR inference engine. This process is called *prediction*. FIR inference engine is a specialization of the k-nearest neighbor rule, commonly used in the pattern recognition field. *Regeneration* is the inverse process of recode. 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, Cellier *et al.* 1996).

3. Genetic algorithm for learning FIR fuzzy partitions

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: (1) genetic representation, (2) initial gene pool, (3) fitness or objective function, (4) genetic operators and (5) genetic parameters. These points are highly important to achieve a good performance of the algorithm.

3.1 Genetic representation

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

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

In figure 2 a temperature of 23 degrees centigrade would hence be recoded 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 qualitative triple. Then, in FIR methodology the tails of the membership functions are discarded and only the parts of

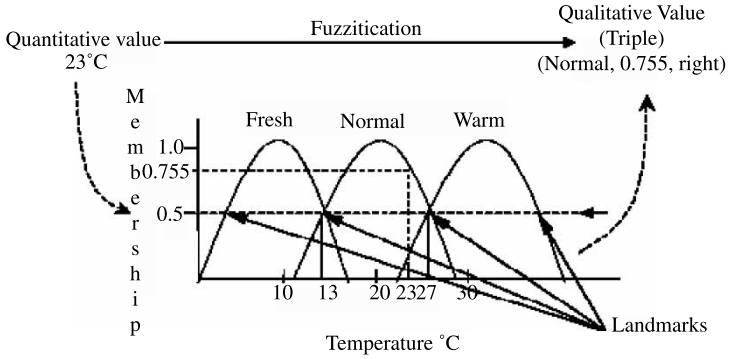

Figure 2. FIR recode (or fuzzification) process of ambient temperature variable.

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 figure 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 recode or fuzzification parameters of the FIR methodology are the number of fuzzy sets (classes) or granularity 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 30% of the data samples available for this variable and the second, third and fourth 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 smallest allowed proportion, $V_{\min}$, is set to 0.05 and the largest proportion, $V_{\max}$, is defined by $V_{\max} = 1 - V_{\min}*(N_{\text{label}} - 1)$, 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.

$$C = C_1 C_2$$

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,

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

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.

3.2 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 chosen randomly), 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 non-uniformly distributed across the variable working range (the data proportion is generated randomly).
(4) In the last group, each chromosome has a 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.

The aim of generating the initial population in this way is to achieve an appropriate diversity. 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.

3.3 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 *optimal mask* process of the FIR methodology with the training data set, using the partition built in the previous step. Therefore, the mask, with the highest quality measure, associated to that partition 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 a part of the training data set*.

As has been explained earlier, in the model identification stage 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,

$$\mathrm{MSE} = \frac{E[(y(t) - \hat{y}(t))^2]}{\mathrm{VAR}[y(t)]} 100\% \tag{2}$$

where $\hat{y}$ (t) is the predicted output, $y(t)$ the system output and VAR denotes variance. The idea is to use a 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 systematic in the evaluation process of a certain fuzzy partition when this cost function is used. Moreover, the prediction 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.

3.4 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 $\mathrm{rank}(C_{\mathrm{best}}) = 1$), by using the following

non-increasing assignment function:

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

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

$$C_t = u\cdot\text{father} + (1 - u)\cdot\text{mother} \quad C'_t = (1 - u)\cdot\text{father} + u\cdot\text{mother} \tag{4}$$

  An advantage of the crossover operator selected is that it 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*: 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. Notice that C_{2i}

"travels" together with its associated granularity, E_i, guaranteeing the preservation of the fundamental requirements of the chosen representation.

(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 $[V_{\min} \ldots \mathrm{MAX}]$ set, also, randomly, where $\mathrm{MAX} = 0.5 - V_{\min}(N_{\mathrm{label}} - 1)$. Remember that N_{label} is the number of classes of the variable. 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 $[V_{\min} \ldots V_{\max}]$, the only possible change is done, i.e. increase or decrease by the proportion factor, respectively.

3.5 Genetic parameters

The values of the probabilities have been established according to Grefenstette (1986). Each application subsection presents the values of the parameters used in the GA.

4. Applications

4.1 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 local electrical needs at 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.* 1998, 1999). Four characteristics of each town are 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 studies (Cordón *et al.* 1998, 1999), the available data was divided into the training set (first 847 towns) and the test set (last 212 towns), corresponding to 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.

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

where, $\hat{y}(t)$ is the predicted output, $y(t)$ the system output, and N is 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 static system. However, this is solved easily by forbidding temporal relations between the system variables, i.e. masks with only one row are allowed.

4.1.1 Previous studies. Table 1 contains the SE prediction errors achieved when classical methods and hybrid evolutionary techniques are used for the same problem (Cordón *et al.* 1998, 1999). With respect to classical methods, Cordón *et al.* have considered linear models fitted by linear least squares, second order polynomial models fitted by nonlinear least squares, and three-layers perceptron NN (of 4-5-1 neurons). The minimization error algorithm was the conjugate gradient method. They also studied GFRBSs for the optimization of three different fuzzy models, i.e. Wang–Mendel (WM), Mamdami and Takagi–Sugeno-Kang (TSK).

Finally, they used two hybrid algorithms, GA-P and interval GA-P, that combine the traditional GA with the genetic programming (GP) paradigm (Howeard and D' 1995). The interval GA-P is a modified version of the GA-P method that uses interval values instead of punctual ones. All of the methodologies use the same training and test data sets explained previously.

The first column of table 1 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

Table 1. Prediction errors (SE) obtained by classical methods and hybrid evolutionary techniques, taken from Cordón *et al.* (1999). Electrical distribution application.

