Efficient mining of association rules based on gravitational search algorithm

Fariba Khademolghorani[1], Ahmad Baraani[2] and Kamran Zamanifar[3]

[1] Young Researchers Club, Islamic Azad University, Najafabad Branch, Isfahan, Iran

**[2] Department of Computer Engineering, University of Isfahan
Isfahan, Iran**

**[3] Department of Computer Engineering, Islamic Azad University , Najafabad Branch
Isfahan, Iran**

Abstract

Association rules mining are one of the most used tools to discover relationships among attributes in a database. A lot of algorithms have been introduced for discovering these rules. These algorithms have to mine association rules in two stages separately. Most of them mine occurrence rules which are easily predictable by the users. Therefore, this paper discusses the application of gravitational search algorithm for discovering interesting association rules. This evolutionary algorithm is based on the Newtonian gravity and the laws of motion. Furthermore, contrary to the previous methods, the proposed method in this study is able to mine the best association rules without generating frequent itemsets and is independent of the minimum support and confidence values. The results of applying this method in comparison with the method of mining association rules based upon the particle swarm optimization show that our method is successful.

Keywords: *Association Rules, Gravitational Search Algorithm, Swarm intelligence.*

1. Introduction

Data mining, popularly referred to as hidden knowledge discovery from data became essential to support decision-making and predict future behavior [1]. There are a lot of data mining tasks such as association rule, sequential patterns, time series, classification, clustering, etc [1, 2].

A basket indicates items purchased by a customer at a specific time. Customer's purchases can be prospected by analyzing market baskets. It helps vendors for future schematizations. The most important application of data mining is discovering association rules. This is one of the most important methods for pattern recognition in unsupervised systems. It was first proposed by Agrawal et.al in 1993 in order to analyze the customers' market basket. So far, many algorithms proposed for discovering association rules are based on Agrawal, Apriori [2], SETM [3], AIS [2], DIC [4] and other methods. However, these algorithms have their limitations. In these methods, rules with high occurrence in the entire database are considered as the best rules, whereas most of these rules can easily be predicted by the users. Therefore, they are not interesting. Also, most of the previous algorithms mine occurrence rules with a large number of attributes, which are not understandable for the user. Therefore, the user will never use them. In these methods, two parameters, minimum support and minimum confidence thresholds, are always determined for any databases. Hence, these algorithms lack both objectiveness and efficiency [3]. This particular makes these methods depended on datasets and it must execute several time.

Therefore, we propose a method, which not require to appointment support and confidence in our method and extract best rules in once executed. Previous method mine association rule in two stages. First they find frequent itemsets and then extract association rules from frequent itemsets.

A few evolutionary algorithms have been used with multi-objective functions. Evolutionary algorithms such as genetic, evolutionary, ant colony, simulated annealing and particle swarm optimization have been used in the area of mining association rules. This paper proposed two algorithms based on gravitational search algorithm and binary gravitational search algorithm without considering minimum support, confidence and interestingness, and extract best rule with high of them. If we do not require using support, we will not use apriori base method for rule mining, whereas in our method we can unused support with change fitness. Our method is very flexible on changing fitness, so user can define any normal multi-objective fitness with support, confidence and etc and obtains his/her interesting rules. Moreover, he can define the fitness function so that the order of items is considered on importance of rules.

The remainder of this paper is organized as follows: Section 2 describes the preliminaries, Section 3 introduces the gravitational search algorithm, Section 4 describes the proposed algorithms in detail, Section 5 reports the computational results and finally, Section 6 presents the conclusions and future research directions.

2. Preliminaries

In this section, we explain the association rule. Then we present a brief overview of related works on association rules.

2.1 Association rules

Usually, Association rules are shown in the form of *A*=>*C* in which *A* is antecedent itemsets of the rule and *C* is consequent itemset of the rule. It is assumed that there exists the transactional database $D=\{T_1,T_2,...,T_n\}$. If $I=\{I_1,I_2,...,I_n\}$ is a set of all items appearing in *D*, then any non-empty subset *Y* from *I* is called an itemset, an itemset that contains *k* items is called a *k*-itemset. The itemset which satisfies the minimum support is called “frequent itemset”.

