PROD. TYPE: COM
PP: 1-38 (col.fig.: Nil)
FSS4439 DTD VER: 5.0.1
ED: Vinutha
PAGN: Murthy.N – SCAN: Roopa
ARTICLE IN PRESS

ELSEVIER

Available online at www.sciencedirect.com
SCIENCE DIRECT®

Fuzzy Sets and Systems III (IIII) III–III

FUZZY sets and systems

www.elsevier.com/locate/fss

Multi-objective hierarchical genetic algorithm for interpretable fuzzy rule-based knowledge extraction☆

Hanli Wang[a,c], Sam Kwong[a,*], Yaochu Jin[b], Wei Wei[c], K.F. Man[d]

[a]*Department of Computer Science, City University of Hong Kong, 83 Tatchee Ave, Kowloon, Hong Kong, People's Republic of China*
[b]*Future Technology Research, Honda R&D Europe (D), 67073 Offenbach/Main, Germany*
[c]*College of Electrical Engineering, Zhejiang University, Hangzhou 310027, People's Republic of China*
[d]*Department of Electronic Engineering, City University of Hong Kong, Hong Kong*

Abstract

A new scheme based on multi-objective hierarchical genetic algorithm (MOHGA) is proposed to extract interpretable rule-based knowledge from data. The approach is derived from the use of multiple objective genetic algorithm (MOGA), where the genes of the chromosome are arranged into control genes and parameter genes. These genes are in a hierarchical form so that the control genes can manipulate the parameter genes in a more effective manner. The effectiveness of this chromosome formulation enables the fuzzy sets and rules to be optimally reduced. Some important concepts about the interpretability are introduced and the fitness function in the MOGA will consider both the accuracy and interpretability of the fuzzy model. In order to remove the redundancy of the rule base proactively, we further apply an interpretability-driven simplification method to newborn individuals. In our approach, we first apply the fuzzy clustering to generate an initial rule-based model. Then the multi-objective hierarchical genetic algorithm and the recursive least square method are used to obtain the optimized fuzzy models. The accuracy and the interpretability of fuzzy models derived by this approach are studied and presented in this paper. We compare our work with other methods reported in the literature on four examples: a synthetic nonlinear dynamic system, a nonlinear static system, the Lorenz system and the Mackey–Glass system. Simulation results show that the proposed approach is effective and practical in knowledge extraction.

Keywords: Interpretability; Hierarchical chromosome formulation; Fuzzy rule base simplification; Multi-objective decision making; Recursive least square method

☆ This work is supported by City University Strategic Grant 7001488.
* Corresponding author. Tel.: +852-2788-7704; fax: +852-2788-8614.
E-mail address: cssamk@cityu.edu.hk (S. Kwong).

0165-0114/$ - see front matter © 2004 Published by Elsevier B.V.
doi:10.1016/j.fss.2004.07.013

1. Introduction

The fundamental concept of fuzzy reasoning was first introduced by Zadeh [35] in 1973 and since then, its use in engineering disciplines has been widely studied. One of the most important motivations for building up a fuzzy model is to let users gain a deep insight into an unknown system through the easily understandable fuzzy rules. Another main attraction undoubtedly lies in the unique characteristics that the fuzzy logic systems possess. They are capable of handling complex, nonlinear, and sometimes mathematically intangible dynamic systems. However, when the fuzzy rules are extracted by the traditional learning methods, there is often a lack of interpretability in the resulting fuzzy rules. Consequently, two common problems are found: (1) the number of rules is usually larger than necessary, and (2) the topology of the fuzzy sets is inappropriate. So there is always a trade-off between the interpretability and the accuracy of the fuzzy model constructed from sampling data. Recently, attentions have been increasingly paid to improve the interpretability of fuzzy systems, and several approaches have been proposed [4,14,15,17,18,24–26,32–34]. Genetic algorithm (GA) is one of such techniques that received a lot of attention owing to its parallel characteristics and its ability in searching for optimal solutions in irregular and high-dimensional solution spaces.

In this paper, we propose a new and efficient approach to construct first-order TS fuzzy models from data, considering both their accuracy and interpretability. First, we use the fuzzy clustering method to preprocess the sampling data and to form the rule antecedents of the initial model. Then, the recursive least square (RLS) method is applied to determine the rule consequents. Thus, a reasonably good initial model instead of random ones is first obtained for the GA. We will then use the multi-objective hierarchical genetic algorithm (MOHGA) to generate the optimized fuzzy models. In this step, we apply the hierarchical chromosome formulation so that it can perform the simultaneous optimization of the rule antecedents and the number of rules. Then we use the RLS method instead of GA to compute the rule consequents. Thus, it can greatly improve the search efficiency of GA and exploit the training data in a more effective way. When comparing our simulation results with those of other approaches in the literature, it shows that the combination of MOHGA and RLS is a very effective approach to obtain interpretable TS fuzzy models of high accuracy. In order to reduce the burden of the GA optimization, we apply an interpretability-driven rule base simplification (IDRBS) method to the newborn individuals during evolutionary optimization to reduce the redundancy of the fuzzy rule base.

The paper is organized as follows. Section 2 discusses the interpretability issues of fuzzy systems. The generation of an initial fuzzy model is given in Section 3. Then the proposed multi-objective hierarchical genetic algorithm and the interpretability-driven rule base simplification method are introduced in Section 4. In Section 5, we compare the proposed approach with existing methods on four examples taken from the literature: a synthetic nonlinear dynamic system, a nonlinear static system, the Lorenz system and the Mackey–Glass system. Comparative simulation results demonstrate that the proposed approach can obtain fuzzy models with better interpretability and with comparable or higher accuracy. Finally, Section 6 draws the conclusion.

2. Interpretability of fuzzy systems

A good method for constructing fuzzy models should not aim to find the best approximation of data only, but to extract knowledge from sampling data in the form of fuzzy rules that can be easily understood and

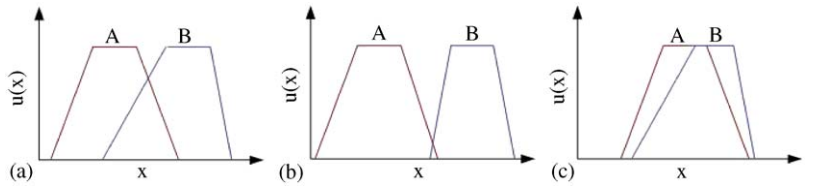


Fig. 1. Fuzzy partitioning: (a) overlap moderately, (b) overlap too little, (c) overlap too much.

interpreted. Interpretability (also called transparency) of fuzzy systems has not received much attention in the field of fuzzy modeling until the last few years. One reason is that most researchers believe that fuzzy rules are always easy for human beings to understand. However, it is not necessarily true especially for complex systems. In the following, we will discuss some important concepts about the interpretability of fuzzy systems.

2.1. Completeness and distinguishability

The discussion of completeness and distinguishability is necessary if fuzzy systems are obtained by automatically learning from data. The partitioning of fuzzy sets for each fuzzy variable should be complete and well distinguishable. The completeness of fuzzy systems means that for each input variable, at least one fuzzy set is fired. We formulate this idea with the following definition.

Definition 2.1 (Completeness). For each input variable x_i (an element of the input vector $\mathbf{X}=[x_1, x_2, \ldots, x_n]^{\mathrm{T}}$), there exists M_i fuzzy sets represented by $A_1(x), A_2(x), \ldots, A_{Mi}(x)$. Then the partition of the fuzzy sets is complete if the following conditions are satisfied:

$$\forall x_i \in U_i, \quad i \in [0, \ldots, n], \quad \exists A_j(x_i) > 0, j \in [1, \ldots, M_i], \tag{1}$$

where U_i is the universe of x_i, n is the dimension of the input vector.

The concepts of completeness and distinguishability of fuzzy systems are usually expressed through a fuzzy similarity measure in the literature [4,10,19,28]. This similarity measure can be interpreted in many different ways depending on the context of the application. However, one important definition given in [28] is that *similarity between fuzzy sets as the degree to which the fuzzy sets are equal.* Based on the similarity measure, three kinds of similarities can be identified: (1) similarity between two fuzzy sets for a given fuzzy variable; (2) similarity of a fuzzy set to the universal set U ($u_U(x) = 1, \forall x \in X$); and (3) similarity of a fuzzy set to a singleton set. We will present the interpretability-driven rule base simplification method to manage these three kinds of similarities in Section 4. In fact, if the similarity of two neighboring fuzzy sets is zero or too small, it means that the fuzzy partitioning in this fuzzy variable is incomplete or the two fuzzy sets do not have enough overlap. On the other hand, if the similarity is too big, then it indicates that the two fuzzy sets overlap too much with each other and the distinguishability between them is poor (Fig. 1).

In the following, let A and B be two fuzzy sets of fuzzy variable x (on the universe U) with the membership functions $u_A(x)$ and $u_B(x)$, respectively. The symbol s represents the similarity value of

these two fuzzy sets: $s = S(A, B)$, $s \in [0, 1]$. A similarity measure will be considered as a possible criterion if it satisfies the following four conditions [28]:

(1) Non-overlapping fuzzy sets should be considered totally non-equal, $s = 0$.

$$S(A, B) = 0 \quad \Leftrightarrow \quad u_A(x)u_B(x) = 0, \quad \forall x \in U \tag{2}$$

(2) Overlapping fuzzy sets should have a similarity value $s > 0$.

$$S(A, B) > 0 \quad \Leftrightarrow \quad \exists x \in U, \quad u_A(x)u_B(x) \neq 0. \tag{3}$$

(3) Only equal fuzzy sets should have a similarity value $s = 1$.

$$S(A, B) = 1 \quad \Leftrightarrow \quad u_A(x) = u_B(x), \quad \forall x \in U. \tag{4}$$

(4) Similarity between two fuzzy sets should not be influenced by scaling or shifting the domain on which they are defined:

$$\begin{aligned} &S(A', B') = S(A, B), \quad u_{A'}(kx + l) = u_A(x), \\ &u_{B'}(kx + l) = u_B(x), \quad k, l \in IR, \ \ k > 0, \ \forall x \in U. \end{aligned} \tag{5}$$

We use the following similarity measure which satisfies the above four criteria to determine the similarity between fuzzy sets:

$$S(A, B) = \frac{M(A \cap B)}{M(A \cup B)} = \frac{M(A \cap B)}{M(A) + M(B) - M(A \cap B)}, \tag{6}$$

where $M(A)$ denotes the cardinality of the fuzzy set A, and the operators $\cap$ and $\cup$ represent the intersection and union, respectively. There are several methods to calculate the similarity. One form on the continuous domain is given in [19]

$$M(A) = \int_{-\infty}^{\infty} u_A(x)\,\mathrm{d}x \tag{7}$$

Another form in [25,26] is described as

$$S(A, B) = \frac{\sum_{j=1}^{m}[u_A(x_j) \wedge u_B(x_j)]}{\sum_{j=1}^{m}[u_A(x_j) \vee u_B(x_j)]} \tag{8}$$

on a discrete universe $U = \{x_j \mid j = 1, 2, \ldots, m\}$. $\wedge$ and $\vee$ in Eq. (8) are the minimum and maximum operators, respectively. In our approach, we use the latter to calculate the similarity of fuzzy sets because it is computationally simple and effective.

However, using the similarity measure does not necessarily guarantee a sound evaluation of the distinguishability of fuzzy systems in the cases where a fuzzy set covers another fuzzy set. In Fig. 2(a)–(c), we can see that the similarity between A and B is moderate, but the fuzzy sets distribution of A and B

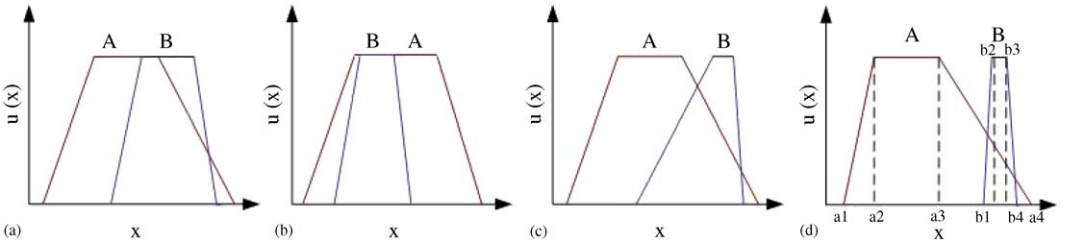


Fig. 2. (a–d) Four cases: fuzzy set A covers fuzzy set B.

is bad because the definition domain of A covers that of B. We use the term *covering* to describe these cases and it is defined as follows.

Definition 2.2 (Covering). For a fuzzy variable x: if there are two fuzzy sets represented by $A(x)$ and $B(x)$, then fuzzy set A is said to cover fuzzy set B if and only if the following conditions are satisfied:

$$\begin{aligned} &U(A) = \{x \mid u_A(x) > 0, x \in U\},\\ &U(B) = \{x \mid u_B(x) > 0, x \in U\},\\ &U(A) \supseteq U(B), \end{aligned} \tag{9}$$

where U is the universe of x, $u_A(x)$ and $u_B(x)$ represent the membership functions of fuzzy sets A and B, respectively.

We will discuss the covering issues with the aid of the trapezoidal membership functions. This is because other types of membership functions can also be easily transformed to the trapezoidal type such as the triangular and Gaussian membership functions. Let the parameter vector $[a1, a2, a3, a4]$ represents the membership function parameters of fuzzy set A and $[b1, b2, b3, b4]$ of fuzzy set B (Fig. 2(d)), where $a1$ is the lower bound of the support of the fuzzy A, $a2$ is the left center, $a3$ is the right center and $a4$ is the upper bound. We define the following terms to describe the degree of covering between fuzzy sets A and B.

Definition 2.3 (Complete covering). Fuzzy set A completely covers fuzzy set B if the following conditions are satisfied (Fig. 2(b)):

$$\begin{aligned} &a1 \leqslant b1, \quad a2 \leqslant b2, \quad a3 \geqslant b3, \quad a4 \geqslant b4,\\ &a1 \leqslant a2 \leqslant a3 \leqslant a4, \quad b1 \leqslant b2 \leqslant b3 \leqslant b4. \end{aligned} \tag{10}$$

Definition 2.4 (Restricted covering). Fuzzy set A restrictedly covers fuzzy set B if the following conditions are satisfied (Fig. 2(a)):

$$\begin{aligned} &a1 \leqslant b1, \quad a4 \geqslant b4, \quad [b2, b3] \not\subset [a2, a3], \quad [b2, b3] \cap [a2, a3] \neq \emptyset,\\ &a1 \leqslant a2 \leqslant a3 \leqslant a4, \quad b1 \leqslant b2 \leqslant b3 \leqslant b4. \end{aligned} \tag{11}$$

Definition 2.5 (Relaxed covering). Fuzzy set A relaxedly covers fuzzy set B if the following conditions are satisfied (Fig. 2(c) and (d)):

$$a1 \leqslant b1, \quad a4 \geqslant b4, \quad [b2, b3] \cap [a2, a3] = \emptyset,$$
$$a1 \leqslant a2 \leqslant a3 \leqslant a4, \quad b1 \leqslant b2 \leqslant b3 \leqslant b4. \tag{12}$$

For fuzzy systems with good interpretability, it is preferable that one fuzzy set does not cover another one. In Section 4, we will discuss how to use the similarity value and the covering ideas to simplify the rule base of the fuzzy systems.

2.2. Non-redundancy

To improve the interpretability of fuzzy rules, it is also needed to reduce the redundancy of fuzzy rule base. In [7], a rule is said to be redundant with respect to the rule base if it brings nothing new to the rule base. From a computational point of view, it is important to improve the non-redundancy of fuzzy rule base since redundancy will sometimes lead to useless computations.