Method	SE_{train}	SE_{test}
Linear	1.64.662	36.819
Second-order polynomial	1.03.032	45.332
Three-layer perceptron 4-5-1	86.469	33.105
GA-P	18.168	21.884
Interval GA-P	16.263	18.325
WM fuzzy model	20.318	27.615
Mamdani fuzzy model	19.679	22.591
TSK fuzzy model ($\alpha = 0$)	25.579	26.450
TSK fuzzy model ($\alpha = 0.2$)	11.074	11.836

models outperform again classical linear and non-linear regression methods as well as NNs. The TSK fuzzy model has obtained the best result. A more detailed discussion of the results presented in table 1 can be found in (Cordón *et al.* 1999). These values are taken in this paper as reference errors to study the performance of the GFS in the same problem.

4.1.2 Fuzzy partitions determination. This section presents the results obtained when the GFS proposed is evaluated for the problem at hand. Thirty executions are preformed for each objective function and stop criteria. Table 2 shows the values of the genetic parameters applied to this problem

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

4.1.2.1. Objective Function $1 - Q$

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 that correspond to the results obtained using the GFS suggested. In row A the best solution obtained for each stop criteria are shown, i.e. the solutions with lowest $1 - Q$ values. Row B presents the worst solution obtained for each stop criteria, i.e. highest $1 - Q$ values. The second column shows the number of evaluations made by the GA. The third column indicates the number of classes (granularity) 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 per variable and the data proportion are the output of the GA, and they constitute the parameters of the recode process of the FIR methodology. The fifth column presents the optimal mask, in position notation, encountered by FIR for this fuzzy partition. 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 last column shows the prediction SE obtained for the test data set.

4.1.2.2. Objective Function $\text{MSE}_{\text{TRAIN}}$

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

As expected, the CPU time needed by the GA when the $\text{MSE}_{\text{train}}$ 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 $1 - Q$

Table 2. Genetic parameters of the GA for the electrical distribution application.

Parameter	*Value*
Population size (# individuals)	50
Crossover probability	0.6
Mutation probability	0.1
Stop criteria (chromosomes evaluations)	{5000, 10000, 20000, 40000, 80000, 160000}

Table 3. Results when the FIR methodology enhanced with a GA is used to learn the granularity and the membership functions. $1 - Q$ cost function. Electrical distribution application.

	# eval	*Granularity*	*Data proportion*	*Optimal mask*	*Q*	*1 − Q*	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	2728
	80000	(7,6,6,2,2)	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	2759
	40000	(2,5,3,5,2,2)	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	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	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	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	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	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)	(1,3,4,5)	0.8979	0.1021	5125

TABLE 3 – *continued*

# eval	*Granularity*	*Data proportion*	*Optimal mask*	*Q*	*1 − Q*	SE_{test}
40000	(4,5,5,2,2)	AB: (0.49,0.51) ES:(0.47,0.53) MC:(0.54,0.46) SLS:(0.08,0.69,0.14,0.09) TA:(0.33,0.13,0.13,0.3,0.11)	(3,4,5)	0.8821	0.1179	5119
20000	(6,7,8,2,2)	AB: (0.15,0.19,0.21,0.17,0.28) ES:(0.47,0.53) MC:(0.48,0.52) 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)	(3,5)	0.8848	0.1152	2,25,848
10000	(5,6,8,2,3)	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) SLS:(0.25,0.23,0.06,0.23,0.23) TA:(0.16,0.16,0.17,0.17,0.17,0.17)	(3,5)	0.8548	0.1452	2,10,251
5000	(6,9,9,2,2)	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) 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)	(3,5)	0.8764	0.1236	2,16,862
		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)				

Table 4. , Results when the FIR methodology enhanced with a GA is used to learn the granularity and the membership functions. MSE_{train} cost function. Electrical distribution application.

	# eval	*Granularity*	*Data proportion*	*Optimal mask*	*Q*	MSE_{train}	SE_{test}
A	160000	(2,4,3,7,8)	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	2936
	80000	(2,7,3,5,8)	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	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	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	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	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	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	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)	(3,4,5)	0.4470	0.1359	2963

TABLE 4 – *continued*

# eval	*Granularity*	*Data proportion*	*Optimal mask*	*Q*	MSE_{train}	SE_{test}
40000	(4,2,4,6,8)	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) SLS:(0.16,0.33,0.24,0.27) TA:(0.54,0.46)	(3,4,5)	0.4646	0.1353	3007
20000	(4,3,7,9,7)	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) SLS:(0.25,0.25,0.25,0.25) TA:(0.36,0.32,0.32)	(3,4,5)	0.5390	0.1361	2978
10000	(4,4,4,3,3)	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) SLS:(0.11,0.53,0.11,0.25) TA:(0.25,0.25,0.25,0.25)	(3,4,5)	0.5566	0.1390	3053
5000	(8,7,2,2,2)	AB: (0.13,0.17,0.35,0.35) ES:(0.45,0.49,0.06) MC:(0.15,0.28,0.57) 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)	(3,4,5)	0.6616	0.1847	7896
		AB: (0.52,0.48) ES:(0.50,0.50) MC:(0.35,0.65)				

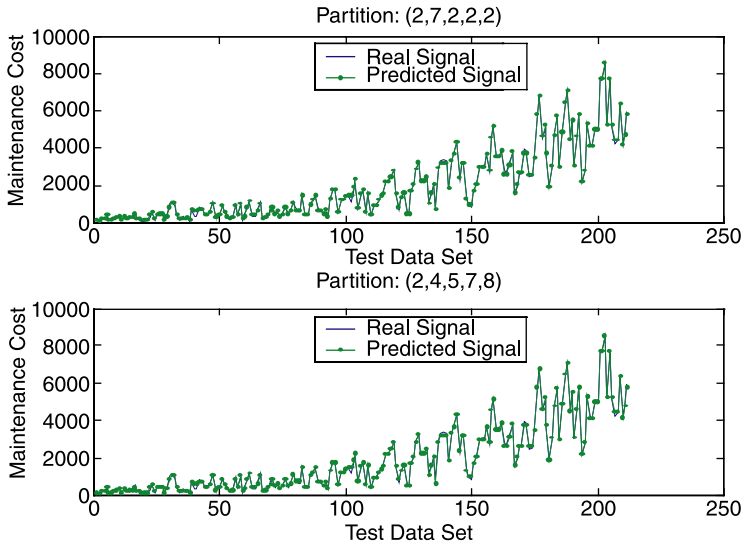

Figure 3. Prediction signals of the test data set obtained by the optimal FIR models identified with the best $1 - Q$ (top) and $\text{MSE}_{\text{train}}$ (bottom) objective functions solutions for the electrical distribution application.

