FUZZY GRAVITATIONAL SEARCH ALGORITHM AN APPROACH FOR DATA MINING

S. H. ZAHIRI

Abstract. The concept of intelligently controlling the search process of gravitational search algorithm (GSA) is introduced to develop a novel data mining technique. The proposed method is called fuzzy GSA miner (FGSA-miner). At first a fuzzy controller is designed for adaptively controlling the gravitational coefficient and the number of effective objects, as two important parameters which play major roles on search process of GSA. Then the improved GSA (namely Fuzzy-GSA) is employed to construct a novel data mining algorithm for classification rule discovery from reference data sets. Extensive experimental results on different benchmarks and a practical pattern recognition problem with nonlinear, overlapping class boundaries and different feature space dimensions are provided to show the powerfulness of the proposed method. The comparative results illustrate that performance of the proposed FGSA-miner considerably outperforms the standard GSA. Also it is shown that the performance of the FGSA-miner is comparable to, sometimes better than those of the CN2 (a traditional data mining method) and similar approach which have been designed based on other swarm intelligence algorithms (ant colony optimization and particle swarm optimization) and evolutionary algorithm (genetic algorithm).

1. Introduction

Recently a new swarm intelligence algorithm has been introduced based on the gravity forces between the masses. This method is called Gravitational Search Algorithm (GSA). The GSA could be considered as an isolated system of masses. It is like a small artificial world of masses obeying the Newtonian laws of gravitation and motion. The position of the mass corresponds to a solution of the problem, and its gravitational and inertial masses are determined using a fitness function.

Thus, agents are considered as objects and their performance is measured by their masses. All these objects attract each other by the gravity force, and this force causes a movement of all objects towards the objects with heavier masses. Hence, masses cooperate using a direct form of communication, through gravitational force. It has been shown that the GSA is able to find the optimum solution for many benchmarks [10].

Received: January 2010; Revised: July 2010 and December 2010; Accepted: January 2011

Key words and phrases: Gravitational search algorithm, Fuzzy controller, Data mining, Rule based classifier.

GSA has some important parameters (e.g. swarm size and gravitational coefficient) which play major roles in its search characteristics, including premature convergence, convergence rate, local capturing, exploitation, exploration, etc.

We know that the value of swarm size considerably affect the powerfulness and effectiveness of GSA. A large value of swarm size reduces the convergence rate and increases the powerfulness of algorithm; whereas a small value of swarm size causes a local minimum capturing and reduces the performance of GSA. On the other hand the search process of GSA (like other swarm intelligence techniques) is very complicated. Thus, it is hard if it is not possible, to model mathematically the search process of GSA. Thus, linearly decreasing the effective objects and an exponential schedule for decreasing the gravitational coefficient is not an optimal selection for tackling different complex engineering problems.

On the other hand, there are some linguistic descriptions and understandings of the search process of GSA. These understandings and linguistic descriptions make a fuzzy system a good candidate for controlling intelligently the parameters of GSA. This idea leads to designing more powerful and efficient gravitational search algorithm (called *Fuzzy-GSA*).

Another topic which is related to this paper is data mining. Data mining (DM) is defined as the process of model abstraction from the data sets and searching for valid, nontrivial patterns, and symptoms within the abstracted model [11].

DM covers a wide range of knowledge discovery methods from the databases including classification, clustering, dependence modeling, etc.

An important branch of DM includes rule discovery techniques for data classification. In this part, a rule based classifier is designed for assigning each pattern (object, record, or instance) in the feature space to one distinct class.

Many approaches, methods and goals have been tried out for designing rule based classifiers. Biology inspired algorithms such as Genetic Algorithms (GA) and swarm intelligence based approaches like particle swarm optimization (PSO) and ant colony optimization (ACO) have been successfully used([13],[9],[3], and [5]).

In this paper a rule discovery method is proposed based on the constructed *Fuzzy-GSA* and is named fuzzy GSA miner (FGSA-miner).
Discovered knowledge is expressed in the form of IF-THEN rules as follows:

$$\text{IF} <term1 \quad AND \quad term2 \quad AND ...> \text{THEN} <\text{Class}>$$

The first part (IF part) is known as rule antecedent and contains a set of conditions, usually connected by logical conjunction operator (AND). As shown above, each condition is usually referred to a term. The second part (THEN part) is consequent rule and specifies the class predicted for cases which their predictor attributes satisfy all the terms specified in the rule antecedent.

This paper is organized as follows:
Section II, consists of a brief overview of GSA. Section III and IV explain the proposed fuzzy GSA and fuzzy GSA miner, respectively. In Section V, the experimental results are presented on six data sets, to evaluate the performance of the proposed FGSA-miner. In particular, in this Section the comparative results are provided for LA-miner, CN2 [1] (a traditional data mining method), Ant-miner (a

rule discovery approach based on the ACO) [9], PSO-miner [13] (a rule discovery method based on the PSO) and GA-miner (an approach based on the GA) [13]. Finally, Section VI concludes the paper.

