CHAPTER 6

GENETIC ALGORITHMS FOR MULTI-OBJECTIVE LINGUISTIC RULE SELECTION

6.1 INTRODUCTION

We explained genetic algorithms for designing fuzzy classification systems in the previous chapter. We considered the following two objectives in constructing fuzzy classification systems:

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

These two objectives were combined into a single scalar fitness function using constant weights in the previous chapter. An idea of a multi-objective genetic algorithm was proposed to find a set of non-dominated solutions of the rule selection problem with the above two objectives in [35,41]. A fuzzy classifier system [42,43] was proposed to handle a rule selection problem with only the first objective for multi-dimensional pattern classification problems involving many features.

The main aim of this chapter is to introduce several methods for finding a set of non-dominated solutions of the rule selection problem with the above two objectives. We reconsider the rule selection problem described in Section 5.3. First we apply three methods based on a genetic algorithm with a single objective for finding a set of non-dominated solutions of the rule selection problem. We also apply a method based on a multi-objective genetic algorithm (MOGA) [41]. Next we introduce a hybrid algorithm by combining a learning procedure [87,88] of linguistic classification rules with the MOGA. The performance of the several methods for finding a set of non-dominated solutions are examined by applying them to iris data [13]. Then we modify our genetic-algorithm-based multi-objective fuzzy rule selection method for handling high-dimensional pattern classification problems with many continuous attributes.

6.2 SINGLE-OBJECTIVE GENETIC ALGORITHMS FOR MULTI-OBJECTIVE LINGUISTIC RULE SELECTION

In this section, we apply three methods based on a genetic algorithm with a single objective described in Section 5.3. First we employ a constant weight genetic algorithm (CWGA) for linguistic rule selection. We assign various weights in order to find a set of non-dominated solutions. Next we employ genetic algorithms with a single objective in which only one of the two objectives is considered as an objective and the other objective is considered as a constraint condition.

6.2.1 Variable weights

We have already introduced a genetic algorithm with a single objective for linguistic rule selection with two objectives in Section 5.3. In the genetic algorithm, we specified a scalar fitness function using constant weights as follows:

$$f(S) = w_{\mathrm{NCP}} \cdot NCP(S) - w_{\mathrm{S}} \cdot |S|, \tag{6.1}$$

where w_{NCP} and w_{S} are non-negative constant weights assigned to the two objectives $NCP(S)$ and $|S|$, respectively.

In the genetic algorithm described in Section 5.3, the weights w_{NCP} and w_{S} were constant. Thus the search direction of the CWGA was fixed as shown in Fig. 6.1. This means that the choice of the weight values in (6.1) has a significant effect on the final solution (*i.e.*, selected linguistic classification rules) obtained by the CWGA. 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} in advance.

One of the basic approach to multi-objective optimization problems is to find not a single solution but a set of non-dominated solutions. The final solution should be determined by decision makers (*i.e.*, human users in the rule selection problem) from a set of non-dominated solutions depending on their preference. Thus we introduce several methods for searching for a set of non-dominated solutions of the two-objective linguistic rule selection problem.

One simple method for searching for a set of non-dominated solutions is to employ variable weights. That is, the execution of the CWGA is repeated using various values of the weights

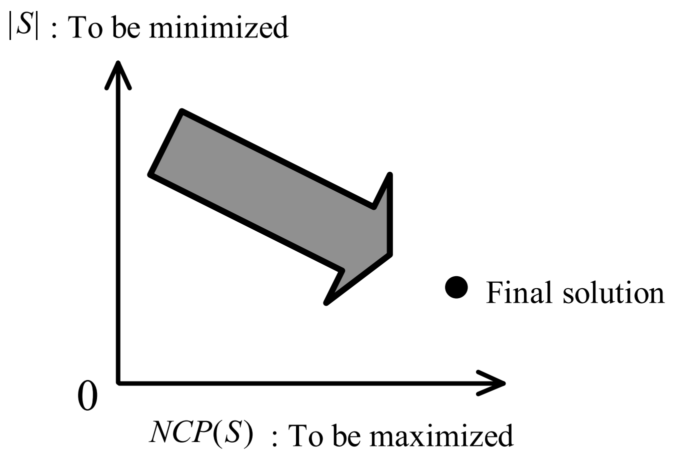


Fig. 6.1 Search direction of the CWGA.

w_{NCP} and w_S. The CWGA in Subsection 5.3.3 was applied to the iris data [13] with the following parameter specification:

Population size: $N_{pop} = 20$,

Crossover probability: 1.00,

Mutation probabilities: $P_m(1 \rightarrow -1) = 0.1$, $P_m(-1 \rightarrow 1) = 0.001$,

Stopping condition: $t_{max} = 1000$ (*i.e.*, 1,000 generations).

The following ten pairs of the weight values were employed:

$$(w_{NCP}, w_S) = (0.1, 1), (0.5, 1), (1, 1), (5, 1), (10, 1), (50, 1), (100, 1), (500, 1), (1000, 1), (5000, 1).$$

The CWGA described in the Section 5.3 was applied to the iris data using each pair of the weight values. From these ten trials, ten solutions in Table 6.1 were obtained. From Table 6.1, we can see that the following solution are non-dominated:

$$\{\ (NCP(S),\ |S|)\ \} = \{\ (142, 3),\ (146, 4),\ (147, 5)\ \}.$$

The final solution should be selected from these three non-dominated solutions by human users depending on their preference.