and $\text{MSE}_{\text{train}}$ objective functions are studied, are 61:16 and 132:17 h, respectively, on a Pentium III computer (0.6 GHz). The errors obtained by the GFS in this application are significantly lower than the ones obtained by the methodologies of table 1. The best result of 11,836 SE obtained by the TSK fuzzy model is much larger than the 2728 SE obtained by FIR methodology enhanced by the GA. From tables 3 and 4 it can also be seen that the results obtained by both cost functions are equivalent. In this case the performance of FIR models when the $\text{MSE}_{\text{train}}$ cost function is used is not superior to the performance of the $1 - Q$ objective function. Therefore, the $1 - Q$ objective function is preferable because of its lower computational cost. Figure 3 presents the best predictions obtained for the two cost functions studied. The top plot shows real and predicted test signals obtained when the best solution of the $1 - Q$ objective function is used to find the optimal mask, whereas the bottom plot shows these signals obtained when the best solution of the $\text{MSE}_{\text{train}}$ objective function is used to find the optimal mask. 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 due to the fact that the SE formula has not been normalized by the variance. However, as can be observed in figure 3, the prediction signals obtained by FIR models are able to follow the real maintenance cost signal very accurately, i.e. the real and predicted values are indistinguishable.

4.2 Ozone models

The problem of estimating the ozone levels is important because this toxic gas can produce harmful effects on the population's health such as eye irritation, respiratory problems and aggravation of cardiovascular diseases. In order to provide adequate early warnings, it is valuable to have accurate and reliable forecasts of future high ozone levels. Therefore, the construction of ozone models that capture as precisely as possible the behavior of this gas in the atmosphere is of great interest not only for environmental scientists but also for

government agencies. To deal with the problem under study, we were provided with the data used by Wieland and Wotawa (1999), stemming from the Viennese basin in the East Austrian region. It was registered mostly in the summer season, when ozone, O_3, occurs in highest concentrations. The ozone values (output variable), measured in parts per billon (ppb) are the average of five measurement points providing 3 h fixed average values. The input variables correspond to weather data originated from the weather prediction model of the European Center for Medium Range Weather Forecasts, i.e. the temperature (T) in °K, the cloud cover (CC) that take values ranging from 0 (*no clouds*) to 1 (*completely cloudy*) and the wind speed (WS) in m/s. Ozone and weather data were available for the periods: 07-07-1995–25-09-1995 (81 values), and 01-05-1996–30-09-1996 (149 values).

In previous studies, the available data was divided into the training set (all data from 1996) and the test set (all data from 1995). This distribution of data constitutes the denominated initial partition (Wieland and Wotawa 1999, Gómez *et al.* 2003), and it is used also in the present study for comparison purposes. Also, the root mean square error (RMS) described in equation 6 is used for the computation of each model prediction error.

$$\mathrm{RMS} = \sqrt{\frac{\sum_{i=1}^{N} (y_i(t) - \hat{y}_i(t))^2}{N}} \tag{6}$$

where, $\hat{y}(t)$ is the predicted output, $y(t)$ the system output, and N is the number of samples.

4.2.1 Previous studies. Table 5 contains the best RMS prediction errors achieved by Wieland and Wotawa (1999) and Gómez *et al.* (2003) when different types of NNs are used for the same problem. In Wieland and Wotawa (1999), the NNs used were a multilayer perceptron (MLP), an elman network (EN), and a modified elman network (MEN). The Wieland and Wotawa's NN models were compared with statistical models, IMPO models and Santiago models, showing that the prediction performance of these NNs was better. The IMPO models use a chemical/physical approach developed by the Institute for Meteorology and Physics of the Universität für Bodenkultur Wien (Stohl *et al.* 1996). On the other hand in Gómez *et al.* (2003), the authors used a MEN network with hyperbolic tangent activation function, and a powerful type of recurrent NN called long short term memory (LSTM).

The first column of table 5 describes the method evaluated. The second column stands for the number of hidden units. The third column shows the learning rate. The fourth column indicates whether bias neurons had been used or not. The fifth column gives the necessary

Table 5. Prediction errors (RMS) obtained by NNs, taken from Wieland and Wotawa (1999), Gómez *et al.* (2003). Ozone application (initial partition).