In our work, non-redundancy of the rule base is based on the similarity degree among fuzzy rules. Definitions about the similarity among fuzzy rules are given in [19]. The similarity of rule antecedents (SRA) and the similarity of rule consequents (SRC) are calculated with the help of aforementioned fuzzy similarity measure. Considering two rules in the rule base:

R_i: If x_1 is $A_{i1}(x_1)$ and x_2 is $A_{i2}(x_2)$ and $\ldots x_n$ is $A_{in}(x_n)$, then y_1 is $B_{i1}(y_1)$ and $\ldots y_m$ is $B_{im}(y_m)$,

R_j: If x_1 is $A_{j1}(x_1)$ and x_2 is $A_{j2}(x_2)$ and $\ldots x_n$ is $A_{jn}(x_n)$, then y_1 is $B_{j1}(y_1)$ and $\ldots y_m$ is $B_{jm}(y_m)$.

Then SRA and SRC of these two rules are defined as follows:

$$SRA(i, j) = \min_{k=1}^{n} S(A_{ik}, A_{jk}) \tag{13}$$

$$SRC(i, j) = \min_{k=1}^{m} S(B_{ik}, B_{jk}). \tag{14}$$

Since we apply the *Takagi–Sugeno* (TS) fuzzy system [30] and use the recursive least square method to obtain the rule consequents, we do not consider the SRC and use the following form to calculate the non-redundancy value among the fuzzy rules:

$$NRdd(R_i, R_j) = 1 - SRA(i, j). \tag{15}$$

One characteristic of the above definition of non-redundancy is that the degree of non-redundancy tends to be high, if the antecedents of the two rules are very different. If the non-redundancy value of the two rules is equal to zero, it means that the antecedents of these rules are the same. So we can eliminate one of them from the rule base resulting in a more compact fuzzy system. We apply Eq. (15) is applied to compute the non-redundancy value for the following two reasons: (1) it is easy and simple to implement, and (2) the other is that with Eq. (38), which is a maximum problem to reduce the similarity degree

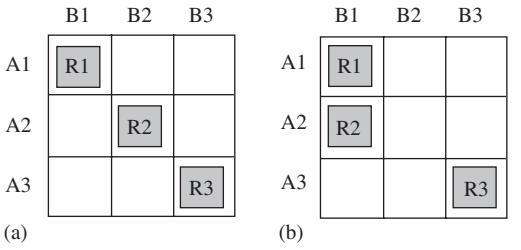


Fig. 3. A fuzzy system with two input variables (three fuzzy sets for each variable) and three rules: (a) sufficient utility, (b) insufficient utility because fuzzy set $B2$ is not utilized by any rules.

among fuzzy rules in the rule base, we can decrease the possibility that rule antecedents of two rules are very similar and the corresponding rule consequents are very different. This is a reason that we use the MOHGA and incorporate the non-redundancy issue as one of the multiple objectives.

2.3. Compactness

A compact fuzzy system means that it has the minimal number of fuzzy sets and fuzzy rules. In addition, the number of fuzzy variables is also worth being considered. In this paper, we only consider the following two issues: the number of fuzzy sets and the number of fuzzy rules for evaluating the compactness of fuzzy systems. A compact fuzzy system is always desirable when the number of input variables increases.

2.4. Utility

We noted that even if the partitioning of the fuzzy variables is complete and distinguishable, it is not guaranteed that each of the fuzzy sets be used by at least one rule. We use the term *utility* to describe such cases. If a fuzzy system is of *sufficient utility*, then all of the fuzzy sets are utilized as antecedents or consequents by the fuzzy rules. Whereas, a fuzzy system of *insufficient utility* indicates that there exists at least one fuzzy set that is not utilized by any of the rules (Fig. 3(b)). In our proposed approach, we will impose some constraints on the hierarchical chromosome to guarantee the sufficient utility of fuzzy systems.

3. Initial fuzzy model

In our proposed approach, we use the fuzzy clustering method to form the rule antecedents of the initial TS fuzzy model, then we apply the recursive least square method to specify the rule consequents. TS fuzzy system [30] is very suitable for the approximation of dynamic systems. Instead of using a linguistic term with an associated membership function, the consequents of the TS fuzzy system are usually constant values (singletons) or expressed as the functions of the inputs:

$$\mathrm{R}_i: \text{ If } x_1 \text{ is } A_{i1} \text{ and } x_2 \text{ is } A_{i2} \text{ and, } \ldots, \text{ and } x_n \text{ is } A_{in}, \text{ then } b_i = g_i(x_1, x_2, \ldots, x_n).$$

The choice of the function g_i depends on the application being considered. The first-order TS model is very common and effective:

$$g_i(x_1, x_2, \ldots, x_n) = a_{i1}x_1 + \cdots + a_{in}x_n + a_{i0}. \tag{16}$$

Here $\mathbf{x} = [x_1, x_2, \ldots, x_n]^{\mathrm{T}}$ is the input vector, b_i is the output of the ith rule, and $A_{i1}, A_{i2}, \ldots, A_{in}$ are the antecedent fuzzy sets. The model output is expressed as follows:

$$y = \frac{\sum_{i=1}^{R} b_i u_i(x)}{\sum_{i=1}^{R} u_i(x)}, \tag{17}$$

where R is the total number of rules, u_i is the fire-strength, also called weight of the ith rule:

$$u_i(x) = u_{A_{i1}}(x_1) \wedge u_{A_{i2}}(x_2) \wedge \cdots \wedge u_{A_{in}}(x_n), \quad i = 1, 2, \ldots, R, \tag{18}$$

$\wedge$ is the *and* operator, minimum and product are the most common *and* operators.

The initial fuzzy model is obtained in two steps. First, we use the fuzzy C-means clustering method [1,10,13] to determine the rule antecedents. Then the recursive least square method is implemented to calculate the consequents from the rule antecedents. For clustering, a regression matrix $\mathbf{X} = [\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N]^{\mathrm{T}}$ and an output matrix $\mathbf{Y} = [y^1, y^2, \ldots, y^N]^{\mathrm{T}}$ are constructed from the sampling data, where N is the number of data pairs. However, fuzzy clustering can be done using the input–output data, input data only, or output data only. In our approach, we want to use all of the available information. We therefore apply the clustering to the product space $\mathbf{X} \times \mathbf{Y}$. Then we project the fuzzy partition matrix already obtained by the clustering method to each of the input variables and approximate the projections with parametric functions.

Concerning the least square estimation methods, the batch least square (BLS) and the recursive least square (RLS) methods are two common choices. While the BLS method has proven to be very successful for a variety of applications [25,26], it is a "batch" method by its very nature [23]. For a small number of sampling data, we could clearly repeat the batch calculation. As more data are gathered, it is almost impossible to compute the inverse of the sampling data matrix $\mathbf{X}^{\mathrm{T}}\mathbf{X}$. Therefore, the RLS method is selected because it allows us to update the parameter vectors recursively. Considering the TS fuzzy system given by Eqs. (16)–(18), we can rewrite Eq. (18) in the following form:

$$y = \frac{\sum_{i=1}^{R} a_{i0}u_i(x)}{\sum_{i=1}^{R} u_i(x)} + \frac{\sum_{i=1}^{R} a_{i1}x_1u_i(x)}{\sum_{i=1}^{R} u_i(x)} + \cdots + \frac{\sum_{i=1}^{R} a_{in}x_nu_i(x)}{\sum_{i=1}^{R} u_i(x)}. \tag{19}$$

Given

$$\xi_i(x) = \frac{u_i(x)}{\sum_{i=1}^{R} u_i(x)}, \tag{20}$$

$$\xi(x) = [\xi_1(x), \ldots, \xi_R(x), x_1\xi_1(x), \ldots, x_1\xi_R(x), \ldots, x_n\xi_1(x), \ldots, x_n\xi_R(x)]^{\mathrm{T}}, \tag{21}$$

$$\theta = [a_{10}, \ldots, a_{R0}, a_{11}, \ldots, a_{R1}, \ldots, a_{1n}, \ldots, a_{Rn}]^{\mathrm{T}}, \tag{22}$$

so that

$$f(x \mid \theta) = \theta^{\mathrm{T}}\xi(x) \tag{23}$$

represents the TS fuzzy system. We use the RLS method to train the parameter vectors $\boldsymbol{\theta}$:

$$P(k) = \frac{1}{\lambda}(I - P(k-1)\xi^k(\lambda I + (\xi^k)^{\mathrm{T}} P(k-1)\xi^k)^{-1}(\xi^k)^{\mathrm{T}})P(k-1),$$
$$\theta(k) = \theta(k-1) + P(k)\xi^k(y^k - (\xi^k)^{\mathrm{T}}\theta(k-1)), \tag{24}$$

where k is the time index, $0 \leqslant k \leqslant N$, λ is the forgetting factor. We need to initialize the RLS method at the time step $k = 0$. In our approach we set $\theta(0) = 0$ and $P(0) = aI$ for some large $a > 0$ (for example 10,000), $\lambda = 1$.

4. Multiobjective hierarchical genetic algorithm and interpretability-driven rule base simplification method

In this section, we will discuss how to use the multi-objective hierarchical genetic algorithm to obtain good candidates for the data-driven fuzzy modeling. A hierarchical chromosome formulation is used to represent the individual solutions with a rule matrix structure. In each generation, an interpretability-driven simplification method is applied to newborn individuals in order to actively reduce the redundancy of the fuzzy system. Unlike other GA-based methods for generating fuzzy rules, the rule consequents are not involved in the chromosome encoding. Instead we use the RLS method to calculate the rule consequents. This approach has a limitation in that it is only suitable for the first-order TS fuzzy modeling. However, a clear advantage of doing so is that it can save the searching time and fully exploit the sampling data. The flowchart of the proposed approach is shown in Fig. 4.

4.1. Individual expression based on the hierarchical chromosome structure

Inspired by the insight of biological DNA structure, a hierarchical chromosome formulation for GA is introduced in [20,31]. The chromosome consists of the control genes and the parameter genes. The activation of the parameter genes is governed by the value of the control genes. When a control gene is "1", then the corresponding parametric gene is activated. Otherwise, it is deactivated. The hierarchical architecture implies that the chromosome contains more information than that of the conventional GA structure. Hence, it is called hierarchical genetic algorithm (HGA). Fig. 5 illustrates the concept further.

Since the HGA is well suitable for solving the topological structure of an unknown system, it is a good candidate for determining the fuzzy membership functions and rules. In our proposed approach, we use the two-level hierarchical structure with restrictions on the control genes and the parameter genes.

4.1.1. Control genes and parameter genes

Given $c\_\max$, $c\_\min$, and the number of control genes are denoted by *num_gene*, it should satisfy the following conditions:

$$c\_\min \leqslant num\_gene \leqslant c\_\max; \quad c\_\min, c\_\max \in N \text{ and } c\_\min > 0. \tag{25}$$

In our approach, we set $c\_\max$ equal to the total number of fuzzy sets of the initial model and $c\_\min$ equal to the number of input variables, because at least one fuzzy set should remain for each input variable.

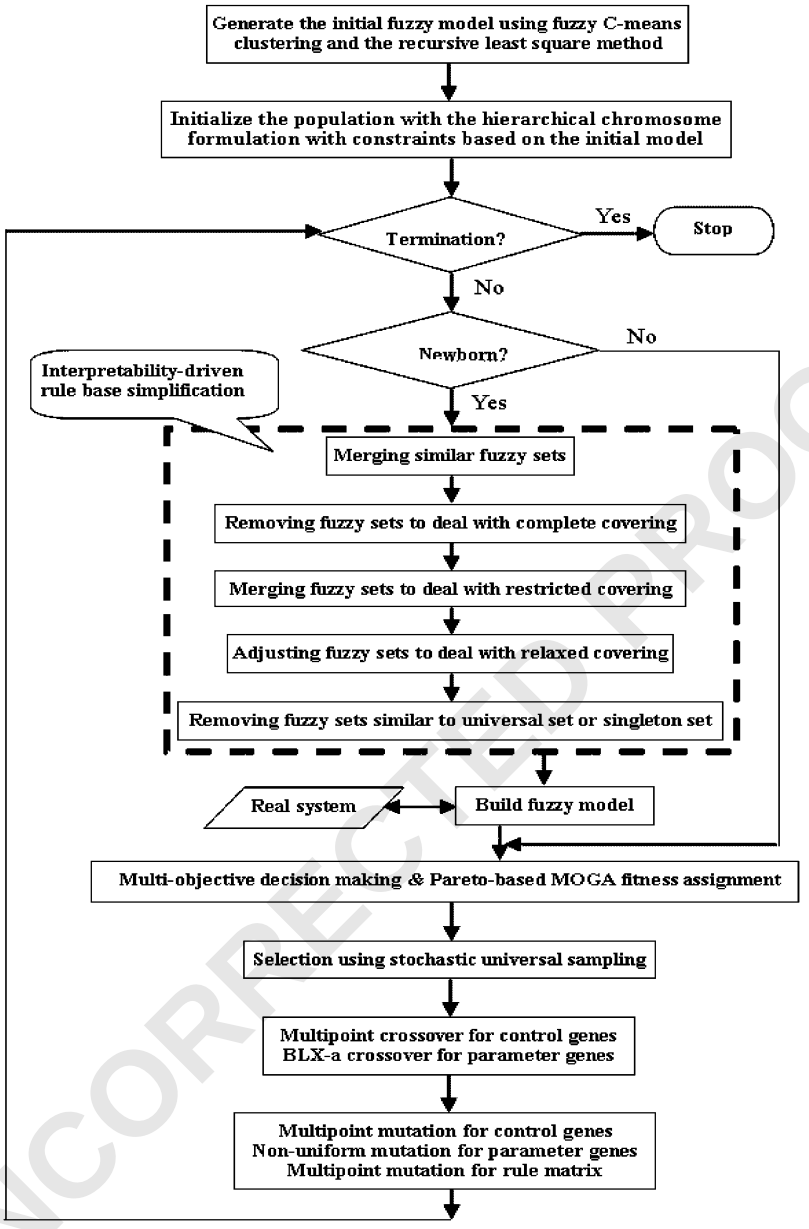


Fig. 4. Flow chart of the proposed approach.

Otherwise, the input variable can be eliminated from the fuzzy system. So strictly speaking, at least one control gene with value "1" should exist in each *control domain* of the input variables. The concept of *control domain* is illustrated in Fig. 6.

We apply the Gaussian combinational membership functions (abbreviated as Gauss2mf) to depict the antecedent fuzzy sets, i.e., a combination of two Gaussian functions. The Gaussian function depends on

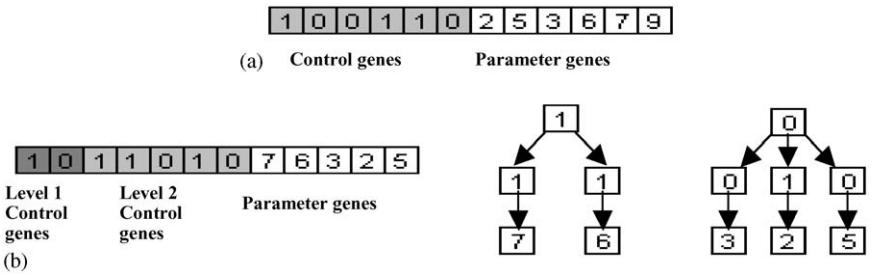


Fig. 5. Example of hierarchical chromosome representation: (a) two level gene structure with phenotype $X_A = (2, 6, 7)$, (b) three level gene structure with phenotype $X_B = (7, 6)$.

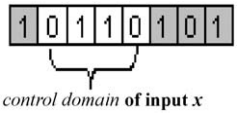


Fig. 6. Considering a three-input fuzzy system with 1, 4 and 3 fuzzy sets, respectively. So the *control domain* of input x has the control genes {1}, the control domain of input x has the control genes {0 1 1 0}, and {1 0 1} for input x.

