Genetic Fuzzy System for Data-Driven Soft Sensors Design

Jérôme Mendes, Francisco Souza, Rui Araújo, and Nuno Gonçalves
Institute for Systems and Robotics (ISR-UC), and
Department of Electrical and Computer Engineering (DEEC-UC),
University of Coimbra, Pólo II, PT-3030-290 Coimbra
jermendes@isr.uc.pt, fasouza@isr.uc.pt,rui@isr.uc.pt and nunogon@isr.uc.pt

Abstract

This paper proposes a new method for Soft Sensors (SS) design for industrial applications based on a Takagi-Sugeno (T-S) fuzzy model. The learning of the T-S model is performed from input/output data to approximate unknown nonlinear processes by a coevolationary genetic algorithm (GA). The proposed method is an automatic tool for SS design since it does not require any prior knowledge concerning the structure (e.g. the number of rules) and the database (e.g. antecedent fuzzy sets) of the T-S fuzzy model, and concerning the selection of the adequate input variables and their respective time delays for the prediction setting. The GA approach is composed by five hierarchical levels and has the global goal of maximizing the prediction accuracy. The first level consists in the selection of the set of input variables and respective delays for the T-S fuzzy model. The second level considers the encoding of the membership functions. The individual rules are defined at the third level, the population of the set of rules is treated in fourth level, and a population of fuzzy systems is handled at the fifth level. To validate and demonstrate the performance and effectiveness of the proposed algorithm, it is applied on two prediction problems. The first is the Box-Jenkins benchmark problem, and the second is the estimation of the flour concentration in the effluent of a real-world wastewater treatment system. Simulation results are presented showing that the developed evolving T-S fuzzy model can identify the nonlinear systems satisfactorily with appropriate input variables and delay selection and a reasonable number of rules. The proposed methodology is able to design all the parts of the T-S fuzzy prediction model. Moreover, presented comparison results indicate that the proposed method outperforms other previously proposed methods for the design of prediction models, including methods previously proposed for the design of T-S models.

1 Introduction

Data-driven soft sensors (DDSS) are inferential models that use on-line available sensor measures for on-line estimation of variables which cannot be automatically measured at all, or can only be measured at high cost, sporadically, or with large time delays (e.g. laboratory analysis). These models are based on measurements which are recorded and provided as historical data. The models themselves are empirical predictive models. They are valuable tools to many industrial applications such as refineries, pulp and paper mills, wastewater treatment systems, just to give a few examples [1].

The development of DDSS can be divided into four main stages: (I) Data collection, and selection of historical data; (II) Data pre-processing; (III) Model selection, training and validation; (IV) Soft sensor maintenance. In the first stage, the data for training and evaluation of the model is selected. The usual steps in pre-processing are the handling of missing data, outliers detection, selection of relevant variables (i.e. feature selection), detecting the delays between the particular variables, and handling of drifting data [2]. One problem in the preprocessing step is that it requires a large amount of manual work and expert knowledge about the underlying process. Next, the model selection, training and validation phase is one of most important in soft sensors development, requiring the correct selection of the model, so that it can correctly reproduce the target variable. The last step is SS maintenance, where the goal is to maintain good SS response even in the presence of process variations, or some data change.

DDSS are built based on empirical observations of the process. Fuzzy models are useful to model systems when they cannot be defined in precise mathematical terms from physical laws. Fuzzy modeling of systems has been worked in many scientific researches. Takagi and Sugeno (T-S) have proposed a search algorithm for a fuzzy controller and generalized their research to fuzzy identification [3]. T-S fuzzy models are suitable to model a large class of nonlinear systems and have gained much popularity because of their rule consequent structure which is a mathematical function. To design the T-S fuzzy model, the global operation of the nonlinear system can be accurately approximated into several local affine models. T-S fuzzy models are a powerful tool to implement the prediction structure of soft sensors.

GAs have proved to be useful in solving a variety of search and optimization problems. This motivates that GAs might be a useful soft computing technique for designing SS based on T-S fuzzy models. Genetic algorithms also have been used for learning classifier systems [4], and nonlinear system identification [5, 6, 7, 8, 9, 10]. In [4], it is proposed a GAs approach to generate optimal fuzzy rules in a classification setting that continuously monitors system states for automatically detecting faults on a HVAC system. In [8], a genetic algorithms based multi-objective optimization technique was utilized in the training process of a feedforward neural network, using noisy data from an industrial iron blast furnace. In [11] an approach is proposed for a dynamic creation and evolving of a first order Takagi-Sugeno fuzzy system model. In [12] it is investigated a technique for modeling and identification of a new dynamic NARX fuzzy model by means of genetic algorithms. The paper proposed the use of a modified genetic algorithm combined with the predictive capability of NARX Takagi-Sugeno model for generating the dynamic NARX T-S fuzzy model. In [13] a fuzzy wavelet neural network model that uses a GA approach for adjusting parameters, is introduced for function approximation and nonlinear dynamic system identification from input-output pairs. [14] proposed a methodology for automatically extracting T-S fuzzy models from data using particle swarm optimization. The structures and parameters of the fuzzy models are encoded into particles and evolve together so that the optimal structure and parameters can be achieved simultaneously. In [15], a technique for the modeling of nonlinear control processes using fuzzy modeling approach based on the Takagi-Sugeno fuzzy model with a combination of a genetic algorithm and the recursive least squares method is proposed.

The above mentioned steps (II) and (III) are essential for correct DDSS development, being directly related to the selection of input variables and the respective time lags and to model identification. The selection of the most adequate input variables *and* the respective time delays is crucial since the use of the correct variables with the correct delays can lead to better prediction accuracy because they can contain more information about the output than incorrect variables and/or variables with incorrect delays. Some studies have used techniques based on variance, such as as principal component analysis (PCA) for variable selection [16]. These methods are designed for linear models, so they can not be the best choice for nonlinear modeling. In the works [5], [6], input variables and respective time delays are automatically selected jointly with the learning of the prediction model. The methods of [4, 7, 8, 9, 10], have the limitation of not being able to perform automatic selection of variables and delays: pre-selection is performed. Pre-selection may be performed completely from human knowledge (e.g. knowledge of the real model, such as in [7, 10, 12, 13, 14, 15, 17]) or using some auxiliary criteria that does not take advantage of taking into account the prediction model being learned, such as correlation coefficients, Kohonen maps and Lipschitz quotients [9], regularity criterion [11], [18] or analysis of "fuzzy curves" [19]. Some approaches have the limitation of not performing the selection of the time delays of input variables (e.g. [4] and [8]). While [5, 7, 8, 9, 10, 18, 19, 17] learn fuzzy models,

[6] learns NARX models. However, fuzzy models have a linguistic interpretation which is an important desirable characteristic for human users/ operators of the SS, such as in industrial applications.

