Interpretability Issues in Fuzzy Genetics-Based Machine Learning for Linguistic Modelling

Hisao Ishibuchi and Takashi Yamamoto

Department of Industrial Engineering, Osaka Prefecture University
1-1 Gakuen-cho, Sakai, Osaka 599-8531, Japan
{hisaoi,yama}@ie.osakafu-u.ac.jp

Abstract. This chapter discusses several issues related to the design of linguistic models with high interpretability using fuzzy genetics-based machine learning (GBML) algorithms. We assume that a set of linguistic terms has been given for each variable. Thus our modelling task is to find a small number of fuzzy rules from possible combinations of the given linguistic terms. First we formulate a three-objective optimization problem, which simultaneously minimizes the total squared error, the number of fuzzy rules, and the total rule length. Next we show how fuzzy GBML algorithms can be applied to our problem in the framework of multi-objective optimization as well as single-objective optimization. Then we point out a possibility that misleading fuzzy rules can be generated when general and specific fuzzy rules are simultaneously used in a single linguistic model. Finally we show that non-standard inclusion-based fuzzy reasoning removes such an undesirable possibility.

1 Introduction

Since Takagi & Sugeno's pioneering work [33], fuzzy modelling has been extensively studied [27]. In the 1990s, many approaches were proposed for fuzzy modelling such as heuristic methods [26, 35], fuzzy-neuro methods [9, 19, 32], and genetic fuzzy methods [2, 7, 24] where emphasis was primarily placed on the improvement in the accuracy of fuzzy models. The interpretability of fuzzy models was also discussed in some studies [29, 31, 34]. Recently the existence of a tradeoff between the accuracy and the interpretability of fuzzy models was recognized [3] and taken into account in many studies on fuzzy modelling [22, 23, 28, 30]. While multiple criteria were simultaneously considered in the design of fuzzy models in those studies, fuzzy modelling was handled in the framework of single-objective optimization. That is, the final goal in those studies was to design a single fuzzy model with high accuracy and high interpretability. The handling of the design of fuzzy models in the framework of multi-objective optimization was first proposed for fuzzy rule-based classification in [11] where the goal was not to find a single fuzzy model but to find multiple non-dominated fuzzy models with respect to the classification accuracy and the number of

J. Lawry, J. Shanahan, A. Ralescu (Eds.): Modelling with Words, LNAI 2873, pp. 209-228, 2003.

fuzzy rules. The two-objective formulation in [11] was extended to the case of three-objective optimization in [13, 14] where the total rule length was used as the third objective. Jimenez et al. [20, 21] discussed multi-objective optimization of Takagi-Sugeno models where the accuracy, the transparency and the compactness were considered. Since Takagi-Sugeno models have a linear function in the consequent part of each fuzzy rule, their linguistic interpretability is not high. Thus we use more descriptive fuzzy rules with linguistic terms in both the antecedent and consequent parts (i.e., Mamdani rules).

Let us assume that we have m input-output pairs $(\mathbf{x}_p, y_p)$, $p = 1,2,...,m$ for an n-input and single-output unknown nonlinear function where $\mathbf{x}_p = (x_{p1}, ..., x_{pn})$ is an n-dimensional input vector and y_p is the corresponding output value. We also assume that a set of linguistic terms has been given by domain experts or human users for each variable. For simplicity of explanation, we use five linguistic terms in Fig. 1 for all the input and output variables. Our task is to linguistically describe the unknown nonlinear function using fuzzy rules of the following form:

$$\text{Rule } R_k : \text{If } x_1 \text{ is } A_{k1} \text{ and ... and } x_n \text{ is } A_{kn} \text{ then } y \text{ is } B_k, \tag{1}$$

where R_k is the label of the kth fuzzy rule, x_i is the ith input variable, A_{ki} is an antecedent fuzzy set on the ith input variable x_i, y is the output variable, and B_k is a consequent fuzzy set. The consequent fuzzy set B_k is one of the five linguistic terms in Fig. 1 while the antecedent fuzzy set A_{ki} can assume *don't care* in addition to the five linguistic terms. Thus the total number of fuzzy rules of the form in (1) is $(5+1)^n \cdot 5$. We do not modify the membership function of each linguistic term because the modification usually degrades the interpretability of fuzzy rules. Thus the design of a linguistic model can be viewed as finding a subset of $(5+1)^n \cdot 5$ fuzzy rules. The size of the search space is 2^N where $N = (5+1)^n \cdot 5$.

In the next section, we formulate our modelling task as a three-objective optimization problem. The three objectives are to minimize the total squared error, the number of fuzzy rules, and the total rule length. The rule length is defined by the number of antecedent conditions. In Section 3, we show the handling of our modelling task in the framework of single-objective optimization where the weighted sum of the three objectives is used as a scalar fitness function. A Pittsburgh-style fuzzy GBML algorithm is used for finding a single linguistic model. In Section 4, we show the handling of our modelling task in the framework of multi-objective optimization. The single-objective fuzzy GBML algorithm is extended using multi-objective genetic algorithms [4, 6]. Our modelling task is tackled by genetic rule selection in Section 5 where we also explain heuristic fuzzy rule generation using rule evaluation measures in data mining [1, 8, 18]. In Section 6, we point out a possibility that misleading fuzzy rules can be extracted when linguistic models include both general and specific fuzzy rules. After explaining why misleading fuzzy rules are extracted, we show that the use of a non-standard fuzzy reasoning method [10] removes such an undesirable possibility. Section 7 concludes this chapter.

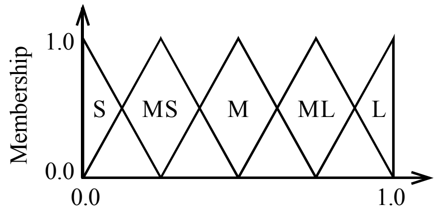

Fig. 1. Membership functions of five linguistic terms (S: *small*, MS: *medium small*, M: *medium*, ML: *medium large*, and L: *large*)

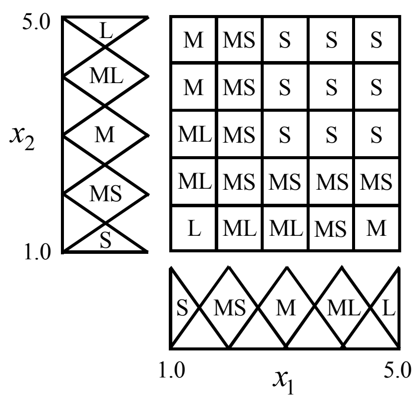

Fig. 2. A fuzzy rule table that linguistically describes the nonlinear function in (2)

2 Formulation of Linguistic Modelling

First we explain the basic idea of linguistic modelling using the following two-input and single-output nonlinear function [33]:

$$y = (1 + x_1^{-2} + x_2^{-1.5})^2, \quad 1 \le x_i \le 5 \ \text{ for } i = 1,2\,. \tag{2}$$

Nozaki et al. [26] extracted 25 linguistic rules in Fig. 2 where the five linguistic terms in Fig. 1 are used as consequent fuzzy sets. While it is not easy to intuitively understand the shape of the nonlinear function from the mathematical description in (2), we can easily grasp a rough three-dimensional shape of the nonlinear function from the linguistic description shown in Fig. 2. Thus the fuzzy rule table in Fig. 2 is an interpretable linguistic model of the nonlinear function in (2). As shown in this example, two-input and single-output nonlinear functions can be linguistically described by fuzzy rule tables in a human understandable manner.