As was explained before, these algorithms are perform in two stages, and therefore both support and confidence values need be calculated. Support is a statistical measure that indicates the ratio of the records that satisfies both the antecedent and the consequent of the rule, and the second parameter which is known as confidence factor or prediction accuracy of a r ule, indicates the strength of implication. These are calculated from 1 and 2 equations, respectively. These rules which satisfy both minimum support and minimum confidence are called strong association rules [1].

$$Support\ (A => C) = \frac{|(A \cup C)|}{TotalRcord} \tag{1}$$

$$Confidence\ (A => C) = \frac{SUP(A \cup C)}{SUP(A)} \tag{2}$$

2.2 Related works

Recently, optimization methods for mining association rules were applied. The idea of using genetic algorithm for extracting frequent itmesets has been applied [5]. In [6], has been extracted association rules based on the genetic algorithm without considering minimum support. In this method, relative confidence was used as the fitness function. But only rules with fixed length are extracted. Also, the fitness function of this method is in such a way that it is trapped into local optimum, and hence many rules are generated. Later, other researchers improved it by defining a new multi-objective fitness function [7].

In [8], a g enetic clustering method has been proposed. A method based on clustering technique for extracting generalized fuzzy association rules has been proposed [9]. A new method for mining fuzzy association rules and membership functions has been proposed by using genetic algorithm and based on clustering [10]. In [11] proposed a method of extracting association rules by using multi-objective genetic algorithm. Other researcher proposed a multi-objective differential evolution algorithm to extract numeric association rules [12].

In [13], a method of extracting frequent itemsets has been proposed by combining PSO with Ant algorithm. This method, in comparison with the GAR algorithm in [5] is faster and has more accuracy but it can discover only frequent itemsets and GAR has the same limitation. Also, in [14], by using particle swarm algorithm has been proposed a method for mining ARs from transactional databases. But, there is no studies exist to extract the association rules by using the gravitational search algorithm.

In this paper, we have tried to provide a useful approach for mining high-quality association rules by using this evolutionary algorithm. In the next section we briefly introduce it.

3. Gravitational Search Algorithm

Gravitational search algorithm is one of the new optimization algorithms that is based on the law of gravity and mass interactions. This algorithm was developed as GSA and BGSA in continuous and binary versions in 2009, 2010 respectively [15, 16].

GSA, the searcher agents are a collection of *N* masses, and their interactions are based on the Newtonian laws of gravity and motion. Each mass is considered as a solution (object), the position of the *i*th mass is defined by (3). x_i^d presents the position of *i*th agent in the *d*th dimension and n is the space dimension.

$$\mathrm{X_i} = (x_i^1,...,x_i^d,...,x_i^n) \qquad for\ \ i = 1,2,...,N \tag{3}$$

This algorithm starts with a random initial population. The heavy masses correspond to good solutions of the problem, as in (4) and (5). The values of masses are calculated using the map of fitness. fit_i(t) represent the fitness value of the agent *i* at *t*. This means that better agents have higher attractions and walk more slowly.

$$m_i = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \qquad (4)$$

$$M_i(t) = \frac{m_i}{\sum_{j=1}^{N} m_j} \qquad (5)$$

All these objects attract each other by a gravity force, and this force causes a movement of all objects globally towards the objects with heavier masses.