An approach using methods for both nonlinear variable selection and learning T-S fuzzy models was proposed by [20] and later by [9]. In [20], it is introduced a hierarchical evolutionary approach to optimize the parameters of Takagi–Sugeno fuzzy systems, where the selection of the variables is performed completely from human knowledge, such as knowledge about the real model. The problems addressed are function approximation and pattern classification. As an evolution or improvement of [20], in the work [9], it was proposed to add to [20] a mechanism for pre-selection of the variables by an auxiliary criteria. The proposed method is addressed for soft sensors applications. It uses T-S fuzzy models learned from available input/output data by means of a coevolutionary GA and a neuro-based technique. The soft sensor design is carried out in two steps. First, the input variables of the fuzzy model are pre-selected from the variables of the dynamical process by means of correlation coefficients, Kohonen maps and Lipschitz quotients. Such selection procedure considers nonlinear relations among the input and output variables. Second, a hierarchical GA is used to identify the fuzzy model itself. The input variable selection approach proposed by [9] has some drawbacks. First, the selection of the number of neurons in the Kohonen maps is not automatically performed. Second, variables and delays selection is not jointly performed with the learning of the fuzzy model (pre-selection is performed), which precludes the global optimization of the prediction setting. Finally, the selection of input variables is not accompanied with the selection of the respective time delays. The later shortcoming can bring low-accuracy results because a variable with the correct delay can contain more information about the output than a variable with an incorrect delay.

This paper proposes a novel methodology for soft sensors development. The proposed method is an automatic tool for SS design, with the following main characteristics: (1) it automatically performs the optimization of the variable and delay selection jointly with the learning and optimization of the system model without the need for any prior human knowledge, (2) the T-S fuzzy model structure is constructed just according to the data characteristics, and (3) it is optimized by means of GAs. This work has been inspired by [9]. However, it will jointly optimize a larger number of components of the prediction setting when compared with in [9]. A hierarchical genetic algorithm (HGA) will be used to optimize a large set of parameters encoded at five different levels to design the T-S fuzzy model. When more complex design decisions involving a large number of parameters must be made, a global formulation of the problem representing all the parameters in just one optimization level can be inadequate. It is well known that computation, search, and optimization problems become more difficult to solve when the dimensionality of the state-space increases. In many cases, this problem is known as the curse of dimensionality. To tackle this issue, in this paper the global problem is divided into various optimization levels, where the genetic evolution (optimization) of each level is performed separately, but is influenced by the current populations and optimization states of all the levels. HGAs make it possible to have different layers optimizing different parts of the T-S model, and facilitate the human interpretation of the optimization structure. The main advancement of is this work in comparison with [9] is the addition of a new hierarchical level responsible for the selection of variables and delays. The hierarchical genetic fuzzy system is constituted by five levels. In the first level, the input variables and respective delays are chosen with the goal of attaining the highest possible prediction accuracy of the T-S fuzzy model. The selection of variables and delays is performed jointly with the learning of the fuzzy model, which increases the global optimization performance. The second level encodes the membership functions. The individual rules are defined at the third level. The population of the set of rules is defined in the fourth level, and a population of fuzzy systems is treated at the fifth level. The least squares method is used to determine the parameters of the rule consequents. Levels two to five were based on [9].

To validate and demonstrate the performance and effectiveness of the proposed algorithm, it is applied on two prediction problems. The first is the Box-Jenkins benchmark problem, and the second is the estimation of the flour concentration in the effluent of a real-world wastewater treatment system. Simulation results are presented showing that the developed evolving T-S fuzzy model can identify the nonlinear systems satisfactorily with appropriate input variables and delay selection and a reasonable number of rules. The results of the proposed methodology are compared with the work [9], and other works. The presented comparison results indicate that the proposed method outperforms other previously proposed methods for the design of prediction models, including T-S models.

The paper is organized as follows. Section 2 presents a method for nonlinear systems modeling using T-S fuzzy models. The hierarchical genetic fuzzy system proposed in this paper is described in Section 3. In Section 4, results of the application of the proposed methodology to nonlinear systems modeling are presented and analyzed. Finally, Section 5 makes concluding remarks.

2 Nonlinear Systems Modeling Using T-S Fuzzy Models

Takagi-Sugeno fuzzy models with simplified linear rule consequents are universal approximators capable of approximating any continuous nonlinear system [21]. For more details about T-S fuzzy models, references [3], [22], are recommended. With a T-S fuzzy model, the global operation of the nonlinear system can be accurately approximated into several local affine models. Specifically, a nonlinear system can be described by a T-S fuzzy model defined by the following fuzzy rules:

$$\begin{aligned} R_i:\ & \text{IF } x_1(k) \text{ is } A_1^i, \text{ and } \dots \text{ and } x_n(k) \text{ is } A_n^i \\ & \text{THEN Y is } y_i(\mathbf{c}_i, \mathbf{x}), \quad i = 1, 2, \dots, N, \end{aligned} \tag{1}$$

where R_i $(i = 1, 2, \dots, N)$ represents the i-th fuzzy rule, N is the number of rules, $x_1(k), \dots, x_n(k)$ are the input variables of the T-S fuzzy system - they can be any variables chosen by the designer. A_j^i are linguistic terms characterized by fuzzy membership functions $\mu_{A_j^i}(x_j)$ which describe the local operating regions of the plant. Vector $\mathbf{x}(k) = [x_1(k), x_2(k), \dots, x_n(k)]^T$ is both the system input, and an independent variable in the consequent functions $y_i(\cdot)$ $(i = 1, 2, \dots, N)$. $\mathbf{c}_i = \left[c_{i0}, \cdots, c_{i(2n+1)}\right]^T$ is a Q-dimensional vector that contains adjustable parameters of $y_i(\cdot)$ $(i = 1, 2, \dots, N)$. In this paper, $Q = 2n + 2$, and nonlinear T-S functions of the following form are considered:

$$\begin{aligned} y_i(\mathbf{c}_i, \mathbf{x}) \ = \ & c_{i0} + c_{i1}x_1(k) + \cdots + c_{in}x_n(k) + \\ & c_{i(n+1)}x_1^2(k) + \cdots + c_{i(2n)}x_n^2(k) + \\ & c_{i(2n+1)}x_1(k)\cdots x_n(k), \quad i = 1, 2, \dots, N. \end{aligned} \tag{2}$$