Table 6.1 Obtained solutions by the CWGA with various weight values.

w_{NCP}	w_{S}	$NCP(S)$	$\lvert S \rvert$
0.1	1	142	3
0.5	1	146	4
1	1	147	5
5	1	147	5
10	1	147	7
50	1	147	5
100	1	147	6
500	1	146	4
1000	1	146	4
5000	1	146	4

6.2.2 Constraint condition on the number of rules

We can also search for a set of non-dominated solutions of the rule selection problem by introducing a constraint condition on the number of rules (*i.e.*, a constraint condition on $|S|$). As we explained in Subsection 5.3.1, we can formulate the rule selection problem with a single objective as follows:

$$\text{Maximize } \; NCP(S), \tag{6.2}$$

$$\text{subject to } \; |S| \le N_{\mathrm{rule}}, \tag{6.3}$$

$$S \subseteq S_{\mathrm{ALL}}, \tag{6.4}$$

where N_{rule} is a constant of the constraint condition on the number of rules. We formulate the following fitness function by introducing large penalty when the constraint condition (6.3) is not satisfied:

$$f(S) = w_{\mathrm{NCP}} \cdot NCP(S) - w_{\mathrm{S}} \cdot \max\{0,\, |S| - N_{\mathrm{rule}}\}, \tag{6.5}$$

where the weights w_{NCP} and w_{S} are specified as $w_{\mathrm{NCP}} << w_{\mathrm{S}}$ in order to attach large penalty to the fitness function when the constraint condition (6.3) is not satisfied. Using different values in the right-hand side of the constraint condition (6.3), we can search for a set of non-dominated solution of the two-objective rule selection problem in (5.10)-(5.11).

Table 6.2 Obtained solutions by the genetic algorithm with a constraint condition on the number of selected linguistic classification rules.

Constraint	$NCP(S)$	$\|S\|$
$\|S\| \leq 3$	142	3
$\|S\| \leq 4$	146	4
$\|S\| \leq 5$	147	5
$\|S\| \leq 6$	147	5
$\|S\| \leq 7$	147	5
$\|S\| \leq 8$	147	5
$\|S\| \leq 9$	147	5
$\|S\| \leq 10$	147	5
$\|S\| \leq 11$	147	5
$\|S\| \leq 12$	147	5

A genetic algorithm which is basically the same as shown in Subsection 5.3 except for the definition of the fitness function was applied to the iris data using each of the following ten values of N_{rule}:

$$N_{\text{rule}} = 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9,\ 10,\ 11,\ 12\,.$$

We assign a value to N_{rule} from three because a classification system should have at least three rules in order to classify all the patterns from three classes in the iris data. By the ten trials of the genetic algorithm with $w_{\text{NCP}} = 1$ and $w_{\text{S}} = 100$, ten solutions were obtained. If the fitness value of the obtained solution is the same or less than that of the solution obtained by a genetic algorithm with more strict constraint condition (*i.e.*, less number of rules), then the solution with better fitness value is regarded as the solution. Table 6.2 shows the obtained solutions. We can see that the following solutions are non-dominated in Table 6.2.

$$\{\ (NCP(S),\ |S|)\ \} = \{\ (142, 3),\ (146, 4),\ (147, 5)\ \}\,.$$

6.2.3 Constraint condition on the number of correctly classified patterns

In the last subsection, we introduced a constraint condition on the number of selected linguistic classification rules. In a similar manner, we can introduce a constraint condition on the number of correctly classified training patterns. As we explained in Subsection 5.3.1, our rule selection problem with a single objective can be written as

$$\text{Minimize} \quad |S|, \tag{6.6}$$

$$\text{subject to} \quad NCP(S) \geq N_{\text{pattern}}, \tag{6.7}$$

$$S \subseteq S_{\text{ALL}}, \tag{6.8}$$

where N_{pattern} is a constant of the constraint condition on the number of correctly classified training patterns. We formulate the following fitness function by introducing large penalty when the constraint condition (6.7) is not satisfied:

$$f(S) = -\, w_{\text{NCP}} \cdot \max\{0,\, N_{\text{pattern}} - NCP(S)\} - w_{\text{S}} \cdot |S|, \tag{6.9}$$

where the weights w_{NCP} and w_{S} are specified as $w_{\text{NCP}} >> w_{\text{S}}$ in order to attach large penalty to the fitness function when the constraint condition (6.7) is not satisfied. A set of non-dominated solutions of the rule selection problem in (5.10)-(5.11) can be obtained using this fitness function with various values of N_{pattern}.

