

A Survey of Multi-objective Evolutionary Algorithms for Association Rule Mining

Munmun Kalita¹ and Chitvan Gupta²

¹M.Tech Student, Department of CSE, Noida Institute of Engineering And Technology, Greater Noida, UP

²Department of CSE, Noida Institute of Engineering And Technology, Greater Noida, UP

E-mail: ¹kalita.munmun75@gmail.com, ²chitvangupta@gmail.com

Abstract—The main aim of data mining is to pull out knowledge from an existing dataset and transform it into a flexible structure. Association rule mining is one of the most important tasks of data mining intended towards decision support. It is the process of finding some relations among the attribute values of a huge database. Evolutionary algorithm has found its strong base in mining Association Rules. Many real life data mining problems involve multiple conflicting measures of performance, or objectives, which need to be optimized simultaneously. Under this context, multi objective evolutionary algorithms are gradually finding more and more applications in the domain of data mining since the beginning of the last decade. Many researchers have proposed multi-objective evolutionary algorithms for mining interesting rules from dataset. This paper provides the major advancements in the approaches for association rule mining using evolutionary algorithms.

1. INTRODUCTION

In recent years, data mining has attracted a great deal of attention in the information industry due to the wide availability of huge amounts of data and the imminent need of extracting useful information and knowledge from those. The process of discovering interesting and unexpected rules from large data sets is known as association rule mining. These rules are the relationships that are found between items of a database. An association rule is an *implication* or *if-then-rule* which is supported by data. Mining of association rules is a field of data mining that has received a lot of attention in recent years [6]. Mining of numeric association rules can be characterized by the presence of more than one objective which is conflicting in nature. These objectives may be high support and confidence values, interestingness, comprehensibility, narrow intervals for numeric attributes. That is why this problem is a multi objective optimization problem. [4]

Most of the association rule algorithms are based on methods proposed by Agrawal, Imielinski, and Swami [1] and Agrawal and Srikant [2], Apriori [1], SETM [1], AIS [1] etc.[7]. However, these algorithms have their limitations. Multi-objective Evolutionary algorithm is used in mining association rule to remove some of the limitations of the existing approaches [3]. MOEA is relatively simple, easy to implement

and easy to use. Furthermore, it follows a database-independent approach which does not rely upon the minimum support and the minimum confidence thresholds which are hard to determine for each database. [4]

The rest of this paper is organized as follows. In Section 2 an overview of Association rule mining (ARM) is provided. Section 3 discusses the multi-objective nature of association rules mining problems. Section 4 gives an overview of Multiobjective evolutionary algorithms (MOEAs). Section 5 covers a comparative study of different MOEAs for ARM. Finally, Section 6 includes the future scope and conclusion.

2. ASSOCIATION RULE MINING (ARM).

Principle of association rule mining (ARM) lies in the market basket or transaction data analysis. The major aim of ARM is to find the set of all subsets of items or attributes that frequently occur in many database records or transactions, and additionally, to extract rules on how a subset of items influences the presence of another subset. ARM algorithms discover high-level prediction rules in the form: IF the condition of the values of the predicting attributes are true, THEN predict values for some goal attributes. The task of mining association rules over market basket data was first introduced by Agrawal et al. [1].

Let $I = \{i_1, i_2, i_3, \dots, i_m\}$ be the set of database items and $T = \{t_1, t_2, \dots, t_m\}$ be the set of transactions in the database, D , with each transaction t_i having a unique identifier and containing a set of items, called an itemset. An association rule is a conditional implication among itemsets, $X \rightarrow Y$, where X and Y are itemsets and $X \cap Y = \emptyset$. An itemset can be a single item or a set of items. An itemset with k items is called a k -itemset. A subset of k elements is called a k -subset.

An association rule (AR) is called frequent if its support exceeds a minimum value *min sup*. The confidence of a rule $X \Rightarrow Y$ in T denotes the percentage of the transactions in T containing X that also contains Y . It is taken to be the conditional probability $P(X|Y)$.