2.1 Least Squares Method

Using the product t-norm, let $\mu_{A^i}(\mathbf{x}_p) = \prod_{j=1}^n \mu_{A_j^i}(x_{jp})$ be the result of antecedent aggregation in ith fuzzy rule $(i = 1, 2, \dots, N)$ for the pth input $\mathbf{x}_p(k) = [x_{1p}(k), x_{2p}(k), \dots, x_{np}(k)]^T$. The final output of the fuzzy model is inferred by a center weighted average defuzzification method as follows (see more details in [22]):

$$y[\mathbf{x}_p(k)] = \sum_{i=1}^{N} \bar{\omega}^i[\mathbf{x}_p(k)] y_i(\mathbf{c}_i, \mathbf{x}_p), \tag{3}$$

$$= \sum_{i=1}^{N} \bar{\omega}^i[\mathbf{x}_p(k)] \mathbf{x}_e^T(k)\, \mathbf{c}_i, \tag{4}$$

$$= \psi[\mathbf{x}_p(k)]^T \mathbf{\Theta},$$

where, for $i = 1, \ldots, N$,

$$\begin{aligned}
\bar{\omega}^i[\mathbf{x}_p(k)] &= \frac{\prod_{j=1}^n \mu_{A_j^i}[\mathbf{x}_p(k)]}{\sum_{q=1}^N \prod_{j=1}^n \mu_{A_j^q}[\mathbf{x}_p(k)]}, && (5)\\
\mathbf{\Theta} &= \left[\mathbf{c}_1^T, \mathbf{c}_2^T, \cdots, \mathbf{c}_N^T\right]^T, && (6)\\
\mathbf{x}_e(k) &= \mathbf{x}_e\left[\mathbf{x}_p(k)\right] = [1, x_{1p}(k), \cdots, x_{np}(k), \\
&\quad x_{1p}^2(k), \cdots, x_{np}^2(k), x_{1p}(k) \cdots x_{np}(k)]^T, && (7)\\
\psi[\mathbf{x}_p(k)] &= \left[\left(\bar{\omega}^1[\mathbf{x}(k)]\right) \mathbf{x}_e^T(k), \ldots, \left(\bar{\omega}^N[\mathbf{x}(k)]\right) \mathbf{x}_e^T(k)\right]^T.
\end{aligned}$$

Let $y_d(\mathbf{x}_p(k))$ be the desired model of the system output, and consider M input patterns $\mathbf{x}_q(k)$, the corresponding desired output patterns $y_d(\mathbf{x}_q(k))$, $q = 1, \ldots, M$, and

$$\begin{aligned}
\mathbf{y}_d &= \left[y_d(\mathbf{x}_1(k)), \cdots, y_d(\mathbf{x}_M(k))\right]^T, && (8)\\
\mathbf{\Psi} &= \left[\psi(\mathbf{x}_1(k)), \cdots, \psi(\mathbf{x}_M(k))\right]^T. && (9)
\end{aligned}$$

Let $\mathbf{c}_i^*$, and $\mathbf{\Theta}^*$ be optimal values of $\mathbf{c}_i$, and $\mathbf{\Theta}$, respectively, in the least squares sense,

$$\begin{aligned}
\mathbf{c}_i^* &= \left[c_{i0}^*, \cdots, c_{i(2n+1)}^*\right]^T, && (10)\\
\mathbf{\Theta}^* &= \left[(\mathbf{c}_1^*)^T, (\mathbf{c}_2^*)^T, \cdots, (\mathbf{c}_N^*)^T\right]^T. && (11)
\end{aligned}$$

Under these conditions, a solution $\mathbf{\Theta}^*$ can be computed using the pseudo-inverse least squares method, as follows [23]:

$$\begin{aligned}
\mathbf{y}_d &= \mathbf{\Psi}\mathbf{\Theta}, && (12)\\
\mathbf{\Theta}^* &= \mathbf{\Psi}^+ \mathbf{y}_d. && (13)
\end{aligned}$$

If $\mathbf{\Psi}$ is full rank, then its pseudo inverse, $\mathbf{\Psi}^+$, can be computed in closed form as follows:

$$\begin{aligned}
\mathbf{\Psi}^+ &= \mathbf{\Psi}^T(\mathbf{\Psi}\mathbf{\Psi}^T)^{-1}, \quad \text{for } M \leq N(2n+2), && (14)\\
\mathbf{\Psi}^+ &= \mathbf{\Psi}^{-1}, \quad \text{for } M = N(2n+2), && (15)\\
\mathbf{\Psi}^+ &= (\mathbf{\Psi}^T\mathbf{\Psi})^{-1}\mathbf{\Psi}^T, \quad \text{for } M \geq N(2n+2). && (16)
\end{aligned}$$

Solution $\mathbf{\Theta}^*$ in (13) is the minimum-norm least squares solution of (12). A reasonable working assumption is that there are more data patterns than model parameters, i.e. $M \geq N(2n+2)$. Note that the full rank condition can always be made satisfied due to the fact that whenever $\mathbf{\Psi}$ is not full-rank, the linearly dependent columns of $\mathbf{\Psi}$ can be iteratively eliminated until $\mathbf{\Psi}$ has full rank [20].

3 Hierarchical Genetic Fuzzy System

The approach used in this paper to coevolve T-S fuzzy systems, which represent the SS, is based on the hierarchical evolutionary scheme introduced by [9]. The main advancement introduced here is the addition of a new hierarchical level responsible for selection of input variables and respective delays, where the variables and respective delays are chosen with the goal of attaining the highest possible prediction accuracy of the T-S fuzzy model. Levels two to five were based on [9]. The proposed coevolutionary model is illustrated in Fig. 1. The approach is constituted by five hierarchical populations, where each population represents different species. The first level consists in a population of a set of input variables and respective time delays. The second level considers the membership functions of the T-S fuzzy system. The population of individual rules is treated at the third level. The population of the set of rules is handled in the fourth level and, the population of fuzzy systems is evolved at the fifth level.