A genetic algorithm which is basically the same as in Subsection 5.3 was applied to the iris data using each of the following ten values of N_{pattern}:

$$N_{\text{pattern}} = 141,\ 142,\ 143,\ 144,\ 145,\ 146,\ 147,\ 148,\ 149,\ 150.$$

By the ten trials of the genetic algorithm with $w_{\text{NCP}} = 100$ and $w_{\text{S}} = 1$, ten solutions were obtained. As in the similar manner in the previous subsection, the better solution obtained by a genetic algorithm with more strict constraint condition (*i.e.*, more number of patterns) is selected as a solution. Table 6.3 shows obtained solutions. We can see that the following solutions are non-dominated in Table 6.3.

$$\{\, (NCP(S),\ |S|) \,\} = \{\, (142, 3),\ (146, 4),\ (147, 6) \,\}.$$

Table 6.3 Obtained solutions by the single-objective genetic algorithm with a constraint condition on the number of correctly classified training patterns.

Constraint	$NCP(S)$	$\|S\|$
$NCP(S) \geq 141$	142	3
$NCP(S) \geq 142$	142	3
$NCP(S) \geq 143$	146	4
$NCP(S) \geq 144$	146	4
$NCP(S) \geq 145$	146	4
$NCP(S) \geq 146$	146	4
$NCP(S) \geq 147$	147	6
$NCP(S) \geq 148$	147	6
$NCP(S) \geq 149$	147	6
$NCP(S) \geq 150$	147	6

6.3 MULTI-OBJECTIVE GENETIC ALGORITHM FOR MULTI-OBJECTIVE RULE SELECTION

In the previous section, we described three methods for searching for a set of non-dominated solutions of the two-objective linguistic rule selection problem by the genetic algorithms with a single objective. The genetic algorithms with a single objective were repeated with different parameter specifications. In this section, we introduce a multi-objective genetic algorithm (MOGA) for searching for a set of non-dominated solutions more directly. Basically, a multi-objective genetic algorithm is the same algorithm as the MOGA described in Chapter 4. In the MOGA in Chapter 4, permutation strings were used because the flowshop scheduling problems were considered. In this chapter, a string which consists of “1”, “0”, and “-1” is treated as an individual. Therefore we should employ the genetic operators for the binary coding in order to construct the MOGA for the two-objective linguistic rule selection problem in this chapter.

A rule set S is treated as a string $S = s_1 s_2 \dots s_r$ in the MOGA as in the genetic algorithm described in the previous section (for details, see Subsection 5.3.2). Crossover and mutation operators in the MOGA are also the same as those in the genetic algorithm in the previous section. Our MOGA differs from the genetic algorithm in the previous section in its selection procedure and elitist strategy. In our MOGA, the selection probability $P_{\mathrm{S}}(S)$ is determined as follows:

$$P_{\mathrm{S}}(S) = \frac{f(S) - f_{\min}(\Psi_t)}{\sum_{S' \in \Psi_t} \{f(S') - f_{\min}(\Psi_t)\}}, \tag{6.10}$$

where

$$f_{\min}(\Psi_t) = \min\{f(S') \mid S' \in \Psi_t\}. \tag{6.11}$$

In (6.10)-(6.11), the fitness function $f(S)$ of each rule set S is specified as follows:

$$f(S) = w_{\mathrm{NCP}} \cdot NCP(S) - w_{\mathrm{S}} \cdot |S|, \tag{6.12}$$

where w_{NCP} and w_{S} are non-negative weights assigned to the two objectives $NCP(S)$ and $|S|$, respectively. These two weights are randomly specified weights. That is, when a pair of parent

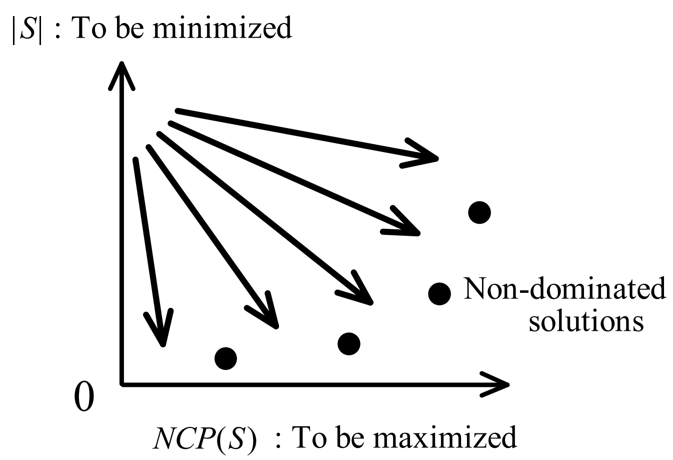


Fig. 6.2 Search direction of the multi-objective genetic algorithm.

strings are selected, the values of the weights w_{NCP} and w_{S} are assigned as

$$w_{\mathrm{NCP}}: \text{a random real number in } [0, 1], \tag{6.13}$$

$$w_{\mathrm{S}}: \ w_{\mathrm{S}} = 1 - w_{\mathrm{NCP}}. \tag{6.14}$$

Random weight values are given by (6.13)-(6.14) for each selection of a pair of parent strings. Thus we can see that the selection procedure in each generation of our MOGA drives the search of the algorithm in various directions in Fig. 6.2.

In the execution of the MOGA, a tentative set of non-dominated solutions is externally preserved in the same manner as in Chapter 4. This means that there are two sets of strings in each generation: one is a current population and the other is a tentative set of non-dominated solutions. A certain number of strings (say, N_{elite} strings) are randomly selected from the tentative set of non-dominated solutions, and the selected strings are added to the current population as elite solutions in our MOGA.

