# Intelligent and Robust Genetic Algorithm Based Classifier

S. H. Zahiri, H. Rajabi Mashhadi and S. A. Seyedin

**Abstract:** The concepts of robust classification and intelligently controlling the search process of genetic algorithm (GA) are introduced and integrated with a conventional genetic classifier for development of a new version of it, which is called Intelligent and Robust GA-classifier (IRGA-classifier). It can efficiently approximate the decision hyperplanes in the feature space.
It is shown experimentally that the proposed IRGA-classifier has removed two important weak points of the conventional GA-classifiers. These problems are the large number of training points and the large number of iterations to achieve a comparable performance with the Bayes classifier, which is an optimal conventional classifier.
Three examples have been chosen to compare the performance of designed IRGA-classifier to conventional GA-classifier and Bayes classifier. They are the Iris data classification, the Wine data classification, and radar targets classification from backscattered signals. The results show clearly a considerable improvement for the performance of IRGA-classifier compared with a conventional GA-classifier.

**Keywords:** Intelligent genetic classifiers, robust genetic classifiers, fuzzy controller, genetic algorithm, optimum decision hyperplanes.

## 1 Introduction

Genetic algorithms (GAs) have been shown to be an effective stochastic search algorithm in high dimensional spaces. They are inspired by the biological process of Darwin's evolution theory, where selection, mutation and crossover play important roles [1].
GAs have been applied to solve pattern recognition and data classification problems by finding decision boundaries and hyperplanes. This new evolutionary classifier is called GA-classifier [2].
It is shown theoretically and experimentally that the performance of a GA-classifier for sufficiently large number of iterations and infinitely number of training data points is comparable to Bayes classifier which is the optimal classifier [3]. It is important to mention that the optimal Bayesian classifier needs a priori knowledge but GA-classifier doesn't need any important priori knowledge.

Also a variable string length GA-classifier (VGA-classifier) proposed evolving the number of hyperplanes automatically [4] and another VGA-classifier with chromosome differentiation (VGACD-classifier) designed for pixel classification in [5]. The fitness functions, defined in all of these researches are the number of misclassified training points. Although the designed GA-classifiers may classify the training points as well as, or better than other conventional classifiers, e.g. multi-layer-perceptron (MLP), k-nearest neighbor and Bayes classifier, but its performance has not this strength against the test points. A multiobjective GA has been recently introduced in [6] for simultaneously optimization three objectives, which are number of misclassified points, class-accuracy and the number of hyperplanes.
As mentioned above, the better performance of GA-classifier, for all of these researches, happens for a *large number of training* points and a *large number of iterations*. In fact these are two conditions, which are necessary for conventional GA-classifier to reach a comparable performance with Bayes classifier as an optimal classifier [2-6].
Obviously, if the number of training points is sufficiently large, the probability distribution functions (PDF) can be estimated and with known PDFs, Bayes classifier is the best candidate to find the decision functions in feature space, because of its simple

Iranian Journal of Electrical & Electronic Engineering, 2005.
Paper first received 17th April 2005 and in revised from 11th March 2006.
S. H. Zahiri is with the Department of Electronics and Communication, Birjand University, Birjand, Iran. P.O. Box: 97175-376. H. Rajabi Mashhadi and S. A. Seyedin are with the Department of Electrical Engineering Ferdowsi University of Mashhad, Mashhad, Iran.
E-mail: hzahiri@birjand.ac.ir, rajabi@ferdowsi.um.ac.ir, seyedin@ferdowsi.um.ac.ir.

structure and the best performance and it is not reasonable to use the GA-classifier with low convergence rate.
In this paper a novel approach is proposed to remove these two weak points. The basic idea is *to maximize the margins of hyperplanes from the different classes* using a proper definition of the fitness function. It has been shown mathematically in [7] that maximizing the margins of hyperplanes from different classes can minimize the risk of error in the classification.
To show the stability of the method against the variation of the number of training points ($n$) a new index, named *robustness* index, is defined as a metric. A GA-classifier with a high value of robustness against the value of $n$, named in this article a *Robust GA-classifier (RGA-classifier)*.
Another important concept introduced in this article is to steer the GA-classifier efficiently to the global solution while genetic algorithm is running. For this purpose, an intelligent mutation and crossover rate controller is designed using a fuzzy structure to develop an *Intelligent and Robust GA-classifier (IRGA-classifier)*. This intelligent fuzzy controller, not only can chase away the genetic algorithm from the local solutions, but also can reduce the necessary number of iterations considerably. Thus a common problem of conventional GA-classifiers in previous researches, i.e. poor convergence of the search process due to the *large number of iterations*, is improved.
The rules for designing the fuzzy controller were extracted from some theoretical and experimental results have reported in researches on GA operators [8-12].
We used *Fuzzy controlled and Robust GA-classifier (FCRGA-classifier)*, and a Simple GA-classifier (SGA-classifier) for determining the hyperplanes for two common benchmark problems and a special problem in pattern recognition. Iris data and Wine data classification are common problems in pattern recognition researches with low and medium feature space dimensions, and automatic target recognition in continuous wave radars is a special pattern recognition problem with high feature space dimensions. We compared the scores of recognition and the number of iterations is needed for convergence for FCRGA-classifier and SGA-classifier. To see the robustness of designed IRGA-classifier, we also compared its performance with the Bayes classifier for different training points, because it is optimal classifier when the probability density function of features is known.
The results show that FCRGA-classifier has more accuracy compared with a SGA-classifier with a less number of iterations and high robustness value. Also the performance of this IRGA-classifier is comparable to Bayes classifier for a low number of training points.
In this paper, Section 2 explains the structure of a real-valued genetic algorithm based classifier (GA-classifier). Intelligent robust genetic classifiers are then described in Section 3. Section 4 considers experimental results on three pattern recognition problems, which are Iris data classification, Wine data classification and radar target classification. Finally, Section 5 concludes the paper.

