Intelligent Gravitational Search Algorithm for Optimum Design of Fuzzy Classifier

Hossein Askari and Seyed-Hamid Zahiri
Department of Electrical Engineering, Faculty of Engineering,
Birjand University, Birjand, Iran
P.O. Box: 97175-376 Phone: +98-561-2227044 Fax: +98-561-2227795
e-mail: Ho.askary@gmail.com, hzahiri@birjand.ac.ir

***Abstract*—In this paper, membership function shapes and types and the fuzzy rules of fuzzy systems are adjusted by using a Intelligent Gravitational Search Algorithm (IGSA) in order to obtain an optimal fuzzy system. The advantages of this method in classifying various data sets are illustrated. Possible extensions of this technique are briefly discussed.**

Keywords- Gravitational search algorithm; fuzzy system; classifier;

I. INTRODUCTION

The most important issue in designing a fuzzy system is to determine appropriate fuzzy variables and their membership functions, optimum fuzzy rules containing Suitabl eantecedents and consequents, and proper fuzzy operators. In fact, these are the structural fuzzy parameters of any fuzzy system which a designer tries to obtain their optimal set up. In other words, one of the most important considerations in designing any fuzzy system is the generation of the optimal fuzzy rules as well as the membership functions for each fuzzy set. In most existing applications, the fuzzy rules are generated by experts in the area, especially for control problems with only a few inputs. With an increasing number of variables, the possible number of rules for the system increases exponentially, which makes it difficult for experts to define a complete rule set for good system performance. An automated way to design fuzzy systems might be preferable. In essence, the design of a fuzzy system can be formulated as a search problem in high dimensional space where each point represents a rule set, membership functions, and the corresponding system's behavior. Given some performance criteria, the performance of the system forms a hyper-surface in the space. Developing the optimal fuzzy system design is equivalent to finding the optimal location of this hyper-surface. These characteristics seem to make the swarm intelligence optimization algorithms good candidates for searching the hyper-surface for optimum point. In this paper the employment of Gravitational Search Algorithms (GSA) for optimizing fuzzy systems' parameters is investigated. Already, the usage of heuristic methods and fuzzy logic with each other as more powerful algorithms were reported [2–6,8]. For an application in data mining and pattern recognition tasks, designing an optimum fuzzy classifier by using GSA is introduced. Also other researches in this area are reported. GA's are commonly used evolutionary algorithms that provide a way to search poorly understood irregular spaces. One of the key issues in the evolutionary design of fuzzy systems using GA's is their genotype representation; that is, what is encoded into the chromosomes. Thrift [14] and Hwang and Thompson [15] encode all the rules into the chromosome while fixing the membership functions. Using several critical points to represent each membership function while using all the possible rules, Karr and Gentry [18] use GA's to evolve these critical points; that is, to adjust the membership functions. Since in a fuzzy system the membership functions and rule set are codependent, they should be designed or evolved at the same time. Homaifar and McCormick [17] use GA's to tune the membership functions and evolve the rule set at the same time. Similar to [17], Lee and Takagi [19] also encode membership functions and all the rules into the chromosome, but have a different way to encode the triangular membership functions. Yuhui Shi [1] implements a fuzzy classifier whose rules and membership functions are optimized by genetic algorithm (namely GAF-classifier) are provided. The optimization of parameters of fuzzy systems by using the swarm intelligence algorithms has been implemented in different applications. Tao et al [9] proposed a fuzzy entropy method incorporating with the ant colony optimization (ACO). The ACO was used to obtain the optimal parameters of fuzzy entropy method. They applied their method to the segmentation of infrared objects and they illustrated that the fuzzy entropy method, incorporating with the ACO, provides improved search performance and requires significantly reduced computations in comparison to GA. Therefore, it may be suitable for real-time vision applications, such as automatic target recognition (ATR). Han and Shi [10] utilized ACO technique for fuzzy clustering in image segmentation. Chatterjee and Siarry [11] employed the PSO algorithm to simultaneously tune the shape of the fuzzy membership functions as well as the rule consequences for the entire neuro-fuzzy rule based classifier. Chen and Zhao [12] proposed a data-driven fuzzy clustering method based on maximum entropy principle (MEP) and PSO. In their algorithm, the memberships of output variables are inferred by maximum entropy principle, and the centers of fuzzy rule base are optimized by PSO. In [13], fuzzy c-mean clustering, particle swarm optimization and recursive least-squares are combined to generate fuzzy modeling system. Tao et al. [25] adopted ACO to propose a chaotic optimization method, called CAS (chaotic ant swarm) for solving the problem of designing a fuzzy system to identify dynamical systems. The position vector of each ant in the