That is, $confidence(X \Rightarrow Y, T) = \frac{support(X \cup Y, T)}{support(X, T)}$

A rule is called confident if its confidence value exceeds a threshold min_conf . The ARM problem can be defined as follows. Find the set of all rules R of the form $X \Rightarrow Y$ such that $R = \{X \Rightarrow Y | X, Y \subseteq I, X \cap Y = \emptyset, X \cup Y \subseteq f(T, min_sup)\}$,

$confidence(X \Rightarrow Y, T) > min_conf\}$.

Generally, the ARM process consists of the following two steps

- 1) Find all frequent itemsets.
- 2) Generate strong ARs from the frequent itemsets.

The number of itemsets grows exponentially with the number of items $|I|$. A commonly used algorithm for generating frequent itemsets is the *apriori* algorithm.[9,7,6]

3. MULTI-OBJECTIVE OPTIMIZATION AND RULE MINING PROBLEMS.

Most of the data mining problems can be thought of as optimization problems, where the aim is to evolve a candidate model that optimizes certain performance criteria. However, the majority of data mining problems have multiple criteria to be optimized. A rule mining problem may optimize several rule objectives such as support count, confidence, comprehensibility, interestingness, J-measure, entropy and lift at the same time. Hence, association rule mining problems are multi-objective in nature. Therefore, it is natural to pose rule mining problems as multiobjective ones. For this reason, over the past decade, several researchers have applied Multi Objective Evolutionary Algorithms (MOEA) for different data mining problems. An MOEA provides a set of nondominated solutions, which the user can compare (it is important to keep in mind that the set of nondominated solutions represents the best possible tradeoffs among the objectives). Then, a single solution from this set can be chosen, based on the user's preferences. In the problem of association rule mining, all the nondominated solutions, representing rules are considered the final solution set. Due to the above reasons, MOEAs have been popularly used for data mining problems. In this paper, a number of different MOEAs techniques applied to ARM mainly focusing on encoding techniques, objective functions, evolutionary operators, and final solution selection strategies.[5,8]

4. MULTIOBJECTIVE EVOLUTIONARY ALGORITHMS (MOEAS).

MOEAs have evolved over several years. First generation MOEAs were traditional aggregating approaches, Pareto-based approaches came after that and more recently indicator-based algorithms are gaining popularity.

In the aggregating approaches, multiple objective functions are combined into a single scalar value using weights, and the resulting single-objective function is then optimized using

conventional evolutionary algorithms. In population based non-Pareto approaches such as the vector evaluated genetic algorithm, a special selection operator is used and a number of subpopulations are generated by applying proportional selection based on each objective function in turn. Among the Pareto-based approaches, multiple objective GA, niched Pareto GA (NPGA), and nondominated sorting GA (NSGA) are the most representative nonelitist MOEAs. In the late 1990s, a number of elitist models of Pareto-based multiobjective evolutionary algorithms were proposed. The most representative elitist MOEAs include strength Pareto evolutionary algorithm (SPEA) and SPEA2, Pareto archived evolutionary strategy (PAES), Pareto envelope-based selection algorithm (PESA) and PESA-II, and nondominated sorting genetic algorithm-II (NSGA-II). Most of the recent applications of MOEAs for data mining problems have used one of these Pareto-based elitist approaches as their underlying optimization strategy. The indicator-based evolutionary algorithm is intended to be adapted to the user's preferences by formalizing such preferences in terms of continuous generalizations of the dominance relation. [8]

5. MULTIOBJECTIVE EVOLUTIONARY ALGORITHMS FOR ARM.

Here, MOEAs used for association rule mining are surveyed. We review many of these approaches with a focus on chromosome representation, objective functions, evolutionary operators, and methods for obtaining the final solution from the non-dominated set. Figure. 1 shows the different MOEAs-based ARM along with the corresponding references.