2. **Gravitational Search Algorithm (GSA)**

The major concept in GSA is interaction between different objects with different masses in the solution space [10]. In fact, in this optimization algorithm, agents are considered as objects and their performance is measured by their masses. All these objects attract each other by the gravity force, and this force causes a movement of all objects towards the objects with heavier masses. Hence, masses cooperate using a direct form of communication, through gravitational force. The heavy masses – which correspond to good solutions – move more slowly than lighter ones. In GSA, each mass (agent) has four specifications: position, inertial mass, active gravitational mass, and passive gravitational mass. The position of the mass corresponds to a solution of the problem, and its gravitational and inertial masses are determined by using a fitness function.

In other words, each mass presents a solution, and the algorithm is navigated by adjusting the gravitational and inertia masses. By lapse of time, we expect that masses be attracted by the heaviest mass. This mass will present an optimum solution in the search space.

More precisely, masses obey the following laws:

Gravity law: each particle attracts any other particle and the gravitational force between two particles is directly proportional to the product of their masses and inversely proportional to R the distance between them.

Motion law: the current velocity of any mass is equal to the sum of the fraction of its previous velocity and the variation in the velocity. Variation in the velocity or acceleration of any mass is equal to the force acted on the system divided by mass of inertia.

For a system with N agents (masses), the position of an agent is defined by:

$$X = (x_1, x_2, \cdots, x_d, x_n) \tag{1}$$

where x_d is the position of the dth dimension of n dimensional solution space.

At a specific time'‘t’, we define the force acting on mass ‘i’ from mass ‘j’ as following:

$$F_d^{ij}(t) = G(t).\frac{M_p^i(t) * M_a^j}{R^{ij}(t) + \varepsilon}.(x_d^j(t) - x_d^i(t)) \tag{2}$$

where M_a^j is the active gravitational mass related to agent j, M_p^i is the passive gravitational mass related to agent i, G(t) is gravitational coefficient at time t, ε is a small constant, and$R^{ij}(t)$ is the Euclidian distance between two agents i and j.

The total force that acts on agent i of dimension d is randomly weighted sum of dth components of the forces exerted from other agents:

$$F_d^i(t) = \sum_{j=1, j\neq i}^{N} rand^j * F_d^{ij}(t) \tag{3}$$

where $rannd^j$ is a random number belongs to [0, 1].
Hence, by the law of motion, the acceleration of the agent i at time t, and in direction d, $a_d^i(t)$, is given as follows:

$$a_d^i(t) = \frac{F_d^i(t)}{M_i^i(t)} \tag{4}$$

where $M^{ii}(t)$ is the inertial mass of ith agent.
Furthermore, the next velocity of an agent is considered as a fraction of its current velocity added to its acceleration.
Therefore, its position and its velocity could be calculated as follows:

$$v_d^i(t+1) = rand^i * v_d^i(t) + a_d^i(t) \tag{5}$$

$$x_d^i(t+1) = x_d^i(t) + v_d^i(t+1) \tag{6}$$

where $rand^i$ is a uniform random variable in the interval [0, 1].
The gravitational coefficient, G, should be initialized at the beginning and will be reduced with time to control the search accuracy. In [10] it was proposed that the gravitational coefficient was decreased by an exponential function as below:

$$G(t) = G_0 \exp(-\alpha \frac{t}{T}) \tag{7}$$

where G_0 and α were selected equal to 100 and 20 respectively. T is the total number of iterations and t is the iteration counter.

Gravitational and inertia masses are calculated by the fitness evaluation. A heavier mass means a more efficient agent. This means that better agents have higher attractions and walk more slowly. Assuming the equality of the gravitational and inertia masses, the values of masses are calculated using the map of fitness. The gravitational and inertial masses are updated by the following equations:

$$M_a^i = M_p^i = M_i^i = M^i \qquad i = 1, \ldots, N$$

$$m^i(t) = \frac{fit^i(t) - worst(t)}{best(t) - worst(t)} \tag{8}$$

$$M^i(t) = \frac{m^i(t)}{\sum_{j=1}^N m^j(t)} \tag{9}$$

where $fit^i(t)$ represents the fitness value of the agent i at time t, and, *worst(t)* and *best(t)* are defined as follows (for a minimization problem):

$$best(t) = \min_{j \in \{1,\ldots,N\}} fit^j(t) \tag{10}$$

$$worst(t) = \max_{j \in \{1,..,N\}} fit^j(t) \tag{11}$$

In [10] it was proposed that for performing a good compromise between exploration and exploitation, reduce the number of agents with lapse of time. In other words, only a set of agents with bigger mass apply their forces to the other. In this manner they proposed only the Kbest agents attract the others. Kbest is a function of time, with the initial value of K_0 at the beginning and decreasing with time. In such a way, at the beginning, all agents apply the force, and as time passes, Kbest is decreased linearly and at the end there will be just one agent applying force to the others. Therefore, equation 3 could be modified as:

$$F_d^i(t) = \sum_{j \in Kbest} rand^j * F_d^{ij}(t) \tag{12}$$

where Kbest is the set of first K agents with the best fitness value and biggest mass. The different steps of the proposed algorithm are the followings:

(a) Identification the search space.
(b) Initialization the swarm and masses.
(c) Fitness evaluation of objects.
(d) Update G(t), best(t), worst(t) and $M^i(t)$ for i = 1,2,. . .,N.
(e) Calculation of the total force in different directions.
(f) Calculation of acceleration and velocity.
(g) Updating objects' position.
(h) Repeat steps c to g until the stop criteria is reached.
(i) End.

3. **Fuzzy-GSA**

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 equation (2)

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 [10]); 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 model mathematically 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 controlling dynamically the GSA parameters. In fact, fuzzy systems are the best tools for controlling difficult process in engineering problems ([6], [7], and [8]).

It should be mentioned that the idea of controlling the search process of the bio-inspired algorithms by fuzzy expert systems, have been successfully implemented for some evolutionary and swarm intelligence techniques ([2], [12], [14], [16], and [17]).

In this paper 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 novel data miner in the next Section. To extract some effective fuzzy rules, at first, a linguistic description about the effects of the GSA parameters on its search process is presented as a subsection.

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

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 controlling intelligently the Kbets by effective fuzzy rules can simulate many of these schemes without any try and error efforts for modeling mathematically the best model of changing the value of Kbest.

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 gravitational coefficient, objects with buildup velocities might explore the search space, but loose 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 equations (2) to (2) that G determines the value of attraction of objects by the Kbest positions found in the present iteration. This means that the convergence characteristic*s* 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 vise versa. Thus an exponential model with reduction property for all iterations (as it was proposed in [10]) is not a good schedule for solving complex problems.

3.2. **Fuzzy Controller in Fuzzy-GSA.** The fuzzy controller is constructed with three inputs and two outputs. The inputs are as follows:

(1) $f_{best}(t)$: The maximum fitness value among the all objects in iteration t.
(2) UN: The number of iterations, which f_{best} is unchanged.
(3) *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:

(1) *Kbest*: The number of effective masses (objects).
(2) 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 and G is high.

Each of these rules has a specific duty to improve the effectiveness and powerfulness of GSA. For example consider the rule 5:

This rule is considered to prevent premature convergence or local capturing. Suppose all the masses have been located in an small area (*VAR_fit(t) is low*) and ineffective of solution space ($f_{best}(t)$ *is low*). Obviously, the fuzzy controller should steer the masses into larger and effective hypervolumes of the solution space. This is done by the consequence of this rule, where the Kbest and G give high values.

The fuzzy controller has been designed with the 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, membership function shapes, membership function locations and fuzzy rules may be introduced and even these

