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

Hanli Wang[1,3], Sam Kwong[1], Yaochu Jin[2], Wei Wei[3], and K. F. Man[4]

[1]Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong, PRC
[2]Honda Research Institute Europe, 67073, Offenbach/Main, Germany
[3]College of Electrical Engineering, Zhejiang University, Hangzhou 310027, PRC
[4]Department of Electronic Engineering, City University of Hong Kong

Abstract—An agent-based evolutionary approach is proposed to extract interpretable rule-based knowledge. In the multi-agent 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 considerations. 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 trade-off between the accuracy and interpretability of fuzzy systems.

Index Terms—Multi-agent system, interpretability and accuracy, hierarchical chromosome formulation, multi-objective decision making

I. INTRODUCTION

The fundamental concept of fuzzy reasoning was first introduced by Zadeh [40] 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 to 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 trade-off 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 [2,7,15-19,25,27-29,35,38,39]. And the book [1] edited by Casillas et al. 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

*This work is supported by City University of Hong Kong Strategic Grant 7001416

fuzzy systems and the trade-off between the accuracy and the interpretability of fuzzy systems autonomously generated from the learning data. So this is a multi-objective optimization problem by its very nature. And the multi-objective evolutionary algorithm is very suitable to solve this problem. In the multi-objective 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 trade-off between the accuracy and the 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 [20-22]. 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 constructing fuzzy systems from training data with the 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 [10,11,36]. In [36] the author used some important end points to distribute membership functions. The number of fuzzy sets is fixed and there are some limitations about the distribution of these fuzzy sets. In [10,11], the fuzzy sets are pre-partitioned without considering more appropriate distributions. More importantly, it is almost impossible to have a good understanding about an unknown complex system, even not to mention giving the linguistic values for each fuzzy variable in advance. In this work, we suggest an agent-based scheme. In this multi-agent system, each agent has the autonomous capability to determine the number 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 in 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 multi-objective 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 in the multi-agent system, whereas the obsolete agents are dead through the arbitrator agent.

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 automatically learning from data should not be limited to find the best approximation of data only, but also and 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:

Definition II-1. Completeness: For each input variable x_i (an element of the input vector X=$[x_1, x_2, \cdots, x_n]^T$), there exists M_i fuzzy sets represented by $A_1(x), A_2(x), \cdots, A_{M_i}(x)$. Then the partition of the fuzzy sets is complete if the following conditions are satisfied:

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

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 [2,7,18,31]. This similarity measure can be interpreted in many different ways depending on the application context. However, an important definition is given in [31]: Similarity between fuzzy sets is defined 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 $U(u_U(x) = 1, \forall x \in X)$; 3) similarity of a fuzzy set to a singleton set. 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).

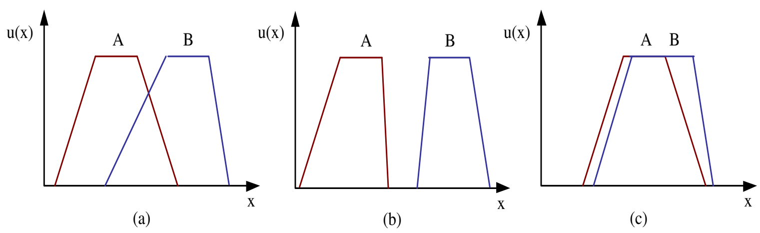


Fig. 1: Fuzzy partitioning. (a) Overlap moderately. (b) Do not overlap. (c) Overlap too much.

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 [31]:

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

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 [28,29] is described as:

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

on a discrete universe $U = \{x_j | j = 1, 2, \cdots, m\}$. $\wedge$ and $\vee$ in Equation (3) are the minimum and maximum operators, respectively. 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 a prior knowledge. Consistency among fuzzy rules means that if two or more rules are simultaneously fired, then their conclusions should be coherent [7], i.e., if two or more rules have the similar antecedents, their consequents should also be similar. The consistency with a prior 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 [18]. 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:

R_1*: If* x_1 *is small and* x_2 *is small and* x_3 *is big, then* y *is big*
R_2*: If* x_1 *is small and* x_2 *is small, then* y *is middle*
R_3*: If* x_1 *is small, then* y *is small* (4)

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 middle,*
Else if x_1 *is small, then* y *is small.* (5)

In [9], it is called inclusion relation. If two fuzzy rules are compatible with an input vector and one rule is include 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* $\cdots$ x_n *is* $A_{in}(x_n)$*, then* y_1 *is* $B_{i1}(y_1)$ *and* $\cdots$ y_m *is* $B_{im}(y_m)$
R_j*: If* x_1 *is* $A_{j1}(x_1)$ *and* $\cdots$ x_n *is* $A_{jn}(x_n)$*, then* y_1 *is* $B_{j1}(y_1)$ *and* $\cdots$ y_m *is* $B_{jm}(y_m)$ (6)

When the inclusion relation $A_{jk} \subseteq A_{ik}$ holds for all the input variables (i.e., for $k = 1, 2, 3, \cdots, n$), 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 *ith* rule is defined as follows:

$$u_i(x) = u_{A_{i1}}(x_1) \wedge u_{A_{i2}}(x_2) \wedge \cdots \wedge u_{A_{in}}(x_n), i = 1, 2, \cdots, R \tag{7}$$

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

$$\lambda_i = \prod_{R_k \subseteq R_i} (1 - u_k(x)), k = 1, 2, \cdots, R, k \neq i \tag{8}$$

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

$$\hat{u}_i = \lambda_i u_i(x), i = 1, 2, \cdots, R \tag{9}$$

Let us illustrate the above ideas by taking the rules R_1, R_2, and R_3 ($R_1 \subseteq R_2 \subseteq R_3$) in (4) for example. Assuming the original fire-strengths of R_1, R_2, and R_3 are u_1, u_2 and u_3, then the inclusion relation factors of the three rules in order are λ_1,λ_2, and λ_3, respectively:

$$\lambda_1 = 1, \lambda_2 = 1 - u_1, \lambda_3 = (1 - u_1)(1 - u_2) \tag{10}$$

The final fire-strengths of rules R_1, R_2, and R_3 are updated as:

$$\hat{u}_1 = \lambda_1 u_1 = u_1 \tag{11}$$

$$\hat{u}_2 = \lambda_2 u_2 = (1 - u_1)u_2 \tag{12}$$

$$\hat{u}_3 = \lambda_3 u_3 = (1 - u_1)(1 - u_2)u_3 \tag{13}$$

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.

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.

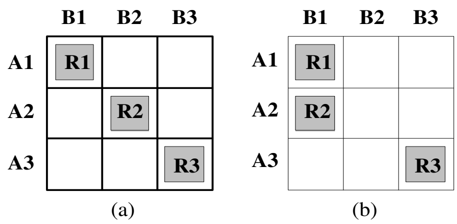


Fig. 2: A fuzzy system with two input variables (three fuzzy sets for each variable) and three rules. (a) Sufficient utility. (b) Insufficient utility because fuzzy set B2 is not utilized by any rules.

III. AGENT-BASED EVOLUTIONARY COMPUTATION APPROACH

In this paper, we propose an agent-based evolutionary computation approach to constructing fuzzy models with considerations of both the accuracy and interpretability. The basic modeling ideas are illustrated in Figure 3. There are two kinds of agents in the multi-agent system: the Arbitrator Agent (AA) and the Fuzzy Set Agent (FSA). These Fuzzy Set Agents are distributed independently and obtain information from the Arbitrator Agent in which the information is expressed in terms of training data in our specified research context. We name the agent as Fuzzy Set Agent 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 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 fuzzy sets agents. After the self-evolving of the FSAs, they send their fitness information in the form of accuracy and interpretability to the Arbitrator Agent. In the current work, the Arbitrator Agent use the NSGA-II algorithm to evaluate the fuzzy set agents, and judge which fuzzy sets 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 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.