Assuming the equality of the gravitational and inertia mass, as in (6), the value of total force that acts on agent *i* in dimension *d*, is equal to the value of the acceleration that acts on agent *i* in dimension *d* [15].The total force that acts on agent *i* in a dimension *d* is a randomly weighted sum of *d*th component of the forces exerted from *Kbest* agents, as in (7). *Kbest* is the set of first *K* agents with the best fitness value and is a function of time, initialized to K_0 at the beginning and decreasing linearly with time. In such a way, at the beginning all agents apply force, and at the end, there will be just two percent of agents apply force to others [15].

$$\mathrm{M}_{ai} = \mathrm{M}_{pi} = \mathrm{M}_{ii} = \mathrm{M}_i, i = 1, 2, \ldots, \mathrm{N} \qquad (6)$$

$$a_i^d(t) = F_i^d(t) = \sum_{j \in kbest, j \neq i} r_i\, G(t) \frac{M_j(t)}{R_{ij}(t) + \varepsilon} (x_j^d(t) - x_i^d(t)) \qquad (7)$$

In (6), Ma_i is the active gravitational mass related to agent *i*, Mp_i is the passive gravitational mass related to agent *i*, Mi_i is the inertia mass related to agent *i*. In (7), *G(t)* is gravitational constant at time *t* such as *G(G0,t)* that is initialized at the beginning and will be reduced with time to control the search accuracy, ε is a small constant, and $R_{ij}(t)$ is the Euclidian distance between two agents *i* and j. r_j is a random number in the interval [0,1].

In GSA, the next velocity of an agent is considered as a fraction of its current velocity added to its acceleration. Therefore, its velocity and its position could be calculated as (8) and (9). r_i is an uniform random variable in the interval [0, 1].

$$v_i^d(t+1) = r_i * v_i^d(t) + a_i^d(t) \qquad (8)$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \qquad (9)$$

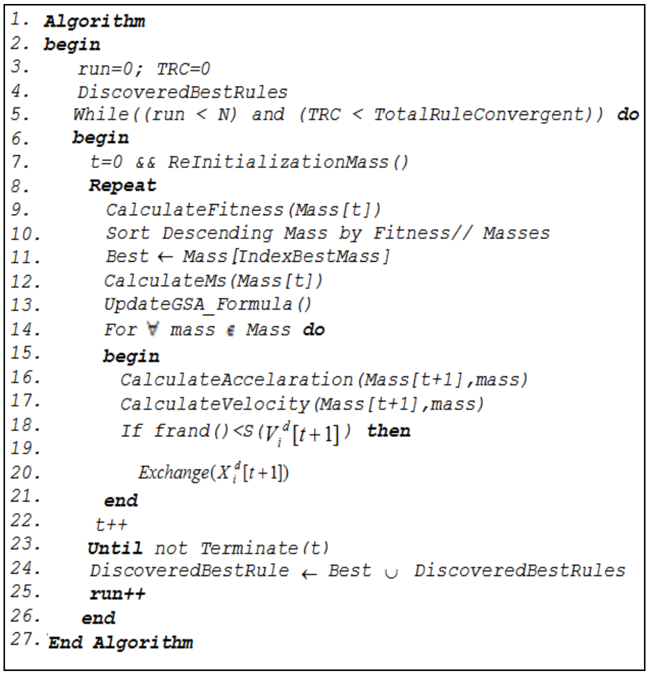
The main difference between continuous and binary GSA is that in the binary algorithm, the position updating means a switching between "0" and "1" values. This switching should be done according to the mass velocity. The current bit value of the position of any mass is changed with a probability that is calculated according to the mass velocity. BGSA updates the velocity based on (8) and considers the new position to be either 1 or 0 with the given probability. A proper probability function should be defined such that for a small $|v_i^d|$, the probability of changing x_i^d must be near zero and vice versa. To achieve a good converge rate, the velocity has been limited, $|v_i^d| < v_{max}$. v_{max} is set to be 6 [16].

$$S(v_i^d(t)) = \left|\tanh(v_i^d(t))\right| \qquad (10)$$

It is to be noted here that the distance, *R*, is computed based on the Hamming distance. In BGSA, *G* is considered as a linear decreasing function as *G_0(1-t/T)*. It will be useful to remind some basic concepts of GSA for BGSA [16].

