Multi-objective Numeric Association Rules Mining via Ant Colony Optimization for Continuous Domains without Specifying Minimum Support and Minimum Confidence

Parisa Moslehi[1], Behrouz Minaei Bidgoli[2], Mahdi Nasiri[3], Afshin Salajegheh[4]

[1] Computer Department, Islamic Azad University South-Tehran Branch
Tehran, Tehran, Iran

[2] Computer Department, Iran University of science and Technology
Tehran, Tehran, Iran

[3] Computer Department, Iran University of science and Technology
Tehran, Tehran, Iran

[4] Computer Department, Islamic Azad University South-Tehran Branch
Tehran, Tehran, Iran

Abstract

Currently, all search algorithms which use discretization of numeric attributes for numeric association rule mining, work in the way that the original distribution of the numeric attributes will be lost. This issue leads to loss of information, so that the association rules which are generated through this process are not precise and accurate. Based on this fact, algorithms which can natively handle numeric attributes would be interesting. Since association rule mining can be considered as a multi-objective problem, rather than a single objective one, a new multi-objective algorithm for numeric association rule mining is presented in this paper, using Ant Colony Optimization for Continuous domains (ACO_R). This algorithm mines numeric association rules without any need to specify minimum support and minimum confidence, in one step. In order to do this we modified ACO_R for generating rules. The results show that we have more precise and accurate rules after applying this algorithm and the number of rules is more than the ones resulted from previous works.

Keywords- *Ant Colony Optimization for Continuous Domains, Numeric association rules mining, Multi objective association rules mining*

1. Introduction

Data mining is the most instrumental tool in discovering knowledge from transactions [1][2][3]. Also data mining is known as an integral part of knowledge discovery in databases (KDD). Transactional database refers to the collection of transaction records, which in most cases are sales records. Data mining on transactional database focuses on the mining of association rules, finding the correlation between items in transaction records. Most of the association rule algorithms are based on methods proposed by Agrawal, Imielinski, and Swami [4] and Agrawal and Srikant [5], which are Apriori [4], SETM [6], AIS [4] and Pincer search [7] etc. Neither the rules with numeric attributes nor the rules in the form of $I_8I_{10}I_{12} \rightarrow I_4I_5I_9$ can be discovered by these methods [8]. That is why numeric association rule mining algorithms have been proposed.

On the other hand while we use these methods we must specify minimum support and minimum confidence. This issue makes these methods dependent on datasets and they must execute several times. We do not require specifying support and confidence thresholds in our method; also we extract best rules in one execution of the algorithm. Previous methods mine association rules in two steps. First they find frequent itemsets and then extract association rules from them. In a numeric association rule, attributes can be Boolean, numeric or categorical. Since, mining numeric association rules is a hard optimization problem rather than being a simple discretization one, algorithms that natively handle numeric attributes usually perform better.

In recent years, the swarm intelligence paradigm received widespread attention in research. Two main algorithms of which that are popular swarm intelligence metaheuristics for data mining, are Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO). Swarm Intelligence is based on social behavior that can be observed in nature, such as ant colonies, flocks of birds, fish schools and bee hives, where a number of individuals with limited capabilities are able to come to intelligent solutions for complex problems [9].

Recently, a continuous version of ACO metaheuristic has been proposed for tackling continuous attributes and solving continuous optimization problems in [10] and [11] by Socha.

There is not any study which uses ACO$_R$ for mining numeric association rules. In this paper we propose a method using ant colony optimization for continuous domain (ACO$_R$) and extract best rules with high support and confidence without any need to specify minimum support and minimum confidence thresholds.

The rest of this paper is organized as follows. In section 2, a brief explanation of Ant Colony optimization algorithm for continuous domain (ACO$_R$) is presented. In section 3, numeric association rule mining and in section 4 related works are discussed. The proposed multi objective association rule mining algorithm via ACO$_R$ is presented in section 5. Experimental setup and results is presented in section 6. Finally we conclude with a summary in section 7.

2. Ant Colony Optimization for continuous domains (ACO$_R$)

While ACO uses a discrete probability distribution for choosing a solution, ACO$_R$ uses a probability density function (PDF) and samples it. A Gaussian function is used as PDF in ACO$_R$.

