Design of Accurate Classifiers With a Compact Fuzzy-Rule Base Using an Evolutionary Scatter Partition of Feature Space

Shinn-Ying Ho, *Member, IEEE*, Hung-Ming Chen, *Student Member, IEEE*, Shinn-Jang Ho, *Member, IEEE*, and Tai-Kang Chen

***Abstract*—An evolutionary approach to designing accurate classifiers with a compact fuzzy-rule base using a scatter partition of feature space is proposed, in which all the elements of the fuzzy classifier design problem have been moved in parameters of a complex optimization problem. An intelligent genetic algorithm (IGA) is used to effectively solve the design problem of fuzzy classifiers with many tuning parameters. The merits of the proposed method are threefold: 1) the proposed method has high search ability to efficiently find fuzzy rule-based systems with high fitness values, 2) obtained fuzzy rules have high interpretability, and 3) obtained compact classifiers have high classification accuracy on unseen test patterns. The sensitivity of control parameters of the proposed method is empirically analyzed to show the robustness of the IGA-based method. The performance comparison and statistical analysis of experimental results using ten-fold cross validation show that the IGA-based method without heuristics is efficient in designing accurate and compact fuzzy classifiers using 11 well-known data sets with numerical attribute values.**

***Index Terms*—Fuzzy classifier, intelligent genetic algorithm, orthogonal experimental design, scatter partition.**

I. Introduction

DESIGNING optimal fuzzy classifiers is equivalent to finding an optimal solution in a high-dimensional search space where each point represents a rule set, membership functions, and the behavior of the corresponding system. Genetic algorithms (GAs) have been proven effective in searching extremely complex spaces, and are particularly suitable for solving multimodal optimization problems [1]. This study investigates how to efficiently partition high-dimensional feature space using GA to produce an accurate classifier with a compact fuzzy-rule base. The following three fundamental issues are simultaneously considered to efficiently achieve this goal.

Manuscript received June 12, 2002; revised August 3, 2003. This work was supported by the National Science Council of R.O.C. under Contract NSC 90-2213-E-035-013. The paper was recommended by Associate Editor H. Ishibuchi.

S.-Y. Ho, H.-M. Chen, and T.-K. Chen are with the Department of Information Engineering and Computer Science, Feng Chia University, Taichung, Taiwan 407, R.O.C. (e-mail address: syho@fcu.edu.tw).

S.-J. Ho is with the Department of Automation Engineering, National Huwei Institute of Technology, Huwei, Yunlin, Taiwan 632, R.O.C. (e-mail address: sjho@nhit.edu.tw).

Digital Object Identifier 10.1109/TSMCB.2003.819160

A. Fuzzy Partition

There are three fuzzy partition approaches: grid partition, tree partition, and scatter partition, and they are briefly described as follows.

1) Grid partition is the most commonly used fuzzy partition approach [2]–[6]. There may be p^n fuzzy rules in the case of p fuzzy sets on each axis of an n-D feature space using grid partition. A major advantage of grid partition is that fuzzy rules obtained from fixed linguistic fuzzy grids are always linguistically interpretable. Efficient high-dimensional GA-based fuzzy classifiers with comprehensible fuzzy-rule bases using linguistic grid partitions can be found in [3]–[6].
2) Tree partition results from a series of guillotine cuts. A guillotine cut is made entirely across the subspace to be partitioned, and each of the regions thus produced can then be subjected to independent guillotine cutting. Tree partition can significantly relieve the problem of rule explosion and accelerate classification, but its application to high-dimensional problems faces practical problems [7]. Janikow proposed a GA-based method for optimizing the fuzzy components of fuzzy trees, in which the optimization is incorporated with the fuzzy tree-building routine [8], [9].
3) Scatter partition uses multi-dimensional antecedent fuzzy sets. From the viewpoint of classification performance, scatter partition may be the most effective approach to designing high-dimensional fuzzy classifiers [3]. Scatter partition usually generates fewer fuzzy regions than the grid and tree partitions owing to the natural clustering property of training patterns. However, scatter partition of high-dimensional feature spaces is difficult, and thus some learning or automatic evolutionary procedures become necessary [7]. The scatter partition approaches can be further divided into three fuzzy partition methods based on the type of fuzzy regions: hyperbox partition [10], ellipsoid partition [11] and polyhedron partition [12].

B. Compact Fuzzy-Rule Base

Compact fuzzy-rule base is an important objective for designing efficient fuzzy classifiers. Some approaches that attempt to achieve this objective are described below.

1) Feature selection. Because not all features are necessary for high-dimensional classification task, a genetic feature selection process is used to determine a set of feature subsets [6]. Thawonmas and Abe [13] proposed an irrelevant feature elimination algorithm based on the analysis of class regions generated by a fuzzy classifier.
2) Rule selection. Ishibuchi *et al.* [3], [4] proposed a GA-based method to minimize the number of linguistic fuzzy rules for high-dimensional fuzzy classifiers.
3) Selecting the best one rule at a time iteratively. The best rule on a training set and a fixed class is the one that is consistent and affects the highest number of examples [5].
4) Partitioning of feature space. Mandal [14] proposed a partitioning method to decompose a feature space into overlapping hyperboxes, depending on the relative positions of the pattern classes found in the training patterns.
5) Fuzzy clustering with model reduction. Roubos and Setnes [2] proposed an approach that fuzzy clustering is first used to obtain an initial rule-based model. Similarity-based simplification and multi-objective GA-based optimization are then used to decrease the complexity of the model while maintaining high accuracy.

C. High Classification Accuracy

Some approaches that can improve classification accuracy are described below.

1) Membership functions must be flexible enough to develop an accurate fuzzy classifier [15]. However, flexible membership functions need additional tuning parameters to adjust the shapes of these membership functions. Inflexible membership function may lead to more fuzzy rules for obtaining an accurate classifier.
2) Homaifar and McCormick [16] showed that simultaneous design of membership functions and fuzzy rules can enhance the performance of fuzzy systems. However, the simultaneous design using GA is generally applied to fuzzy controllers with few input variables [16], [17]. For high-dimensional patterns, there are few evolutionary fuzzy classifier designs using the simultaneous design of flexible membership functions and fuzzy rules [18].
3) It has been confirmed that the performance of fuzzy rules can be improved by adjusting the certainty grade of each rule [19]. To alleviate the load of GA, an efficient heuristic rule generation procedure for determining the consequent class and the certainty grade of the fuzzy rule is used in [3], [4].

According to the above-mentioned analysis, if flexible membership functions and fuzzy rules with both certainty grade and consequent class are determined simultaneously to obtain an accurate and compact fuzzy-rule base, the evolutionary design of high-dimensional fuzzy classifiers can be regarded as an optimization problem with lots of system's tuning parameters. The performance of GA would be greatly degraded when applied to a large parameter optimization problem (LPOP) that is shown by theoretical analysis in [20]. As a result, the success of the approach to formulating the fuzzy classifier design to an LPOP mainly relies on a powerful optimization algorithm to solve the LPOP.

In this paper, an evolutionary approach to designing accurate classifiers with a compact fuzzy-rule base is proposed, in which all the elements of the fuzzy classifier design problem have been moved in parameters of a complex optimization problem. An intelligent genetic algorithm IGA based on orthogonal experimental design (OED) [21] is used to effectively solve the design problem of high-dimensional fuzzy classifiers with many tuning parameters. The OED-based evolutionary algorithms can effectively solve the applications of LPOP [22]–[24].

The merits of the proposed method are threefold: 1) the proposed method has high search ability to efficiently find fuzzy rule-based systems with high fitness values, 2) obtained fuzzy rules have high interpretability, and 3) obtained compact classifiers have high classification accuracy on unseen test patterns. The sensitivity of control parameters of the proposed method is empirically analyzed to show the robustness of the IGA-based method. The performance comparison and statistical analysis of experimental results using ten-fold cross validation show that the IGA-based method without heuristics is efficient in designing accurate and compact fuzzy classifiers using 11 well-known data sets with numerical attribute values.

The next section introduces the proposed evolutionary fuzzy classifier design. An efficient algorithm IGA for solving the design problem of accurate classifiers with a compact fuzzy-rule base is described in Section III. In Section IV, the proposed method is demonstrated on well-known classification problems. Section V concludes the paper.

II. Evolutionary Fuzzy Classifier Design

The proposed evolutionary fuzzy classifier design involves: 1) designing membership functions and determining a proper fuzzy partition approach for efficiently partitioning feature spaces, 2) determining a fuzzy reasoning method and fuzzy if-then rules corresponding to fuzzy regions, and 3) determining a fitness function and a chromosome representation for using IGA to optimize the system's tuning parameters.

A. Membership Function and Fuzzy Partition

Flexible generic parameterized fuzzy region can be determined by flexible generic parameterized membership functions (FGPMF's) and a hyperbox-type fuzzy partition of feature space. Each fuzzy region corresponds to a parameterized fuzzy rule. The major advantage of the parameterized fuzzy region approach is that only few overlapping fuzzy regions can cover all training patterns with high classification accuracy. For simplicity of explanation, each attribute value is assumed to be a real number in the unit interval [0,1] [3]. In experiments of our study, every attribute value is normalized into a real number in the unit interval [0,1]. An FGPMF $\mu(x)$ with a single fuzzy set is defined as

$$\mu(x) = \begin{cases} 0 \text{ if } x \le a \text{ or } x \ge d \\ \frac{x-a}{b-a} \text{ if } a < x < b \\ 1 \text{ if } b \le x \le c \\ \frac{d-x}{d-c} \text{ if } c < x < d \end{cases} \quad (1)$$

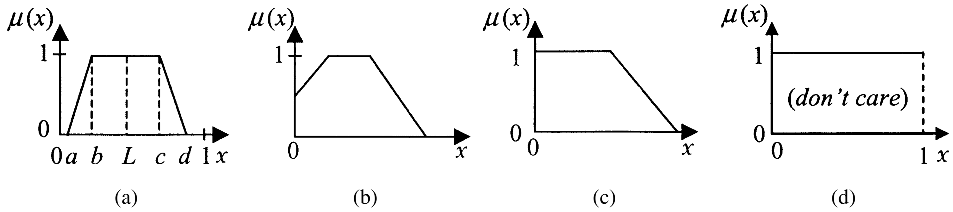

Fig. 1. Examples of FGPMF. (a) $a > 0$ and $d < 1$. (b) $a < 0 < b$. (c) $b \le 0$. (d) $b \le 0$ and $c \ge 1$.