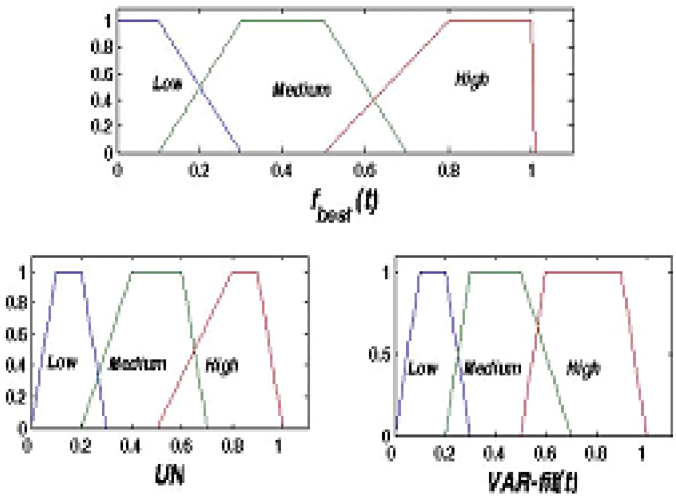

FIGURE 1. Normalized Inputs Membership Functions

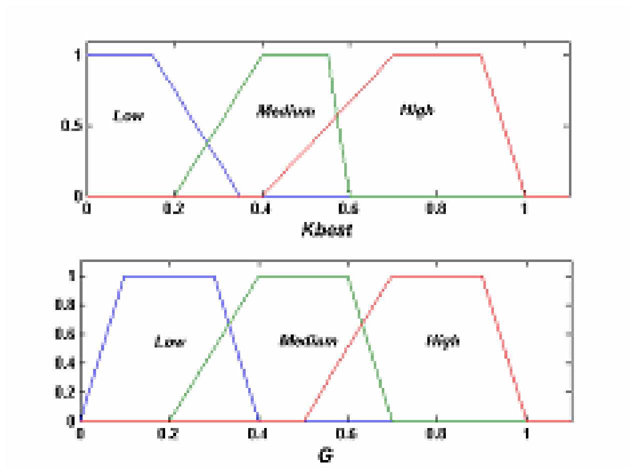

FIGURE 2. Normalized Outputs Membership Functions

parameters can be optimized by another optimization algorithm. In this paper the fuzzy inputs, membership functions, and their locations are selected and tuned manually. The block diagram of Fuzzy-GSA is shown in Figure 3.

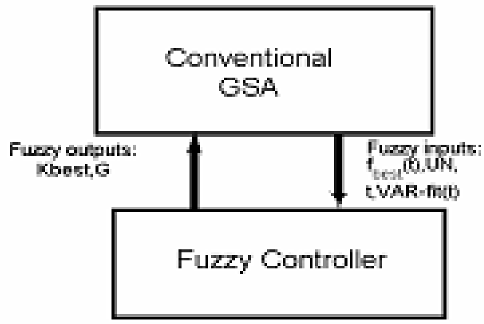

FIGURE 3. Block Diagram of Fuzzy GSA

4. **The Proposed Fuzzy GSA-Miner**

A. Object and Fitness Representation

As mentioned in section I, our proposed fuzzy GSA miner (FGSA-miner) is designed based on the Fuzzy-GSA as a function optimization algorithm which was explained in the previous Section.

To explain how FGSA-miner extracts the IF-THEN rules, consider the structure of a rule:

$$\begin{array}{l} IF\ (att_1\ is\ quant_1)\ AND\ (att_2\ is\ quant_2)\ AND \\ ...\ (att_n\ is\ quant_n)\ THEN\ x\ belongs\ to\ Class\ j. \end{array} \tag{13}$$

where att_i $(i = 1, 2, ..., n)$ is i'th attribute of the training point and $x = (att_1, att_2, ..., att_n)$ is the training point which has been expressed in the feature space. The $quant_i$ $(i = 1, 2, ..., n)$ are values belonging to the domain of corresponding attributes.

It should be mentioned that the proposed LA-miner discovers rules referring only to categorical attributes (like Ant-miner [9]). Therefore, continuous attributes have to be discretized in a preprocessing step. Since in the experimental results it is needed to compare the performance of the FGSA-miner with similar swarm intelligence based methods, the adopted dicretization method is C4.5-Disc [4] which has been utilized [9].

C4.5-Disc uses the well-known C4.5 algorithm for discretizing continuous attributes. Basically, for each attribute to be discretized it is extracted from the training set, a reduced data set containing only two attributes: the attribute to be discretized and the goal (class) attribute. C4.5 is then applied to this reduced data set. Therefore, C4.5 constructs a decision tree which all internal nodes refer to the attribute being discretized. Each path in the constructed decision tree corresponds to the definition of a categorical interval produced by C4.5.

The basic duty of FGSA-miner is to extract K distinct rules for classifying M reference classes in the n-dimensional feature space. In fact, for each rule there are $(n+1)$ unknown parameters ($n$values of $quant_i$and$j$, the index of a reference class).

Obviously, for K rules, FGSA-miner should extract $K \times (n+1)$ unknown parameters from the solution space. Thus, each object in FGSA-miner includes unknown sets of antecedents and consequences, and is of the form:

$$\mathrm{O^i} = \{\ \mathrm{quant_{11}}, \mathrm{quant_{12}}, ..., \mathrm{quant_{1n}}, j_1,\ \mathrm{quant_{21}}, \mathrm{quant_{22}}, ..., \mathrm{quant_{2n}}, j_2, ..., \\ \mathrm{quant_{k1}}, \mathrm{quant_{k2}}, ..., \mathrm{quant_{kn}}, j_k\} \quad (14)$$

where O^idenotes ith object in Fuzzy-GSA. Other parameters have similar definitions as (4).

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

$$Q = \frac{TP}{TP + FN} \cdot \frac{TN}{FP + TN} \quad (15)$$

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:

$$Fit(O^i) = \sum_{l=1}^{K} Q_l \quad (16)$$

where $O^i$is ith object, Q_l is the fitness of the lth rule of the K rules in the rule set. Note that to find the optimum rule set, the defined fitness function in (4) should be maximized.