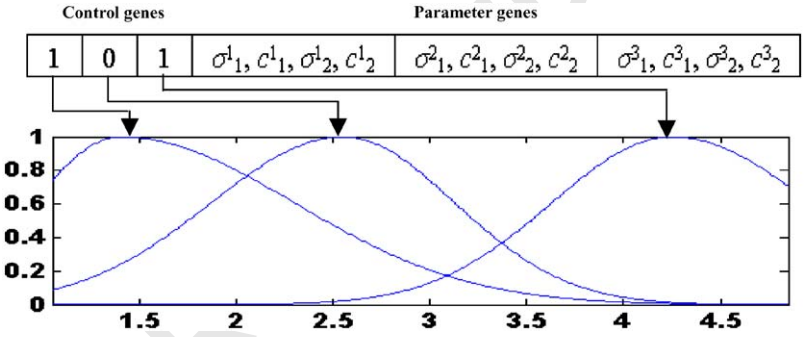


Fig. 7. An example of Hierarchical formulation in our approach.

the two parameters, σ and c, is given by the following:

$$u(x, \sigma, c) = \exp\left\{\frac{-(x-c)^2}{2\sigma^2}\right\}. \tag{26}$$

So we use the parameter list $[\sigma_1, c_1, \sigma_2, c_2]$ to represent one parameter gene, where σ_1 and c_1 determine the shape of the leftmost curve. The shape of the rightmost curve is specified by σ_2 and c_2. The Gauss2mf is a kind of smooth membership functions, so the resulting model will in general have a high accuracy in fitting the training data. Another characteristic of Gauss2mf is that the completeness of fuzzy system is guaranteed because the it covers the universe sufficiently. An example of the relationship between control genes and parameter genes is given in Fig. 7.

4.1.2. Rule structure

The rule structure coding is important because the size of a fuzzy system is fully specified by the rule structure. When the hierarchical chromosome is used, the number of fuzzy sets may vary from one chromosome to another. So it becomes impossible to set a universal rule table for all chromosomes as the traditional approach. Unlike the work proposed in [20,31] in which only one rule chromosome is associated with the subgroup specified by the type of control genes, we instead adopt the strategy that a rule structure is embedded in the chromosome. Suppose that there are n input variables and each input variable x_i has a maximum number of fuzzy sets M_i, then the rule base has at most $R = M_1 \times M_2 \times \cdots \times M_n$ fuzzy rules. We use a multi-dimensional matrix RM called *rule matrix* to describe the rule structure. The RM has n dimensions with size $(M_1 + 1) \times (M_2 + 1) \times \cdots \times (M_n + 1)$, where the cell value is "1" or "0". The cell with value "1" in RM means that the corresponding fuzzy sets form the antecedents of a rule, whereas the one with value "0" does not form a rule. If $RM(\ldots, M_i + 1, \ldots) = 1$, then it indicates that the variable x_i does not appear in the resulting rule (called incomplete rule in [10]). Note that the cell $RM(M_1 + 1, \ldots, M_i + 1, \ldots, M_n + 1)$ should be equal to zero, otherwise, it is meaningless. In order to guarantee the sufficient utility of fuzzy systems, at least one cell in RM corresponding to that active control gene should have the cell value "1". In the following, we will give a concrete example to show the ideas about the rule structure.

Example. Considering an individual chromosome of three input variables (x_1, x_2 and x_3) with equally three fuzzy sets for each of them (A_{ij} denotes the jth fuzzy set of the ith input variable), the control genes are given as {1 0 1 1 1 1 0 1 0}. So the rule matrix RM is a three-dimensional matrix with size $4 \times 4 \times 4$ and the control domain for input x_1, x_2 and x_3 is {1 0 1}, {1 1 1} and {0 1 0}, respectively. Giving the cell values of RM as: $RM(1, 1, 2) = 1$, $RM(3, 2, 4) = 1$, $RM(2, 2, 2) = 1$, and $RM(4, 3, 4) = 1$; the other cell values are equal to 0. We can extract three rules based on the values of RM, the rule antecedents are: (1) if x_1 is A_{11} and x_2 is A_{21} and x_3 is A_{32}; (2) if x_1 is A_{13} and x_2 is A_{22}; (3) if x_2 is A_{23}. Rules 2 and 3 are incomplete rules because the antecedent is defined by a subset of the available variables only. Noted that the cell $RM(2, 2, 2)$ does not play a role in rule generation even if $RM(2, 2, 2) = 1$ because the 2nd control gene of x_1 is not active. The utility of this fuzzy system represented by this chromosome is sufficient because all of the active fuzzy sets are utilized by rules.

We use the RLS method to calculate rule consequents in our work, thus the coding of consequent structure is unnecessary.

4.2. Interpretability-driven rule base simplification

In GA, the newborn individuals often introduce redundancy to the rule base leading to bad interpretability of fuzzy systems. Although some individuals representing interpretable rule base may emerge after a certain number of iterations, the searching efficiency of GA is sometimes unsatisfactory. To address this problem, an interpretability-driven rule base simplification method is applied to the newborn individuals in each generation in order to actively reduce the redundancy of the rule base. As we have discussed in the previous section, the completeness and the sufficient utility of fuzzy systems are guaranteed by means of the hierarchical chromosome expression. The simplification method is mainly based on the similarity measure and the judgment of the type of covering. In this method, we define a fuzzy set A that uses the membership function $u_A(x; a_1, a_2, a_3, a_4)$, where a_1, a_2, a_3, and a_4 are the lower bound, left

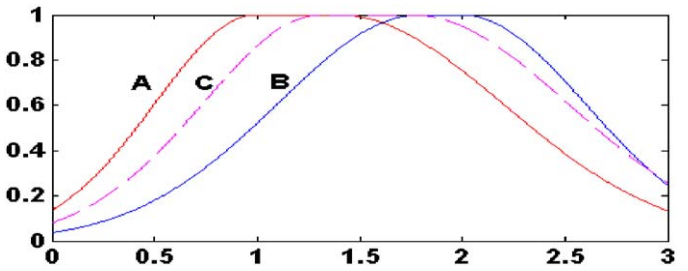


Fig. 8. Merging A and B to create C.

center, right center and upper bound of the definition domain, respectively ($a_1 \leqslant a_2 \leqslant a_3 \leqslant a_4$). In our proposed approach, we use the Gauss2mf as the membership function. So this is not easy to obtain a_1 and a_4 as the triangular or the trapezoidal ones. We need to calculate a_1 and a_4 using a very small number ε (for example 0.001) which is very close to zero, i.e. $u_A(a_1; a_1, a_2, a_3, a_4) = u_A(a_4; a_1, a_2, a_3, a_4) = \varepsilon$. Nevertheless, the interpretability-driven simplification method is also applicable to all other types of membership functions besides Gauss2mf. Because the hierarchical chromosome expression is used, the interpretability-driven simplification method is implemented to those active fuzzy sets when the control genes' values are equal to one.

4.2.1. Merging similar fuzzy sets

An example of the similarity measure between two fuzzy sets is given as in Eq. (8). If the similarity value is greater than a given threshold, then we merge these two fuzzy sets to generate a new one. Considering two fuzzy sets A and B with the membership functions $u_A(x; a_1, a_2, a_3, a_4)$ and $u_B(x; b_1, b_2, b_3, b_4)$, the resulting fuzzy set C with the membership function $u_C(x; c_1, c_2, c_3, c_4)$ is defined from merging A and B by:

$$\begin{aligned} c_1 &= \min(a_1, b_1), \\ c_2 &= \lambda_2 a_2 + (1 - \lambda_2) b_2, \\ c_3 &= \lambda_3 a_3 + (1 - \lambda_3) b_3, \\ c_4 &= \max(a_4, b_4). \end{aligned} \tag{27}$$

The parameters $\lambda_2, \lambda_3 \in [0, 1]$ determines which of the fuzzy sets A and B has the most influence on C. The threshold for merging similar fuzzy sets plays an important role in the improvement of interpretability. According to our experience, values in the range [0.4, 0.7] may be a good choice. In our approach, we set the threshold to 0.45. Fig. 8 illustrates the case for merging A and B to create C.

After merging the similar fuzzy sets, the control genes and the rule matrix should be adjusted accordingly. If C replaces A and B in the rule antecedents, then the control gene associated with B is set to zero.

4.2.2. Removing fuzzy sets similar to the universal set or similar to a singleton set

If the similarity value of a fuzzy set to the universal set $U(u_U(x) = 1,\ \forall x \in X)$ is greater than a upper threshold (θ_U) or smaller than a lower threshold (θ_S), and if this fuzzy set is not the only one fuzzy set of its input variable, then we can remove it from the rule base. In the first case, the fuzzy set is very similar to

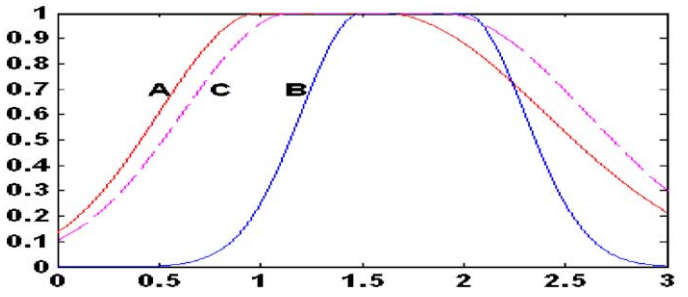


Fig. 9. Merging A and B to create C based on the covering judgment.

the universal set. In fact, the latter case is similar to a singleton set. Neither of these cases is desirable for interpretable rule base generation. However, if the fuzzy set is the only one of its input variable, it should be kept in the rule base because for each given input variable at least one fuzzy set should be defined, otherwise this input variable should not be introduced into the fuzzy system. We set θ_U to 0.8 and θ_S to 0.05 in this work. If a fuzzy set is removed, then the corresponding control gene will update its value from 1 to 0 and the rule antecedents associated with this fuzzy set is removed from the corresponding rules. This is done by adjusting the cell value in the rule matrix.

4.2.3. Removing fuzzy sets to deal with complete covering

As we have discussed in Section 2.1, using the similarity measure does not necessarily guarantee a sound evaluation of the distinguishability of fuzzy systems due to *covering*. If fuzzy set A completely covers fuzzy set B, then we should remove B from the rule base in order to maintain the distinguishable distribution of the fuzzy sets. Then the control gene associated with set B is set to zero and B is replaced by A in the corresponding rule antecedents.

4.2.4. Merging fuzzy sets to deal with restricted covering

If fuzzy set A restrictedly covers set B, then we use the merging similar fuzzy sets method to create a set C. Fig. 9 illustrates the case for merging A and B to create C.

4.2.5. Adjusting fuzzy sets to deal with relaxed covering

If fuzzy set A relaxedly covers set B, then we adjust the membership function parameters according to the relative position between A and B. If set A is in the left of set B, i.e., $a_3 < b_2$, the newborn set A^* and B^* are described as:

Case 1: If $a_3 < b_2$, Then

$$a_1^* = a_1, \quad a_2^* = a_2, \quad a_3^* = a_3, \quad a_4^* = b_4,$$

$$b_1^* = b_1, \quad b_2^* = b_2, \quad b_3^* = b_3, \quad b_4^* = a_4. \tag{28}$$

Otherwise set A is in the right of set B, i.e., $b_3 < a_2$,

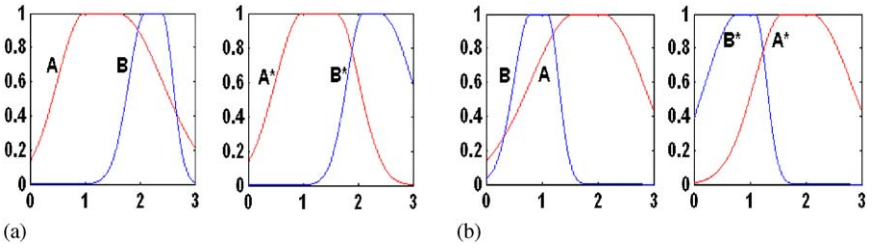


Fig. 10. Set A relaxedly covers set B: (a) Case 1: A in the left of B (b) Case 2: A in the right of B.

Case 2: If $b_3 < a_2$, *then*

$$a_1^* = b_1, \quad a_2^* = a_2, \quad a_3^* = a_3, \quad a_4^* = a_4,$$
$$b_1^* = a_1, \quad b_2^* = b_2, \quad b_3^* = b_3, \quad b_4^* = b_4. \tag{29}$$

Fig. 10 shows both of these two cases.

4.3. Multi-objective hierarchical genetic algorithm

The interpretability and performance of fuzzy systems are greatly dependent on the learning algorithm and it is usually impossible to achieve both of aims at the same time. There is often a trade-off between the interpretability and the accuracy of fuzzy models. Thus, it is a kind of multi-objective optimization problems by nature. In other words, we can get only a set of Pareto-optimal solutions of which the improvement in one of the objectives will degrade other objectives. There are a lot of evolutionary multi-objective optimization approaches to solve this problem [5,6]. Among these approaches, we use the Pareto-based multi-objective genetic algorithm (MOGA) introduced by Fonseca and Fleming in 1993 [9]. Because we use the hierarchical chromosome structure and a number of corresponding operators, we call our approach as multi-objective hierarchical genetic algorithm (MOHGA).

4.3.1. Initial population

As previously mentioned, an initial fuzzy model is generated using the fuzzy clustering method, and this is directly copied to the first generation. The remaining individuals are initialized in the following way. Suppose that there are n input variables and each input variable x_i has a maximum number of fuzzy sets M_i: consider that the initial individual s_1: $s_1.c\_gene$ represents the control genes with length *clength* equal to the total number of fuzzy sets introduced by the fuzzy clustering method, i.e., $M_1+M_2+\cdots+M_n$; $s_1.p\_gene(i)$ represents the ith parameter gene corresponding to the ith control gene $s_1.c\_gene(i)$ in the form of $[\sigma_{i1}^1, c_{i1}^1, \sigma_{i2}^1, c_{i2}^1]$ representing a Gauss2mf; $s_1.RM$ is the rule matrix with size $(M_1+1)\times(M_2+1)\times\cdots\times(M_n+1)$. Then the jth individual $s_j.c\_gene$ is generated randomly with the constraint that at least one control gene of value "1" exists in each *control domain* of the input variables (*CONSTRAINT* 1) and the length of $s_j.c\_gene$ is equal to *clength*. The rule matrix $s_j.RM$ which is the same size as $s_1.RM$ is initialized randomly with the constraint that at least one cell in $s_j.RM$ associated with the active control genes should have the cell value "1" (*CONSTRAINT* 2). This is to guarantee the sufficient utility of the fuzzy systems. The parameter gene $s_j.p\_gene(i)$ which represents a fuzzy set of the input variable x_k in

the form of $[\sigma_{i1}^j, c_{i1}^j, \sigma_{i2}^j, c_{i2}^j]$ is initialized with random values within certain permissible ranges, that is as follows:

$$\sigma_{i1}^j = \sigma_{i1}^1 \, rand_1 \frac{M_k}{2}, \quad \sigma_{i2}^j = \sigma_{i2}^1 \, rand_2 \frac{M_k}{2} \tag{30}$$

and

$$c_{i1}^j = \min\{U(x_k)\, rand_3 + \min(x_k), U(x_k)\, rand_4 + \min(x_k)\},$$

$$c_{i2}^j = \max\{U(x_k)\, rand_3 + \min(x_k), U(x_k)\, rand_4 + \min(x_k)\}, \tag{31}$$

where $rand_1$ and $rand_2$ are random numbers in (0, 1], $rand_3$ and $rand_4$ are random numbers in [0, 1], $U(x_k)$ is the universe of the input variable x_k which has M_k fuzzy sets. By doing so, the left center c_{i1}^j is smaller than or equal to the right center c_{i2}^j, and the left width σ_{i1}^j and the right width σ_{i2}^j are larger zero. It satisfies the topological conditions of Gauss2mf (*CONSTRAINT* 3).