## 2 Structure of a real- valued GA-classifier

A general hyperplane is in the form

$$d(X) = w_1 x_1 + w_2 x_2 + .... + w_n x_n + w_{n+1} \qquad (1)$$

Where $X = (x_1, x_2, ..., x_n, 1)'$ and $W = (w_1, w_2, ..., w_n, w_{n+1})'$ are called the augmented feature (pattern) and weight vectors respectively.
In a general case, there are a number of hyperplanes that separate the feature space to different regions, which each region distinguishes an individual class (Fig. 1). In Fig. 1, IR denotes the indeterminate region.

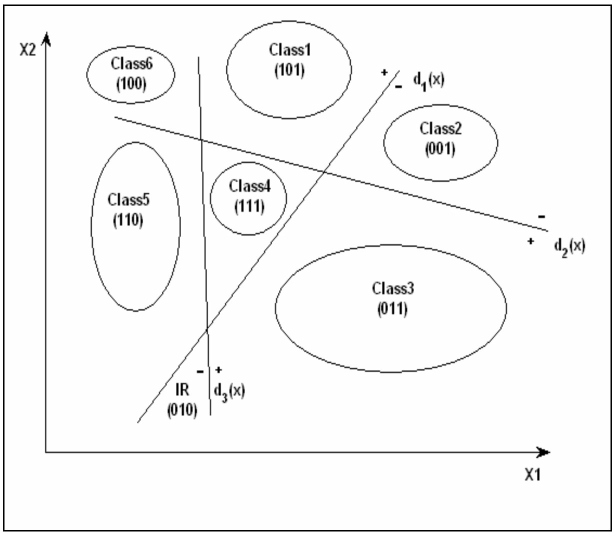


**Fig. 1** A general case, which each region can identify an individual class by its code obtained from the sign of hyperplanes.

Some especial cases are described in the text books of Pattern Recognition course [e.g. 13].
A real-valued GA–classifier should find $W_i$ (i=1,2,…,M) in solution space. We consider *real-valued* GA-classifier because binary coded GAs are less efficient when applied to multi dimensional problems. The bit-strings can become very long and the search space blows up. In real-valued GAs, the variables appear directly in the chromosome and are modified by mutation and recombination (crossover) operators. Various real-valued-GA were reviewed in [14].The basic steps for designing a real-valued GA-classifier are as follows:

### 1-Generation

At the first step a random initial population, S(0) is generated . S(0) includes N chromosomes and the i'th chromosome is of the form $[W^0_{i1}, W^0_{i2}, ..., W^0_{iM}]'$ for M classes.

**2-Fitness evaluation**

At the second step the value of fitness function of S(0) is computed . In a simple GA-classifier, which introduced in previous researches the fitness function is defined as the number of misclassified training points [2-6].

**3-Basic loop**

The body of a GA is:

```
While (Termination Condition)
    Loop :
        Compute fitness(S(q));
        q=q+1;
        S(q)=Select (S(q-1));
        S(q)=Crossover S(q);
        S(q)=Mutate (S(q));
    End Loop;
End While;
```

The termination condition can be implemented using the best fitness value or a default maximum number of iterations. In q'th iteration the fitness value is computed and then a new population is generated by three important genetic operators: selection, crossover (recombination), and mutation. These operators described as follows:

**-Selection**

Selection is the process of determining the number of times or trials that a particular individual is chosen for reproduction and, thus the number of offsprings that an individual will produce. Many selection techniques employ "roulette wheel" mechanism to probabilistically select individuals proportional to their fitness value. Also the best individuals are transmitted to the next generation without any additional process (elitism strategy).

**-Crossover**

The basic operator for producing new chromosomes in the GA is the crossover. Like its counterpart in nature, crossover produces new individuals that have some parts of both parent's genetic material. We used a simple arithmetic form of crossover, which is single point crossover, described as follows:

Assume $W_{ik}^{q}$ and $W_{jp}^{q}$ are the i'th and j'th hyperplane of k'th and p'th chromosomes in the q'th iteration respectively. Then $W_{ik}^{q}$ and $W_{jp}^{q}$ are crossed over at the l'th position. The resulting offsprings are

$$W_{ik}^{q+1} = (w_{i1}, w_{i2}, \ldots, w_{il}, w_{jl+1}, w_{jl+2}, \ldots, w_{jn+1}) \quad (2)$$

$$W_{jk}^{q+1} = (w_{j1}, w_{j2}, \ldots, w_{jl}, w_{il+1}, w_{il+2}, \ldots, w_{in+1}) \quad (3)$$

Where *l* is a random number from {2,…,n} and n is the feature space dimension. Crossover rate (CR) is the number of times that crossover operator is applied to the population.