B. The structure of FGSA-Miner

Based on the above definitions and descriptions, the structure of the proposed *FGSA-miner* is as follows:

Step 1: **Initialization of Population and Internal Parameters**

In this step a random population is created and internal parameters are initialized. Internal parameters are:

-Masses of objects

-the initial value of gravitational coefficient

-the value of Kbest

-iteration counter (t)

-Number of total iterations (T)

-Other random variables

Step 2: **Evaluation of Fuzzy Inputs**

In this step the fuzzy inputs $(f_{best}(t), Var\_fit(t),\ and UN)$ of fuzzy controller are evaluated.

Step 3: **Evaluation of Fuzzy Outputs**

In this step the fuzzy controller updates its fuzzy outputs (Kbest, G) and prepares them for the next generation of Fuzzy-GSA.

Step 4: **Running with Updated Parameters**

The search loop is executed with the updated values of Kbest and G as the following stages:

-Calculation of best(t), worst(t) and $M^i(t)$ for i = 1,2,. . .,N.
-Calculation of the total force in different directions.
-Calculation of acceleration and velocity.
-Updating objects' position.
-Fitness evaluation of objects.

Step 5: **Checking Stop Criteria**

If the termination condition is reached, stop the algorithm; else go to step 2.

C. Rule Pruning

Rule pruning is a post processing stage in extracting effective rule set. This post-processing step is needed because rule pruning can avoid the overfitting of the rule-based classifier; increase the score of recognition and simplicity of the rules.

The commonly used algorithm for rule pruning is to iteratively remove one term at a time from the rule while this process improves the fitness of the rule. In other words, in the first iteration, one starts with the full rule. Then it is iteratively tried to remove each of the terms of the rule and the fitness of the resulting rule is computed using equation (4). The term whose removal most improves the fitness of the rule is effectively removed from it, completing the first iteration. In the next iteration, the term whose removal most improves the fitness of the rule is again removed and so on. This process is repeated until there is no term whose removal will improve the fitness of the rule.

5. Computational and Comparative Results

In this section, performance evaluation of the proposed FGSA-miner is investigated. Also the comparative results with GSA-miner, Ant-miner, PSO-miner, GA-miner, and CN2 are provided.

GSA-miner is a data miner which uses the standard GSA without any fuzzy controller. Ant-miner is a rule discovery approach based on the ant colony optimization technique [9]. PSO-miner is a rule discovery method which is developed based on the particle swarm optimization algorithm [13]. GA-miner is an approach which utilizes genetic algorithm for rule discovery [13]. CN2 is a traditional and well-known data mining method which is widely used for rule discovery from databases and searches for a rule list in an incremental fashion [1].

Two performance aspects are considered for comparing above mentioned data mining approaches. Those are recognition score for testing data (predictive accuracy) and rule simplicity.

Five pattern classification benchmarks and a practical pattern recognition problem with different feature vector dimensions (4 to 128), are used for performance evaluation and comparison of the results. A description of the data sets is given in the next subsection.

5.1. **Data Sets.** Five well-known data sets are as follows[1]:

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: 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.

Tic-Tac-Toe: This database encodes the complete set of possible board configurations at the end of tic-tac-toe games and contains 958 categorical cases in two different classes.

Hepatitis: This dataset contains 155 instances of 13 categorical and 6 continuous attributes which are separated into two classes.

The piratical pattern recognition problem is defined in below:

Radar Targets Data: An application of pattern recognition is Automatic Target Recognition (ATR) for continuous wave radars. In this paper Jet Engine Modulations approach (JEM) is used for this purpose. In this way the modulation of the radar wave by rotating propellers and jet engine blades of targets is considered [15]. Ten different flying objects were chosen as introduced in Table 1 for classification in 20 elevation angle. After sampling from backscattered signals and data reduction preprocess, we took 128 points FFT (Fast Fourier Transform) as feature vectors for each target in 10 dB signal to noise ratio for fifty times. In fact these FFT coefficients are the feature vectors for each flying object.

5.2. **Experimental Results.** For comparing five data miner techniques, performance measures are considered. Those are predictive accuracy and simplicity of obtained rule-sets.

The simplicity is measured by the number of discovered rules and the average number of terms (conditions) per rule [13].

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 traditional training set for rule discovery. 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 data mining method is run 10 times, 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 testing data sets.

The FGSA-miner, Ant-miner, PSO-miner, GA-miner and CN2, are tested on the data sets described in previous subsection.