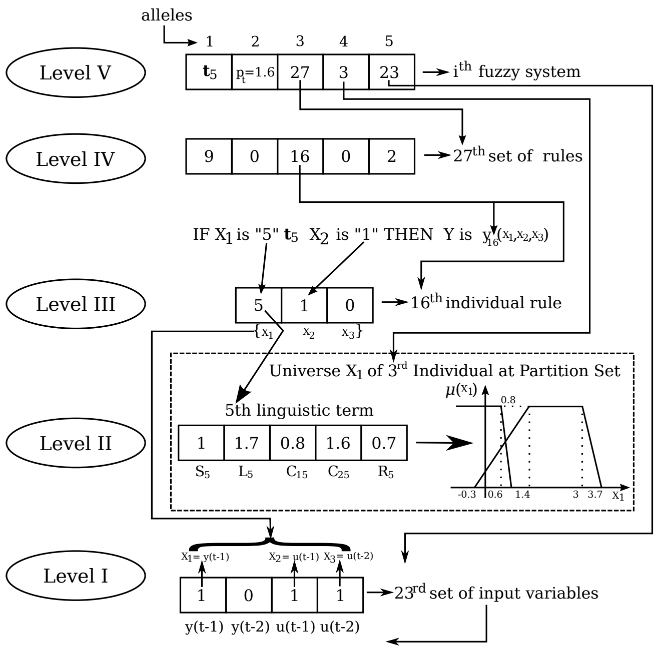

Figure 1: Encoding and hierarchical relations among the individuals of the different levels of the genetic hierarchy.

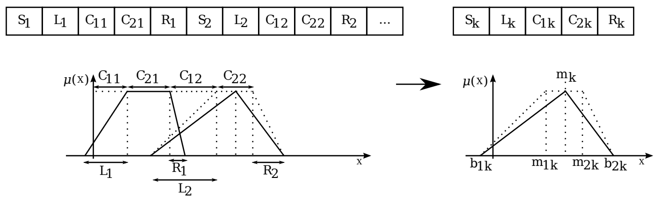

Figure 2: Membership functions encoding in Level II.

3.1 Hierarchical Structure

The population of Level I is formed by a set of input variables and respective delays that will be used in the T-S fuzzy model to design the SS. The chromosome of Level I is represented by a binary encoding, where each allele (element of the chromosome that is located at a specific position) corresponds to each input variable and respective delay (see Fig. 1). The length of the chromosome is given by the total number of pairs of system variables and respective delays that are considered as possible candidates to be used as inputs of the T-S model. In the example of Fig. 1 the selected pairs correspond to $x_1 = y(t-1)$, $x_2 = u(t-1)$, and $x_3 = u(t-2)$.

Level II contains the representation of all antecedent membership functions (including their corresponding parameters) defined in the universe of the variables involved. The chromosome is formed by the aggregations, one after another, of all partition sets associated with each input variable. A partition set of a variable is a collection of fuzzy sets associated to the variable. For each variable the range of possible values that the variable may take is fixed by the designer. Also, the designer may optionally choose to divide the set of input variables into several disjoint groups, and then enforce that for each individual, all variables within each group have the same partition set of Level II. Level II is represented by integer and real encoding. Fig. 2 shows the detail of the chromosome of Level II. Each first allele uses integer encoding to represent the type of membership function. In this paper three different types of membership function are assumed: trapezoidal ($S_k = 1$), triangular ($S_k = 2$), and Gaussian ($S_k = 3$). Zero values at the first allele ($S_k = 0$) represent an unused chromosome. Alleles 2-5 use real encoding to represent the parameters of the membership function. Considering the kth membership function, for trapezoidal functions, alleles 2-5 are converted into absolute values, given by (see Fig. 2):

$$m_{1k} = m_{2,k-1} + C_{1k}, \tag{17}$$

$$m_{2k} = m_{1k} + C_{2k}, \tag{18}$$

$$b_{1k} = m_{1k} - L_k, \tag{19}$$

$$b_{2k} = m_{2k} + R_k, \tag{20}$$

where m_{20} can be initialized by the first value of the universe of respective variable. For triangular membership functions, the center is found by the average between m_{1k} and m_{2k} (see Fig. 2). For Gaussian functions, the central value is calculated in the same way as in the triangular case, and the dispersion is given by $\sigma_{kj} = \Delta_k/3$ where $\Delta_k = (L_k + R_k)/2$. In the example presented in Fig. 1, Level II of the GA hierarchy is illustrated by the 5th membership function of x_1, which in this case is a trapezoidal membership function ($S_5 = 1$). Using (17)-(20), its parameters are:

$$m_{15} = m_{2,4} + C_{15} = 0.6 + 0.8 = 1.4, \tag{21}$$

$$m_{25} = m_{15} + C_{25} = 1.4 + 1.6 = 3, \tag{22}$$

$$b_{15} = m_{15} - L_5 = 1.4 - 1.7 = -0.3, \tag{23}$$

$$b_{25} = m_{25} + R_5 = 3 + 0.7 = 3.7, \tag{24}$$

as can be deduced from Figs. 1, and 2.

Level III is formed by a population of individual rules. The length of the chromosome is determined by the maximum number of antecedent variables. The chromosome is represented by integer encoding where each allele is formed by the index that identifies the corresponding antecedent membership function (defined at Level II). Null index values indicate the absence of membership function for the corresponding variable (i.e. the absence of the variable) in the rule. In the example of Fig. 1, Level III of the GA hierarchy is illustrated by describing the 16th individual rule. As can be seen, in this rule x_1 is represented by its 5th membership function, x_2 is represented by its 1st membership function and x_3 is not used.

The population of Level IV is formed by a set of fuzzy rules, where each allele contains the index of the corresponding individual rule that is being included in the set. Null values indicate that the corresponding allele does not contribute to the inclusion of any rule to the set of fuzzy rules. The chromosome is represented by integer encoding. The length of the chromosome is determined by the maximum number of fuzzy rules. In the example of Fig. 1, Level IV of the GA hierarchy is illustrated by the 27th set of fuzzy rules that contains the 9th, 16th, and 2nd individual rules, where these rules are described/represented in Level III of the hierarchy (but only the 16th rule is illustrated at the Level III of Fig. 1).

Each individual at Level V represents a fuzzy system. The chromosome is represented by integer and real encoding. The first allele represents the aggregation method used in the antecedent part of the rules. In this paper, only the $\mathbf{t}_5$ t-norm is used for aggregation, where

$$a\,\mathbf{t}_5\,b = \frac{ab}{p_t + (1 - p_t)(a + b - ab)}, \tag{25}$$

p_t is represented by allele 2, and in this paper its range is defined by$p_t \in [0, 10]$. In allele 1, other aggregation operators can be used (see more in [22]). Allele 3 chooses a kth set of fuzzy rules specified

Algorithm 1 Proposed algorithm.

