# Agent-Based Evolutionary Approach for Interpretable Rule-Based Knowledge Extraction

Hanli Wang, Sam Kwong, Yaochu Jin, *Senior Member, IEEE*, Wei Wei, and Kim-Fung Man

***Abstract*—An agent-based evolutionary approach is proposed to extract interpretable rule-based knowledge. In the multiagent system, each fuzzy set agent autonomously determines its own fuzzy sets information, such as the number and distribution of the fuzzy sets. It can further consider the interpretability of fuzzy systems with the aid of hierarchical chromosome formulation and interpretability-based regulation method. Based on the obtained fuzzy sets, the Pittsburgh-style approach is applied to extract fuzzy rules that take both the accuracy and interpretability of fuzzy systems into consideration. In addition, the fuzzy set agents can cooperate with each other to exchange their fuzzy sets information and generate offspring agents. The parent agents and their offspring compete with each other through the arbitrator agent based on the criteria associated with the accuracy and interpretability to allow them to remain competitive enough to move into the next population. The performance with emphasis upon both the accuracy and interpretability based on the agent-based evolutionary approach is studied through some benchmark problems reported in the literature. Simulation results show that the proposed approach can achieve a good tradeoff between the accuracy and interpretability of fuzzy systems.**

***Index Terms*—Hierarchical chromosome formulation, interpretability and accuracy, multiagent system, multiobjective decision making.**

## I. Introduction

THE fundamental concept of fuzzy reasoning was first introduced by Zadeh [1] in 1973, and the past few years have witnessed a rapid growth in a number of applications of fuzzy systems. 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 characteristics that fuzzy 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. This is essentially due to two main factors: 1) The number of rules and fuzzy sets are usually larger than necessary, and 2) the topology of fuzzy sets is inappropriate. So there is always a tradeoff between the interpretability and accuracy of fuzzy systems constructed from training data. Recently, increasing attention has been paid to improve the interpretability of fuzzy systems [2]–[15], and [16] presents an up-to-date state of the current research.

In this work, our main purpose is to propose an approach to study the interpretability of fuzzy systems and the tradeoff between the accuracy and interpretability of fuzzy systems autonomously generated from the learning data. So this is a multiobjective optimization problem by its very nature. And the multiobjective evolutionary algorithm is very suitable to solve this problem. In the multiobjective evolutionary algorithm, a main advantage is that many solutions, each of which represents an individual fuzzy system, can be obtained in a single run, and the accuracy and interpretability issues can be incorporated into the multiple objectives to evaluate the solutions. Thus, the improvement of interpretability and the tradeoff between the accuracy and interpretability can be easily studied. On the other hand, the neural-network-based method is very effective to generate fuzzy systems from the sampling data, such as the methods in [17]–[19]. However, there is only one fuzzy system that can be obtained by the neural-network-based method. Additionally, in order to generate interpretable fuzzy rules, not only the accuracy, but also, the interpretability conditions should be considered. This means that in a neural-network-based approach, extra regularization terms that guarantee the interpretability should be added alongside the accuracy index. One difficulty in this approach is how to properly select the regularization term and determine its relative importance in the whole cost function.

In this paper, we propose an agent-based evolutionary approach to construct fuzzy systems from training data with emphasis on both the accuracy and interpretability. We want to explore a more compact fuzzy system considering not only the number of rules but also the number of fuzzy sets. In addition, we also hope to get more appropriate distributions of fuzzy sets with no interference from human beings. It is a very difficult task compared with the methods stated in [20]–[22]. In [22], the author used some important endpoints to distribute membership functions. The number of fuzzy sets is fixed and there are some limitations about the distribution of these fuzzy sets. In [20] and [21], the fuzzy sets are prepartitioned without considering more appropriate distributions. More important, it is almost impossible to have a good understanding about an unknown complex system, not to mention giving the linguistic values for each fuzzy variable in advance. In this work, we suggest an agent-based scheme. In this multiagent system, each agent has the autonomous capability to determine the number

Manuscript received September 1, 2003; revised February 27, 2004. This work was supported in part by the City University of Hong Kong under Strategic Grant #7001416 and in part by the Key Natural Science Foundation of Zhejiang Province #ZD0107. This paper was recommended by Guest Editor Y. Jin.

H. Wang is with the Department of Computer Science, City University of Hong Kong, Hong Kong. He is also with the College of Electrical Engineering, Zhejiang University, Hong Kong (e-mail: wanghl@cityu.edu.hk).

S. Kwong is with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: cssamk@cityu.edu.hk).

Y. Jin is with Honda Research Institute Europe, Offenbach/Main 63073, Germany (e-mail: yaochu.jin@honda-ri.de).

W. Wei is with the College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: wwei@cee.zju.edu.cn).

K.-F. Man is with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong (e-mail: eekman@cityu.edu.hk).

Digital Object Identifier 10.1109/TSMCC.2004.841910

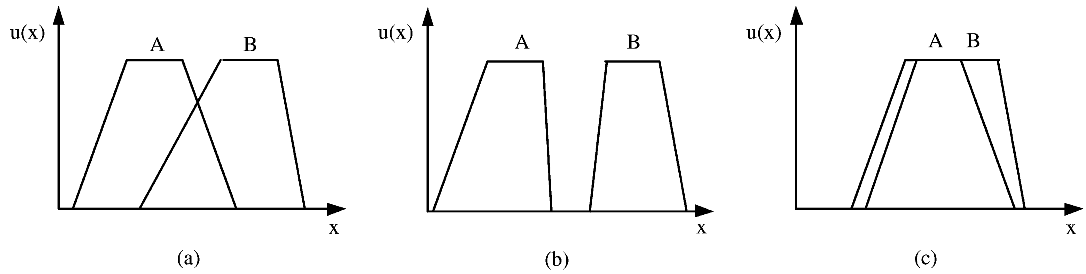


Fig. 1. Fuzzy partitioning: (a) Overlap moderately. (b) Do not overlap. (c) Overlap too much.

of fuzzy sets and the distribution of these fuzzy sets considering the interpretable issues of fuzzy systems. We achieve these goals by means of the hierarchical chromosome formulation and an interpretability-based regulation method. Then, with these fuzzy sets at hand, the agents will apply the Pittsburgh-style approach to extract interpretable fuzzy rules. The reason for us to adopt the Pittsburgh-style approach is because the fuzzy rule set can be treated as one solution, and many solutions can be obtained simultaneously in a single run so that we can compare the performance of the solutions based on the accuracy and interpretability. The agents apply NSGA-II [23] multiobjective decision-making method to evaluate fuzzy rule sets candidates. After the agents have finished self-evolving, they can interact with each other by switching fuzzy sets information and also give birth to new agents. Based on the multiple criteria about the accuracy and interpretability of fuzzy systems, the elite agents are retained, whereas the obsolete agents are dead.

The paper is organized as follows. Section II discusses the interpretability issues of fuzzy systems. The agent-based evolutionary approach used to construct interpretable fuzzy systems is discussed in Section III. In Section IV, the experimental results are given on some benchmark problems. Finally, we conclude this paper and give the future work prospect in Section V.

## II. Interpretability of Fuzzy Systems

The most important motivation to use a fuzzy system is that it uses linguistic rules to infer knowledge, making it similar to the way that humans think. Methods for constructing fuzzy models from the training data should not be limited to finding the best approximation of data only. It is more important to extract knowledge from training data in the form of fuzzy rules that can be easily understood and 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 take it for granted that fuzzy rules are easy for human beings to understand. However, it is not necessarily true for complex systems. In the following, we will discuss some important concepts about the interpretability of fuzzy systems.

### *A. 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 can describe this idea in the following definition.

*Completeness*: For each input variable $x_i$ (an element of the input vector $\mathbf{x} = [x_1, x_2, \ldots, x_{\mathrm{n}}]^{\mathrm{T}}$), 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:

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

where $U_i$ is the universe of $x_i$, and $n$ 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 [2], [3], [7], and [24]. This similarity measure can be interpreted in many different ways depending on the application context. However, an important definition is given in [24]: Similarity between fuzzy sets is defined as the degree to which fuzzy sets are equal. 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 overlap enough. On the other hand, if the similarity is too big, then it indicates that the two fuzzy sets overlap too much, 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 $s = S(A, B), s \in [0, 1]$. We use the following similarity measure between fuzzy sets [24]:

$$S(A, B) = \frac{M(A \cap B)}{M(A \cup B)} = \frac{M(A \cap B)}{M(A) + M(B) - M(A \cap B)} \quad (2)$$

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 in [11] and [12] is described as

$$S(A, B) = \frac{\sum_{j=1}^{m} [u_A(x_j) \wedge u_B(x_j)]}{\sum_{j=1}^{m} [u_A(x_j) \vee u_B(x_j)]} \quad (3)$$

on a discrete universe $U = \{x_j | j = 1, 2, \ldots, m\}$. $\wedge$ and $\vee$ in (3) are the minimum and maximum operators. In our approach, we use this form to calculate the similarity of fuzzy sets because it is computationally simple and effective.

### B. Consistency

Another important issue about interpretability is the consistency among fuzzy rules and the consistency with *apriori* knowledge. Consistency among fuzzy rules means that if two or more rules are simultaneously fired, then their conclusions should be coherent [3] (i.e., if two or more rules have the similar antecedents, their consequents should also be similar). The consistency with *apriori* knowledge means that the fuzzy rules generated from data should not be in conflict with the expert knowledge or heuristics. A definition of consistency and its calculation method among fuzzy rules is given in [7]. Also one important factor about the consistency is that the antecedents of one rule may include those of another rule. Take the following three rules for example:

$\mathrm{R}_1$) If $x_1$ is small and $x_2$ is small and $x_3$ is big, then $y$ is big.

$\mathrm{R}_2$) If $x_1$ is small and $x_2$ is small, then $y$ is medium;

$\mathrm{R}_3$) If $x_1$ is small, then $y$ is small.

Usually, we express the above three rules in the following hierarchical form:

If $x_1$ is small and $x_2$ is small and $x_3$ is big, then $y$ is big.
Else if $x_1$ is small and $x_2$ is small, then $y$ is medium.
Else if $x_1$ is small, then $y$ is small.

