H. Ishibuchi, T. Murata, and I. B. Turksen, " Selecting linguistic classification rules by two-objective genetic algorithms," *Proc. of 1995 IEEE International Conference on Systems, Man and Cybernetics*, pp. 1410-1415, Vancouver, Canada, October 1995.

Selecting Linguistic Classification Rules by Two-Objective Genetic Algorithms

Hisao Ishibuchi, Tadahiko Murata
Department of Industrial Engineering, University of Osaka Prefecture
Gakuen-cho 1-1, Sakai, Osaka 593, Japan

I.B.Türkşen
Department of Industrial Engineering, University of Toronto
Toronto, Ontario, M5S 1A4, Canada

ABSTRACT

In this paper, we show how two-objective genetic algorithms can be applied to a rule selection problem of linguistic classification rules. First we briefly describe a generation method of linguistic classification rules from numerical data. Next we formulate a rule selection problem of linguistic classification rules. This problem has two objectives: to maximize the number of correctly classified training patterns and to minimize the number of selected rules. Then we propose a two-objective genetic algorithm for finding non-dominated solutions of the rule selection problem. Last we extend our two-objective genetic algorithm to a hybrid algorithm where a learning method is applied to each individual (*i.e.*, each rule set) generated in the execution of the two-objective genetic algorithm.

1. INTRODUCTION

Fuzzy systems based on fuzzy if-then rules have been applied to various problems (for example, see Lee [1]). One advantage of fuzzy-rule-based systems is their clarity. Human users of such systems can easily understand each fuzzy if-then rule because its antecedent and consequent are related to linguistic values such as "small", "medium" and "large". The number of fuzzy if-then rules is also closely connected to the clarity of fuzzy systems. If a single fuzzy system consists of thousands of fuzzy if-then rules, it is difficult for human users to carefully examine each rule. Therefore we should choose a small number of fuzzy if-then rules for constructing a fuzzy system that is easily understood by human users.

Recently a genetic-algorithm-based approach [2-3] was proposed for constructing a compact fuzzy classification system with a small number of fuzzy if-then rules. In that approach, a rule selection problem with the following two objectives was formulated:

(i) To maximize the number of correctly classified training patterns.
(ii) To minimize the number of selected rules.

These two objectives were combined into a single scalar fitness function, and genetic algorithms were applied to the rule selection problem in [2-3].

Because various antecedent fuzzy sets were used in [2-3], the linguistic interpretation of selected fuzzy if-then rules was not always easy. Thus we have already formulated a rule selection problem of linguistic classification rules in [4]. A single-objective genetic algorithm was employed to select linguistic classification rules in a similar manner as in [2-3].

The main aim of this paper is to extend single-objective genetic algorithms in [2-4] to two-objective genetic algorithms for finding non-dominated solutions of the rule selection problem of linguistic classification rules. In this paper, we employ the following linguistic rules for classification problems in an n-dimensional pattern space.

$$\text{Rule } R_j : \text{If } x_1 \text{ is } A_{j1} \text{ and } \ldots \text{ and } x_n \text{ is } A_{jn} \text{ then Class } C_j \text{ with } CF_j, \quad (1)$$

where R_j is a label of the rule, $\mathbf{x} = (x_1, \ldots, x_n)$ is a pattern vector in the n-dimensional pattern space, A_{ji} is an antecedent fuzzy set with a linguistic label, C_j is a consequent class, and CF_j is the grade of certainty of this rule. This rule can be generated from numerical data [4] and the grade of certainty CF_j of this rule can be adjusted by a learning method [5].

In this paper, first we describe a generation method of linguistic classification rules from numerical data [4]. Next we formulate a rule selection problem of linguistic classification rules as a two-objective optimization problem. Then we propose a two-objective genetic algorithm for finding non-dominated solutions of this problem. Last we combine a learning method [5] of linguistic classification rules into our two-objective genetic algorithm.

2. GENERATING LINGUISTIC RULES

In this section, we describe a generation method of linguistic classification rules from numerical data [4]. This method is basically the same as that of fuzzy if-then rules for classification problems in [2-3].