CAS algorithm corresponds to the parameter vector of the selected fuzzy system. At each learning time step, the CAS algorithm is iterated to give the optimal parameters of fuzzy systems based on the fitness theory. Then the corresponding CAS-designed fuzzy system is built and applied to the identification of the unknown nonlinear dynamical systems. This paper is organized as follows:

Section II explains the proposed intelligent GSA. We discuss about fuzzy system in Section III and we implement an optimized fuzzy system using GSA in section IV respectively. In Section V, the experimental results are presented on different data sets to evaluate the performance of the proposed IGSF-classifier. In particular, in this Section the comparative results are provided for PSF-classifier (a fuzzy classifier based on the particle swarm optimization) [7], and GAF-classifier (a fuzzy classifier based on the genetic algorithm) [1]. Finally, Section VI concludes the paper.

II. Intelligent Gravitational Search Algorithm

In standard GSA the swarm size was considered a constant value (50 for their experiments), and the effective number of objects was set to the swarm size and was decreased to one lineally. Also the gravitational coefficient was decreased by an exponential function as Eq. (1)

$$G(t) = G_0 \exp(-\alpha \frac{\mathrm{t}}{\mathrm{T}}) \tag{1}$$

Linearly, exponentially, or other schedules for mathematically modeling the search process of a swarm intelligence algorithm may be useful for tackling some benchmark functions (as it was shown in [20]); but for solving complex engineering problems, this planning is not practical, generally. Because in complex optimization problems, like data mining, the search process of GSA is non-linear and very complicated and it is hard if not impossible, to mathematically model the search process. Thus adjusting the GSA parameters by predefined mathematical models reduces the performance of GSA and it may lead to premature convergence, local capturing, poor exploitation, poor exploration, etc.

On the other hand, some understanding of the GSA search process has been accumulated, and linguistic description of its search process is available. This understanding and linguistic description make a fuzzy system a good candidate for dynamically controlling the GSA parameters.

In this Section a fuzzy system is introduced to control the effective number of objects (*Kbest*) and gravitational coefficient to improve the efficiency and performance of GSA The proposed optimizing method is called Fuzzy-GSA and is utilized to design a fuzzy classifier in the next Section. This version of GSA is just similar to Fuzzy-GSA that has been introduced by the authors in [26]. It is adopted because our experiences on its powerfulness and effectiveness in pattern recognition tasks.

A. Linguistic Description on the Effect of GSA Parameters on Its Search Process

1) Number of Effective Objects (Kbest)

Number of effective objects (Kbest) has a significant effect on the search process of GSA. A large value of Kbest means considering more objects which interacting with each other by gravitational force. It means more movement, more computational costs, and lower convergence rate; whereas a small value of Kbest causes a local minimum capturing and reduces the performance of GSA. In fact by tracking the search process of GSA, when GSA has no effective improvement in the best fitness, Kbest should be increased to escape from the local regions in the solution space. It means confirming the exploitation. On the contrary by receiving better regions, the value of Kbest should be decreased to improve the convergence rate and fortify the exploration instead of exploitation. Obviously, in each complex engineering and practical problem, the reduction and increasing schemes of Kbest are different. Thus, the idea of intelligently controlling the Kbets by effective fuzzy rules can simulate many of these schemes without any try and error efforts for mathematically modeling the best model of changing the value of Kbest.

2) Gravitational Coefficient

The application of gravitational coefficient (G) allows control over the dynamical characteristics of the particle swarm, including its exploration versus exploitation propensities. In fact, gravitational coefficient prevents a buildup of velocity because of the effect of object inertia. Without the gravitational coefficient, objects with buildup velocities might explore the search space, but lose the ability to fine-tune a result. On the other hand, preventing the objects speed too much might damage the search space exploration. Thus the value of gravitational coefficient affects the global versus local abilities of the GSA. Also it can be concluded from Eq. (2) to (6) that G determines the value of attraction of objects by the Kbest positions found in the present iteration. This means that the convergence characteristics of GSA can be controlled by gravitational coefficient. As the fitness value of the objects system becomes better and better, the part of search space, which the objects explore should be smaller and smaller. It means that G should be decreased to emphasize the local search instead of global. A less improvement in the objects fitness causes a bigger search space for the exploration. This means an increasing should be happen on the value of G, to emphasize the global search instead of local. Since the search process is randomized based, it might be needed to increase the gravitational coefficient in medium values of iterations, and vice versa. Thus an exponential model with reduction property for all iterations (as it was proposed in [20]) is not a good schedule for solving complex problems.