NN model	*#HU*	*LR*	*Bias*	*Steps*	RMS_{test}
MLP	2	1	N	3000	15.053
MLP	5 + 1	1	Y	1000	11.200
MLP	5	0.4	N	100	11.176
MLP	5 + 1	0.2	Y	100	10.813
MLP	2 + 1	0.2	Y	1000	10.665
MEN ($\alpha = 0.2$)	5 + 1	0.2	Y	3000	10.515
EN	8 + 1	0.2	Y	5000	10.318
MEN ($\alpha = 0.4$)	5 + 1	0.2	Y	5000	9.957
MEN-HYP ($\alpha = 0.2$)	5 + 1	0.2	Y	1000	9.193
LSTM	–	0.00125	–	3500	9.796

Table 6. Prediction errors (RMS) obtained by FIR, taken from Gómez *et al.* (2003). Ozone application (initial partition). Granularity (3,2,2,2). EFP method

Mask type	*Mask relations*	*Q*	RMS_{test}
Opt	(7,12,14,15,16)	0.4306	13.676
Subopt	(8,12,13,16)	0.3680	10.550
Subopt	(1,4,13,16)	0.3158	10.538
Subopt	(4,13,14,16)	0.4033	10.183
Subopt	(7,12,13,16)	0.3456	10.111
Subopt	(1,12,13,16)	0.3566	10.022
Subopt	(9,12,13,16)	0.2830	9.827
Subopt	(8,9,13,16)	0.2866	9.757

number of learning steps needed by the NN in order to obtain the best results. The last column indicates the RMS error of the test set. In Gómez *et al.* (2003), the results of the FIR methodology when used to obtain the best models for the problem at hand are presented. In this study, default and heuristic recode (or fuzzification) parameter values were used. The best result were obtained with the EFP method and discretizing the temperature into 3 classes, whereas the rest of the variables were discretized into 2 classes. Table 6 shows the optimal and the best suboptimal masks. The first column indicates whether it is the optimal mask or a suboptimal one. The second column describes the causal relations of each mask in a position notation. The third column prints the quality measure of each mask. The last column shows the RMS error obtained when the test set is predicted. As can be seen from tables 5 and 6, the MEN network with hyperbolic tangent activation function has the lowest prediction error. A more detailed discussion of these results can be found in (Gómez *et al.* 2003).

4.2.2 Fuzzy partitions determination. In this section the utility of the GFS developed is evaluated for the problem at hand. Thirty executions were performed for each objective function and stop criteria. Table 7 shows the values of the genetic parameters applied to this problem.

Tables 8 and 9 show a summary of the results obtained for each objective function, i.e. $1 - Q$ and $\text{MSE}_{\text{train}}$, respectively.

In this application the last 8% of the training signal is used for $\text{MSE}_{\text{train}}$ objective function evaluation and the first 92% of the signal is used to obtain the FIR models (masks and pattern rule bases). The reduced amount of data available does not allow increasing the percentage of the signal used to compute the $\text{MSE}_{\text{train}}$ objective function. Both tables are organized in the same way as tables 3 and 4. The granularity and data proportion per variable are the output of the GFS and the input parameters of the FIR recode process.

Table 7. Genetic parameters of the GA for the ozone application.

Parameter	*Value*
Population size (# individuals)	40
Crossover probability	0.6
Mutation probability	0.1
Stop criteria (chromosomes evaluations)	{500, 1000, 2000, 4000, 8000, 16,000}

Table 8. Results when the FIR methodology enhanced with a GA is used to learn the granularity and the membership functions. $1 - Q$ cost function. Ozone application.

	# eval	*Granularity*	*Data proportion*	*Optimal mask*	*Q*	*1 − Q*	RMS_{test}
A	16000	(2,3,7,2)	TMP:(0.65,0.35) NUB:(0.29,0.40,0.31) VV: (0.14,0.14,0.17,0.16,0.06,0.16,0.17) O3:(0.95,0.05)	(3,16)	0.9823	0.0177	20.9833
	8000	(3,4,7,2)	TMP:(0.28,0.42,0.30) NUB:(0.26,0.27,0.22,0.25) VV: (0.14,0.15,0.15,0.16,0.06,0.16,0.18); O3:(0.95,0.05)	(3,16)	0.9823	0.0177	20.1400
	4000	(2,3,3,2)	TMP:(0.68,0.32) NUB:(0.26,0.16,0.58) VV: (0.33,0.33,0.34); O3:(0.95,0.05)	(13,14,16)	0.9814	0.0186	11.9828
	2000	(2,3,2,2)	TMP:(0.70,0.30) NUB:(0.36,0.28,0.36) VV: (0.56,0.44); O3:(0.94,0.06)	(6,13,16)	0.9808	0.0192	11.8037
	1000	(9,7,8,2)	TMP:(0.07,0.08,0.09,0.05,0.13,0.19,0.05,0.17,0.17) NUB:(0.14,0.14,0.14,0.14,0.14,0.15,0.15) VV: (0.27,0.12,0.06,0.05,0.12,0.15,0.06,0.17) O3:(0.95,0.05)	(5,16)	0.9771	0.0229	16.7516
	500	(2,7,2,2)	TMP:(0.66,0.34) NUB:(0.14,0.14,0.14,0.14,0.14,0.15,0.15) VV: (0.20,0.80); O3:(0.93,0.07)	(14,16)	0.9548	0.0452	13.0337
B	16000	(2,3,5,3)	TMP:(0.64,0.36) NUB:(0.20,0.23,0.57) VV: (0.16,0.16,0.16,0.38,0.14); O3:(0.90,0.05,0.05)	(10,13,16)	0.9580	0.0420	10.8162
	8000	(4,3,3,4)	TMP:(0.36,0.19,0.16,0.29) NUB:(0.25,0.29,0.46) VV: (0.29,0.46,0.25); O3:(0.83,0.07,0.05,0.05)	(2,13,16)	0.9078	0.0922	10.8603
	4000	(2,4,9,3)	TMP:(0.64,0.36) NUB:(0.19,0.17,0.17,0.47) VV: (0.09,0.05,0.22,0.12,0.20,0.10,0.11,0.05,0.06) O3:(0.90,0.05,0.05)	(10,13,16)	0.9567	0.0433	10.8910
	2000	(3,3,2,2)	TMP:(0.30,0.41,0.29) NUB:(0.19,0.20,0.61) VV: (0.38,0.62); O3:(0.91,0.09)	(11,13,16)	0.9452	0.0548	10.5619
	1000	(3,4,2,2)	TMP:(0.38,0.27,0.35) NUB:(0.25,0.25,0.25,0.25)	(1,3,13,16)	0.7446	0.2554	9.8303

500	(3,7,3,2)	VV: (0.50,0.50); O3:(0.76,0.24) TMP:(0.33,0.33,0.34) NUB:(0.15,0.15,0.15,0.15,0.15,0.10,0.15) VV: (0.33,0.33,0.34); O3:(0.80,0.20)	(13,15,16)	0.7593	0.2407	8.8196

Table 9. Results when the FIR methodology enhanced with a GA is used to learn the granularity and the membership functions. Prediction error of the last 8% of the training data set (MSE_{train}) cost function. Ozone application.