In ACO a pheromone table is used to store pheromone information. ACO$_R$ uses a solution archive of size k in order to describe the pheromone distribution over the search space. Here, k is the number of complete solutions to the problem. Considering the solution archive as a matrix, each entry is called s_j^i where i=1,2,…,n is the number of dimensions and j=1,2,…,k is the number of rows.

First the archive is initialized with k random solutions. These solutions are ranked based on their quality. The weight ω_j of a complete solution S$_j$ is calculated according to its rank:

$$\omega_j = \frac{1}{qk\sqrt{2\pi}} e^{\frac{-(j-1)^2}{2q^2k^2}} \qquad (1)$$

where q is a parameter of the algorithm. The effect of q is that, if it is small the best-ranked solutions are strongly preferred and when it is large, the probability becomes more uniform [11].

Each ant chooses a solution from the archive probabilistically for building its own solution. The probability of choosing S$_j$ by an ant is:

$$p_j = \frac{\omega_j}{\sum_{r=1}^{k} \omega_r} \qquad (2)$$

After choosing a solution s_j^i in the archive, each ant samples a Gaussian function:

$$P(x) = g(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \qquad (3)$$

where μ and σ are the mean and standard deviation of the Gaussian function respectively. If an ant chooses a solution s_j^i then the value of s_j^i is assigned to μ, and the standard deviation is assigned as follows:

$$\sigma \leftarrow \xi \sum_{r=1}^{k} \frac{|s_r^i - s_j^i|}{k-1} \qquad (4)$$

where ξ is a parameter of the algorithm that has the same effect as the pheromone evaporation parameter in ACO algorithm. The higher the value of ξ, the lower the convergence speed of the algorithm [11].

3. Numeric association rule mining

In a basket market transactions where transactions are a list of items purchased by a customer, the knowledge that association rules give us are something like: “70% of customers who buy A also buy B”. The applications of association rules are in discovering customer buying patterns for cross-marketing and attached mailing applications, catalog design, product placement, customer segmentation, etc., based on their buying patterns [12].

Given a set of transactions, the problem of mining association rules is to find all association rules that have support and confidence greater than the user-specified minimum support and minimum confidence respectively. Association rules can be boolean or numeric. Numeric association rules can have some numeric attributes, like age, height and etc. they also can have categorical attributes like gender, brand, and etc. numeric attributes need to be dicretized in order to transform the problem into a Boolean one, before mining association rules.

An example of a numeric association rule in an employee database is like this [13]:
“Age ∈ [25,36] ∧ Sex=Male → Salary ∈ [2000,2400] ∧ Have-Car=Yes”
(Support=4%, Confidence=80%)
In this numeric association rule, “Age ∈ [25,36] ∧ Sex=Male” is antecedent and “Salary ∈ [2000,2400] ∧ Have-Car=Yes” is consequent part. This numeric association rule states that “4% (support) of the employees are males aged between 25 and 36 and earning a salary of between $2.000 and $2.400 and have a car”, while “80 % (confidence) of males aged between 25 and 36 are earning a salary of between $2.000 and $2.400 and have a car”.

In a transaction database the support and confidence of a rule is calculated by following equations [14]:

$$Support, s(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N} \qquad (5)$$

$$Confidence, c(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)} \qquad (6)$$

where N is the total number of transactions. $\sigma(X \cup Y)$ and $\sigma(X)$ is the frequency of occurrence of the itemset X and $X \cup Y$ respectively, which is called support count. Support determines how often a rule is satisfied in the transaction, and confidence determines how often items in Y appear in transactions that contain X [14].

4. Related works

There has been proposed many numeric association rule mining algorithms. Each of these algorithms use a method to deal with numeric attributes.

Sirkant and Agrawal in 1996 [15], proposed an algorithm for mining association rules in large relational tables containing both quantitative and categorical attributes, by partitioning numeric attributes into a number of intervals. Fukuda et al in 1996 [16], proposed an algorithm for mining optimized association rules for numeric attributes which uses computational geometry for computing optimized ranges.