**-Mutation**

In GAs, mutation is randomly applied with a known probability. It modifies elements in the chromosomes. It means that a position in an individual is selected randomly and the value in this position is changed.

We used Gaussian mutation mechanism that mutates some elements of an individual such that

$$W_{ik}^{q+1} = (w_{i1}, w_{i2}, \ldots, w'_{il_1}, \ldots, w'_{il_2}, \ldots, w_{in+1}) \quad (4)$$

Where $l_r$ belongs to [1,n+1] interval and is randomly selected. Also $w'_{il_r} = w_{il_r} + z_{l_r}$. Here $z_{l_r}$ is a random number drown from a Gaussian distribution with zero mean and adaptive variance.

## 3 What is intelligent and robust GA-classifier (IRGA-classifier)?

As it mentioned in Section 1 the *large number of training points* and *iterations* are two necessity, which are necessary for conventional GA-classifier to reach a comparable performance with Bayes classifier as an optimal classifier.

In this article it has been tried to remove the aforesaid problems in GA-classifiers. For this reason at first a review on optimal hyperplane for separation of two classes is presented. Then we define an efficient fitness function for obtaining these optimal hyperplanes by a GA-classifier. At the next step a new concept is defined and is called the *robustness* of performance of a GA-classifier against the number of data points.

Eventually designing an intelligent mutation rate and crossover rate controller is considered to help GA-classifier to obtain near the optimal hyperplanes.

### A. Optimum Hyperplanes

A conceptual problem for computing a linear decision function to separate two different classes is determining a hyperplane which has a better performance against receiving the next data points (or test points).

Since we suppose, there is no *priori* knowledge about the distribution of data points in feature space, the optimal hyperplanes are defined as the linear decision function with maximal margin between the feature vectors of different classes this margin is calculated based on the available training points of the classes. This strategy is similar to what has been done in Support Vector Machine (SVM) studies [7].

Fig. 2 shows three hyperplanes ($d_1(x)$, $d_2(x)$ and $d_{opt}(x)$) that all of them can be considered as the decision functions, which separate two Class1 and Class2 successfully. It can be seen from this figure that if any small noise is added to training points, as a test data (or a new training point) near this decision functions, it can impair the recognition score ( or change the position) of $d_1(x)$ and $d_2(x)$.

Among $d_1(x)$, $d_2(x)$ and $d_{opt}(x)$, only $d_{opt}(x)$ separates the training data with a maximal margin from two classes. Since the margins of two classes from the $d_{opt}(x)$ are equal, the new noisy testing points in the classes (or new training points in them) can have a little effect on the recognition score (or variation of) $d_{opt}(x)$.

### B. Fitness Function Definition

It should be mentioned that the fitness function, which has been used in previous researches [2-6] is "*the number of miss-classified data points*". Thus a conventional GA-classifier may converge to each of $d_1(x)$ or $d_2(x)$ in Fig.2 that have the minimum miss-classified data points, but are not optimal decision functions. In fact, fitness function definition as the misclassified data points, need to have *many data points* in the training phase for convergence to the best hyperplanes in a simple GA-classifier. This is a weakness aspect for conventional GA-classifiers, which has been proved theoretically and experimentally.

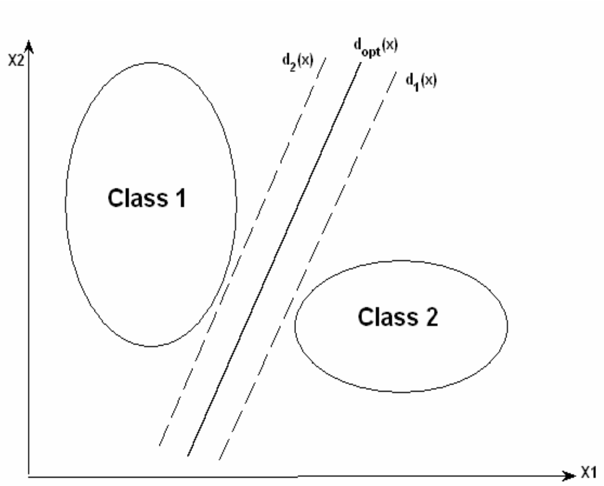


**Fig. 2** $d_{opt}(x)$ is the optimal decision function among other decision functions.

Maximizing the minimum value of the average of the Euclidean distances of all data points in each class is a good algorithm, which can displace $d_1(x)$ and $d_2(x)$ toward $d_{opt}(x)$. We named this algorithm Max-Min algorithm. To see the efficiency of Max-Min algorithm for chancing away $d_1(x)$ and $d_2(x)$ toward $d_{opt}(x)$, suppose that a conventional GA-classifier converged to $d_1(x)$, which has minimum average distance from Class2. By entering the Max-Min algorithm in fitness function, GA-classifier must maximize the margin of $d_1(x)$ from Class2. Thus $d_1(x)$ tends to $d_{opt}(x)$. Same condition is appeared if GA-classifier is converged to $d_2(x)$.