1. The autonomous Fuzzy Set Agents' intra behavior

In the multi-agent system, the Fuzzy Set Agents employ the fuzzy sets distribution strategy, 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.

A. Fuzzy sets distribution strategy

Inspired by the insight of biological DNA structure, a hierarchical chromosome formulation for GA is introduced in [23,24,33], 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. Figure 4 illustrates the concept further.

For each fuzzy variable x_i (i.e., input variable or pattern attribute), 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 or linguistic values. 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 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 σ_1, c_1, σ_2 and c_2 as given by:

$$f(x;\sigma_1,c_1,\sigma_2,c_2)=\left\{\begin{array}{rcl}\exp\left[\frac{-(x-c_1)^2}{2\sigma_1^2}\right] & : & x<c_1\\ 1 & : & c_1\le x\le c_2\\ \exp\left[\frac{-(x-c_2)^2}{2\sigma_2^2}\right] & : & c_2<x\end{array}\right\} \tag{14}$$

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. So we use the parameter list $[\sigma_1,c_1,\sigma_2,c_2]$ to represent one parameter

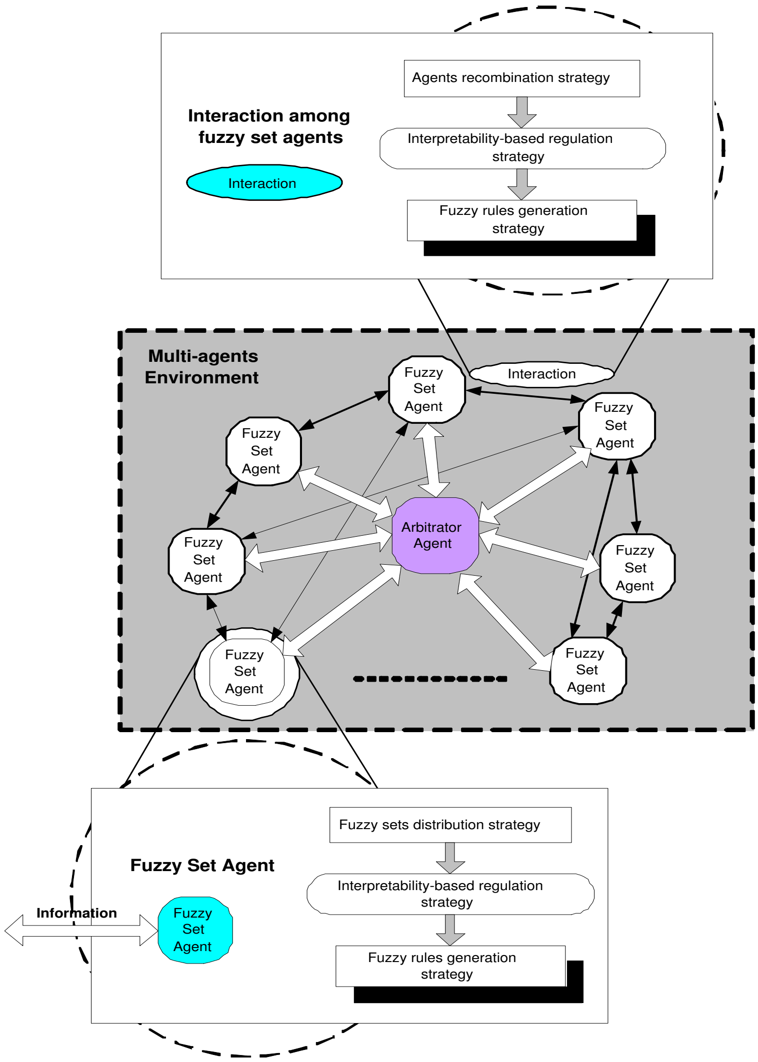


Fig. 3: Multi-agents system framework

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 Figure 5. The Fuzzy Set Agent initializes its own control genes and parameter genes randomly.

B. Interpretability-based regulation strategy

Although the Fuzzy Set Agent initializes the fuzzy sets, the interpretability issues such as distinguishability of fuzzy partitions 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, and 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

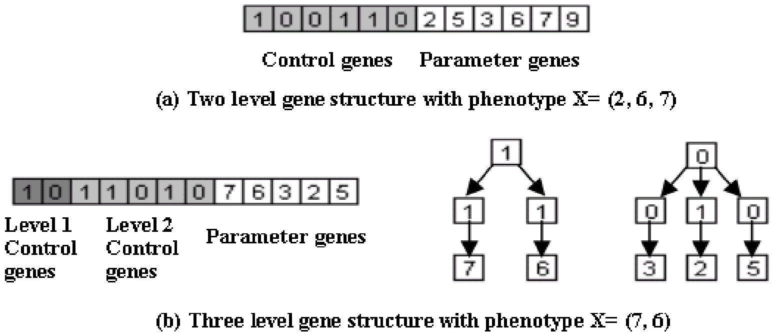


Fig. 4: Example of hierarchical chromosome representation

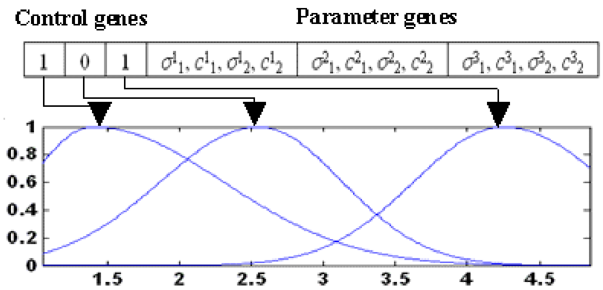


Fig. 5: An example of Hierarchical formulation

a_4 just like 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 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.

B.1 Merging similar fuzzy sets

An example of the similarity measure between two fuzzy sets is given as in Equation (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:

$$c_1 = min(a_1, b_1) \tag{15}$$

$$c_2 = \lambda_2 a_2 + (1 - \lambda_2) b_2 \tag{16}$$

$$c_3 = \lambda_3 a_3 + (1 - \lambda_3) b_3 \tag{17}$$

$$c_4 = max(a_4, b_4) \tag{18}$$

The parameters λ_2,$\lambda_3 \in [0, 1]$ determines the relative importance about the influence 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. Figure 6 illustrates the case for merging A and B to create C.

B.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

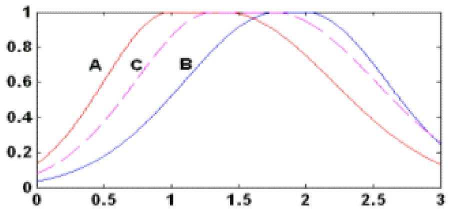


Fig. 6: Merging A and B to create C

than a upper threshold (θ_U) or smaller than a lower threshold (θ_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 of these cases 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 Fuzzy Set Agent obtains a fuzzy system with $M_1^a + M_2^a + \cdots + M_N^a$ sets, where $0 \leq M_i^a \leq M_i$ and the case that M_i^a is equal to 0 indicates that the corresponding fuzzy variable does not be involved in the modeling of fuzzy systems resulting in the dimensionality reduction by one.