The main difficulty in the application of fuzzy rule tables to high-dimensional nonlinear functions is the exponential increase in the number of fuzzy rules, which is often referred to as the curse of dimensionality. Let K be the number of linguistic terms for each input variable (e.g., $K = 5$ in Fig. 2), the number of fuzzy rules in an n-dimensional fuzzy rule table is K^n. Thus the interpretability of fuzzy rule tables is severely deteriorated by the increase in the number of input variables. For example, let us consider the following nonlinear function [12]:

$$y = \frac{1}{2\left(1+\exp\{\sum_{i=1}^{3}(-60x_i+55)\}\right)}, \quad 0 \le x_i \le 1 \text{ for } i = 1,2,3. \tag{3}$$

When we have the five linguistic terms in Fig. 1 for all the three input and single output variables, we can easily generate 125 fuzzy rules using a heuristic method (e.g., [26, 35]). It is, however, not easy for human users to understand the nonlinear function from the generated 125 fuzzy rules. This is because the number of the generated fuzzy rules is too large. It should be noted that the understanding of the nonlinear function from the mathematical description in (3) is also difficult.

Even when the number of fuzzy rules is small, linguistic models are not always interpretable. Another difficulty in the handling of high-dimensional problems is the increase in the rule length. It is not easy for human users to intuitively understand long fuzzy rules with many antecedent conditions. Thus the length of each fuzzy rule should be small when we design linguistic models with high interpretability. In this chapter, the number of antecedent conditions of each fuzzy rule is referred to as the rule length. For generating short fuzzy rules for high-dimensional problems, we use "*don't care*" as an additional antecedent fuzzy set. Since *don't care* is fully compatible with any input values, its membership function is defined as

$$\mu_{don't\ care}(x) = 1 \text{ for } \forall x. \tag{4}$$

Since *don't care* conditions are usually omitted from the antecedent part, fuzzy rules with many *don't care* conditions are short and interpretable. As an example, let us consider the following fuzzy rule:

If x_1 is *don't care* and x_2 is *don't care* and x_3 is *large* then y is *medium large*. (5)

We omit the two *don't care* conditions as

If x_3 is *large* then y is *medium large*. (6)

Short and long fuzzy rules are referred to as general and specific rules, respectively.

The use of *don't care* is also supported from the viewpoint of the number of fuzzy rules required for covering the whole input space. As we can see from Fig. 1, each linguistic term covers the following fraction of the domain interval [0, 1] of each input variable:

small: 1/4 $(0 \le x_i < 0.25)$, *medium small*: 1/2 $(0 < x_i < 0.5)$,
medium: 1/2 $(0.25 < x_i < 0.75)$, *medium large*: 1/2 $(0.5 < x_i < 1)$,
large: 1/4 $(0.75 < x_i \le 1)$.

Thus we can see that each linguistic term covers on average 2/5 of the domain interval [0, 1] where

$$\frac{2}{5} = \frac{1}{5} \times \left(\frac{1}{4} \times 2 + \frac{1}{2} \times 3 \right). \tag{7}$$

Since each fuzzy rule has n antecedent conditions, it covers on average $(2/5)^n$ of the n-dimensional input space $[0, 1]^n$ if no *don't care* conditions are included. That is, the fraction covered by each fuzzy rule is exponentially decreased by the increase in the dimensionality of the input space. The minimum number of fuzzy rules required for covering the whole input space is roughly estimated as $(5/2)^n$. This becomes huge in the case of high-dimensional problems. For example, $(5/2)^n$ is 9537 for $n = 10$ and about 91 million for $n = 20$. This discussion clearly shows the necessity of *don't care* conditions when we try to linguistically describe high-dimensional nonlinear functions. General fuzzy rules with many *don't care* conditions can cover a large portion of the input space. Thus the whole input space can be covered by a small number of general fuzzy rules. For example, the following two fuzzy rules were generated in [12] for the nonlinear function in (3).

y is *small*, (8)

If x_1 is *large* and x_2 is *large* and x_3 is *large* then y is *medium*. (9)

The first fuzzy rule has no antecedent conditions (i.e., it has *don't care* conditions on all the three input variables). The whole input space $[0, 1]^3$ is covered by these two fuzzy rules (Actually it is covered by the first fuzzy rule). We can easily grasp a rough shape of the nonlinear function in (3) from the two fuzzy rules in (8)-(9).

A linguistic model with only a small number of general fuzzy rules has high interpretability. If the approximation accuracy is also high, we may be able to correctly understand the nonlinear function from the linguistic model. On the other hand, the linguistic model is unreliable if its approximation accuracy is very low. Thus not only the interpretability but also the approximation accuracy should be high when we design a linguistic model for linguistically describing a nonlinear function.

When we use K linguistic terms and *don't care* in the antecedent part and K linguistic terms in the consequent part, the total number of possible fuzzy rules is $(K+1)^n \cdot K$. Let S be a subset of those fuzzy rules. Our linguistic modelling task is formulated as a three-objective combinatorial optimization problem where the following objectives are to be minimized:

$f_1(S)$: The total squared error by the rule set S.

$f_2(S)$: The number of fuzzy rules in the rule set S.

$f_3(S)$: The total rule length of fuzzy rules in the rule set S.

A similar three-objective problem was formulated for fuzzy rule-based classification in [13, 14]. It should be noted that the third objective is not the average rule length but the total rule length. This is because the average rule length does not appropriately measure the complexity of linguistic models. For example, let us consider a linguistic model with three fuzzy rules of the average length 3. If we include an additional fuzzy rule of the length 1, the average rule length is decreased from 3 to 2.5 while the actual complexity of the linguistic model is increased.