Due to above descriptions, we defined a modified fitness function as:

$$\text{fitness}(W) = \text{penalty} + \min(\overline{\Sigma d_i}, \overline{\Sigma d_j}) \tag{5}$$

In (5), *fitness (W)* is the value of fitness function for the chromosome *W*. The *penalty* is a negative, absolutely large value, which is used if the hyperplane obtained by *W,* doesn't classify all data points in different classes $C_i$ and $C_j$. In this paper penalty is defined as *10*min(*$\overline{\Sigma d_i}$*,* $\overline{\Sigma d_j}$*)*Miss*, which *Miss* is the number of misclassified data points by hyperplanes obtained by a chromosome *W*. $\overline{\Sigma d_i}$ is the average of the Euclidean distances of all data points in class *i* from the hyperplane *W*, and $\min(\overline{\Sigma d_i}, \overline{\Sigma d_j})$ is the minimum value of $\overline{\Sigma d_i}$ and $\overline{\Sigma d_j}$. The GA-classifier proposed in this article maximizes the fitness function (or minimize its negative value).

Obviously, when the second term in definition of *fitness (W) is* set to zero, the same fitness of a conventional GA-classifier is obtained.

The first and necessary condition for a good performance of Equation 5 is that the classes are separable from each other. Your comment is correct when classes have overlap in some patterns. In this case we attach the misclassified patterns to one class and then we use Equation 5 as the fitness function. It means that we accept a little error to adjust decision hyperplanes just between classes.

### C. Robust GA-classifier (RGA-classifier)

To evaluate the performance of modified fitness function defined by (5) and compare it with conventional fitness function definition in previous researches, a new concept of *robustness* is introduced.

In this article the concept of robustness is defined as the inverse of the sensitivity of the solutions (hyperplanes) of a GA-classifier against the variation of number of data points for each class. More robustness for a GA-classifier concludes more stability for the obtained optimum solution. We defined the robustness of a GA-classifier as follows:

$$Robustness = \left[\frac{\Delta(\text{Perf})}{\Delta n} * \frac{n}{\text{Perf}}\right]^{-1} \tag{6}$$

In (6) $Perf$ is the performance of the best hyperplanes have been found by GA-classifier. *n* is the number of data points and $\Delta Perf$ is defined as the difference between two obtained performance in two different number of training points. A robust GA-classifier (RGA-classifier) has a low sensitivity against changing the number of training points.

In fact the robustness is a good metric to see how the modified fitness function can remove one of the important defects in conventional GA-classifier, which is *need to a large number of training points to reach to an optimal performance.*

### D. Intelligent and Robust GA-classifier (IRGA-classifier)

The search mechanism of an evolutionary algorithm likeness GA is based on three important operators: Selection, Mutation and Crossover. All of these operators have probabilistically events, but by different effects on search process, thus *intelligently controlling* them can help the genetic algorithm to escape from local solutions and converge to global solution by a faster rate.

A RGA-classifier with an intelligent CR and MR controller is called *intelligent and robust genetic classifier* (IRGA-classifier). In this paper, we designed a fuzzy structure to control adaptively CR and MR in each iteration and called it Fuzzy Controlled and Robust GA-classifier (FCRGA-classifier). The fuzzy controller is constructed on some fuzzy (IF antecedents THEN consequents) rules. Each input and output variable are defined with their membership functions.

We defined three inputs for fuzzy controller in a FCRGA-classifier as follows:

*Fit-dist*: the distance between fitness value of the best individual in q'th iteration and maximum of fitness function. Based on Equation (5) the fitness of each chromosome is related to the number of misclassified training points and margins between different classes. The number if training points is known and the minimum distance of the patterns which exist in each class can be found. Thus *an approximated value* of *Fit-dist* is available in each problem.

*UN*: The number of iterations whose fitness value is constant.

*t*: the number of iterations.

Two outputs of fuzzy controller are *crossover rate (CR)* and *mutation rate (MR)*.

The normalized membership functions of Fit-dist, UN and t are shown in Fig. 3.

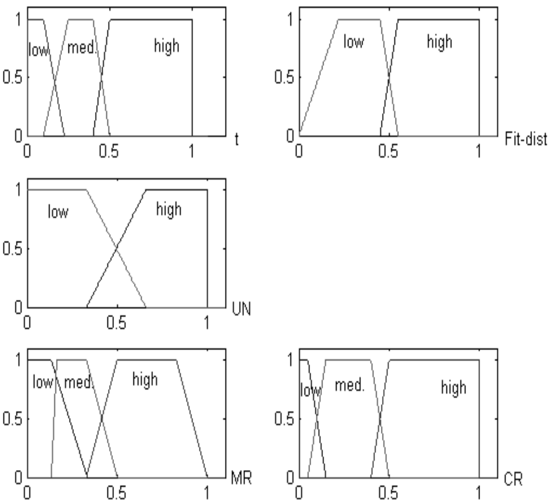


**Fig. 3** Membership functions of inputs (Fit-dist, UN, and t) and outputs (CR, MR) in fuzzy controller.

The selection of the shapes of membership functions and their locations are based on widely study on researches were related to GA and appeared as a survey in [8] and other references (some of them are [9-12]).