4.3.2. Crossover operators

Considering there are two types of genes in the chromosome, the crossover operation is applied to control genes and parameter genes separately. For control genes, multi-point crossover is applied considering the reason of simplicity. Then the rule matrix exchanges the corresponding cell values. Note that the crossover operation may violate *CONSTRAINT* 1 and *CONSTRAINT* 2. If a newborn individual does not satisfy these two constraints, then we adjust the control genes and the rule matrix randomly to obtain qualified candidates. As for the parameter genes which are represented in real numbers, BLX-α crossover [8] is applied because BLX-α (in particularly $\alpha = 0.5$) crossover has turned out to be the best crossover operators for the real-coded GA based on the experiment results reported in [12]. In our approach, we apply BLX-α crossover with *CONSTRAINT* 3.

4.3.3. Mutation operators

The mutation is applied to control genes, parameter genes and the rule matrix, separately. Multi-point mutation is applied to the control genes with *CONSTRAINT* 1 for its simplicity, then we will check if the rule matrix violates *CONSTRAINT* 2 or not. If *CONSTRAINT* 2 is not satisfied, we will adjust the rule matrix accordingly. Regarding the parameter genes, Non-uniform mutation is applied [21] with *CONSTRAINT* 3, because it has been demonstrated that Non-uniform mutation is very appropriate for the real-coded GA [12]. In addition, multi-point bit mutation is used for the rule matrix with *CONSTRAINT* 2. The selected cell of the rule matrix for mutation is flipped ("1" or "0") if a probability test is satisfied (a random number is smaller than a predefined rate). In the current work, the hierarchical genetic algorithm is used. As far as the parameter genes are concerned, the typical BLX-α crossover operator and non-uniform mutation operator are implemented because of their advantages stated in [12]. We also notice the mutation operators applied in the evolutionary strategies, especially the self-adaptive parameter control mechanisms [11,27]. The distinguishing feature of self-adaptation mechanism is that the control parameters (different from the objective parameters that define points in search space) are evolved by the evolutionary algorithms, rather than exogenously defined or modified according to some fixed schedule [3]. The self-adaptation mechanism uses evolutionary learning principles on two levels at the same time: the level of solutions and the level of the search strategy, and works well. We are inspired by such

beautiful ideas and the mutation operators with the self-adaptive parameter control mechanism would be incorporated and explored into our work in the future research.

4.3.4. Multi-objective decision making and Pareto-based fitness assignment

In our approach, we consider both the accuracy and the interpretability of fuzzy systems. The accuracy is measured in terms of mean squared error (MSE). Whilst the concept of interpretability is translated into the completeness and the distinguishability of fuzzy sets expressed through fuzzy similarity measure, the non-redundancy of fuzzy rules by means of non-redundancy measure, and the compactness of fuzzy systems expressed in terms of the number of rules and the number of fuzzy sets. Because we have already guaranteed the sufficient utility of fuzzy systems through the chromosome formulation and genetic operators with constraints, we do not incorporate the utility item into the interpretability considerations for fitness evaluation.

(a) *The objective for accuracy*: The MSE is described as follows:

$$fit_{\text{acc}} = \frac{1}{N}\sum_{i=1}^{N}[(y_1^i - \hat{y}_1^i)^2 + (y_2^i - \hat{y}_2^i)^2 + \cdots + (y_m^i - \hat{y}_m^i)^2], \tag{32}$$

where $\mathbf{Y} = [y_1, y_2, \ldots, y_m]^{\mathrm{T}}$ is the true output vector, $\hat{\mathrm{Y}} = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m]^{\mathrm{T}}$ is the model output vector, N is the number of sampling data pairs.

(b) *The objective for completeness and distinguishability*: The similarity measure is given in Eq. (8). We consider two cases about the number of fuzzy sets of input variables: one case is that a variable has only one fuzzy set, and the other is that a variable has more than one fuzzy set. In the former case, we denote: $\hat{S} = \{x_i \mid M_i^{\mathrm{a}} = 1\}$ where M_i^{a} is the number of active fuzzy sets of variable x_i. In the latter case, $\check{S} = \{x_i \mid M_i^{\mathrm{a}} > 1\}$. For each input variable $x_i \in \hat{S}$, A_i is the only fuzzy set of variable x_i and U_i is the universal set of x_i. Two parameters θ_l and θ_{u} are the desired lower and upper bounds of the similarity measure between the fuzzy set and the universal set. We define the following:

$$\text{If } \theta_l \leqslant S(A_i, U_i) \leqslant \theta_{\mathrm{u}}, \quad \widehat{\beta}_i = 1; \quad \text{otherwise } \widehat{\beta}_i = 0; \quad \forall x_i \in \hat{S}, \tag{33}$$

$$\widehat{\beta} = \sum_{i=1}^{n} \widehat{\beta}_i, \quad i = \{j \mid x_j \in \hat{S}\}, \tag{34}$$

where n is the size of $\hat{S}$. In our approach, we set $\theta_l = 0.4$ and $\theta_{\mathrm{u}} = 0.8$. On the other hand, for each input variable $x_i \in \check{S}$, A_k^i and A_{k+1}^i are two neighboring fuzzy sets of variable x_i. Two parameters θ_{low} and θ_{up} are the lower and upper bounds of the similarity measure between fuzzy sets and they are defined as follows:

$$\text{If } \theta_{\mathrm{low}} \leqslant S(A_k^i, A_{k+1}^i) \leqslant \theta_{\mathrm{up}}, \check{\beta}_k^i = 1; \quad \text{otherwise } \check{\beta}_k^i = 0, \;\; k = 1, \ldots, M_i^{\mathrm{a}} - 1; \;\; \forall x_i \in \check{S}. \tag{35}$$

$$\check{\beta} = \sum_{l=1}^{m}\sum_{k=1}^{M_l^{\mathrm{a}}-1} \check{\beta}_k^l, \quad l = \{j \mid x_j \in \check{S}\}, \tag{36}$$

where m is the size of $\check{S}$. In our approach, we set $\theta_{\text{low}} = 0$ and $\theta_{\text{up}} = 0.4$. Then the fitness function for completeness and distinguishability evaluation is shown as below:

$$fit_{\text{sim}} = \frac{\widehat{\beta} + \breve{\beta}}{n + \sum_{l=1}^{m}(M_l^{\text{a}} - 1)}. \tag{37}$$

The fitness item fit_{sim} ranges from 0 to 1.

(c) *The objective for non-redundancy*: In our approach, the non-redundancy value of rule R_i and R_j is given as in Eq. (15). Based on this definition, the non-redundancy of fuzzy rule base is evaluated by the following equation:

$$fit_{\text{NRdd}} = \frac{\sum_{i=1}^{R-1}\sum_{j=i+1}^{R} NRdd(R_i, R_j)}{(R-1)R/2} \quad \text{if } R > 1,\ fit_{\text{cons}} = 1 \text{ if } R = 1, \tag{38}$$

where R is the number of fuzzy rules. It is a maximization problem and the values of fit_{NRdd} range from 0 to 1.

(d) *The objective for compactness*: The compactness of fuzzy systems can be expressed in terms of the number of fuzzy sets fit_{nFS} and the number of fuzzy rules fit_{nRule}:

$$fit_{\text{nFS}} = \sum_{i=1}^{n} M_i^{\text{a}} \tag{39}$$

$$fit_{\text{nRule}} = R \tag{40}$$

where n is the number of input variables, M_i^{a} is the number of active fuzzy sets for input variable x_i and R is the number of rules.

Based on the above discussions, the objectives of the multi-criteria in our approach is to find the maxima in Eqs. (37) and (38) and the minima in Eqs. (32), (39) and (40). We will use the MOGA Fitness Assignment Procedure with dynamically updating of the sharing parameter σ_{share} [6] to assign the fitness among the candidates. In fitness assignment, we will determine the dominance relationship between the two selected solutions by comparing these five objectives. This procedure requires $O(MN^2)$ comparisons, where N is the size of population and M is the number of objectives, i.e., $M = 5$ in our approach. For comparison, we pre-define the preference over these five objectives, for example, the first priority for fit_{acc}, the second priority for fit_{nFS} and fit_{nRule}, and the third and the fourth priority for fit_{sim} and fit_{NRdd}, respectively. Thus we do not need to have a prior knowledge about the weights among those objectives, and it is similar to the human decision-making process. The drawback is that the comparison takes more computational time than the weighted method [16,22]. However, different sets of fuzzy rules that emphasize different aspects of interpretability and accuracy may be built based on the preferences. This is a topic worthwhile further study and is not the concern of this paper.

4.3.5. Selection operators

In the MOHGA, we use the stochastic universal sampling (SUS) [2] to select the next population. SUS provides zero bias and minimum spread.

4.4. Recursive least square method to train rule consequents

For each individual, we use its control genes, parameter genes and rule matrix to train the rule consequents using the recursive least square method. Then we will build the corresponding fuzzy model and compute its accuracy in terms of the MSE.

5. Comparative experiment results

We have set up a few experiments to test the effectiveness of our proposed method. The following sub-sections describe the results of our work and it shows that the proposed approach is effective.

5.1. Example: nonlinear plant with two inputs and one output

The second-order nonlinear plant is studied by Wang and Yen in [32–34]; Roubos and Setnes [25,26] and Jiménez et al. [14]:

$$y(k) = g(y(k-1), y(k-2)) + u(k), \tag{41}$$

where

$$g(y(k-1), y(k-2)) = \frac{y(k-1)y(k-2)(y(k-1)-0.5)}{1+y^2(k-1)+y^2(k-2)}. \tag{42}$$

The goal is to approximate the nonlinear component $g(y(k-1), y(k-2))$ of the plant with a fuzzy model. In [14,25,26,32], 400 sampling data points were generated from the plant model. Two hundred samples of training data were obtained with a random input signal $u(k)$ uniformly distributed in the interval $[-1.5\ 1.5]$, while the last 200 validation data points were obtained by using a sinusoid input signal $u(k) = \sin(2\pi k/24)$. The 400 simulated data points are shown in Fig. 11.

We compare our results with those obtained by different approaches in [14,25,26,32–34]. However, we put more focus on [25,26] because the approaches used there are more similar to ours especially in the use of initial rule base construction and the rule base simplification method. These approaches are described below, and the best results are summarized in Table 1.

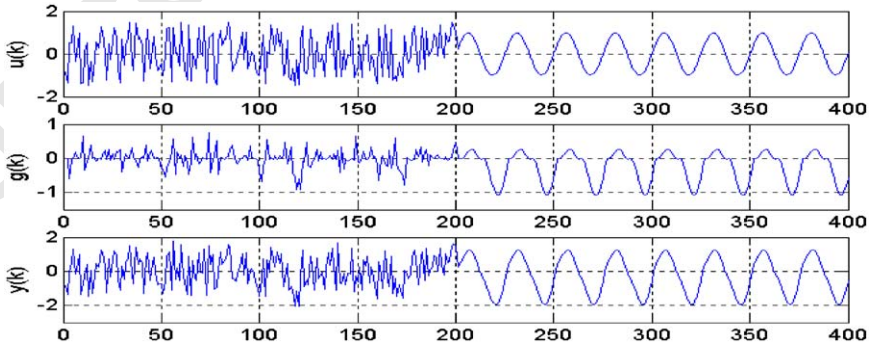


Fig. 11. Input $u(k)$, unforced system $g(k)$, and output $y(k)$ of the plant in (41).

Table 1
Fuzzy models of the nonlinear plant of Section 5.1

Ref.	No. of rules	No. of sets	Consequent	MSE train	MSE validation
[32]	40 rules (initial)	40 Gauss.	Singleton	3.2884e − 4	6.9152e − 4
	28 rules (optimized)	28 Gauss.	Singleton	3.3299e − 4	5.9595e − 4
[34]	25 rules (initial)	25 Gauss.	Singleton	2.3092e − 4	4.0717e − 4
	20 rules (optimized)	20 Gauss.	Singleton	6.8341e − 4	2.3836e − 4
[33]	36 rules (initial)	12 B-splines	Singleton	2.7743e − 5	5.1163e − 3
	23 rules (optimized)	12 B-splines	Singleton	3.1746e − 5	1.4776e − 3
	36 rules (initial)	12 B-splines	Linear	1.9465e − 6	2.9211e − 3
	24 rules (optimized)	12 B-splines	Linear	1.9835e − 6	6.4120e − 4
[25]	7 rules (initial)	14 triangular	Singleton	1.6e − 2	1.2e − 3
	7 rules (optimized)	14 triangular	Singleton	3.0e − 3	4.9e − 4
	5 rules (initial)	10 triangular	Linear	5.8e − 3	2.5e − 3
	5 rules (optimized)	8 triangular	Linear	7.5e − 4	3.5e − 4
	4 rules (optimized)	4 triangular	Linear	1.2e − 3	4.7e − 4
[26]	5 rules (initial)	10 triangular	Linear	4.9e − 3	2.9e − 3
	5 rules (optimized)	10 triangular	Linear	1.4e − 3	5.9e − 4
	5 rules (optimized)	5 triangular	Linear	8.3e − 4	3.5e − 4
[14]	5 rules (optimized)	5 trapezoidal	Linear	2.0e − 3	1.3e − 3
	5 rules (optimized)	6 trapezoidal	Linear	5.9e − 4	8.8e − 4
This paper					
Fig. 12(a)	5 rules (initial)	10 Gauss2mf.	Linear	1.4032e − 3	2.6267e − 3
Fig. 12(b)	5 rules (optimized)	3 Gauss2mf.	Linear	2.3773e − 4	3.0116e − 4
Fig. 12(c)	4 rules (optimized)	3 Gauss2mf.	Linear	5.4611e − 4	5.4360e − 4
Fig. 12(d)	4 rules (optimized)	3 Gauss2mf.	Linear	5.6086e − 4	2.4885e − 4

In [32], the authors proposed an algorithm that combines the advantages of GA's search capability and the Kalman filter's fast convergence. The antecedent fuzzy sets of 40 rules encoded by Gaussian membership functions were determined initially by clustering and kept fixed. A binary GA was used to select a subset of the initial 40 rules in order to produce a more compact rule base. Then the consequents were calculated by the Kalman filter, and the Schwarz–Rissanen criterion (SRC) was used as evaluation function to balance the trade-off between the number of rules and the model accuracy.

In [34], the authors proposed several orthogonal transformation-based methods for rule selection. They used an initial model with 25 rules. Finally, 20 rules remained and five redundant rules were eliminated from the rule base.

In [33], several information theoretic optimality criterions were used to pick up rules from a set of 36 rules in order to obtain a compact and accurate model. The role of these optimality criteria in fuzzy

modeling is discussed and their practical applicability is illustrated using the nonlinear system example in Eq. (41).

In [25], a two-step approach is proposed for data-driven fuzzy modeling. First, the fuzzy clustering method and the BLS method are used to obtain an initial rule-based model with 5 rules and 10 fuzzy sets encoded with the triangular membership functions. Then the initial fuzzy model is optimized by a real-coded GA subjected to certain constraints for maintaining the semantic properties of fuzzy rules.

In [26], an iterative fuzzy identification technique starting at fuzzy clustering with some redundancy of fuzzy rule base is proposed. First, fuzzy clustering method is applied to obtain an initial rule base model from the sampling data. Successively, similarity driven rule base simplification and GA-based optimization are applied in an iterative manner resulting in a compact rule base of low complexity and high accuracy. Finally, a GA-based optimization is performed to increase accuracy and interpretability of the fuzzy rule base.

shows the distribution of the fuzzy sets and the simulation results about the initial model and the optimized models, respectively. We present the antecedent and consequent parameters of the fuzzy rules in Table 2.

In [14] a Pareto-based multi-objective evolutionary algorithm for fuzzy modeling is proposed. This algorithm has a variable-length, real-coded representation. Each individual of the population contains a variable number of rules between 1 and *max*, where *max* is defined by the decision maker.