C. Fuzzy rules generation strategy

In the stage of fuzzy rules generation, Fuzzy Set Agents 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 $(M_1^a + 1) \times (M_2^a + 1) \times \cdots \times (M_N^a + 1)$ for N-dimensional problems. The task of Fuzzy Set Agents 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 Fuzzy Set Agents achieve these goals.

C.1 Initialization of the rules population

In the Pittsburgh-style genetic based machine learning approach, the search of 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 N-length array in the computer program, and the *ith* element of the array indicates which fuzzy set of the *ith* fuzzy variable is fired. The *ith* 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 *ith* fuzzy variable is fired, whereas if c_i is equal to zero, this means that the *ith* fuzzy variable does not play a role in the rule generation. As far as the *ith* 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 *ith* 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 *ith* fuzzy variable. Then the Fuzzy Set Agent sets the population size N_{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 sets population, it is represented as a concatenated string of the length $N \times N_{rule}$, where N_{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 [26] and the heuristic procedure in [8,10,12,13] 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.

C.2 Crossover and Mutation

Offspring rule sets are generated by crossover and mutation. As far as the crossover is concerned, one-point crossover is used (seen in Figure 7). The crossover operation randomly selects a different cutoff point for each parent to generate offspring rule sets. A mutation operation randomly replaces each element of the rule sets string with another linguistic 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 Fuzzy Set Agent will check the offspring fuzzy rule base to delete the same rules and maintain single among all the rules after the crossover and mutation operations, so the consistency of fuzzy systems is guaranteed.

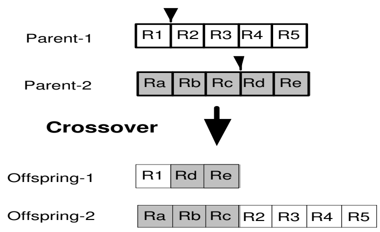


Fig. 7: Crossover operation

C.3 Evaluation criteria and selection mechanism

The Fuzzy Set Agent uses the following three criteria to evaluate fuzzy rule set candidates: 1) Accuracy: the accuracy is measured in terms of Mean-Square-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 ([25]): 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 [32] is generated. The Takagi-Sugeno fuzzy system is very suitable for the approximation of dynamic systems. And the first-order TS fuzzy system is very common and effective. And 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 so 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 Equation (9) to infer the conclusion.

As far as the classification problems, the heuristic procedure is applied to generate rule consequents from the training pattern data. For each n-dimension training pattern data $X^i = [x_1^i, x_2^i, \cdots, x_n^i]$, the fire-strength of rule R_i considering the inclusion relation is calculated using Equation (9). Then for each of the c classes, the sum of the fire-strength related to rule R_i is calculated as:

$$\beta_{Classj}(R_i) = \sum_{X^k \in Classj} \hat{u}_i(X^k), j = 1, 2, \cdots, c \tag{19}$$

Find the class C_i as the consequent of rule R_i which has the maximum value of β_{Classj}. If the maximum value of β_{Classj} can not be uniquely specified, that is, there are some classes that have 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 [14]. For each training pattern data X^i, the winner rule R_i rule is determined as:

$$\hat{u}_i(X^i) = max\{\hat{u}_k(X^i) \mid k = 1, 2, \cdots, R\} \tag{20}$$

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 by one.

Based on the foregoing three criteria, the Fuzzy Set Agent uses the NSGA-II [5] algorithm to evaluate the fuzzy rule sets candidates. In order to compare the fuzzy rule base 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 firstly compare two fuzzy rule base 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 base candidate is better. If one rule base candidate is better than the other based on the accuracy preference, then it 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 base 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. If we do not pay more attention to the accuracy, the NSGA-II in the long run maybe generate solutions with a high accuracy, however, this is a passive measure and maybe takes a lot of computational time to achieve such a goal. It is a topic worthwhile further study and is not the main concern of this paper. Suppose that there are $N_{pop} + N_{offs}$ rule sets candidates, where N_{pop} is the parent population size and N_{offs} is the number of offspring resulting from crossover and mutation operations. The Fuzzy Set Agent selects N_{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 Fuzzy Set Agent 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.

2. The Interaction among agents

The Fuzzy Set Agents can interact with each other. In the current work, we assume that the number of offspring Fuzzy Set Agents (N_{aoff}) that we want to generate is even and less than or equal to the number of Fuzzy Set Agents (N_{acur}) in the current population: $N_{aoff} \leq N_{acur}$. We select N_{aoff} Fuzzy Set Agents from the current agent population and use the crossover and mutation operations to generate N_{aoff} offspring agents, i.e., two parent agents generate two offspring agents. The N_{aoff} Fuzzy Set Agents are different with one another and selected randomly with the same probability. It is because such a selection mechanism is simple and easy to implement, and the mating restriction is not incorporated in the current research. 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 fuzzy rule base. Thus, the cooperation among the fuzzy sets agents 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 Arbitrator Agent. As mentioned above, the accuracy is predefined the first priority and the other three criteria are predefined the same second priority. The Arbitrator Agent implements the NSGA-II algorithm to evaluate the parent and offspring fuzzy set agents and select N_{acur} best agents to become the next agent population. *Strong* fuzzy set agents considering both the accuracy and interpretability survive from the competition, whereas the *weak* 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 the fuzzy systems constructed by the 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 is downloaded from the University of California, Irvine (UCI) database [34].

A. Example: Nonlinear plant with two inputs and one output

The 2^{nd} order nonlinear plant is studied by Wang and Yen in [35,38,39]; Roubos and Setnes, et al. in [28,29] and Jim*acute*eenez, et al. in [15]:

$$y(k) = g(y(k-1), y(k-2)) + u(k) \tag{21}$$

$$g(y(k-1), y(k-2)) = \frac{y(k-1)y(k-2)(y(k-1)-0.5)}{1+y^2(k-1)+y^2(k-2)} \tag{22}$$

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 Figure 8.

In this agent-based approach, we use eight fuzzy set agents each of which has five fuzzy rule sets solutions, so there are totally forty 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 multi-agent system is given in Figure 9.

From Figure 9, we can see the sawtoothed curve and the tendency plots of the four criteria all go downward indicating that the agents autonomously learn to construct fuzzy systems with considerations of both the accuracy and interpretability within 100 iterations. It is mainly because that the NSGA-II algorithm is applied and the selection mechanism in NSGA-II is an elitism mechanism. So the four criteria are optimized simultaneously. On the other hand, the tendency curve is sawtoothed means that there is a trade-off between the accuracy and interpretability of fuzzy systems. Also we give the *Pareto front* [4] after 100 iterations of the evolution. *Pareto optimum* is the most commonly accepted term used in the literature of multiple objective optimization. The *Pareto optimal* is defined as: A vector of decision variables $\vec{x}^* \in \mathcal{F}$ is *Pareto optimal* if there does not exist another $\vec{x} \in \mathcal{F}$ such that $f_i(\vec{x}) \leq f_i(\vec{x}^*)$ for all $i = 1, \cdots, k$