In [25], it is called inclusion relation. If two fuzzy rules are compatible with an input vector and one rule is included in the other rule, the former should have a larger weight than the latter in the fuzzy inference to calculate the output value. Let us consider the following two rules $R_i$ and $R_j$:

$R_i$) 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 $B_{\mathrm{jm}}(y_m)$.

When the inclusion relation $A_{jk} \subseteq A_{ik}$ holds for all of the input variables (i.e., for $k = 1, 2, 3, \ldots, n$), we say that the rule $R_j$ is included in the rule $R_i$ (i.e., $R_j \subseteq R_i$). For the rule $R_i$, the fire-strength $u_i$, also called weight of the $i$ th rule is defined as follows:

$$u_i(x) = u_{A_{i1}}(x_1) \wedge u_{A_{i2}}(x_2) \wedge \cdots \wedge u_{A_{in}}(x_n), \quad i = 1, \ldots, R \tag{4}$$

where $R$ is the total number of fuzzy rules in the rule base, $\wedge$ is the *and* operator, and *minimum* and *product* are the most common *and* operators. As far as the inclusion relation is concerned, a factor $\lambda$ related to the rule $R_i$ is defined as

$$\lambda_i(x) = \prod_{R_k \subseteq R_i} (1 - u_k(x)), \qquad k = 1, \ldots, R; k \neq i. \tag{5}$$

Then, the fire-strength of the rule $R_i$ considering the inclusion factor is updated as

$$u_i = \lambda_i u_i(x), \qquad i = 1, \ldots, R. \tag{6}$$

### C. 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. A compact fuzzy system is always desirable when the number of input variables increases.

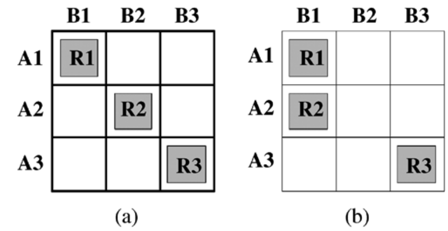


Fig. 2. Fuzzy system with two input variables (three fuzzy sets for each variable) and three rules. (a) Sufficient utility. (b) Insufficient utility because fuzzy set B2 is not utilized by any rules.

### D. Utility

Even if the partitioning of 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 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. 2(b)]. Then, the unused fuzzy sets should be removed from the rule base resulting in a more compact fuzzy system.

## III. Agent-Based Evolutionary Approach

In this paper, we propose an agent-based evolutionary approach to constructing fuzzy models with considerations of both the accuracy and interpretability. The basic modeling ideas are illustrated in Fig. 3. There are two kinds of agents in the multiagent system: the arbitrator agent (AA) and the fuzzy set agent (FSA). These fuzzy set agents are distributed independently and obtain information from the AA in which the information is expressed in terms of training data in our specified research context. We name the agent as FSA because it can autonomously determine its own fuzzy sets information, such as the number and distribution, and then learn to construct fuzzy rule base based on the obtained fuzzy sets. As far as the social behavior is concerned, the FSA is able to cooperate and compete with other fuzzy set agents. Different from the parallel GA where an individual in one subpopulation can migrate into another subpopulation and no subpopulations will be dead (i.e., removed from the evolutionary process). While in our agent-based evolutionary approach, the FSAs cooperatively exchange their fuzzy sets information by ways of crossover and mutation of the hierarchical chromosome and generate offspring FSAs. After the self-evolving of the FSAs, they send their fitness information in the form of accuracy and interpretability to the AA. In the current work, the AAs use the NSGA-II algorithm to evaluate the FSAs and judge which fuzzy set agents should survive and be kept to the next population, whereas the obsolete agents are dead. The agent-based evolutionary approach has the characteristics of parallel GA. However, the agents have the ability of competing with each other based on the considerations of accuracy and interpretability. They do not exchange individuals just

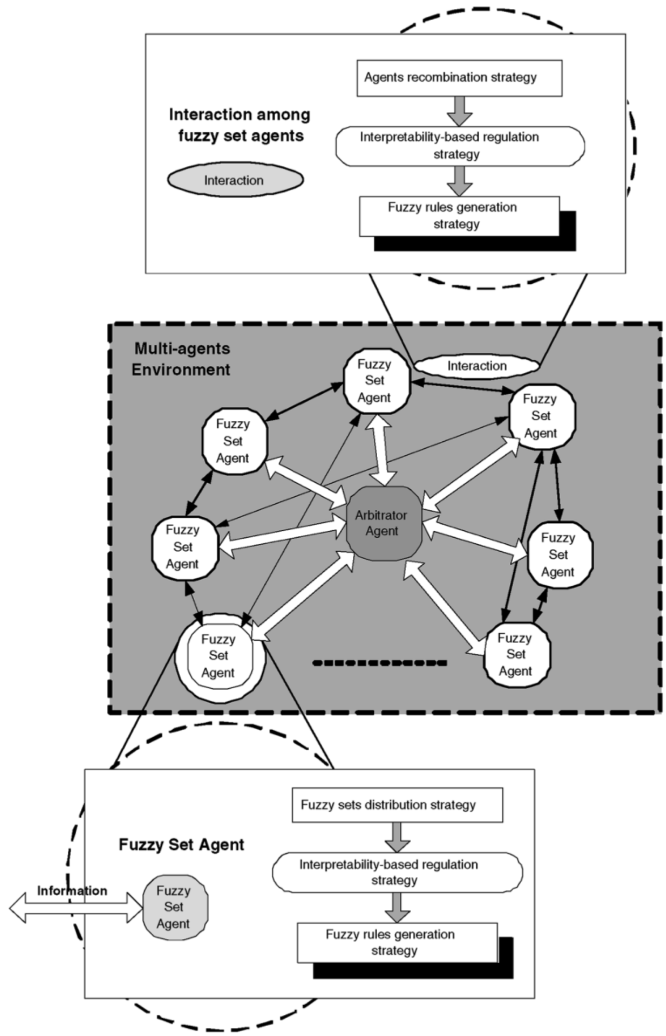


Fig. 3. Multiagents system framework.

like parallel GA subpopulations; instead, they cooperatively exchange information about the fuzzy sets. In the following, we will discuss how the proposed agent-based approach constructs accurate and interpretable fuzzy systems.

### A. Autonomous FSAs' Intra Behavior

In the multiagent system, the FSAs employ the fuzzy sets distribution strategy, the interpretability-based regulation strategy, as well as fuzzy rules generation strategy to build accurate and interpretable fuzzy systems. The details of the strategies are discussed below.

*1) Fuzzy Sets Distribution Strategy:* Inspired by the insight of biological DNA structure, a hierarchical chromosome formulation for GA is introduced in [26]–[28], where the genes of the chromosome are classified into two different types: control genes and parameter genes. These genes are arranged in a hierarchical form so that the control genes are able to manipulate the parameter genes in a more effective manner. To indicate the activation of the control genes, an integer 1 is assigned for each control gene that is ignited, whereas 0 is for turning off. When 1 is assigned, the associated parameter gene corresponding to that active control gene is activated. The effectiveness of this chromosome formulation enables the number as well as the distribution of fuzzy sets to be optimized. Fig. 4 illustrates the concept further.

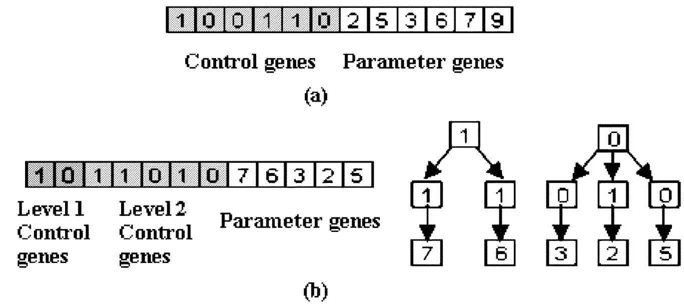


Fig. 4. Example of hierarchical chromosome representation. (a) Two-level gene structure with phenotype $X = (2, 6, 7)$. (b) Three-level gene structure with phenotype $X = (7, 6)$.

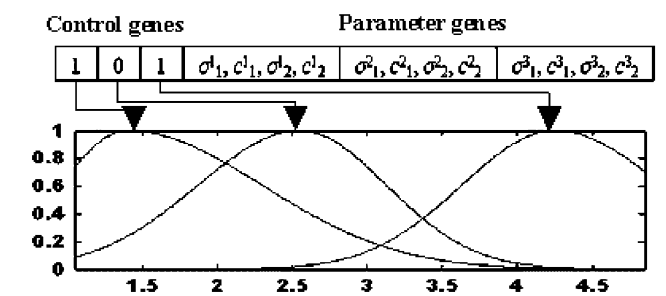


Fig. 5. Example of hierarchical formulation.

For each fuzzy variable $x_i$, we determine the possible maximal number of fuzzy sets $M_i$ so that it can sufficiently represent this fuzzy variable. For $N$ dimensional problems, there are totally $M_1 + M_2 + \cdots + M_N$ possible fuzzy sets. So there are $M_1+M_2+\cdots+M_N$ control genes coded as bits 0 or 1, where 1 is assigned to represent that the corresponding parameter gene, which is dominated by this control gene, is selected for involvement in an evolutionary process; otherwise, 0 is for turning off. 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 Gauss2mf function depends on four parameters $\sigma_1, c_1, \sigma_2$, and $c_2$ as given by

$$\mu(x;\sigma_1,c_1,\sigma_2,c_2) = \left\{ \begin{array}{c} \exp\left[\frac{-(x-c_1)^2}{2\sigma_1^2}\right] : x < c_1 \\ 1 : c_1 \le x \le c_2 \\ \exp\left[\frac{-(x-c_2)^2}{2\sigma_2^2}\right] : c_2 < x \end{array} \right\} \quad (7)$$

where $\sigma_1$ and $c_1$ determine the shape of the leftmost curve. The shape of the rightmost curve is specified by $\sigma_2$ and $c_2$. So we use the parameter list $[\sigma_1, c_1, \sigma_2, c_2]$ to represent one parameter gene (i.e., a fuzzy set expressed in the form of a Gauss2mf membership function). 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 Gauss2mf covers the universe sufficiently. An example of the relationship between control genes and parameter genes is given in Fig. 5. The FSA initializes its own control genes and parameter genes randomly.