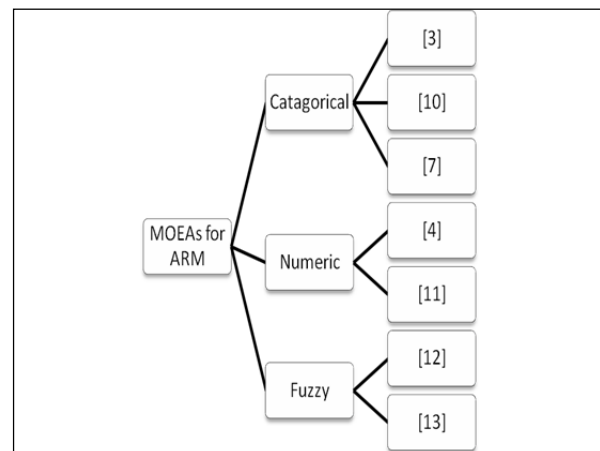


Fig. 1: MOEAs for association rule mining.

In the past decade, several MOEAs have been proposed for ARM. These techniques can broadly be classified into three categories, namely categorical association rules, numeric association rules, and fuzzy association rules.[9]

5.1 Categorical Association Rules.

Categorical association rules are generated from a binary or categorical dataset. In a binary dataset, a rule like $ABC \Rightarrow DE$ can be interpreted as follows: if items A , B , and C are purchased, then items D and E are also purchased. Thus, these rules do not say anything about the number of items that are to be purchased; they simply imply the presence or absence of items. For categorical data, if some item has multiple categorical values, then each attribute-value pair is treated as a separate item. In this way the dataset is converted into a binary dataset.[9]

5.1.1 Underlying MOEAs.

Different standard and nonstandard MOEAs have been used in various works on categorical ARM. We call a MOEA non-standard if it does not follow any of the standard MOEA approaches directly, but uses instead some combination of operators. In [3], a multiobjective GA (MOGA) is used. In [10] and [7], some non-standard MOEAs are used for the ARM problem.[9]

5.1.2 Chromosome Representation.

There are mainly two chromosome representation techniques for categorical ARM. In the first approach (Pittsburgh approach), a set of possible association rules are encoded in each chromosome. This approach is more suitable for classification rule mining, where the objective is to identify a good set of rules. However, in ARM, the objective is to find a set of rules each of which is good. Therefore, for this case, the Michigan approach, in which each chromosome represents exactly one rule, is more suitable [3]. Most of the MOEA-based categorical ARM techniques use this chromosome representation. In an early work [3], the authors adopted the Michigan approach as follows: each chromosome has length $2k$, where k is the number of items. The chromosomes are binary strings where each attribute is given two bits. If these two bits are 00 or 11, then the attribute appears in the antecedent or consequent parts of the rule, respectively; otherwise, the attribute is absent from the rule. In a similar approach, the presence of an attribute in the antecedent and consequent part are represented by bits 10 and 01, whereas other bit combinations represent the absence of the attribute from the rule. The above encoding schemes [3] can only be adopted for binary datasets, that is, when an item is either present or absent in a transaction. If someone wants to use this encoding for more general categorical data, where an item may be present in a transaction with certain value (a categorical state), the dataset will first need to be transformed into a binary one by considering each attribute-value pair as an item. The main disadvantage of using a binary encoding scheme is that it gives rise to a large chromosome length when the number of attributes is large, since at least two bits are needed for each attribute. An integer encoding may come handy in this respect. Such an integer encoding scheme has been proposed in association rule mining using multiobjective

genetic algorithm (ARMMGA) [7], where the chromosomes encode the index of the attributes. A chromosome encoding a k -rule, k being the total number of items in the antecedent and the consequent, has $k + 1$ genes. The first gene position indicates the separating position of the chromosome where the antecedent and the consequent attributes are separated. For example, if A_i represents the i th item, then the chromosome $\{3 | 2 5 4 1 3\}$ represents the rule $A_2A_5A_4 \Rightarrow A_1A_3$. This representation significantly reduces the length of the chromosome, but not effectively the search space, because now for each position, a large number of alternative indices are to be searched. Moreover, this representation scheme gives rise to a variable chromosome length, thus requiring a specialized crossover operator. Also, there remains a possibility of finding duplicate indices in a chromosome after crossover/mutation, which must be taken care of during the evolutionary process.[9]