---

[1] $Thesedatasetsareavailablefromthesite : http : //archive.ics.uci.edu/ml/datasets$

Number	***Target***	***Application***
1	V.F-3	Training
2	PC-7	Training
3	ANTONOV AN-12	Military
4	FFA AS ZZO118A	Training
5	BAE-248 SERIES 2B	Transportation
6	KJ 500-3S	Military
7	ROLLS ROYCEALI-SON	Military
8	KUZNETSORNK-8-2	Transportation
9	TUMMANSKY R-11 F2S	Military
10	ROLLS ROYCE 535 E1 H4	Military

Table 1. Ten Targets as Reference Classes

For FGSA-miner the initial population is set to 30 and maximum number of iterations is set to 10000. Minimum and maximum values of G are set to 0.0001 and 100 respectively. This set up is similar to GSA-miner where its population size is equal to 30 and maximum number of iterations is 10000. Also additional settings for GSA-miner are as follows:

G is set using equation (2). The initial value of Kbest is set to N (total number of agents) and is decreased linearly to 1. Extensive experiments were executed to find the best functions for G in each problem. Many mathematical functions were adopted and implemented for this reason to explore the best ones. Those are linear decreasing (LD) function, linear increasing (LI) function, linear decreasing and then increasing (LDI) function, linear increasing and then decreasing (LID) function, increasing with order two (IO2) function, decreasing and then increasing with order two (DIO2) function, and so on. Among these different schedules it was seen that the performance of exponential function (as reported in Ref [10]) has more superiority cases in comparison to other functions. Thus it has been adopted.

Parameter settings for Ant-miner, PSO-miner, GA-miner, and CN2 are exactly similar to which reported in [13], [9], and [1].

Table 2 shows the obtained average predictive accuracies by FGSA-miner and GSA-miner for six data sets. The results in this Table demonstrate that the performance of fuzzy GSA considerably outperforms the standard GSA in rule discovery problem. In fact, Table 2 conform that the standard GSA captures in local solutions when it is utilized for data mining task. Also the effectiveness of designed fuzzy controller is confirmed, where the obtained results by FSGA-miner is better than those of GSA-miner by at least 8.5%.

Tables 3 and 4 present the obtained predictive accuracy results with the estimated rule sets by five data miner.

	FGSA-miner	*GSA-miner*
Iris	96.7	86.1
Cancer	94.0	82.3
Dermatology	95.6	84.0
Tic-Tac-Toe	87.2	78.7
Hepatitis	94.1	85.3
Radar Targets	87.2	78.4

Table 2. Average Predictive Values (%) for FGSA-miner and GSA-miner

It can be seen from Tables 3 and 4 that, for Iris data the obtained minimum, maximum, and average predictive accuracy by FGSA-miner are better than other methods for testing data.

With regard to Cancer data, FGSA-miner has better minimum predictive accuracy in comparison to Ant-miner, PSO-miner, and GA-miner. But the obtained maximum average predictive accuracy by FGSA-miner outperforms other data miner techniques.

For Dermatology data, FGSA-miner outperforms other methods in minimum and average predictive accuracy values. The improvement of average predictive accuracy results for FGSA-miner with respect to Ant-miner, PSO-miner, GA-miner, and CN2 are 1.5%, 2.7%, 3.1%, and 5.4% respectively.

Although for Tic-Tac-Toe dataset, CN2 has better average accuracy, but the obtained maximum value of recognition score by the proposed FGSA-miner is better than other methods.

For Hepatitis data, FGSA-miner outperforms other data mining techniques in minimum, maximum, and average obtained accuracies.

Radar targets dataset is a high dimensional feature space problem. As it is shown in Tables 3 and 4, both maximum and average found values are surpass other techniques. It means that the proposed FGSA-miner can effectively find appropriate rule set in high dimensional spaces too.

FGSA-miner				*Ant-miner*			*PSO-miner*		
	Min.	Max.	Ave.	Min.	Max.	Ave.	Min.	Max.	Ave.
Iris	**93.4**	**98.3**	**96.7**	92.4	96.1	94.0	90.7	95.1	92.7
Cancer	92.8	**98.0**	94.0	91.5	98.0	95.9	90.3	97.0	94.6
Dermatology	**93.0**	96.9	**95.6**	91.4	97.2	94.1	88.7	95.1	92.9
Tic-Tac-Toe	79.0	**94.2**	87.2	70.4	74.3	73.0	85.8	91.6	89.1
Hepatitis	**90.2**	**95.6**	**94.1**	87.9	93.6	90.1	85.0	91.4	88.6
Radar Targets	85.6	**90.7**	**87.2**	80.0	84.9	83.3	83.8	87.4	85.5

Table 3. Minimum, Maximum and Average Predictive Accuracy (%) for FGSA-miner, Ant-miner, PSO-miner, GA-miner and CN2

GA-miner				*CN2*		
	Min.	Max.	Ave.	Min.	Max.	Ave.
Iris	90.0	97.1	95.5	88.9	95.2	93.1
Cancer	91.3	98.1	94.7	93.2	96.8	94.0
Dermatology	89.4	93.8	92.5	87.5	92.9	90.2
Tic-Tac-Toe	78.4	83.6	81.5	87.1	94.1	92.4
Hepatitis	86.9	92.5	90.7	88.6	95.0	90.0
Radar Targets	86.0	90.3	87.1	79.8	84.3	82.5

Table 4. The Continuation of Table 3

Tables 5 and 6 shows the simplicity measures which have been obtained by five data mining approaches. In this table the first column shows the averages number of discovered rules and standard deviations, and the second column presents relative number of terms per rule.