We construct the following multi-objective genetic algorithm where t is the number of generations and $t_{\max}$ is the maximum number of generations that is prespecified to terminate the algorithm:

Step 0 (Initialization): Let $t := 0$. Generate an initial population containing N_{pop} strings in the same manner as in the genetic algorithm in the previous section.

Step 1 (Rule elimination): Classify all the given training patterns by linguistic classification rules included in each string S. Exclude non-active rules from S. This rule elimination procedure is applied to all the strings in the current population.

Step 2 (Evaluation): Calculate the values of the two objectives $NCP(S)$ and $|S|$ for the strings in the current population. Update the tentative set of non-dominated solutions.

Step 3 (Selection): Let Ψ_t be the population in the t-th generation. Calculate the fitness value of each string using random weights in (6.13)-(6.14). Select a pair of strings from the current population according to the selection probability $P_s(S)$ in (6.10). This procedure is repeated for selecting $N_{pop} / 2$ pairs of parent strings from the current population Ψ_t.

Step 4 (Crossover): To each of the selected pairs, apply the uniform crossover operator in order to generate two strings in the same manner as in the genetic algorithm in the previous section.

Step 5 (Mutation): To each bit value of the generated strings by the crossover operator, apply the mutation operator.

Step 6 (Elitist strategy): Randomly remove N_{elite} strings from the generated N_{pop} strings, and add N_{elite} strings that are randomly selected from the tentative set of non-dominated solutions to the current population.

Step 7 (Termination test): Let $t := t + 1$. If $t = t_{max}$, stop the algorithm. Otherwise, return to Step 1.

We applied the MOGA to the iris data. In order to compare the MOGA with the genetic algorithms in the previous section under the same computation load, the execution of the multi-objective algorithm was repeated ten times. That is, the same number of solutions were examined in order to form a set of non-dominated solutions by the MOGA. The number of elite solutions N_{elite} was specified as $N_{elite} = 3$. By the ten trials of the MOGA the following non-dominated solutions were obtained:

$$\{ (NCP(S), |S|) \} = \{ (0,0), (50,1), (100,2), (142,3), (146,4), (147,5), (148,6) \}. \qquad (6.15)$$

Here we summarize the non-dominated solutions obtained by each method in the last section (see Table 6.1 ～ Table 6.3):

(1) By the method based on variable weights in Subsection 6.2.1:

$$\{ (NCP(S), |S|) \} = \{ (142,3), (146,4), (147,5) \}. \tag{6.16}$$

(2) By the method based on the constraint condition $|S| \le N_{\mathrm{rule}}$ in Subsection 6.2.2:

$$\{ (NCP(S), |S|) \} = \{ (142,3), (146,4), (147,5) \}. \tag{6.17}$$

(3) By the method based on the constraint condition $NCP(S) \ge N_{\mathrm{pattern}}$ in Subsection 6.2.3:

$$\{ (NCP(S), |S|) \} = \{ (142,3), (146,4), (147,6) \}. \tag{6.18}$$

From the comparison of the result in (6.15) by the MOGA with these results in (6.16)-(6.18) by the genetic algorithms with a single objective, we can see that a bit better set of non-dominated solutions was obtained by the MOGA. For example, a rule set that can correctly classified 148 patterns was not found by any method based on the genetic algorithms with a single objective in the previous section (see (6.16)-(6.18)).

6.4 MULTI-OBJECTIVE GENETIC ALGORITHM WITH LEARNING PROCEDURE

The learning procedure of the grade of certainty CF_j [87,88] is combined with our MOGA in the same manner as in Subsections 5.3.3 and 5.4.4. Since the learning procedure is applicable to any rule set in *S*, we apply it to all the rule sets (*i.e.*, all the strings) generated by the crossover and mutation operators in the genetic algorithm. That is, the following procedure is inserted between Step 6 and Step 7 of the MOGA described in Subsection 6.3:

[Learning procedure of the grade of certainty]

Step 6.5 (Learning): Apply the learning procedure to each rule set *S* generated by the crossover and mutation operators. The learning procedure for each rule set *S* is iterated N_{learning} times for all the training patterns.

The hybrid algorithm was applied to the iris data using the same parameter specifications as in the MOGA in the previous section. The learning rates η_1 in (5.31) and η_2 in (5.32) were specified as $\eta_1 = 0.001$ and $\eta_2 = 0.1$. We examined four specifications of N_{learning}, *i.e.*, $N_{\text{learning}} = 0$, 1, 2, 10. Table 6.4 shows non-dominated solutions by ten trials of the hybrid algorithm with each specification of N_{learning}. For example, we can see from Table 6.4 that the following non-dominated solutions were obtained by specifying N_{learning} as $N_{\text{learning}} = 10$:

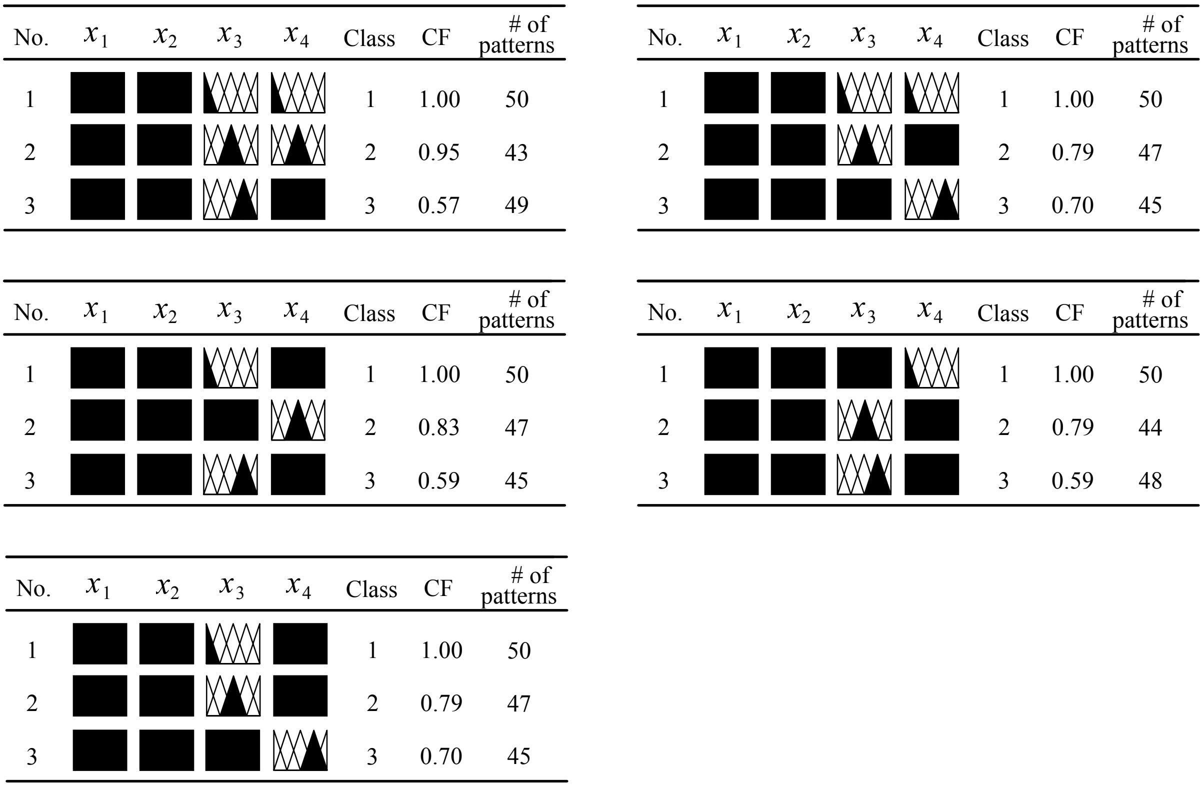
$$\{\, (NCP(S),\ |S|) \,\} = \{\, (0, 0), (50, 1), (100, 2), (145, 3), (147, 4), (148, 5) \,\}. \tag{6.19}$$

From Table 6.4, we can see that the classification performance of the selected linguistic classification rules was improved by combining the learning procedure into the multi-objective genetic algorithm. For example, three linguistic classification rules selected by the multi-objective algorithm with no learning (*i.e.*, $N_{\text{learning}} = 0$) correctly classified 142 patterns while 145 patterns were correctly classified by three rules selected by the hybrid algorithm with $N_{\text{learning}} = 2$ and $N_{\text{learning}} = 10$. In Fig. 6.3, we show rule sets with three linguistic classification rules obtained by the non-hybrid algorithm. The five rule sets in Fig. 6.3, which

Table 6.4 Obtained solutions by the hybrid algorithm with various specifications of the number of iterations of the learning method (*i.e.*, N_{learning}). The non-hybrid multi-objective genetic algorithm corresponds to the case of $N_{\text{learning}} = 0$. "*" denotes that a non-dominated solution with the corresponding number of selected rules was not obtained.

The number of selected rules: $\lvert S \rvert$	The number of correctly classified training patterns: $NCP(S)$			
	$N_{\text{learning}} = 0$	$N_{\text{learning}} = 1$	$N_{\text{learning}} = 2$	$N_{\text{learning}} = 10$
0	0	0	0	0
1	50	50	50	50
2	100	100	100	100
3	142	143	145	145
4	146	147	147	147
5	147	*	148	148
6	148	148	149	*
7	*	149	*	*

had the same classification performance (*i.e.*, which can correctly classify 142 patterns), were obtained by all the ten trials of the non-hybrid algorithm. On the other hand, three rule sets with three linguistic classification rules that could correctly classify 145 patterns were obtained by all the ten trials of the hybrid algorithm with $N_{\text{learning}} = 10$. The three rule sets are shown in Fig. 6.4. The first rule set in each figure consists of the same three linguistic classification rules except for their grades of certainty (*i.e.*, *CF* in each figure).