1. Set Generation = 1;
2. Initialize populations of all levels;
3. Compute the optimal parameters of the consequent of T-S fuzzy model for all individuals at Level V using the Least squares method (Section 2.1);
4. Compute the fitness of each individual, from Level V to Level I:
   (a) Level V - Fuzzy system: the fitness function is $f_{FS}(i) = 1/MSE(i)$, where $MSE(i)$ is the mean square error of the ith fuzzy system. The MSE is given by: $MSE(i) = \frac{1}{L}\sum_{g=1}^{L}(y_k - \hat{y}_k)^2$ where L is the number of data patterns, $\hat{y}_k$ the predicted output pattern and y_k is the target output pattern;
   (b) Level IV - Rule base: the fitness function is $f_{RB}(k) = max(f_{FS}(b), \cdots, f_{FS}(d))$, where $b, \cdots, d$ are the fuzzy systems at Level V that contain rule-base k (set of fuzzy rules);
   (c) Level III - Individual rule: the fitness function is $f_{IR}(j) = mean(f_{RB}(m), \cdots, f_{RB}(p))$, where $m, \cdots, p$ are the rule-bases at Level IV that contain individual rule j;
   (d) Level II - Partition set: the fitness function is $f_{PS}(q) = max(f_{FS}(x), \cdots, f_{FS}(z))$, where $x, \cdots, z$ are the fuzzy systems at Level V that contain partition set q;
   (e) Level I - Inputs and delays selection: the fitness function is $f_{IS}(l) = max(f_{FS}(e), \cdots, f_{FS}(h))$, where $e, \cdots, h$ are the fuzzy systems at Level V that contain the selection number l of inputs and delays;
5. Each level is evolved, considering it as a separate genetic algorithm. There may be only one stop common condition that is used for all the levels. If the stop condition does not hold, do:
   (a) Generation = Generation + 1;
   (b) For each level, apply the following evolutionary operators to form a new population: (1) selection, (2) crossover and (3) mutation;
   (c) For each level, replace the current population with the new evolved population;
   (d) Return to step 3.

at Level IV. Allele 4 contains a qth partition set individual at Level II, and Allele 5 represents the lth set of input variables and delays at Level I.

Fig. 1 presents an example of the encoding and the hierarchical relations. In this example, the ith fuzzy system at Level V uses the $\mathbf{t}_5$-norm (25) with the associated parameter $p_t = 1.6$, the 27th set of fuzzy rules at Level IV, the 3rd partition set of Level II, and the 23rd set of selected input variables and delays. The 27th set of fuzzy rules contains the 9th, 16th, and 2nd individual rules, where the 16th individual rule is composed by two input variables with membership functions 5 and 1, respectively, i.e.:

$$\begin{aligned} R_{16}: \quad & \text{IF } x_1(k) \text{ is "5", and } x_2(k) \text{ is "1"} \\ & \text{THEN } y_{16}(\mathbf{c}_{16}, [x_1, x_2, x_3]). \end{aligned} \tag{26}$$

The linguistic term "5" is defined in the 3rd chromosome at Level II, and the input variables x_1, x_2, and x_3 are defined/selected at Level I.

The main steps to learn/improve the T-S fuzzy model parameters are presented in Algorithm 1. The fitness functions of each individual for Levels I to V are defined in Algorithm 1. Each level of the genetic hierarchy is evolved separately as an independent genetic algorithm using its own population and its own different fitness function. However, since the (values of the) fitness functions of every levels depend on the populations of all the levels, then evolution of each level also influences the evolution of all other levels. The flowchart of Figure 3 describes the operation of the method for identification

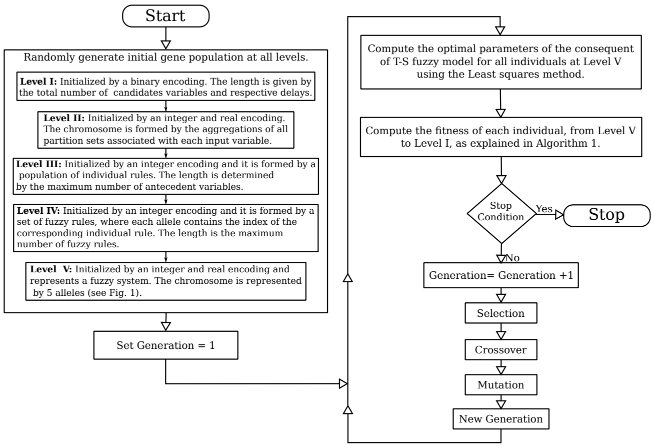

Figure 3: Flowchart of the hierarchical genetic algorithm for learning the T-S fuzzy system.

of the T-S fuzzy model using GAs.

3.2 Genetic Operators

As can be seen in Algorithm 1, the first genetic operator that is used, is selection. In this paper roulette wheel selection is used. The principle of roulette selection is a linear search through a roulette wheel with the slots in the wheel weighted in proportion to the individual's fitness values [24]. The second operator is single point crossover, where the two mating chromosomes are cut once at corresponding points and the sections after the cuts exchanged. The crossover point is selected randomly [24]. The last operator is the mutation: uniform mutation is employed. This operator replaces the value of a randomly selected allele with a uniform random value selected between the upper and lower bounds for that allele.

4 Experiments and Results

This section presents simulation results to demonstrate the feasibility, performance and effectiveness of the proposed SS design methodology. The proposed methodology is quantitatively compared with the work [9]. Two nonlinear system identification application problems are analyzed: the Box-Jenkins gas furnace benchmark study and the estimation of the flour concentration in the effluent of a real-world wastewater treatment system. In both simulations, a maximum of 5000 generations is used, and the population of each species is fixed: 60 individuals for Level V, 80 individuals for Level IV, 200 individuals for Level III, 15 individuals for Level II, and 200 individuals for Level I. The proposed methodology and the method of [9] were implemented in the Matlab Software with the main functions being implemented in the C language to reduce computational time. In the Box-Jenkins benchmark, the proposed methodology is additionally compared to the works [10, 18, 19, 17, 5].

4.1 Application to the Box-Jenkins Gas Furnace Data

The Box-Jenkins gas furnace process data[1] was recorded from a combustion process of a methane-air mixture, and consists of 296 data points $[y(t), u(t)]$. The input $u(t)$ is the gas flow rate into the furnace and the output $y(t)$ is the carbon dioxide (CO_2) concentration in the outlet gas. The sampling

[1]Provided by IEEE Neural Networks Council Standards Committee Working Group on Data Modeling Benchmarks. May be found at: http://www.stat.wisc.edu/~reinsel/bjr-data/gas-furnace .