Miller and Yang in 1997 [17], proposed an algorithm for mining association rules over interval data using Birch, an adaptive clustering algorithm. Lent and Swami in 1997 [18], proposed an algorithm for the problem of clustering two-dimensional association rules in large databases. They used a geometric-based algorithm, BitOp for clustering.

The idea of using an evolutionary algorithm (EA) for mining only frequent sets was applied in Mata et al.(2002) [19]. Aumannn and Lindell in 2003 [20], proposed a statistical theory for quantitative association rules, based on the distribution of values of the quantitative attributes. Kaya and Alhajj in 2005 [2], proposed a genetic clustering method. Ke et al in 2006 [21], proposed an information theoretic approach to quantitative association rule mining. They used discretizing numeric attributes and constructing a graph based on mutual information of attributes.

David and Yanhong in 2007 [22], proposed a fuzzy weighted association rule mining algorithm by transforming numeric and categorical data into fuzzy values. Alatas and Akin in 2007 [23], proposed a multi-objective differential evolution algorithm for mining numeric association rules, later they proposed a rough particle swarm optimization algorithm and presented its applications in data mining especially numeric association rule mining problems. Their algorithm had some improvements in performance and precision in comparison with previous one in [13]. Also they proposed a genetic algorithm for automated mining of both positive and negative quantitative association rules in [24]. Qodmanan et al. in 2010 [8], proposed a multi-objective genetic algorithm for association rule mining and proposed a method without taking into account of minimum support and minimum confidence. Nasiri et al. in 2010 [25], proposed a multi-objective numeric association rule mining algorithm using simulated annealing.

Also, there are some researches which applied ACO for combinatorial optimization problems to association rule mining. The first data mining algorithm using ACO, Ant-Miner, was proposed by Parpinelli et al. in 2002 [26], as a classification algorithm, Since then, there have been proposed many data mining algorithms using ACO in field of clustering and classification. For the first time, Kuo et al in 2007 [27], proposed an algorithm for mining association rules through integration of clustering analysis and ant colony system for health insurance database in Taiwan. Atabaki in [28], used Ant System for mining association rules in a distributed system.

These algorithms mine Boolean association rules. In this paper we use $\mathrm{ACO_R}$ which is an extension of ACO to continuous domain for mining numeric association rules.

5. $\mathrm{ACO_R}$ in numeric association rule mining

5.1. Solution representation

Since, $\mathrm{ACO_R}$ uses a Gaussian function as a PDF, for dealing with continuous variables; we got this idea and used the concept of Gaussian functions in our process of generating intervals. Based on this concept, each solution member in archive is assumed to be the central point of a Gaussian normal distribution. The structure of the solution archive is slightly modified in order to store the standard deviation of each solution member in the archive to store the interval information of numeric attributes and specify whether each attribute is in the antecedent of a rule or the consequent of it.

In $\mathrm{ACO_R}$, the ants move through the archive and choose one row of it based on its associated weight (ω), which is calculated through equation (1). Then they construct a new solution by sampling the Gaussian function g of the selected solution's values of each dimension.

Each numeric attribute is considered as one dimension of the solution archive which has three parts in such a way that each complete solution is considered as a numeric association rule. The structure of each complete solution in proposed algorithm is shown in Fig. 1 where n is the number of attributes of database being mined.

		1			……..			n	
S_j	AC_j^1	s_j^1	σ_j^1				AC_j^n	s_j^n	σ_j^n

Fig.1. The structure of a complete solution in proposed algorithm

The first part of each complete solution represents the antecedent or consequent of the rule and can take values between 0 and 1like the work in [13]. If the first part of the decision variable takes values between $0.00 \leq AC_j \leq 0.33$, it means that this item will be in the antecedent of the rule and if $0.33 < AC_j \leq 0.66$, this item will be in the consequent of the rule. If $0.66 < AC_j \leq 1.00$, it means that this item will not be involved in the rule. All decision variables which are in the interval 0.00 and 0.33 will form the antecedent of the rule while decision variables which have values in the interval between 0.33 and 0.66 will form the consequent of the rule.

The second part represents the value of the solution and the third part represents the standard deviation of that solution which is used to build the intervals of numeric data. Our numeric association rule mining algorithm is described in the following.