A. Classification Problems

Let us consider a classification problem in an n-dimensional pattern space $[0,1]^n$. It is assumed that m patterns $\mathbf{x}_p = (x_{p1}, x_{p2}, ..., x_{pn})$, $p = 1,2,...,m$, are given as training data from c classes (Class 1, Class 2, ..., Class c). Fig.1 shows an example of the classification problem with $n = 2$, $c = 2$ and $m = 121$.

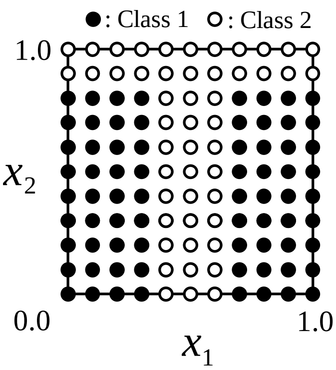

Fig.1 A two-class classification problem in the two-dimensional pattern space $[0,1]^2$

B. Fuzzy Partition

In this paper, we use six fuzzy sets with linguistic labels in the antecedent part of linguistic classification rules. Five of them are shown in Fig.2, which are related to the five linguistic labels: small, medium small, medium, medium large and large, respectively.

The other fuzzy set is the unit interval [0,1] that is used for denoting a "don't care" attribute. Therefore we assign the linguistic label "don't care" to the unit interval. The membership function of the unit interval (*i.e.,* the linguistic label "don't care") is defined as

$$\mu_{don't\ care}(x) = \begin{cases} 1, & \text{if } 0 \le x \le 1, \\ 0, & \text{otherwise}. \end{cases} \quad (2)$$

This membership function is always equal to 1 for all the possible values of any attribute because we consider the classification problem in the n-dimensional unit hyper-cube $[0,1]^n$. Therefore the attributes with "don't care" in each linguistic rule are negligible.

For example, the first attribute x_1 is negligible in the linguistic rule:

If x_1 is don't care and x_2 is small then Class 1.

Actually this linguistic rule is the same as the following linguistic rule with no condition on x_1:

If x_2 is small then Class 1.

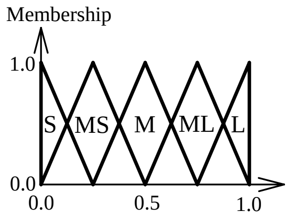

Fig.2 Membership functions of the five fuzzy sets (S: Small, MS: Medium Small, M: Medium, ML: Medium Large and L: Large)

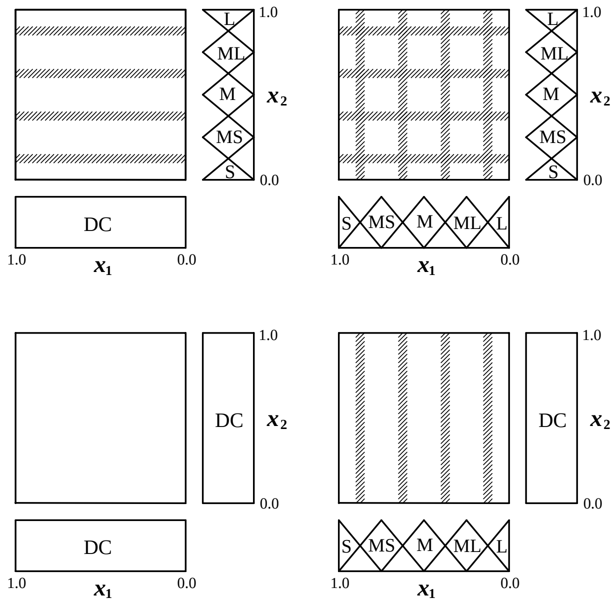

Fig.3 Fuzzy partitions of the two-dimensional pattern space corresponding to the 36 linguistic rules (DC: Don't Care)

Because we use the six fuzzy sets for each axis (*i.e.,* for each attribute) of the n-dimensional pattern space $[0,1]^n$, we can generate 6^n linguistic rules. For example, we can generate $6^2 = 36$ linguistic rules for the two-dimensional pattern space $[0,1]^2$. In Fig.3, we show the fuzzy partitions of the pattern space corresponding to these 36 linguistic rules. From Fig.3, we can see that several linguistic rules are overlapping in the pattern space. This means that some of the 36 linguistic rules in Fig.3 may be redundant for the classification task.