Table 1: Comparison results for the Box-Jenkins problem.

Method	Number of rules	Number of inputs	Inputs	$f_{FS} = 1/MSE$
Du and Zhang [5]	4	3	$y(t-4)$, $u(t-4)$, $u(t-5)$	$\frac{1}{0.06} = 16.6667$
Delgado and Nagai [9]	10	4	$y(t-1)$, $y(t-2)$, $u(t-1)$, $u(t-2)$	13.8170
Kim *et al.* [10]	2	2	$y(t-1)$, $u(t-4)$	$\frac{1}{0.129} = 7.7519$
Y.M. Chen and Chun-Ta Li [11]	3	2	$y(t-1)$, $u(t-4)$	$\frac{1}{0.252^2} = 15.7470$
Liang Zhao *et al.* [14]	3	2	$y(t-1)$, $u(t-4)$	$\frac{1}{0.1275} = 7.8431$
Nie [17]	45	4	$y(t-1)$, $u(t-3)$, $u(t-4)$, $u(t-5)$	$\frac{1}{0.169} = 5.9172$
Sugeno and Yasukawa [18]	6	3	$y(t-1)$, $u(t-3)$, $u(t-4)$	$\frac{1}{0.190} = 5.2632$
Lin and Cunningham [19]	4	5	$y(t-1)$, $y(t-2)$, $u(t-3)$, $u(t-5)$, $u(t-6)$	$\frac{1}{0.261} = 3.8314$
The method proposed in this paper	10	7	$y(t-1)$, $y(t-2)$, $y(t-3)$, $y(t-4)$, $u(t-1)$, $u(t-5)$, $u(t-6)$	16.9299

interval is 9 [s]. To predict $y(t)$ the following combinations of process variables and delays are used as the candidates for inputs of the T-S model, which reduces to 290 the effective number data points: $[y(t-1), y(t-2), y(t-3), y(t-4), u(t-1), u(t-2), u(t-3), u(t-4), u(t-5), u(t-6)]$. The input variables were divided into two groups: one group for variables $[y(t-1), y(t-2), y(t-3), y(t-4)]$ and the other for $[u(t-1), u(t-2), u(t-3), u(t-4), u(t-5), u(t-6)]$. Level II of the hierarchical genetic fuzzy system (Sec. 3) was configured so that (i) variables belonging to the same group have the same range of possible values, and (ii) for each individual, all variables within each group were forced to share the same partition set.

Both Algorithm 1, and the method proposed in [9], were applied to perform the estimation of the output $y(t)$ in the Box-Jenkins gas furnace dataset. The complete dataset was used both for training and evaluation. Table 1 presents the results of both the presently proposed methodology (Algorithm 1), and the method proposed in [9]. Table 1 also presents the results of the application of the seven methods presented from [10, 18, 19, 17, 5, 11, 14] to the Box-Jenkins dataset (these results were obtained in [5, 11, 14]). Drawbacks of these approaches have been discussed in Sec. 1. Here, the discussion will be expanded with a quantitative analysis. The performance index used to evaluate and compare the methods is f_{FS}, which is the same as the fuzzy system fitness function (Level V) defined in Algorithm 1. As can be seen in Table 1 the algorithm proposed in this paper outperforms the method proposed in [9], where the same parameters were used in both methods, and also outperforms all other methods shown in Table 1.

Fig. 4 shows the predicted and desired (real) values of the target variable to be estimated, and Fig. 5 shows the prediction error. The membership functions learned by the coevolutionary genetic algorithm are shown in Figs. 6 and 7. Concerning variable selection, Table 1 shows the difference between the T-S fuzzy model input variables and delays selected by both methods. The presently proposed method selects a larger number of input variables than the method proposed in [9], but shows higher precision in the prediction results.

4.2 Application to Wastewater Treatment System

In the second application experiment the objective is to estimate the flour concentration in the effluent of a real-world urban wastewater treatment plant (WWTP). The dataset of plant variables that is available for learning consists of 10 input variables, $u_1 \dots u_{10}$, and one target output variable to be

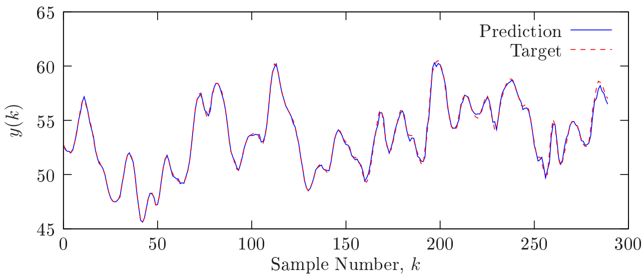

Figure 4: Modeling performance of the proposed algorithm for the Box-Jenkins furnace data.

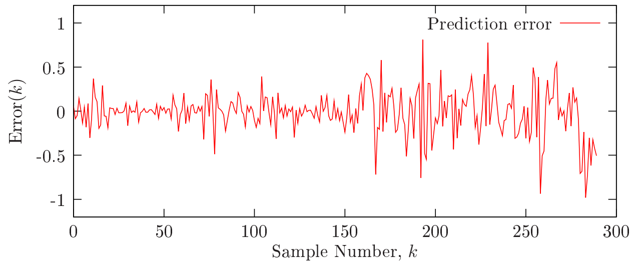

Figure 5: Prediction error of the proposed algorithm for the Box-Jenkins furnace data.

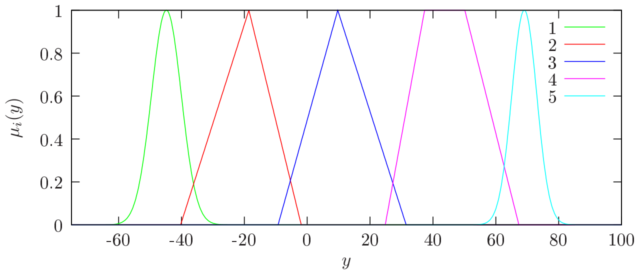

Figure 6: Membership functions for input variable y of the T-S fuzzy model. y is the output of the Box-Jenkins process.