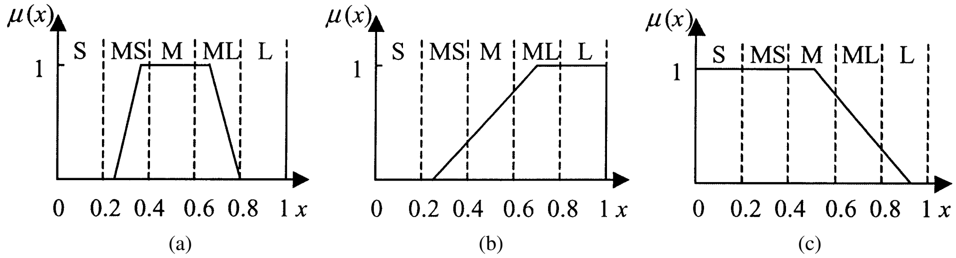

Fig. 2. Examples of an antecedent fuzzy set A_{ji} with linguistic values (S: small, MS: medium small, M: medium, ML: medium large, L: large). (a) A_{ji} represents {MS, M, ML}. (b) A_{ji} represents {MS, M, ML, L} or NOT small. (c) A_{ji} represents {S, MS, M, ML, L} or ALL.

where $x \in [0,1]$ and $a \le b \le c \le d$. The variables a, b, c, and d determining the shape of a trapezoidal fuzzy set are to be optimized.

Five parameters $V^1, V^2, \ldots, V^5 \in [0,1]$ instead of a, b, c and d are encoded in a chromosome for facilitating IGA. Let an additional variable $L = V^1$ where $b \le L \le c$. L determines location of the fuzzy set characterizing the occurrence of training patterns. Variables a, b, c, and d can be derived as

$$\begin{aligned} a &= L - (V^2 + V^3) \\ b &= L - V^3 \\ c &= L + V^4 \\ d &= L + (V^4 + V^5). \end{aligned} \tag{2}$$

Some examples of FGPMF are shown in Fig. 1. The advantages of the transformation are described as follows.

1) Confining all genetic searches within feasible regions. Notably, no inequality constraints are needed to define the relationship among parameters V^i like those for a, b, c, and d in (1). This transformation can always make the derived values of a, b, c, and d feasible. It is well recognized that confining genetic searches within feasible regions is often much more reliable than penalty approaches for handling constrained problems [25].
2) If $b \le 0$ and $c \le 1$, the condition is viewed as a "*don't care*" condition [see Fig. 1(d)]. Since *don't care* conditions can be omitted, short fuzzy rules with a less number of antecedent conditions can be obtained.
3) Reducing interaction effects between genes. The evolutionary search for the optimal location of fuzzy regions could become more efficient by evaluating V^1 independently. One of two parametric gene pairs, (V^2, V^3) and (V^4, V^5), is used to adjust one side of a trapezoidal fuzzy set. Two sides can be separately and independently adjusted. Reducing interaction effects between genes benefits not only IGA but also the standard GA (see experiments in Section IV-A).

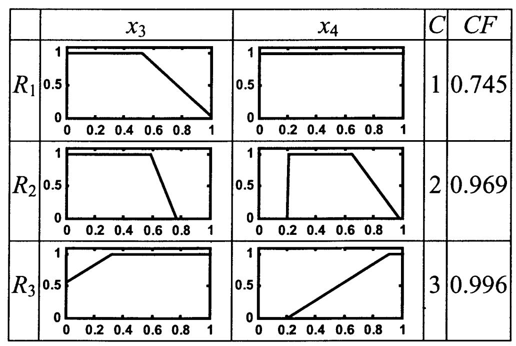

Fig. 3. Fuzzy rules for an iris classification problem using 50% of patterns for training and the remainder for testing. The recognition rates for training and test data are both 97.33% (73/75). The antecedent part of rule R_1 has a "don't care" condition for the feature x_4.

B. Fuzzy Rule and Fuzzy Reasoning Method

The following fuzzy if-then rules for n-dimensional pattern classification problems are used in our design of fuzzy classifier systems

R_j: If x_1 is A_{j1} and ... and x_n is A_{jn} then Class C_j with CF_j, $j = 1, \ldots, N$

where R_j is a rule label, x_i denotes a feature variable, A_{ji} is an antecedent fuzzy set, $C_j \in \{1, \ldots, C\}$ denotes a consequent class, C is a number of classes, CF_j is a certainty grade of this rule in the unit interval [0,1], and N is a number of fuzzy rules in the initial fuzzy-rule base.

To enhance interpretability of fuzzy rules, linguistic variables and fuzzy rules can be used in our classifiers. Each variable x_i takes values in [0,1] and has a linguistic set $U = \{\mathrm{S}, \mathrm{MS}, \mathrm{M}, \mathrm{ML}, \mathrm{L}\}$ for $i \in \{1, \ldots, n\}$. Each linguistic value of x_i equally represents 1/5 of the domain [0,1]. For example, $x_i = 0.3$ is MS (medium small). An antecedent fuzzy set $A_{ji} \in A_U$ where A_U denotes a set of subsets of U [5]. Examples of linguistic antecedent fuzzy sets are shown in Fig. 2. For example, in a computer simulation described in Section IV-B, we obtained the following three rules with two features for an iris classification problem (see Fig. 3).

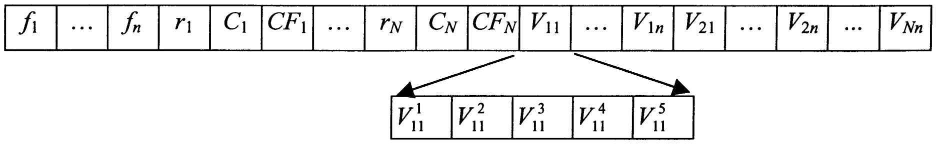

Fig. 4. Chromosome representation.

R_1: If x_3 is ALL then Class 1 with $CF_1 = 0.745$,
R_2: If x_3 is NOT large and x_4 is NOT small then Class 2 with $CF_2 = 0.969$, and
R_3: If x_3 is ALL and x_4 is NOT small then Class 3 with $CF_3 = 0.996$.

In the training phase, all the variables C_j and CF_j are treated as parametric genes encoded in chromosomes and their near-optimal values are obtained using IGA. In the test phase, to determine the class of an input pattern $x_p = (x_{p1}, x_{p2}, \ldots, x_{pn})$ based on voting by multiple fuzzy if-then rules that are compatible with x_p, the following fuzzy reasoning method is adopted.

Step 1) Calculate score $S_{\text{Class}v} (v = 1, \ldots, C)$ for each class as follows:

$$S_{\text{Class}\,v} = \sum_{\substack{R_j \in FC \\ C_j = \text{Class}\,v}} \mu_j(x_p) \cdot CF_j \tag{3}$$

where FC denotes the fuzzy classifier, the scalar value $\mu_j(x_p) = \mu_{j1}(x_{p1}) \ldots \mu_{jn}(x_{pn})$, and $\mu_{ji}(\cdot)$ represents the membership function of the antecedent fuzzy set A_{ji}.

Step 2) Classify x_p as the class with a maximum value of S_{Classv}.

C. Fitness Function and Chromosome Representation

Three objectives of designing an efficient fuzzy classifier FC using IGA are as follows:

1) to maximize the number NCP of the correctly classified training patterns;
2) to minimize the number N_{r} of fuzzy rules;
3) to minimize the number N_{f} of used features.

A three-objective GA can be used to find nondominated rule sets [3]. In this study, we combine these three objectives into a single scalar fitness function as

$$\max Fit(FC) = NCP - W_{\text{r}} \cdot N_{\text{r}} - W_{\text{f}} \cdot N_{\text{f}} \tag{4}$$

where W_{r} and W_{f} are positive weights. The weights should be specified based on the users' preference. In this study, we aim to obtain high classification accuracy. If it is in a tie situation, minimizing rule number is the second optimizing criterion. Similarly, minimizing feature number is the last optimizing criterion. Therefore, we use $W_{\text{r}} = 0.1$ and $W_{\text{f}} = 0.001$ for the desired optimizing criteria, i.e., to maximize classification accuracy, in a tie situation, to minimize rule number, and in a new tie situation also to minimize the used feature number.

A chromosome consists of control genes for selecting useful features and significant fuzzy rules, and parametric genes for encoding the membership functions and fuzzy rules. This design means that feature selection, rule selection, membership function tuning, consequent class determination, and rule certainty grade tuning are simultaneously determined to obtain a minimal number of fuzzy regions which can cover all training patterns with high classification accuracy.