The first objective is calculated from the difference between the actual output value y_p and the estimated output value $\hat{y}(\mathbf{x}_p)$. The latter is calculated as

$$\hat{y}(\mathbf{x}_p) = \frac{\sum_{R_k \in S} \mu_{\mathbf{A}_k}(\mathbf{x}_p) \cdot b_k}{\sum_{R_k \in S} \mu_{\mathbf{A}_k}(\mathbf{x}_p)}, \tag{10}$$

where $\mu_{\mathbf{A}_k}(\mathbf{x}_p)$ is the compatibility grade of the antecedent part $\mathbf{A}_k = (A_{k1}, ..., A_{kn})$ of the fuzzy rule R_k with the input vector $\mathbf{x}_p$, and b_k is a representative real number of the consequent fuzzy set B_k. As the representative real number of each linguistic term in Fig. 1, we use the center of its triangular membership function (i.e., 0.0 for *small*, 0.25 for *medium small*, 0.5 for *medium*, 0.75 for *medium large*, and 1.0 for *large*). The compatibility grade $\mu_{\mathbf{A}_k}(\mathbf{x}_p)$ is calculated by the product operation as

$$\mu_{\mathbf{A}_k}(\mathbf{x}_p) = \mu_{A_{k1}}(x_{p1}) \cdot \mu_{A_{k2}}(x_{p2}) \cdot \ ... \ \cdot \mu_{A_{kn}}(x_{pn}), \tag{11}$$

where $\mu_{A_{ki}}(\cdot)$ is the membership function of the antecedent fuzzy set A_{ki}.

The total squared error over the m input-output pairs $(\mathbf{x}_p, y_p)$, $p = 1,2,...,m$ is calculated from the actual output value y_p and the estimated output value $\hat{y}(\mathbf{x}_p)$ as

$$f_1(S) = \frac{1}{2} \sum_{p=1}^{m} | \hat{y}(\mathbf{x}_p) - y_p |^2 . \tag{12}$$

If there is no compatible fuzzy rule with the input vector $\mathbf{x}_p$, the estimated output value $\hat{y}(\mathbf{x}_p)$ cannot be calculated by (10). In this case, we use a pre-specified large penalty value as the corresponding squared error. In our computer simulations, we specify the penalty value as $| \hat{y}(\mathbf{x}_p) - y_p |^2 = 1$ because the range of the output variable is the unit interval [0, 1] in numerical examples of this chapter.

3 Single-Objective Fuzzy GBML Algorithm

3.1 Problem Specification

When our three-objective linguistic modelling problem is handled in the framework of single-objective optimization, a scalar fitness function is defined from the three objectives. We use the following weighted sum of the three objectives as the scalar fitness function to be maximized in fuzzy GBML algorithms:

$$f(S) = -w_1 \cdot f_1(S) - w_2 \cdot f_2(S) - w_3 \cdot f_3(S), \tag{13}$$

where w_i is a user-definable positive weight for the ith objective $f_i(S)$, $i = 1,2,3$. Our three-objective linguistic modelling problem is reduced to the task of finding the optimal rule set that maximizes the scalar fitness function in (13). In this case, the obtained optimal rule set totally depends on the specification of the three weights.

3.2 Pittsburgh-Style Fuzzy GBML Algorithm

Many fuzzy GBML algorithms can be classified into two categories: Michigan Approach and Pittsburgh Approach (see [5] for various fuzzy GBML algorithms). Each fuzzy rule is handled as an individual in Michigan-style algorithms while a set of fuzzy rules is handled as an individual in Pittsburgh-style algorithms. In general, Michigan-style algorithms need much less computational load than Pittsburgh-style algorithms. The optimization of rule sets is indirectly executed through the evolution of fuzzy rules in Michigan-style algorithms while rule sets are directly optimized in Pittsburgh-style algorithms through the evolution of rule sets.

Since the scalar fitness function in (13) involves the minimization of the number of fuzzy rules, the application of Michigan-style algorithms is difficult. This is because the minimization of the number of fuzzy rules means the minimization of the population size in Michigan-style algorithms. Thus we use a Pittsburgh-style algorithm. The outline of our Pittsburgh-style algorithm is written as follows:

[Outline of Pittsburgh-Style Fuzzy GBML Algorithm]

Step 1: Randomly generate a number of rule sets as an initial population.
Step 2: Repeat the following procedures for generating new rule sets.
(a) Select a pair of parent rule sets from the current population.
(b) Generate a new rule set from the selected pair by a crossover operation.
(c) Apply mutation operations to the generated rule set.
Step 3: Update the current population using the newly generated rule sets.
Step 4: If a pre-specified stopping condition is not satisfied, return to Step 2.

In our fuzzy GBML algorithm, the fuzzy rule R_k in (1) is coded by its n antecedent and single consequent fuzzy sets as $R_k = A_{k1}A_{k2} \cdots A_{kn}B_k$. A rule set S is represented by a concatenated string where each substring of the length $(n+1)$ corresponds to a single fuzzy rule. Initial rule sets are generated by randomly assigning a linguistic term or *don't care* to A_{ki} and a linguistic term to B_k.

From the current population, two parent rule sets are selected according to their fitness values. We use the binary tournament selection where two rule sets are randomly drawn with replacement from the current population and the better one with the higher fitness value is chosen as a parent. The binary tournament selection is iterated for selecting a pair of parent rule sets.

Since the number of fuzzy rules is minimized in our fuzzy GBML algorithm, the string length is not fixed. The number of fuzzy rules in each rule set is modified by a crossover operation, which generates a new string whose length is different from its parent strings. We use a kind of one-point crossover with different cutoff points illustrated in Fig. 3 where R_k denotes a substring of the length $(n+1)$. One of the two children in Fig. 3 is randomly selected as a new rule set while it is also possible to use both children. The crossover operation is applied to each pair of selected parents with a pre-specified crossover probability. When the crossover operation is not applied, one of the two parents is handled as a new rule set. The crossover operation in Fig. 3 can be viewed as a special form of the cut and splice crossover used in messy genetic algorithms (see [5] for details of the cut and splice crossover). For efficiently searching for compact rule sets, we use a heuristic procedure after the crossover operation. The heuristic procedure imposes an upper bound on the number of fuzzy rules in each rule set. In our computer simulations, only the first 20 fuzzy rules from the left of each string are used and the other rules are removed from the string when the number of fuzzy rules exceeds 20.

A mutation operation is applied with a pre-specified mutation probability after the crossover operation. Our mutation operation randomly replaces each antecedent (and consequent) fuzzy set with another one. It should be noted that *don't care* is used only in the antecedent part. We also use a different kind of mutation, which randomly removes each fuzzy rule from the rule set with a pre-specified probability. We can also use heuristic-based mutation operations for improving the search ability of our fuzzy GBML algorithm. For example, the consequent fuzzy set of each fuzzy rule is probabilistically replaced with more appropriate one using compatible input-output pairs with its antecedent part. Moreover a new fuzzy rule can be directly generated from an input-output pair with the largest error in a heuristic manner and added to a rule set. See [16] for details of these heuristic-based mutation operations.

Let N_{pop} be the population size. The selection, crossover and mutation are iterated for generating $(N_{\text{pop}} - 1)$ rule sets as a new population. The best rule set with the largest fitness value in the current population is added to the generated new population with no modifications as an elite rule set.