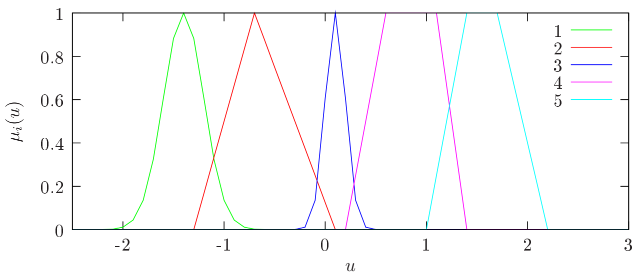

Figure 7: Membership functions for input variable u of the T-S fuzzy model. u is the input of the Box-Jenkins process.

estimated, $u_{11} = y$. The variables correspond to physical values, such as pH, turbidity, color of the water and others. The plant variables are described in Table 2. The input variables are measured

Table 2: Variables of the wastewater treatment plant dataset.

Variables	Description
u_1	Amount of chlorine in the influent;
u_2	Amount of chlorine in the effluent;
u_3	Turbidity in the raw water;
u_4	Turbidity in the influent;
u_5	Turbidity in the effluent;
u_6	Ph in the raw water;
u_7	Ph in the effluent;
u_8	Color in the raw water;
u_9	Color in the influent;
u_{10}	Color in the effluent;
$u_{11} = y$	Flour in the effluent.

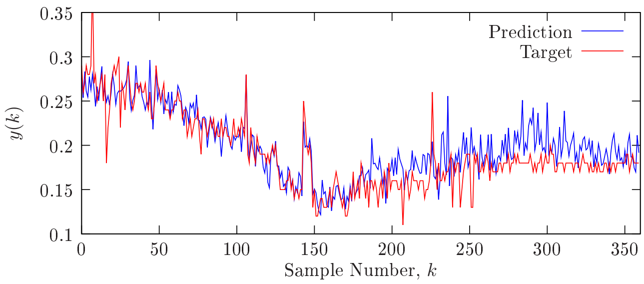

Figure 8: Modeling performance of the proposed algorithm for wastewater treatment system data set.

online by plant sensors, and the output variable in the dataset is measured by laboratory analysis. The sampling interval is 2 [hours]. The proposed algorithm jointly selects the best variables and delays, and learns the best T-S fuzzy system for the flour prediction setting. The second half of the dataset was used for training and the remaining data was used for evaluation. The first three delayed versions of each variable were chosen as candidates for inputs of the T-S model. Specifically, the following combinations of process variables and delays are used as the candidates for inputs of the T-S model to predict $y(t)$: $[u_1(t-1), u_1(t-2), u_1(t-3), \ldots, u_{11}(t-1), u_{11}(t-2), u_{11}(t-3)]$. The input variables were divided into two groups: one group for all the delayed versions of u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8, and $u_{11} = y$, and the other group for all the delayed versions of u_9, and u_{10}. Level II of the hierarchical genetic fuzzy system (Sec. 3) was configured so that (i) variables belonging to the same group have the same range of possible values, and (ii) for each individual, all variables within each group were forced to share the same partition set.

Fig. 8 shows the predicted and desired (real) values of the target variable to be estimated, for the WWTP experiment. The prediction error can be seen in Fig. 9. As can be seen in Figs. 8 and 9 the accuracy of the modeling is acceptable. The membership functions obtained by the coevolutionary genetic algorithm are shown in Figs. 10 and 11.

Numerical results comparing the performance of the proposed method and the work [9] are presented in Table 3. The same parameters were used in both methods. As can be seen the higher fitness function $f_{FS} = 1/MSE$ is obtained with the method proposed in this paper which includes automatic selection of input variables and respective delays. The proposed method selects a larger number of (variable,

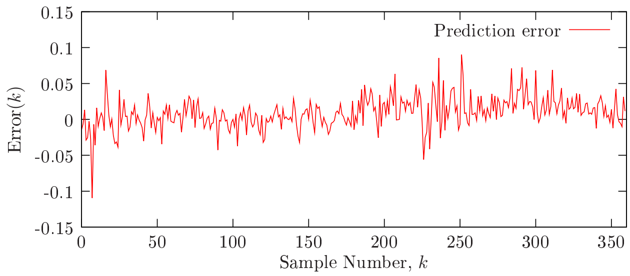

Figure 9: Prediction error of the proposed algorithm for wastewater treatment system data set.

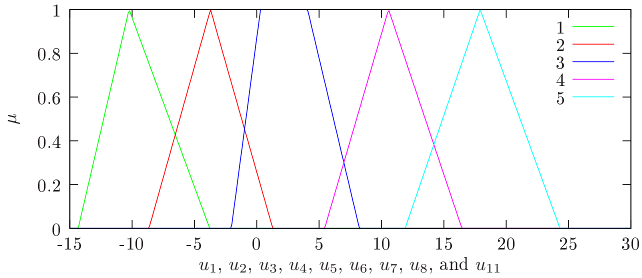

Figure 10: Membership functions for input variables u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8, and $u_{11} = y$ of the T-S fuzzy model. $u_1, \ldots, u_8$ are inputs of the process, and $u_{11} = y$ is the output of the process.

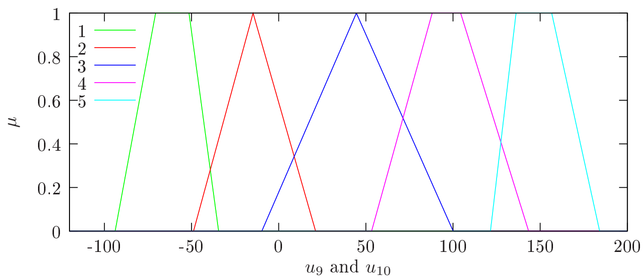

Figure 11: Membership functions for input variables u_9, and u_{10} of the T-S fuzzy model. u_9, u_{10} are inputs of the process.

delay) pairs.

5 Conclusion

This paper proposed a new method for SS design for industrial applications based on the Takagi-Sugeno fuzzy model. A coevolutionary genetic algorithm is used to identify a T-S fuzzy model from input/output data to approximate the unknown nonlinear process. The proposed method is an automatic tool for SS design because it does not require any prior knowledge concerning the structure (e.g. the number of rules) and the database (e.g. antecedent fuzzy sets) of the T-S fuzzy model, and concerning the selection of the adequate input variables and their respective time delays for the prediction setting. Input variables and delay selection, antecedent aggregation operators, fuzzy rules, type, location and shape of membership functions are learned by coevolutionary hierarchical GAs. To vali-

Table 3: Comparison results for the wastewater treatment system dataset.