The control gene comprises two types of parameters. One is parameter r_j, $j = 1, \ldots, N$, represented by one bit for eliminating unnecessary fuzzy rules. If $r_j = 0$, the fuzzy rule R_j is excluded from the rule base. Otherwise, R_j is included. The other is parameter f_i, $i = 1, \ldots, n$, represented by one bit for eliminating useless features. If $f_i = 0$, the feature x_i is excluded from the classifier. Otherwise, x_i is included. The parametric genes consist of three types:

1) $V_{ji}^k \in [0, 1]$, $k = 1, \ldots, 5$, for determining the antecedent fuzzy set A_{ji} for each feature variable x_i in rule R_j;
2) $C_j \in \{1, \ldots, C\}$ for determining the consequent class of rule R_j;
3) $CF_j \in [0, 1]$ for determining the certainty grade of rule R_j;

where $j = 1, \ldots, N$ and $i = 1, \ldots, n$. A rule base with N fuzzy rules is represented as an individual, as shown in Fig. 4. The number of encoding parameters to be optimized is equal to $N_{\text{p}} = n + 3N + 5Nn$. A chromosome representation uses a binary string for encoding control and parametric genes. There are 8 b for encoding one of parameters V_{ji}^k and CF_j.

Since each fuzzy region defines a fuzzy rule, the setting of number N is independent of value n but dependent on the number of fuzzy regions. Generally, N is set to the maximal number of possible fuzzy regions. In this study, N is set to $3C$. The design of an efficient fuzzy classifier is formulated as an LPOP. If the optimal or near-optimal solution to the LPOP can be found, an efficient fuzzy classifier can be obtained.

III. Solving the Design Problem Using IGA

The orthogonal experimental design (OED) of intelligent crossover is described in Section III-A. Section III-B presents the main power of IGA, i.e., the intelligent crossover. Finally, Section III-C gives the algorithm IGA.

A. Orthogonal Experimental Design

An efficient way to study the effect of several factors simultaneously is to use OED with both orthogonal array (OA) and factor analysis [21]. Many design experiments use OED for determining which combinations of factor levels (or treatments) to use for each experiment and for analyzing the experimental results. The factors are the variables (parameters), which affect the chosen response variable (fitness function), and a setting (or a discriminative value) of a factor is regarded as a level of the factor. The term "main effect" designates the effect on the response variable that one can trace to a design parameter.

OA is a matrix of numbers arranged in rows and columns where each row represents the levels of factors in each run and each column represents a specific factor that can be changed from each experiment. The array is called orthogonal because all columns can be evaluated independently of one another, and the main effect of one factor does not bother the estimation of the main effect of another factor.

Factor analysis using the OA's tabulation of experimental results can allow the main effects to be rapidly estimated, without the fear of distortion of results by the effects of other factors. Factor analysis can evaluate the effects of individual factors on the evaluation function, rank the most effective factors, and determine the best level for each factor such that the evaluation is optimized.

OED uses well-planned and controlled experiments in which certain factors are systematically set and modified, and then main effects of factors on the response can be observed. Therefore, OED using OA and factor analysis is regarded as a systematical reasoning method [21]. The merit of intelligent crossover is that the systematic reasoning ability of OED is incorporated to economically identify the good genes of parents and intelligently combine these good genes to generate offspring. The two-level OA used in the intelligent crossover is described below.

Let there be α factors with two levels for each factor. The number of experiments is 2^α for the popular "one-factor-at-a-time" study. Generally, levels 1 and 2 of a factor represent selected genes from parents 1 and 2, respectively. To use an OA of α factors with two levels, we obtain an integer $\beta = 2^{\lceil \log_2(\alpha+1) \rceil}$, build an orthogonal array $L_\beta(2^{\beta-1})$ with β rows and $\beta - 1$ columns, use the first α columns, and ignore the other $\beta - \alpha - 1$ columns. For instance, Table I shows an OA $L_8(2^7)$. OA can reduce the number of experiments for factor analysis. The number of OA experiments required to analyze a single factor is only β where $\alpha + 1 \leq \beta \leq 2\alpha$. An algorithm of constructing OA's can be found in [26].

After proper tabulation of experimental results, the summarized data are analyzed using factor analysis to determine the relative effects of levels of various factors. Let y_t denote a fitness value to be maximized for experiment t, where $t = 1, \ldots, \beta$. Define the main effect of factor j with level k as S_{jk} where $j = 1, \ldots, \alpha$ and $k = 1, 2$

$$S_{jk} = \sum_{t=1}^{\beta} y_l \cdot F_t \tag{5}$$

where $F_{\rm t} = 1$ if the level of factor j of experiment t is k; otherwise, $F_{\rm t} = 0$. If $S_{j1} > S_{j2}$, the level 1 of factor j makes a better contribution to the fitness function than level 2 of factor j does. Otherwise, level 2 is better. The most effective factor j has the largest main effect difference $|S_{j1} - S_{j2}|$.

Note that the main effect holds only when no interaction exists or when it is weak, and that makes the experiment meaningful. In order to achieve an effective design, experiments should be prepared so as to reduce interaction effects. In addition, to accurately estimate the main effect, all candidate solutions corresponding to the β conducted combinations need to be feasible for constrained problems if possible. The aim of the encoding scheme of FGPMF using parameters V^i instead of variables a, b, c and d is to simultaneously maintain feasibility of chromosomes and reduce interaction effects.

TABLE I
ORTHOGONAL ARRAY $L_8(2^7)$

Experiment	Factors							Fitness
no.	1	2	3	4	5	6	7	value
1	1	1	1	1	1	1	1	y_1
2	1	1	1	2	2	2	2	y_2
3	1	2	2	1	1	2	2	y_3
4	1	2	2	2	2	1	1	y_4
5	2	1	2	1	2	1	2	y_5
6	2	1	2	2	1	2	1	y_6
7	2	2	1	1	2	2	1	y_7
8	2	2	1	2	1	1	2	y_8

B. Intelligent Crossover

Each parameter is encoded in a chromosome using binary codes. Like traditional GA's, two parents P_1 and P_2 produce two children C_1 and C_2 in one crossover operation. If specific control parameters f_i or r_j in two parent chromosomes are all equal to zero, the corresponding governed parameters are not necessary to participate the crossover operation. The parameters having identical values in two parents do not participate the crossover operation such that the chromosomes can be temporally shorten possibly resulting in using a small OA table. Let the number of all participated parameters be randomly divided into α segments where each segment is treated as a factor. The following steps describe how to use OED to achieve intelligent crossover.

- Step 1) Use the first α columns of OA $L_\beta(2^{\beta-1})$ where $\beta = 2^{\lceil \log_2(\alpha+1) \rceil}$.
- Step 2) Let levels 1 and 2 of factor j represent the jth parameter of a chromosome coming from parents P_1 and P_2, respectively.
- Step 3) Evaluate the fitness values y_t for experiment t where $t = 2, \ldots, \beta$. The value y_1 is the fitness value of P_1.
- Step 4) Compute the main effect S_{jk} where $j = 1, \ldots, \alpha$ and $k = 1, 2$.
- Step 5) Determine the better one of two levels of each factor. Select level 1 for the jth factor if $S_{j1} > S_{j2}$. Otherwise, select level 2.
- Step 6) The chromosome of C_1 is formed using the combination of the better genes from the derived corresponding parents.
- Step 7) The chromosome of C_2 is formed similarly as C_1, except that the factor with the smallest main effect difference adopts the other level.
- Step 8) The best two individuals among P_1, P_2, C_1, C_2, and $\beta - 1$ combinations of OA are used as the final children C_1 and C_2 for elitist strategy.

One intelligent crossover operation takes $\beta + 1$ fitness evaluations, where $\alpha + 1 \leq \beta \leq 2\alpha$, to explore the search space of 2^α combinations. Generally, C_1 is a potentially good approximation to the best one of 2^α combinations. The larger the value of α, the more efficient it is the intelligent crossover if there exists no or weak interaction effect among gene segments. Considering the interaction effect, the smaller the value of α, the more accurate it is the estimated main effects of gene segments. Considering the tradeoff, an efficient criterion is to minimize the interaction effects while maximizing the value of α.

TABLE II
DATA SETS WITH NUMERICAL ATTRIBUTE VALUES. N_{p} IS THE NUMBER OF ENCODING PARAMETERS IN THE CHROMOSOMES OF IGA

Data set	Pattern number	Dimension n	Class number C	N_{p}
iris	150	4	3	211
wine	178	13	3	625
wdbc	569	30	2	948
heart-c[†]	297	13	5	1033
pima-diabetes	768	8	2	266
living-disorder	345	6	2	204
new-thyroid	215	5	3	257
haberman	306	3	2	111
glass	214	9	6	873
cmc	1473	9	3	441
sonar	208	60	2	1878

[†] Six patterns with missing attribute values are excluded.

For practical use, the proper value of α depends on the number of encoding parameters and their interaction effects. Generally, there are two approaches to specifying the value of α. One is to adaptively change the value of α during the evolution process [22]–[24]. To achieve an efficient coarse-to-fine search, α is gradually increased when the evolution proceeds [24]. The other is to use a constant value of α according to domain knowledge and simulation results.

C. Intelligent Genetic Algorithm IGA

IGA of the proposed method is given as follows:

Step 1) Initiation: Randomly generate an initial population with N_{pop} individuals.

Step 2) Evaluation: Evaluate fitness values of all individuals.

Step 3) Selection: Use the simple ranking selection that replaces the worst $P_{\mathrm{s}} \cdot N_{\mathrm{pop}}$ individuals with the best $P_{\mathrm{s}} \cdot N_{\mathrm{pop}}$ individuals to form a new population, where P_{s} is a selection probability. Let I_{best} be the best individual in the population.

Step 4) Crossover: Randomly select $P_{\mathrm{c}} \cdot N_{\mathrm{pop}}$ individuals including I_{best}, where P_{c} is a crossover probability. Perform intelligent crossover operations for all selected pairs of parents.

Step 5) Mutation: Apply a conventional bit-inverse mutation operator to the population using a mutation probability P_{m}. To prevent the best fitness value from deteriorating, mutation is not applied to the best individual.

Step 6) Termination test: If a prespecified termination condition is satisfied, stop the algorithm. Otherwise, go to Step 2.