In our work, we first use the fuzzy C-means clustering method and the RLS method to construct the initial fuzzy model. In order to compare the results with [25,26], we also set the cluster number equal to 5. So the initial rule base is obtained by partitioning each of the two inputs $y(k-1)$ and $y(k-2)$ into five fuzzy sets. Unlike the triangular membership functions used in [25,26], we use the Gaussian combination membership functions. After obtaining an initial fuzzy model, the MOHGA-RLS was applied with a population size: $L = 40$ and the number of generation T=200. Table 1 describes the experimental results when compared with those in [14,25,26,32–34]. Fig. 12

5.2. *Example: nonlinear static system with two inputs and one output*

Let us consider a nonlinear static system with two inputs x_1 and x_2, and a single output y studied by Sugeno and Yasukawa [29] and by Rojas et al. [24]:

$$y = (1 + x_1^{-2} + x_2^{-1.5})^2, \quad 1 \leqslant x_1, x_2 \leqslant 5. \tag{43}$$

In [29], the fuzzy clustering method is used to identify the structure of fuzzy models. Then the parameter identification is applied to obtain the parameters of fuzzy models. In [24], a three-step approach for fuzzy system generation is proposed. Step 1 outlines a very simple initial fuzzy system. A new and more suitable topology is decided for the fuzzy system in Step 2. A best fuzzy system considering both the accuracy and the complexity of the fuzzy rules is selected in Step 3. In order to make a quantitative comparison of the results obtained in [24,29], we used the same 50 data points described in [29]. The initial model was obtained with six clusters, resulting in a model with six rules and twelve fuzzy sets. Table 3 describes the experimental results compared with those in [24,29]. Fig. 13 shows the distribution of the fuzzy sets and the simulation results about the initial model and the optimized models, respectively. We present the antecedent and consequent parameters of the fuzzy rules in Table 4.

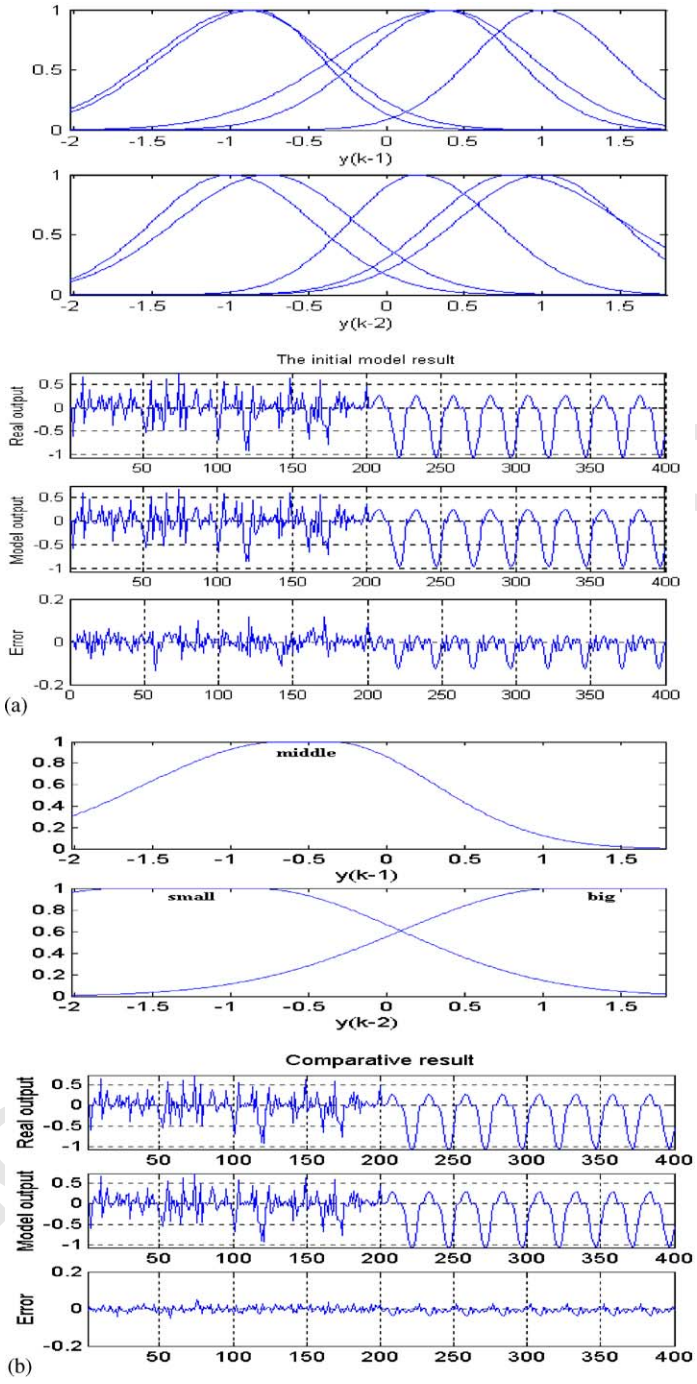
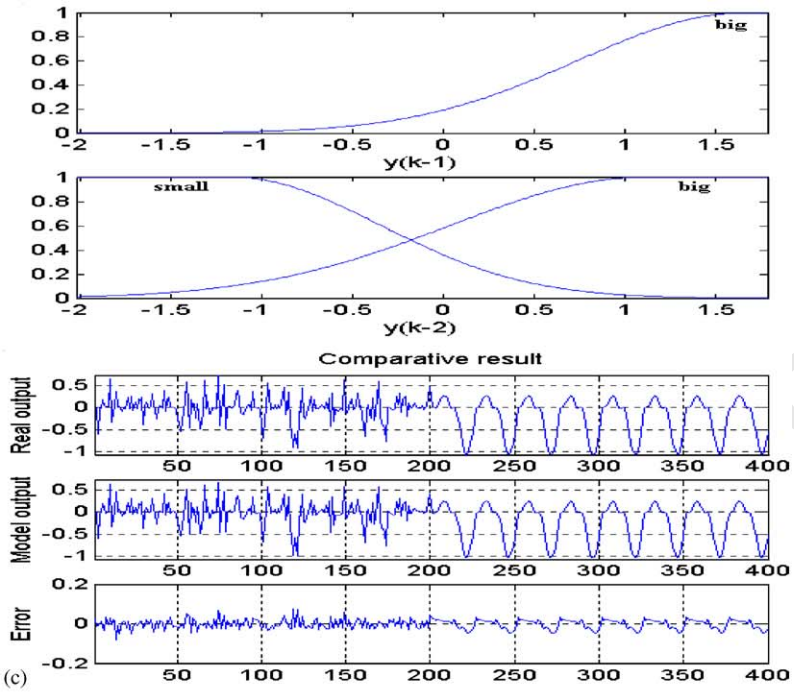
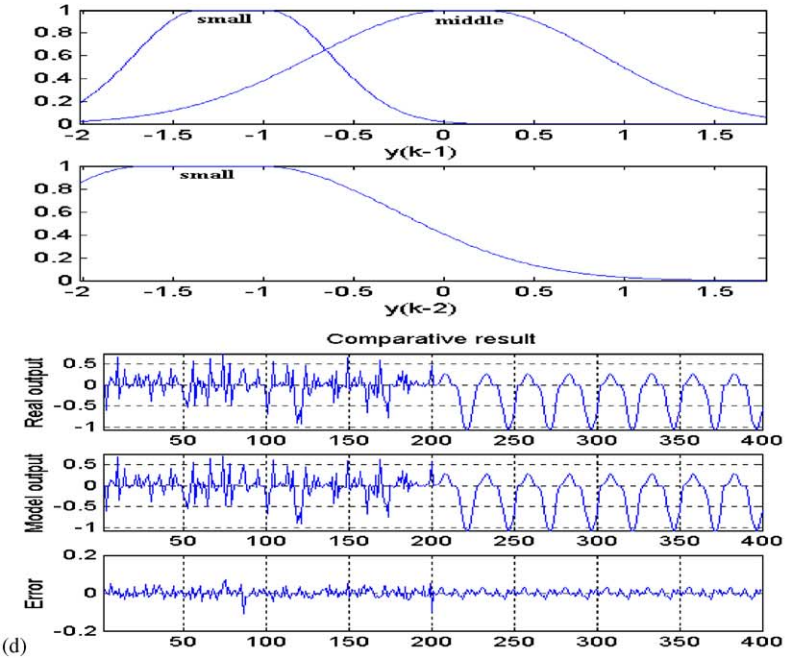


Fig. 12. Fuzzy sets distribution and the simulation results of Section 5.1: (a) initial model with 5 rules and 10 sets, (b) optimized model with 5 rules and 3 sets, (c) and (d) optimized model with 4 rules and 3 sets.



(c)



(d)

Fig. 12. (*Continued.*)

Table 2
Fuzzy model parameters for Fig. 12(b)–(d)

(b) **Rule expression**
R1: If $y(k-1)$ is middle and $y(k-2)$ is small, then $g(k) = 0.9595y(k-1) - 0.1801y(k-2) - 0.8555$
R2: If $y(k-1)$ is middle and $y(k-2)$ is big, then $g(k) = -2.4551y(k-1) - 0.4372y(k-2) - 1.4300$
R3: If $y(k-1)$ is middle, then $g(k) = 0.8033y(k-1) - 0.0883y(k-2) + 1.3722$
R4: If $y(k-2)$ is small, then $g(k) = 0.1030y(k-1) + 0.1406y(k-2) - 0.5434$
R5: If $y(k-2)$ is big, then $g(k) = -0.1503y(k-1) + 0.0591y(k-2) + 0.6716$

Antecedent parameters
$y(k-1)$: middle = [0.8810, −0.6619, 0.6714, −0.3693]
$y(k-2)$: small = [1.2114, −1.6839, 0.9514, −0.8601], big = [1.0104, 1.1024, 1.1600, 2.1794]

(c) **Rule expression**
R1: If $y(k-1)$ is big and $y(k-2)$ is small, then $g(k) = -1.3585y(k-1) + 0.2023y(k-2) + 0.5208$
R2: If $y(k-1)$ is big and $y(k-2)$ is big, then $g(k) = 0.9723y(k-1) + 0.0871y(k-2) - 0.1519$
R3: If $y(k-2)$ is small, then $g(k) = 0.5563y(k-1) - 0.1032y(k-2) - 0.1863$
R4: If $y(k-2)$ is big, then $g(k) = -0.4673y(k-1) + 0.0124y(k-2) + 0.0341$

Antecedent parameters
$y(k-1)$: big = [0.9133, 1.6594, 0.3006, 2.3074]
$y(k-2)$: small = [1.2491, −2.2310, 0.8097, −1.1593], big = [1.0651, 1.1110, 1.1608, 2.1766]

(d) **Rule expression**
R1: If $y(k-1)$ is small and $y(k-2)$ is small, then $g(k) = 1.1417y(k-1) - 0.3714y(k-2) - 0.4983$
R2: If $y(k-1)$ is small, then $g(k) = -0.5174y(k-1) - 0.1560y(k-2) + 0.2069$
R3: If $y(k-1)$ is middle and $y(k-2)$ is small, then $g(k) = -0.0049y(k-1) - 0.4959y(k-2) + 0.1679$
R4: If $y(k-2)$ is small, then $g(k) = 0.1070y(k-1) + 0.4692y(k-2) - 0.1884$

Antecedent parameters
$y(k-1)$: small = [0.3652, −1.3556, 0.3526, −0.9812], middle = [0.7459, 0.0324, 0.6682, 0.2102]
$y(k-2)$: small = [0.6629, −1.6519, 0.7631, −1.0252]

5.3. Example: Lorenz system

The Lorenz system studied in [19] is described by the following differential equations:

$$\dot{x} = -y^2 - z^2 - a(x - F), \tag{44}$$

$$\dot{y} = xy - bxz - y + G, \tag{45}$$

$$\dot{z} = bxy + xz - z. \tag{46}$$

In order to make a comparison with the results obtained in [19], we use the same means to generate the sampling data. That is to say, $a = 0.25$, $b = 4.0$, $F = 8.0$ and $G = 1.0$. In the simulation, we predict $x(t)$ from $x(t-1)$, $y(t-1)$ and $z(t-1)$. Two thousand data points are obtained from the Eqs. (44)–(46) using the fourth order Runge–Kutta method with a step length of 0.05, where 1000 pairs of data are used for training and the other 1000 for test. The sampling data pairs are shown in Fig. 14.

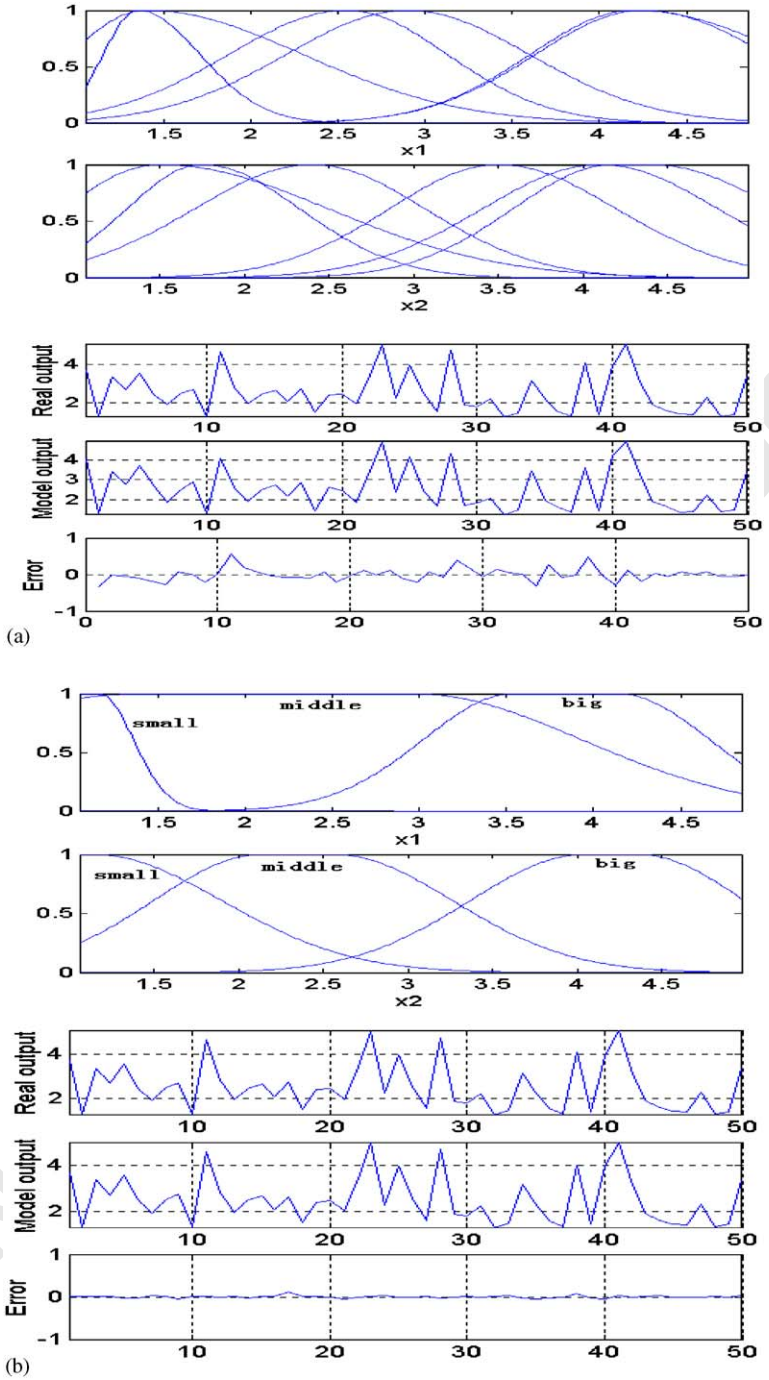
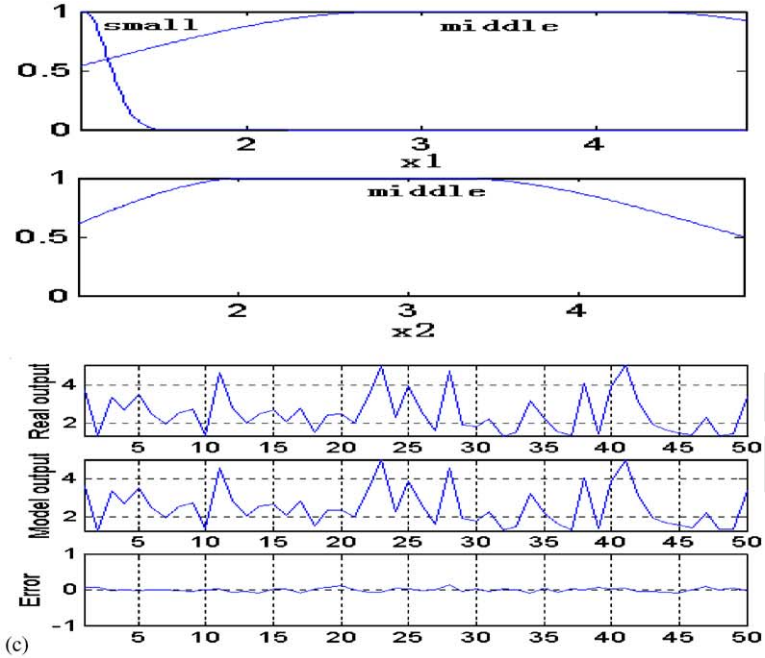
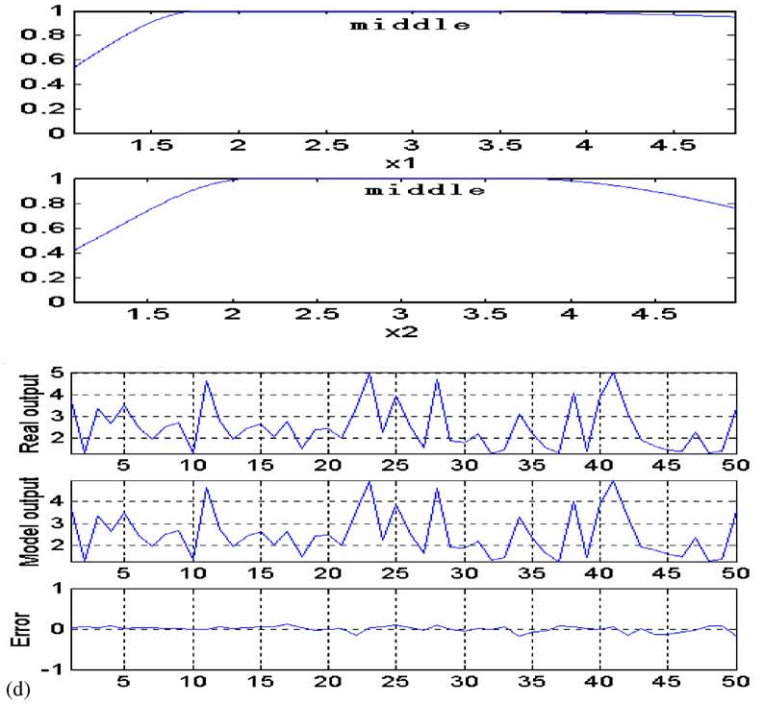