As it can be seen from the results of Tables 3 and 4 the rule list discovered by FGSA-miner is simpler for Iris data; because it has a smaller number of rules and terms per rule in comparison to other data mining techniques. For this data set, FGSA-miner discovered rule lists with the average number of rules of 4.9 and 2.17 terms per rule, whereas other techniques have larger values. Regarding the number of rules, the nearest result to FGSA-miner is for Ant-miner with the average number of rules of 5.3 and regarding the number of terms per rules, the nearest result is related to PSO-miner with 2.43.

In the Cancer data FGSA-miner outperforms other data miners regarding the number of rules; where it has the average number of rules of 6.1. PSO-miner competes with FGSA-miner by the average number of rules of 6.5. For this data set Ant-miner has the best conditions per rule by a little different of 0.04 in comparison to FGSA-miner.

In Dermatology data set FGSA-miner and PSO-miner have the minimum averages number of discovered rules with a little difference of 0.2. In fact both FGSA-miner and PSO-miner are the best data mining techniques among five data miners. This similarity again is repeated for the number of terms per rule, where these measures are 3.01 (for FGSA-miner) and 3.4 (for PSO-miner). Due to the large value of average of the number of rules, CN2 has the best number of terms per rule in comparison to other techniques.

For Tic-Tac-Toe the superiority of FGSA-miner for discovering simpler rule lists can be seen again; where its average number of rule is 7.5 and its number terms per number of rules is 1.85. It may be useful if we refer to the obtained average scores of recognition for Tic-Tac-Toe data (Tables 3 and 4). It was seen that CN2 outperforms the proposed FGSA-miner in average accuracy by 5.2%. Referring to Tables 5 and 6 appears that the rule lists discovered by CN2 are very more complex than those of FGSA-miner. The average number of rules for FGSA-miner to achieve 87.2% predictive accuracy is only 7.5, whilst the number of rules for obtaining 92.4% predictive accuracy is 40.1 for CN2.

	Number of rules				
	FGSA-miner	*Ant-miner*	*PSO-miner*	*GA-miner*	*CN2*
Iris	**4.9±0.25**	5.3±0.23	5.5±0.12	5.9±0.16	9.4±0.20
Cancer	**6.1±0.20**	6.8±0.53	6.5±0.30	7.0±0.24	17.8±0.5
Dermatology	7.1±0.24	7.7±0.21	6.9±0.27	7.9±0.19	18.8±0.41
Tic-Tac-Toe	**7.5±0.32**	9.1±0.54	9.0±0.67	12.3±0.36	40.1±3.31
Hepatitis	4.0±0.28	3.8±0.23	5.0±0.61	5.4±0.31	7.5±0.31
Radar Targets	**10.5±0.55**	12.5±0.8	18.5±0.47	16.9±0.52	32.7±0.7

Table 5. Simplicity Measures for Discovered Rules by FGSA-miner, Ant-miner, PSO-miner, GA-miner and CN2

	Number of terms / Number of rules				
	FGSA-miner	*Ant-miner*	*PSO-miner*	*GA-miner*	*CN2*
Iris	**2.17**	2.75	2.43	2.53	2.64
Cancer	2.12	**2.08**	2.26	2.35	2.46
Dermatology	3.01	3.55	3.40	3.71	2.53
Tic-Tac-Toe	**1.85**	1.89	2.54	2.37	3.40
Hepatitis	2.25	2.40	2.40	1.79	1.73
Radar Targets	**3.09**	3.38	5.07	4.60	5.12

Table 6. The Continuation of Table 5

The best number of rules and number of conditions per rules in Hepatitis data are appeared for Ant-miner and CN2 respectively. FGSA-miner competes with Ant-miner by a little value of 0.2 number of rules. Also, it follows CN2 by a minor difference of 0.52 number of terms per number of rules.

The simplicity results obtained for Radar targets dataset shows the effectiveness of the proposed FGSA-miner in extracting simpler rule lists in high dimensional feature space. Both the average number of rules and number of terms per number of rules are the best values for FGSA-miner.