To extract some effective fuzzy rules, we know that crossover facilitates exploration while mutation facilitates exploitation in solution space. This means that when the best fitness stuck at one value for a long time (*UN* is High), the GA is often stuck at a local minimum, so the crossover rate should be decreased and mutation rate should be increased. Low fitness values (or High values of *Fit-dist*) often happen in the start of GA (*t* is low) and we need more exploitation and less exploration. Thus crossover rate should be decreased and mutation rate should be increased and contrariwise if fitness value is increased (*Fit-dist* tends to Low values) crossover rate should be increased and mutation rate should be decreased.

On the other hand it should be mentioned that although all of the theoretical and experimental researches have been done on the optimal MR and CR were constructed under some conditions or for a few test functions, but they have a common aspect, which is a decreasing schedule for MR and CR as the number of iterations (*t*) is increased. Specially in [11], Schmitt presented an annealing schedules for MR and CR with respect to *t* which guaranty the convergence to global solution:

$$MR(t) = \phi_m . t^{-1/\kappa . L_p} < \frac{1}{2} \tag{7}$$

$$CR(t) = \phi_c . [MR(t)]^{1/m} \tag{8}$$

Where $\phi_m \in R^+ \setminus \{0\}$ and $\kappa \in [1, \infty)$ can be chosen and $L_p$ is the population size and both $\phi_c \in (0, 2^{1/m}]$ and $m \in [1, \infty)$ can be chosen.

Of course decreasing the MR and CR must be based on an improvement in fitness values (or decreasing the *Fit-dist*).

From above linguistic descriptions seven rules for the fuzzy controller are defined as below:

a) IF (t) is low and (Fit-dist) is high THEN (MR) is high and (CR) is low.
b) IF (t) is medium and (Fit-dist) is high THEN (MR) is medium and (CR) is low.
c) IF (t) is high and (Fit-dist) is low THEN (MR) is low and (CR) is high.
d) IF (Fit-dist) is high and UN is low THEN (MR) is medium and (CR) is medium.
e) IF (Fit-dist) is high and UN is high THEN (MR) is high and (CR) is low.
f) IF (Fit-dist) is low and UN is high THEN (MR) is medium and (CR) is low.
g) IF (Fit-dist) is low and UN is low THEN (MR) is low and (CR) is high.

It must be mentioned that another inputs, outputs, membership function shapes and fuzzy rules may be introduced and even these parameters can be optimized by another optimization algorithm [15-17].

## 4 Implementation and Results

Three pattern recognition problems with different augmented feature vectors dimensions (5,4,129) were used to demonstrate the effectiveness of the IRGA-classifier. A description of the data sets is given here:

### A. Data Sets

*Iris Data*: Iris Data contains 50 measurements of four features from each three species Iris setosa, Iris versicolor, and Iris virginica [18]. Features are sepal length, sepal width, petal length and petal width.

*Wine Data:* Wine data contains the chemical analysis of wines grown in the same region in Italy but derived from different cultivars [19]. The 13 continuous attributes are available for classification. Total number of instances is 178 which we classified them in five classes.

*Radar Targets:* An application of pattern recognition is Automatic Target Recognition (ATR) for continuous wave radars. In this paper Jet Engine Modulations (JEM) is used for this purpose. In this approach the modulation of the radar wave by rotating propellers and jet engine blades of targets is considered [20,21]. Ten different flying objects were chosen as introduced in [21] for classification in 20 º elevation angle. After sampling from backscattered signals and data reduction preprocess, we took a 128 points FFT as feature vectors for each target.

### B. Comparison with Existing Methods

The performance of proposed IRGA-classifier is compared with the performance of a simple GA-classifier (SGA-classifier) and Bayes classifier. The genetic parameters in SGA-classifier were selected as conventional GA-classifiers were proposed in previous researches [2-6]. The crossover probability is fixed at 0.8. Available value of mutation probability is selected from the range [0.015, 0.333]. A fixed population size of 20 is chosen for both SGA-classifier and IRGA-classifier. For Bayes classifier, a *priori* probabilities equal to $tr_i/tr$ , for $tr_i$ patterns from class i, and totally *tr* training samples , and a multivariate normal distribution of the samples are considered. This is similar to Bayesian classifier is used in [2-5] to show the improvements compared to SGA-classifiers introduced in these researches.