C. Rule Generation

Let us denote the 6^n linguistic rules for the n-dimensional pattern space $[0,1]^n$ as

Rule R_j: If x_1 is A_{j1} and ... and x_n is A_{jn} then Class C_j with CF_j, j=1,2,...,r, (3)

where r is the number of linguistic rules (*i.e.*, $r = 6^n$). Since the antecedent fuzzy sets A_{ji}'s are given as in Fig.3, our rule generation is to determine the consequent class C_j and the grade of certainty CF_j of each linguistic rule.

The consequent C_j and the grade of certainty CF_j of each linguistic rule can be determined from the given patterns $\boldsymbol{x}_p = (x_{p1}, x_{p2}, ..., x_{pn})$, $p = 1,2,...,m$ as in [2-4]. First let us define the grade of compatibility of $\boldsymbol{x}_p$ to the j-th linguistic rule R_j in (3) as

$$\mu_j(\boldsymbol{x}_p) = \mu_{A_{j1}}(x_{p1}) \cdot \ ... \ \cdot \mu_{A_{jn}}(x_{pn}), \qquad (4)$$

where $\mu_{A_{ji}}(x_{pi})$ is the membership function of the antecedent fuzzy set A_{ji}. Thus the total grade of compatibility to the j-th linguistic rule R_j is calculated for each class as

$$\beta_{\text{Class } h} = \sum_{\boldsymbol{x}_p \in \text{Class } h} \mu_j(\boldsymbol{x}_p) = \sum_{\boldsymbol{x}_p \in \text{Class } h} \mu_{A_{j1}}(x_{p1}) \cdot \ ... \ \cdot \mu_{A_{jn}}(x_{pn}), \quad h = 1,2,...,c, \qquad (5)$$

where $\beta_{\text{Class } h}$ is the total grade of compatibility of the given patterns in Class h to the j-th linguistic rule in (3).

The consequent C_j of the j-th linguistic rule R_j is determined as the class with the maximum total grade of compatibility. That is, C_j is determined as Class $\hat{h}$ by

$$\beta_{\text{Class } \hat{h}} = \max\{\beta_{\text{Class } 1}, \beta_{\text{Class } 2}, ..., \beta_{\text{Class } c}\}. \qquad (6)$$

If Class $\hat{h}$ is not determined uniquely (*i.e.*, if two or more classes have the same maximum value in (6)), we assign ϕ to C_j where ϕ means an empty class. For example, the consequent class C_j is determined as Class 1 in Fig.4(a)~(c) while ϕ is assigned to C_j in Fig.4(d). The consequent C_j also becomes ϕ when $\beta_{\text{Class } h} = 0$ for all classes. This means that a linguistic rule with ϕ in the consequent part is generated when there are no patterns compatible with that rule. In this paper, linguistic rules with ϕ in the consequent part are referred to as "dummy rules" because those rules have no effect on the classification phase of new patterns.