4. The proposed algorithms

We proposed two algorithms which are called ARMGSA and ARMBGSA. These algorithms are presented for exploring association rules from transactional databases. The following of this section are some important parts of the algorithm which are explained: masses encoding, fitness function, and finally the last part of the section, the pseudo-code existed in Fig. 1 is explained.



```
Algorithm
begin
    run=0; TRC=0
    DiscoveredBestRules
    While((run < N) and (TRC < TotalRuleConvergent)) do
    begin
      t=0 && ReInitializationMass()
      Repeat
        CalculateFitness(Mass[t])
        Sort Descending Mass by Fitness// Masses
        Best ← Mass[IndexBestMass]
        CalculateMs(Mass[t])
        UpdateGSA_Formula()
        For ∀ mass ∈ Mass do
        begin
          CalculateAccelaration(Mass[t+1],mass)
          CalculateVelocity(Mass[t+1],mass)
          If frand()<S($V_i^d[t+1]$) then

            Exchange($X_i^d[t+1]$)
        end
       t++
      Until not Terminate(t)
      DiscoveredBestRule ← Best ∪ DiscoveredBestRules
      run++
    end
End Algorithm
```

Fig. 1 Pseudo-code of ARMBGSA.

4.1. Masses encoding

In this paper, each mass represents a r ule and each rule contains of a series of decision variables which represent the status of every item in the rule

In the ARMBGSA, every mass can take only "0" or "1". Therefore, according to Fig. 2 in masses encoding, each mass has 2*n* bits in lieu of *n* items in any dataset. If these two bits are 00 then the attribute appears in the antecedent part and if it is 11 then the attribute appears in the consequent part. And the other two combinations, 01 and 1 will indicate the lack of the attribute in either of these parts.

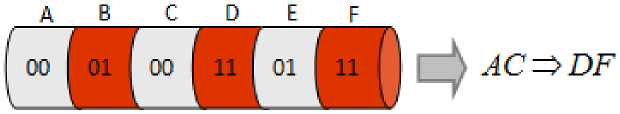


Fig. 2 Masses encoding in ARMBGSA.

In masses encoding in the ARMGSA, every mass has *n* decision variables in lieu of *n* items in any dataset. This means that the *i*th variable which is known as ES_i indicates the status of *i*th item and can take values between “0” and “1”. In this way, if $0.00 \leq ES_i \leq 0.33$, the *i*th attribute is in the antecedent of the rule and if $0.33 < ES_i \leq 0.66$, this attribute is in the consequence of the rule and if $0.66 < ES_i \leq 1$, it means the lack of *i*th attribute in the rule.

4.2. Fitness function

The fitness function provided in this study is in (12). Although non-low support value of the rule is a basic criterion in extracting association rules, high support value of the rule is not the indication of its being interesting for the user. Since the mining association rule is a part of a process of extracting hidden information, it must discover those rules which are interesting for the user; that is those which have comparatively less occurrence in the entire database; discovering such rules is more difficult.

$$fit(i) = \alpha_1 [\frac{Sup(A \cup C)}{Sup(A)}].[\frac{Sup(A \cup C)}{Sup(C)}].[1 - \frac{Sup(A \cup C)}{|D|}] + \alpha_2 \frac{NumberField(i)}{MaxField} \qquad (12)$$

Therefore, interestingness measure in [11] is used in fitness function according to the first parameter. In this parameter |*D*| is the total number of records in the database. This relation has three parts:

- $[Sup(A \cup C) / Sup(A)]$ indicates the probability of creating the rule depending on the antecedent part.
- $[Sup(A \cup C) / Sup(C)]$ shows the probability of creating rule depending on the antecedent part.

In fact most of these are interesting rules in which the rate of acquired information is approximately the same in both antecedent and consequent parts of the rule. In this parameter the support count of the rule antecedent and the support count of the rule consequent are used.

- In the third part of this parameter, those rules which have a very high support count and high frequency will be less interesting, because such rules are easily predictable by the users.

In databases with the large number of attributes we have the problem of creating rules with the several attributes. However, all these rules are not useable for the user. Thus, the second parameter of the fitness function prevents from creating these rules.

It should be noted that α_1 and α_2 will be specified by the user and one might increase or decrease the effects of parameters of fitness function.