B. Fuzzy Controller in Fuzzy-GSA

The fuzzy controller is constructed with four inputs and two outputs. The inputs are as follows:

- $f_{best}(t)$: The maximum fitness value among the all objects in iteration *t*.
- UN : The number of iterations, which f_{best} is unchanged.
- *VAR_fit(t)*: The variance of the obtained fitnesses in iteration *t*.

UN is introduced as an input of fuzzy controller to know when the object system converged (or captured) to a local optimum and *VAR_fit(t)* is introduced as a metric of *objects diversity*. Obviously large values of *VAR_fit(t)* show large objects diversity and vice versa.

Two outputs are:

- $Kbest$: The number of effective masses (objects).
- G : The gravitational coefficient.

The following eight fuzzy rules can be extracted from the linguistic descriptions in previous subsection, to control intelligently the search process of GSA:

1. IF UN is high, and $f_{best}(t)$ is low, THEN $Kbest$ is high and G is high.
2. IF *VAR_fit(t)* is medium and UN is low and $f_{best}(t)$ is medium, THEN $Kbest$ is low and G is medium.
3. IF $f_{best}(t)$ is medium and UN is medium, THEN $Kbest$ is medium and G is medium.
4. IF UN is high and $f_{best}(t)$ is high, THEN $Kbest$ is low and G is low.
5. IF $f_{best}(t)$ is low and *VAR_fit(t)* is low, THEN $Kbest$ is high and G is high.
6. IF $f_{best}(t)$ is medium and *VAR_fit(t)* is high, THEN $Kbest$ is high and G is medium.
7. IF $f_{best}(t)$ is high and *VAR_fit(t)* is medium, THEN $Kbest$ is low and G is medium.
8. IF $f_{best}(t)$ is high and *VAR_fit(t)* is high, THEN $Kbest$ is low and G is high.

The fuzzy controller has been designed with above fuzzy rules and its normalized inputs and outputs membership functions are shown in figure 1 and figure 2, respectively.

It must be mentioned that different kinds of inputs, outputs, membership function shapes, membership function locations and fuzzy rules may be introduced and even these parameters can be optimized by another optimization algorithm. In this here the membership functions and their locations are selected and tuned manually. The block diagram of Fuzzy-GSA is shown in figure 3.

III. Fuzzy Logic and Classification

Classification is a supervised learning technique that takes labeled data samples and generates a model (classifier) that classifies new data samples into different predefined