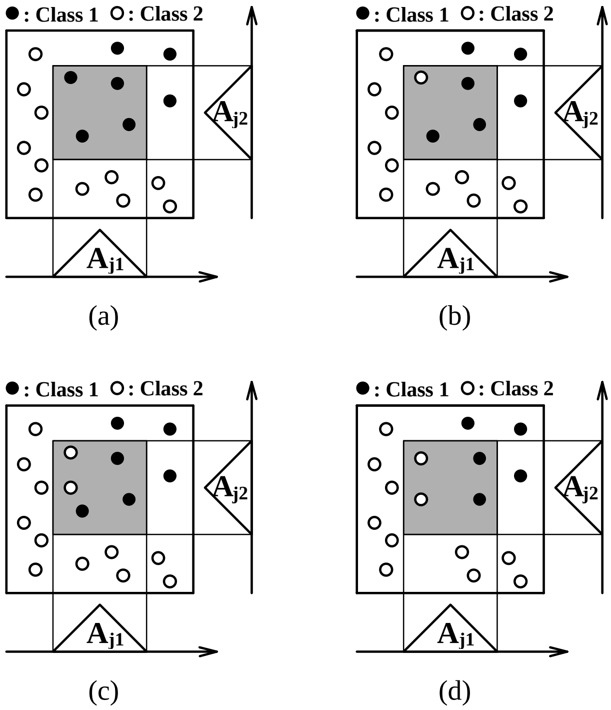

Fig.4 Antecedent fuzzy sets and given patterns

The grades of certainty of all dummy rules are specified as $CF_j = 0$. For non-dummy rules, the grade of certainty CF_j is determined as

$$CF_j = \frac{\beta_{\text{Class } \hat{h}} - \bar{\beta}}{\sum_{h=1}^{c} \beta_{\text{Class } h}}, \qquad (7)$$

where

$$\bar{\beta} = \Big(\sum_{h \neq \hat{h}} \beta_{\text{Class } h}\Big) / (c-1). \qquad (8)$$

The grade of certainty CF_j is maximum (*i.e.*, $CF_j = 1$) when $\beta_{\text{Class } \hat{h}} > 0$ and $\beta_{\text{Class } h} = 0$ for $h \neq \hat{h}$. That is, if all the patterns compatible with the j-th linguistic rule R_j belong to the same class, the grade of certainty CF_j of this rule is equal to 1 (the maximum certainty). On the contrary, if the total grades of compatibility for the c classes are similar to one another (*i.e.*, $\beta_{\text{Class } 1} \cong \ ... \ \cong \beta_{\text{Class } c}$), the grade of certainty is nearly equal to 0 (the minimum certainty). Among the four situations in Fig.4, the grade of certainty CF_j is maximum in Fig.4(a), and it is minimum in Fig.4(d). The grade of certainty CF_j in Fig.4(b) is larger than that in Fig.4(c).

By applying the rule generation procedure to all the linguistic rules in (3), we have r ($r = 6^n$) linguistic rules including dummy rules. Let us denote the set of the generated *r* linguistic rules by S_{ALL} :

$$S_{ALL} = \{\text{Rule } R_j \mid j = 1,2,...,r\}. \tag{9}$$

D. Classification of New Patterns

Let us denote a subset of the rule set S_{ALL} by *S*. A new pattern $\mathbf{x}_p = (x_{p1},...,x_{pn})$ is classified by linguistic rules in *S* as follows [2-4].

Step 1: Calculate $\alpha_{\text{Class } h}$ for $h = 1, 2, ..., c$ as

$$\alpha_{\text{Class } h} = \max\{\mu_j(\mathbf{x}_p) \cdot CF_j \mid C_j = \text{Class } h \text{ and Rule } R_j \in S\}, \tag{10}$$

where $\mu_j(\mathbf{x}_p)$ is the grade of compatibility of $\mathbf{x}_p$ to the *j*-th linguistic rule R_j, which is defined by (4).

Step 2: Find the maximum value of $\alpha_{\text{Class } h}$'s as

$$\alpha_{\text{Class } \hat{h}} = \max\{\alpha_{\text{Class } 1}, ..., \alpha_{\text{Class } c}\}. \tag{11}$$

If two or more classes take the same maximum value in (11), then the classification of $\mathbf{x}_p$ is rejected (*i.e.*, $\mathbf{x}_p$ is left as an unclassifiable pattern), else assign $\mathbf{x}_p$ to Class $\hat{h}$ determined by (11).

In this procedure, a new pattern $\mathbf{x}_p = (x_{p1},...,x_{pn})$ is classified by the linguistic rule that has the maximum product of $\mu_j(\mathbf{x}_p)$ and CF_j.

3. LINGUISTIC RULE SELECTION BY A TWO-OBJECTIVE GENETIC ALGORITHM

A. Problem Formulation

Our rule selection problem is to select a small number of linguistic rules from the rule set S_{ALL} to construct a compact classification system S with high classification power. Therefore our problem can be written as follows:

$$\text{Maximize } NCP(S) \text{ and minimize } |S|, \tag{12}$$

$$\text{subject to } S \subseteq S_{ALL}, \tag{13}$$

where *NCP*(*S*) is the number of correctly classified training patterns by linguistic rules in a rule set *S*, and $|S|$ is the number of the linguistic rules in *S*.