4.3. Description of the proposed algorithms

This section describes our approach. Fig. 1 shows the pseudo-code of the ARMBGSA algorithm.

The difference between the ARMGSA and ARMBGSA methods is in steps 19 to 21. However, the implementation details of these two algorithms have some differences. Furthermore for computing the distance of two masses in ARMBGSA their Hamming distance and in ARMGSA the Euclidian distance is used.

The algorithm is run as times as *N* or the number of desired rules. Moreover, each run includes a number of generations. In addition to this termination condition, there is another condition called *TRC*. It is zero in the first algorithm run, then, it is increased by each discovered repetitive solution. If the quantity of *TRC* is more than or equal to *"TotalRuleConvergent"* value, future algorithm runs are avoided. At the beginning of the algorithm, sets of the *"DiscoveredBestRules"* and "*Best"* are empty. "*Best*" is a vector for keeping the best solutions in each generation.

At the first iteration of algorithm, each individual is initialized randomly as a solution.

In each of generation, until the reaching the termination conditions, the masses are evaluated and then, the best and worst of masses are determined. In each generation, the best discovered rule is added to Best vector. The individuals of the population are sorted in descending order according to their fitness value.

Then, for each individual, its mass value is calculated. Finally, the quantities of the parameters concerning the gravitational search algorithm are updated. Each time, the *Kbest* of the masses with elitist strategy in order to influence the all masses are selected. In the first generation, all masses affect each other and the quantity of *Kbest* is decreased to 2% by the "*Linear"* function. Then, the acceleration and the velocity of each mass are computed and the position of masses are updated and evaluated; the rule is valid if it has at least one attribute in the antecedent of the rule and one in the consequence of the rule.

In ARMBGSA algorithm, the updating of the masses position in each dimensions such as *d,* is based on steps 19 to 21. The *frand()* function makes a randomly real number in the interval [0,1]. Based on the mentioned concepts, for a small $|v_i^d|$, $S(v_i^d)$ is small, also.

As a result, the probability of changing x_i^d is near zero and for a large $|v_i^d|$, the probability of x_i^d movement is high.

The position of each mass in the ARMGSA is calculated in (9) such that the position of i*th* item should always be between 0 and 1; otherwise, it is modified.

After reaching the termination condition in each iteration, that the best mined rule from the *Best* is added to the *DiscoveredBestRule* set. This process is continued until termination conditions. The termination conditions in our proposed method consist of:

1) Number of generation (*No.Gen*).
2) No individuals improving (*Best-Worst* < *BW*). *BW* is a small constant that determined before start of the algorithm.
3) No better solutions discovering (MinGenConvergent).

5. Computational results

ARMGSA and ARMBGSA were implemented and executed in C# with .Net frame work 3.5, on a PC with Intel Dual-Core 2.1 GHz operator on a 3GB Ram and with Vista Home Premium Windows.

The setting of the used parameters is shown in Table 1. Factors α_1 and α_2 that have been used in fitness function were selected as 0.8 and 0.1, respectively.

The first experiment in this section is evaluating the efficiency and usefulness of the algorithm provided in this study. Therefore, we compare it with the method of mining association rules based on the particle swarm optimization in [14]. In order to compare the experiments justly, we examined it by our presented fitness function.

Table 1: Used parameter values for ARMGSA and BARMGSA.

MassSize	No.Gen	N
40	70	20
TotalRuleConvergent	MinGenConvergent	BW
10	8	0.05

The initial setting of this algorithm's parameters is given in Table 2. For doing this examination we used the basket-market dataset existing in Clementine 12.0 tool [15] and two datasets in UCI at http://www.ics.uci.edu/~mlearn. The specifications of datasets used from UCI are given in Table 3. The market dataset includes 11 binary attributes and 1000 records. All attributes in datasets are categorical; therefore we converted them into the Boolean datasets. Every attribute with any amount is considered as an item.

Table 2: Used parameter values for proposed PSO in [14].

SwarmSize	No.Gen	C1, C2	W	N
40	70	2,2	0.9-0.4	20

Table 3: The specifications of datasets used from UCI.