5.2. Objective function

In ACO$_R$ algorithm, our goal is to optimize a function which is called objective function. The mined rules have to acquire high support and confidence. Our algorithm has been designed to find the intervals of each numeric attribute that conform an interesting rule, by the use of Gaussian functions, in such a way that the objective function itself is the one that decides the frequency and the length of the intervals. The objective function is shown in Eq. (7)

$$Objective = \alpha 1 \times \text{Support} + \alpha 2 \times \text{Confidence} - \alpha 3\ \text{Interestingness} - \alpha 4 \times Int \qquad (7)$$

This objective function has four parts. The first part can be considered as support of the rule that is statistical significance of an association rule. The second part can be considered as confidence value. The third part is used for number of attributes in the rule. This parameter rewards the shorter rules with a smaller number of attributes. By interestingness measure, readability, comprehensibility, and ease of understanding that are important in data mining are increased. It is known that larger rules are more likely to contain redundant or unimportant information, and this can obscure the basic components that make the rule successful and efficiently processable. The last part of the objective function is used to penalize the amplitude of the intervals that conform the itemset and rule. In this way, between two solutions that cover the same number of records and have the same number of attributes, the one whose intervals are smaller gives the best information. *Int* has been computed as shown in Eq. (8)

$$Int = \sum_{i=0}^{n} \frac{(UB_i - LB_i)}{max\ bound_i - min\ bound_i} \qquad (8)$$

Where n is the number of attributes in a rule, and *max bound* and *min bound* are Maximum and minimum allowable values of an attribute in a database, and UB_i and LB_i are upper bound value and lower bound value of an attribute in a rule. Since in our proposed algorithm the notion of Gaussian functions is used, these values can be acquired by adding a coefficient of standard deviation to the value of a solution s_j^i, in fact the upper bound and lower bound of the intervals are calculated by the equation (9):

$$UB_i = s_j^i + \alpha_5 \sigma \quad \text{and} \quad LB_i = s_j^i - \alpha_5 \sigma \qquad (9)$$

This way, the distribution of the original data is kept. α_1, α_2, α_3, α_4 and α_5 are user specified parameters and one might increase or decrease the effects of parts of objective function.

5.3. Rule generation

In ACO$_R$ pheromone table is replaced by solution archive, to keep track of the solutions and the pheromone update procedure adds a number of new solutions, each of which is generated by one ant, and eliminates the same number of bad solutions from the archive after ranking its solutions. This way, the best solutions are always on top of the solution archive, so the best solution in each execution of the ACO$_R$ can be considered as a rule, as the structure of a complete solution in proposed algorithm represents a rule. To obtain more rules the final user can execute the algorithm as many times as it is required. Also he can consider a number of top solutions from the archive as the best ranking rules of each execution.

5.4. Initialization and parameter control

First the solution archive is filled by some uniform random data, based on a range that is defined by the user in run-time; this range is usually selected in the way that covers the upper bound and lower bound of the numeric attribute value in the database. Furthermore, as this range has a vital effect on solutions of the algorithm, it can be selected in a way to focus on some particular parts of its range. Then the weights are calculated according to the equation (1) and one solution is selected probabilistically according to equation (2).

After that, the vector of standard deviations is calculated according to the equation (4), considering s_r^i as a solution member and the central point of intervals, and k as the size of the solution archive. Choosing a proper value for ξ, affects the ability of the algorithm to find the correct solutions [11]. Fig. (2) shows the effect of ξ on generated intervals that will be used by ACO$_R$ process.

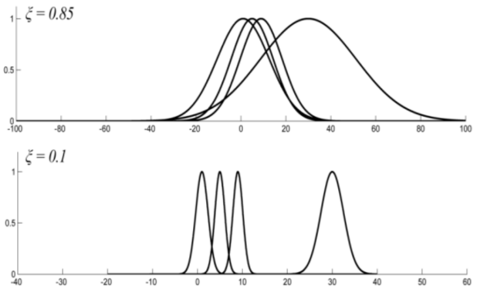

Fig. (2) The impact of ξon generated intervals

In this work, a fixed and predefined value for ξ is used, but if its value changes dynamically, it will result in a more efficient algorithm.