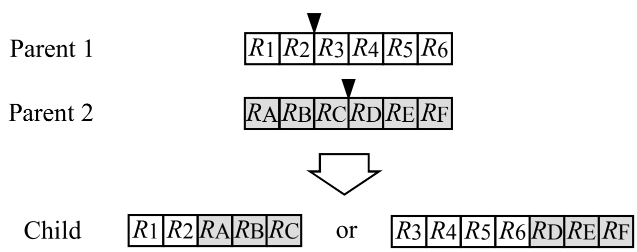

Fig. 3. A kind of one-point crossover with different cutoff points

4 Three-Objective Fuzzy GBML Algorithm

4.1 Problem Specification

Our task is to find all non-dominated rule sets (i.e., Pareto-optimal solutions) with respect to the three objectives when linguistic modelling is handled in the framework of multi-objective optimization. First we briefly describe the concept of Pareto-optimality. A rule set S_A is said to dominate another rule set S_B (i.e., S_A is better than $S_B : S_A \prec S_B$) if all the following three inequalities hold:

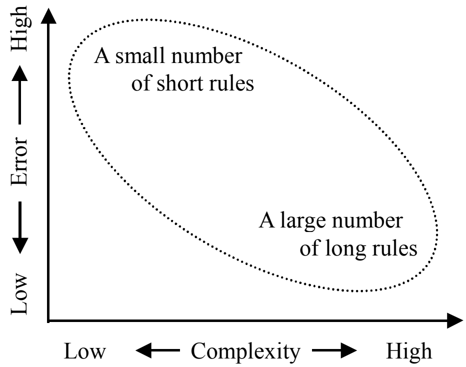

Fig. 4. Illustration of the tradeoff between the error and the complexity of rule sets

$$f_1(S_A) \le f_1(S_B),\ \ f_2(S_A) \le f_2(S_B),\ \ f_3(S_A) \le f_3(S_B), \tag{14}$$

and at least one of the following three inequalities holds:

$$f_1(S_A) < f_1(S_B),\ \ f_2(S_A) < f_2(S_B),\ \ f_3(S_A) < f_3(S_B). \tag{15}$$

The first condition (i.e., all the three inequalities in (14)) means that no objective of S_A is worse than S_B. The second condition (i.e., one of the three inequalities in (15)) means that at least one objective of S_A is better than S_B. When a rule set S is not dominated by any other rule sets, S is said to be a Pareto-optimal solution with respect to the three objectives. Our three-objective linguistic modelling problem is to find all Pareto-optimal solutions. Since there exists a tradeoff between the accuracy and the complexity of linguistic models [3], our linguistic modelling problem has many Pareto-optimal solutions with different accuracy and different complexity. The tradeoff between the error and the complexity of rule sets is illustrated in Fig. 4.

4.2 Multi-objective Fuzzy GBML Algorithm

Our Pittsburgh-style fuzzy GBML algorithm in the previous section can be extended to the case of three-objective optimization as in [13, 14] for fuzzy rule-based pattern classification. Recently many multi-objective genetic algorithms (MOGAs) have been proposed [4, 6] together with various performance measures [25]. Since most MOGAs are general-purpose search algorithms, they can be used for finding Pareto-optimal solutions of our three-objective linguistic modelling problem.

The main difference between single-objective and multi-objective fuzzy GBML algorithms is the fitness calculation for each rule set. The fitness calculation was very simple in the previous section because the three objectives were integrated into the scalar fitness function using the user-definable weight values. On the other hand, we do not assume any *a priori* knowledge about the relative importance of the three objectives in this section. Thus the fitness value of each rule set is calculated based on the Pareto-dominance relation defined by (14)-(15). Larger fitness values are usually assigned to non-dominated rule sets than dominated ones. For maintaining the diversity of solutions (i.e., finding a variety of Pareto-optimal solutions), the concept of fitness sharing or crowding is also used in the fitness calculation. A larger fitness value is usually assigned to a rule set that is less similar to other rule sets. The concept of elite solutions should be modified in multi-objective optimization. While the best solution with the largest fitness value was used as an elite solution in the previous section, each non-dominated solution in the current population can be viewed as an elite solution. For various implementations of MOGAs, see [4, 6].

Some MOGAs have a secondary population where tentative non-dominated solutions are stored separately from the current population. The secondary population is updated by comparing it with the current population in every generation. When we use a MOGA with a secondary population, the outline of our single-objective fuzzy GBML algorithm in the previous section is extended to the case of multi-objective optimization as follows:

[Outline of Pittsburgh-Style Multi-Objective Fuzzy GBML Algorithm]

Step 1: Randomly generate a number of rule sets as an initial population. A copy of each non-dominated rule set in the initial population is included in the secondary population.

Step 2: Repeat the following procedures for generating new rule sets.
(a) Select a pair of parent rule sets from the current population.
(b) Generate a new rule set from the selected pair by a crossover operation.
(c) Apply mutation operations to the generated rule set.

Step 3: Update the secondary population using the newly generated rule sets in Step 2. Generate a new population using the current population, the newly generated rule sets, and the secondary population.

Step 4: If a pre-specified stopping condition is not satisfied, return to Step 2.

When the execution is terminated, non-dominated rule sets stored in the secondary population are presented to human users as solutions of the three-objective linguistic modelling problem. Those rule sets are used for examining the tradeoff between the accuracy and the interpretability of linguistic models. When a single linguistic model should be chosen, the choice depends on the preference of human users. In general, the choice of a single linguistic model from multiple non-dominated ones is much easier than the pre-specification of the weight value to each objective.

5 Genetic Rule Selection

5.1 Basic Idea of Genetic Rule Selection

The design of a linguistic model for an n-input and single-output nonlinear function can be viewed as finding a subset of $(K+1)^n \cdot K$ fuzzy rules where K is the number of linguistic terms given for each variable. When n is small, we can handle linguistic modelling as a rule selection problem where a small number of fuzzy rules are selected from $(K+1)^n \cdot K$ candidate rules. Single-objective and multi-objective genetic algorithms are directly applicable to such a rule selection problem because each rule set is naturally represented by a binary string of the length $(K+1)^n \cdot K$. The size of the search space is 2^N where $N = (K+1)^n \cdot K$.

Genetic rule selection was originally proposed for fuzzy rule-based classification by Ishibuchi et al. [15] where the weighted sum of the classification accuracy and the number of fuzzy rules was used as a fitness function. Their study was extended to two-objective rule selection in [11] and three-objective rule selection in [13, 14]. Since the number of candidate rules exponentially increases with the number of input variables, the computational load and the memory storage for genetic rule selection also exponentially increase. As a result, genetic rule selection is much slower than fuzzy GBML algorithms as shown in [16] except for the case of low-dimensional problems. Moreover, it is impractical to use all the $(K+1)^n \cdot K$ fuzzy rules as candidate rules when the number of input variables is large (i.e., when n is large).