6. Conclusion

The idea of intelligently controlling the search process of gravitational search algorithm was introduced and a fuzzy controller was designed for this purpose. Based on defined effective fuzzy inputs, fuzzy controller updates gravitational coefficient (G) and Kbest as two important parameters of GSA. The proposed intelligent GSA, (called *Fuzzy-GSA*) was used to introduce a novel data mining method for rule discovery task.

The proposed algorithm, named FGSA-miner, searches the rule space, to find the optimum rule set in order to maximize a fitness function which is related to the total true positives, false positives, true negatives, and false negatives.

The performance of the FGSA-miner was tested on five well-known benchmarks and a practical radar targets recognition problem. The obtained results were compared with a conventional GSA-miner, Ant-miner, PSO-miner, GA-miner, and CN2.

With regard to the *predictive accuracy* and rule set *simplicity*, FGSA-miner outperforms other techniques in many cases.

Study on capabilities of other controllers (like neural networks) instead of fuzzy controller for designing other intelligent GSAs is topic for future works.

References

[1] P. Clark and T. Niblet, *The CN2 induction algorithm,* Mach. Learn., **3(4)** (1989), 261-283.

[2] A. E. Eiben, R. Hinterding and Z. Michalewicz, *Parameter control in evolutionary algorithms*, IEEE Transactions on Evolutionary Computation, **3(2)** (1999), 124-141.

[3] A. Freitas, *A survey of evolutionary algorithms for data mining and knowledge discovery*, In: A. Ghosh, S. Tsutsui, eds., Advances in Evolutionary Computation, Springer-Verlag, 2001.

[4] R. Kohavi and M. Sahami, *Error-based and entropy-based discretization of continuous features*, Proc. 2^{nd} Int. Conf. Knowledge Discovery and Data Mining, Menlo Park, CA, (1996), 114-119.

[5] B. Liu, H. A. Abbass and B. Mckay *Classification rule discovery with ant colony optimization,* In Proceeding of the IEEE/WIC International Conference on Intelligent Agent Technology, Beijing, China, (2003), 83-88.

[6] P. M. Mary and S. Marimuthu, *Minimum time swing up and stabilization ofrotary inverted pendulum using pulse step control,* Iranian Journal of Fuzzy Systems, **6(3)** (2009), 1-15.

[7] E. Mehdizadeh, S. Sadi-nezhad and R. Tavakkoli-moghaddam, *Optimization of fuzzy clustering criteria by a hybrid PSO and fuzzy c-mean clustering algorithm,* Iranian Journal of Fuzzy Systems, **5(1)** (2008), 1-14.

[8] F. Moayedi, R. Boostani, A. R. Kazemi, S. Katebi and E. Dashti, *Subclass fuzzy-svm classifier an efficient method to enhance the mass detection in mammograms,* Iranian Journal of Fuzzy Systems, **7(1)** (2010), 15-31.

[9] R. S. Parpinelli, H. S. Lopes and A. A. Freitas, *Data mining with an ant colony optimization algorithm*, IEEE Transactions on evolutionary computing, **6(4)** (2002), 321-332.

[10] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, *GSA: A Gravitational Search Algorithm,* Information Sciences, **179(13)** (2009), 2232-2248.

[11] R. Sarkar, H. abbas and C. Newton, *Introducing data mining and knowledge discovery*, Heuristic and Optimization for Knowledge Discovery, Idea Group Publishing, (2008), 1-23.

[12] Y. Shi, R. Eberhart and Y. Chen, *Implementation of evolutionary fuzzy systems*, IEEE Transactions On Fuzzy Systems, **7(2)** (1999), 109-119.

[13] T. Sousa, A Silva and A. Neves, *Particle swarm based data mining algorithms for classification tasks*, Parallel Computing, **30(1)** (2004), 767-783.

[14] M. Stenes and H. Robous, *GA-fuzzy modeling and classification: complexity and performance*, IEEE Transactions on Fuzzy Systems, **8(5)** (2000), 509-522.

[15] S. H. Zahiri, H. Zareie and M. R. Agha-ebrahimi, *Automatic target recognition using jet engine modulation on backscattered signals,* In the Communication Proc. of 8'th Iranian Conf. on Electrical Engineering, Isfahan, (1996), 296-303.

[16] S. H. Zahiri and S. A. Seyedin, *Swarm intelligence based classifiers*, Int. J. of the Franklin Inst., **344(2)** (2007), 362-376.

[17] S. H. Zahiri, H. Rajabi Mashhadi and S. A. Seyedin, *Intelligent and robust genetic algorithm based classifier,* Iranian Journal of Electrical and Electronic Engineering, **1(3)** (2005), 1-9.

Seyed Hamid Zahiri, Department of Electrical Engineering, Faculty of Engineering, Birjand University, Birjand, Iran

E-mail address: hzahiri@@birjand.ac.ir