IV. PERFORMANCE EVALUATION

The 11 well-known data sets with numerical attribute values, as shown in Table II, are used to demonstrate the superiority of the proposed method. All the data sets are available from [27]. The set of test classification problems is composed of problems with various dimensions from 3 to 60 and various degrees of overlapping that the general test accuracy ranges from 50% to 100%. All the feature values are normalized to real numbers in the unit interval [0,1].

A standard GA (SGA) with elitist strategy derived by MIT GALib [28] is used to demonstrate high search ability of IGA. The C4.5 release 8 algorithm [29] is compared with to demonstrate high test classification accuracy and compactness of the IGA fuzzy classifiers. Both IGA and SGA are implemented using C++ on a PC with Pentium III/1 GHz processor. The parameters of the IGA-based method are as follows unless otherwise specified: $N_{\mathrm{pop}} = 20$, $P_{\mathrm{c}} = 0.7$, $P_{\mathrm{m}} = 0.01$, $P_{\mathrm{s}} = 1 - P_{\mathrm{c}} = 0.3$, and $\alpha = 15$. For handling classification problems with various dimensions, the stopping criterion of IGA is to use $N_{\mathrm{fit}} = 100N_{\mathrm{p}}$ for all problems. The genetic parameters of SGA are the same as those of IGA. The default parameter settings of C4.5 are used.

In Section IV-A, the sensitivity of control parameters of the proposed method is empirically analyzed to show the robustness of the IGA-based method. In Section IV-B, some experiments are used to demonstrate high performance of the proposed method.

A. Sensitivity Analysis

The performance of classifiers using an evolutionary design should not be influenced too much by the control parameters of GA. In order to show the robustness of the proposed method, we analyze the performance of IGA using various parameter combinations with 30 independent runs each and determine a set of parameter values in default of domain knowledge. Each data set is randomly divided into two disjoint sets of equal size. One set is used for training and the other for testing.

1) Sensitivity of Genetic Parameters P_{c} and P_{m}: Generally, crossover probability P_{c} and mutation probability P_{m} are the major factors influencing the performance of GA. Usually, the value of P_{c} is greater than 0.5 and P_{m} takes values in [0.005, 0.05] [30]. Conventionally, $P_{\mathrm{s}} = 1 - P_{\mathrm{c}}$. We analyze the performance of the proposed method with various combinations of P_{c} and P_{m}, where P_{c} takes values in {0.6,0.7,0.8,0.9} and P_{m} takes values in {0.005,0.01,0.05}.

The statistical results of IGA based on fitness values for three data sets (iris, wine, and heart-c) are shown in Table III. The results show that the variance of fitness values for all combinations is relatively small and there is no combination of P_{c} and P_{m} which is the best one in all the three classification problems. It reveals that P_{c} and P_{m} are not sensitive to the performance of

TABLE III
STATISTICAL RESULTS OF IGA BASED ON FITNESS VALUES FOR THREE DATA SETS USING 12 COMBINATIONS OF P_c AND P_m. THE VALUES OF P_c ARE 0.6, 0.7, 0.8, AND 0.9. THE VALUES OF P_m ARE 0.005, 0.01, AND 0.05

Data set	Worst (P_c, P_m)	Mean	Variance	Best (P_c, P_m)
iris	72.598 (0.6, 0.05)	73.793	0.146	74.698 (0.7, 0.01)
wine	86.695 (0.8, 0.01)	87.696	0.413	88.697 (0.8, 0.005)
heart-c	104.093 (0.7 0.05)	109.514	6.826	114.394 (0.9, 0.05)

TABLE IV
PERFORMANCE OF CLASSIFIERS USING VARIOUS COMBINATIONS OF W_r AND W_f. THE VALUES OF W_r ARE 0.1, 0.3, AND 0.5. THE VALUES OF W_f ARE 0.001, 0.005, AND 0.01. $TrCR$ IS THE TRAINING CLASSIFICATION RATE

Data set	*TrCR* (%)		N_r		N_f	
	Mean	Variance	Mean	Variance	Mean	Variance
iris	98.87	0.12	3.35	0.01	1.80	0.02
wine	99.83	0.08	3.52	0.02	4.54	0.03
heart-c	73.61	0.24	5.80	0.16	6.46	0.08

IGA. Due to the number of encoding parameters in a chromosome is large where $N_p = 1033$, the variance of heart-c can be further decreased using a large value of N_{fit}.

2) Sensitivity of Weights W_r and W_f: Since the preferred order of objectives is as follows: 1) classification accuracy, 2) rule number, and 3) feature number, the relationship of weights is specified as $W_f < W_r < 1$. To show that the weights W_r and W_f in the fitness function are not sensitive to the obtained classifiers, nine combinations of W_r and W_f are conducted, where W_f takes values in $\{0.001, 0.005, 0.01\}$ and W_r takes values in $\{0.1, 0.3, 0.5\}$. The experimental results are shown in Table IV. There is no combination of W_r and W_f which is the best one in all the three classification problems. The small variances of training classification rate $(TrCR)$, rule numbers (N_r), and feature numbers (N_f) of obtained classifiers reveal that IGA is efficient for finding good solutions and the weights W_r and W_f are not sensitive.

3) Sensitivity of Factor Number α: To understand how does α affect the performance of IGA, various fixed values of α are conducted. To make use of all columns of OA, α is usually set to $2^k - 1$ where k is an integer. Let MAX be the number of parameters participated in the intelligent crossover operation. $\alpha = MAX$ means that one parameter is treated as a factor. The $\alpha - 1$ cut points are randomly specified. Considering two high-dimensional data sets, typical results of convergence for various values of α are shown in Fig. 5.

From the simulation results, some observations are given as follows.

i) The performance of $\alpha = MAX$ is relatively inefficient in a limited amount of computation time. One reason is that the interaction effect between factors is relatively large. The other is that one generation takes many fitness evaluations.
ii) The performance of $\alpha = 3$ is better in the early evolution but worse in the later evolution because the effect of OED in the intelligent crossover is weak.
iii) The performance of $\alpha = 15$ is generally the best for all experiments in this study, including the results of Fig. 5.
iv) Compared with IGA, it is difficult for SGA with one-cut-point crossover to effectively improve the fitness value

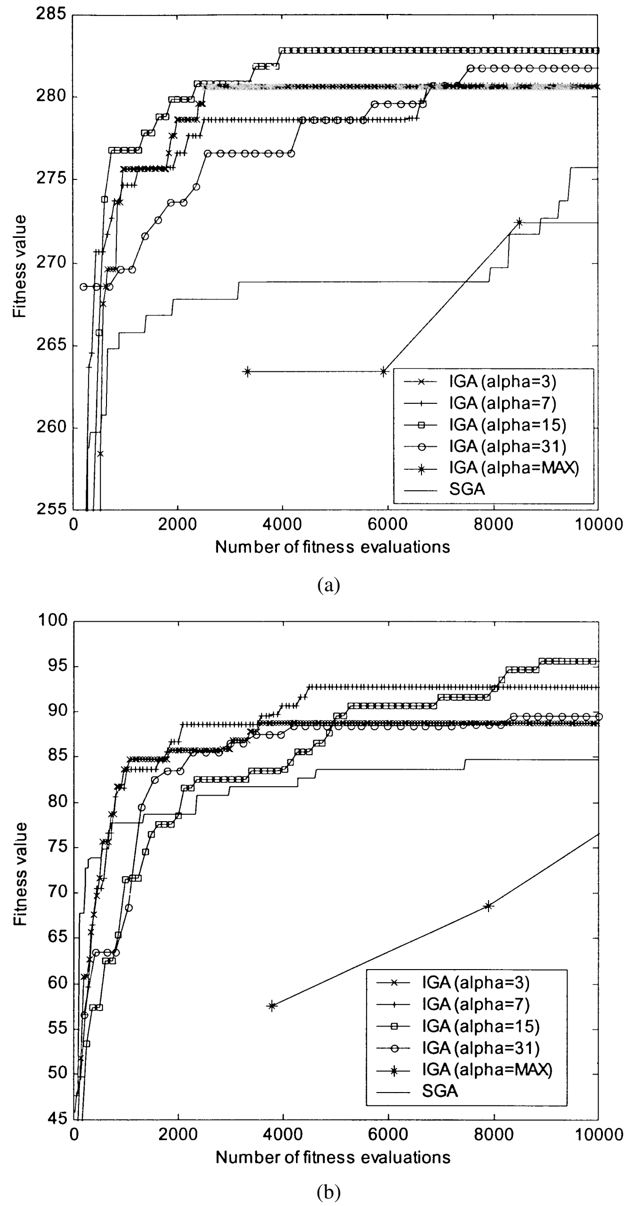

Fig. 5. Convergences of various values of the factor number α (alpha). Each mark on the curve denotes as the result of one generation: (a) wdbc data set and (b) sonar data set.

in the later evolution, especially for the case with lots of encoding parameters.

If the computation time can be properly increased, the final fitness values of IGA with various values of α are almost similar according to computer simulation.

4) Sensitivity of FGPMF Parameters: The degree of freedom of a trapezoid fuzzy set is 4. In order to facilitate IGA, the proposed encoding method for FGPMF uses five encoding parameters without constraints. The variables a, b, c, and d to be optimized can also be derived using four encoding parameters without constraints: $b = v^1$, $a = b - v^2$, $c = b + v^3$, and $d = c + v^4$.

Both encoding methods are used to encode fuzzy sets. The typical performance for IGA and SGA using the sonar data set is shown in Fig. 6. From the experimental results, it can be found that:

i) the five-parameter encoding method performs better than the four-parameter one for both IGA and SGA. Although the number of encoding parameters is larger, the experimental results reveal that reducing interaction effects between genes is important and the proposed encoding method is effective;
ii) IGA performs better than SGA for both four- and five-parameter encoding methods. The fitness value of IGA with the four-parameter method is slightly better than that of SGA with five-parameter method. Generally, for a small parameter optimization problem, the contribution of reducing interaction effects to evolution performance is larger than using IGA instead of SGA. For a very large parameter optimization problem, IGA plays an important role in solving the investigated design problem.