	# eval	*Granularity*	*Data proportion*	*Optimal mask*	*Q*	MSE_{train}	RMS_{test}
A	16000	(6,4,6,6)	TMP:(0.10,0.15,0.18,0.22,0.27,0.08) NUB:(0.22,0.26,0.27,0.25) VV: (0.09,0.11,0.21,0.14,0.11,0.34) O3:(0.05,0.41,0.07,0.07,0.07,0.33)	(13,14,16)	0.4298	41.44	22.9025
	8000	(2,3,3,2)	TMP:(0.36,0.64); NUB:(0.34,0.30,0.36) VV: (0.24,0.32,0.44); O3:(0.06,0.48,0.46)	(11,13,14,16)	0.5170	37.98	11.7773
	4000	(4,4,4,4)	TMP:(0.10,0.22,0.19,0.49) NUB:(0.19,0.40,0.11,0.30) VV: (0.09,0.15,0.55,0.21) O3:(0.06,0.41,0.26,0.27)	(13,14,16)	0.3721	46.74	10.0564
	2000	(8,7,5,9)	TMP:(0.11,0.11,0.11,0.11,0.12,0.13,0.13,0.18) NUB:(0.11,0.29,0.21,0.05,0.19,0.08,0.07) VV: (0.08,0.22,0.07,0.06,0.57) O3:(0.09,0.05,0.14,0.17,0.06,0.08,0.08,0.06,0.27)	(13,14,16)	0.2863	48.31	24.5347
	1000	(9,6,7,9)	TMP:(0.11,0.11,0.11,0.11,0.11,0.11,0.11,0.11,0.12) NUB:(0.23,0.34,0.06,0.16,0.10,0.11) VV: (0.37,0.24,0.12,0.05,0.05,0.08,0.09) O3:(0.07,0.09,0.07,0.19,0.18,0.08,0.13,0.12,0.07)	(13,14,16)	0.3283	50.03	32.8391
	500	(6,6,5,8)	TMP:(0.12,0.10,0.27,0.15,0.12,0.24) NUB:(0.05,0.37,0.19,0.22,0.05,0.12) VV: (0.23,0.11,0.26,0.09,0.31) O3:(0.12,0.12,0.12,0.12,0.13,0.13,0.13,0.13)	(13,14,16)	0.2972	52.60	11.1948
B	16000	(3,3,4,3)	TMP:(0.13,0.23,0.64); NUB:(0.47,0.31,0.22) VV: (0.32,0.13,0.27,0.28); O3:(0.05,0.66,0.29)	(13,14,16)	0.5847	46.68	8.8591
	8000	(3,2,6,4)	TMP:(0.36,0.13,0.51) NUB:(0.55,0.45) VV: (0.14,0.32,0.16,0.11,0.15,0.12) O3:(0.05,0.73,0.17,0.05)	(13,14,16)	0.7417	54.16	9.4123
	4000	(3,3,4,3)	TMP:(0.12,0.26,0.62) NUB:(0.44,0.28,0.28) VV: (0.18,0.18,0.17,0.47); O3:(0.05,0.56,0.39)	(13,14,16)	0.4549	48.27	9.3005
	2000	(2,3,3,7)	TMP:(0.41,0.59); NUB:(0.36,0.58,0.06) VV: (0.56,0.34,0.10) O3:(0.08,0.30,0.21,0.07,0.11,0.17,0.06)	(1,13,14,16)	0.3226	51.68	9.7052

1000	(2,3,9,13)	TMP:(0.47,0.53); NUB:(0.44,0.28,0.28) VV: (0.11,0.11,0.11,0.11,0.11,0.11,0.11,0.11,0.12) O3:(0.50,0.50)	(12,13,14,16)	0.3688	57.10	9.7278
500	(3,2,4,4)	TMP:(0.53,0.25,0.22); NUB:(0.43,0.57) VV: (0.24,0.44,0.26,0.06) O3:(0.05,0.38,0.05,0.52)	(13,14,16)	0.4333	62.84	9.4563

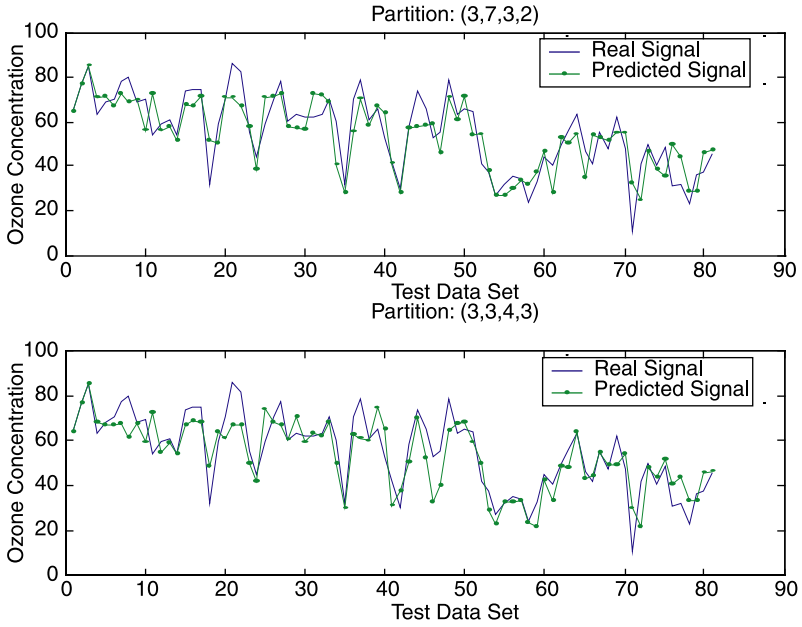

Figure 4. Prediction signals of the test data set obtained by the optimal FIR models identified with the best $1 - Q$ (top) and $\text{MSE}_{\text{train}}$ (bottom) objective functions solutions for the ozone application.