No.	x_1	x_2	x_3	x_4	Class	CF	# of patterns
1					1	1.00	50
2					2	0.95	43
3					3	0.57	49

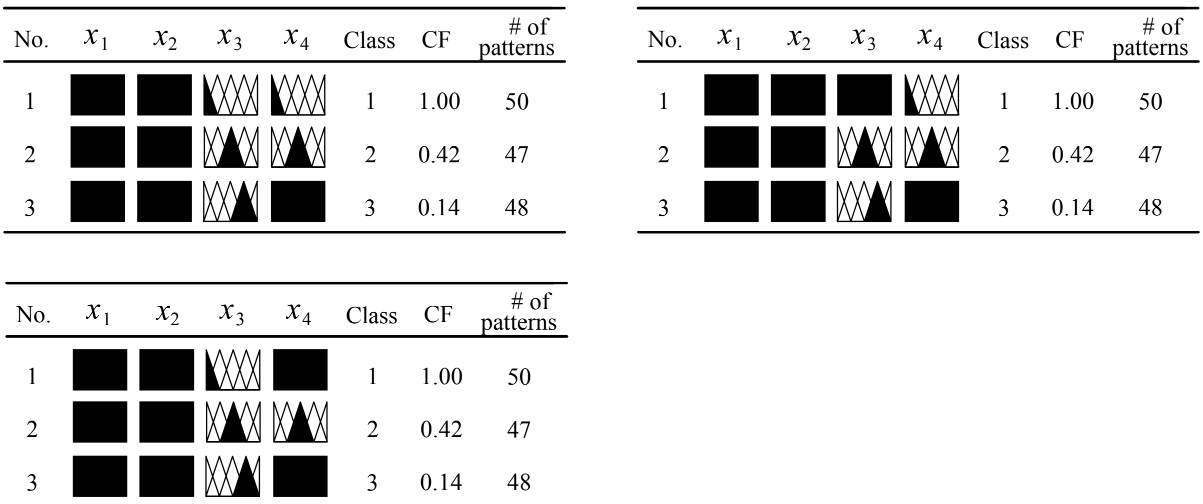
No.	x_1	x_2	x_3	x_4	Class	CF	# of patterns
1					1	1.00	50
2					2	0.79	47
3					3	0.70	45

No.	x_1	x_2	x_3	x_4	Class	CF	# of patterns
1					1	1.00	50
2					2	0.83	47
3					3	0.59	45

No.	x_1	x_2	x_3	x_4	Class	CF	# of patterns
1					1	1.00	50
2					2	0.79	44
3					3	0.59	48

No.	x_1	x_2	x_3	x_4	Class	CF	# of patterns
1					1	1.00	50
2					2	0.79	47
3					3	0.70	45

Fig. 6.3 Rule sets obtained by the non-hybrid algorithm.



No.	x_1	x_2	x_3	x_4	Class	CF	# of patterns
1					1	1.00	50
2					2	0.42	47
3					3	0.14	48

No.	x_1	x_2	x_3	x_4	Class	CF	# of patterns
1					1	1.00	50
2					2	0.42	47
3					3	0.14	48

No.	x_1	x_2	x_3	x_4	Class	CF	# of patterns
1					1	1.00	50
2					2	0.42	47
3					3	0.14	48

Fig. 6.4 Rule sets obtained by the hybrid algorithm with $N_{\text{learning}} = 10$.

6.5 MULTI-OBJECTIVE GENETIC ALGORITHM FOR HIGH-DIMIENSIONAL CLASSIFICATION PROBLEMS

It has often been claimed that the grid-type fuzzy partition can not handle high-dimensional problems with many input variables due to the curse of dimensionality (see, for example, Carse *et al.*[5]). That is, when we use the grid-type fuzzy partition, the number of fuzzy rules exponentially increases as the number of input variables increases.

In Chapter 5 and Chapter 6, we employed one of six linguistic values in Fig. 5.2 as the antecedent fuzzy set A_{ji} in linguistic classification rules. The antecedent part with "*don't care*" is removable from the linguistic classification rule. For example, the following two linguistic classification rules are the same:

(i) If x_1 is *don't care* and x_2 is *small* then Class C_j with $CF = CF_j$,

(ii) If x_2 is *small* then Class C_j with $CF = CF_j$.

These linguistic classification rules correspond to the fuzzy partitions in Fig. 5.3.

Because we have the six linguistic values for each axis of the *n*-dimensional pattern space (see, Fig. 5.2), the total number of linguistic classification rules is 6^n. The relation between the number of attributes (*i.e.*, *n*) and the number of linguistic classification rules (*i.e.*, 6^n) are shown in Table 6.5. From this table, we can see that the number of linguistic classification rules is too large to be included in a single fuzzy rule-based classification system, which was described in Chapter 5 and the previous section, when the number of attributes is large (*e.g.*, $n \geq 4$). The genetic-algorithm-based rule selection methods can not handle such a large number of linguistic classification rules.

Table 6.5 The number of linguistic classification rules.