B. Two-Objective Genetic Algorithm

In our former work [4], we applied a single-objective genetic algorithm to the rule selection problem by combining the two objectives into a scalar fitness function:

$$f(S) = W_{NCP} \cdot NCP(S) - W_S \cdot |S|, \tag{14}$$

where W_{NCP} and W_S are positive constant weights. Because the weight for each objective in the fitness function is constant, the search direction of the genetic algorithm in [4] is also constant as shown in Fig.5. This means that the choice of the weight values in (14) has a significant effect on the final solution (*i.e.*, rule set *S*) obtained by the genetic algorithm. Because the importance of each objective in the rule selection problem depends on the preference of human users, it is not easy to assign constant values to the weights W_{NCP} and W_S. Therefore we propose a two-objective genetic algorithm to find multiple non-dominated solutions of the two-objective rule selection problem in (12)-(13). Human users will choose a final solution (*i.e.*, rule set *S*) from the obtained non-dominated solutions.

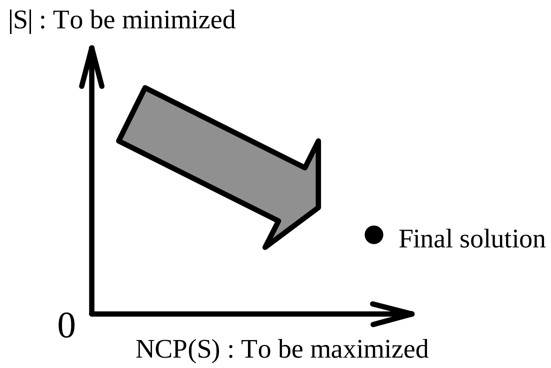

Fig.5 Search direction of a single-objective genetic algorithm

As in [4], each rule set *S* is treated as an individual in our two-objective genetic algorithm. Each rule set *S* (*i.e.*, each individual) is represented by a string as $S = s_1 s_2 ... s_r$ where *r* is the number of all the linguistic rules in S_{ALL} and $s_j = 1, -1 \text{ or } 0$ means the following:

$s_j = 1$ means that the *j*-th rule is included in the rule set *S*,
$s_j = -1$ means that the *j*-th rule is not included in *S*,
$s_j = 0$ means that the *j*-th rule is a dummy rule.

Since dummy rules have no effect on the classification phase of new patterns, they should be excluded from a rule set *S*. Therefore the special coding $s_j = 0$ is assigned to them. A string $S = s_1 s_2 ... s_r$ is decoded as

$$S = \{\text{Rule } R_j \mid s_j = 1;\ j = 1,2,\ldots,r\}. \tag{15}$$

Our two-objective genetic algorithm differs from single-objective algorithms in its selection procedure and elitist strategy. The selection probability in our two-objective genetic algorithm is specified according to the fitness function $f(S)$ in (14) with randomly specified weight values. That is, when each pair of parent individuals are selected, the values of the weights W_{NCP} and W_S are assigned as

W_{NCP} : a random real number in [0, 1], (16)

W_S : $W_S = 1 - W_{NCP}$. (17)

In our two-objective genetic algorithm, multiple solutions are preserved from the current generation to the next generation as elite solutions. Those elite solutions are randomly selected from a tentative set of non-dominated solutions that is stored and updated at each generation of our two-objective genetic algorithm. Multiple search directions in Fig.6 are realized by the selection procedure with random weight values and the elitist strategy with multiple elite solutions.