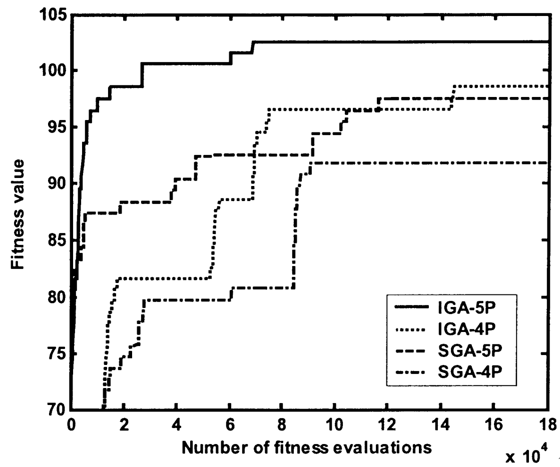

Fig. 6. Performances of IGA and SGA with encoding methods of four parameters (4P) and five parameters (5P) using the sonar data set. Final fitness values: $\mathrm{IGA-5P} = 102.591, \mathrm{IGA-4P} = 98.595, \mathrm{SGA-5P} = 97.485$, and $\mathrm{SGA-4P} = 91.792$.

B. Performance Comparisons

The proposed method using a scatter partition tries to maximize classification accuracy and minimize the numbers of used features and fuzzy rules. Due to different aims and merits of both grid partition and tree partition approaches described in Section I, the performance of the proposed approach cannot be directly compared with those of nonscatter partition approaches in justice. However, some performance comparisons with the fuzzy grid partition approach [3], [4], the tree partition method C4.5 [29], and the fuzzy scatter partition methods [10]–[12] are given to demonstrate the following three merits of the proposed method: 1) the proposed method has high search ability to efficiently find fuzzy rule-based systems with high fitness values, 2) obtained fuzzy rules have high interpretability, and 3) obtained compact classifiers have high test classification accuracy.

A ten-fold cross validation method (10-CV) [31] is adopted to compare the test performance of C4.5 with that of the proposed method. The performance is based on multiple independently formed training and test sets. For 10-CV, each data set is randomly divided into 10 disjoint sets of equal size. Each set in turn is used for testing and the remainder for training. The classifier is trained 10 times, each time with a different set held out as a test set. The estimated performance is the mean of these 10 results.

1) Search Ability of IGA: The average performances of the IGA classifiers using 10-CV from 30 independent runs per classifier are shown in Table V. Since fitness value is the only guide for GAs in the evolution, the search abilities of GAs can be compared by the fitness value using the same value of N_{fit}. The average performance of SGA using $N_{\mathrm{fit}} = 100\ N_{\mathrm{p}}$ from 30 runs is shown in Table VI. A paired t-test with 29 degrees of freedom on the fitness value is also given to show that the search ability of IGA is statistically significantly better than that of SGA. In the paired t-test the null hypothesis is that the average of the differences between the paired observations in two samples is zero. If the calculated P-value is less than the conventional 0.05, the conclusion is that the mean difference between the paired observations is statistically significantly different from zero. Table VI reveals that the fitness performance of IGA is indeed statistically significantly better than that of SGA for all classification problems where the probabilities (P-values) are less than 0.005. As a result, the training classification rates $(TrCR)$ of IGA are better than those of SGA. It is worthwhile to mention that the computation time of IGA is much smaller than that of SGA for all problems (65.8% on an average). The reason is that one intelligent crossover of IGA uses 17 ($\alpha = 15$ and $\beta = 16$) fitness evaluations and thus IGA takes a smaller number of generations than SGA. In other words, the intelligent crossover can make GAs more efficient in both fitness performance and convergence speed for large parameter optimization problems in spite of the multiple fitness evaluations per recombination [32].

To further show that the IGA-based method has high search ability for designing fuzzy classifiers, computation experiments using all patterns in each one of two well-known data sets iris and wine as training data will be examined. Figs. 7 and 8 show the encouraging results, i.e., a 100% training classification rate with three rules for three-class problems. It is well known that there are three patterns in the iris data set that is difficult to be accurately classified using the fuzzy grid-partition method with a small number of fuzzy rules [3]. Furthermore, for the wine data set, the least number of fuzzy rules with 100% classification rate is 6 and the maximal classification rate of the classifier with

TABLE V

AVERAGE PERFORMANCES OF THE IGA CLASSIFIERS USING 10-CV FROM 30 INDEPENDENT RUNS PER CLASSIFIER

Data set	Fitness			*TrCR*(%)	N_r	N_f	*TeCR*(%)	N_r/C
	Avg.	Best	Worst					
iris	132.500	132.947	131.967	98.41	3.54	2.61	95.09	1.2
wine	159.088	159.714	158.294	99.55	3.90	5.92	93.70	1.3
wdbc	502.678	503.682	500.951	98.23	3.26	9.14	95.55	1.6
heart-c	182.431	187.383	179.813	68.52	7.25	6.90	55.71	1.5
pima-diabetes	553.401	556.664	550.515	80.12	3.84	5.03	75.19	1.9
living-disorder	234.371	239.925	228.856	75.60	3.61	4.39	67.22	1.8
new-thyroid	188.859	190.186	187.066	97.82	4.24	3.44	93.97	1.4
haberman	219.283	220.367	218.238	79.75	3.34	2.58	73.90	1.7
glass	140.780	144.645	137.465	73.47	7.15	5.20	62.19	1.2
cmc	770.993	785.255	750.445	58.20	5.15	4.92	54.28	1.7
sonar	163.078	165.541	160.553	87.31	3.51	17.98	71.54	1.8
Average	295.224	298.755	291.288	83.36	4.44	6.19	76.18	1.55

TABLE VI

SEARCH ABILITY COMPARISON OF IGA AND SGA USING STATISTICAL ANALYSIS, A PAIRED T-TEST WITH 29 DEGREES OF FREEDOM ON FITNESS VALUES

	IGA			SGA			t-test on Fitness	
Data set	Fitness	*TrCR*(%)	Time(s)	Fitness	*TrCR*(%)	Time(s)	*t*	*P*
iris	132.500	98.41	3.4	132.184	98.17	4.1	3.099	0.004
wine	159.088	99.55	24.5	158.495	99.18	36.6	3.993	0.000
wdbc	502.678	98.23	147.5	501.967	98.08	213.9	3.360	0.002
heart-c	182.431	68.52	109.9	176.717	66.34	172.5	9.951	0.000
pima-diabetes	553.401	80.12	26.2	545.165	78.92	27.3	12.626	0.000
living-disorder	234.371	75.60	6.7	225.832	72.84	7.9	11.572	0.000
new-thyroid	188.859	97.82	6.5	186.274	96.48	9.3	12.268	0.000
haberman	219.283	79.75	2.6	216.417	78.70	3.1	7.476	0.000
glass	140.780	73.47	50.5	138.441	72.22	96.0	4.174	0.000
cmc	770.993	58.20	101.7	739.818	55.84	114.5	14.301	0.000
sonar	163.078	87.31	209.7	161.557	86.48	361.9	5.094	0.000
Average	295.224	83.36	62.65	289.352	82.11	95.19	7.992	0.001

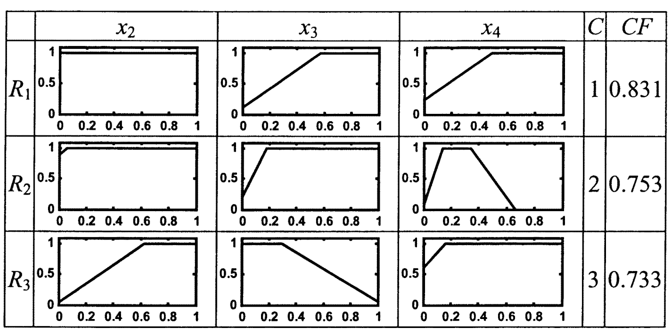

Fig. 7. Rule base for the iris classifier with a 100% training classification rate using all patterns as training data.

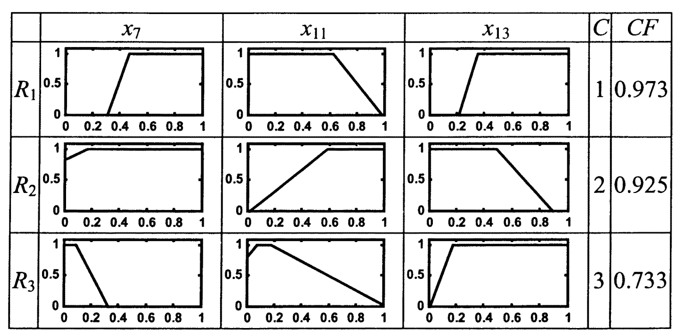

Fig. 8. Rule base for the wine classifier with a 100% training classification rate using all patterns as training data.

three rules is 97.2% [3]. Moreover, a 100% classification rate was obtained by three fuzzy rules in [4]. The excellent performance arises from both the proposed FGPMF with its encoding method and the high search ability of IGA for solving the large parameter optimization problem.

2) Interpretability of Fuzzy Rules: Table V shows that the average number of fuzzy rules per class is smaller than 2 for each IGA classifier that $N_r/C = 1.55$ on an average. Generally, the scatter partition method cannot compete with the conventional fixed linguistic grid partition method in the interpretability of fuzzy rules. However, the number of fuzzy rules obtained by the proposed method is much smaller than those of the grid-partition-based method [3], [4]. One of the advantages of the proposed method is that only few overlapping fuzzy regions can cover all training patterns with high classification accuracy.