5.2 Heuristic Rule Generation Using Data Mining Criteria

When the number of input variables is small, we can use all the $(K+1)^n \cdot K$ fuzzy rules as candidate rules in genetic rule selection. On the other hand, we need some prescreening procedure of candidate rules in the application of genetic rule selection to high-dimensional problems. We proposed the use of heuristic rule evaluation criteria for candidate rule prescreening in [17] for fuzzy rule-based classification. More specifically, two rule evaluation measures (i.e., *support* and *confidence*) were employed for evaluating fuzzy rules. The proposed idea can be also used for linguistic modelling. The two rule evaluation measures, which were originally used for evaluating association rules in the area of data mining [1], were extended to the case of fuzzy rules in [8, 18].

The confidence $c(R_k)$ of the fuzzy rule R_k in (1) is defined using the given m input-output pairs $(\mathbf{x}_p, y_p)$, $p = 1,2,...,m$ as

$$c(R_k) = \frac{\sum_{p=1}^{m} \mu_{\mathbf{A}_k}(\mathbf{x}_p) \cdot \mu_{B_k}(y_p)}{\sum_{p=1}^{m} \mu_{\mathbf{A}_k}(\mathbf{x}_p)}, \tag{16}$$

where $\mu_{\mathbf{A}_k}(\mathbf{x}_p)$ is the compatibility grade of the input vector $\mathbf{x}_p$ with the antecedent part $\mathbf{A}_k = (A_{k1}, ..., A_{kn})$ of the fuzzy rule R_k, and $\mu_{B_k}(y_p)$ is the compatibility grade of the output value y_p with the consequent part B_k of R_k. The denominator of (16) corresponds to the number of input-output pairs that are compatible with the antecedent part $\mathbf{A}_k$ of the fuzzy rule R_k. The numerator corresponds to the number of input-output pairs that are compatible with both the antecedent and consequent parts of R_k.

The support $s(R_k)$ of the fuzzy rule R_k is defined as

$$s(R_k) = \frac{\sum_{p=1}^{m} \mu_{\mathbf{A}_k}(\mathbf{x}_p) \cdot \mu_{B_k}(y_p)}{m}. \tag{17}$$

When both the antecedent and consequent parts of the fuzzy rule R_k are specified by non-fuzzy concepts, these two definitions in (16) and (17) are exactly the same as those used for non-fuzzy association rules in data mining [1].

The two rule evaluation measures are employed for extracting a pre-specified number of candidate rules in various manners. For example, we can use one of the following rule extraction criteria:

(1) Choose candidate rules using the confidence measure.
(2) Choose candidate rules using the support measure.
(3) Choose candidate rules using the confidence measure from fuzzy rules whose support values are not less than a pre-specified minimum support level.
(4) Choose candidate rules using the support measure from fuzzy rules whose confidence values are not less than a pre-specified minimum confidence level.
(5) Choose candidate rules using a composite criterion of the confidence and support measures. A simple example of such a composite criterion is their product.

The length of fuzzy rules can be used as a constraint condition on candidate rules. That is, candidate rules are chosen using a rule extraction criterion from fuzzy rules that are shorter than or equal to a pre-specified maximum length. The use of the upper bound on the length of candidate rules is consistent with the third objective of our linguistic modelling problem (i.e., minimization of the total rule length).

5.3 Genetic Algorithms for Rule Selection

Let us assume that we have N candidate rules for genetic rule selection. Any subset S of those candidate rules is denoted by a binary string of the length N as

$$S = s_1 s_2 \cdots s_N, \tag{18}$$

where $s_j = 1$ and $s_j = 0$ mean that the jth candidate rule is included in S and excluded from S, respectively.

When the weight values for the three objectives are given from domain experts or human users, we can use the weighted sum in (13) as a scalar fitness function. In this case, we can use standard genetic algorithms for finding the optimal rule set that maximizes the scalar fitness function. On the other hand, genetic rule selection is performed using multi-objective genetic algorithms [4, 6] when no *a priori* knowledge is given for the relative importance of the three objectives.

As shown in (18), the length of the binary string S is N (i.e., the number of candidate rules). Thus the size of the search space is 2^N. This means that long computation time and large memory storage are needed for executing genetic rule selection when the number of candidate rules is large. Two heuristic procedures were used for improving the efficiency of genetic rule selection for fuzzy rule-based classification [11, 13, 14, 15, 17]. One is the use of biased mutation where a larger mutation probability is assigned to the mutation from 1 to 0 than that from 0 to 1. The biased mutation is for efficiently decreasing the number of fuzzy rules in each rule set. The other is the removal of unnecessary fuzzy rules. If the antecedent part of a fuzzy rule is not compatible with any input-output pair, we can remove the fuzzy rule without deteriorating the approximation accuracy. At the same time, the removal of such an unnecessary rule improves the second and third objectives of our linguistic modelling problem. While the efficiency of genetic rule selection mainly depends on the choice of candidate rules, the biased mutation and the removal of unnecessary rules also improve the search ability to efficiently find good rule sets.

6 Modification of Fuzzy Reasoning

6.1 Computer Simulations on Simple Numerical Examples

As a test problem, we generated 9261 input-output pairs $(x_{p1}, x_{p2}, x_{p3}, y_p)$, $p = 1$, 2, ..., 9261, from the three-input and single-output nonlinear function in (3) using the $21 \times 21 \times 21$ uniform grid of the input space $[0, 1]^3$: $x_{pi} = 0.00, 0.05, 0.10, ..., 1.00$ for $i = 1,2,3$. The five linguistic terms in Fig. 1 were used for all the three input and single output variables. We also used *don't care* as an additional antecedent fuzzy set.

We assumed that the following scalar fitness function was given:

$$f(S) = -100 f_1(S) - f_2(S) - f_3(S)\,. \tag{19}$$

We used the fuzzy GBML algorithm in Section 3 for finding the optimal rule set with respect to this scalar fitness function. As explained in Section 3, the heuristic procedure with the upper bound on the number of fuzzy rules (i.e., 20 rules) was used. The other heuristic procedures were not utilized in computer simulations. Our fuzzy GBML algorithm was executed under the following parameter specifications:

Population size: 200,
The number of fuzzy rules in each initial rule set: 10,
Crossover probability: 0.8,
Mutation probability for replacing each fuzzy set with another one: 0.1,
Mutation probability for removing each fuzzy rule: 0.1,
Stopping condition: 5000 generations.

We applied our fuzzy GBML algorithm to the generated 9261 input-output pairs 10 times using different initial populations. A rule set with the following two rules was obtained from 4 out of 10 runs:

$$R_{\mathrm{A}}: y \text{ is } \textit{small}, \tag{20}$$

$$R_{\mathrm{B}}: \text{If } x_1 \text{ is } \textit{large} \text{ and } x_2 \text{ is } \textit{large} \text{ and } x_3 \text{ is } \textit{large} \text{ then } y \text{ is } \textit{large}. \tag{21}$$

The total squared error over the 9261 input-output pairs was 1.045. Thus we can see that the accuracy of the rule set with R_{A} and R_{B} is high. In all the other six runs, these two rules were obtained as a part of larger rule sets with additional fuzzy rules. The total squared error of those larger rule sets was slightly better than the case of the rule set with the above two fuzzy rules. Actually it was between 0.738 and 0.968. These simulation results were much more sensitive to the specification of the weight values in (19) than the above parameter values in our fuzzy GBML algorithm.