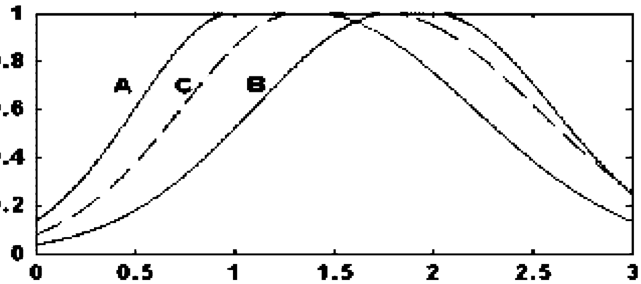


Fig. 6. Merging $A$ and $B$ to create $C$.

*2) Interpretability-Based Regulation Strategy:* Although the FSA initializes the fuzzy sets, the interpretability issues, such as distinguishability, are not guaranteed automatically. So the agent applies the interpretability-based regulation strategy on the active fuzzy sets to obtain a better distribution of fuzzy sets and a more compact fuzzy system. In this work, we define a fuzzy set $A$ using the membership function $u_A(x; a_1, a_2, a_3, a_4)$, where $a_1, a_2, a_3, a_4$ are the lower bound, left center, right center, and upper bound of the definition domain, respectively $(a_1 \leq a_2 \leq a_3 \leq a_4)$. However, we use the Gauss2mf as the membership function, so it is not easy to obtain $a_1$ and $a_4$ just like the triangular or the trapezoidal ones. We need to calculate $a_1$ and $a_4$ using a very small number $\varepsilon$ (for example, 0.001), which is regarded as equal to zero $u_A(a_1; a_1, a_2, a_3, a_4) = u_A(a_4; a_1, a_2, a_3, a_4) = \varepsilon$. Nevertheless, the interpretability-based regulation method is also applicable to all other types of membership functions besides Gauss2mf. The interpretability-based regulation strategy includes the following two actions.

*a) Merging Similar Fuzzy Sets:* An example of the similarity measure between two fuzzy sets is given as in (3). 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

$$c_1 = \min(a_1, b_1), c_2 = \lambda_2 a_2 + (1 - \lambda_2) b_2$$
$$c_3 = \lambda_3 a_3 + (1 - \lambda_3) b_3, c_4 = \min(a_4, b_4). \quad (8)$$

The parameters $\lambda_2, \lambda_3 \in [0, 1]$ determine the relative importance about the influence of the fuzzy sets $A$ and $B$ have 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 equal to 0.55. Fig. 6 illustrates the case for merging $A$ and $B$ to create $C$.

*b) 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 $U(u_U(x) = 1)$ is greater than an upper threshold $\theta_U$ or smaller than a lower threshold $\theta s$, then we can remove it from the rule base. In the first case, the fuzzy set is very similar to the universal set and in the latter case, similar to a singleton set. Neither case is desirable for interpretable rule base generation. We set $\theta_U = 0.9, \theta s = 0.05$ in this work.

After implementing the interpretability-based regulation strategy, we have the assumption that the FSA obtains a fuzzy system with $M_1^a + M_2^a + \cdots + M_N^a$ sets, where $0 \leq M_i^a \leq M_{\rm i}$ and the case that $M_i^a$ is equal to 0 indicates that the corresponding fuzzy variable is not involved in the modeling of fuzzy systems resulting in the dimensionality reduction by one.

*3) Fuzzy Rules Generation Strategy:* In the stage of fuzzy rules generation, FSAs use the Pittsburgh-style approach to extract rules. Assume there are $N$ fuzzy variables, $M_i^a$ is the number of active fuzzy sets for variable $x_i$. We also consider the "don't care" conditions (also called incomplete rules) so the total maximum number of possible fuzzy rules is $(M_1^a + 1) \times (M_2^a + 1) \ldots \times (M_N^a + 1)$ for $N$-dimensional problems. The task of FSAs in this stage is to find a small number of rules considering both the accuracy and interpretability. In the following, we will discuss how the FSAs achieve these goals.

*a) Initialization of the Rules Population:* In the Pittsburgh-style genetic-based machine learning approach, the search for a compact rule set with high-performance ability corresponds to the evolution of a population of fuzzy rule sets. In this work, each fuzzy rule is coded as a string of the length $N$. We express the string as an array in the computer program, and the $i$th element of the array indicates which fuzzy set of the $i$th fuzzy variable is fired. The $i$th element is denoted as $c_i$ and initially set to an integer between 0 and $M_i^a$ with the same probability $1/(M_i^a + 1)$. If $c_i$ is greater than zero, it is indicated that the $c_i$th fuzzy set of the $i$th fuzzy variable is fired, whereas if $c_i$ is equal to zero, this means that the $i$th fuzzy variable does not play a role in the rule generation. As far as the $i$th fuzzy variable is concerned, in the stage of fuzzy rules generation strategy, there are $M_i^a$ active fuzzy sets related to this variable. We initialize $c_i$ equal to zero considering the incomplete fuzzy rule (i.e., the $i$th fuzzy variable does not participate in the rule generation), and $c_i$ should be equal to or less than $M_i^a$ because there are only $M_i^a$ active fuzzy sets that exist for the $i$th fuzzy variable. Then, the FSA sets the population size $N_{\rm pop}$ (i.e., the number of individuals or solutions involved in the evolutionary algorithm). In the fuzzy rules generation strategy of this work, each individual is a fuzzy rule sets that represents a fuzzy rule base. For each individual of the fuzzy rule set population, it is represented as a concatenated string of the length $N \times N_{\rm rule}$, where $N_{\rm rule}$ is a predefined integer to describe the size of the initial fuzzy rule base. In this concatenated string, each substring of the length $N$ represents a single fuzzy rule. Note that we use the recursive least square method [29] and the heuristic procedure in [20], [30]–[32] to determine rule consequents for function approximation problems and classification problems, respectively, so the rule consequents are not coded as parts of the concatenated string. The fuzzy rule sets are randomly initialized so that the cell value of the concatenated string represents one of the fuzzy sets of the corresponding fuzzy variable or is equal to zero indicating "don't care" conditions.

*b) Crossover and Mutation:* Offspring rule sets are generated by crossover and mutation. As far as the crossover is concerned, one-point crossover is used (Fig. 7). The crossover operation randomly selects a different cut-off point for each parent to generate offspring rule sets. A mutation operation randomly replaces each element of the rule sets string with another linguistic

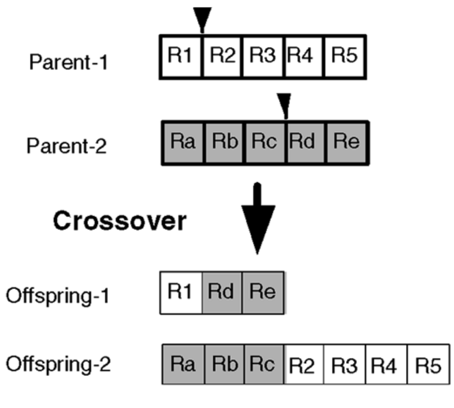


Fig. 7. Crossover operation.

value if a probability test is satisfied. Elimination of existing rules and addition of new rules can also be used as mutation operations. Such mutation operations change the number of rules in the rule sets string. Note that the crossover and mutation operations maybe introduce the same rules, the FSA will check the offspring fuzzy rule base to delete the same rules and maintain single among all of the rules after the crossover and mutation operations, so the consistency of fuzzy systems is guaranteed.

*c) Evaluation Criteria and Selection Mechanism:* The FSA uses the following three criteria to evaluate fuzzy rule set candidates: 1) Accuracy: the accuracy is measured in terms of mean-squared error (MSE) for function approximation problems and classification error rates for classification problems; 2) the number of fuzzy rules; 3) the total length of fuzzy rules [9]: the total number of the rule antecedents displayed in the rule base.

For the function approximation problems, the first-order Takage–Sugeno (TS) fuzzy system [33] is generated. The TS fuzzy system is very suitable for the approximation of dynamic systems, and the first-order TS fuzzy system is very common and effective. In our current work, unlike other GA-based methods for generating fuzzy rules, the rule consequents are not involved in the chromosome encoding. Instead, we use the recursive least square method to calculate the rule consequents for function approximation problems. So this approach has a limitation in that it is suitable for the first-order TS fuzzy modeling. However, a clear advantage of doing this is that it can save the searching time and fully exploit the sampling data. During the computation, we use the updated rule fire-strength in (6) to calculate the conclusion.

As far as the classification problems are concerned, the heuristic procedure is applied to generate rule consequents from the training pattern data. For each $n$-dimension training pattern data $\mathbf{X}^i = [\mathrm{x}_1^i, \mathrm{x}_2^i, \ldots, \mathrm{x}_n^i]$, the fire-strength of rule $R_i$ considering the inclusion relation is calculated using (6). Then, for each of the $c$ classes, the sum of the fire-strength related to rule $R_i$ is calculated as

$$\beta_{\mathrm{Class}j}(R_i) = \sum_{X^k \in \mathrm{Class}j} \hat{u}_i(X^k), \qquad j = 1, 2, \ldots, c. \quad (9)$$

Find the class $C_i$ as the consequent of rule $R_i$ which has the maximum value of $\beta_{\mathrm{Class}j}$. If the maximum value of $\beta_{\mathrm{Class}j}$ cannot be uniquely specified, that is, there is more than one class that has the same maximum value, the fuzzy rule $R_i$ is removed from the rule base. After the rule base is constructed, we calculate the classification accuracy through the single winner rule method [34]. For each training pattern data $\mathbf{X}^{\mathrm{i}}$, the winner rule $R_i$ rule is determined as

$$\hat{u}_i(X^i) = \max\{\hat{u}_k(X^i) | k = 1, 2, \ldots, R\} \qquad (10)$$

where $R$ is the number of fuzzy rules. If the class result is not the actual one or more than one fuzzy rules have the same maximum fire-strength, the classification error increases one.