### C. Experimental Results

The proposed IRGA-classifier (i.e. FCRGA-classifier), SGA-classifier and Bayes classifier are tested on the data sets described in section 4-A. We implemented different classifiers for ten times with random selected of training sets. Thus the results report the average score of recognition for ten times repeats. Table 2 and Table 3 present the results corresponding to Iris data and Wine data classification for different number of training samples (*n*=5,10,15), for 5 number of hyperplanes (H=5).

The calculated robustness (defined by (6)) in some different number of training points for Iris data and Wine data are given in Table 4 and Table 5, respectively . The robustness of the IRGA-classifier and a SGA-classifier are appeared in these Tables, with respect to 5 training points as a reference for (*n*, *Perf*).

The results in Table 2 to Table 5 have some meaningful concepts:

i) The performance of an IRGA-classifier is better than or comparable with the Bayes classifier, which is an optimal classifier, for any number of training points, appeared in Table 2 and Table 3, for both Iris data and wine data classification.
ii) The performance of a SGA-classifier for these two benchmark problems depends on the number of training points [Table 2 and Table 3]. The larger number of training points, the better performance for SGA-classifier. (It is compatible to the theorem 1 in [3] and other experimental results in [2-6]. It shows low *robustness* of SGA-classifier [Table 4 and Table 5].
iii) On the contrary, IRGA-classifier have a good *robustness* [Table 4 and Table 5], because a low dependence on the number of training points. Thus the *large number of training* points is not necessary for good performance of an IRGA-classifier. In fact it has robustness comparable to Bayesian classifier.

Radar targets classification is done by ten hyperplanes (H=10) and for ten numbers of training points (*n*=10). In this experiment we waited until the IRGA-classifier and SGA-classifier converged to their optimum solutions for different signal to noise ratios (changing the variances of Gaussian noise produces different powers of noise). Table 6 shows obtained results.

Table 6 shows that the hyperplanes have been found by IRGA-classifier perform more accurate than a SGA-classifier and are comparable to Bayesian classifier. This means that designed intelligent fuzzy controller steers the GA- classifier to find better hyperplanes, near those have been found by Bayesian classifier for a low number of training points (*n*=10).

At another experiment, to show the effective role of fuzzy controller in the reduction of the number of iterations, 10 out of 50 measurements are considered as training data and the rest as the test data. The average scores of recognition (%) with respect to the number of generations for FCRGA-classifier and a SGA-classifier have been shown in Figures 4,5,6 for each case study. In this figures the numbers of generations are normalized by 10. In Fig. 6 the SNR is 10 dB.

These also mean that the convergence rate of an IRGA-classifier has a considerable improvement compared with a SGA-classifier.

Since the Equation 5 is more complex than the fitness definition in a SGA-classifier, the number of generations for a RGA-classifier is more than a SGA-classifier and in turn for FCRGA-classifier and FCSGA-classifier. Thus comparing the performances of a FCRGA and FCSGA classifiers we found the better reduction of number of generations than it has been shown in Figures 4-6.

**Table 2** Recognition scores (%) for Iris data classification with H=5.

| | training points=5 | | | training points=10 | | | training points=15 | | |
|---|---|---|---|---|---|---|---|---|---|
| | SGA | IRGA | Bayes | SGA | IRGA | Bayes | SGA | IRGA | Bayes |
| class1 | 89.1 | 98.1 | 91.2 | 97.7 | 97.5 | 96.4 | 100 | 100 | 100 |
| class2 | 92.2 | 93.3 | 95.4 | 87.9 | 95.1 | 97.3 | 96.3 | 96.5 | 97.1 |
| class3 | 78.8 | 95.1 | 85.1 | 96.2 | 98.2 | 98.6 | 92.1 | 98.2 | 96.1 |
| average | 86.7 | 95.5 | 90.6 | 93.9 | 96.9 | 97.43 | 96.1 | 98.2 | 97.7 |

**Table 3** Recognition scores (%) for Wine data classification with H=5.

| | training points=5 | | | training points=10 | | | training points=15 | | |
|---|---|---|---|---|---|---|---|---|---|
| | SGA | IRGA | Bayes | SGA | IRGA | Bayes | SGA | IRGA | Bayes |
| class1 | 70.1 | 88.2 | 83.1 | 83.2 | 92.1 | 95.2 | 89.1 | 95.3 | 95.4 |
| class2 | 73.2 | 90.2 | 92.1 | 87.2 | 93.5 | 90.1 | 90.1 | 92.3 | 91.5 |
| class3 | 80.2 | 89.1 | 95.2 | 87.6 | 92.3 | 92.7 | 86.3 | 97.3 | 98.1 |
| class4 | 65.7 | 78.4 | 80.9 | 73.5 | 82.3 | 84.5 | 89.2 | 91.1 | 90.6 |
| class5 | 71.0 | 89.1 | 81.5 | 87.1 | 92.3 | 95.3 | 89.1 | 97.8 | 96.0 |
| average | 72.0 | 87.0 | 86.6 | 83.7 | 90.5 | 91.6 | 88.8 | 94.8 | 94.3 |