We can easily understand a rough shape of the nonlinear function from the above two fuzzy rules. That is, one may think that the output value is *small* except for the region with *large* x_1, *large* x_2 and *large* x_3 where the output value is *large*.

Using the confidence measure, we evaluated the two fuzzy rules. The confidence value of each fuzzy rule was calculated as follows:

$$c(R_{\mathrm{A}}) = 0.994 \text{ and } c(R_{\mathrm{B}}) = 0.000. \tag{22}$$

We can see that the confidence value of R_{B} is very small while that of R_{A} is large. For comparison, we also calculated the confidence values of the following fuzzy rules that have the same antecedent part as R_{B} but different consequent fuzzy sets:

R_{C} : If x_1 is *large* and x_2 is *large* and x_3 is *large* then y is *small*,
R_{D} : If x_1 is *large* and x_2 is *large* and x_3 is *large* then y is *medium small*,
R_{E} : If x_1 is *large* and x_2 is *large* and x_3 is *large* then y is *medium*,
R_{F} : If x_1 is *large* and x_2 is *large* and x_3 is *large* then y is *medium large*.

The confidence value of each fuzzy rule was calculated as follows:

$$c(R_{\mathrm{C}}) = 0.660,\ c(R_{\mathrm{D}}) = 0.148,\ c(R_{\mathrm{E}}) = 0.192, \text{ and } c(R_{\mathrm{F}}) = 0.000. \tag{23}$$

We can see that the fuzzy rules R_{C}, R_{D} and R_{E} are more compatible with the given input-output pairs than the obtained fuzzy rule R_{B}. That is, the output value is not *large* but *small*, *medium small* or *medium* for input vectors with *large* x_1, *large* x_2 and *large* x_3. In this sense, the obtained fuzzy rule R_{B} in (21) is misleading.

For visually examining why such a misleading fuzzy rule was obtained, we applied our fuzzy GBML algorithm to a two-input and single-output nonlinear function in Fig. 5 in the same manner as the previous computer simulation. We first generated 441 input-output pairs (x_{p1}, x_{p2}, y_p), $p = 1, 2, ..., 441$, from the nonlinear function using the 21×21 uniform grid of the input space $[0, 1]^2$. Then we applied our fuzzy GBML algorithm to the generated 441 input-output pairs 20 times using different initial populations. A rule set with the following three rules was obtained from 17 out of 20 runs:

$$R_{\mathrm{I}} : y \text{ is } small, \tag{24}$$

$$R_{\mathrm{II}} : \text{If } x_1 \text{ is } small \text{ then } y \text{ is } medium, \tag{25}$$

$$R_{\mathrm{III}} : \text{If } x_1 \text{ is } small \text{ and } x_2 \text{ is } small \text{ then } y \text{ is } large. \tag{26}$$

The total squared error was 0. Actually we depicted the nonlinear function in Fig. 5 by applying the fuzzy reasoning method in (10) to the three fuzzy rules in (24)-(26).

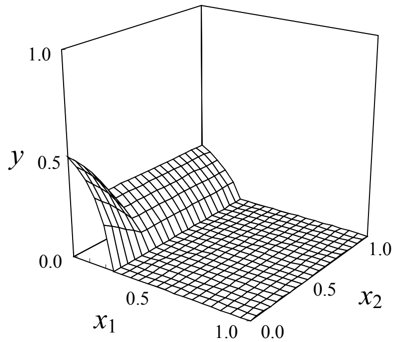

Fig. 5. Nonlinear function to be linguistically described

Let us try to imagine a three-dimensional shape of a nonlinear function from the three fuzzy rules in (24)-(26). From the fuzzy rule R_{III}, we think that the output is *large* for *small* x_1 and *small* x_2. From the fuzzy rule R_{II}, we think that the output is *medium* for *small* x_1. For other input vectors, the output values seem to be *small*. As a result, we may have a three-dimensional shape that is similar to Fig. 6. It should be noted that the intuitively imagined shape is different from Fig. 5 from which the three fuzzy rules in (24)-(26) were derived by our fuzzy GBML algorithm. We further examined the three fuzzy rules using the confidence measure. The confidence value of each fuzzy rule was calculated from the 441 input-output pairs as follows:

$$c(R_{\mathrm{I}}) = 0.819, \quad c(R_{\mathrm{II}}) = 0.078 \quad \text{and} \quad c(R_{\mathrm{III}}) = 0.000. \tag{27}$$

While the first fuzzy rule R_{I} has a large confidence value, the confidence values of the other fuzzy rules are very small. This means that the fuzzy rules R_{II} and R_{III} are not consistent with the given input-output pairs.

Let us explain why the misleading fuzzy rules were obtained for the nonlinear functions in (3) and Fig. 5. In our approach to linguistic modelling, we use *don't care* as an additional antecedent fuzzy set for generating short fuzzy rules and covering the whole input space by a small number of fuzzy rules. As we have already explained in Section 2, the use of *don't care* is necessary for linguistically explaining a nonlinear function using a small number of fuzzy rules. Thus our linguistic model is a mixture of general and specific fuzzy rules. When we intuitively estimate an output value from general and specific rules, specific rules have usually higher priority than general rules. For example, we may mainly use the most specific fuzzy rule R_{III} in (26) when we estimate an output value for *small* x_1 and *small* x_2 using the three fuzzy rules in (24)-(26). In this case, the output is intuitively estimated as *large* (see Fig. 6). On the other hand, most fuzzy reasoning methods are based on the interpolation of compatible fuzzy rules. Thus the estimated output for *small* x_1 and *small* x_2 is usually calculated as *medium* (see Fig. 5). This difference between our intuition and fuzzy reasoning leads to linguistic models with misleading fuzzy rules.

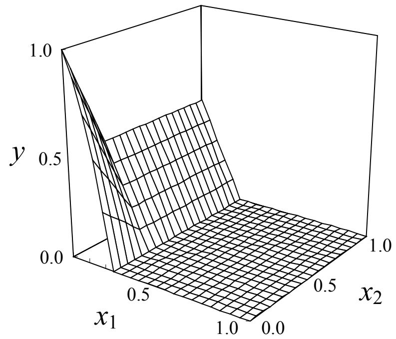

Fig. 6. Nonlinear function that is intuitively depicted from R_{I}, R_{II} and R_{III} in (24)-(26)

6.2 Non-standard Inclusion-Based Fuzzy Reasoning

A non-standard fuzzy reasoning method based on an inclusion relation among fuzzy rules was proposed for obtaining intuitively acceptable fuzzy reasoning results [10]. The proposal of such an inclusion-based fuzzy reasoning method was motivated by the above-mentioned difficulty of standard interpolation-based fuzzy reasoning.