Number of attributes	Number of linguistic classification rules
2	36
4	1,296
6	46,6556
8	1,679,616
10	60,466,176

As shown in Table 6.5, the total number of linguistic classification rules is too large to be handled by the genetic-algorithm-based rule selection methods. Thus we generate only a part of the 6^n linguistic classification rules as candidate rules. Let us define the length of a linguistic classification rule by the number of its antecedent fuzzy sets except for "*don't care*." For example, the length of the following linguistic classification rule is three.

$$\text{If } x_1 \text{ is } \textit{small} \;\&\; x_2 \text{ is } \textit{don't care} \;\&\; x_3 \text{ is } \textit{don't care}$$
$$\&\; x_4 \text{ is } \textit{large} \;\&\; x_5 \text{ is } \textit{small} \;\&\; x_6 \text{ is } \textit{don't care}$$
$$\text{then Class } C_j \text{ with } CF = CF_j.$$

This linguistic classification rule can be rewritten as follows by removing the attributes with "*don't care*":

$$\text{If } x_1 \text{ is } \textit{small} \;\&\; x_4 \text{ is } \textit{large} \;\&\; x_5 \text{ is } \textit{small}$$
$$\text{then Class } C_j \text{ with } CF = CF_j.$$

Thus we can see that the length of the linguistic classification rule is the same as the number of conditions in its antecedent part. In this section, we reduce the number of candidate linguistic classification rules in the genetic-algorithm-based multi-objective rule selection method which was described in the previous section by the constraint on their length. Let us consider a 10-dimensional pattern classification problem. As shown in Table 6.5, the total number of linguistic classification rules in this problem is $6^{10} \cong 6.0 \times 10^7$. In Table 6.6, we show the number of linguistic classification rules according to their length in the case of a 10-dimensional pattern classification problem. From this table, we can see that the number of candidate linguistic classification rules is not large if we only generate linguistic classification rules whose length is less than or equal to two.

Table 6.6 The number of linguistic classification rules.

Length of linguistic classification rules	Number of linguistic classification rules
0	1
1	50
2	1,125
3	15,000
4	131,250
5	787,500
6	3,281,250
7	9,375,000
8	17,578,125
9	19,531,250
10	9,76,5625
Total number of rules	60,466,176

One alternative approach to such a high-dimensional pattern classification problem is a fuzzy classifier system [42,43] where each fuzzy if-then rule was coded as a string. Because each population consisted of a relatively small number of fuzzy if-then rules (*e.g.*, 60 rules were used in [42,43]), the fuzzy classifier system can be applied to high-dimensional pattern classification problems. The effectiveness of the fuzzy classifier system was also demonstrated in Yuan & Zhuang [120] for high-dimensional pattern classification problems with many discrete attributes. While the fuzzy classifier system can find fuzzy if-then rules with high classification performance for high-dimensional pattern classification problems, the number of fuzzy if-then rules can not be efficiently decreased because a fitness value is not assigned to a rule set but to each fuzzy if-then rule.

In this section, we use the grid-type fuzzy partition for pattern classification problems with many continuous attributes because such a fuzzy partition has an inherent advantage of fuzzy rule-based systems: the comprehensibility of fuzzy if-then rules. Horte [31] pointed out that a simple classification rule which utilized only one attribute out of multiple attributes performed well on a lot of data sets. Therefore we can restrict the number of antecedent fuzzy sets in each linguistic classification rule. Our approach in this section tackles the curse of dimensionality by (i) utilizing "*don't care*" as an antecedent fuzzy set, (ii) efficiently generating a tractable number of candidate linguistic classification rules, (iii) selecting only a small number of linguistic classification rules from the candidate rules, and (iv) constructing a compact fuzzy rule-based classification system by the selected linguistic classification rules. As shown in Table 6.6, the effect of "*don't care*" on the reduction of the number of linguistic classification rules is much more significant in the case of high-dimensional pattern classification problems with many continuous attributes.

6.5.1 Restriction on candidate linguistic classification rules

In this section, we apply the genetic-algorithm-based rule selection method, which was described in Chapter 5 and the previous section, to wine data in [16] in order to examine its ability to find compact rule sets with high classification performance for high-dimensional pattern classification problems with many continuous attributes. The wine data consist of 178 samples with 13 continuous attributes from three classes. We used the wine data because (i) this data set is available from UC Irvine Database [16], (ii) this data set has many continuous attributes, and (iii) our rule selection method can be compared with other genetics-based machine learning methods.

As we have already mentioned, compact rule sets consists of general linguistic classification rules with "*don't care*" attributes. Therefore it seems to be a promising strategy to generate only general linguistic classification rules with many "*don't care*" attributes as candidate rules. For preventing the exponential increase of the number of candidate linguistic classification rules, we only generate linguistic classification rules with short length as candidate rules. We generated linguistic classification rules by restricting the length of candidate rules less than or equal to two. In this case, 2016 linguistic classification rules were generated where 182 dummy rules were included. That is, 1834 linguistic classification rules were generated with their consequent classes and grades of certainty. Thus each of the generated candidate rules has two conditions in its antecedent part at most. Using those candidate rules, we applied the MOGA to the wine classification problem with 13 continuous attributes. We employed the following parameter specifications:

Population size: $N_{\text{pop}} = 50$,

Crossover probability: 1.00,

Mutation probabilities: $P_{\text{m}}(1 \to -1) = 0.1$, $P_{\text{m}}(-1 \to 1) = 0.001$,

Number of elite solutions: $N_{\text{elite}} = 3$,

Stopping condition: $t_{\max} = 10000$ (*i.e.*, 10,000 generations).