5.1.3 Objective Functions.

Although support and confidence are two popular objectives that are to be maximized, there are several other metrics to measure the interestingness of association rules. These metrics, which have been used by different algorithms for optimization in a multiobjective framework, include coverage, lift, comprehensibility, cosine, prevalence, recall, Laplace, conviction, surprise, Jaccard, J-measure, and so on. In [3], the rule mining problem has been modeled as a three-objective optimization problem where confidence, comprehensibility, and interestingness have been optimized simultaneously. They defined the comprehensibility of a rule as $\log(1 + |C|)/\log(1 + |A \cup C|)$, where $|C|$ and $|A \cup C|$ denote the number of attributes in the consequent part and total rule, respectively. They considered that the lower value of comprehensibility, that is, less number of attributes in the consequent of the rule, leads to better understandability of the rule. The interestingness measure, on the other hand, is defined as a product of three probabilities, namely, the probability of generating the rule given the antecedent (ratio of the support of the rule to the support of the antecedent), the probability of generating the rule given the consequent (ratio of the support of the rule to the support of the consequent), and the probability of generating the rule given both antecedent and consequent (ratio of the support of the rule to the total number of transactions). A rule becomes more interesting if it has a high interestingness value. In [10], five objective functions, that is, support, confidence, J-measure, interest, and surprise have been simultaneously optimized. They found five different groups of correlated measures. To make the objective functions contradictory and uncorrelated, they selected these five measures from five different groups. In [7], the classical measures (support and confidence of the rules) are simultaneously optimized. Thus, it is apparent from the above discussion that different sets of rule-interestingness measures have been chosen by various authors as their objective functions. [9]

5.1.4 Evolutionary Operators.

When binary encoding has been adopted, standard crossover and mutation operators have been used. For example, in [3], multipoint crossover and bit-flip mutation have been used. In [10], although the authors did not explain the encoding strategy explicitly, from the description of the evolutionary operators, it appears that they used an approach in which the categorical values of the attributes participating in the rule are encoded. Here, the authors used value exchange and insertion crossover operators. If two parents have some common attributes in the antecedent part, then a value exchange crossover is performed by exchanging the categorical values of one of the common attributes. When the parents do not have any common attribute, then one random attribute selected from one parent is inserted into the other with a probability that is inversely proportional to the length of the latter chromosome. Four mutation operators are applied with equal probabilities. A value mutation randomly replaces a chosen categorical value with another random value from the same domain. An attribute mutation randomly replaces an attribute with another one. An insertion mutation inserts a new attribute-value pair, and a deletion mutation deletes a randomly chosen attribute-value pair. In [7], where integer encoding of the attributes is used, an order-1 crossover strategy is adopted. In this strategy, first a segment is chosen from two parent chromosomes and these are copied to the two offspring. Then, starting from the right side of the segment, the values of the genes that do not exist in the selected segment of the first parent, are copied to the first offspring. The same procedure is repeated for the second offspring as well. The mutation operator replaces a chosen item from the chromosome with a random item not present in the chromosome.[9]

5.1.5 Obtaining the Final Solution.

All the works for categorical rule mining using MOEAs that have been discussed in this paper use a Michigan type of encoding, where each chromosome encodes one association rule. Hence, the final generation produces a set of non-dominated solutions each of which are given to the user as the association rules generated from the input dataset. Thus, in this case, there is no specific need of selecting a single solution from the non-dominated front.[9]

5.2 Numeric Association Rules.

For datasets having continuous attribute domains, the ARM algorithms designed for categorical attributes do not work well. This is because such algorithms need categorization of the continuous attributes. Hence, the results of the ARM algorithms depend a lot on the categorization technique adopted. To overcome this limitation, many numeric/quantitative ARM algorithms have been proposed and some of them adopted a multiobjective optimization approach. A quantitative association rule is represented as [4], [11]

$(l1 \leq A1 \leq h1) \wedge (l2 \leq A2 \leq h2) \Rightarrow (l3 \leq A3 \leq h3)$. Here A_i represents the i th attribute. l_i and h_i represent the lower and upper bound of the attribute values, respectively. Thus, $[l_i, h_i]$ defines an interval of values for the attribute A_i . Here, we discuss two different works on quantitative ARM.[9]

5.2.1 Underlying MOEAs.

In this section, we review a two EAs. First one is multiobjective differential evolution based numeric association rule mining algorithm (MODENAR) [4].