First we explain an inclusion relation using the following two fuzzy rules:

$$R_k : \text{If } x_1 \text{ is } A_{k1} \text{ and ... and } x_n \text{ is } A_{kn} \text{ then } y \text{ is } B_k, \tag{28}$$

$$R_q : \text{If } x_1 \text{ is } A_{q1} \text{ and ... and } x_n \text{ is } A_{qn} \text{ then } y \text{ is } B_q. \tag{29}$$

When the inclusion relation $A_{qi} \subseteq A_{ki}$ holds between the antecedent fuzzy sets for $i = 1,2,...,n$, we say that the fuzzy rule R_q is included in the fuzzy rule R_k (i.e., $R_q \subseteq R_k$). In order to implement the preference for more specific fuzzy rules, the standard fuzzy reasoning method in (10) is modified as

$$\hat{y}(\mathbf{x}_p) = \frac{\sum_{R_k \in S} \phi(R_k, \mathbf{x}_p) \cdot b_k \cdot \mu_{\mathbf{A}_k}(\mathbf{x}_p)}{\sum_{R_k \in S} \phi(R_k, \mathbf{x}_p) \cdot \mu_{\mathbf{A}_k}(\mathbf{x}_p)}, \tag{30}$$

where $\phi(R_k, \mathbf{x}_p)$ is a weight determined by the inclusion relation between R_k and the other fuzzy rules in the rule set *S*. The value of $\phi(R_k, \mathbf{x}_p)$ is small when R_k includes more specific rules compatible with the input vector $\mathbf{x}_p$. In this case, the weight of R_k is discounted in fuzzy reasoning. Actually $\phi(R_k, \mathbf{x}_p)$ is defined using a user-definable non-negative parameter β as

$$\phi(R_k, \mathbf{x}_p) = \prod_{\substack{R_q \subseteq R_k \\ q \neq k}} (1 - \mu_{\mathbf{A}_q}(\mathbf{x}_p))^{\beta}. \tag{31}$$

When no fuzzy rule is included in R_k, $\phi(R_k, \mathbf{x}_p)$ is specified as $\phi(R_k, \mathbf{x}_p) = 1$ because the weight of R_k should not be discounted in this case.

The nonlinear function in Fig. 6 was the fuzzy reasoning result where the inclusion-based fuzzy reasoning method with $\beta = 1$ in (30) was applied to the three fuzzy rules R_{I}, R_{II} and R_{III} in (24)-(26). From the comparison between the three fuzzy rules in (24)-(26) and the three-dimensional shape of the nonlinear function in Fig. 6, we can see that an intuitively acceptable fuzzy reasoning result was obtained by the non-standard inclusion-based fuzzy reasoning method.

6.3 Computer Simulations Using Non-standard Fuzzy Reasoning

In the same manner as Subsection 6.1, we applied our fuzzy GBML algorithm to the nonlinear function in Fig. 5. When the total squared error was calculated, we used the non-standard inclusion-based fuzzy reasoning method with $\beta = 2$ instead of the standard interpolation-based fuzzy reasoning method in (10). The computer simulation was iterated 20 times using different initial populations. A rule set with the following three fuzzy rules was obtained from 15 out of 20 runs:

y is *small*, (32)

If x_1 is *small* then y is *medium small*, (33)

If x_1 is *small* and x_2 is *small* then y is *medium*. (34)

From the comparison between the obtained three fuzzy rules and Fig. 5, we can see that the nonlinear function in Fig. 5 is linguistically described in an intuitively acceptable manner. That is, the obtained fuzzy rules in (32)-(34) are consistent with the three-dimensional shape of the nonlinear function in Fig. 5.

In the same manner as Subsection 6.1, we also applied our fuzzy GBML algorithm to the nonlinear function in (3) using the non-standard fuzzy reasoning method with $\beta = 5$. The computer simulation was iterated 10 times using different initial populations. A rule set with the following two fuzzy rules was obtained from all the 10 runs.

R_{A} : y is *small*, (35)

R_{E} : If x_1 is *large* and x_2 is *large* and x_3 is *large* then y is *medium*. (36)

As we have already explained in Subsection 6.1, these two fuzzy rules have large confidence values (i.e., $c(R_{\mathrm{A}}) = 0.994$ and $c(R_{\mathrm{E}}) = 0.192$). While the misleading fuzzy rule R_{B} with a very small confidence value (i.e., $c(R_{\mathrm{B}}) = 0.000$) was obtained in Subsection 6.1 using the standard interpolation-based fuzzy reasoning method, the fuzzy rule R_{E} with a larger confidence value was obtained using the non-standard inclusion-based fuzzy reasoning method.

7 Concluding Remarks

In this chapter, we discussed linguistic modelling for linguistically describing nonlinear functions in a human understandable manner. That is, we discussed linguistic modelling for obtaining linguistic models with high interpretability as well as high accuracy. We assumed that a set of linguistic terms was given for each variable from domain experts or human users. Thus we did not discuss the interpretability of fuzzy partitions. The interpretability of linguistic models was defined by the number of fuzzy rules and the total rule length. In this context, we explained the validity of the use of the total rule length instead of the average rule length as a complexity measure of linguistic models. We also explained the necessity of the use of *don't care* as an additional antecedent fuzzy set for linguistically describing high-dimensional nonlinear functions using a small number of fuzzy rules with high interpretability.

Linguistic modelling was formulated as a three-objective optimization problem where the total squared error, the number of fuzzy rules and the total rule length were minimized. We explained how the formulated linguistic modelling problem can be handled by single-objective and multi-objective genetic algorithms. We showed two approaches to our linguistic modelling problem: fuzzy genetics-based machine learning and genetic rule selection. Then we pointed out a possibility that misleading fuzzy rules can be obtained from our linguistic modelling problem. Finally we demonstrated that the use of the non-standard inclusion-based fuzzy reasoning method removed such an undesirable possibility.

References

[1] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., and Verkamo, A. I.: Fast Discovery of Association Rules, in Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R. (eds.) *Advances in Knowledge Discovery and Data Mining*, AAAI Press, Metro Park (1996) 307-328.

[2] Carse, B., Fogarty, T. C., and Munro, A.: Evolving Fuzzy Rule Based Controllers Using Genetic Algorithms, *Fuzzy Sets and Systems* 80 (1996) 273-293.

[3] Casillas, J., Cordon, O., Herrera, F., and Magdalena, L. (Eds.): *Trade-off between Accuracy and Interpretability in Fuzzy Rule-Based Modelling*, Physica-Verlag (2002).

[4] Coello Coello, C. A., Van Veldhuizen, D. A., and Lamont, G. B.: *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer Academic Publishers, Boston (2002).

[5] Cordon, O., Herrera, F., Hoffmann, F., and Magdalena, L.: *Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*, World Scientific (2001).

[6] Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms, John Wiley & Sons, Chichester (2001).