Corcoran & Sen [6] reported the following results of their genetics-based machine learning system with 60 non-fuzzy rules in each rule set, 1500 rule sets in each population, and 300 generations (*i.e.*, $1500 \times 300 =$ 450,000 rule sets of 60 non-fuzzy if-then rules were examined in each trial):

Best classification rate: 100%,

Average classification rate: 99.5%,

Worst classification rate: 98.3%.

These results were classification rates obtained by ten independent trials where all the 178 samples were used as training data.

We summarize the non-dominated solutions (*i.e.*, non-dominated rule sets) obtained by the MOGA in Table 6.7. These non-dominated rule sets were obtained after examining 500,000

Table 6.7 Non-dominated solutions obtained by the MOGA for the wine classification problem.

NCP(*S*)	|*S*|
0	0
62	1
118	2
165	3
171	4
175	5
177	6
178	7

rule sets (*i.e.*, 10,000 populations with 50 rule sets) by the MOGA. From the comparison between Table 6.7 and the above-mentioned results by Corcoran & Sen [6], we can see that more compact rule sets were obtained by the MOGA. As shown in Table 6.7, eight non-dominated rule sets were obtained by the MOGA. For example, one non-dominated rule set consists of the following seven linguistic classification rules that can correctly classify 178 patterns (*i.e.*, all the given patterns):

If x_1 is *medium large* and x_4 is *medium small* then Class 1 with $CF = 0.89$,
If x_7 is *medium* and x_{11} is *medium* then Class 1 with $CF = 0.56$,
If x_{10} is *small* then Class 2 with $CF = 0.94$,
If x_{10} is *medium small* and x_{13} is *small* then Class 2 with $CF = 0.81$,
If x_{11} is *medium* and x_{13} is *medium small* then Class 2 with $CF = 0.66$,
If x_4 is *medium* and x_{11} is *small* then Class 3 with $CF = 0.93$,
If x_7 is *small* and x_{11} is *medium small* then Class 3 with $CF = 0.92$.

This rule set can be easily understood by human users because (i) the number of linguistic classification rules is very small, (ii) each linguistic classification rule has only a few conditions (*i.e.*, only a few attributes) in its antecedent part, and (iii) each condition is represented by a linguistic value.

6.5.2 Generation of candidate fuzzy if-then rules

A. Candidate fuzzy if-then rules generated by fuzzy classifier systems

Fuzzy if-then rules generated by the fuzzy classifier system in [42,43] can be used as candidate rules for the rule selection. While the generation procedure of candidate rules in the previous subsection can not generate a long linguistic classification rule with many conditions in the antecedent part, the fuzzy classifier system can generate long rules as well as short rules from training patterns. Because the fuzzy classifier system finds a relatively small number of fuzzy if-then rules with high classification performance for high-dimensional pattern classification problems, the MOGA can be applied to the fuzzy if-then rules generated by the fuzzy classifier system. That is, the combination of the fuzzy classifier system and the MOGA makes a hybrid algorithm that can maximize the classification performance and minimize the number of fuzzy if-then rules for high-dimensional pattern classification problems.

B. Candidate fuzzy if-then rules from Neural Networks

Fuzzy if-then rules extracted from neural networks [96] can be also used as candidate rules for the rule selection. Ishibuchi & Nii [44] proposed an extraction method of linguistic classification rules from trained neural networks. Because fuzzy if-then rules with linguistic values can be extracted from standard feedforward neural networks, the rule extraction method is used for the linguistic analysis of the trained neural networks. Such linguistic analysis becomes much easier if a small number of linguistic classification rules are selected by our MOGA.

6.6 SUMMARY

In this chapter, we considered genetic-algorithm-based method to find a set of non-dominated solutions of the two-objective rule selection problem described in Section 5.3. First we applied three methods based on a genetic algorithm with a single objective for finding the non-dominated solutions of the rule selection problem. We also applied a method based on a multi-objective genetic algorithm [81]. Next we introduced a hybrid algorithm by combining a learning method [87,88] of linguistic classification rules with the multi-objective genetic algorithm. The performance of the several methods for finding a set of non-dominated solutions were examined by applying them to the iris data. Then we modified our genetic-algorithm-based multi-objective fuzzy rule selection method for handling high-dimensional pattern classification problems with many continuous attributes. We applied the modified method to the wine classification problem with 13 continuous attributes.

The advantages of our genetic-algorithm-based method to the design of fuzzy rule-based classification systems can be summarized as follows:

(i) Human users can choose the final rule set from several alternative rule sets by considering the tradeoff between the performance and the compactness of the fuzzy classification system.

(ii) These two criteria are simultaneously handled in the MOGA.

(iii) A small number of linguistic classification rules with high classification performance can be selected for multi-dimensional pattern classification problems with many continuous attributes. For example, our MOGA selected eight linguistic classification rules that can correctly classify all the 178 patterns in the wine classification problem with 13 attributes.

(iv) Selected linguistic classification rules can be linguistically interpreted by human users. This means that linguistic knowledge is extracted from numerical data by our genetic-algorithm-based method.

(v) Our MOGA is a general algorithm that can be applied to rule selection problems in various areas. For example, it can be applied to the rule selection of non-fuzzy classification rules. It can be also extended from pattern classification problems to others areas such as function approximation problems.