	Balance Scale	Care Evaluation
No. record	625	1728
No. attribute	23	26

Table 4 shows the comparison of mean results obtained from ARMBGSA, ARMGSA and the proposed PSO in [14] with our fitness function. The mean number of best different mined rule, the mean number of attributes contained in the rules and the mean of the support, confidence and cosine value of these rules are shown in this table.

According to Table 4, the proposed algorithms which are based on the gravitational search algorithm have better results compared to the AR mining method based on the proposed PSO. ARMBGSA and ARMGSA obtained rules with higher confidence and support value. In addition the user's comprehension of these rules and the rules interestingness for the user are acceptable; also the number of attributes obtained from this method is smaller. This means that the algorithm of mining association rules based on the gravitational search algorithm is more capable of discovering global solutions in comparison with the mining association rules based on the particle swarm optimization.

Table 4: Comparisons of the results.

	Balance scale			Car Evaluation			Basket-Market		
	ARMBGSA	ARMGSA	PSO	ARMBGSA	ARMGSA	PSO	ARMBGSA	ARMGSA	PSO
Sup (%)	9.86	9.62	8.54	9.13	8.44	7.21	15.55	15.06	14.42
Conf (%)	66.48	62.28	61.88	62.93	60.97	57.66	64.11	61.32	52.41
N.Att	6.5	7.2	7.2	6.2	6.4	6.8	2.2	2.3	22
N.Rue	15	16	16	18	18	20	10	10	11
Cossine	0.59	0.58	0.55	0.61	0.58	0.52	0.58	0.56	0.52

On the other hand, ARMBGSA method presents better results in comparison with ARMGSA method.

Fig. 3 shows the average execution time of 10 replications scalability of these algorithms in lieu of discovering a rule from basket-market data set and based upon the record size increase. These algorithms were implemented and executed under the same condition.

According to this diagram in Fig. 3, ARMBGSA presented in this study has faster convergence speed (red line) and also generates acceptable and high-quality global solutions. Then, in next priority, ARMGSA has a better mining capability and speed (blue line).

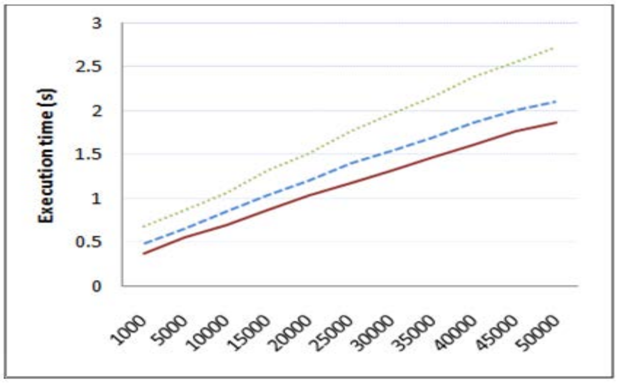


Fig. 3 Relationship between population size and execution time.

As is seen, the gradient of execution time is steeper in the algorithm of mining association rules based on particle swarm optimization. In the same way, we repeated our experiment on the other datasets considered in this paper. The results indicate an appropriate execution time for the proposed method. This appropriate is due to:

1) the Nature and efficiency of the gravitational search algorithm in comparison with the particle swarm optimization.
2) Proper proposed termination conditions which avoid useless algorithm run.

According to the comparison in Table 5, the gravitational search algorithm has better global search capability compared to particle swarm optimization.