Fig. 13. Fuzzy sets distribution and the simulation results of Section 5.2: (a) initial model with 6 rules and 12 sets, (b) optimized model with 7 rules and 6 sets, (c) optimized model with 4 rules and 3 sets, (d) optimized model with 3 rules and 2 sets.



(c)



(d)

Fig. 13. (*Continued.*)

Table 3
Fuzzy models of the nonlinear plant of Section 5.2

Ref.	No. of rules	No. of Fuzzy sets	MSE train
[29][a]			
Before parameter identification	6 rules	18 trapezoid	0.318
After parameter identification	6 rules	18 trapezoid	0.079
Using position gradient model	6 rules	12 trapezoid	0.010
[24]			
Configuration 1	9 rules	6 triangular	0.263
Configuration 2	16 rules	8 triangular	0.126
Configuration 3	25 rules	10 triangular	0.0523
This paper			
Fig. 13(a)	6 rules (initial)	12 Gauss2mf.	3.0852e − 2
Fig. 13(b)	7 rules (optimized)	6 Gauss2mf.	8.9507e − 4
Fig. 13(c)	4 rules (optimized)	3 Gauss2mf.	2.7389e − 3
Fig. 13(d)	3 rules (optimized)	2 Gauss2mf.	5.1733e − 3

[a]Note that the 42nd data point in Table 2 of [29] should be corrected.
The value of "y" should not be "1.97" but "3.11".

In [19] an initial fuzzy system is generated using the evolutionary algorithm based method. Then this fuzzy system is converted to an RBF neural network and continued to be trained with the conventional gradient method. After the training algorithm converges, the adaptive weight sharing algorithm for extracting fuzzy rules is implemented.

In our approach, an initial fuzzy model is generated using the fuzzy C-means clustering method and the RLS method with 5 rules and 15 fuzzy sets (equally 5 sets for each of the three input variables $x(t-1)$, $y(t-1)$, and $z(t-1)$). Then we use the interpretability-driven rule base simplification method, multi-objective hierarchical genetic algorithm based on the interpretability and accuracy objectives to construct fuzzy models. Table 5 describes the experimental results compared with those in [19]. Table 6 presents the antecedent and consequent parameters of the fuzzy rules obtained by our approach. The distribution of the fuzzy sets and the simulation results are shown in Fig. 15.

5.4. Example: Mackey–Glass time series

The Mackey–Glass time series studied in [18] is described as follows:

$$\dot{x} = \frac{ax(t-r)}{1+x^b(t-r)} - cx(t). \tag{47}$$

We set the same value to the parameters in order to make a comparison with the results obtained in [18]. That is to say, $a = 0.2$, $b = 10$, $c = 0.1$ and $r = 30$. The goal is to predict $x(t)$ from $x(t-1)$, $x(t-2)$ and $x(t-3)$. 1000 data points are obtained from Eq. (47) using the fourth order Runge–Kutta method with a step length of 1 and the initial condition $x(0) = 1.2$, where 500 pairs of data are used for training and the other 500 for test. The sampling data pairs are shown in Fig. 16.

Table 4
Fuzzy model parameters for Fig. 13(b)–(d)

(b) **Rule expression**
R_1: If x_1 is middle and x_2 is middle, then $y = -0.0746x_1 - 1.5580x_2 + 7.8487$
R_2: If x_1 is middle and x_2 is big, then $y = -0.4408x_1 + 0.0018x_2 + 2.7997$
R_3: If x_1 is middle and x_2 is small, then $y = -1.9667x_1 - 5.7073x_2 + 14.2118$
R_4: If x_1 is small and x_2 is middle, then $y = -7.7450x_1 - 1.0939x_2 + 18.0300$
R_5: If x_1 is small and x_2 is big, then $y = -3.5432x_1 + 2.0367x_2 - 0.0838$
R_6: If x_1 is big and x_2 is big, then $y = 0.0284x_1 - 0.1485x_2 + 2.0844$
R_7: If x_1 is big and x_2 is small, then $y = -0.7829x_1 - 2.2814x_2 + 10.9001$

Antecedent parameters
x_1: small = [0.1972, 1.0500, 0.1972, 1.1612], middle = [1.4037, 1.4443, 0.9690, 2.9655]
big = [0.5281, 3.5462, 0.5281, 4.1387]
x_2: small = [0.1697, 1.0380, 0.7601, 1.1421], middle = [0.6491, 2.1408, 0.7046, 2.5604]
big = [0.6846, 4.0472, 0.6323, 4.3567]

(c) **Rule expression**
R_1: If x_1 is middle and x_2 is middle, then $y = 0.3725x_1 - 2.5924x_2 - 19.2852$
R_2: If x_1 is middle, then $y = -0.5159x_1 - 2.2213x_2 + 15.8310$
R_3: If x_1 is small and x_2 is middle, then $y = 1.9520x_1 - 0.6571x_2 + 6.1666$
R_4: If x_2 is middle, then $y = -0.2506x_1 - 1.4928x_2 + 13.1501$

Antecedent parameters
x_1: small = [0.4093, 0.9737, 0.1357, 1.0719], middle = [1.6026, 2.8305, 1.6420, 4.1938]
x_2: middle = [1.0368, 2.0892, 1.5006, 3.2139]

(d) **Rule expression**
R_1: If x_1 is middle and x_2 is middle, then $y = -7.1554x_1 + 1.0645x_2 - 7.6321$
R_2: If x_1 is middle, then $y = -1.0802x_1 - 0.5307x_2 + 11.4291$
R_3: If x_2 is middle, then $y = 7.1143x_1 - 1.4932x_2 + 7.8529$

Antecedent parameters
x_1: middle = [0.6706, 1.7988, 6.5565, 2.7640]
x_2: middle = [0.8107, 2.1321, 1.7888, 3.6582]

In [18], the evolution strategies are used to generate an initial fuzzy model using a scalar function. Then this fuzzy model is converted to an RBF network to refine the obtained knowledge. Finally, the adaptive weight sharing regularization technique is applied to extract interpretable fuzzy rules. In our approach, an initial fuzzy model is generated with 5 rules and 15 fuzzy sets (equally 5 sets for each of the three input variables $x(t-1)$, $x(t-2)$ and $x(t-3)$). Then the multi-objective hierarchical genetic algorithm is implemented to extract interpretable fuzzy systems with the interpretability-driven rule base simplification method applied to the newborn individuals. Table 7 describes the experimental results compared with those in [18]. Fig. 17 shows the distribution of the fuzzy sets and the simulation results about the initial model and the optimized model, respectively. We give the antecedents and consequents of the fuzzy rules in Table 8.

Remarks and analysis: Simulation results show that a better interpretable fuzzy system with a high accuracy can be obtained by using our proposed approach.

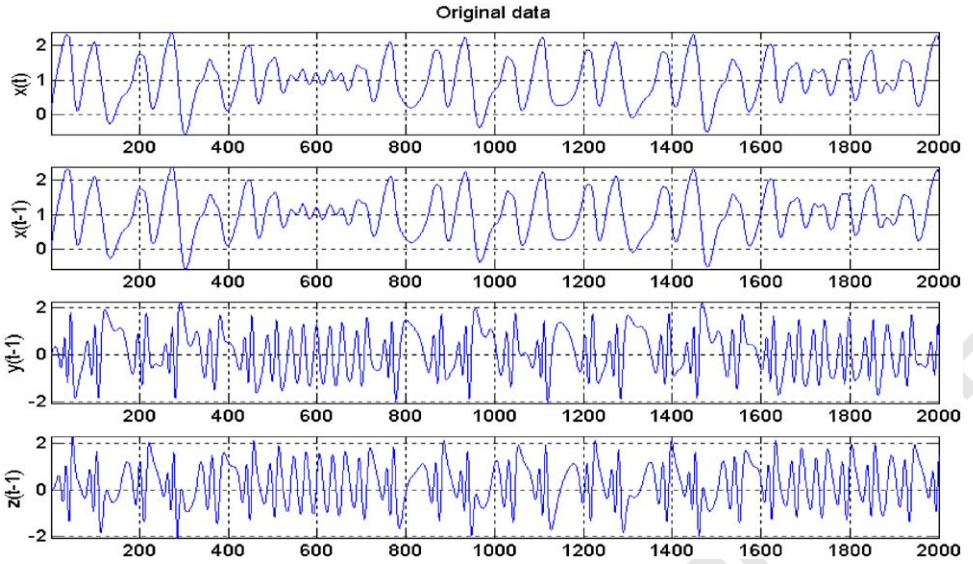


Fig. 14. Input $x(t-1)$, $y(t-1)$ and $z(t-1)$, output $x(t)$ of the Lorenz system.

Table 5
Fuzzy models of the nonlinear plant of Section 5.3

Ref.	No. of rules	No. of fuzzy sets	MSE train	MSE validation
[19]	4 rules	7 Gauss.	Not mentioned	Not mentioned
This paper				
Fig.15(a)	5 rules (initial)	15 Gauss2mf.	5.0523e − 4	4.6257e − 4
Fig.15(b)	3 rules (optimized)	3 Gauss2mf.	6.8435e − 5	6.5679e − 5
Fig.15(c)	2 rules (optimized)	3 Gauss2mf.	1.1907e − 4	9.5102e − 5
Fig.15(d)	2 rules (optimized)	3 Gauss2mf.	2.6534e − 4	2.5116e − 4

Firstly, we compare our results with those obtained by other approaches reported in the literature mainly in terms of the number of rules, the number of fuzzy sets, the MSE for training and validation (test) data. From Tables 1, 3, 5 and 7, it can be clearly seen that our proposed approach can obtain solutions with better interpretability and a comparable or higher accuracy than those obtained by other approaches reported in the literature.

Secondly, diverse solutions can be found in our approach due to the multi-objective hierarchical genetic algorithm. We only present three typical solutions from the different alternatives that we have obtained for each of the four examples. Diversity means that not only the number of fuzzy rules, the number of fuzzy sets or the MSE is different among the alternatives, but also the distribution of fuzzy sets are different from one solution to another when considering the same or similar factors above. For example, in Fig. 12(c) vs. (d), Fig. 15(c) vs. (d), and Fig. 17(c) vs. (d), the number of rules and the number of fuzzy sets of the two compared fuzzy models are the same, the MSE between them are similar, whereas the fuzzy partitions are quite different.

Table 6
Fuzzy model parameters for Fig. 15(b)–(d)

(b) **Rule expression**
R1: If $x(t-1)$ is middle and $y(t-1)$ is small and $z(t-1)$ is big,
then $x(t) = 0.9690x(t-1) + 0.4707y(t-1) - 0.1118z(t-1) + 0.5942$
R2: If $x(t-1)$ is middle and $z(t-1)$ is big,
then $x(t) = 0.9952x(t-1) - 0.1856y(t-1) - 0.1673z(t-1) + 1.1332$
R3: If $x(t-1)$ is middle, then $x(t) = 0.9792x(t-1) + 0.0430y(t-1) - 0.1719z(t-1) - 0.9265$

Antecedent parameters
$x(t-1)$: middle = [3.1992, −0.0243, 0.2455, 1.4921]
$y(t-1)$: small = [0.9359, −1.6004, 1.4568, −1.2401]
$z(t-1)$: big = [2.0055, 1.4497, 1.9562, 1.8957]

(c) **Rule expression**
R1: If $x(t-1)$ is middle and $y(t-1)$ is small and $z(t-1)$ is big,
then $x(t) = 0.9783x(t-1) + 0.1987y(t-1) - 0.2612z(t-1) + 1.1554$
R2: If $x(t-1)$ is middle, then $x(t) = 0.9834x(t-1) + 0.0966y(t-1) - 0.0630z(t-1) - 0.5092$

Antecedent parameters
$x(t-1)$: middle = [0.2170, 1.0405, 0.1218, 1.5640]
$y(t-1)$: small = [1.8230, −1.5130, 1.7763, −1.2380]
$z(t-1)$: big = [2.0119, 1.4863, 2.0147, 1.8887]

(d) **Rule expression**
R1: If $x(t-1)$ is middle and $y(t-1)$ is small and $z(t-1)$ is middle,
then $x(t) = 1.0240x(t-1) + 0.3557y(t-1-0.2190z(t-1) + 1.5143$
R2: If $x(t-1)$ is middle, then $x(t) = 0.9486x(t-1) - 0.1905y(t-1) + 0.2012z(t-1) - 1.2809$

Antecedent parameters
$x(t-1)$: middle = [1.1800, 0.3837, 1.1654, 1.1581]
$y(t-1)$: small = [3.1054, −1.3900, 3.5833, −1.2057]
$z(t-1)$: middle = [1.9911, −0.6432, 1.5705, 0.6044]

Thirdly, our approach is able to search for interpretable fuzzy models with a high accuracy. From the simulation results on the four examples, we can see that better interpretable rule base with a comparable or higher accuracy than the initial one can be obtained starting from the initial fuzzy model. From the initial fuzzy model with five rules, ten fuzzy sets and MSE equal to 1.4032×10^{-3} for the training data in Section 5.1, a better interpretable rule base with a higher accuracy is obtained such as the fuzzy model with four rules, three fuzzy sets and MSE equal to 5.6086×10^{-4} (Fig. 12(c)). Similar results have also been obtained from Sections 5.2–5.4.