**Table 4** The robustness for different training points (n) for Iris data with respect to n=5 as a reference.

| | n=10 | n=15 | n=20 | n=25 |
|---|---|---|---|---|
| SGA-classifier | 6.52 | 6.81 | 6.95 | 7.01 |
| IRGA-classifier | 34.60 | 24.24 | 32.05 | 29.33 |
| Bayesian classifier | 7.13 | 9.17 | 25.34 | 32.11 |

**Table 5** The robustness for different training points (n) for Wine data with respect to n=5 as a reference.

| | n=10 | n=15 | n=20 | n=25 |
|---|---|---|---|---|
| SGA-classifier | 3.58 | 3.52 | 4.76 | 5.36 |
| IRGA-classifier | 12.92 | 8.10 | 11.67 | 13.33 |
| Bayesian classifier | 9.16 | 8.16 | 12.15 | 15.38 |

**Table 6** Recognition scores (%) with respect to different SNRs with n=10 and H=10.

| | different SNRs(dB) | | | | | | |
|---|---|---|---|---|---|---|---|
| | -15 | -10 | -5 | 0 | 5 | 10 | 15 |
| SGA | 17.9 | 25.5 | 33.4 | 43.7 | 57.5 | 70.7 | 82.5 |
| IRGA | 23.5 | 33.4 | 40.2 | 65.5 | 77.2 | 93.3 | 96.3 |
| Bayes | 25.8 | 31.6 | 43.2 | 63.7 | 78.8 | 92.5 | 95.1 |

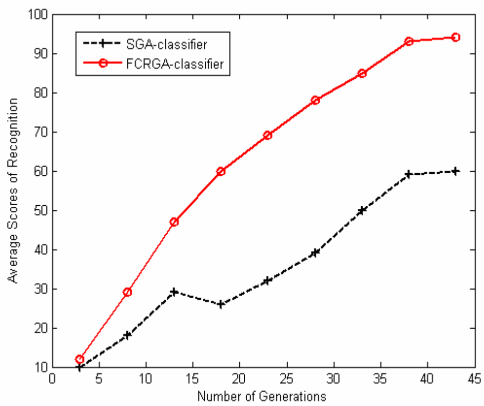


**Fig. 4** The average scores of recognition (%) with respect to the number of generations(*10) for Iris data classification ($n$=10 and H=5).

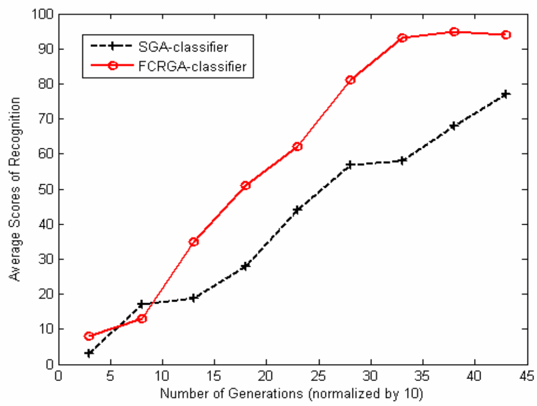


**Fig. 5** The average scores of recognition (%) with respect to the number of generations (*10) for Wine data classification ($n$=10 and H=5).

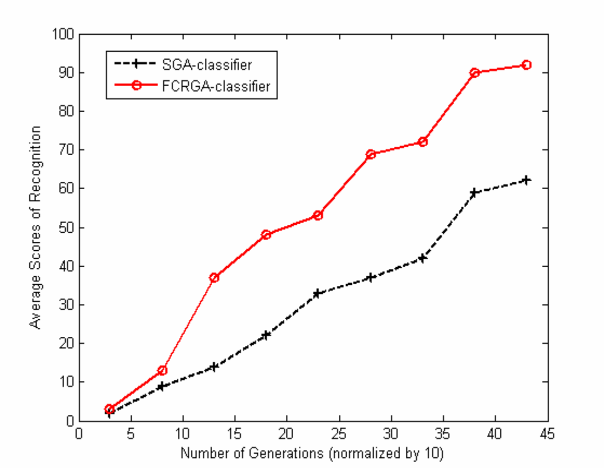


**Fig. 6** The average scores of recognition with respect to the number of generations (*10) for radar targets classification ($n$=10, H=10 and SNR=10 dB).

## 5 Conclusion

An evolutionary computation method is proposed for obtaining optimal hyperplanes in feature space, designing intelligent and robust GA-classifier (IRGA-classifier).

Conventional GA-classifiers, which have been introduced in previous researches, have two important defects. They need a *large number of training points* and a *large number of iterations* to converge to optimum hyperplanes. Both of these two prerequisites are usually unreachable in practice and can restrict the performance of conventional GA-classifiers.

In this article a new concept, named the *robustness* of a GA-classifier, has been proposed. It has been defined as the insensitivity of performance of a GA-classifier under increasing the number of training points, to remove the first weakness aspect of conventional GA-classifiers. On the other hand the idea of designing the intelligent controllers for adapting the crossover and mutation rate in a GA-classifier has been proposed to steer the GA-classifier to optimum solution and to escape it from local solutions. It can remove another weakness of usual GA-classifiers, which is the need for large number of iterations to converge to optimum hyperplanes. The IRGA-classifier find the decision hyperplanes which are fine tuned between different classes, no closer to one class.