Based on the foregoing three criteria, the FSA uses the NSGA-II [23] algorithm to evaluate the fuzzy rule set candidates. In the multiobjective evolutionary optimization problem, *Pareto optimum* [35] is the most commonly accepted term in the literature. The *Pareto optimal* is defined as: A vector of decision variables $\boldsymbol{x}^* \in F$ is *Pareto optimal* if there does not exist another $\boldsymbol{x} \in F$ such that $f_i(\boldsymbol{x}) \le f_i(\boldsymbol{x}^*)$ for all $i = 1, \ldots, k$ and $f_j(\boldsymbol{x}) < f_j(\boldsymbol{x}^*)$ for at least one $j$. It is based on the minimization problems and $F$ denotes the decision variable space and $f_i$ is one of the $k$ function objectives. This definition says that $\boldsymbol{x}^*$ is Pareto optimal if there exists no feasible vector of decision variables $\boldsymbol{x} \in F$ which would decrease some criteria without causing a simultaneous increase in at least one other criterion. This concept always gives not a single solution, but rather a set of solutions called the *Pareto optimal set*. The vectors $\boldsymbol{x}^*$ corresponding to the solutions included in the Pareto optimal set are called *nondominated*. The NSGA-II is a very famous Pareto-based multiobjective evolutionary algorithm. In the current work, we mainly apply the selection mechanism of the NSGA-II algorithm. In order to compare the fuzzy rule set candidates, we predefine the preference for the three criteria. The accuracy is predefined the first priority and the other two criteria about the interpretability are predefined the same second priority. In other words, we first compare two fuzzy rule set candidates according to the accuracy only. If these two candidates have the same accuracy level based on our preference, then we compare the other two criteria to determine which rule set candidate is better. If one rule set candidate is better than the other based on the accuracy preference, then there is no need to compare the other two criteria and we can know which candidate is better. In the current work, we use the difference of the accuracy value of fuzzy rule set candidates to design the preference. If the difference is less than or equal to a predefined value, then it is considered that the candidates have the same accuracy level. Otherwise, if the difference is greater than the predefined value, then we can determine which candidate is better without continuing to compare the other two criteria. We take such measures because a fuzzy system constructed by learning from data is meaningful with a certain degree of accuracy. Suppose that there are $N_{\mathrm{pop}} + N_{\mathrm{offs}}$ rule set candidates, where $N_{\mathrm{pop}}$ is the parent population size and $N_{\mathrm{offs}}$ is the number of offspring resulting from crossover and mutation operations. The FSA selects $N_{\mathrm{pop}}$ best candidates from the mixed populations. It is an elitism strategy by nature.

Notice that during the course of rules generation, the sufficient utility that we have discussed in Section II is not guaranteed. So the FSA recognizes the unutilized active fuzzy sets and flips their corresponding control genes from 1 to 0 to guarantee the sufficient utility of fuzzy systems at the end of the evolution.

### B. Interaction Among Agents

The FSAs can interact with each other. In the current work, we assume that the number of offspring FSAs $(N_{a\,\mathrm{off}})$ that we want to generate is even and less than or equal to the number of FSAs $(N_{a\,\mathrm{cur}})$ in the current population: $N_{a\,\mathrm{off}} \leq N_{a\,\mathrm{cur}}$. We select $N_{a\,\mathrm{off}}$ FSAs from the current agent population and use the crossover and mutation operations to generate $N_{a\,\mathrm{cur}}$ offspring agents (i.e., two parent agents generate two offspring agents). The $N_{a\,\mathrm{off}}$ FSAs are different and selected randomly with the same probability. It is because such a selection mechanism is simple and easy to implement. Then, crossover and mutation operations are implemented on both the control genes and parameter genes of two paired parent agents, and two offspring agents are generated. The offspring agents use the interpretability-based regulation strategy and fuzzy rules generation strategy to obtain the fuzzy rule base. Thus, the cooperation among the FSAs are achieved by exchanging fuzzy sets information and generating child agents. Then, four criteria including the three foregoing criteria and the number of fuzzy sets are transferred to the AA. As mentioned above, the accuracy is predefined the first priority and the other three criteria are predefined the same second priority. The AA implements the NSGA-II algorithm to evaluate the parent and offspring fuzzy set agents and select $N_{a\,\mathrm{cur}}$ best agents to become the next agent population. Better FSAs considering both the accuracy and interpretability survive from the competition, whereas the worst ones are discarded from the evolutionary process. We endow the agents with the ability to cooperate and compete with other agents to achieve the global goal: constructing fuzzy systems considering both the accuracy and interpretability.

## IV. Experimental Results

In order to examine the performance of our agent-based evolutionary approach, we use three benchmark problems in the literature. Matlab 6.1 is applied to implement the experiments. To prepare the training and test data for Example $A$ and $B$, we use the Simulink Toolbox of Matlab to build the simulated model to generate the sampling data (see the description in the corresponding part). As far as Example $C$ Iris Data is concerned, the sampling data are downloaded from the University of California, Irvine (UCI) database [36].

### A. Nonlinear Plant With Two Inputs and One Output

The second-order nonlinear plant is studied by Wang and Yen in [13]–[15], Roubos and Setnes *et al.* in [11] and [12], and Jiménez, *et al.* in [4]

$$y(k) = g(y(k-1), y(k-2)) + u(k) \quad (11)$$

$$g(y(k-1), y(k-2)) = \frac{y(k-1)y(k-2)(y(k-1)-0.5)}{1+y^2(k-1)+y^2(k-2)}. \quad (12)$$

The goal is to approximate the nonlinear component $g(y(k-1), y(k-2))$ of the plant with a fuzzy model. In this work, 400 sampling data points were generated from the plant model. 200 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 $u(k) = \sin(2\pi k/25)$. The 400 simulated data points are shown in Fig. 8.

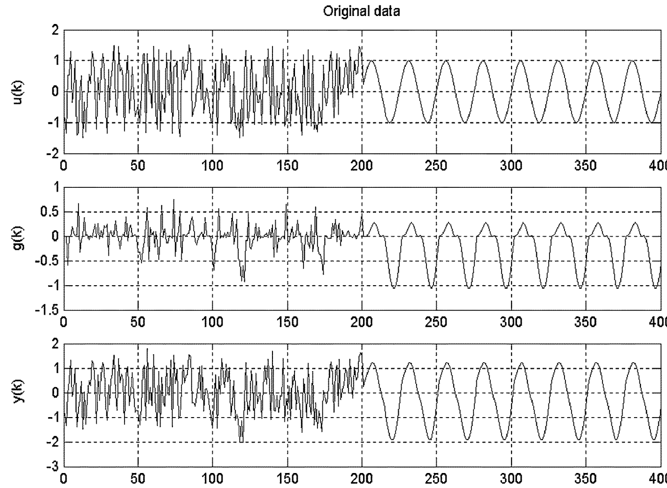


Fig. 8. Input $u(k)$, unforced system $g(k)$, and output $y(k)$ of plant in (11).

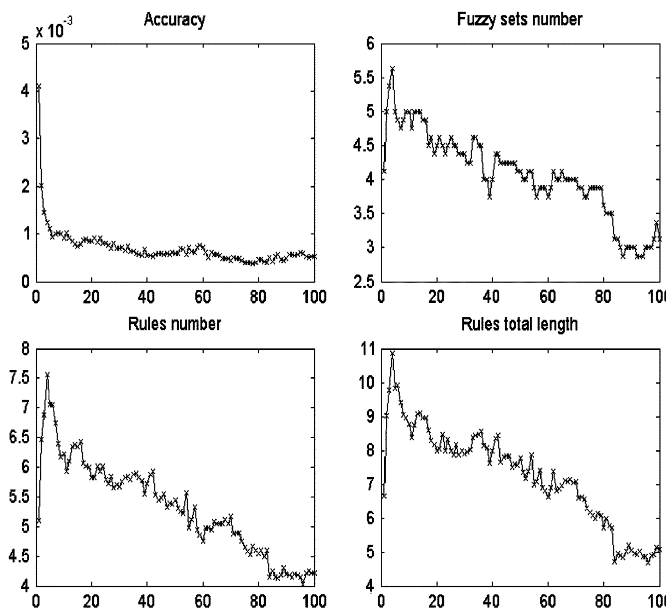


Fig. 9. Trends of average accuracy, fuzzy sets number, rules number, and rule base total length of the plant in (11).

In this agent-based approach, we use eight fuzzy set agents each of which has five fuzzy rule set solutions, so there are 40 fuzzy systems obtained. The trends plot about four criteria including average accuracy (MSE), average fuzzy sets number, average fuzzy rules number, and average fuzzy rule base total length among the multiagent system is given in Fig. 9.

Fig. 9 shows the average tendency of the four criteria in 100 generations. It can be seen that the agents are able to improve both the accuracy and interpretability of the fuzzy systems continuously due to the fact that the four criteria are optimized simultaneously in our approach. It can also be noticed that fluctuations in performance still exist, although NSGA-II is known

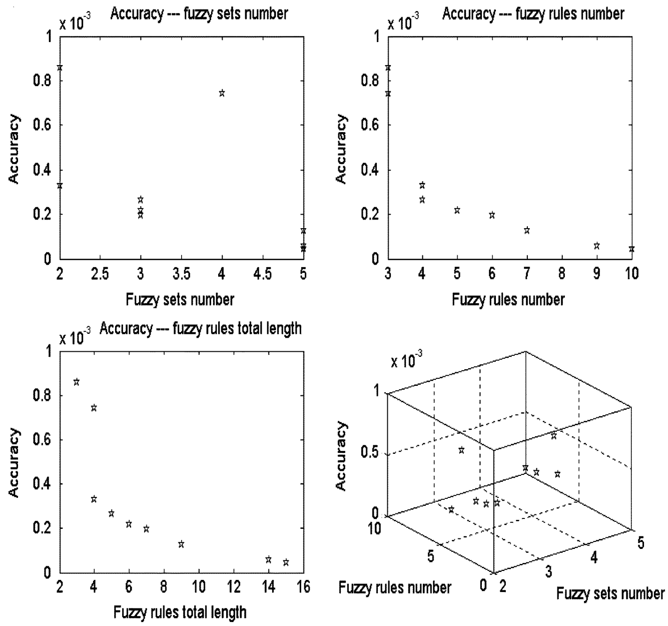


Fig. 10. Pareto front about the fuzzy systems of the plant in (11).