To further realize the performance of the proposed method in the aspects of rule number and test classification rate, we compare the IGA classifier with the existing scatter partition methods: a) fuzzy classifier with hyperbox regions (Hyperbox) [10], b) fuzzy classifier with ellipsoidal regions (Ellipsoidal) [11], and c) neural network classifier with polyhedron regions (Polyhedron) [12]. The test performances of various iris

TABLE VII
TEST PERFORMANCES OF VARIOUS IRIS CLASSIFIERS

Method	*TeCR* (%)	# rules
IGA (hyperbox)	98.67	3
Hyperbox [10]	97.33	17
Hyperbox [10]	92.00	5
Ellipsoidal [11]	98.67	3
Polyhedron [12]	97.33	48

classifiers using 50% of patterns for training and the remainder for testing are shown in Table VII. The classification result of the IGA classifier with $TrCR = 100\%$ and three rules is $TeCR = 98.67\%$ (one misclassification). Table VII shows that the IGA classifier using a hyperbox type fuzzy partition is superior to the Hyperbox and Polyhedron methods, and performs as well as the Ellipsoidal method. Few results of high-dimensional fuzzy classifiers with ellipsoidal regions reported for fair comparisons with the proposed method. Notably, design of high-dimensional fuzzy classifiers with hyperbox fuzzy regions needs fewer tuning parameters than that of the classifiers with ellipsoidal fuzzy regions.

In the proposed method, "*don't care*" condition and genetic feature selection can be used to shorten the length of fuzzy rules that can make the rule base more compact. It is well recognized that a compact rule base is more easily interpretable than a complex one. Fig. 3 is a typical example of a fuzzy-rule base for the iris classifier and its interpretable fuzzy rules can be found in Section II-B. Considering the scatter-partition-based fuzzy classifiers, the proposed IGA-based method can obtain compact fuzzy rule-based systems with high interpretability.

Considering the fitness function in (4) where $W_r = 0.1$ and $W_f = 0.001$, the preferred order of objectives is as follows: 1) classification accuracy, 2) rule number, and 3) feature number. It means that the training classification rate $TrCR$ must be maximized first and then the compactness of the fuzzy system can be minimized. Generally, maximizing $TrCR$ may involve the risk of overtraining resulting in high generalization errors for conventional classifier designs. Due to 1) the nature clustering property of patterns, 2) the flexible generic parameterized fuzzy region, and 3) the strong search ability of IGA, the proposed method using $N = 3C$ (the maximal number of fuzzy rules per class is three) can maximize $TrCR$ and further minimize N_r and N_f without fear of overtraining.

In the following performance comparison with C4.5, we will show that the IGA classifiers are compact and accurate for unseen test patterns. Therefore, the proposed method using IGA with control genes can be widely used for feature selection [22] and knowledge acquisition because its advantages: 1) no additional domain knowledge is required, 2) default parameter settings are efficient enough, 3) no additional problem-dependent parameters are needed, and 4) the global feature selection considering interaction and system performance can be simultaneously optimized.

3) Classification Accuracy and Compactness: Since there are few results of high-dimensional fuzzy classifiers with scatter partitions reported for fair comparisons, we compare the IGA classifier with C4.5 using the significance analysis on 10-CV

TABLE VIII
PERFORMANCE OF C4.5 WITH UNPRUNED TREES

Data set	*TrCR*(%)	N_r	N_f	*TeCR*(%)	N_r/C
iris	98.00	4.8	2.0	94.67	1.6
wine	98.75	6.1	4.5	90.46	2.0
wdbc	99.16	12.4	8.4	93.49	6.2
heart-c	87.24	56.7	12.9	48.46	11.3
pima-diabetes	83.83	23.2	7.1	69.80	11.6
living-disorder	86.22	28.0	6.0	67.81	14.0
new-thyroid	98.24	8.0	4.4	93.92	2.7
haberman	76.91	4.8	2.4	72.20	2.4
glass	92.99	25.9	8.7	65.39	4.3
cmc	79.62	257.4	9.0	48.07	85.8
sonar	97.92	14.5	12.3	74.55	7.3
Average	90.81	40.16	7.06	74.44	13.56

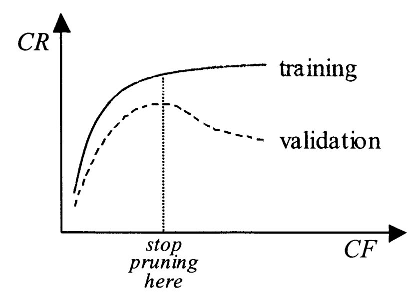

Fig. 9. For 10-CV, the data set is randomly split into two parts. The training set (90% of patterns) is used to set free parameters in the classifier model; the validation set (10%) is used to estimate the generalization rate of a classifier. The C4.5 classifiers are obtained by tuning the parameter CF to maximize test classification rate (CR).

to demonstrate that the proposed method can obtain compact and accurate fuzzy classifiers on unseen test patterns. Table VIII shows the performance of C4.5 with unpruned trees. For the average case of 11 classification problems, the training classification rate $TrCR$ is as high as 90.81%. The high classification accuracy of the C4.5 classifiers for training data doesn't mean it has high accuracy on unseen test patterns. The test classification rate $TeCR = 74.44\%$ and the number of rules per class $N_r/C = 13.56$. Note that the IGA classifiers have average performance $TrCR = 83.36\%$, $TeCR = 76.18\%$, and $N_r/C = 1.55$.

Typically, tree pruning can make the C4.5 classifiers more compact while maintaining high test classification rates. Therefore, we fairly compare the IGA classifiers with the best pruned classifiers of C4.5 having high test classification rate and compactness. We adjust the certainty factor parameter, CF, (default value $CF = 25$ in the tool) of C4.5 to prune the decision trees for obtaining high-performance classifiers with low generalization errors. Typically, the classification rate (CR) on the training set increases monotonically while the value of CF increases, resulting in an increasing number of rules used, as sketched in Fig. 9. For most problems, the accuracy on the validation set increases, but then decreases. An indication is that the classifier may overfit the training data. Therefore, in validation, the parameter adjustment is stopped at the maximum of the test classification rate.

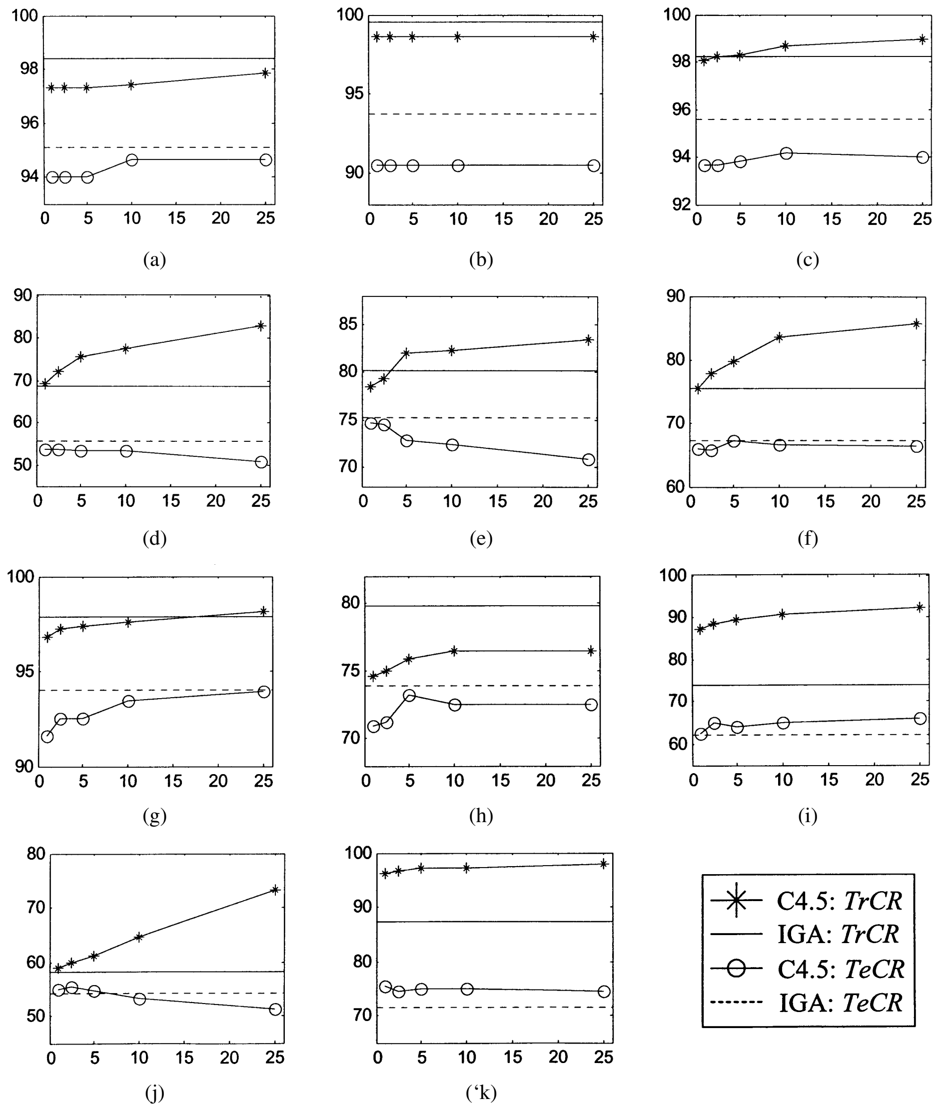

Fig. 10. Classification rates of the IGA and C4.5 classifiers. Horizontal and vertical axes are certainty factor (CF) and classification rate $(CR$, percentage), respectively: (a) iris, (b) wine, (c) wdbc, (d) heart-c, (e) pima-diabetes, (f) living-disorder, (g) new-thyroid, (h) haberman, (i) glass, (j) cmc, and (k) sonar.