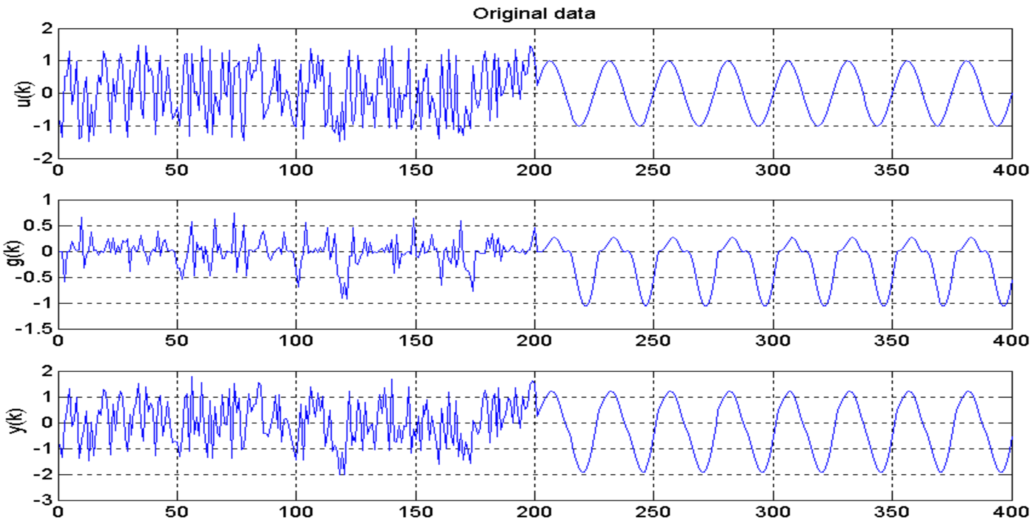


Fig. 8: Input u(k), unforced system g(k), and output y(k) of the plant in (21)

and $f_j(\vec{x}) < f_j(\vec{x}^*)$ for at least one j. It is based on the minimization problems and $\mathcal{F}$ denotes the decision variable space and f_i is one of the k function objectives. This definition says that $\vec{x}^*$ is Pareto optimal if there exists no feasible vector of decision variables $\vec{x} \in \mathcal{F}$ which would decrease some criteria without causing a simultaneous increase in at least one other criterion. Unfortunately, this concept almost always gives not a single solution, but rather a set of solutions called the *Pareto optimal set*. The vectors $\vec{x}^*$ corresponding to the solutions included in the Pareto optimal set are called *non-dominated*. The plot of the objective functions whose non-dominated vectors are in the Pareto optimal set is called the *Pareto front*. Figure 10 shows the trade-off among the multiple objectives within the non-dominated fuzzy system solutions. The upper left subfigure illustrates the trade-off relation between the accuracy and fuzzy sets number, the upper right subfigure shows the trade-off between that accuracy and fuzzy rules number, the lower left subfigure for the trade-off between the accuracy and fuzzy rules total length and the lower right one shows the trade-off among three objectives: accuracy, fuzzy sets number as well as fuzzy rules number. There are 14 non-dominated solutions out of the forty (only 9 different forms). Then we use the 14 non-dominated fuzzy system solutions to test the validation data set. Figure 11 shows the test results. For comparison, we use all the forty fuzzy system solutions to test the validation data set and show the non-dominated solutions based on the test accuracy and the other three criteria in Figure 12. There are 8 non-dominated solutions (only 3 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 the models of the compared methods in [15,28,29,35,38,39] 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 non-dominated solutions about test data are given in Table II. For brevity, the fuzzy distribution and the fuzzy rules expression are not given in this paper.

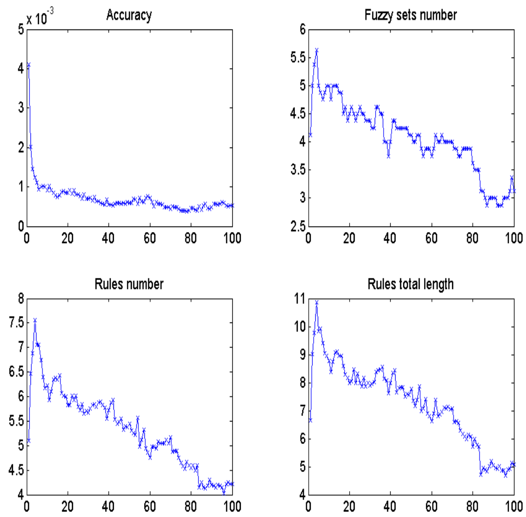


Fig. 9: Trends of average accuracy, fuzzy sets number, rules number, and rule base total length of the plant in (21)

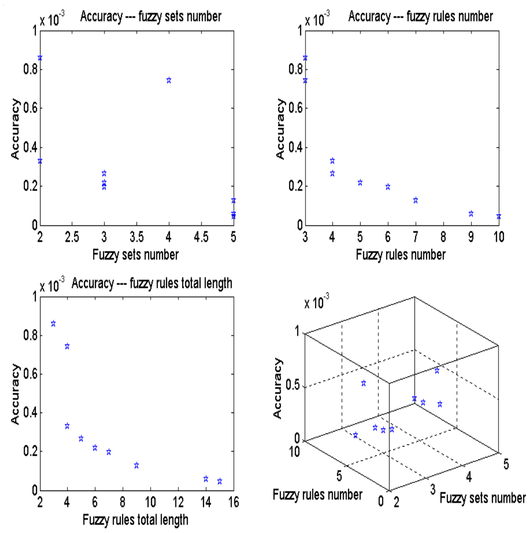


Fig. 10: Pareto front about the fuzzy systems of the plant in (21)

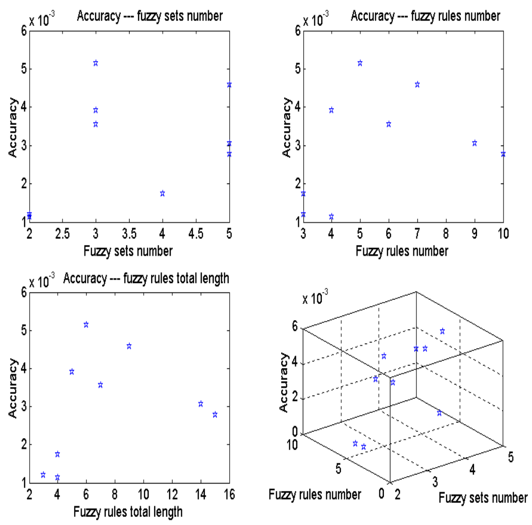


Fig. 11: Test results of the non-dominated fuzzy systems of the plant (21)

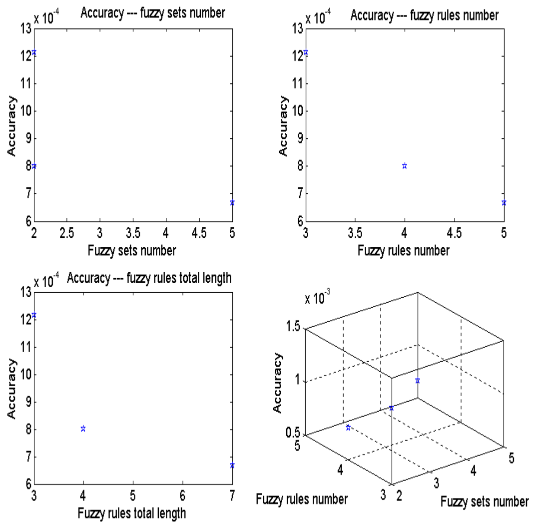


Fig. 12: Pareto front about the fuzzy systems of the plant in (21) for the test data set