Fourthly, from the simulation results summarized in Tables 1, 3, 5 and 7, it is noticed that better interpretability usually leads to a lower approximation accuracy among the optimized models obtained by our approach. For instance, in Section 5.2 we obtain a better interpretable rule base with four rules and three fuzzy sets (Fig. 13(c)) compared with another one with seven rules and six fuzzy sets (Fig. 13(b)), whereas the accuracy in terms of MSE is improved from 2.7389×10^{-3} to 8.9607×10^{-4}. This is consistent with the human common sense, and the trade-off between accuracy and interpretability of fuzzy systems is also easily understood.

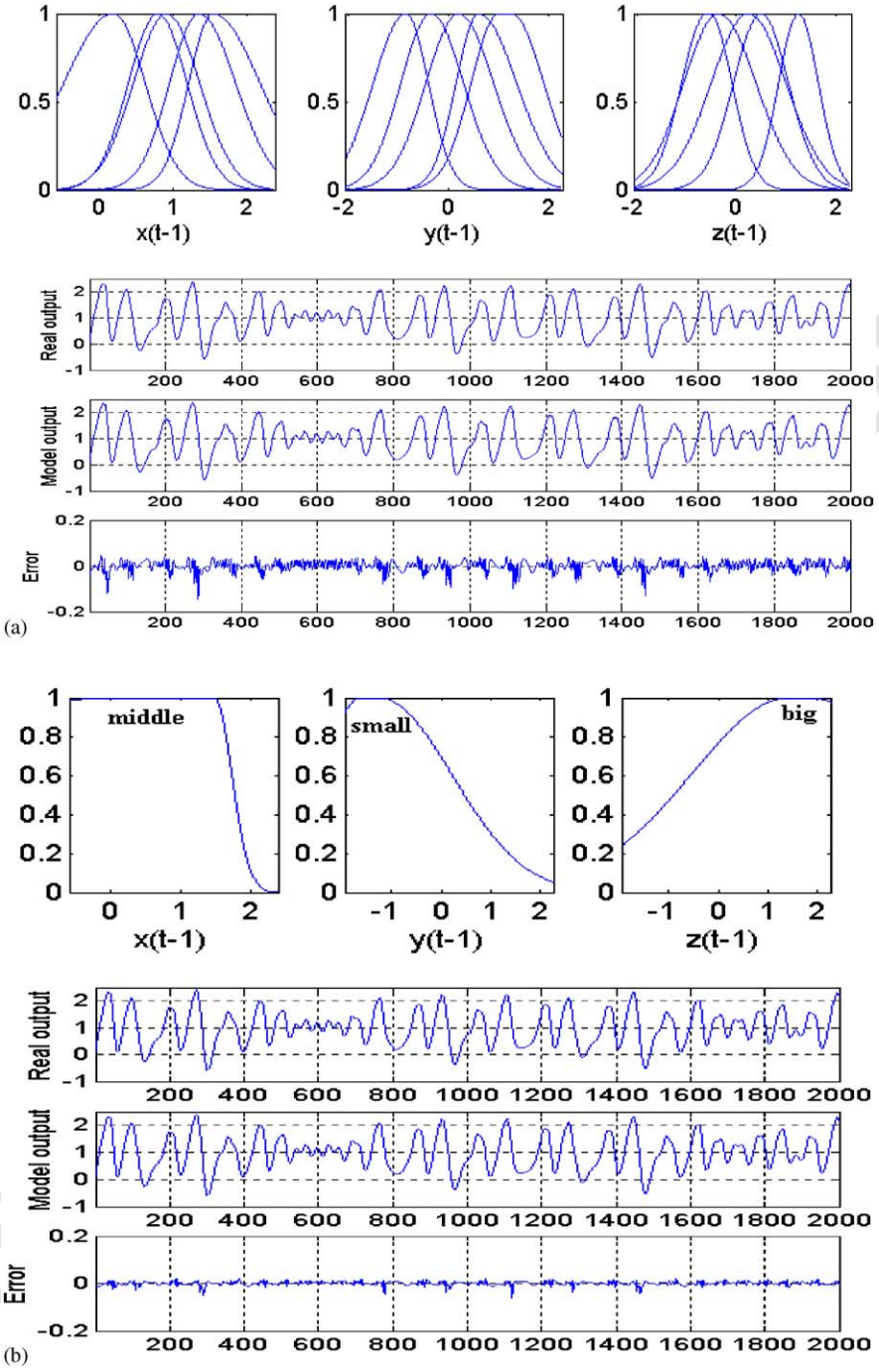
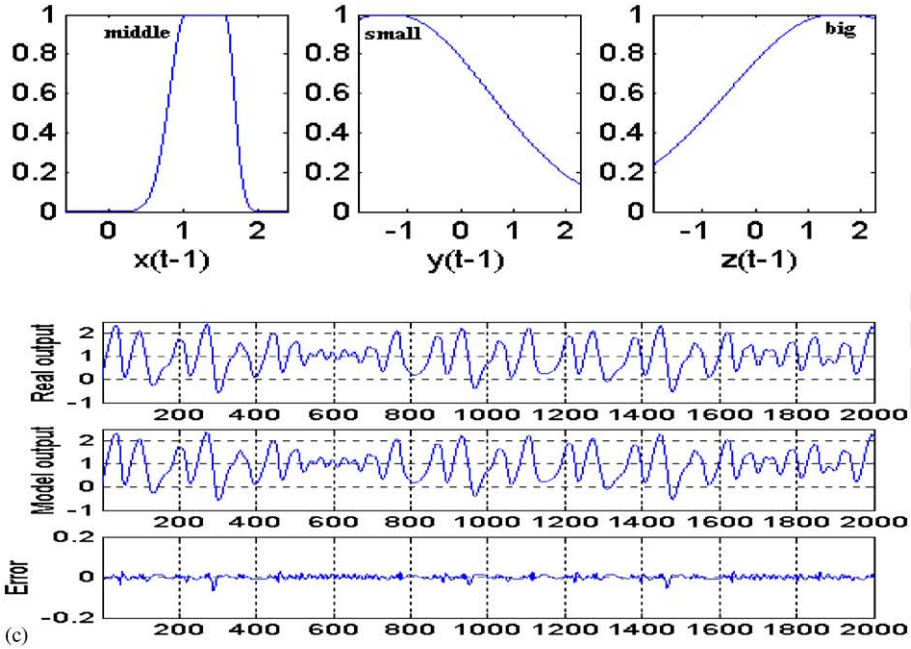
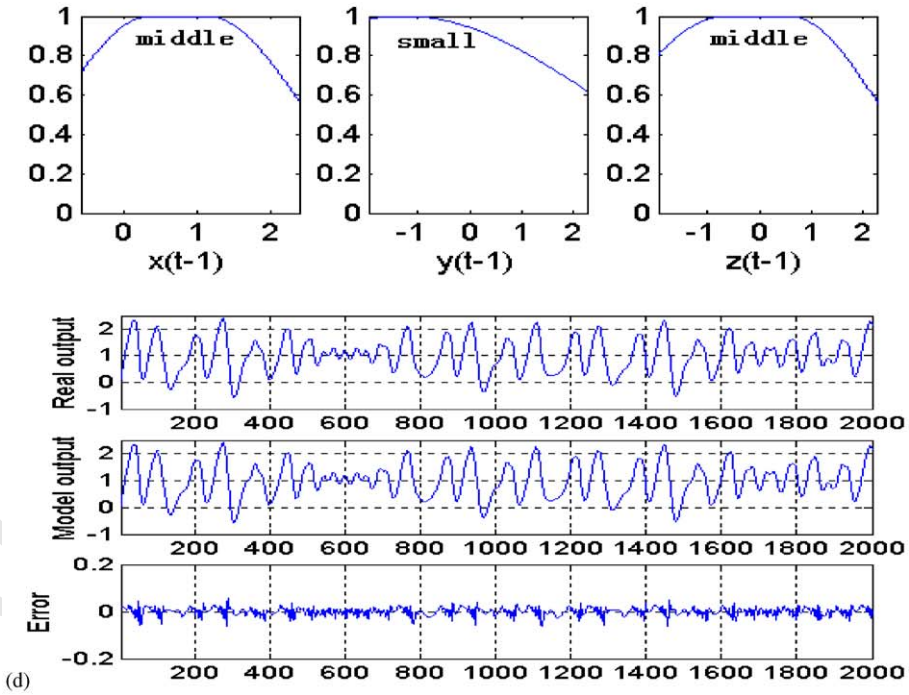


Fig. 15. Fuzzy sets distribution and the simulation results of Section 5.3: (a) initial model with 5 rules and 15 sets, (b) optimized model with 3 rules and 3 sets, (c) and (d) optimized model with 2 rules and 3 sets.



(c)



(d)

Fig. 15. (*Continued*.)

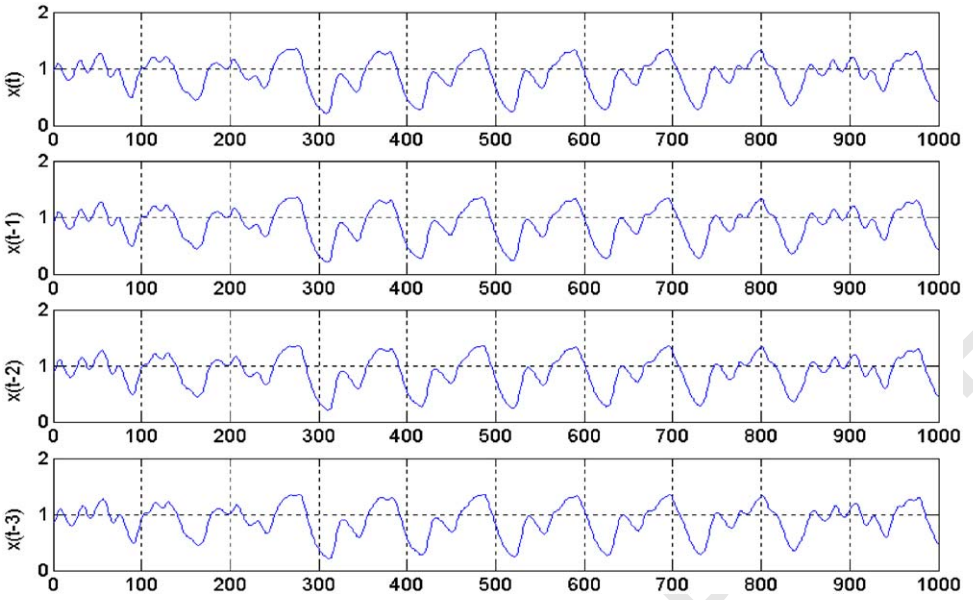


Fig. 16. Input $x(t-1)$, $x(t-2)$ and $x(t-3)$, output $x(t)$ of the Mackey–Glass system.

Fifthly, in order to guarantee a good trade-off between the interpretability and accuracy of fuzzy systems, the hierarchical chromosome formulation is used to optimize the structure of fuzzy systems at the same time to keep accuracy. And the interpretability-driven rule base simplification method and three constraints are applied to get good interpretability of fuzzy systems. Assuming there are m fuzzy variables each of which has n fuzzy sets initially. In the worst case, the computational complexity for the interpretability-driven rule base simplification method is about $O(mn^2)$. This is not the general case. When we initialize the active control genes, the parameters c_max and c_min are used to get the number of active control genes: *num_gene*. In our approach, c_max is equal to n and c_min equal to 1. And the parameter *num_gene* has the same probability to be set an integer between 1 and n. So the average computational complexity for the interpretability-driven rule base method is $O(m(n+1)^2/4)$. However, we merge similar fuzzy sets, merge fuzzy sets to deal with restricted covering, remove fuzzy sets to deal with complete covering and remove fuzzy sets which is similar to universal set or singleton set. So the number of active control genes would be reduced and the average computational complexity decreases. Another step which costs much computational time is the multi-objective decision making mechanism. As we have mentioned, this procedure requires $O(MN^2)$ comparisons, where N is the size of population and M is the number of objectives. In our simulation work, we use the Matlab 6.1 to implement experiments. The CPU is 1.8 GHZ and the RAM is 128M. The average computational time is about 130 min for the first example Nonlinear plant, 100 min for the second example Nonlinear static system, 220 min for the third example Lorenz system, and 200 min for the fourth example Mackey–Glass Time Series.

Finally, the highlight of our approach is that we apply the combination of MOGA and hierarchical GA to study the interpretability of fuzzy systems and the trade-off between interpretability and accuracy. We proposed the covering and utility concepts to study the interpretability of fuzzy systems generated from learning data for the first time we have ever known in the literature. Based on the understanding

Table 7
Fuzzy models of the nonlinear plant of Section 5.4

Ref.	No. of rules	No. of fuzzy sets	Consequent	MSE train	MSE validation
[17]	5 rules	6 Gauss.	Singleton	0.016	0.016
This paper					
Fig. 17(a)	5 rules (initial)	15 Gauss2mf.	Linear	6.0589e − 6	4.5485e − 6
Fig. 17(b)	2 rules (optimized)	3 Gauss2mf.	Linear	3.8582e − 6	3.5776e − 6
Fig. 17(c)	1 rules (optimized)	3 Gauss2mf.	Linear	3.9637e − 6	3.2976e − 6
Fig. 17(d)	1 rules (optimized)	3 Gauss2mf.	Linear	3.9637e − 6	3.2976e − 6

of interpretability of fuzzy systems, we apply MOHGA to extract interpretable fuzzy systems from data. The completeness and distinguishability and covering issues are guaranteed through the interpretability-driven rule base simplification method. Additionally, the shapes of fuzzy sets are controlled within such a method by regulating the parameters of membership functions. We use the hierarchical chromosome formulation to optimize both the structure of fuzzy systems and their parameters. Five objectives are applied and the Pareto-based MOGA decision making and fitness assignment is implemented to study the improvement of interpretability and the trade-off between interpretability and accuracy of fuzzy systems.

6. Conclusion

In this paper, we presented an approach to construct TS fuzzy models that take both the accuracy and the interpretability of fuzzy systems into accounts. Some important concepts such as covering and utility are introduced in discussing different aspects of interpretability like distinguishability and completeness. A fuzzy clustering method and the recursive least square method are employed to obtain an initial fuzzy model for the GA-based optimization. Then the interpretability-driven rule base simplification and the multi-objective hierarchical genetic algorithm are used to generate optimized fuzzy models with a high accuracy and good interpretability. Instead of encoding the consequent parameters of rules in the chromosome, we used the recursive least square method to determine them.

The proposed approach has successfully been applied to four problems taken from the literature: a synthetic nonlinear dynamic system, a nonlinear static system, the Lorenz system and the Mackey–Glass system. Comparative simulation results demonstrate that the proposed approach can obtain fuzzy models with better interpretability without deteriorating the approximation accuracy. Our proposed method shows a comparable or higher accuracy compared to other fuzzy models reported in the literature. A systematic framework for describing interpretability issues and how to effectively express the trade-off between accuracy and interpretability in the context of multi-objective optimization are open for future research.

Acknowledgements

The authors would like to thank the referees for their valuable comments.