TABLE I: Comparison of Different MOEAs for Association Rule Mining.

Algorithm	MOO tool	Type	Encoding	Objective Functions	Evolutionary operator
Kaya & Alhadj, 2003[12]	SPEA Variant	Fuzzy	Real Valued (membership functions)	Number of large itemsets, time taken to find all large itemsets	Multi point crossover, standard real value mutation
Ghosh & Nath, 2004[3]	MOGA	Categorical	Michigan approach	confidence, comprehensibility and interestingness values	multi-point crossover and mutation
Alatas et al 2007 [4]	MODENAR	Numeric	Mixed(Integer+real), Michigan	Support, Confidence, Comprehensibility, amplitude of interval	Multiobjective DE, rounding, repairing and filtrating operator
Khabzaoui, et al, 2008[10]	Enumerative Procedure	Categorical	Not mentioned	Support, Confidence, Jmeasure, Interest and Surprise	Value Exchange crossover, insertion crossover, value/attribute mutation, insertion/deletion mutation
Qodmanan et al, 2011[7]	ARMMGA	Categorical	Integer(Michigan)	Support, confidence	Order, crossover,, random replacement mutation
Mathews et al, 2011[13]	NSGA-II	Fuzzy	Mixed(Integer+real), Michigan	Temporal support, temporal confidence, Fuzzy support, membership function width	Modified uniform crossover, random change mutation

Troncoso et al, 2015[11]	MOQAR	Numeric	Mixed(Integer+real), Michigan	Support of the rule, support of the antecedent, support of the consequent, confidence, leverage, accuracy, lift, gain, certainty factor, amplitude and finally, number of attributes of the rule	Crossover, mutation
--------------------------	-------	---------	-------------------------------	--	---------------------

In this case, a multiobjective differential evolution (MODE) algorithm is used as the underlying optimization framework. In another work, an NSGA-II-based quantitative association rule mining algorithm (MOQAR) is proposed [11]. [9]

5.2.2 Chromosome Representation.

The chromosomes representing numeric or quantitative association rules need to encode the lower and upper bounds of the intervals of the attributes participating in a rule. In [4], where the MODENAR algorithm has been proposed, the following encoding technique has been adopted for the chromosomes. They used chromosomes where each attribute has three components. The first component indicates whether the attribute is present or absent in the rule, and if present, in which part (antecedent or consequent) in the rule it is. The second and third components indicate the lower and upper bounds of the ranges of the attribute. The first component can have integer values 0, 1, or 2, which indicate the presence of the attribute in the antecedent of the rule, the presence of the attribute in the consequent of the rule, and the absence of the attribute from the rule, respectively. The second and third components can take real values from the corresponding attribute ranges. It is to be noted that as MODENAR uses differential evolution as an optimizer and works on real-valued chromosomes, the authors used a round-off operator to handle the integer part of the chromosome. A similar encoding scheme is adopted in MOQAR. [9]

5.2.3 Objective Functions.

MODENAR optimizes four criteria of the rules [4]: support, confidence, comprehensibility, and amplitude of the intervals that make up the itemset and the rule. Comprehensibility is used to bias the search process toward shorter rules, under the assumption that shorter rules provide more non-redundant information. They also proposed that the amplitude of the intervals must be smaller for interesting rules, but the rationale for this is not explained. In MOQAR [11], these objective

functions are simultaneously optimized: Support of the rule, support of the antecedent, support of the consequent, confidence, leverage, accuracy, lift, gain, certainty factor, amplitude and finally, number of attributes of the rule. [9]

5.2.4 Evolutionary Operators.

MODENAR [4] used the standard version of the crossover and mutation operators adopted by the version of differential evolution called DE/rand/1. Additionally, a rounding operator is used to roundoff the first part of the attribute that requires an integer (0,1,2) for computing the objective function values.