TABLE I. Fuzzy models of the nonlinear plant of example A							
Ref.	No. of Rules	No. of Sets	Rules leng.	Consequent	MSE Train	MSE Test	No. of generation
[35]	40 rules (initial)	40 Gauss.	80	Singleton	3.2884e-4	6.9152e-4	/
	28 rules (optimized)	28 Gauss.	56	Singleton	3.3299e-4	5.9595e-4	100
[39]	25 rules (initial)	25 Gauss.	50	Singleton	2.3092e-4	4.0717e-4	N/A
	20 rules (optimized)	20 Gauss.	40	Singleton	6.8341e-4	2.3836e-4	N/A
[38]	36 rules (initial)	12 B-splines	72	Singleton	2.7743e-5	5.1163e-3	N/A
	23 rules (optimized)	12 B-splines	46	Singleton	3.1746e-5	1.4776e-3	N/A
	36 rules (initial)	12 B-splines	72	Linear	1.9465e-6	2.9211e-3	N/A
	24 rules (optimized)	12 B-splines	48	Linear	1.9835e-6	6.4120e-4	N/A
[29]	7 rules (initial)	14 triangular	14	Singleton	1.6e-2	1.2e-3	/
	7 rules (optimized)	14 triangular	14	Singleton	3.0e-3	4.9e-4	2000
	5 rules (initial)	10 triangular	10	Linear	5.8e-3	2.5e-3	/
	5 rules (optimized)	8 triangular	10	Linear	7.5e-4	3.5e-4	2000
	4 rules (optimized)	4 triangular	8	Linear	1.2e-3	4.7e-4	2000
[28]	5 rules (initial)	10 triangular	10	Linear	4.9e-3	2.9e-3	/
	5 rules (optimized)	10 triangular	10	Linear	1.4e-3	5.9e-4	600
	5 rules (optimized)	5 triangular	10	Linear	8.3e-4	3.5e-4	600
[15]	5 rules (optimized)	5 trapezoidal	10	Linear	2.0e-3	1.3e-3	N/A
	5 rules (optimized)	6 trapezoidal	10	Linear	5.9e-4	8.8e-4	N/A
This paper							
solution 1*6	3 rules	2 Gauss2mf.	3	Linear	8.5782e-4	1.2154e-3	100
solution 2*1	3 rules	4 Gauss2mf.	4	Linear	7.4202e-4	1.7401e-3	100
solution 3*1	10 rules	5 Gauss2mf.	15	Linear	4.5181e-5	2.7872e-3	100
solution 4*1	4 rules	3 Gauss2mf.	5	Linear	2.6503e-4	3.9176e-3	100
solution 5*1	9 rules	5 Gauss2mf.	14	Linear	5.6968e-5	3.0596e-3	100
solution 6*1	5 rules	7 Gauss2mf.	9	Linear	1.2806e-4	4.5867e-3	100
solution 7*1	4 rules	2 Gauss2mf.	4	Linear	3.3038e-4	1.1428e-3	100
solution 8*1	6 rules	3 Gauss2mf.	7	Linear	1.9523e-4	3.5593e-3	100
solution 9*1	5 rules	3 Gauss2mf.	6	Linear	2.1698e-4	5.1407e-3	100

TABLE II. Non-dominated fuzzy models based on the test data in example A					
Solutions	No. of Rules	No. of Sets	Rules leng.	MSE Train	MSE Test
solution 1*6	3 rules	2 Gauss2mf.	3	8.5782e-4	1.2154e-3
solution 2*1	4 rules	2 Gauss2mf.	4	3.3299e-4	8.0147e-4
solution 3*1	5 rules	5 Gauss2mf.	7	7.0870e-4	6.6750e-4

B. Example: Lorenz system

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

$$\dot{x} = -y^2 - z^2 - a(x - F) \tag{23}$$

$$\dot{y} = xy - bxz - y + G \tag{24}$$

$$\dot{z} = bxy + xz - z \tag{25}$$

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 $z(t-1)$. 400 data points are obtained from the Equations (23), (24) and (25) 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 for test. The sampling data pairs are shown in Figure 13.

In this work, we use eight fuzzy set agents and five fuzzy rule sets solutions for each agent, so there are totally forty fuzzy systems. The trends plot about the same four criteria as those

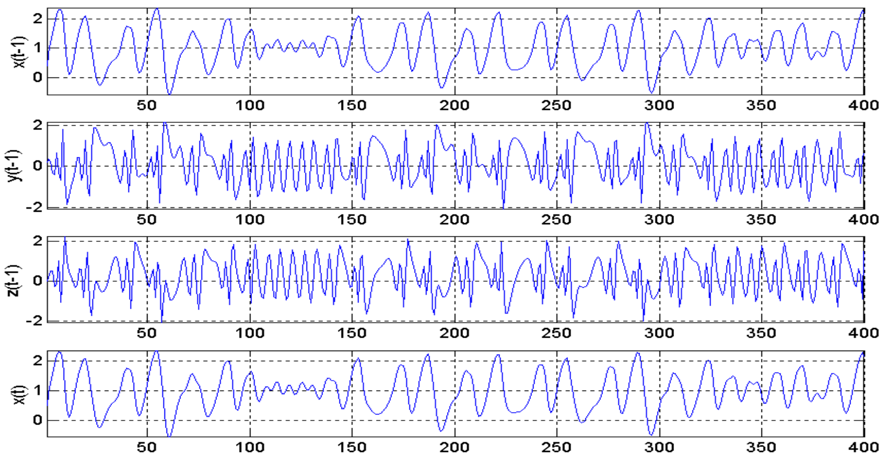


Fig. 13: Input x(t-1), y(t-1) and z(t-1) , output x(t) of the Lorenz system

in Figure 9 is given in Figure 14. Figure 15 shows the trade-off among the multiple objectives within the non-dominated fuzzy system solutions based on the training data. There are 17 non-dominated solutions out of the forty (only 8 different forms).

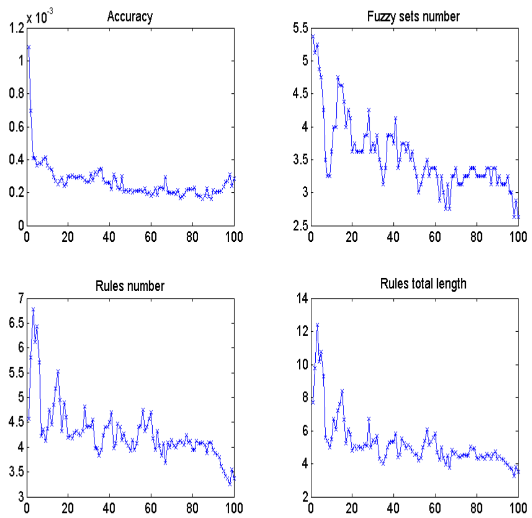


Fig. 14: Trends of average accuracy, fuzzy sets number, rules number, and rule base total length of the Lorenz system.

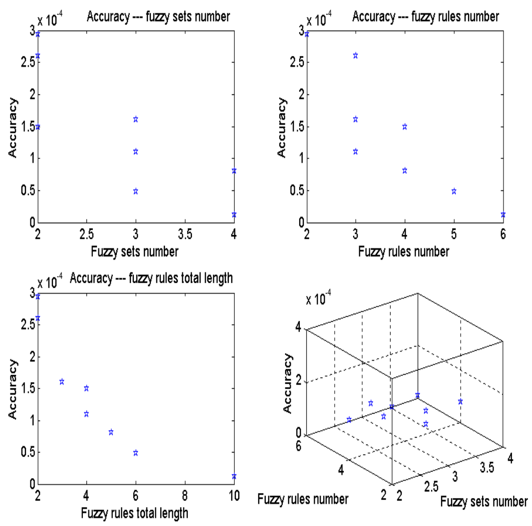


Fig. 15: Pareto front about the Lorenz system

Then we use the 17 non-dominated fuzzy system solutions to test the validation data set. Figure 16 shows the test results. For comparison, we use all the forty fuzzy system solutions to test the validation data set and show the non-dominated solutions based on the test accuracy and the other three criteria in Figure 17. There are 10 non-dominated solutions (only 7 different forms) associated with the test data. In Table III, we compare the results with that in [19]. The MSE result is not given in [19], so we use "N/A" in Table III to denote such a case. However, we think our results with respect to MSE are satisfying. The non-dominated solutions about test data are given in Table IV.