5. Experimental setup and results

First the proposed algorithm was evaluated in a synthetic database with a size of 1,000 records formed by 4 numeric attributes like previous works in [13] and [25]. All of the domains of values were set to [0, 100]. The values were uniformly distributed in attributes in such a way that they were grouped in pre-determined sets as shown in Table 1.These sets are completely separated from each other. This distribution of the values was completely arbitrary. Some intervals had small size and others have larger size. Support and confidence values for these sets were 25 and 100%, respectively. Other values outside these sets were distributed in such a way that no other better rules than these rules exist.

Table 1: Predetermined sets

sets
$A_1 \in [1-10] \land A_2 \in [15-30]$
$A_1 \in [15-45] \land A_3 \in [60-75]$
$A_2 \in [65-90] \land A_4 \in [15-45]$
$A_3 \in [80-100] \land A_4 \in [80-100]$

The aim of the proposed algorithm was to optimize the objective function and mining the numeric association rules which result in optimum values for the objective function. This test was performed in order to show the accuracy of the association rules found by the proposed algorithm.

In Table 2 the association rules found by the proposed algorithm, are shown. The resulted rules have high support and confidence and fall into the pre-determined sets. The algorithm is database-independent, since it does not rely upon support/confidence thresholds which are hard to choose for each database. No other rules with higher support and confidence can be found in case of specifying the minimum support and minimum confidence thresholds. Also, the intervals of the attributes are open and have a proper length because of using the Gaussian functions for building the intervals. The proposed algorithm is able to automatically find all the rules without relying upon the minimum support and the minimum confidence thresholds and the intervals of the numeric attributes are exact and accurate.

Table 2: The association rules found by the proposed algorithm

Association Rules	Support (%)	Confidence (%)
$A_3 \in$(79/44-100/02) → $A_4 \in$(79/78-100/17)	25	100
$A_2 \in$(14/92-30/13) → $A_1 \in$(0/67-10/31)	25	100
$A_1 \in$(0/95-10/08) → $A_2 \in$(14/34-30/98)	25	100
$A_1 \in$(14/94-45/01) → $A_3 \in$(59/91-75/02)	25	100
$A_4 \in$(79/45-100/48) → $A_3 \in$(79/21-100/05)	25	100
$A_3 \in$(59/66-75/65) → $A_1 \in$(14/77-45/40)	25	100
$A_2 \in$(64/12-90/62) → $A_4 \in$(14/16-45/39)	25	100
$A_4 \in$(14/40-45/41) → $A_2 \in$(64/28-90/12)	25	100

To test the efficiency of the proposed algorithm, it has been executed on noisy synthetic database like the work in [13] and [25].The noise is applied in the way that a percentage p of records exists that is not fulfilled in the pre established interval of the second item. For example, for the first set there is a percentage p of records that are distributed in the ranges [0–14] or [31–100]. This test was performed with three levels of noise to show the ability of finding correct intervals even when they contain a percentage of noise. The experimental results have been shown in Table 3. It can be seen that the algorithm is able to find the ranges of the intervals which are almost exactly adjusted to the pre-determined ones. This shows that the proposed algorithm is able to overcome certain levels of noise among the tested data.

Table 3: The results of mining the synthetic data with noise by the proposed algorithm