In both [4] and [11], during mutation/crossover, it may happen that the lower bound becomes larger than the upper bound, or they go outside the bounds. For this, some repairing operators are also adopted to make the chromosome a valid one. [9]

5.2.5 Obtaining Final Solution.

Both MODENAR and MOQAR use a Michigan approach of rule mining by encoding one rule in one chromosome. Thus, the final nondominated set gives a set of numeric rules. Thus, there is no need to select any particular solution from the final nondominated set. All the solutions will serve as the final selected rule set. [9]

5.3 Fuzzy Association Rules.

One of the major problems of mining numeric association rules is that these algorithms deal with sharp boundaries between consecutive intervals. Thus, they cannot represent smooth changes from one interval to another, which can be easily handled by fuzzy association rules. A number of MOEA-based fuzzy ARM techniques have been developed in the past decade. Here, we describe several of these algorithms and discuss different approaches that incorporate them. The general form of a fuzzy association rule is as [12] *If* $X = \{x_1, x_2, \dots, x_p\}$ *is* $A = \{f_1, f_2, \dots, f_p\}$ *Then* $Y = \{y_1, y_2, \dots, y_q\}$ *is* $B = \{g_1, g_2, \dots, g_q\}$.

Here X and Y represent two sets of attributes, and $X \cap Y = \varphi$. A and B represent the fuzzy sets (linguistic values) of the corresponding attributes in X and Y , respectively. Therefore, if a rule is encoded in a chromosome, both the attributes and their linguistic values should be encoded in it. A number of studies have been done on the application of MOEAs for fuzzy association rule mining. [9]

5.3.1 Underlying MOEAs.

Different MOEAs have been employed in various works on fuzzy ARM. Kaya *et al.* [12] used a variant of SPEA for fuzzy rule mining. Mathews *et al.*, [13] used a NSGA-II. [9]

5.3.2 Chromosome Representation.

There are two categories of chromosome representations for fuzzy ARM. In the first approach, a chromosome represents a set of fuzzy clusters corresponding to each attribute. The objective is to find a suitable set of fuzzy clusters that partition

the range of values in each attribute domain. This approach is adopted in a series of works done by Kaya *et al.* in [12]. In these works, each chromosome represents the base values of a variable number of membership functions representing the fuzzy sets for each quantitative attribute. Standard triangular membership functions are used to represent the fuzzy sets. Real-valued representation of the chromosomes is used for this purpose. Here, a chromosome does not represent association rules. It represents a suitable fuzzy clustering of the attribute domains. The evolved fuzzy membership functions are then used as the linguistic values of the corresponding attributes. Fuzzy association rules are mined using standard algorithms based on minimum support and minimum confidence criteria. [9]

5.3.3 Objective Functions.

In the works of Kaya *et al.* [12], the authors optimize two criteria, that is, number of large itemsets and time spent to obtain the large itemsets. Thus, here the objective is to evolve a possible fuzzy clustering of the numeric attributes that maximizes the number of large itemsets while minimizing the time required to obtain all large itemsets given the clustering. After optimizing the clustering, the authors then use the membership functions as the linguistic values for the fuzzy association rules extracted based on minimum support and minimum confidence criteria. [9]

5.3.4 Evolutionary Operators.

[12] have used standard multipoint crossover operations. The mutation operator is used to slightly change the center of the fuzzy set being mutated. It is to be noted that when mutation takes place at the center of a fuzzy membership function, it may disrupt the order of the resulting fuzzy membership functions. Hence, these fuzzy membership functions need rearrangement according to their center values after the mutation. [9]