[7] Homaifar, A., and McCormick, E.: Simultaneous Design of Membership Functions and Rule Sets for Fuzzy Controllers Using Genetic Algorithms, *IEEE Trans. on Fuzzy Systems* 3 (1995) 129-139.
[8] Hong, T. -P., Kuo, C. -S., and Chi, S. -C.: Trade-off between Computation Time and Number of Rules for Fuzzy Mining from Quantitative Data, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 9 (2001) 587-604.
[9] Horikawa, S., Furuhashi, T., and Uchikawa, Y.: On Fuzzy Modelling using Fuzzy Neural Networks with the Back-Propagation Algorithm, *IEEE Trans. on Neural Networks* 3 (1992) 801-806.
[10] Ishibuchi, H.: Fuzzy Reasoning Method in Fuzzy Rule-based Systems with General and Specific Rules for Function Approximation, *Proc. of 8th IEEE International Conference on Fuzzy Systems* (1999) 198-203.
[11] Ishibuchi, H., Murata, T., and Turksen, I. B.: Single-Objective and Two-Objective Genetic Algorithms for Selecting Linguistic Rules for Pattern Classification Problems, *Fuzzy Sets and Systems* 89 (1997) 135-149.
[12] Ishibuchi, H., Nakashima, T.: Genetic-Algorithm-Based Approach to Linguistic Approximation of Nonlinear Functions with Many Input Variables, *Proc. of 8th IEEE International Conference on Fuzzy Systems* (1999) 779-784.
[13] Ishibuchi, H., Nakashima, T., and Murata, T.: Three-Objective Genetics-Based Machine Learning for Linguistic Rule Extraction, *Information Sciences* 136 (2001) 109-133.
[14] Ishibuchi, H., Nakashima, T., and Murata, T.: Multiobjective Optimization in Linguistic Rule Extraction from Numerical Data, in Zitzler, E., Deb, K., Thiele, L., Coello Coello, C. A., and Corne, D. (eds.) *Evolutionary Multi-Criterion Optimization: Lecture Notes in Computer Science 1993*, Springer-Verlag, Berlin (2001) 588-602.
[15] Ishibuchi, H., Nozaki, K., Yamamoto, N., and Tanaka, H.: Selecting Fuzzy If-Then Rules for Classification Problems Using Genetic Algorithms, *IEEE Trans. on Fuzzy Systems* 3 (1995) 260-270.
[16] Ishibuchi, H., Takeuchi, D., and Nakashima, T.: GA-Based Approaches to Linguistic Modelling of Nonlinear Functions, *Proc. of 9th IFSA World Congress and 20th NAFIPS International Conference* (2001) 1229-1234.
[17] Ishibuchi, H., and Yamamoto, T.: Fuzzy Rule Selection by Data Mining Criteria and Genetic Algorithms, *Proc. of Genetic and Evolutionary Computation Conference* (2002) 399-406.
[18] Ishibuchi, H., Yamamoto, T., and Nakashima, T.: Fuzzy Data Mining: Effect of Fuzzy Discretization, *Proc. of 1st IEEE International Conference on Data Mining* (2001) 241-248.
[19] Jang, J. -S. R.: ANFIS: Adaptive-Network-Based Fuzzy Inference System, *IEEE Trans. on Systems, Man, and Cybernetics* 23 (1993) 665-685.
[20] Jimenez, F., Gomez-Skarmeta, A. F., Roubos, H., and Babuska, R.: Accurate, Transparent and Compact Fuzzy Models for Function Approximation and Dynamic Modelling Through Multi-Objective Evolutionary Optimization, in Zitzler, E., Deb, K., Thiele, L., Coello Coello, C. A., and Corne, D. (eds.) *Evolutionary Multi-Criterion Optimization: Lecture Notes in Computer Science 1993*, Springer-Verlag, Berlin (2001) 653-667.

[21] Jimenez, F., Gomez-Skarmeta, A. F., Roubos, H., and Babuska, R.: A Multi-Objective Evolutionary Algorithm for Fuzzy Modelling, *Proc. of 9th IFSA World Congress and 20th NAFIPS International Conference* (2001) 1222-1228.
[22] Jin, Y.: Fuzzy Modelling of High-dimensional Systems: Complexity Reduction and Interpretability Improvement, *IEEE Trans. on Fuzzy Systems* 8 (2000) 212-221.
[23] Jin, Y., Von Seelen, W., and Sendhoff, B.: On Generating FC^3 Fuzzy Rule Systems from Data Using Evolution Strategies, *IEEE Trans. on Systems, Man and Cybernetics - Part B: Cybernetics* 29 (1999) 829-845.
[24] Karr, C. L., and Gentry, E. J.: Fuzzy Control of pH Using Genetic Algorithms, *IEEE Trans. on Fuzzy Systems* 1 (1993) 46-53.
[25] Knowles, J. D., and Corne, D. W.: On Metrics for Comparing Non-Dominated Sets, *Proc. of 2002 Congress on Evolutionary Computation* (2002) 711-716.
[26] Nozaki, K., Ishibuchi, H., and Tanaka, H.: A Simple but Powerful Heuristic Method for Generating Fuzzy Rules from Numerical Data, *Fuzzy Sets and Systems* 86 (1997) 251-270.
[27] Pedrycz, W.: *Fuzzy Modelling: Paradigms and Practice*, Kluwer Academic Publishers, Boston (1996).
[28] Roubos, H., and Setnes, M.: Compact and Transparent Fuzzy Models and Classifiers Through Iterative Complexity Reduction, *IEEE Trans. on Fuzzy Systems* 9 (2001) 516-524.
[29] Setnes, M., Babuska, R., Kaymak, U., and Van Nauta Lemke, H. R.: Similarity Measures in Fuzzy Rule Base Simplification, *IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics* 28 (1998) 376-386.
[30] Setnes, M., Babuska, R., and Verbruggen, B.: Rule-based Modelling: Precision and Transparency, *IEEE Trans. on Systems, Man, and Cybernetics - Part C: Applications and Reviews* 28 (1998) 165-169.
[31] Sugeno, M., and Yasukawa, T.: A Fuzzy-Logic-Based Approach to Qualitative Modelling, *IEEE Trans. on Fuzzy Systems* 1 (1993) 7-31.
[32] Sun, C. -T.: Rule-Base Structure Identification in an Adaptive-Network-Based Fuzzy Inference System, *IEEE Trans. on Fuzzy Systems* 2 (1994) 64-73.
[33] Takagi, T., and Sugeno, M.: Fuzzy Identification of Systems and Its Applications to Modelling and Control, *IEEE Trans. on Systems, Man, and Cybernetics* 15 (1985) 116-132.
[34] Valente de Oliveira, J.: Semantic Constraints for Membership Function Optimization, *IEEE Trans. on Systems, Man, and Cybernetics - Part A: Systems and Humans* 29 (1999) 128-138.
[35] Wang, L. X., and Mendel, J. M.: Generating Fuzzy Rules by Learning from Examples, *IEEE Trans. on Systems, Man, and Cybernetics* 22 (1992) 1414-1427.