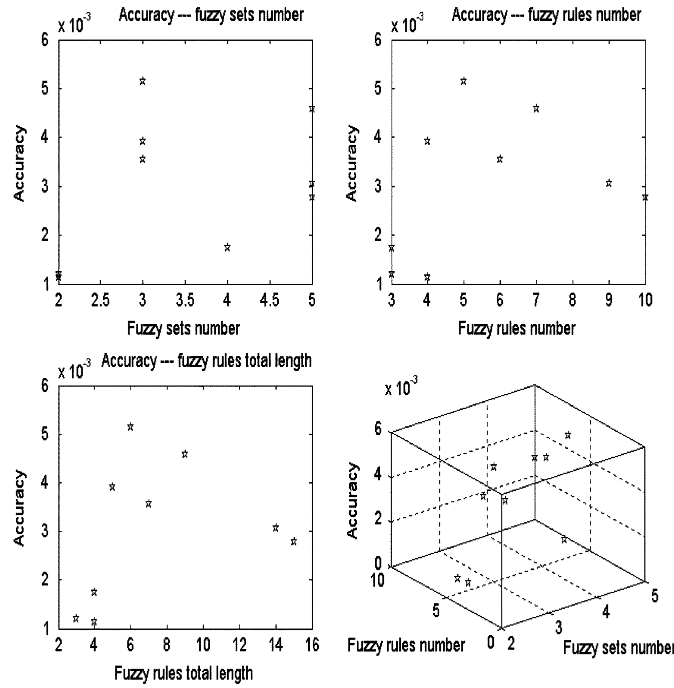


Fig. 11. Test results of the nondominated fuzzy systems of the plant in (11).

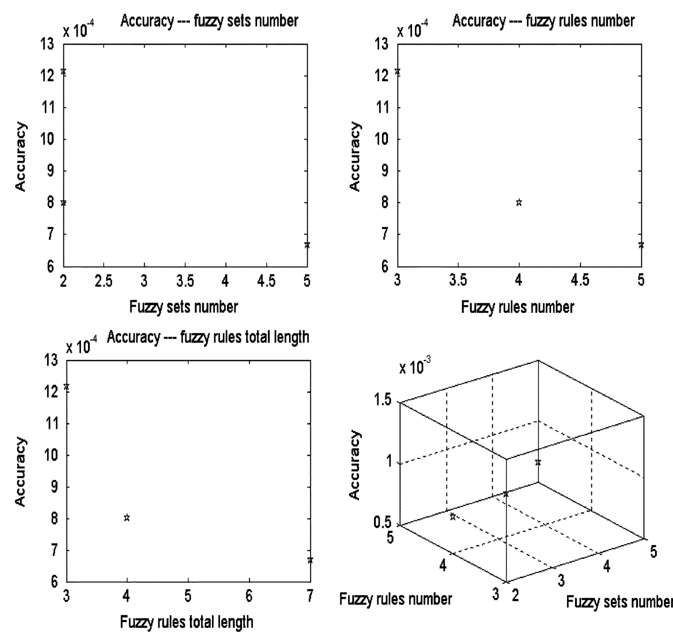


Fig. 12. Pareto front of the fuzzy systems of the plant in (11) for the test data set.

as an elitist strategy. This fluctuation is probably caused by the randomness in the NSGA-II selection mechanism.

Also we give the *Pareto front* (the plot of the objective functions whose nondominated vectors are in the Pareto optimal set is called the *Pareto front*) [35] after 100 iterations of the evolution. Fig. 10 shows the tradeoff among the multiple objectives within the nondominated fuzzy system solutions. The upper left figure illustrates the trade-off relation between the accuracy and fuzzy sets number, the upper right figure shows the tradeoff between that accuracy and fuzzy rules number, the lower left figure for the tradeoff between the accuracy and fuzzy rules total length and the lower right one shows the tradeoff among three objectives: accuracy, fuzzy sets number, as well as fuzzy rules number. There are 14 nondominated solutions out of the 40 (only nine different forms). Then, we use the 14 nondominated fuzzy system solutions to test the validation data set. Fig. 11 shows the test results. For comparison, we use all of the 40 fuzzy system solutions to test the validation data set and show the nondominated solutions based on the test accuracy and the other three criteria in Fig. 12. There are eight nondominated solutions (only three different forms) associated with the test data. In Table I, we compare our results with those of other methods in the literature.

In this example, we use the first-order TS fuzzy system (i.e., the TS fuzzy system with the linear consequents), and all of the models of the compared methods in [4], [11]–[15] are of the TS fuzzy systems. However, not all of the models have the linear form of consequents, some of them have the singleton form. We listed the consequent type in Table I. Because the first-order TS fuzzy system is applied, so the recursive least square method is very suitable to calculate the rule consequents. The training iteration number of the recursive least square method is identical to the number of training sample data (i.e., 200 in Example A) for each generation. We also list the number of generations performed by our approach and the compared methods (except the methods that do not use the evolutionary algorithm). The nondominated solutions about test data are given in Table II. Due to space limitations, we do not give the fuzzy distribution and fuzzy rules expression in this paper.

## B. Lorenz System

The Lorenz system studied in [8] is described by the following differential equations:

$$\dot{x} = -y^2 - z^2 - a(x - F) \tag{13}$$

$$\dot{y} = xy - bxz - y + G \tag{14}$$

$$\dot{z} = bxy + xz - z. \tag{15}$$

TABLE I
FUZZY MODELS OF THE NONLINEAR PLANT OF EXAMPLE A

| Ref. | No. Rules | No. Sets | Rules length | Consequent | MSE Train | MSE Test | Iteration |
|---|---|---|---|---|---|---|---|
| [13] | 40 (initial) | 40 Gauss. | 80 | Singleton | 3.2884e-4 | 6.9152e-4 | / |
| | 28 (optimized) | 28 Gauss. | 56 | Singleton | 3.3299e-4 | 5.9595e-4 | 100 |
| [15] | 25 (initial) | 25 Gauss. | 50 | Singleton | 2.3092e-4 | 4.0717e-4 | N/A |
| | 20 (optimized) | 20 Gauss. | 40 | Singleton | 6.8341e-4 | 2.3836e-4 | N/A |
| [14] | 36 (initial) | 12 B-splines | 72 | Singleton | 2.7743e-5 | 5.1163e-3 | N/A |
| | 23 (optimized) | 12 B-splines | 46 | Singleton | 3.1746e-5 | 1.4776e-3 | N/A |
| | 36 (initial) | 12 B-splines | 72 | Linear | 1.9465e-6 | 2.9211e-3 | N/A |
| | 24 (optimized) | 12 B-splines | 48 | Linear | 1.9835e-6 | 6.4120e-4 | N/A |
| [12] | 7 (initial) | 14 triangular | 14 | Singleton | 1.6e-2 | 1.2e-3 | / |
| | 7 (optimized) | 14 triangular | 14 | Singleton | 3.0e-3 | 4.9e-4 | 2000 |
| | 5 (initial) | 10 triangular | 10 | Linear | 5.8e-3 | 2.5e-3 | / |
| | 5 (optimized) | 8 triangular | 10 | Linear | 7.5e-3 | 3.5e-4 | 2000 |
| | 4 (optimized) | 4 triangular | 8 | Linear | 1.2e-3 | 4.7e-4 | 2000 |
| [11] | 5 (initial) | 10 triangular | 10 | Linear | 4.9e-3 | 2.9e-3 | / |
| | 5 (optimized) | 10 triangular | 10 | Linear | 1.4e-3 | 5.9e-4 | 600 |
| | 5 (optimized) | 5 triangular | 10 | Linear | 8.3e-4 | 3.5e-4 | 600 |
| [4] | 5 (optimized) | 5 triangular | 10 | Linear | 2.0e-3 | 1.3e-3 | N/A |
| | 5 (optimized) | 6 triangular | 10 | Linear | 5.9e-4 | 8.8e-4 | N/A |
| 1*6 | 3 | 2 Gauss2mf. | 3 | Linear | 8.5782e-4 | 1.2154e-3 | 100 |
| 2*1 | 3 | 4 Gauss2mf. | 4 | Linear | 7.4202e-4 | 1.7401e-3 | 100 |
| 3*1 | 10 | 5 Gauss2mf. | 15 | Linear | 4.5181e-5 | 2.7872e-3 | 100 |
| 4*1 | 4 | 3 Gauss2mf. | 5 | Linear | 2.6503e-4 | 3.9176e-3 | 100 |
| 5*1 | 9 | 5 Gauss2mf. | 14 | Linear | 5.6968e-5 | 3.0596e-3 | 100 |
| 6*1 | 5 | 7 Gauss2mf. | 9 | Linear | 1.2806e-4 | 4.5867e-3 | 100 |
| 7*1 | 4 | 2 Gauss2mf. | 4 | Linear | 3.3038e-4 | 1.1428e-3 | 100 |
| 8*1 | 6 | 3 Gauss2mf. | 7 | Linear | 1.9523e-4 | 3.5593e-3 | 100 |
| 9*1 | 5 | 3 Gauss2mf. | 6 | Linear | 2.1698e-4 | 5.1407e-3 | 100 |

TABLE II
NONDOMINATED FUZZY MODELS OF TEST DATA OF EXAMPLE A

| Ref. | No. Rules | No. Sets | Rules length | MSE Train | MSE Test |
|---|---|---|---|---|---|
| 1*6 | 3 | 2 Gauss2mf. | 3 | 8.5782e-4 | 1.2154e-3 |
| 2*1 | 4 | 2 Gauss2mf. | 4 | 3.3299e-4 | 8.0147e-4 |
| 3*1 | 5 | 5 Gauss2mf. | 7 | 7.0870e-4 | 6.6750e-4 |

In order to make a comparison with the results obtained in [8], 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 $z(t-1)$. Four-hundred data points are obtained from (13)–(15) using the fourth-order Runge–Kutta method with a step length of 0.05, where 200 pairs of data are used for training and the other 200 are for test. The sampling data pairs are shown in Fig. 13.

In this work, we use eight fuzzy set agents and five fuzzy rule set solutions for each agent, so there are 40 fuzzy systems. The trends plot about the same four criteria as those in Fig. 9 is given in Fig. 14. Fig. 15 shows the tradeoff among the multiple objectives within the nondominated fuzzy system solutions based on the training data. There are 17 nondominated solutions out of the forty (eight different forms).

Then, we use the 17 nondominated fuzzy system solutions to test the validation data set. Fig. 16 shows the test results. For comparison, we use all of the 40 fuzzy system solutions to test the validation data set and show the nondominated solutions based on the test accuracy and the other three criteria in Fig. 17. There are ten nondominated solutions (seven different forms) associated with the test data. In Table III, we compare the results with those of [8]. The MSE result is not given in [8], so we use "N/A" in Table III to denote such a case. The nondominated solutions about test data are given in Table IV.