Noise level (%)	Association rules	Support (%)	Confidence (%)
4	$A_1 \in$(0.60-10.75) → $A_2 \in$(14.87-30.31)	24	97
	$A_2 \in$(14.56-30.55) → $A_1 \in$(0.22-10.38)	24	97
	$A_1 \in$(14.99-45.24) → $A_3 \in$(59.01-75.56)	24	97
	$A_3 \in$(59.93-75.28) → $A_1 \in$(14.13-45.23)	24	97
	$A_2 \in$(64.95-90.76) → $A_4 \in$(14.34-45.72)	24	97
	$A_4 \in$(45.57-14.86) → $A_2 \in$(64.46-90.51)	24	97
	$A_3 \in$(79.96-100.99) → $A_4 \in$(79.10-100.91)	24	97
	$A_4 \in$(79.70-100.57) → $A_3 \in$(79.57-101.44)	24	97
6	$A_1 \in$(0.87-10.42) → $A_2 \in$(14.81-30.36)	23.5	96
	$A_2 \in$(14.66-30.40) → $A_1 \in$(0.15-11.05)	23.5	96
	$A_1 \in$(15.48-47.35) → $A_3 \in$(59.42-75.27)	23.5	95
	$A_3 \in$(61.93-75.93) → $A_1 \in$(14.95-47.59)	21	96
	$A_2 \in$(64.98-86.41) → $A_4 \in$(14.91-45.17)	21	97
	$A_4 \in$(14.38-45.00) → $A_2 \in$(64.36-90.48)	23.5	95
	$A_3 \in$(79.91-100.22) → $A_4 \in$(79.40-100.84)	23.5	96
	$A_4 \in$(79.44-100.40) → $A_3 \in$(79.55-100.91)	23.5	96
8	$A_1 \in$(1.97-10.03) → $A_2 \in$(14.19-31.27)	21	95
	$A_2 \in$(14.91-31.27) → $A_1 \in$(1.69-10.02)	21	95
	$A_1 \in$(14.60-45.50) → $A_3 \in$(59.70-76.10)	23	95
	$A_3 \in$(60.92-76.14) → $A_1 \in$(14.60-45.28)	21	95
	$A_2 \in$(64.97-90.85) → $A_4 \in$(14.16-45.02)	23	95
	$A_4 \in$(16.18-45.09) → $A_2 \in$(64.28-90.50)	21	94
	$A_3 \in$(79.61-100.04) → $A_4 \in$(78.78-100.20)	23	94
	$A_4 \in$(79.88-100.88) → $A_3 \in$(79.68-100.71)	23	95

Multi objective association rules mining using ACO for continuous domains algorithm was also evaluated in five public domain databases: Basketball, Bolts, Pollution, Quake, and Sleep in order to compare with related works. These databases are available from Bilkent University

Function Approximation Repository [29]. The proposed algorithm is stochastic so, it has fluctuations in different runs. In order to get a better result, the user needs to execute several trials of the algorithm to get the result with the best solutions.

Most of the numeric association rule mining algorithms need to build the intervals of numeric attributes before mining process. They discretize numeric attributes manually and then start mining association rules from them. So the proposed algorithm is compared to five evolutionary computation-based algorithms in the literature that discretize numerical attributes and search for association rules simultaneously.

Table 4 shows the number of records and the number of numeric attributes for each database. Table 5 shows the mean number of different rules and the mean of confidence value, which can be considered as strength of the rule, found by previous works and the proposed algorithm. The experimental comparison in terms of number of rules and confidence values has been performed because the algorithms RPSOA [13], the Genetic Association Rule Mining algorithm [24] and SA [25] find directly numeric association rules without finding frequent itemsets and search for numeric intervals while mining association rules in one step. For the algorithm proposed in [24], the population size was set to 100 and it has been modified to find only positive ARs. It can be shown that, number of rules found by the proposed algorithm is more than that reported by [13], [24] and [25], in all databases. The most effective aspect of the proposed algorithm is the way it builds numeric intervals and that's why it is able to find more different rules. The results obtained in these domains seem to indicate that the proposed algorithm is competitive with the other algorithms in terms of confidence values and number of rules.

Table 4: The number of records and numeric attributes in each database

Sleep		Quake	
8	Number of attributes	4	Number of attributes
57	Number of records	2178	Number of records
%2	Missing values	0	Missing values
Pollution		Bolts	
16	Number of attributes	8	Number of attributes
60	Number of records	40	Number of records
0	Missing values	0	Missing values
Basketball			
5	Number of attributes		
96	Number of records		
0	Missing values		

Table 5: The mean number of different rules and confidence value compared with other works

Database	Number of rules			
	ACO$_R$	SA	RPSOA	GA
Basketball	37	12	33.8	33.8
Bolts	42	3.1	39.0	39.0
Pollution	49	3.5	41.2	41.2
Quake	58	7.5	43.8	43.8
Sleep	41	3.1	32.8	32.8
Database	Confidence (%)			
	ACO$_R$	SA	RPSOA	GA
Basketball	78	93	60±2.8	60±1.2
Bolts	89	80	60±2.0	65±1.9
Pollution	78	75	66±4.7	68±4.8
Quake	85	73	63±2.8	62±5.1
Sleep	84	78	64±2.8	64±2.3

Table 6 shows the comparison of obtained results from the proposed algorithm, RPSOA [13], the Genetic Association Rule Mining algorithm proposed in [24], the work proposed in [19] and SA [25] in terms of support which refers to the usefulness of the rule, and size which refers to number of attributes in the rule. The GAR algorithm uses an EA for mining only frequent itemsets. That is why; comparisons about the values according to the rules cannot be made [13]. The value of the column "Support (%)" indicates the mean of support, while the value of the column "Size" shows the mean number of attributes contained in the rules.