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., 3 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. For clear speaking, 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 order. For brevity, we give only one complete rule base related to the solution 8 in Table III. Figure 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.

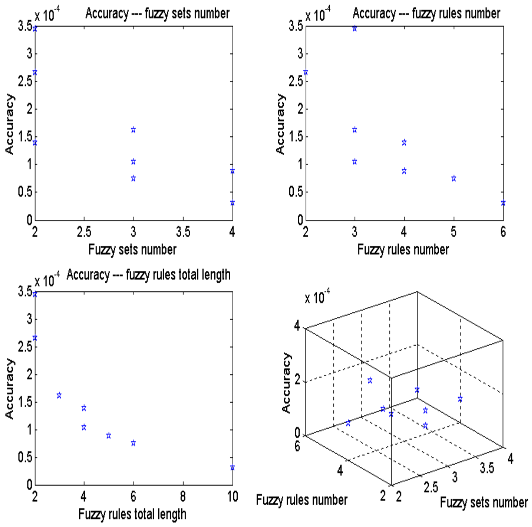


Fig. 16: Test results of the non-dominated fuzzy systems of the Lorenz system

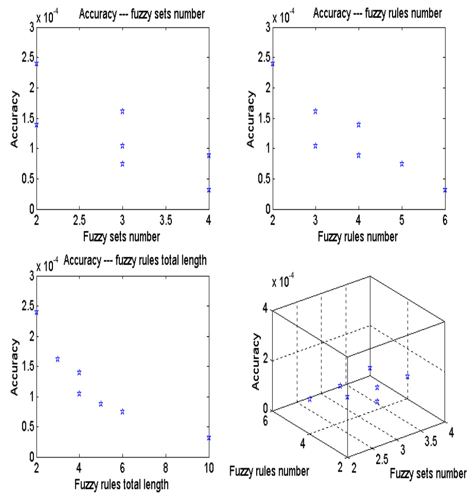


Fig. 17: Test results of the non-dominated fuzzy systems of the Lorenz system

TABLE III. Fuzzy models of the Lorenz system

Ref.	Input	No. of Rules	No. of Sets	Rules leng.	MSE Train	MSE Test
[19]	3{4,2,1}	4 rules	7 Gauss.	8	N/A	N/A
This paper						
solution 1*1	2{0,2,2}	6 rules	4 Gauss2mf.	10	1.2798e-5	3.1480e-5
solution 2*1	2{0,2,2}	4 rules	4 Gauss2mf.	5	8.1136e-5	8.8000e-5
solution 3*1	2{0,1,2}	5 rules	3 Gauss2mf.	6	4.8548e-5	7.4819e-5
solution 4*1	2{0,1,2}	3 rules	3 Gauss2mf.	3	1.6088e-4	1.6160e-4
solution 5*1	2{0,1,2}	3 rules	3 Gauss2mf.	4	1.1023e-4	1.0447e-4
solution 6*2	2{0,1,1}	2 rules	2 Gauss2mf.	2	2.9419e-4	2.6617e-4
solution 7*4	2{0,1,1}	4 rules	2 Gauss2mf.	4	1.4979e-4	1.3927e-4
solution 8*6	2{0,1,1}	3 rules	2 Gauss2mf.	2	2.6085e-4	3.4412e-4

TABLE IV. Non-dominated fuzzy models based on the test data for the Lorenz system					
Solutions	No. of Rules	No. of Sets	Rules leng.	MSE Train	MSE Test
solution 1*1	6 rules	4 Gauss2mf.	10	1.2798e-5	3.1480e-5
solution 2*1	4 rules	4 Gauss2mf.	5	8.1136e-5	8.8000e-5
solution 3*1	2 rules	2 Gauss2mf.	2	3.0443e-4	2.3970e-4
solution 4*1	5 rules	3 Gauss2mf.	6	4.8548e-5	7.4819e-5
solution 5*1	3 rules	3 Gauss2mf.	3	1.6088e-4	1.6160e-4
solution 6*1	3 rules	3 Gauss2mf.	4	1.1023e-4	1.0447e-4
solution 7*4	4 rules	2 Gauss2mf.	4	1.4979e-4	1.3927e-4

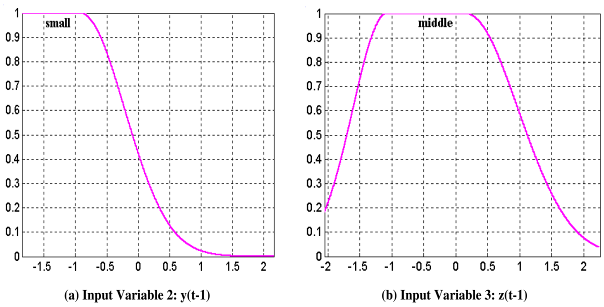


Fig. 18: Fuzzy sets of Lorenz system

Table V. Rule base for Lorenz system
R_1: If $y(t-1)$ is small, then $x(t) = 0.989x(t-1) + 0.227y(t-1) + 0.009z(t-1) + 0.018$;
R_2: If $z(t-1)$ is middle, then $x(t) = 0.975x(t-1) - 0.119y(t-1) + 0.049z(t-1) + 0.163$;
R_3: Else $x(t) = 0.963x(t-1) + 0.071y(t-1) + 0.009z(t-1) - 0.210$.

C. Example: Iris Data

The Iris Data contains 150 pattern instances with 4 attributes from 3 classes available from the University of California, Irvine (UCI) database [34]. We use all of the data to train ten fuzzy set agents each of which has eight fuzzy rule sets solutions, so there are totally eighty fuzzy systems. The trends plot about four criteria including average accuracy (classification error rate), average fuzzy sets number, average fuzzy rules number, and average fuzzy rule base total length among the multi-agent system is given in Figure 19. Figure 20 shows the trade-off among the multiple objectives within the non-dominated fuzzy classifier system solutions for iris data. There are 13 non-dominated solutions out of the eighty (only 5 different forms related to the four criteria). Also we compare our results with other works reported in the literature. The comparative 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 the compactness issues. We illustrate only one fuzzy classifier system corresponding to the solution 3 in Table VI owing to the brief considerations. The distribution of fuzzy sets is shown in Figure 21, and Table VII lists the fuzzy rules.

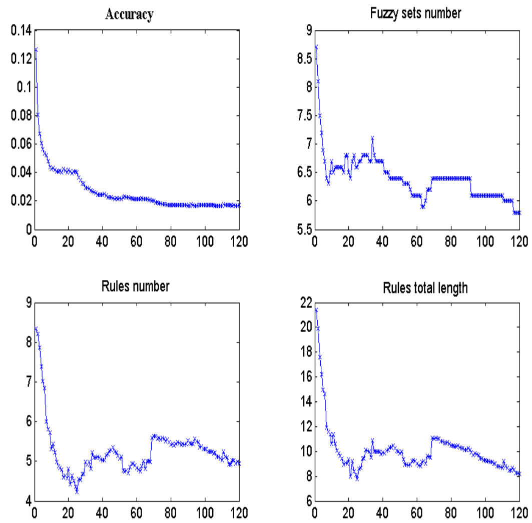


Fig. 19: Trends of average accuracy, fuzzy sets number, rules number, and rule base total length for Iris Data.