Analyzing table 8 it can be seen that, although the GFS does not assure the optimal solution, all solutions have a mask quality larger than the FIR masks of table 6. The prediction errors of the test sets (last column) are also presented to show the accuracy of each model obtained. Notice that although the RMS_{test} is usually smaller when the GFS is used, this is not always true. This is due to the fact that the test data sets were not used in the FIR model identification process. Only the suboptimal solutions (row *B* for each objective function) present lower prediction errors than those shown in tables 5 and 6. The best result

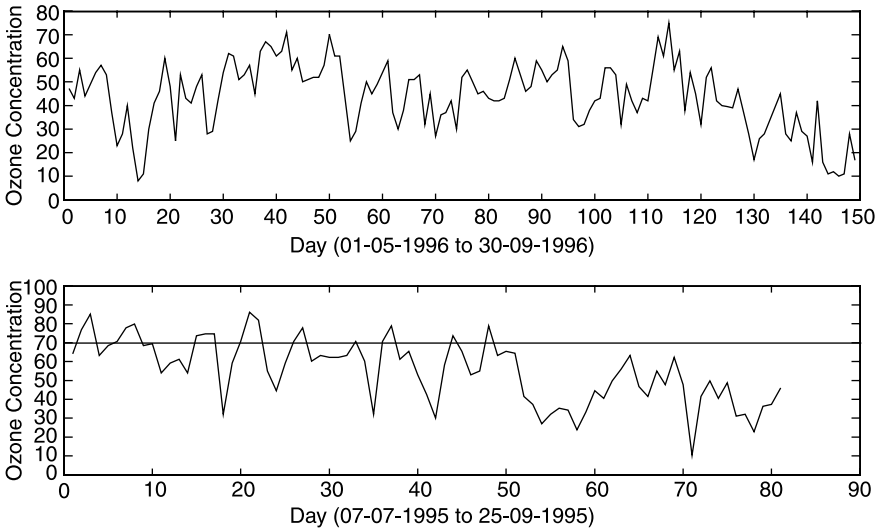

Figure 5. Training and test data sets. Ozone application.

of 9.193 RMS obtained by the MEN network with hyperbolic tangent activation function is bigger than the 8.819 and 8.859 RMS errors obtained by FIR methodology enhanced with a GA. Also, as expected, the MSE_{train} objective function is more expensive computationally. For example, the computational time needed to perform 30 executions for 16000 evaluations when the $1 - Q$ and MSE_{train} objective functions are studied is 41:19 and 58:05 h, respectively, on a Pentium IV computer (2.66 Hz). It is important to remark that the determination of the recode or fuzzification parameters is an offline process. Therefore, once the FIR model is available, the prediction becomes a real-time process.

As in the previous application, the performance of FIR models when the MSE_{train} objective function is used is not superior to the performance of $1 - Q$ objective function. Therefore, once again, the $1 - Q$ objective function is preferable. The user should decide, which objective function to use taking into account the characteristics of the problem to be solved, the size of the optimization problem and the required accuracy.

Figure 4 presents the best predictions obtained for the two cost functions studied. The top plot shows real and predicted test signals obtained when the best solution of the $1 - Q$ objective function is used to find the optimal mask, whereas the bottom plot shows these signals obtained when the best solution of the MSE_{train} objective function is used to find the optimal mask. The RMS errors for the top and bottom plots are 8.819 and 8.859, respectively. They, basically, fail on the upper peaks of the first 45 data points. The reason of the poor ability of FIR models to correctly forecast the upper peaks of the first part of the test signal is quite clear. FIR is an inductive methodology that captures the pattern rule base (behavior matrix) from the observed data (identification set), and bases the prediction on the knowledge derived from it. Therefore, it is not able to predict system behaviors that do not appear in the identification (training) set, i.e. data that the FIR model has never seen before. This is exactly the case here. Figure 5 shows the identification (upper plot) and the test (lower plot) signals of the ozone concentration. Analyzing carefully the identification signal, it can be seen that from April to September of 1996 (full data set) the ozone concentration has never reached values higher than 70 ppb. However, from the beginning of July to middle of August of the test data set (first portion of the lower plot of figure 5) all the ozone upper peaks go beyond this level. Therefore, FIR is not able to predict this behavior because it is not included in the training set. Notice that FIR "memorizes" de training data (in the behavior matrix) and, therefore, is not able to predict system's behavior that has never been seen before.

5. Conclusions

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

Two real problems are presented in this research, i.e. the estimation of the maintenance cost of medium voltage lines in Spanish towns and the prediction of ozone levels in Austria. The performance of FIR models encountered using the new approach were superior when

compared with the performance of other methodologies presented in previous studies such are linear models, second order polynomial models, NNs, hybrid genetic programming and different fuzzy models, for the same problems.

References

Acosta, J., Aprendizaje de Particiones Difusas para Razonamiento Inductivo, PhD thesis, Universitat Politècnica de Catalunya 2006, p. 296.