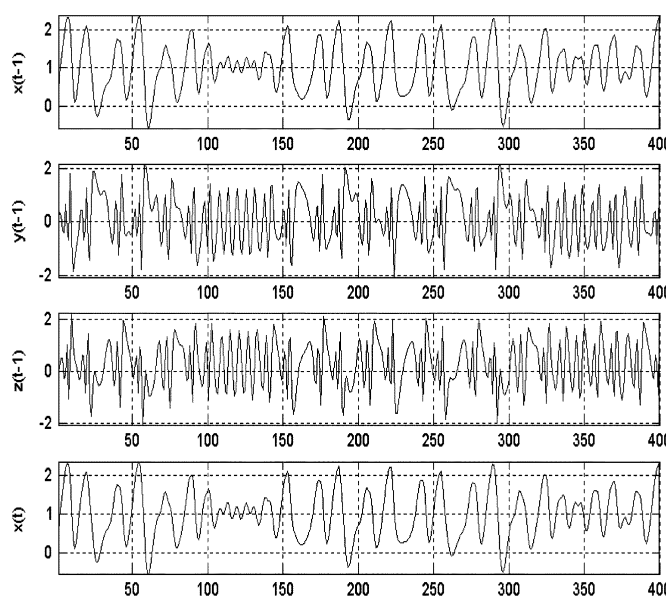


Fig. 13. Input $x(t-1), y(t-1)$, and $z(t-1)$, output $x(t)$ of the Lorenz System.

From Table III, we can see that the number of fuzzy sets of some solutions is less than the number of the input variables (i.e., three in the Lorenz system). This indicates that our agent-based evolutionary approach can use a more compact set of input variables to train the fuzzy system. To show more clearly, we add the second column named the number of input variables in Table III to illustrate such cases. The number before the brace represents the number of input variables of which the corresponding solutions make use, whereas the numbers in the brace mean the number of fuzzy sets for each input variable in

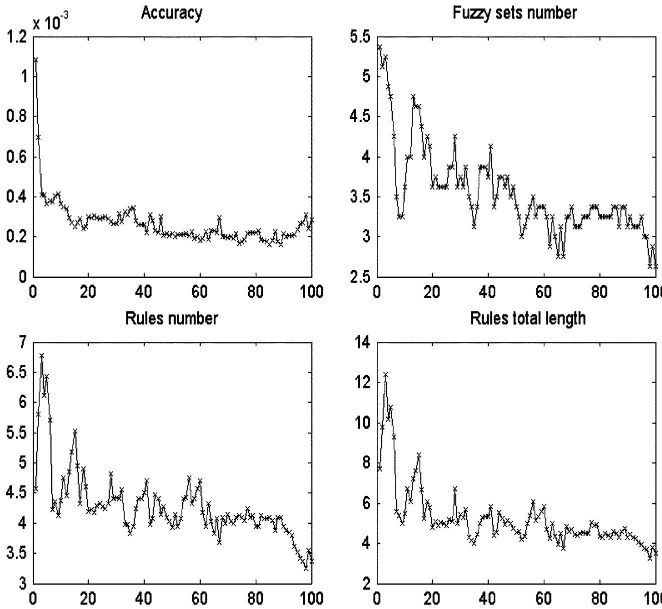


Fig. 14. Trends of average accuracy, fuzzy sets number, rules number, and rule base total length of the Lorenz System.

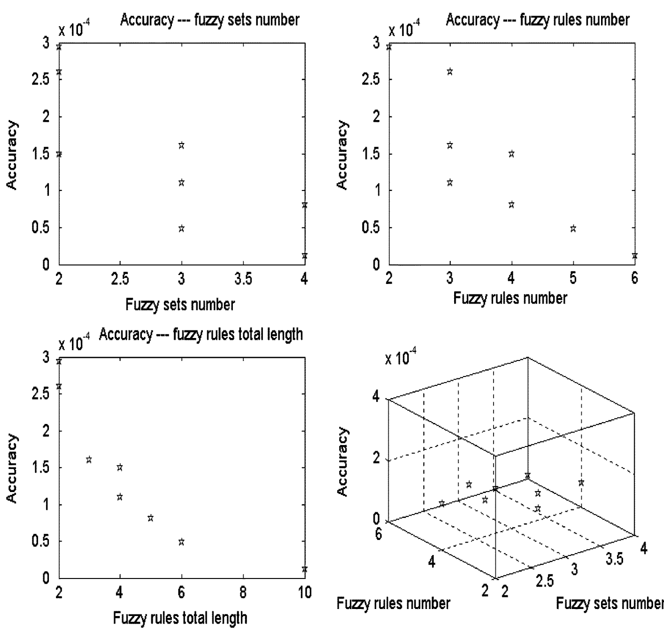


Fig. 15. Pareto front of the Lorenz System.

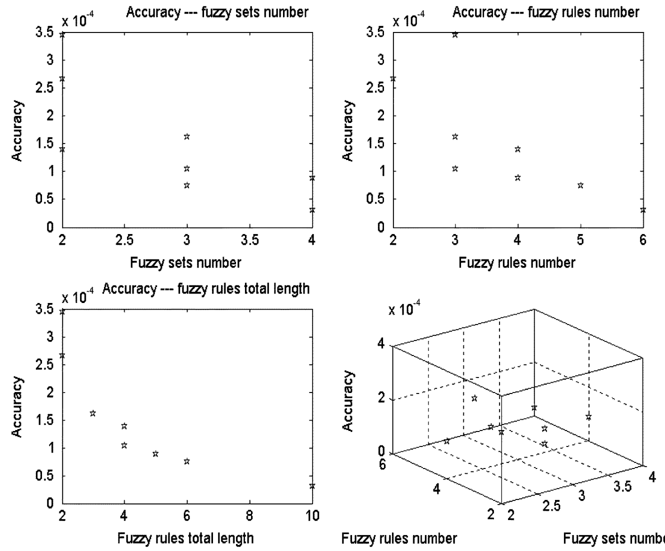


Fig. 16. Test results of the nondominated fuzzy systems of the Lorenz System.

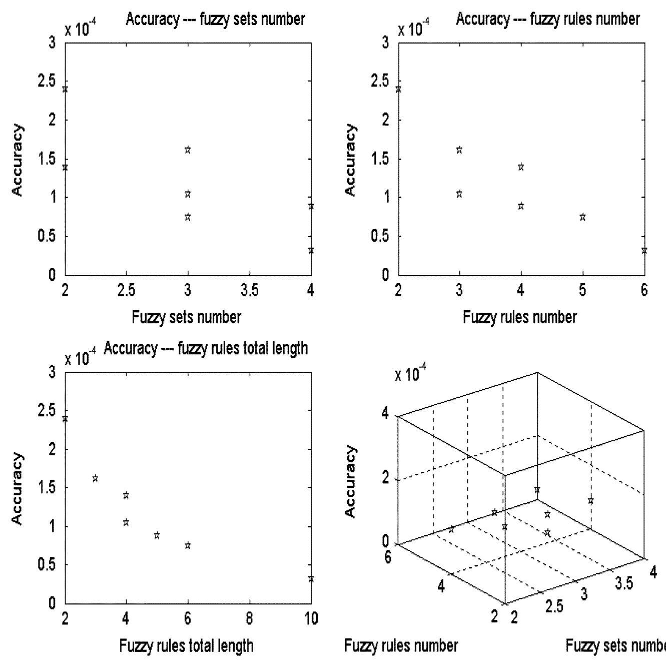


Fig. 17. Pareto front of the Lorenz System for the test data set.

TABLE III
FUZZY MODELS OF THE LORENZ SYSTEM

| Ref. | Input | No. Rules | No. Sets | Rules length | MSE Train | MSE Test |
|---|---|---|---|---|---|---|
| [8] | 3{4,2,1} | 4 | 7 | 8 | N/A | N/A |
| 1*1 | 2{0,2,2} | 6 | 4 | 10 | 1.2798e-5 | 3.1480e-5 |
| 2*1 | 2{0,2,2} | 4 | 4 | 5 | 8.1136e-5 | 8.8000e-5 |
| 3*1 | 2{0,1,2} | 5 | 3 | 6 | 4.8548e-5 | 7.4819e-5 |
| 4*1 | 2{0,1,2} | 3 | 3 | 3 | 1.6088e-4 | 1.6160e-4 |
| 5*1 | 2{0,1,2} | 3 | 3 | 4 | 1.1023e-4 | 1.0447e-4 |
| 6*2 | 2{0,1,1} | 2 | 2 | 2 | 2.9419e-4 | 2.6617e-4 |
| 7*4 | 2{0,1,1} | 4 | 2 | 4 | 1.4979e-4 | 1.3927e-4 |
| 8*6 | 2{0,1,1} | 3 | 2 | 2 | 2.6085e-4 | 3.4412e-4 |

order. For brevity, we give only one complete rule base related to the solution eight in Table III. Fig. 18 shows the distribution of fuzzy sets. The fuzzy rules are listed in Table V. From Table V, we know that $R_1$ and $R_2$ are specific rules and $R_3$ is a general rule, all of them are incomplete rules.