5.3.5 Obtaining the Final Solution.

As in [12], a chromosome encodes a possible fuzzy clustering of the attribute values. It is necessary to select a suitable solution from the final non-dominated set, based on which of the final association rules are extracted. In [13], the authors used a Michigan type of encoding of temporal fuzzy association rules. Therefore, all the rules encoded in the final non-dominated set are considered as extracted rules. There is no specific need of choosing any particular solution from the non-dominated set. [9]

6. FUTURE SCOPE AND CONCLUSION.

MOEAs are being applied in association rule mining tasks over the past decade, still some important future research issues are there. A systematic comparison to guide new users in choosing most suitable method for his/her application would be valuable as it is still missing in the literature. Computational efficiency of MOEAs used in association rule mining is another promising research area.

In this paper, we surveyed several MOEAs used for association rule mining. Importance has been given on the chromosome representation, objective functions, evolutionary operators, and final solution selection. Moreover, a comparison among different methods in each category is provided. [9]

REFERENCES

- [1] Agrawal, R., Imielinski, T., & Swami, A. (1993). "Mining association rules between sets of items in large databases." In Proceedings of ACM SIGMOD conference on management of data (pp. 207–216)
- [2] Agrawal, R., & Srikant, R. (1994). "Fast algorithms for mining association rules". In Proceedings of the 20th international conference on very large databases, Santiago, Chile.
- [3] A. Ghosh and B. Nath, "Multi-objective rule mining using genetic algorithms," *Inf. Sci.*, vol. 163, nos. 1–3, pp. 123–133, Jun. 2004.
- [4] Alatas B, Akin E, Karci A (2008) "MODENAR: multi-objective differential evolution algorithm for mining numeric association rules". *Appl Soft Comput* 8(1):646–656
- [5] Peter P. Wakabi-Waiswa, Venansius Baryamureeba, "Extraction of interesting association rules using genetic algorithms, *International Journal of Computing and ICT Research*", Vol. 2 No. 1, June 2008.
- [6] Soumadip Ghosh, Sushanta Biswas, Debasree Sarkar, Partha Pratim Sarkar, "International Journal of Artificial Intelligence & Applications (IJAIA)", Vol.1, No.4, October 2010.
- [7] Hamid Reza Qodmanan, Mahdi Nasiri, Behrouz Minaei-Bidgoli, "Multi objective association rule mining with genetic algorithm without specifying minimum support and minimum confidence", 2010 Elsevier Ltd.
- [8] Anirban Mukhopadhyay, Ujjwal Maulik, Sanghamitra Bandyopadhyay, and Carlos Artemio Coello Coello, "Survey of Multiobjective Evolutionary Algorithms for Data Mining: Part I", *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, VOL. 18, NO. 1, FEBRUARY 2014.
- [9] Anirban Mukhopadhyay, Ujjwal Maulik, Sanghamitra Bandyopadhyay, and Carlos Artemio Coello Coello, "Survey of Multiobjective Evolutionary Algorithms for Data Mining: Part II", *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, VOL. 18, NO. 1, FEBRUARY 2014.
- [10] M. Khabzaoui, C. Dhaenens, and E.-G. Talbi, "Combining evolutionary algorithms and exact approaches for multiobjective knowledge discovery", *RAIRO—Oper. Res.*, vol. 42, no. 1, pp. 69–83, 2008.
- [11] M. Martínez-Ballesteros, A. Troncoso, F. Martínez-Álvarez, J. C. Riquelme1, "Improving a multi-objective evolutionary algorithm to discover quantitative association rules", Springer-Verlag London 2015
- [12] M. Kaya and R. Alhajj, "Facilitating fuzzy association rules mining by using multiobjective genetic algorithms for automated clustering", in *Proc. 3rd IEEE ICDM*, 2003, pp. 561–564.
- [13] S. G. Matthews, M. A. Gongora, and A. A. Hopgood, "Evolving temporal fuzzy association rules from quantitative data with a multiobjective evolutionary algorithm," in *Proc. Int. Conf. HAIS—Vol. Part I*, 2011, pp. 198–205.