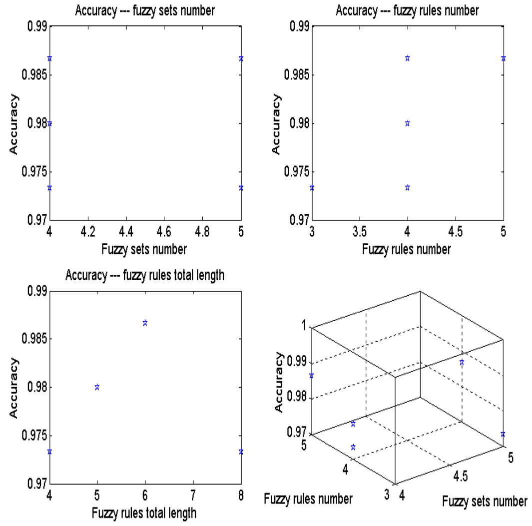
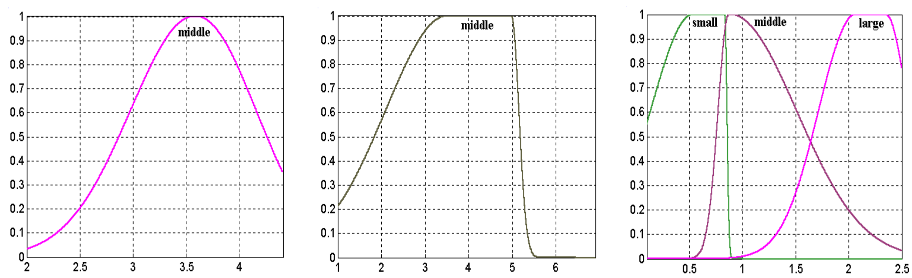


Fig. 20: Pareto front about the Iris Data

Table VI. Comparison results for Iris Data					
Ref.	Classification rate	Input	No. of Sets	No. of Rules	Rules leng.
[28]	0.973	2{0,0,3,2}	5	3	6
[2]	0.993	4{3,3,3,3}	12	3	12
[10]	0.973	4{5,5,5,5}	20	4.6	N/A
[30]	1.000	4{4,5,4,5}	18	5	18
[37]	0.993	4{3,3,3,3}	12	6	24
This paper					
solution 1*8	0.980	3{0,1,1,2}	4	4	5
solution 2*1	0.987	3{0,1,1,2}	4	5	6
solution 3*2	0.987	3{0,1,1,3}	5	4	6
solution 4*1	0.973	3{0,1,1,3}	5	3	8
solution 5*1	0.973	3{0,1,1,2}	4	4	4

Table VII. Rule base for Iris Data
R_1: If x_2 is middle and x_3 is middle, then class 2;
R_2: If x_4 is small, then class 1;
R_3: If x_3 is middle and x_4 is middle, then class 2;
R_4: If x_4 is large, then class 3.



(a) Attribute 2: Sepal width **(b) Attribute 3: Petal length** **(c) Attribute 4: Petal width**

Fig. 21: Fuzzy sets of fuzzy classifier system for Iris Data

From Figures 9, 14 and 19, we can see that our agent-based approach can guarantee good convergence among the multiple objectives. The objectives considering both the accuracy and interpretability in our research context can co-evolve in progress together. Another advantage of our approach is that we can obtain multiple non-dominated fuzzy systems concentrating on both the accuracy and interpretability of fuzzy systems. It is obviously illustrated in Figures 10, 15 and 20, and quantified in Tables I, III and VI. From these tables, we also demonstrate that the accuracy of our results is compatible with or better than other methods known in the literature. And more important, most solutions that we get have better interpretability. The trade-off between accuracy and interpretability of fuzzy systems is also easily understood. 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 three out of four 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 based system 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 in future research.

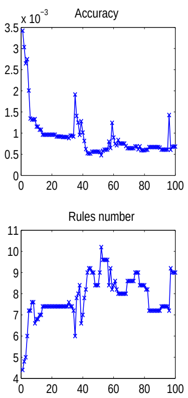


Fig. 22: Example A using one agent

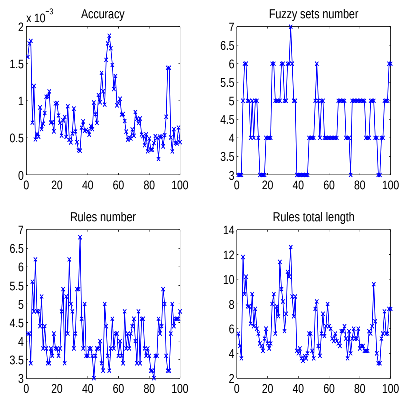


Fig. 23: Example B using one agent

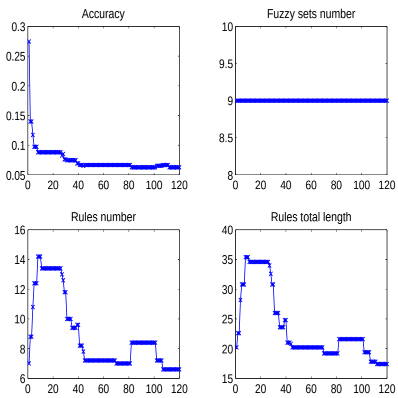


Fig. 24: Example C using one agent

In order to show the effectiveness of the multi-agent approach, we did the simulation experiments based on only one agent, i.e., just use one agent to learn the fuzzy rule base without changing other parameters. For brevity, we only give the trends plot of Example A, B and C in Figures 22, 23 and 24, respectively. We compare the trends plot with those of the multi-agent approach (see Figures 9, 14, and 19). We can see that the average classification rate, fuzzy sets, fuzzy rules number, and fuzzy rules total length of the evolutionary approach using multiple agents are better than those of the single agent approach and have a better convergency. It means that the multi-agent approach has good abilities to explore interpretable rule base with the accuracy consideration based on the obtained fuzzy sets. In the multi-agent 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. Hence, the multi-agent approach can obtain a more compact fuzzy system. It is also a main goal for us to design such a multi-agent mechanism: to explore more appropriate fuzzy sets distribution and uses a smaller number of fuzzy sets.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed an agent-based evolutionary approach to construct interpretable fuzzy systems. In the multi-agent system, the Fuzzy Set Agents autonomously implement the intra-tasks: 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 multi-objective 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 and create offspring agents. The arbitrator agent evaluates the parent and offspring agents based on the criteria on 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 trade-off 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. 2) Appling some data mining techniques related to dimension reduction such as SUD [3], RELIEF, and SCM [6], etc to our multi-agent system, hopefully using more important attributes to train the agents leading to a more compact fuzzy system construction.