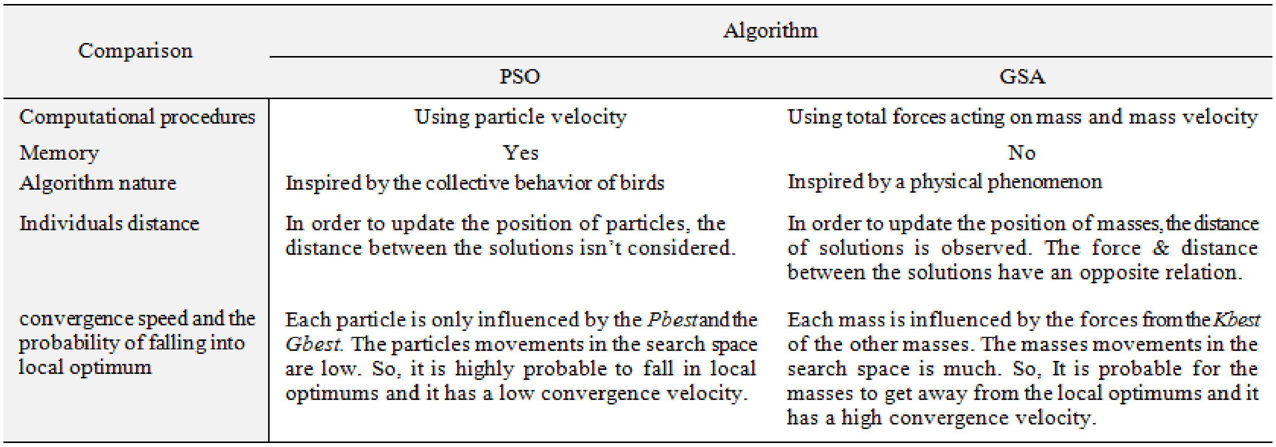
6. Conclusions and future research

Mining associating rules from a large dataset takes a lot of time. In this paper, we proposed a new approach in two versions based on the binary and continues gravitational search algorithm for mining interesting and understandable association rules, we have named these algorithms ARMBGSA and ARMGSA.

We tested our algorithms on three datasets with several experiments for evaluating their behavior. It can be said as time passes, masses have a better effect on each other and individuals evolve into the convergent proper positions and reach a h igher fitness value. The results show the appropriate success of these methods in comparison with the method of extracting rules by the use of the PSO, both in its being useful and in its execution time and they are more capable of discovering global solutions. Also, ARMBGSA method has a higher speed and discovers proper solutions in compared to ARMGSA. Some other specifications of our algorithms are as the following:

- Observing the proper termination conditions.
- It is able to automatically find all rules without relying upon the minimum support and the minimum confidence thresholds and it generates association rules without generating frequent itemsets.
- In this algorithm an appropriate multi-objective fitness function has been used to mine interesting and understandable rules for users.

Table 5: Comparison table of PSO and GSA.



Comparison	Algorithm	
	PSO	GSA
Computational procedures	Using particle velocity	Using total forces acting on mass and mass velocity
Memory	Yes	No
Algorithm nature	Inspired by the collective behavior of birds	Inspired by a physical phenomenon
Individuals distance	In order to update the position of particles, the distance between the solutions isn't considered.	In order to update the position of masses, the distance of solutions is observed. The force & distance between the solutions have an opposite relation.
convergence speed and the probability of falling into local optimum	Each particle is only influenced by the *Pbest* and the *Gbest*. The particles movements in the search space are low. So, it is highly probable to fall in local optimums and it has a low convergence velocity.	Each mass is influenced by the forces from the *Kbest* of the other masses. The masses movements in the search space is much. So, It is probable for the masses to get away from the local optimums and it has a high convergence velocity.

Several works are suggested to improve the proposed algorithm: extending the proposed approach in this study is suggested in order to search categorical and quantitative rules. Considering that only positive rules discovered in these algorithms, discovering both positive and negative rules is suggested for future works.

REFERENCES

[1] J. Han, and M. Kamber, Data Mining: Concepts and Techniques(2nd ed.), Elsevier, 2006.

[2] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases" in: Proceedings of ACM SIGMOD Conference on Management of Data, Washington, Vol. 22, No. 2, 1993, pp. 207–216

[3] M. Houtsma, and A. Swami, "Set-oriented mining of association rules", Research Report RJ 9567, IBM Almaden Reseach Center, 1993.

[4] S. Brin and R. Motwani and J.D. Ullman and S. Tsur, "Dynamic itemset counting and implication rules for market basket data," Proc. of ACM SIGMOD International Conf. on Management of Data. Tucson, Vol. 26, No. 2, 1997, pp. 255 - 264.