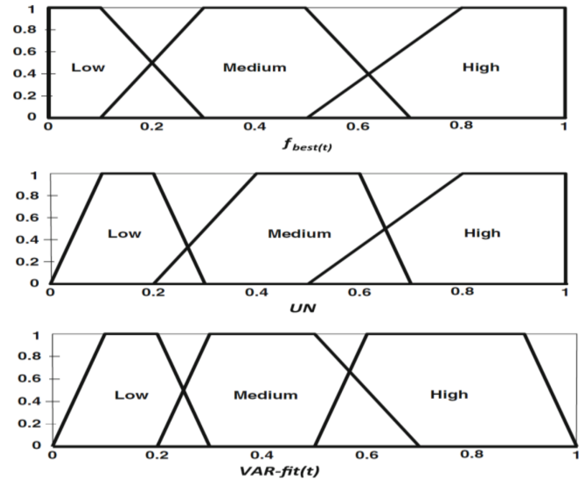

Figure 1. Normalized inputs membership functions

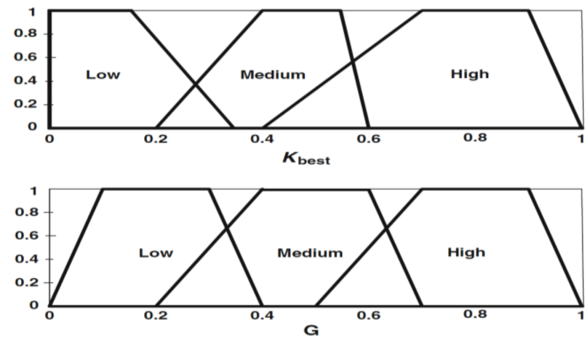

Figure 2. Normalized outputs membership functions

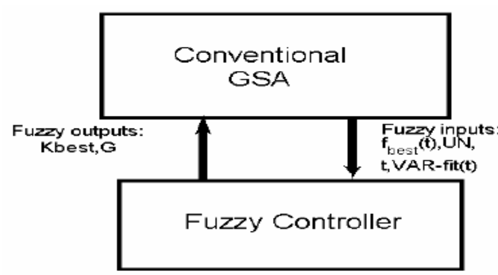

Figure 3. Block diagram of Fuzzy GSA

groups or classes. This classification problem can be easily solved by fuzzy logic with interpretable if-then rules and membership function Fuzzy logic provides a general concept for description and measurement. Most fuzzy logic systems encode human reasoning into a program to make decisions or control a system. Fuzzy logic compromises fussy sets, which are a way of representing non-statistical uncertainty and approximate reasoning, which includes the operations used to make inferences in fuzzy logic. Fuzzy rule-based systems have been successfully applied to various engineering problems (e.g. pattern recognition [1], [21], and control problems [22]). In this section the basic concepts and definitions of fuzzy systems are presented.

A. Membership Functions

Unlike traditional two-valued logic, in fuzzy logic, fuzzy set membership occurs for a fuzzy variable by degree over the range [0,1]. Which is represented by a membership function? It is this function that is the fuzzy set. The function can be linear or nonlinear. Commonly used are the left-trapezoidal, right-trapezoidal, triangle, Gaussian, and sigmoid functions, as shown in figure 4. Definitions of these membership functions as used in this chapter are as follows.

a) Left-trapezoidal membership function:

$$LTrap-MF(x)=\begin{cases}1 & if\ x<a\\ \frac{b-x}{b-a} & if\ a\le x\le b\\ 0 & if\ x>b\end{cases}$$

b) Right-trapezoidal membership function

$$RTrap-MF(x)=\begin{cases}0 & if\ x<a\\ 2\frac{x-a}{b-a} & if\ a\le x\le b\\ 2\frac{b-x}{b-a} & if\ \frac{b+a}{2}\le x\le b\\ 0 & if\ x>b\end{cases}$$

c) Triangle membership function:

$$Triangle-MF(x)=\begin{cases}0 & if\ x<a\\ 2\frac{x-a}{b-a} & if\ a\le x\le\frac{b+a}{2}\\ 2\frac{b-x}{b-a} & if\ \frac{b+a}{2}\le x\le b\\ 0 & if\ x>b\end{cases}$$

d) Gaussian membership function

$$Gaussian-MF(x)=e^{-0.5y^2}\ where\ y=\frac{8(x-a)}{b-a}-4$$

e) Sigmoid membership function

$$Sig-MF(x)=\frac{1}{1+e^{(-y+6)}}\quad where\ y=\frac{12(x-a)}{b-a}$$

f) Reverse-sigmoid membership function

$$Rsig-MF(x)=1-Sig-MF(x)$$

From the definitions, it can be seen that each abovementioned membership function is determined by two values (the start-point a and the end-point b).

B. Fuzzy Rules

The general form of a Mamdani-type fuzzy rule in a fuzzy system is

If x_1 is A_1 AND x_2 is A_2 $\ldots x_n$ is A_n THEN y_1 is C_1

y_2 *is* $C_2, \ldots, y_n$ is C_k

Where each y_i is the consequent (output) variable whose value is inferred, each x_i is an antecedent (input) variable and each A_i and C_i is a fuzzy represented by a membership function. The antecedents are combined by AND fuzzy operator. AND'ed antecedents are usually calculated by T-norm [23]. Other fuzzy operators are defined (e.g. OR, Aggregation operator, and Implication operator). In our

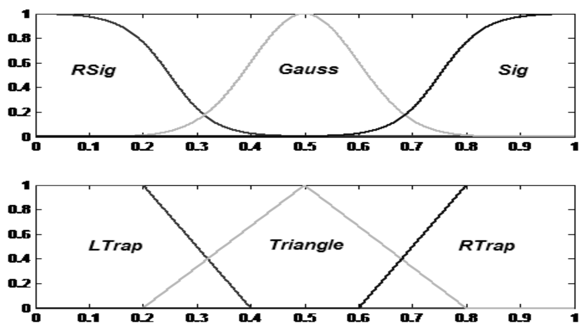

Figure 4. Left-trapezoidal, right-trapezoidal, triangle, Gaussian, and sigmoid membership functions

application, a fuzzy system is utilized as a fuzzy classifier. In fuzzy classifiers the most utilized operator for feature vectors are AND operator. All the fuzzy rules in a fuzzy system are fired in parallel. The fuzzy system works as follows:

1. Determine the fuzzy membership values activated by the inputs.
2. Determine which rules are fired in the rule set.
3. Combine the membership values for each activated rule using the AND operator.
4. Trace rule activation membership values back through the appropriate output fuzzy membership functions.
5. Utilize defuzzification to determine the value for each output variable.
6. Make decision according to the output values.