### C. Iris Data

The Iris Data contains 150 pattern instances with four attributes from three classes available from the University of California, Irvine (UCI) database [36]. We use all of the data to train ten fuzzy set agents each of which has eight fuzzy rule set solutions, so we can get eighty fuzzy systems. The trends plot about the four criteria showed the tradeoff among the multiple objectives within the nondominated fuzzy classifier system solutions for Iris Data. Considering the brevity of the paper, we did not give the figures in this work. There are 13 nondominated solutions out of the 80 (five different forms). Also, we compare our

TABLE IV
NONDOMINATED MODELS FOR TEST DATA OF THE LORENZ SYSTEM

| Ref. | No. Rules | No. Sets | Rules length | MSE Train | MSE Test |
|---|---|---|---|---|---|
| 1*1 | 6 | 4 Gauss2mf. | 10 | 1.2798e-5 | 3.1480e-5 |
| 2*1 | 4 | 4 Gauss2mf. | 5 | 8.1136e-5 | 8.8000e-5 |
| 3*1 | 2 | 2 Gauss2mf. | 2 | 3.0443e-4 | 2.3970e-4 |
| 4*1 | 5 | 3 Gauss2mf. | 6 | 4.8548e-5 | 7.4819e-5 |
| 5*1 | 3 | 3 Gauss2mf. | 3 | 1.6088e-4 | 1.6160e-4 |
| 6*1 | 3 | 3 Gauss2mf. | 4 | 1.1023e-4 | 1.0447e-4 |
| 7*4 | 4 | 2 Gauss2mf. | 4 | 1.4979e-4 | 1.3927e-4 |

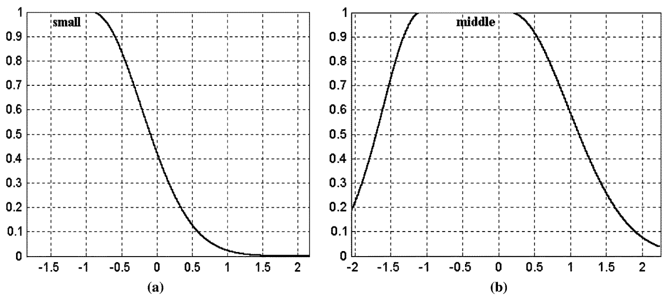


Fig. 18. Fuzzy sets of the Lorenz System. (a) Input variable 2: $y(t-1)$. (b) Input variable 3: $z(t-1)$.

TABLE V
RULE BASE FOR THE LORENZ SYSTEM

R1: If y(t-1) is small, then x(t)=0.989x(t-1)+0.227y(t-1)+0.009z(t-1)+0.018;
R2: If z(t-1) is middle, then x(t)= 0.975x(t-1)-0.119y(t-1)+0.049z(t-1)+0.163;
R3: Else x(t)= 0.963x(t-1)+0.071y(t-1)+0.009z(t-1)-0.210.

TABLE VI
COMPARISON RESULTS FOR IRIS DATA

| Ref. | Classification Rate | Input | No. Sets | No. Rules | Rules length |
|---|---|---|---|---|---|
| [11] | 0.973 | 2{0,0,3,2} | 5 | 3 | 6 |
| [2] | 0.993 | 4{3,3,3,3} | 12 | 3 | 12 |
| [20] | 0.973 | 4{5,5,5,5} | 20 | 4.6 | N/A |
| [37] | 1.000 | 4{4,5,4,5} | 18 | 5 | 18 |
| [38] | 0.993 | 4{3,3,3,3} | 12 | 6 | 24 |
| 1*8 | 0.980 | 3{0,1,1,2} | 4 | 4 | 5 |
| 2*1 | 0.987 | 3{0,1,1,2} | 4 | 5 | 6 |
| 3*2 | 0.987 | 3{0,1,1,3} | 5 | 4 | 6 |
| 4*1 | 0.973 | 3{0,1,1,3} | 5 | 3 | 8 |
| 5*1 | 0.973 | 3{0,1,1,2} | 4 | 4 | 4 |

results with other works. The comparison results are shown in Table VI. We also noticed that we can use only three attributes instead of four to train fuzzy classifier systems resulting in an improvement of interpretability associated with compactness.

From Figs. 9 and 14, we can see that our agent-based approach can guarantee good convergence among the multiple objectives. The objectives considering both the accuracy and interpretability can co-evolve. Another advantage of our approach is that we can obtain multiple nondominated fuzzy systems concentrating on both the accuracy and interpretability. It is obviously illustrated in Figs. 10 and 15 and quantified in Tables I, III, and VI. From these tables, we also demonstrate that the accuracy of our results is comparable to or better than other methods known in the literature, and more important, most solutions we get have better interpretability. The tradeoff between accuracy and interpretability of fuzzy systems is also easily understood.

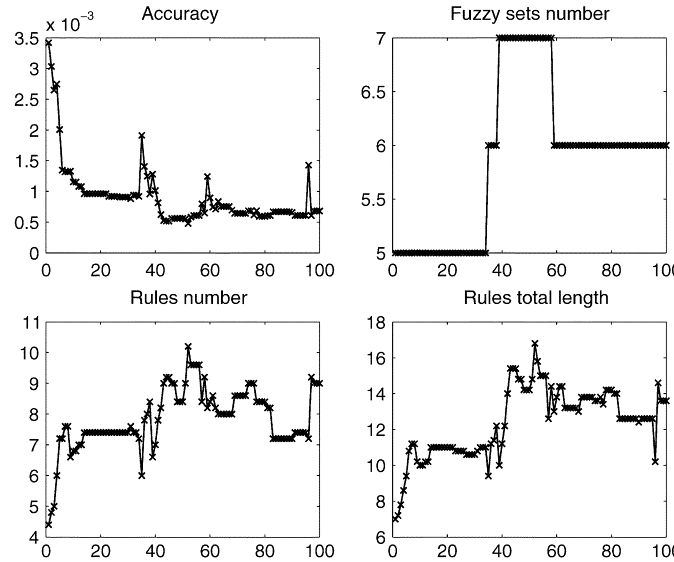


Fig. 19. Example A using one agent.

Different sets of fuzzy rules and fuzzy sets that emphasize different aspects of interpretability and accuracy may be built. In this work, the number of fuzzy variables can be automatically learned, for example, only two out of three input variables participate in the fuzzy system construction for the Lorenz system, and only three out of four attributes play roles in the fuzzy classifier system construction for the Iris Data. This leads to more compactness not only associated with the number of fuzzy sets but also related to the number of fuzzy variables. We are inspired by this aspect that more important variables can be determined by the proposed approach and rule-based systems can be built based on these important variables only. The irrelevant variables are removed from the system construction. Thus, the complexity of rule base construction is reduced greatly, especially for the high-dimensional problems. We hope that it will work in the real-world nonlinear plant modeling and classification problems, and so on. It will be worth paying much attention to in future research.

In order to show the effectiveness of the multiagent approach, we did experiments based on only one agent (i.e., just use one agent to learn the fuzzy rule base without changing the other parameters). Due to space limitations, we only give the trends plot of example $A$ and $B$ in Figs. 19 and 20. We compare the trends plot with those of the multiagent approach (Figs. 9 and 14). We can see that the average classification rate, fuzzy sets number, fuzzy rules number, and fuzzy rules total length using multiple agents are better than those of the single agent approach and have a better convergence. It means that the multiagent approach has good abilities to explore interpretable rule base with the accuracy consideration based on the obtained fuzzy sets. In the multiagent system, the fuzzy sets number can reduce gradually with the co-evolution of the other three criteria. The interpretability improves related to the compactness issue. When multiple agents participate in the evolutionary process, they can cooperate and compete with each other to exchange the fuzzy sets information to obtain compact fuzzy systems. It is also a

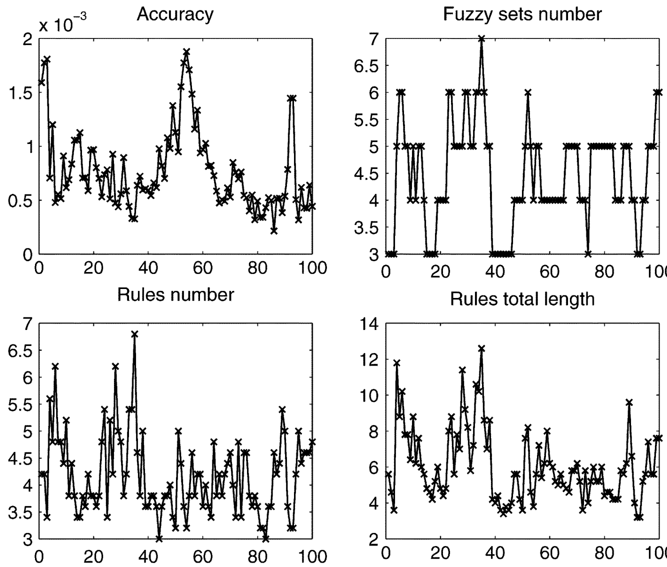


Fig. 20. Example B using one agent.

main goal for us to design such a multiagent mechanism: to explore more appropriate fuzzy sets distribution and use a smaller number of fuzzy sets.

## V. Conclusion and Future Works

In this paper, we proposed an agent-based evolutionary approach to construct interpretable fuzzy systems. In the multiagent system, the FSAs autonomously implement the following intratasks: i) Use the hierarchical chromosome formulation and interpretability-based regulation strategy to obtain compact and distinguishable fuzzy sets distribution, and ii) apply the Pittsburgh-style approach based on the obtained fuzzy sets to extract interpretable fuzzy rules by means of NSGA-II multiobjective decision-making method and the recursive least square method for function approximation problems as well as the heuristic procedure for classification problems. Then, the fuzzy set agents cooperate with each other by exchanging fuzzy sets information to create offspring agents. The arbitrator agent evaluates the parent and offspring agents based on the criteria of accuracy and interpretability. During competition, the elite agents survive to the next population and obsolete ones are dead. Simulation results show that our proposed approach can generate multiple fuzzy systems with a good tradeoff between the accuracy and interpretability. In future research, we will concentrate ourselves on the following issues to improve the performance of our agent-based evolutionary approach: 1) further studying the interaction mechanism among the agents to realize a more effective manner associated with cooperation and competition and 2) applying some data mining techniques related to dimension reduction such as SUD [39], RELIEF, and SCM [40], etc. to our multiagent system, hopefully using more important attributes to train the agents leading to a more compact fuzzy system construction.