Alpaydin, G., Dündar, G. and Balktr, S., 2002, Evolution-based design of neural fuzzy networks using self-adapting genetic parameters. *IEEE Transactions on Fuzzy Systems*, **10**(2), pp. 211–221.

Baker, J.E., 1985. Adaptive selection methods for genetic algorithms. In *Proceedings of First International Conference on Genetic Algorithms*, J.J. Grefenstette (Ed.), pp. 101–111, Hillsdale.

Baker, J.E., 1987. Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the Second International Conference on Genetic Algorithms (ICGA'87)*, J.J. Grefenstette (Ed.), pp. 14–21, Hillsdale.

Bonissone, P.P., 1997, Soft computing: the convergence of emerging reasoning technologies. *Soft Computing*, **1**(1), pp. 6–18.

Camargo, H.A., Pires, M.G. and Castro, P.A.D., 2004. Genetic design of fuzzy knowledge bases—a study of different approaches. *Proceedings of the IEEE Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS'04)*, pp. 954–959.

Carmona, P., Castro, J.L. and Zurita, J.M., 2004, Strategies to identify fuzzy rules directly from certainty degrees: a comparison and a proposal. *IEEE Transactions on Fuzzy System*, **12**(5), pp. 631–640.

Casillas, J., Cordón, O., del Jesús, M.J. and Herrera, F., 2005, Genetic tuning of fuzzy rule deep structures preserving interpretability and its interactison with fuzzy set reduction. *IEEE Transactions on Fuzzy Systems*, **13**(1), pp. 29.

Cellier, F.E., Nebot, A., Mugica, F. and De Albornoz, A., 1996, Combined qualitative/quantitative simulation models of continuous-time processes using fuzzy inductive reasoning techniques. *International Journal of General Systems*, **24**(1–2), pp. 95–116.

Chen, S.M., Horng, Y.J. and Lee, C.H., 2001, Document retrieval using fuzzy-valued concept networks. *IEEE Transactions on Systems, Man, and Cybernetics Part B*, **31**(1), pp. 111–118.

Chi, Z., Yan, H. and Pham, T., 1996, *Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition* (Singapore: World Scientific).

Chung, I.-F., Lin, C.J. and Lin, C.T., 2000, A GA-based fuzzy adaptive learning control network. *Fuzzy Sets and Systems*, **112**(1), pp. 65–84.

Cordón, O., Herrera, F. and Sánchez, L., 1998. Computing the spanish medium electrical line maintenance costs by means of evolution-based learning processes. *Proceedings of the 11th International Conference on Industrial and Engineering Applications of AI and ES (IEA-98-AIE)* (Spain: Castellón), pp. 478–482.

Cordón, O., Herrera, F. and Sánchez, L., 1999, Solving electrical distribution problems using hybrid evolutionary data analysis techniques. *Applied Intelligence*, **10**, pp. 5–24.

Cordón, O., Herrera, F., Hoffmann, F. and Magdalena, L., 2001. Genetic fuzzy systems. *Evolutionary Tuning and Learning of Fuzzy Knowledge Bases* (Singapore: World Scientific).

Cordón, O., Gomide, F., Herrera, F., Hoffmann, F. and Magdalena, L., 2004, Ten years of genetic fuzzy systems: current framework and new trends. *Fuzzy Sets and Systems*, **141**(1), pp. 5–31.

Dote, Y. and Ovaska, S.J., 2001, Industrial applications of soft computing: a review. *Proceedings of the IEEE*, **89**(9), pp. 1243–1265.

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

González, A. and Pérez, R., 2001, An experimental study about the research mechanism in SLAVE learning algorithm: hill-climbing methods versus genetic algorithm. *Information Sciences*, **136**(1–4), pp. 159–174.

Grefenstette, J.J., 1986, Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, **16**(1), pp. 122–128.

Gudwin, R., Gomide, F. and Pedrycz, W., 1998, Context adaptation in fuzzy processing and genetic algorithms. *International Journal of Intelligent Systems*, **13**(10–11), pp. 929–948.

Gómez, P., Nebot, A., Ribeiro, S., Alquézar, R., Mugica, F. and Wotawa, F., 2003, Local maximum ozone concentration prediction using soft computing methodologies. *Systems Analysis Modelling Simulation*, **43**(8), pp. 1011–1031.

Gürocak, H.B., 1999, A genetic-algorithm-based method for tuning fuzzy logic controllers. *Fuzzy Sets and Systems*, **108**(1), pp. 39–47.

Hall, L.O., Bezdek, J.C., Boggavarpu, S. and Bensaid, A., 1994. Genetic fuzzy clustering. *Proceedings of NAFIPS'94*, pp. 411–415, San Antonio.

Heider, H. and Drabe, T., 1997, A cascaded genetic algorithm for improving fuzzy-system design. *International Journal of Approximate Reasoning*, **17**(4), pp. 351–368.

Herrera, F., Lozano, M. and Verdegay, J.L., 1995, Tuning fuzzy controllers by genetic algorithms. *International Journal of Approximate Reasoning*, **12**(3–4), pp. 299–315.

HIROTA, K. and SUGENO, M., 1995, *Industrial Applications of Fuzzy Technology in the World* (Singapore: World Scientific).

HOFFMANN, F., 2001, Evolutionary algorithms for fuzzy control system design. *Proceedings of IEEE*, **89**(9), pp. 1318–1333.

HOFFMANN, F. and PFISTER, G., 1997, Evolutionary design of a fuzzy knowledge base for a mobile robot. *International Journal of Approximate Reasoning*, **17**(4), pp. 447–469.

HOLLAND, J., 1975, *Adaptation in Natural and Artificial Systems* (Ann Arbor: The University of Michigan Press).

HOWEARD, L. and D'ANGELO, D., 1995, The GA-P: A genetic algorithm and genetic programming hybrid. *IEEE Expert*, pp. 11–15.