The performance of designed IRGA-classifier, which is FCRGA-classifier compared with a simple GA-classifier and Bayesian classifier for three pattern recognition problems with low, medium and high feature space dimensions. The experimental results show a better robustness, performance and convergence rate for IRGA-classifier compared with a SGA-classifier. Also similar performances have been obtained for IRGA-classifier and Bayesian classifier for low number of data points. Both of these results are two evidences of removing two essential problems in conventional GA-classifiers using IRGA-classifier.

Other intelligent controllers (e.g. Neural Network structures) and other evolutionary classifiers (e.g. Particle Swarm classifier) should be studied to investigate their performance compared with proposed IRGA-classifier in this paper.

## Acknowledgement

This research has been supported by the Iranian Telecommunication Research Center (ITRC) through grant 500-6618.

## References

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

[2] S. K. Pal, S. Bandyopadhyay and C.A.Murthy, " Genetic Classifiers for Remotely Sensed Images: Comparison with Standard Methods", International Journal of Remote Sensing, Vol.22, No.13, pp.2545 – 2569, 2001.

[3] S. Bandyopadhyay, C. A. Murthy and S.A.Pal, " Theoretical Performance of Genetic Pattern Classifier", Journal of The Franklin Institute 336, pp. 387-422, 1999.

[4] S. Bandyopadhyay, C. A. Murthy and S.k.Pal, " VGA – Classifier : Design and Application " , IEEE Transactions On Systems , Man, and Cybernetics– Part B: Cybernetics , Vol.30 , No.6, pp. 890-895, Dec.2000.

[5] S. Bandyopadhyay and S.K.Pal, "Pixel Classification Using Variable String Genetic Algorithms With Chromosome Differentiation", IEEE Transactions on Geoscience and Remote Sensing, Vol .20, No.2, pp. 303-308, Feb.2001.

[6] S. Bandyopadhyay, S.K.Pal and B.Aruna, "Multibjective GAs, Quantitative Indecies, and Pattern Classification ", IEEE Transactions On Systems , Man, and Cybernetics-Part B: Cybernetics, Vol. 34, No. 5, Oct. 2004.

[7] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition", Data Mining and Knowledge Discovery, 2, pp. 121-167, 1998.

[8] A. E. Eiben, R. Hinterding and Z. Michalewicz, "Parameter Control in Evolutionary Algorithms", IEEE Transactions on Evolutionary Computation, Vol. 3, No. 2, pp. 124-141, July 1999.

[9] T. Bäck and M. Schütz, "Intelligent Mutation Rate Control in Canonical Genetic Algorithms", in Foundations of Intelligent Systems (Lecture Notes in Artificial Intelligence, 1079), Z.Ras and M. Michalewicz, Eds. New York: Springer-Verlag, pp. 158-167, 1996.

[10] J. Hesser and R. Männer, "Toward an Optimal Mutation Probability for Genetic Algorithms", in Proc. 1st conf. Parallel Problem Solving from Nature, (Lecture Notes in Computer Science, Vol. 496), H. Schwefel, and R. Männer, Eds. Berlin, Germany: Springer-Verlog, pp.23-32, 1991.

[11] L. M. Schmitt, "Theory of Genetic Algorithm II: Models for Genetic Operators Over the String – Tensor Representation of Populations and Convergence to Global Optima for Arbitrary Fitness Function Under Scaling", Theoret. Comput. Sci. 310, 2004.

[12] I. Wegener, "Theoretical Aspects of Evolutionary Algorithms", a technical report published by: University of Dortmund, May 2001.

[13] J. T. Tou and R.C. Gonzalez, " Pattern Recognition Principles", Addison-Wesely, Reading MA, 1974.

[14] F. Herrera, M. Lozano and J.L.Verdegay, "Tackling real-coded genetic algorithms: Operators and tools for behavioral analysis", Artificial Intell., Rev., Vol. 12, pp. 265-319, 1998.

[15] Y. Shi, R. Eberhart and Y. Chen, " Implementation Of Evolutionary Fuzzy Systems ", IEEE Transactions On Fuzzy Systems, Vol .7, No. 2, pp. 109-119, April 1999.

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

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

[18] R. A. Fisher, "The use of Multiple Measurements in Toxonomic Problems", Ann. Eugen, Vol. 7, pp. 179-188,1936.

[19] University of California, Irvine, via anonymous ftp ftp.ics.uci.edu/pub/machine-learning-databases.

[20] M. R. Bell, and R.A. Grubbs, "JEM Modeling and Measurement for Radar Target Identification", IEEE Transactions on Aerospace and Electronic Systems, Vol.29, No. 1, pp. 73-87, Jan. 1993.

[21] S. H. Zahiri, H. Zareie, and M. R. Agha-Ebrahimi, "Automatic Radar Target Recognition Using Modulation of Targets on Radar Signal", In Proc. of ICEE 2000 in Farsi, pp. 348-353, Isfahan, May 2000.