[5] J. Mata, J.L. Alvarez, and J.C. Riquelme, "An evolutionary algorithm to discover numeric association rules" ,In Proceedings of the ACM symposium on Applied computing SAC, 2002, pp. 590–594.

[6] X.Yan, CH. Zhang, and SH. Zhang, "Genetic algorithm-based strategy for identifying association rules without specifying actual minimum support", Elsevier Expert Systems with Applications, Vol. 36, No. 2, 2008, pp.3066-3076.

[7] H. Qodmanan, M. Nasiri, and B. Minaei-Bidgoli, "Multi objective association rule mining with genetic algorithm without specifying minimum support and minimum confidence", Elsevier Expert Systems with Applications, Vol. 38, No. 1, July 2010 , pp. 288-298.

[8] M. Kaya, and R. Alhajj, "Genetic algorithm based framework for mining fuzzy association rules", Elsevier Fuzzy Sets and Systems, Vol. 152, No. 3, 2005, pp. 587-601.

[9] H-P. Chiu, Y-T. Tang, and K-L. Hsieh, "A Cluster-Based method for mining generalized fuzzy association rules", IEEE First International Conference on I nnovative Computing, Information and Control , Beijing, China, Vol. 2, No. 6, Sep 2006, pp. 249-262.

[10] Ch. Chun-Hao, T.-P. Hong, and V.S.Tseng, "A cluster-based fuzzy-genetic mining approach for association rules and membership functions", IEEE ICICIC Fuzzy Systems , Vancouver, BC, 2006, pp. 1411-1416.

[11] A. Ghosh, and B. Nath, "Muti Objective Association Rule Mining Using Genetic Algorithm", Elsevier Information Sciences, Vol. 163, No. 1, 2004, pp.123–133.

[12] G. Alatas, and E. Akin, "MODENAR: Multi-objective differential evolution algorithm for mining numeric association rules", Elsevier Applied Soft Computing, Vol. 8, No. 1, 2008a, pp.646-656.

[13] O. M. Badawy, M. I. Habib, and A. A. Sallam, "Quantitative Association Rule Mining Using a Hybrid PSO/ACO Algorithm (PSO/ACO-AR)", ACIT'2008, Hammamet, Tunisia, Dec 2008, pp.1-9.

[14] R.J. Kuo, C.M. Chao, and Y.T. Chiu, "Application of particle swarm optimization to association rule mining", Elsevier Applied Soft Computing, Vol. 11, No. 1, 2009, pp. 326-336.

[15] E. Rashedi, H. Nezamabadi, and S. Saryazdi, "A Gravitational Search Algorithm Information", Elsevier Information Sciences, Vol. 179, No. 13, Jun 2009, pp. 2232–2248.

[16] E. Rashedi, H. Nezamabadi, and S. Saryazdi, "BGSA: binary gravitational search algorithm", Springer Natural Computing, Vol. 9, No. 3, Jun 2010, pp. 727–745.

[17] M. Clerc, Think Locallly, Act Locally: The Way of Life of Cheap-PSO, an adaptive PSO. Technical report, http://clerc.maurice.free.fr/pso/, 2001

[18] SPSS Inc. "Clementine 12.0 Algorithms Guide". http://www.spss.com.

Fariba Khademolghorani received her BS and M S degrees in computer engineering from Islamic Azad University of Najafabad, Iran in 2007 and 2011. She has written about 7 l ocal papers. Her research interests include swarm Intelligence, data mining, soft computing and evolutionary algorithms.

Dr Ahmad Baraani is an assistant professor in the department of Computer Engineering of University of Isfahan. His areas of research include data mining, data bases & security, machine learning, software engineering.

Dr Kamran Zamanifar is an assistant professor in the department of Computer Engineering of University of Isfahan and also he works in the department of Computer Engineering from Islamic Azad University of Najafabad. His subjects are on aspect for parallel algorithms, soft computing, evolutionary algorithms and distributed systems.