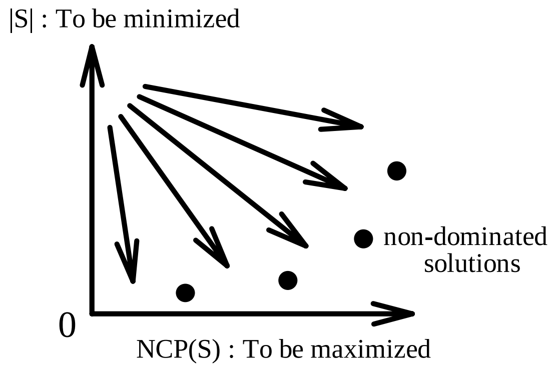

Fig.6 Search directions of our two-objective genetic algorithm

The outline of our two-objective genetic algorithm can be written as follows:

Step 0 (Initialization): Generate an initial population containing N_{pop} strings where N_{pop} is the number of strings in each population.

Step 1 (Evaluation): Calculate the values of the two objectives for the generated strings. Update the tentative set of non-dominated solutions.

Step 2 (Selection): Calculate the fitness value of each string using random weight values. Select a pair of strings from the current population according to the following selection probability. The selection probability $P(S)$ of a string S in a population Ψ is specified as

$$P(S) = \frac{f(S) - f_{\min}(S)}{\sum_{S \in \Psi}\{f(S) - f_{\min}(S)\}} , \quad (18)$$

where

$$f_{\min}(S) = \min\{f(S) \mid S \in \Psi\} . \quad (19)$$

This procedure is repeated for selecting $N_{pop}/2$ pairs of parent strings.

Step 3 (Crossover): For each selected pair, apply a crossover operation to generate two strings.

Step 4 (Mutation): For each bit value of the generated strings by the crossover operation, apply a mutation operation with a pre-specified mutation probability.

Step 5 (Elitist strategy): Randomly remove N_{elite} strings from the generate N_{pop} strings, and add N_{elite} solutions that are randomly selected from the tentative set of non-dominated solutions.

Step 6 (Termination test): If a pre-specified stopping condition is not satisfied, return to Step 1.

C. Computer Simulation

We applied the two-objective genetic algorithm to the well-known iris data (see, for example, Fisher [6]) for selecting linguistic rules. The classification problem of the iris data is a three-class problem with four attributes. In each class, 50 patterns are given (total 150 patterns). Since the iris data has four attributes, $6^4 = 1296$ linguistic rules were generated as candidate rules.

By the two-objective genetic algorithm, non-dominated solutions in Table 1 were obtained. In Table 1, 146 patterns (97.3%) are correctly classified by six linguistic rules, 145 patterns (96.7%) by five linguistic rules and so on. If the human user prefers a high classification rate, he/she would choose the rule set with six linguistic rules in Table 1. On the contrary, if the human user prefers the compactness of rule sets to the high classification performance, he/she would choose the rule set with three linguistic rules in Table 1. Those three rules are shown in Fig.7. We can linguistically interpret the three rules in Fig.7 as follows by ignoring "don't care" attributes:

If x_1 is medium small and x_4 is small then Class 1,

If x_3 is medium then Class 2,

If x_4 is medium large then Class 3.

Table 1 Non-dominated solutions obtained by the two-objective genetic algorithm

$NCP(S)$	146	145	144	141	99	50	0
$\|S\|$	6	5	4	3	2	1	0

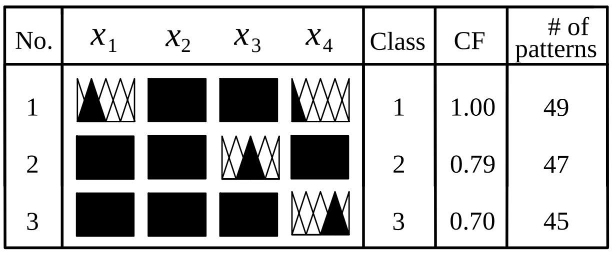

Fig.7 Selected three linguistic rules

4. A HYBRID GENETIC ALGORITHM

In this section, we show that the classification performance of selected linguistic rules can be improved by combining a learning method of fuzzy if-then rules [5] into our two-objective genetic algorithm.