## Acknowledgment

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

## References

[1] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, pp. 28–44, Jan. 1973.
[2] G. Castellano, A. M. Fanelli, E. Gentile, and T. Roselli, "A GA-based approach to optimization of fuzzy models learned from data," in *Proc. Genetic Evolutionary Computation Conf.*, Jul. 2002, pp. 5–8.
[3] S. Guillaume, "Designing fuzzy inference systems from data: An interpretability oriented review," *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 3, pp. 426–443, Jun. 2001.
[4] F. Jiménez, A. F. Gómez-Skarmeta, H. Roubos, and R. Babuška, "Accurate, transparent, and compact fuzzy models for function approximation and dynamic modeling through multiobjective evolutionary optimization," in *Proc. Int. Conf. Evolutionary Multi-criterion Optimization*, Mar. 2001, pp. 653–667.
[5] Y. Jin, "Fuzzy modeling of high-dimensional systems: Complexity reduction and interpretability improvement," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 2, pp. 212–221, Apr. 2000.
[6] Y. Jin, W. von Seelen, and B. Sendhoff, "An approach to rule-based knowledge extraction," in *Proc. IEEE Conf. Fuzzy Syst.*, vol. 2, May 1998, pp. 1188–1193.
[7] ——, "On generating $FC^3$ fuzzy rule systems from data using evolution strategies," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 29, no. 6, pp. 829–845, Dec. 1999.
[8] Y. Jin and B. Sendhoff, "Extracting interpretable fuzzy rules from RBF networks," *Neural Processing Lett.*, vol. 17, no. 2, pp. 149–164, 2003.
[9] T. Murata, S. Kawakami, H. Nozawa, M. Gen, and H. Ishibuchi, "Three-objective genetic algorithms for designing compact fuzzy rule-based systems for pattern classification problems," in *Proc. Genetic Evolutionary Computation Conf.*, Jul. 2001, pp. 485–492.
[10] I. Rojas, H. Pomares, J. Ortega, and A. Prieto, "Self-organized fuzzy system generation from training examples," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 1, pp. 23–36, Feb. 2000.
[11] H. Roubos and M. Setnes, "Compact and transparent fuzzy models and classifiers through iterative complexity reduction," *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 4, pp. 516–524, Aug. 2001.
[12] ——, "GA-fuzzy modeling and classification: Complexity and performance," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 5, pp. 509–522, Oct. 2000.
[13] L. Wang and J. Yen, "Exacting fuzzy rules for system modeling using a hybrid of genetic algorithms and Kalman filter," *Fuzzy Sets Syst.*, vol. 101, pp. 353–362, 1999.
[14] J. Yen and L. Wang, "Application of statistical information criteria for optimal fuzzy model construction," *IEEE Trans. Fuzzy Syst.*, vol. 6, no. 3, pp. 362–372, Aug. 1998.
[15] ——, "Simplifying fuzzy rule-based models using orthogonal transformation methods," *IEEE Trans. Syst., Man, Cybern. B*, vol. 29, no. 1, pp. 13–24, Feb. 1999.
[16] J. Casillas, O. Cordón, F. Herrera, and L. Magdalena, *Interpretability Issues in Fuzzy Modeling*. Berlin, Germany: Springer, 2003.
[17] C. F. Juang and C. T. Lin, "An on-line self-constructing neural fuzzy inference network and its applications," *IEEE Trans. Fuzzy Syst.*, vol. 6, no. 1, pp. 12–32, Feb. 1998.
[18] C. T. Lin, "A neural fuzzy control system with structure and parameter learning," *Fuzzy Sets Syst.*, vol. 70, pp. 183–212, 1995.
[19] C. T. Lin and C. S. G. Lee, "Neural-network-based fuzzy logic control and decision system," *IEEE Trans. Comput.*, vol. 40, no. 12, pp. 1320–1336, Dec. 1991.
[20] H. Ishibuchi, T. Nakashima, and T. Kuroda, "A hybrid fuzzy GBML algorithm for designing compact fuzzy rule-based classification systems," in *Proc. 9th IEEE Int. Conf. Fuzzy Syst.*, May 2000, pp. 706–711.
[21] H. Ishibuchi, T. Nakashima, and T. Kuroda, "A hybrid fuzzy genetics-based machine learning algorithm: Hybridization of Michigan approach and Pittsburgh approach," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 1999, pp. 296–301.
[22] N. Xiong, "Evolutionary learning of rule premises for fuzzy modeling," *Int. J. Syst. Sci.*, vol. 32, no. 9, pp. 1109–1118, 2001.
[23] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[24] M. Setnes, R. Babuška, U. Kaymak, and H. R. van Nauta Lemke, "Similarity measures in fuzzy rule base simplification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 28, no. 3, pp. 376–386, Jun. 1998.
[25] H. Ishibuchi and T. Nakashima, "Three-objective optimization in linguistic function approximation," in *Proc. Evol. Comput.*, May 2001, pp. 340–347.

[26] K. F. Man, K. S. Tang, and S. Kwong, "Genetic algorithm: Concepts and applications," *IEEE Trans. Ind. Electron.*, vol. 43, no. 5, pp. 519–534, Oct. 1996.

[27] K. S. Tang, K. F. Man, S. Kwong, and Q. He, "Genetic algorithms and their applications," *IEEE Signal Process. Mag.*, vol. 13, no. 6, pp. 22–37, Nov. 1996.

[28] K. S. Tang, K. F. Man, Z. F. Liu, and S. Kwong, "Minimal fuzzy memberships and rules using hierarchical genetic algorithms," *IEEE Trans. Ind. Electron.*, vol. 45, no. 1, pp. 162–169, Feb. 1998.

[29] K. M. Passino and S. Yurkovich, *Fuzzy Control*. Reading, MA: Addison-Wesley , 1998, ch. 5.

[30] H. Ishibuchi and T. Nakashima, "Effect of rule weights in fuzzy rule-based classification systems," *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 4, pp. 506–515, Aug. 2001.

[31] H. Ishibuchi, T. Nakashima, and T. Murata, "Multiobjective optimization in linguistic rule extraction from numerical data," in *Proc. Int. Conf. Evolutionary Multi-criterion Optimization*, Mar. 2001, pp. 588–602.

[32] ——, "Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 29, no. 5, pp. 601–618, Oct. 1999.

[33] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 1, pp. 116–132, Jan. 1985.

[34] H. Ishibuchi, K. Nozaki, and H. Tanaka, "Distributed representation of fuzzy rules and its application to pattern classification," *Fuzzy Sets Syst.*, vol. 52, no. 1, pp. 21–32, Nov. 1992.

[35] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, New York: Wiley, 2001, pp. 28–46.

[36] UCI Machine Learning Repository [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html

[37] M. Russo, "Genetic fuzzy learning," *IEEE Trans. Evol. Comput.*, vol. 4, no. 3, pp. 259–273, Sep. 2000.

[38] N. Xiong and L. Litz, "Identifying flexible structured premises for mining concise fuzzy knowledge," in *Interpretability Issues in Fuzzy Modeling*, J. Casillas *et al.*, Eds. Berlin, Germany: Springer, 2003, pp. 54–76.

[39] M. Dash, H. Liu, and J. Yao, "Dimensionality reduction for unsupervised data," in *Proc. 9th Int. Conf. Tools Artif. Intell.*, Nov. 1997, pp. 532–539.

[40] X. Fu and L. Wang, "Data dimensionality reduction with application to simplifying RBF network structure and improving classification performance," *IEEE Trans. Syst., Man, Cybern. B*, vol. 33, no. 3, pp. 399–409, Jun. 2003.

**Hanli Wang** received the B.S. and M.S. degrees from the College of Electrical Engineering, Zhejiang University, Hong Kong, in 2001 and 2004, respectively.

From 2003 to 2004, he was a Research Assistant in the Department of Computer Science, City University of Hong Kong. His current research interests include evolutionary computation, fuzzy systems, image processing, and digital watermarking.

**Sam Kwong** received the B.Sc. and M.A.Sc. degrees in electrical engineering from the State University of New York at Buffalo and the University of Waterloo, Waterloo, ON, Canada, respectively, and the Ph.D. degree from the University of Hagen, Hagen, Germany.

Currently, he is an Associate Professor in the Department of Computer Science, City University of Hong Kong. In 1990, he joined the City University of Hong Kong as a Lecturer in the Department of Electronic Engineering. His research interests include evolutionary algorithms, signal processing, and intelligent computing.

**Yaochu Jin** (M'98–SM'02) received the B.Sc. and M.Sc. degrees from Zhejiang University, Hangzhou, China, in 1988 and 1991, respectively, the Ph.D. degree from Zhejiang University in 1996, and the Ph.D. degree from Ruhr-Universität Bochum, Bochum, Germany, in 2001.

Currently, he is a Principal Scientist at Honda Research Institute Europe, Offenbach/Main, Germany. In 1991, he joined the Electrical Engineering Department, Zhejiang University, where he became an Associate Professor in 1996. From 1996 to 1998, he was with the Institut für Neuroinformatik, Ruhr-Universität Bochum, first as a Visiting Scholar and then as a Researcher. He was a Postdoctoral Associate with the Industrial Engineering Department, Rutgers University, Piscataway, NJ, from 1998 to 1999. In 1999, he joined Future Technology Research, Honda R&D Europe. His research interests are in neural and evolutionary fuzzy systems, knowledge extraction from and knowledge incorporation into neural and evolutionary systems, evolutionary computation in dynamic and uncertain environments, multiobjective evolutionary algorithms, and their application to systems modeling and control, design optimization, robotics and the self-organization of distributed systems. He is the author of the book entitled *Advanced Fuzzy Systems Design and Applications* (Heidelberg, Germany: Springer, 2003), the Editor of the book titled *Knowledge Incorporation in Evolutionary Computation* (Berlin, Germany: Springer, 2004) and many journal and conference papers.

Dr. Jin is currently an Associate Editor of the IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY. He has been a Guest Editor of the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS and *Journal of Soft Computing*. He serves as the Chair of the Working Group on "Evolutionary Computation in Dynamic and Uncertain Environments" within the Evolutionary Computation Technical Committee of the IEEE Computational Intelligence Society. He has been a Co-Chair or a member of program committee of several international workshops and conferences. He is a member of Council of Authors of the International Society for Genetic and Evolutionary Computation. He received *the Science and Progress Award* from the Ministry of Education of China in 1995.

**Wei Wei** received the B.S., M.S., and Ph.D. degrees in electrical engineering from Zhejiang University, Hangzhou, China, in 1983, 1986, and 1994, respectively.

Currently, he is with Zhejiang University, where he has been a Professor of automation since 1999. He joined Zhejiang University in 1986. From 1993 to 1994, he was a Visiting Scholar in the Department of Engineering, University of Reading, Reading, U.K. His research interests include intelligent control, robot, and computer vision.

**Kim-Fung Man** received the Ph.D. degree from Cranfield Institute of Technology, Bedford, U.K., in 1983.

Currently, he is a Professor in the Electronic Engineering Department, City University of Hong Kong. He is the Editor-In-Chief of *Real-Time Systems Journal*. His research interests include evolutionary computing, chaos, and control.

Dr. Man is an editorial member of IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, IEEE TRANSACTIONS ON MOBILE COMPUTING, and *Advances in Industrial Control Series*, Springer Verlag, London, U.K.