References

[1] J. Casillas, O. Cordón, F. Herrera, and L. Magdalena (Eds.), Interpretability Issues in Fuzzy Modeling, Verlag/Jahr: SPRINGER, BERLIN 2003, ISBN 3-540-02932-X

[2] G. Castellano, A.M. Fanelli, E. Gentile and T. Roselli, "A GA-based approach to optimization of fuzzy models learned from data", GECCO-2002 Program, New York, pp. 5-8, July 9, 2002

[3] M. Dash, H. Liu, and J. Yao, "Dimensionality Reduction for Unsupervised Data", 9th International Conference on Tools with Artificial Intelligence, pp. 532-539, Newport Beach, CA, November 3-8, 1997

[4] K. Deb, "Multi-Objective Optimization using Evolutionary Algorithms", John Willy & Sons, Chichester, UK, 2001

[5] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II", IEEE Transactions on Evolutionary Computation, Vol. 6, No. 2, pp. 182-197, April 2002

[6] X. Fu and L. Wang, "Data Dimensionality Reduction With Application to Simplifying RBF Network Structure and Improving Classification Performance", IEEE Transactions on Systems, Man, and Cybernetics-Part B, Vol. 33, No. 3, pp. 399-409, June 2003

[7] S. Guillaume, "Designing Fuzzy Inference Systems from Data: An interpretability Oriented Review", IEEE Transactions on Fuzzy System, Vol.9, No.3, pp. 426-443, June 2001

[8] H. Ishibuchi, and T. Nakashima, "Effect of Rule Weights in Fuzzy Rule-Based Classification Systems", IEEE Transactions on Fuzzy Systems, Vol. 9, No. 4, pp.506-515, August 2001

[9] H. Ishibuchi, T. Nakashima, "Three-Objective Optimization in Linguistic Function Approximation", Proc. of Congress on Evolutionary Computation 2001 (Seoul, Korea), pp. 340-347, May 27-30, 2001

[10] H. Ishibuchi, T. Nakashima, and T. Kuroda, "A Hybrid Fuzzy GBML Algorithm for Designing Compact Fuzzy Rule-Based Classification Systems", Proc. of 9th IEEE International Conference on Fuzzy Systems (San Antonio), pp.706-711, May 7-10, 2000

[11] H. Ishibuchi, T. Nakashima, and T. Kuroda, "A Hybrid Fuzzy Genetics-based Machine Learning Algorithm: Hybridization of Michigan Approach and Pittsburgh Approach", Proc. of IEEE International Conference on Systems, Man and Cybernetics (Tokyo, Japan), pp. 296-301, October 12-15, 1999

[12] H. Ishibuchi, T. Nakashima, and T. Murata, "Multiobjective Optimization in Linguistic Rule Extraction from Numerical Data", EMO 2001, LNCS 1993, pp. 588-602, 2001

[13] H. Ishibuchi, T. Nakashima, and T. Murata, "Performance Evaluation of Fuzzy Classifier Systems for Multidimensional Pattern Classification Problems", IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 29, No. 5, pp. 601-618, October 1999

[14] H. Ishibuchi, K. Nozaki, and H. Tanaka, "Distributed representation of fuzzy rules and its application to pattern classification", Fuzzy Sets and Systems, Vol.52, No.1, pp.21-32, November, 1992

[15] F. Jiménez, A. F. Gómez-Skarmeta, H. Roubos, and R. Babuška, Accurate, Transparent, and Compact Fuzzy Models for Function Approximation and Dynamic Modeling through Multi-objective Evolutionary Optimization, First International Conference on Evolutionary Multi-criterion Optimization, pp. 653-667, 2001

[16] Y. Jin, "Fuzzy Modeling of high-dimensional systems: Complexity reduction and interpretability improvement", IEEE Transactions on Fuzzy Systems, Vol. 8, No. 2, pp. 212-221, 2000

[17] Y. Jin, W. von Seelen and B. Sendhoff, "An approach to Rule-Based Knowledge Extraction", Proceedings of IEEE Conference on Fuzzy System, Vol. 2, pp. 1188-1193, May 1998

[18] Y. Jin, W. von Seelen, B. Sendhoff, "On Generating FC3 Fuzzy Rule Systems from Data Using Evolution Strategies", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 29, No. 6, pp.829-845, December 1999

[19] Y. Jin and B. Sendhoff, "Extracting Interpretable Fuzzy Rules from RBF Networks", Neural Processing Letters, 17(2), pp. 149-164, 2003

[20] C. F. Juang and C. T. Lin, "An On-Line Self-Constructing Neural Fuzzy Inference Network and Its Applications", IEEE Transactions on Fuzzy Systems, Vol. 6, No. 1, pp. 12-32, 1998

[21] C. T. Lin, "A Neural Fuzzy Control System with Structure and Parameter Learning", Fuzzy Sets and Systems, Vol. 70, pp. 183-212, 1995

[22] C. T. Lin and C. S. G. Lee, "Neural-Network-Based Fuzzy Logic Control and Decision System", IEEE Transactions on Computers, Vol. 40, No. 12, pp. 1320-1336, 1991

[23] K. F. Man, K. S. Tang and S. Kwong, "Genetic Algorithms Concepts and Designs", Springer-Verlag London Limited 1999, Printed in Great Britain

[24] K. F. Man, K. S. Tang, S. Kwong and W. A. Halang, "Genetic algorithms for control and signal processing", Springer-Verlag London Limited 1997, Printed in Great Britain

[25] 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", Proceedings on Genetic and Evolutionary Computation Conference, pp. 485-492, San Francisco, July 2001

[26] K. M. Passino and S. Yurkovich, "Fuzzy Control", Copyright © 1998 by Addison Wesley Longman, Inc

[27] I. Rojas, H. Pomares, J. Ortega, and A. Prieto, "Self-Organized Fuzzy System Generation from training Examples", IEEE Transactions on Fuzzy Systems, Vol. 8, No. 1, pp. 23-36, February 2000

[28] H. Roubos and M. Setnes, "Compact and Transparent Fuzzy Models and Classifiers Through Iterative Complexity Reduction", IEEE Transactions on Fuzzy Systems, Vol. 9, No. 4, pp. 516-524, August 2001

[29] H. Roubos and M. Setnes, "GA-Fuzzy Modeling and Classification: Complexity and Performance", IEEE Transactions on Fuzzy Systems, Vol. 8, No. 5, pp. 509-522, October 2000

[30] M. Russo, "Genetic Fuzzy Learning", IEEE Transactions on Evolutionary Computation, Vol. 4, No. 3, pp. 259-273, 2000

[31] M. Setnes, R. Babuška, U. Kaymak, and H. R. van Nauta Lemke, "Similarity Measures in Fuzzy Rule Base Simplification", IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 28, No. 3, pp. 376-386, June 1998

[32] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 15, No. 1, pp. 116-132, 1985

[33] K. S. Tang, K. F. Man, Z. F. Liu and S. Kwong, "Minimal Fuzzy Memberships and Rules Using Hierarchical Genetic Algorithms", IEEE Transactions on Industrial Electronics, Vol. 45, No. 1, pp. 162-169, February 1998

[34] UCI Machine Learning Repository, http://www.ics.uci.edu/ mlearn/MLRepository.html

[35] L. Wang and J. Yen, "Exacting fuzzy rules for system modeling using a hybrid of genetic algorithms and Kalman filter", Fuzzy Sets and Systems, Vol. 101, pp. 353-362, 1999

[36] N. Xiong, "Evolutionary learning of rule premises for fuzzy modeling", International Journal of Systems Science, Vol. 32, No. 9, pp. 1109-1118, 2001

[37] N. Xiong and L. Litz, "Identifying Flexible Structured Premises for Mining Concise Fuzzy Knowledge", Interpretability Issues in Fuzzy Modeling, Eds.: J. Casillas et al., pp. 54-76, 2003

[38] J. Yen and L. Wang, "Application of Statistical Information Criteria for Optimal Fuzzy Model Construction", IEEE Transactions on Fuzzy systems, Vol. 6, No. 3, pp. 362-372, 1998

[39] J. Yen and L. Wang, "Simplifying Fuzzy Rule-Based Models Using Orthogonal Transformation Methods", IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 29, No. 1, pp. 13-24, 1999

[40] L.A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes", IEEE Transaction on Systems, Man, and Cybernetics, Vol. SMC-3, pp. 28-44, January 1973