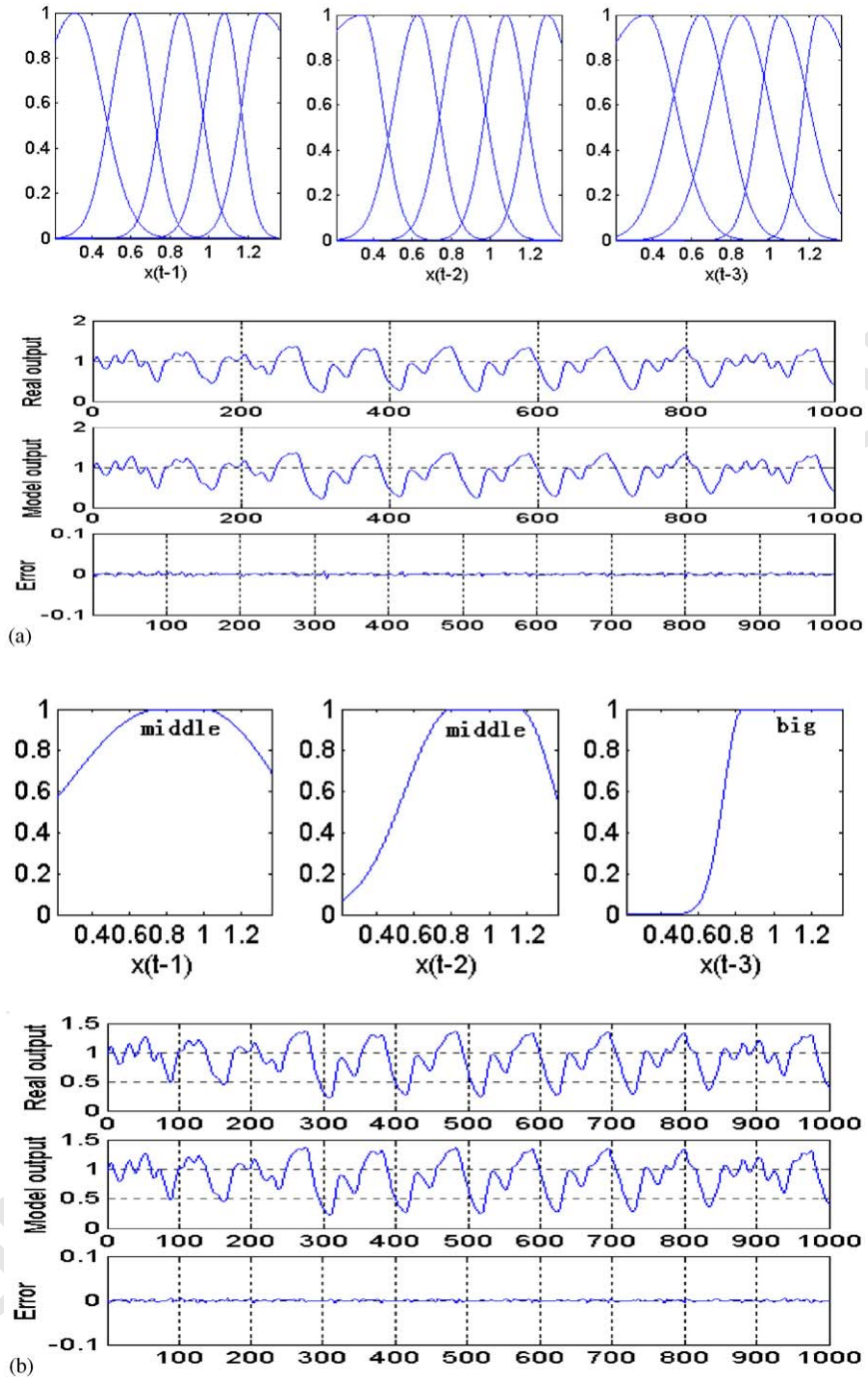
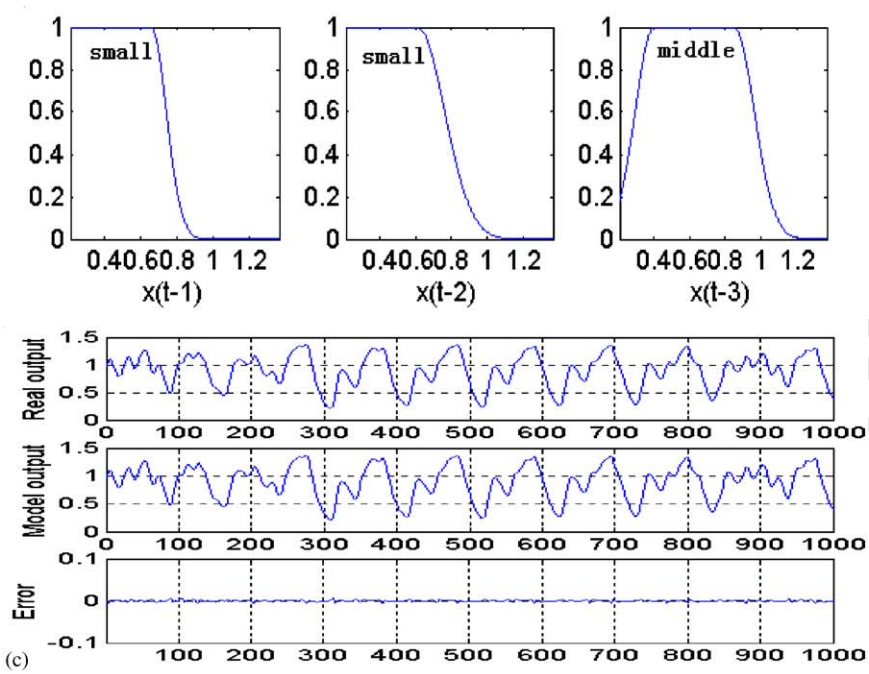
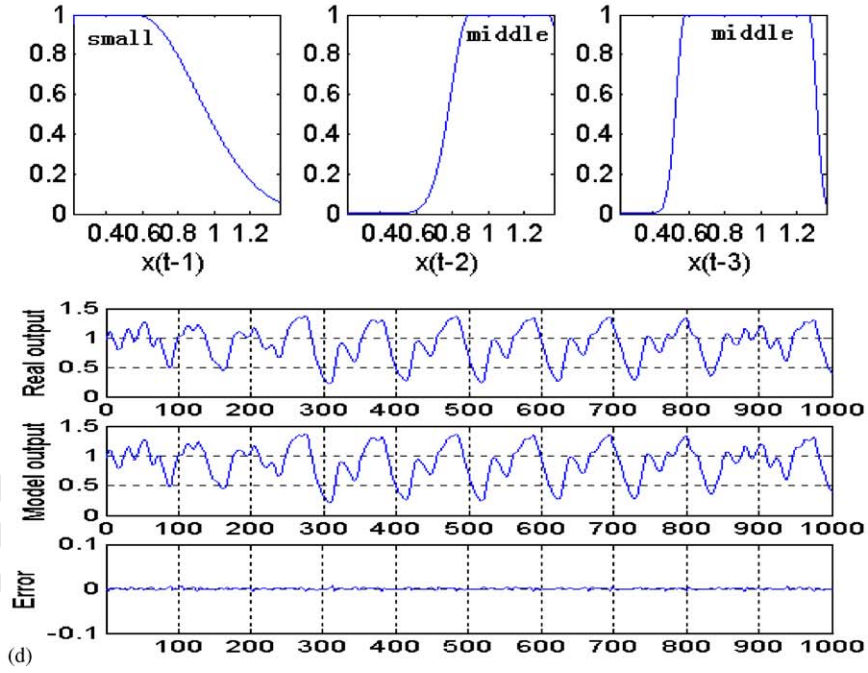


Fig. 17. Fuzzy sets distribution and the simulation results of Section 5.4: (a) initial model with 5 rules and 15 sets, (b) optimized model with 2 rules and 3 sets, (c) and (d) optimized model with 1 rules and 3 sets.



(c)



(d)

Fig. 17. (*Continued*.)

Table 8
Fuzzy model parameters for Fig. 17(b)–(d)

(b) **Rule expression**
R1: If $x(t-1)$ is middle and $x(t-2)$ is middle and $x(t-3)$ is big, then $x(t) = 2.5176x(t-1) - 2.0750x(t-2) + 0.5605x(t-3) - 0.0016$
R2: If $x(t-3)$ is big, then $x(t) = 3.0216x(t-1) - 3.1014x(t-2) + 1.0763x(t-3) + 0.0018$

Antecedent parameters
$x(t-1)$: middle = [0.5179, 0.7645, 0.4260, 0.9987]
$x(t-2)$: middle = [0.2527, 0.8045, 0.1904, 1.1599]
$x(t-3)$: big = [0.0961, 0.8324, 1.7097, 1.2256]

(c) **Rule expression**
R1: If $x(t-1)$ is small and $x(t-2)$ is small and $x(t-3)$ is middle, then $x(t) = 2.8415x(t-1) - 2.7308x(t-2) + 0.8881x(t-3) + 0.0011$

Antecedent parameters
$x(t-1)$: small = [0.1995, 0.1041, 0.0826, 0.6594]
$x(t-2)$: small = [0.0566, 0.1603, 0.1485, 0.6137]
$x(t-3)$: middle = [0.1034, 0.4042, 0.1110, 0.8492]

(d) **Rule expression**
R1: If $x(t-1)$ is small and $x(t-2)$ is middle and $x(t-3)$ is middle, then $x(t) = 2.8415x(t-1) - 2.7308x(t-2) + 0.8881x(t-3) + 0.0011$

Antecedent parameters
$x(t-1)$: small = [0.0351, 0.2237, 0.3354, 0.5683]
$x(t-2)$: middle = [0.1048, 0.8976, 0.0909, 1.3329]
$x(t-3)$: middle = [0.0487, 0.5793, 0.0400, 1.2668]

References

[1] R. Babus˘ka, Fuzzy Modeling for Control, Kluwer, Boston, MA, 1998.
[2] J.E. Baker, Reducing bias and inefficiency in the selection algorithm, Proc. Second Internat. Conf. on Genetic Algorithms, 1987, pp. 14–21.
[3] T. Bäck, Introduction to the special issue: self-adaptation, Evolutionary Comput. 9 (2) (2001) iii–iv.
[4] G. Castellano, A.M. Fanelli, E. Gentile, T. Roselli, A GA-based approach to optimization of fuzzy models learned from data, GECCO-2002 Program, New York, 2002, pp. 5–8.
[5] C.A.C. Coello, D.A. Van Veldhuizen, G.B. Lamont, Evolutionary Algorithms for Solving Multi-Objective Problems, Kluwer Academic Publishers, New York, 2002.
[6] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, Wiley, Chichester, UK, 2001.
[7] D. Dubois, H. Prade, L. Ughetto, Checking the coherence and redundancy of fuzzy knowledge bases, IEEE Trans. Fuzzy Systems 5 (3) (1997) 398–417.
[8] L.J. Eschelman, J.D. Schaffer, Real-coded genetic algorithms and interval schemata, in: D. Whitley (Ed.), Foundations of Genetic Algorithms-2, Los Altos, CA, Morgan Kaufmann, 1993, pp. 187–202.
[9] C.M. Fonseca, P.J. Fleming, Genetic algorithms for multiobjective optimization: formulation, discussion, and generation, in: Proc. Fifth Internat. Conf. on Genetic Algorithms,1993, pp. 416–423.
[10] S. Guillaume, Designing fuzzy inference systems from data: an interpretability oriented review, IEEE Trans. Fuzzy System 9 (3) (2001) 426–443.

[11] N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, Evolutionary Comput. 9 (2) (2001) 159–195.
[12] F. Herrera, M. Lozano, J.L. Verdegay, Tackling real-coded genetic algorithm: operators and tools for behavioral analysis, Artif. Intell. Rev. 12 (1998) 265–319.
[13] F. Höppner, F. Klawonn, R. Kruse, T. Runkler, Fuzzy Clustering Analysis, Wiley, New York, 1999.
[14] F. Jiménez, A.F. Gómez-Skarmeta, H. Roubos, R. Babuˇka, Accurate, transparent, and compact fuzzy models for function approximation and dynamic modeling through multi-objective evolutionary optimization, in: First Internat. Conf. on Evolutionary Multi-criterion Optimization,2001, pp. 653–667.
[15] Y. Jin, Fuzzy modeling of high-dimensional systems: complexity reduction and interpretability improvement, IEEE Trans. Fuzzy Systems 8 (2) (2000) 212–221.
[16] Y. Jin, M. Olhofer, B. Sendhoff, Dynamic weighted aggregation for evolutionary multi-objective optimization: why does it work and how?, in: Proc. Genetic and Evolutionary Computation Conf., San Francisco, 2001, pp. 1042–1049.
[17] Y. Jin, B. Sendhoff, Extracting interpretable fuzzy rules from RBF networks, Neural Process. Lett. 17 (2) (2003) 149–164.
[18] Y. Jin, W. von Seelen, B. Sendhoff, An approach to rule-based knowledge extraction, Proc. IEEE Conf. Fuzzy System 2 (1998) 1188–1193.
[19] Y. Jin, W. von Seelen, B. Sendhoff, On generating FC^3 fuzzy rule systems from data using evolution strategies, IEEE Trans. Systems Man Cybernet. 29 (6) (1999) 829–845.
[20] K.F. Man, K.S. Tang, S. Kwong, Genetic Algorithms Concepts and Designs, Springer-Verlag, London Limited, 1999.
[21] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer, New York, 1992.
[22] T. Murata, S. Kawakami, H. Nozawa, M. Gen, H. Ishibuchi, Three-objective genetic algorithms for designing compact fuzzy rule-based systems for pattern classification problems, in: Proc. Genetic and Evolutionary Computation Conf., San Francisco, 2001, pp. 485–492.
[23] K.M. Passino, S. Yurkovich, Fuzzy Control, Addison-Wesley Longman Inc., Reading, MA, 1998.
[24] I. Rojas, H. Pomares, J. Ortega, A. Prieto, Self-organized fuzzy system generation from training examples, IEEE Trans. Fuzzy Systems 8 (1) (2000) 23–36.
[25] H. Roubos, M. Setnes, GA-fuzzy modeling and classification: complexity and performance, IEEE Trans. Fuzzy Systems 8 (5) (2000) 509–522.
[26] H. Roubos, M. Setnes, Compact and transparent fuzzy models and classifiers through iterative complexity reduction, IEEE Trans. Fuzzy Systems 9 (4) (2001) 516–524.
[27] H.-P. Schwefel, Imitating evolution: collective, two level learning processes, in: U. Witt (Ed.), Explaining Process and Change-Approaches to Evolutionary Economics, University of Michigan Press, Ann Arbor, Michigan, 1992, pp. 49–63.
[28] M. Setnes, R. Babuˇka, U. Kaymak, H.R. van Nauta Lemke, Similarity measures in fuzzy rule base simplification, IEEE Trans. Systems Man Cybernet.—B: Cybernet. 28 (3) (1998) 376–386.
[29] M. Sugeno, T. Yasukawa, A fuzzy-logic-based approach to qualitative modeling, IEEE Trans. Fuzzy Systems 1 (1) (1993) 7–31.
[30] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Trans. Systems Man Cybernet. 15 (1) (1985) 116–132.
[31] K.S. Tang, K.F. Man, Z.F. Liu, S. Kwong, Minimal fuzzy memberships and rules using hierarchical genetic algorithms, IEEE Trans. Ind. Electron. 45 (1) (1998) 162–169.
[32] L. Wang, J. Yen, Exacting fuzzy rules for system modeling using a hybrid of genetic algorithms and Kalman filter, Fuzzy Sets and Systems 101 (1999) 353–362.
[33] J. Yen, L. Wang, Application of statistical information criteria for optimal fuzzy model construction, IEEE Trans. Fuzzy Systems 6 (3) (1998) 362–372.
[34] J. Yen, L. Wang, Simplifying fuzzy rule-based models using orthogonal transformation methods, IEEE Trans. Systems Man Cybernet.—B: Cybernet. 29 (1) (1999) 13–24.
[35] L.A. Zadeh, Outline of a new approach to the analysis of complex systems and decision processes, IEEE Trans. Systems Man Cybernet. SMC-3 (1973) 28–44.