Five values of CF, 1, 2.5, 5, 10, and 25, are adopted to sketch the cross validation performance of C4.5 with pruned trees. The classification rates and rule numbers of the C4.5 classifiers with various values of CF are shown in Figs. 10 and 11. The number of rules of the C4.5 classifier is calculated by looking at the number of leaves in the tree. For clear comparisons by visualization, the average values of $TrCR$, $TeCR$, and $N_{\rm r}$ of the IGA classifiers are also shown in Figs. 10 and 11. From the average test classification rates of 11 classification problems, $CF = 5$ can obtain the highest test performance $TeCR = 75.56\%$, as shown in Table IX. Table IX reveals that C4.5 with $CF = 5$ can obtain more accurate and compact classifiers ($N_{\rm r}/C = 4.14$ on an average). The results of a paired t-test with 29 degrees of freedom on $TeCR$, $N_{\rm r}$, and $N_{\rm f}$ for comparisons of IGA and C4.5 with $CF = 5$ are shown in Table X. The statistical analysis of Table X is discussed below.

Considering the performance on $TeCR$ for 11 classifiers, IGA has eight wins over C4.5 and three losses (glass, cmc, and sonar). When carefully check the classifier living-disorder with an associate P value of $P = 0.732$, and $TeCR = 67.22\%$ and 67.21% for IGA and C4.5, respectively, the two classifiers are statistically significantly equivalent to each other. However, IGA uses 3.61 rules, 4.39 features and C4.5 uses 13.0 rules, 5.0 features for living-disorder. Considering the three lost classifiers (glass, cmc, sonar) for which accuracy is lower, the values of $TeCR$ are (62.19%, 54.28%, 71.54%) and (63.98%, 54.65%, 75.02%) for IGA and the C4.5 classifiers, respectively. However, the IGA classifiers use much smaller numbers of rules (7.15, 5.15, 3.51) than C4.5 (18.6, 22.0, 13.0).

Considering the performance on $N_{\rm r}$, IGA has 10 wins over C4.5 and only one loss that all P-values are less than 0.001. The only lower accuracy classifier was from the low-dimensional haberman data set which has dimension $n = 3$ and $C = 2$. The result comes from that IGA aims to obtain compact and accurate classifiers and the obtained classifier is indeed compact ($N_{\rm r} = 3.34$ and $N_{\rm f} = 2.58$) and accurate $(TeCR = 73.90\%)$ The $TeCR = 73.18\%$ for the C4.5 classifier. Considering the average case, $N_{\rm r}/C = 1.55$ and 4.14 for IGA and C4.5, respectively. The t and P values of $N_{\rm r}$ reveal that the IGA classifier is accurate and has a significantly small number of fuzzy rules.

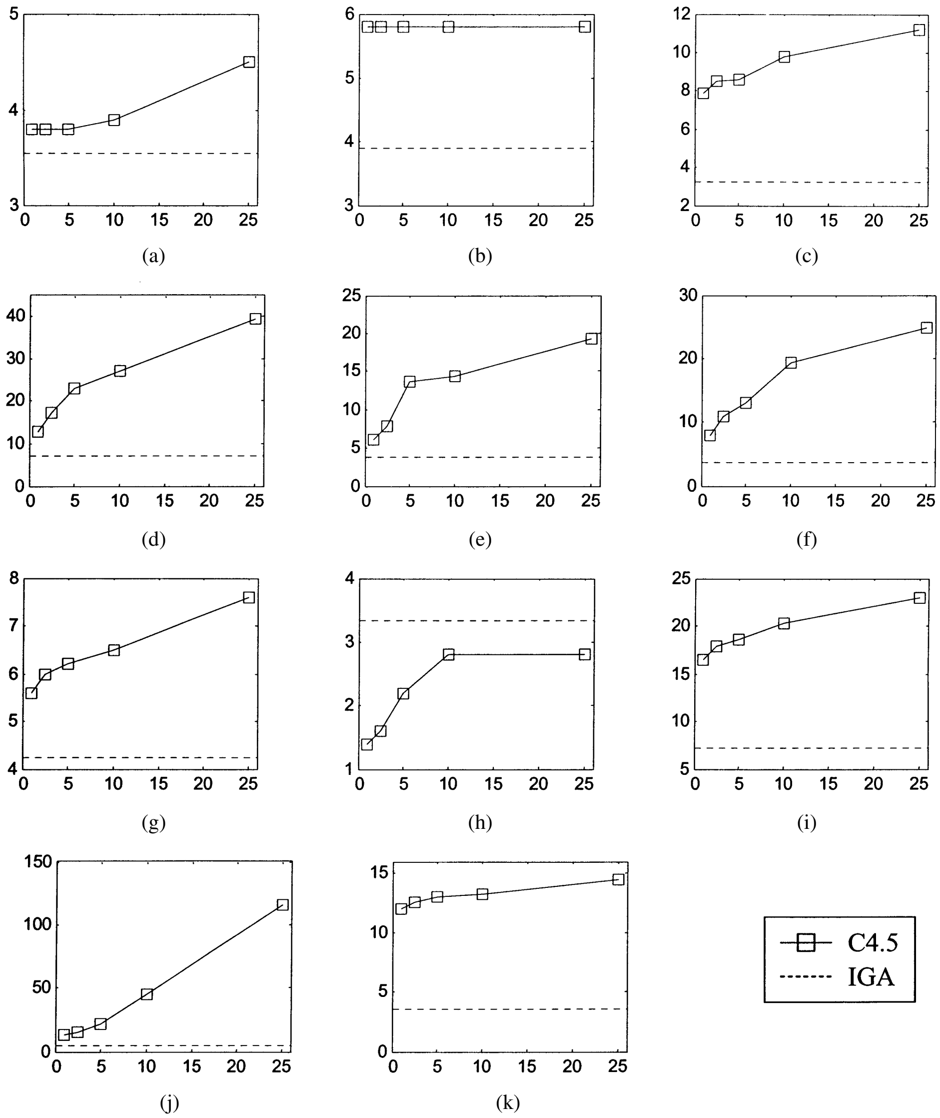

Fig. 11. Rule numbers of the IGA and C4.5 classifiers. Horizontal and vertical axes are certainty factor (CF) and number of rules (N_r), respectively: (a) iris, (b) wine, (c) wdbc, (d) heart-c, (e) pima-diabetes, (f) living-disorder, (g) new-thyroid, (h) haberman, (i) glass, (j) cmc, and (k) sonar.

TABLE IX
PERFORMANCE OF C4.5 WITH THE PRUNED TREES OF $CF = 5$ USING 10-CV

Data set	*TrCR*(%)	N_r	N_f	*TeCR*(%)	N_r/C
iris	97.33	3.8	2.0	94.00	1.3
wine	98.63	5.8	5.0	90.46	1.9
wdbc	98.30	8.6	5.0	93.84	4.3
heart-c	75.49	23.0	12.0	53.48	4.6
pima-diabetes	82.05	13.6	6.0	72.79	6.8
living-disorder	79.71	13.0	5.0	67.21	6.5
new-thyroid	97.37	6.2	4.0	92.49	2.1
haberman	75.82	2.2	2.0	73.18	1.1
glass	89.35	18.6	8.0	63.98	3.1
cmc	61.13	22.0	7.0	54.65	7.3
sonar	97.06	13.0	12.0	75.02	6.5
Average	86.57	11.80	6.18	75.56	4.14

Considering the performance on N_f, IGA has six wins over C4.5 and five losses that all P values are smaller than 0.001. From the average performance of 11 classification problems that IGA has $N_f = 6.19$ and C4.5 has $N_f = 6.18$, it can be recognized that the used feature numbers of two methods are nearly the same. It is worthwhile to mention that minimization of the used feature number is the last objective of the IGA classifier. For feature selection, the proposed method performs as well as C4.5 with the best value of CF does.

Generally speaking, the optimal design of fuzzy classifiers is a three-objective optimization problem in essence [3]. Unlike the single-objective optimization, multi-objective optimization problems cannot satisfactorily be characterized by a single performance measure, but often can be characterized by distinct measures of multiple incommensurable and competing objectives. Due to the nature of tradeoffs involved, the optimal design of fuzzy classifiers seldom has a unique solution [33]. Considering the performances of the classifier on the three objectives $TeCR$, N_r, and N_f simultaneously, the classifier A is said to dominate the classifier B if there exists at least one objective performance of A is statistically significantly better than that of B and the remainder (if any) are statistically significantly equivalent to that of B. Table X reveals that the IGA classifier dominates the C4.5 classifier in the four data sets heart-c, pima-diabetes, living-disorder, and new-thyroid. On the other

TABLE X
RESULTS OF A PAIRED T-TEST WITH 29 DEGREES OF FREEDOM FOR IGA AND C4.5 WITH THE PRUNED TREES OF $CF = 5$

Data set	t-test on *TeCR*		t-test on N_r		t-test on N_f	
	t	*P*	*t*	*P*	*t*	*P*
iris	6.614	0.000	-6.542	0.000	18.712	0.000
wine	9.250	0.000	-46.627	0.000	13.560	0.000
wdbc	13.788	0.000	-128.969	0.000	35.204	0.000
heart-c	7.149	0.000	-271.125	0.000	-70.634	0.000
pima-diabetes	19.310	0.000	-159.260	0.000	-15.200	0.000
living-disorder	0.348	0.732	-162.444	0.000	-12.223	0.000
new-thyroid	6.485	0.000	-35.235	0.000	-10.933	0.000
haberman	3.156	0.004	25.612	0.000	25.829	0.000
glass	-4.160	0.000	-131.910	0.000	-54.278	0.000
cmc	-2.266	0.031	-328.030	0.000	-34.780	0.000
sonar	-6.998	0.000	-181.939	0.000	29.712	0.000
IGA: Win/Loss	8/3		10/1		6/5	

hand, the C4.5 classifier does not dominate the IGA classifier in any data set. The performance comparisons show that the IGA-based method can generate accurate and compact fuzzy classifiers with rules of high interpretability.