Method	Number of rules	Number of inputs	Inputs	$f_{FS} = 1/MSE$
Delgado and Nagai [9]	20	10	$u_1(t-1), u_2(t-1), u_3(t-1), u_4(t-1), u_5(t-1), u_6(t-1), u_7(t-1), u_8(t-1), u_9(t-1), u_{10}(t-1)$	1328.1
The method proposed in this paper	20	17	$u_1(t-2), u_2(t-1), u_3(t-2), u_4(t-1), u_4(t-2), u_5(t-1), u_5(t-2), u_6(t-2), u_6(t-3), u_7(t-2), u_7(t-3), u_8(t-1), u_8(t-3), u_9(t-1), u_{10}(t-1), u_{10}(t-2), u_{11}(t-1)$	1804.4

date and demonstrate the performance and effectiveness of the proposed methodology, the algorithm was applied on two prediction problems. The first problem is the Box-Jenkins benchmark problem, and the second is the estimation of the flour concentration in the effluent of a real-world wastewater treatment system. The presented results have shown that the developed evolving T-S fuzzy model can identify the nonlinear systems satisfactorily with appropriate input variables and delay selection, and with a reasonable number of rules. The proposed methodology is able to design all the parts of the T-S fuzzy prediction model. Moreover, presented comparison results indicate that the proposed method outperforms other previously proposed methods for the design of prediction models, including a method previously proposed for the design of T-S models. A limitation in the proposed method is the computational time. To improve the search performance, in the future, other optimization approaches of the antecedent membership functions will investigated, and some self-adaptive strategies will be added.

Acknowledgment

This work was supported by Mais Centro Operacional Program, financed by European Regional Development Fund (ERDF), and Agência de Inovação (AdI) under Project SInCACI/3120/2009.

Jérôme Mendes and Francisco Souza were supported by Fundação para a Ciência e a Tecnologia (FCT) under grants SFRH/BD/63383/2009, and SFRH/BD/63454/2009, respectively.

References

[1] L. Fortuna, S. Graziani, A. Rizzo, Soft Sensors for Monitoring and Control of Industrial Processes (Advances in Industrial Control), Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[2] P. Kadlec, B. Gabrys, S. Strandt, Data-driven soft sensors in the process industry, Computers & Chemical Engineering 33 (4) (2009) 795–814.

[3] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Transactions on Systems, Man, and Cybernetics 15 (1) (1985) 116–132.

[4] C. Lo, P. Chan, Y. Wong, A. Rad, K. Cheung, Fuzzy-genetic algorithm for automatic fault detection in hvac systems, Applied Soft Computing 7 (2) (2007) 554–560.

[5] H. Du, N. Zhang, Application of evolving takagi-sugeno fuzzy model to nonlinear system identification, Applied Soft Computing 8 (1) (2008) 676–686.

[6] S. Ghosh, S. Maka, Genetic algorithm based narx model identification for evaluation of insulin sensitivity, Applied Soft Computing 11 (1) (2011) 221–226.

[7] L. P. Tan, A. Lotfi, E. Lai, J. B. Hull, Soft computing applications in dynamic model identification of polymer extrusion process, Applied Soft Computing 4 (4) (2004) 345–355.

[8] F. Pettersson, N. Chakraborti, H. Saxén, A genetic algorithms based multi-objective neural net applied to noisy blast furnace data, Applied Soft Computing 7 (1) (2007) 387–397.

[9] M. R. Delgado, E. Y. Nagai, L. V. R. de Arruda, A neuro-coevolutionary genetic fuzzy system to design soft sensors, Soft Computing - A Fusion of Foundations, Methodologies and Applications 13 (5) (2009) 481–495.

[10] M.-S. Kim, C.-H. Kim, J.-J. Lee, Evolving compact and interpretable takagi-sugeno fuzzy models with a new encoding scheme, IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics 36 (5) (2006) 1006–1023.

[11] Y. Chen, C.-T. Lin, Dynamic parameter optimization of evolutionary computation for on-line prediction of time series with changing dynamics, Applied Soft Computing 7 (4) (2007) 1170–1176.

[12] H. P. H. Anh, K. K. Ahn, Identification of pneumatic artificial muscle manipulators by a mga-based nonlinear narx fuzzy model, Mechatronics 19 (1) (2009) 106–133.

[13] S.-T. Tzeng, Design of fuzzy wavelet neural networks using the ga approach for function approximation and system identification, Fuzzy Sets and Systems 161 (19) (2010) 2585–2596.

[14] L. Zhao, F. Qian, Y. Yang, Y. Zeng, H. Su, Automatically extracting t-s fuzzy models using cooperative random learning particle swarm optimization, Applied Soft Computing 10 (3) (2010) 938–944.

[15] R. Yusof, R. Z. A. Rahman, M. Khalid, M. F. Ibrahim, Optimization of fuzzy model using genetic algorithm for process control application, Journal of the Franklin Institute In Press, Corrected Proof (2010).

[16] K. Warne, G. Prasad, S. Rezvani, L. Maguire, Statistical and computational intelligence techniques for inferential model development: a comparative evaluation and a novel proposition for fusion, Engineering Applications of Artificial Intelligence 17 (8) (2004) 871–885.

[17] J. Nie, Constructing fuzzy model by self-organizing counterpropagation network, IEEE Transactions on Systems, Man and Cybernetics 25 (6) (1995) 963–970.

[18] M. Sugeno, T. Yasukawa, A fuzzy-logic-based approach to qualitative modeling, IEEE Transactions on Fuzzy Systems 1 (1) (1993) 7–31.

[19] Y. Lin, G. A. Cunningham, III, A new approach to fuzzy-neural system modeling, IEEE Transactions on Fuzzy Systems 3 (2) (1995) 190 –198.

[20] M. R. Delgado, F. V. Zuben, F. Gomide, Hierarchical genetic fuzzy systems, Information Sciences 136 (1-4) (2001) 29 – 52.

[21] H. Ying, General miso takagi-sugeno fuzzy systems with simplified linear rule consequent as universal approximators for control and modeling applications, IEEE International Conference on Systems, Man, and Cybernetics 2 (1997) 1335–1340.

[22] L.-X. Wang, A Course in Fuzzy Systems and Control, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1997.

[23] A. Ben-Israel, T. N. Greville, Generalized Inverses: Theory and Applications, 2nd Edition, Springer-Verlag, New York, USA, 2003.

[24] S. N. Sivanandam, S. N. Deepa, Introduction to Genetic Algorithms, Springer Publishing Company, Incorporated, 2007.