Determination of the fuzzy membership values of the inputs is often called fuzzification. Each input may activate one or more fuzzy sets of that input variable according to the definitions of the fuzzy membership functions. Only the rules with at least one antecedent set activated are said to be fired by the inputs. The AND operator is typically used to combine the membership values for each fired rule to generate the membership values for the fuzzy sets of output variables in the consequent part of the rule. Since there may be several rules fired in the rule sets, for some fuzzy sets of the output variables there may be different membership values obtained from different fired rules. There are many ways to combine these values. One commonly used way is to use the OR operator, that is to take the maximum value as the membership value of the fuzzy set. Next, a defuzzification method is used to produce a single scalar value for each output variable. A common way to do the defuzzification is called centroid method [23]. Then according to the output values, some decisions can be made to solve the problem. For example, for M-class classification problem, the range of the output variable of a fuzzy classifier can be divided into M evenly distributed parts, then the input pattern belongs to class I if the inferred output value is located inside the ith part. This is the approach taken for constructing the fuzzy classifiers in this here.

IV. DESIGNING A FUZZY CLASSIFIER USING INTELLIGENCE GRAVITATIONAL SEARCH ALGORITHM

Let us assume that our pattern classification problem is an M-class problem in the n- dimensional feature space with continuous attributes. The general form of a fuzzy classifier rule is as follows:

If x_1 is A_1 AND x_2 is $A_2 \ldots x_n$ is A_n THEN y is C

where each y is the output of the rule C is a feature vector (input) each A_i and C is a fuzzy represented by a membership function. As mentioned in the previous section, the range of the output variable of the rule is divided into M evenly distributed parts, then the input pattern belongs to class i if the inferred output value is located inside the ith part. For example, for three classification problem, the output range is divided into three fuzzy regions of Low, Medium, and High, corresponding to class 1 to 3 respectively. The major aim in this section is obtaining the optimum fuzzy rule set and membership functions in a fuzzy classifier, using GSA. The designed fuzzy classifier by employing GSA is called GSF-classifier.

A. Masses Representation

The first important consideration is mass representation strategy, which is how to encode the fuzzy classifier into the mass form. To completely represent a fuzzy classifier (system), each mass must contain all the needed information about the rule set and the membership functions. For more explanation, suppose a classification problem for four feature vector dimensions and three reference classes. Each variable has three fuzzy sets representing the linguistic descriptions: Low, Medium, and High. In this case, we can use the integers 1-3 to represent each of these three terms, use the integer 0 to represent the absence of a term, and use a minus sign '-' to encode the term "not". For example, the rule "IF input-1 is not Low AND input-2 is not Medium AND input-4 is High, THEN output is high" can be encoded as "-1-2033". The total of six types of membership functions (defined in the subsection 4-1) are used as the membership functions candidates; each is represented by an integer from 1 to 6. A membership function in our problems is completely determined by three values: the start–point a, the end-point b, and the function type value. Here, real values are chosen to represent the start-point and end-point. Assume for the variable x that its dynamic range is [A,B] and that it has n fuzzy sets. If the fuzzy membership functions are uniformly distributed over the range with half-way overlap [1], then the center point C_i $(i = 1,..., \mathrm{n})$ of the ith membership function is located at

$$C_i = S + i * step \qquad i = 1,..., \mathrm{n}$$

where

$$step = \frac{A - B}{n + 1}$$

We constraint the start–point a_i of the ith membership function to vary only between c_{i-1} and c_i, and the end-point b_i of the ith membership function can vary only between c_i and c_{i+1}. Assume for our example fuzzy classifier that the number of rules is 5, then the length of the mass is 4*(3*(2+1)) +5*5==61 and its form is as follows:

$$p = (s_1, s_2, s_3, \ldots, s_{14}, s_{15}, s_{16}, s_{17}, \ldots, s_{56}, s_{57}, s_{58}, s_{59}, s_{60}, s_{61})$$

Where s_1, s_2 represent the start point and end point for the first fuzzy set of the first input variable. s_2 represents the membership function type for the first input variable and can vary between one and six. s_4 to s_{36} encode the remaining fuzzy membership functions (start point, and point, and type, s_{37} to s_{41} represent the first fuzzy rule and s_{57} to s_{61} represent the last fuzzy rule. Each possible rule is checked to see whether it represents a feasible rule or not. A rule without a nonzero antecedent or consequent part is not a feasible rule and will not be included in the rule set. This schedule is similar to chromosome representation, where Shi et al. [1] try to implement an evolutionary fuzzy classifier using genetic algorithm. Since we are interested in comparing the powerfulness of the evolutionary algorithms and swarm intelligence in task of optimizing fuzzy systems, the schedule of implementation of a fuzzy classifier using GA and PSO adopted from [1], [7] respectively to reach more meaningful comparison results.