V. CONCLUSION

This paper proposes an automated method for designing accurate classifiers with a compact fuzzy-rule base using an evolutionary scatter partition of feature space. A novel flexible generic parameterized membership function associated with an efficient encoding method is proposed to achieve an efficient evolutionary scatter partition. The proposed method includes almost all aspects related to the design of compact fuzzy rule-based classification systems: feature selection, rule selection, membership function tuning, consequent class determination, and certainty grade tuning. Thus, the efficient fuzzy classifier system design is formulated as a large parameter optimization problem (LPOP).

To solve the LPOP, an efficient optimization algorithm IGA is used. The superiority of the proposed method has been demonstrated by computer simulation on 11 well-known classification problems in the following three aspects: 1) the proposed method has high search ability to efficiently find fuzzy rule-based systems with high fitness values, 2) obtained fuzzy rules have high interpretability, and 3) obtained compact classifiers have high classification accuracy on unseen test patterns. The performance comparison and statistical analysis of experimental results using ten-fold cross validation have shown that the IGA-based method without heuristics is efficient in designing accurate and compact fuzzy classifiers with rules of high interpretability.

REFERENCES

[1] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[2] H. Roubos and M. Setnes, "Compact and transparent fuzzy models and classifiers through iterative complexity reduction," *IEEE Trans. Fuzzy Syst.*, vol. 9, pp. 516–524, Aug. 2001.

[3] H. Ishibuchi, T. Nakashima, and T. Murata, "Three-objective genetics-based machine learning for linguistic rule extraction," *Information Sci.*, vol. 136, no. 1–4, pp. 109–133, Aug. 2001.

[4] H. Ishibuchi and T. Yamamoto, "Fuzzy rule selection by data mining criteria and genetic algorithms," in *Proc. Genetic Evolutionary Computat. Conf. (GECCO-2002)*, 2002, pp. 399–406.

[5] L. Castillo, A. Gonzalez, and R. Perez, "Including a simplicity criterion in the selection of the best rule in a genetic fuzzy learning algorithm," *Fuzzy Sets Syst.*, vol. 120, no. 2, pp. 309–321, June 2001.

[6] J. Casillas, O. Cordon, M. J. Del Jesus, and F. Herrera, "Genetic feature selection in a fuzzy rule-based classification system learning process for high-dimensional problems," *Information Sci.*, vol. 136, pp. 135–157, Aug. 2001.

[7] J. Yen, "Fuzzy logic—a modern perspective," *IEEE Trans. Knowledge Data Eng.*, vol. 11, pp. 153–165, Jan.–Feb. 1999.

[8] C. Z. Janikow, "A genetic algorithm for optimizing fuzzy decision trees," *Information Sci.*, vol. 89, no. 3–4, pp. 275–296, Mar. 1996.

[9] ——, "Fuzzy decision tree: issues and methods," *IEEE Trans. Syst., Man, Cybern. B*, vol. 28, pp. 1–14, Feb. 1998.

[10] S. Abe and M.-S. Lan, "A method for fuzzy rules extraction directly from numerical data and its application to pattern classification," *IEEE Trans. Fuzzy Syst.*, vol. 3, pp. 18–28, Feb. 1995.

[11] S. Abe and R. Thawonmas, "A fuzzy classifier with ellipsoidal regions," *IEEE Trans. Fuzzy Syst.*, vol. 5, pp. 358–368, Aug. 1997.

[12] V. Uebele, S. Abe, and M.-S. Lan, "A neural-network-based fuzzy classifier," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, pp. 353–361, Feb. 1995.

[13] R. Thawonmas and S. Abe, "A novel approach to feature selection based on analysis of class regions," *IEEE Trans. Syst., Man, Cybern. B*, vol. 27, pp. 196–207, Apr. 1997.

[14] D. P. Mandal, "Partitioning of feature space for pattern classification," *Pattern Recog.*, vol. 30, no. 12, pp. 1971–1990, Dec. 1997.

[15] S. Medasani, J. Kim, and R. Krishnapuram, "An overview of membership function generation techniques for pattern recognition," *Int. J. Approx. Reasoning*, vol. 19, no. 3–4, pp. 391–417, Oct. 1998.

[16] A. Homaifar and E. McCormick, "Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms," *IEEE Trans. Fuzzy Syst.*, vol. 3, pp. 129–139, May 1995.

[17] K.-S. Tang, K.-F. Man, Z.-F. Liu, and S. Kwong, "Minimal fuzzy memberships and rules using hierarchical genetic algorithms," *IEEE Trans. Ind. Electron.*, vol. 45, pp. 162–169, Feb. 1998.

[18] T. Murata, S. Kawakami, H. Nozawa, M. Gen, and H. Ishibuchi, "Three-objective genetic algorithms for designing compact fuzzy rule-based systems for pattern classification problems," in *Proc. Genetic Evolutionary Computat. Conf.*, 2001, pp. 485–492.

[19] H. Ishibuchi and T. Nakashima, "Effect of rule weights in fuzzy rule-based classification systems," *IEEE Trans. Fuzzy Syst.*, vol. 9, pp. 506–515, Aug. 2001.

[20] K. Kumar, S. Narayanaswamy, and S. Garg, "Solving large parameter optimization problems using a genetic algorithm with stochastic coding," in *Genetic Algorithms in Engineering and Computer Science*, G. Winter, J. Périaux, M. Galán, and P. Cuesta, Eds. New York: Wiley, 1995.

[21] S. H. Park, *Robust Design and Analysis for Quality Engineering*. London, U.K.: Chapman & Hall, 1996.

[22] S.-Y. Ho, C.-C. Liu, and S. Liu, "Design of an optimal nearest neighbor classifier using an intelligent genetic algorithm," *Pattern Recog. Lett.*, vol. 23, no. 13, pp. 1495–1503, Nov. 2002.

[23] S.-Y. Ho and Y.-C. Chen, "An efficient evolutionary algorithm for accurate polygonal approximation," *Pattern Recog.*, vol. 34, no. 12, pp. 2305–2317, Nov. 2001.

[24] H.-L. Huang and S.-Y. Ho, “Mesh optimization for surface approximation using an efficient coarse-to-fine evolutionary algorithm,” *Pattern Recog.*, vol. 36, no. 5, pp. 1065–1081, June 2003.
[25] Z. Michalewicz, D. Dasgupta, R. G. Le Riche, and M. Schoenauer, “Evolutionary algorithms for constrained engineering problems,” *Comput. Ind. Eng.*, vol. 30, no. 4, pp. 851–870, Sept. 1996.
[26] Y.-W. Leung and Y. Wang, “An orthogonal genetic algorithm with quantization for global numerical optimization,” *IEEE Trans. Evol. Comput.*, vol. 5, pp. 41–53, Feb. 2001.
[27] UCI Repository of Machine Learning Databases, C. L. Blake and C. J. Merz. (1998). *http://www.ics.uci.edu/~mlearn/MLRepository.html* [Online]
[28] M. B. Wall. (1999) *The GALib Genetic Algorithm Package* [Online] http://lancet.mit.edu/ga/
[29] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kauffman, 1993.
[30] D. A. Coley, *An Introduction to Genetic Algorithms for Scientists and Engineers*. Singapore: World Scientific Publishing, 1999.
[31] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York: Wiley-Interscience, 2000.
[32] S.-Y. Ho, J.-H. Chen, and M.-H. Huang, “Inheritable genetic algorithm for bi-objective 0/1 combinatorial optimization problems and its applications,” *IEEE Trans. Syst., Man, Cybern. B*, to be published.
[33] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York: Wiley, 2001.

Shinn-Ying Ho (M’00) was born in Taiwan, R.O.C., in 1962. He received the B.S., M.S., and Ph.D. degrees in computer science and information engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1984, 1986, and 1992, respectively.

He is currently a Professor in the Department of Information Engineering and Computer Science, Feng Chia University, Taichung, Taiwan. His research interests include image processing, pattern recognition, virtual reality applications of computer vision, fuzzy classifier, evolutionary algorithms, large parameter optimization problems, and system optimization.

Dr. Ho is listed in *Who’s Who in Science and Engineering* and *Who’s Who in Finance and Industry*.

Hung-Ming Chen (S’03) was born in Taiwan, R.O.C., in 1977. He received the B.A. and M.S. degrees in information engineering and computer science from Feng Chia University, Taichung, Taiwan, in 1999 and 2001, respectively, where he is currently pursuing the Ph.D. degree in the Department of Information Engineering.

His research interests include evolutionary computation, fuzzy systems, and pattern recognition.

Shinn-Jang Ho (M’03) was born in Taiwan, R.O.C., in 1960. He received the B.S. degree in power mechanic engineering from National Tsing Hua University, Hsinchu, Taiwan, in 1983, and the M.S. and Ph.D. degrees in mechanical engineering from National Sun-Yat-Sen University, Kaohsiung, Taiwan, in 1985 and 1992, respectively.

He is currently an Associate Professor in the Department of Automation Engineering, National Huwei Institute of Technology, Huwei, Yulin, Taiwan. His research interests include optimal control, fuzzy systems, genetic algorithms, and system optimization.

Tai-Kang Chen was born in Taiwan, R.O.C., in 1976. He received the B.S. degree in computer science from Tatung Institute of Engineering, Taichung, Taiwan, in 1998 and the M.S. degree in information engineering and computer science from Feng Chia University, Taichung, in 2000.

His research interests include fuzzy systems, control systems, and genetic algorithms.