Table 6: The comparison of obtained results from the proposed algorithm in terms of support and size

Database	Support (%)				
	ACO$_R$	SA	RPSOA	GA	GAR
Basketball	45	42	36.44	32.21	36.69
Bolts	48	41	28.48	27.04	25.97
Pollution	54	41	43.85	38.95	46.55
Quake	60	45	38.74	36.96	38.65
Sleep	53	46	36.52	37.25	35.91
Database	Size				
	ACO$_R$	SA	RPSOA	GA	GAR
Basketball	2.7	2.58	3.21	3.21	3.38
Bolts	3.88	2.71	5.14	5.14	5.29
Pollution	6.71	2.3	6.46	6.21	7.32
Quake	2.32	2.02	2.22	2.10	2.33
Sleep	4.1	2.01	4.19	4.19	4.21

Multi objective association rule mining via ACO$_R$ algorithm has found rules with high values of support in all databases. The size values obtained from this algorithm are smaller than the values obtained from the

GAR in five out of five databases and they are smaller than the values obtained from the GA algorithm proposed in [19] in three out of five databases. The size of the intervals is based on the attributes' standard deviations and is so flexible and controllable. There is no need to refine their size by decreasing the length of them.

Another point is that, by tuning up the parameters of the algorithm many different rules can be discovered which have lower support and confidence or have 100% confidence and lower support. Using a method which can tune the parameter **ξ** will result in even more different rules.

6. Conclusion

Ant colony optimization for continuous domain (ACO_R) is a new metaheuristic approach. This study proposed a new algorithm that uses the notion of Gaussian functions and the modification of the solution archive of ACO_R in order to build numeric intervals and search for association rules in one step without specifying minimum support and minimum confidence.

The lower bound and upper bound values are introduced by adding and subtracting a coefficient of a partial solution's standard deviation to/from its value. The proposed algorithm has been used in data mining within databases that can take numeric attributes and has given satisfactory results in its first applications. This algorithm seems to provide useful extensions for practical applications specially while using feature selection before applying the algorithm, since ant algorithms cannot distinguish variable correlations. But still despite of this fact, the ACO_R algorithm is able to find accurate and exact rules with reasonable interval lengths for numeric attributes because of the way of building intervals using Gaussian functions. Proposing an efficient way of tuning the algorithm's parameters which would result in better solutions and another ant algorithm which can handle both numeric and categorical attributes may be presented as further works.

References

[1] Chen C.-H., Hong T.-P., and Tseng V.S., *A Cluster-Based Fuzzy-Genetic Mining Approach for Association Rules and Membership Functions*, IEEE International Conference on Fuzzy Systems, pp. 1411 - 1416, 2006.

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

[3] Tsay Y. J., Chiang J. Y., *CBAR: an efficient method for mining association rules*, Knowledge] Based Systems, Vol. 18, No. 2-3, pp. 99-105, 2005.

[4] Agrawal R., Imielinski T., and Swami, A., *Mining association rules between sets of items in large databases,* In proceedings of ACM SIGMOD conference on management of data, pp. 207-206, 1993.

[5] Agrawal R., Sirkant R., *Fast algorithms for mining association rules,* In proceedings of the 20th international conference on very large databases, Santiago, Chile, 1994.

[6] Houtsma A., Swami M., *Set-oriented mining of association rules,* Research Report, 1993.

[7] Lin D. I., Kedem Z. M., *Pincer-search: An efficient algorithm for discovering the maximal frequent set,* In Proceedings of sixth European conference on extending database technology, 1998.