B. Fitness Function Definition

To evaluate the quality of each rule set, at first, a fitness value is defined for a rule as below:

$$Q = \frac{TP}{TP + FN} \cdot \frac{TN}{FP + TN}$$

Where

TP: True Positives = number of instances covered by the rule that are correctly classified, i.e., its class matches the training target class.

FP: False Positives = number of instances covered by the rule that are wrongly classified, i.e., its class differs from the training target class.

TN: True Negatives = number of instances not covered by the rule, whose class differs from the training target class.

FN: False Negatives = number of instances not covered by the rule, whose class matches the training target class.

Then the total fitness of a rule set is defined as follows:

$$Fit(\mathrm{Rule_{set}}) = \sum_{l=1}^{k} Q_l$$

Where Q_l is the fitness of the l'th rule of the K rules in the rule set.

V. EXPERIMENTAL RESULTS

The performance evaluation of the optimized fuzzy classifier (IGSF-classifier) is investigated in this Section. Also the comparative results with a fuzzy classifier whose rules and membership functions are optimized by GA [1] (namely GAF-classifier) and fuzzy classifier which designed by PSO PSF-Classifier are provided. recognition score for training data and testing data is performance aspect considered for comparing above mentioned optimized fuzzy classifiers. Three pattern classification problems [1] with different feature vector dimensions (4, 9, 34), are used for performance evaluation and comparison of the results. A description of the data sets is given as follows:

Iris data: The Iris data contains 50 measurements of four features of each three species: Iris setosa, Iris versicolor, and Iris virginica. Features are sepal length, sepal width, petal length and petal width.

Cancer data: This breast cancer database, obtained from the University of Wisconsin Hospital, Madison, has 683 breast mass samples belonging to two classes Benign and Malignant, in a nine dimensional feature space.

Dermatology data: The aim for this dataset is to determine the type of Eryhemato-Squamous Disease. This database contains 34 attributes, 33 of which are linear valued and one of them is nominal.

To estimate more accurate performance measures, ten-fold cross validation is used. It means 10% of whole training samples are randomly considered as testing points (validation sets) and others as training set for discovery and optimization of fuzzy rules and membership functions. The validation sets is used to estimate the generalization of classifier. The whole training set is randomly divided into 10 disjoint sets of equal size. Then the PSO, GA and IGSA method is run 10 times for designing PSF-classifier and GAF-classifier and IGSF-classifier respectively. Each time with a different set held out as a validation. The estimated predictive accuracy values are the mean values of these 10 scores of recognition for training data sets and testing data sets. The population size is 40 and the maximum number of iterations is set to 1000 for PSO, GA and IGSA. The mutation and crossover rates are chosen equal to 0.01 and 0.7

A. *Performance evaluation in the classification of iris data set*

Table 1 presents the obtained predictive accuracy results by PSF-classifier, GAF-classifier and IGSF-classifier for training Iris data set. Also Table 2 presents the obtained predictive accuracy results by PSF-classifier, GAFclassifier and IGSF-classifier for testing Iris data set. Table 1 shows that the proposed IGSF-classifier outperforms other methods by obtained minimum, maximum, and average recognition scores for Iris data in training stage. Table 2 shows that the Minimum, maximum, and average recognition scores (%) for

TABLE I. MINIMUM, MAXIMUM, AND AVERAGE RECOGNITION SCORES (%) FOR **IRIS DATA TRAINING POINTS**, OBTAINED BY PSF-CLASSIFIER, GAF-CLASSIFIER AND IGSF-CLASSIFIER

Iris	*Classification rate*	*IGSF-classifier*	*PSF-classifier*	*PSF-classifier*
	Min	**96.00**	94.7	92.6
	Max	**99.3**	98.6	97.34
	Average	**98.1**	96.1	93.5

[1] These data sets is available at: http;//www.ics.uci.edu/~mlearn/MLRepository.html

TABLE II. MINIMUM, MAXIMUM, AND AVERAGE RECOGNITION SCORES (%) FOR **IRIS DATA TESTING POINTS**, OBTAINED BY PSF-CLASSIFIER, GAF-CLASSIFIER AND IGSF-CLASSIFIER

Iris	*Classification rate*	*IGSF-classifier*	*PSF-classifier*	*PSF-classifier*
	Min	92.00	92.00	91.34
	Max	**98.00**	97.34	96.00
	Average	**96.1**	95.07	93.2