A. Learning of Linguistic Classification Rules

From the classification procedure in Subsection 2.D, we can see that a pattern $\boldsymbol{x}_p = (x_{p1}, ..., x_{pn})$ is classified by a linguistic rule $R_{\hat{j}}$ that satisfies the following relation:

$$\mu_{\hat{j}}(\boldsymbol{x}_p) \cdot CF_{\hat{j}} = \max\{\mu_j(\boldsymbol{x}_p) \cdot CF_j \mid \text{Rule } R_j \in S\}. \quad (20)$$

If the consequent class $C_{\hat{j}}$ of this rule is the same as the actual class of $\boldsymbol{x}_p$, $\boldsymbol{x}_p$ is correctly classified, otherwise $\boldsymbol{x}_p$ is misclassified.

When $\boldsymbol{x}_p$ is correctly classified by the linguistic rule $R_{\hat{j}}$, the grade of certainty $CF_{\hat{j}}$ of this rule is increased as the reward of the correct classification [5]:

$$CF_{\hat{j}}^{new} = CF_{\hat{j}}^{old} + \eta_1 \cdot (1 - CF_{\hat{j}}^{old}), \quad (21)$$

where η_1 is a positive learning constant for increasing the grade of certainty. On the contrary, when $\boldsymbol{x}_p$ is misclassified by the linguistic rule $R_{\hat{j}}$, the grade of certainty $CF_{\hat{j}}$ of this rule is decreased as the punishment of the misclassification [5]:

$$CF_{\hat{j}}^{new} = CF_{\hat{j}}^{old} - \eta_2 \cdot CF_{\hat{j}}^{old}, \quad (22)$$

where η_2 is a positive learning constant for decreasing the grade of certainty.

B. A Hybrid Genetic Algorithm

The learning method of the grade of certainty described in the last subsection is incorporated into our two-objective genetic algorithm. Since the learning method can be applicable to any rule set *S*, we apply it to half of the rule sets (*i.e.*, half of the individuals) generated by the crossover and the mutation in the two-objective genetic algorithm.

C. Simulation Result

We apply the hybrid two-objective genetic algorithm to the classification problem of the iris data. We show the obtained non-dominated solutions in Table 2. From the comparison of Table 2 with Table 1, we can see that the learning method incorporated in the two-objective genetic algorithm improved the classification performance (*i.e.*, $NCP(S)$ in each table) of the selected linguistic rules.

Table 2 Non-dominated solutions obtained by the hybrid two-objective genetic algorithm

$NCP(S)$	150	149	147	145	100	50	0
$\|S\|$	11	6	5	4	2	1	0

5. CONCLUSION

In this paper, we proposed a two-objective genetic algorithm to find non-dominated solutions of the rule selection problem of linguistic classification rules with two objectives: to maximize the number of correctly classified training patterns and to minimize the number of selected rules. We also extended the two-objective genetic algorithm to a hybrid algorithm where a learning method was applied to rule sets generated by genetic operations. The selection of a final rule set from the obtained non-dominated solutions should be done based on the preference of human users.

REFERENCES

[1] C.C.Lee, Fuzzy logic in control systems: fuzzy logic controller - Part I and Part II, *IEEE Trans. on SMC* **20** (1990) 404-435.

[2] H.Ishibuchi, K.Nozaki, N.Yamamoto and H.Tanaka, Selecting fuzzy if-then rules for classification problems using genetic algorithms, *IEEE Trans. on Fuzzy Systems* **3** (1995, in press).

[3] H.Ishibuchi, K.Nozaki, N.Yamamoto and H.Tanaka, Construction of fuzzy classification systems with rectangular fuzzy rules using genetic algorithms, *Fuzzy Sets and Systems* **65** (1994) 237-253.

[4] H.Ishibuchi, T.Murata and I.B.Turksen, A genetic-algorithm-based approach to the selection of linguistic classification rules, *Proc. of the 3rd EUFIT* (August 28-31, 1995, Aachen, Germany, in press).

[5] K.Nozaki, H.Ishibuchi and H.Tanaka, Trainable fuzzy classification systems based on fuzzy if-then rules, *Proc. of 3rd FUZZ-IEEE* (Orlando, USA, June 26-29, 1994) 498-502.

[6] R.A.Fisher, The use of multiple measurements in taxonomic problems, *Annals of Eugenics* **7** (1936) 179-188.