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

[9] Martens, D., Baesens, B., and Fawcett, T., *Editorial Survey: Swarm Intelligence for Data Mining,* Machine Learning, Vol. 82, No. 1, pp. 1-42, 2010.

[10] Socha K., *ACO for Continuous and Mixed-Variable Optimization,* Ant Colony Optimization and Swarm Intelligence, Computer Science, Vol. 3172, pp. 53-61, 2004.

[11] Socha K., *Ant Colony Optimization For Continuous and Mixed-Variable Domains*, Ph.D. Thesis, Universit´e Libre de Bruxelles, Brussels, Belgium, 2008.

[12] Rungswang A., Tangpong P., Laohawee T., and Khampachua T., *Novel Query Expansion Technique using Apriori Algorithm.* In proceedings of TREC' 1999.

[13] Alatas B., Erhan A., *Rough particle swarm optimization and its applications in data mining,* Soft Computing – A Fusion of Foundations, Methodologies and Applications, Vol.12, No. 12, pp. 1205-1218, 2008.

[14] Tan P.-N., Steinbach M., and Kumar V., *Introduction to Data Mining*, Pearson International Edition Pearson Addison Wesley, 2006.

[15] Srikant R., Agrawal R., *Mining quantitative association rules in large relational tables*, In: Proceedings of ACM SIGMOD international conference on Management of data, Vol. 25, No. 2, pp. 1-12, 1996.

[16]] Fukuda T., Yasuhiko M., Sinichi M., Tokuyama T. Mining *optimized association rules for numeric attributes*. In: Proceedings of ACM SIGACT-SIGMOD-SIGART symposium on principles of database systems. New York, pp. 182–191, 1996.

[17] Miller R.J., Yang Y., *Association rules over interval data*, In: Proceedings of ACM SIGMOD international conference on management of data, vol. 29, No. 2, pp. 452–461, 1997.

[18] Lent B., Swami A., and Widom J., *Clustering association rules*, In: Proceedings of IEEE international conference on data engineering, pp. 220-231, 2002.

[19] Mata J, Alvarez JL, Riquelme JC, *Discovering numeric association rules via evolutionary algorithm*. In: Sixth Pacific–Asia conference on knowledge discovery and data mining PAKDD-02 (LNAI), Taiwan 2336, pp 40–51, 2002.

[20] Aumann Y., Lindell Y., *A statistical theory for quantitative association rules*, Journal of Intelligent Information Systems, Vol. 20, No. 3, pp.255–283, 2003.

[21] Ke K., Cheng J., and Ng W., *An information-theoretic approach to quantitative association rule mining*, Journal Knowledge and Information Systems, Vol. 16, No. 2, pp. 112-114, 2008.

[22] David L. O., Yanhong L., *Mining Fuzzy Weighted Association Rules,* In: 40th Annual Hawaii international conference on system, sciences HICSS, pp 53–62, 2007.

[23] Alatas B., Akin E., Karci A., *MODENAR: multi-objective differential evolution algorithm for mining numeric association rules*, Appl Soft Comput. doi:10.1016/j.asoc.2007.05.003, 2007

[24] Alatas B, Akin E, *An efficient genetic algorithm for automated mining of both positive and negative quantitative association rules*. Soft Comput 10(3):230–237, 2006.

[25] Nasiri M., Taghavi L., Minaei Bidgoli B., *Multi-Objective Rule Mining using Simulated Annealing Algorithm*, Journal of Convergence Information Technology, Vol. 5, No. 1, 2010.

[26] Parpinelli R.S., Lopes H.S., and Freitas A.A., *Data mining with an ant colony optimization algorithm,* Evolutionary Computation, IEEE Transactions, pp. 321 – 332, Vol. 6, No. 4, 2002.

[27] Kuo R J., Shih C.W., *Association rule mining through the ant colony system for national health insurance research database in Taiwan.*. In Journal of Computers and Mathematics with Applications, pp. 1303-1318, 2007.

[28] Atabaki G., Kangavari M., *Mining association rules in Distributed Environment through Ant Colony Optimization Algorithm,* M.Sc thesis (in Persian), Iran University of Science and Technology, 2009.

[29] Guvenir H.A., Uysal I., Bilkent University Function Approximation Repository. http://funapp.cs.bilkent.edu.tr, 2000.