B. *Performance evaluation in the classification of Cancer data set*

TABLE III. MINIMUM, MAXIMUM, AND AVERAGE RECOGNITION SCORES (%) FOR **CANCER DATA TRAINING POINTS**, OBTAINED BY PSF-CLASSIFIER, GAF-CLASSIFIER AND IGSF-CLASSIFIER

Cancer	*Classification rate*	*IGSF-classifier*	*PSF-classifier*	*PSF-classifier*
	Min	93.6	94.57	95.2
	Max	97.2	96.86	98.00
	Average	95.4	95.3	96.5

TABLE IV. MINIMUM, MAXIMUM, AND AVERAGE RECOGNITION SCORES (%) FOR **CANCER DATA DATA TESTING POINTS**, OBTAINED BY PSF-CLASSIFIER, GAF-CLASSIFIER AND IGSF-CLASSIFIER

Cancer	*Classification rate*	*IGSF-classifier*	*PSF-classifier*	*PSF-classifier*
	Min	92.85	93.40	93.70
	Max	**96.8**	95.70	96.56
	Average	94.3	94.2	95.1

Iris data training points, obtained by PSF-classifier, GAF-classifier and IGSF-Classifier obtained average recognition scores by the proposed method are better than PSF-classifier and GAF-classifier. Table 3 and Table 4 presents the obtained predictive accuracy results by PSF-classifier, GAF-classifier and IGSF-classifier for training Cancer data set and testing Cancer data set respectively. for Cancer data set, the obtained average recognition score by GAF-classifier is better than other methods. But the difference of the performance of GAF-classifier and IGSF-classifier is only 1.1%. It demonstrates that the performance of the proposed technique is comparable to GAF-classifier for Cancer data. only, for Cancer data, GAF-classifier outperforms the proposed IGSF-classifier in average recognition score by little value of 0.8%. For this data set, IGSF-classifier has the best performance with respect to maximum recognition score.

C. *Performance evaluation in the classification of Dermatology data set*

Table 5 and Table 6 presents the obtained predictive accuracy results by PSF-classifier, GAF-classifier and IGSF-classifier for training Dermatology data set and testing Cancer data set respectively. Regarding dermatology data

TABLE V. MINIMUM, MAXIMUM, AND AVERAGE RECOGNITION SCORES (%) **FOR DERMATOLOGY DATA TRAINING POINTS**, OBTAINED BY PSF-CLASSIFIER, GAF-CLASSIFIER AND IGSF-CLASSIFIER

Dermatology	*Classification rate*	*IGSF-classifier*	*PSF-classifier*	*PSF-classifier*
	Min	91.0	91.8	89.7
	Max	**96.3**	95.5	95
	Average	**94.1**	93.9	93.0

TABLE VI. ON SCORES (%) FOR **DERMATOLOGY DATA TESTING POINTS**, OBTAINED BY PSF-CLASSIFIER, GAF-CLASSIFIER AND IGSF-CLASSIFIER

Dermatology	*Classification rate*	*IGSF-classifier*	*PSF-classifier*	*PSF-classifier*
	Min	90.0	90.2	89.1
	Max	**95.4**	95.0	93.2
	Average	**93.5**	93.3	91.2

set, IGSF-classifier outperforms other methods for both maximum and average obtained scores of recognition. It should be mentioned that the improvements of the performance are not considerable and the performances of three methods are comparable to each other. These results illustrate the capability of the IGSF classifier in estimation of optimum parameters of a fuzzy classifier.

I. CONCLUSION

The bottleneck of the fuzzy logic based system for any application is the development of rule base and the formation of the membership function. This paper has proposed an approach based on Gravitational Search Algorithm that is adapted with fuzzy controller for the optimal design of the fuzzy classifier system. In the proposed approach, both rule base and the membership functions are evolved simultaneously with the objective of maximizing the correctly classified class.

ACKNOWLEDGMENT

This research has been supported by ITRC.

REFERENCES

[1] Y. Shi, R. Eberhart and Y. Chen, “Implementation of evolutionary fuzzy systems,” IEEE Trans on Fuzzy Systems , vol. 2, pp. 109-119, Apr 1999.

[2] Storn R, Price K, “Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces,” J Glob Optim vol. 11, pp. 341–359, 1997.

[3] Wang X-Z, He Y-L, Dong L-CZhao H-Y, “Particle swarm optimization for determining fuzzy measures from data,” Inform Sci, vol. 181, pp. 4230–4252, 2011.