ISHIBUCHI, H., NAKASHIMA, T. and MURATA, T., 1999, Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems. *IEEE Transactions on Systems, Man, and Cybernetics Part B*, **29**(5), pp. 601–618.

JANG, J.S.R., SUN, C.T. and MIZUTANI, E., 1997, *Neuro-fuzzy and Soft Computing* (New York: Prentice-Hall).

KLIR, G. and ELIAS, D., 2002, *Architecture of Systems Problem Solving*, 2nd ed. (NY: Plenum Press).

KUNCHEVA, L.I., 2000, *Fuzzy Classifier Design* (Heidelberg: Physica-Verlag).

LEONDES, C.T., 1999, *Fuzzy Theory Systems: Techniques and Applications* (San Diego, CA: Academic Press) **Vol. 1–4**.

MAGDALENA, L., 1999, Adapting the gain of an FLC with genetic algorithms. *International Journal of Approximate Reasoning*, **17**(4), pp. 327–349.

MICHALEWICZ, Z., 1996, *Genetic Algorithms + Data Structures = Evolution Programs* (New York: Springer-Verlag).

MIYAMOTO, S., 1989, Two approaches for information retrieval through fuzzy associations. *IEEE Transactions on Systems, Man, and Cybernetics*, **19**(1), pp. 123–130.

MUGICA, F. and CELLIER, F.E., 1994. Automated synthesis of a fuzzy controller for cargo ship steering by means of qualitative simulation. *Proceedings of ESM'94, European Simulation MultiConference*, (Barcelona, June 1–3), pp. 523–528.

NEBOT, A., 1994, Qualitative modeling and simulation of biomedical systems using fuzzy inductive reasoning, PhD thesis, Dept. Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Barcelona, Spain.

NEBOT, A., CELLIER, F.E. and LINKENS, D., 1996, Synthesis of an anaesthetic agent administration system using fuzzy inductive reasoning. *Artificial Intelligence in Medicine*, **8**(3), pp. 147–166.

NEBOT, A., MUGICA, F. and GÓMEZ, P., 2001. Long term prediction of maximum ozone concentration using fuzzy inductive reasoning. *Proceedings of EUNITE'01: European Symposium on Intelligent Technologies, Hybrid Systems and their implementation on Smart Adaptive Systems* (Tenerife, December 13–15), pp. 91–101.

PEDRYCZ, W., 1996, *Fuzzy Modelling: Paradigms and Practice* (London: Kluwer Academic Press).

POMARES, H., ROJAS, I., GONZÁLEZ, J., DAMAS, M., PINO, B. and PRIETO, A., 2004, Online global learning in direct fuzzy controllers. *IEEE Transactions on Fuzzy Systems*, **12**(2), pp. 218–229.

RUSSO, M., 1998, FuGeNeSys: a fuzzy genetic neural system for fuzzy modeling. *IEEE Transactions on Fuzzy Systems*, **6**(3), pp. 373–388.

SATTAR, F. and TAY, D.B.H., 1999, Enhancement of document images using multiresolution and fuzzy logic techniques. *IEEE Signal Processing Letters*, **6**(10), pp. 249–252.

STOHL, A., WOTAWA, G. and KROMP-KOLB, H., 1996, The IMPO modeling system description, sensitivity studies and applications, Technical Report, Universität für Bodenkultur, Institut für Meteorologie und Physik, Vienna, Austria.

SUZUKI, Y., ITAKURA, K., SAGA, S. and MAEDA, J., 2001, Signal processing and pattern recognition with soft computing. *Proceedings of the IEEE, Special Issue on Industrial Innovations Using Soft Computing*, **89**(9), pp. 1297–1317.

THRIFT, P., 1991. Fuzzy logic synthesis with genetic algorithms. *Fourth International Conference on Genetic Algorithms (ICGA'91)*, San Diego, CA pp. 509–513.

VAN LE, T., 1995. Evolutionary fuzzy clustering. *Proceedings of 2nd IEEE Conference on Evolutionary Computation (ICEC'95)* **2**, pp. 753–758, Perth.

VAPNIK, V., 1998, *Statistical Learning Theory* (New York: Wiley & Sons).

WIELAND, D. and WOTAWA, W., 1999. Local maximum ozone concentration prediction using neural networks. *AAAI Workshop on Environmental Decision Support Systems and Artificial Intelligence*, pp. 47–54.

YUAN, B., KLIR, G.J. and SWAN-STONE, J.F., 1995. Evolutionary fuzzy c-means clustering algorithm. *Proceedings of 4th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'95)*, pp. 2221–2226, Yokohama.

DEL JESÚS, M.J., HOFFMAN, F., NAVASCUÉS, I.J. and SÁNCHEZ, L., 2004, Induction of fuzzy-rule based classifiers with evolutionary boosting algorithms. *IEEE Transactions on Fuzzy Systems*, **12**(3), pp. 296–308.

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 MS 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 PhD degree from Polytechnical 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 optimisation of fuzzy partitions for inductive reasoning at the Soft Computing Group, UPC. His research interests are in areas including control, fuzzy logic, GAs, pattern recognition and machine learning.Mr Acosta received a PhD fellowship in 2001–2003 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.

Àngela Nebot received her BS and PhD 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, GAs, 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 like "IEEE Transactions on Fuzzy Systems", "International Journal of General Systems", "Neurocomputing", "Artificial Intelligence in Environmental Engineering" and "Artificial Intelligence Communication".

Pedro Villar was born in Granada, Spain, on August 15, 1968. He received his MS degree in Computer Science in 1991 from the University of Granada, Spain and his PhD 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 GFSs, evolutionary algorithms and multi-objective GAs.

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 Polytechnical 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 m 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.