[4] Wang X-Z and Dong C-R, “Improving generalization of fuzzy if-then rules by maximizing fuzzy entropy,” IEEE Trans on Fuzzy Systems vol. 17, pp. 556–567, 2009.

[5] Lin C-M, Li M-C, Ting A-B, Lin M-H, “A robust self learning PID control system designs for nonlinear systems using a particle swarm optimization algorithm,” Int J Mach Learn Cyber, vol. 2, pp. 225-234, 2011.

[6] G. S. MahapatraT, T. K. Mandal, G. P. Samanta, “A production inventory model with fuzzy coefficients using parametric geometric programming approach,” Int J Mach Learn Cyber, vol. 2, pp. 99-105,2011.

[7] S.H Zahiri,“Swarm intelligence and fuzzy systems,” Nova publishers. 2010.

[8] V. L. Huang, P. N. Suganthan, J. J. Liang ,“Comprehensive learning particle Swarm optimizer for solving multiobjective optimization problems,” Int J Int Syst, vol. 21, pp. 209–226,2011.

[9] W. Tao, H. Jin and L. Liu, “Object segmentation using ant colony optimization algorithm and fuzzy entropy,” Pattern Recognition Letters, vol. 28, pp. 788–796, 2007.

[10] Y. Han and P. Shi, “An improved ant colony algorithm for fuzzy clustering in image segmentation,” Neurocomputing, vol. 70, pp. 665–671, 2007.

[11] A. Chatterjee and P. Siarry, “A PSO-aided neuro-fuzzy classifier employing linguistic hedge concepts,” Expert Systems with Applications, vol. 33, pp. 1097–1109, 2007.

[12] D. Chen and C. Zhao, “Data-driven fuzzy clustering based on maximum entropy principle and PSO,” Expert Systems with Applications, vol. 36 , pp. 625-633, 2007.

[13] H. M Feng , “ Hybrid stages particle swarm optimization learning fuzzy modelling systems design,” Tamkang Journal of Science and Engineering, vol. 9 , pp. 167–176, 2006.

[14] P. Thrift, “ Fuzzy logic synthesis with genetic algorithms, In Proc 4th Int Conf Genetic Algorithms (ICGA) ,” pp. 509–513, 1991.

[15] W. R. Hwang and W. E. Thompson, “Design of intelligent fuzzy logic Controllers using genetic algorithms,” In Proc. IEEE Int. Conf. Fuzzy Systems, vol. 2 , pp. 1383-1388, 1994.

[16] R. R. Yager and L. A. Zadeh, “Adaptive control with fuzzy logic and genetic algorithms,” In Fuzzy Sets Neural Networks and Soft Computing, vol. 37, 1993, Doi: 10.1145/175247.175255.

[17] A. Homaifar and E. McCormick, “Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms,” IEEE Trans on Fuzzy Systems , vol. 3,: pp. 129-139, 1995.

[18] C. L. Karr and E. J. Gentry, “ Fuzzy control of pH using genetic algorithms,” IEEE Trans on Fuzzy Systems, vol. 1,: pp. 46–53, 1993.

[19] M. A. Lee and H. Takagi, “Dynamic control of genetic algorithms using Fuzzy logic techniques,” In Proc. Int. Conf. Genetic Algorithms pp. 76–83,1993.

[20] E. Rashedi, H. Nezamabadi-pour and S. Saryazdi, “GSA: A Gravitational Search Algorithm,” Inform Sci, vol. 179, pp. 2232-2248. 2009.

[21] H. Ishubichi, T. Nakashima and T. Murata, “Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems,” IEEE Trans on Fuzzy Systems man and Cybernetics-Part b: Cybernetics , vol. 29, 601-618, 1999.

[22] M. Sugeno, “An introductory survey of fuzzy control,” Inform Sci, vol. 36, pp. 59-83,1985.

[23] J. S. R. Jang, C. T. Sun and E. Mizutani,“ Neuro-Fuzzy and Soft Computing,” prentice hall, 1997.

[24] R. S. Parpinelli, H. S. Lopes and A. A. Freitas, “Data mining with an ant colony optimization algorithm,” IEEE Transactions on evolutionary computing, vol. 6, pp. 321-332, 2002.

[25] L. Li, Y. Yang, and H. Peng, “Fuzzy system identification via chaotic ant swarm Chaos,” Solitons & Fractals, vol. 41, pp. 401-409, 2008.

[26] H. Askari, S.-H. Zahiri, “Decision Function Estimation Using Gravitational Search Algorithm,” International journal of machine learning and cybernetics, 2011, DOI 10.1007/s13042-011-0052.