

# Multi-objective Bat Algorithm for Mining Interesting Association Rules

Kamel Eddine Heraguemi<sup>1</sup>(✉), Nadjat Kamel<sup>1</sup>, and Habiba Drias<sup>2</sup>

<sup>1</sup> Depart. Computer Science, Fac-Sciences, Univ-Setif, Setif, Algeria  
k.heragmi@univ-setif.dz, nkamel@usthb.dz

<sup>2</sup> Depart. Computer Science, USTHB, Algiers, Algeria  
hdrias@usthb.dz

**Abstract.** Association rule mining problem attracts the attention of researchers inasmuch to its importance and applications in our world with the fast growth of the stored data. Association rule mining process is computationally very expensive because rules number grows exponentially as items number in the database increases. However, Association rule mining is more complex when we introduce the quality criteria and usefulness to the user. This paper deals with association rule mining issue in which we propose Multi-Objective Bat algorithm for association rules mining Known as MOB-ARM. With the aim of extract more useful and understandable rules. We introduce four quality measures of association rules: Support, Confidence, Comprehensibility, and Interestingness in two objective functions considered for maximization. A series of experiments are carried out on several well-known benchmarks in association rule mining field and the performance of our proposal are evaluated and compared with those of other recently published methods including mono-objective and multi-objective approaches. The outcomes show a clear superiority of our proposal in-face-of mono objective methods in terms generated rules number and rule quality. Also, The analysis also shows a competitive outcomes in terms of quality against multi-objective optimization methods.

**Keywords:** Association rules mining · ARM · Bat algorithm · Multi-objective optimization · Support · Confidence · Comprehensibility · Interestingness

## 1 Introduction

Association rule mining [1] (ARM) is one of the most active, attractive and useful research area in knowledge discovery. Basically, it finds practical and interesting relations between items in huge transactional databases to help for decision making. The extracted relationships can be represented by *IF-THEN* statement, *IF <some conditions are satisfied> THEN <some values of other attributes>*. The conditions in *IF* statement called *Antecedent* and those within the *THEN* clause are *Consequence*. ARM applications varies from market basket analysis

as first innovation [1] toward more important and hypersensitive fields, such as: business intelligence, Medical and natural language processing, which make ARM process and relationships among attributes of datasets indispensable.

Nowadays, discovery such relationships in large database is NP-Complete problem [3]. The huge quantity of stored data makes classical approaches applied to extract association rules in such database running slowly. In these methods the growth of features number results in a dramatic increase of the processing time. This is why researchers in ARM headed to optimization with intelligent algorithms which presents robust and efficient approaches to explore a massive search space. Intelligent algorithms have already shown their efficiency to solve combinatorial problems (NP-Complete). Generally, an evolutionary algorithm maintains a population of individuals; each one represents a solution to the given problem. Each individual is evaluated by a fitness function which determines solution quality. Assuming that databases are simple search spaces, same concepts exist in association rule problem where the algorithm maintains a set of rules which are individuals and evaluate them using different quality measures (confidence, comprehensibility, interesting ...). The most popular intelligent methods that have been applied to ARM problem are: Genetic algorithm, particle swarm algorithm, bees swarm algorithm, and bat algorithm. Most of these approaches deal with the ARM problem as a single-objective optimization problem. However, they still generate useless rules for decision making process because they utilize just support and confidence to evaluate the rules. Recently, many works on association rules deal with ARM as multi-objective optimization perspective to extract a small set of useful and comprehensible rules by introducing several measures in assessment of rules.

In this paper we propose a multi-objective method to mine interesting and useful association rules within transactional databases, starting from a minimum support and confidence threshold specified by the final users according to their needs, based on the multi-objective bat algorithm. In order to improve the efficiency of our algorithm some new contributions have been embedded in our proposal. We use four measures to evaluate extracted rules quality and define two global objective functions considered for optimization (maximization) in order to extract better promising association rules.

The rest of this paper is organized as follows. The next section presents a general background on multi-Objective Optimization and association rule mining problems. Also, we recall the modified bat algorithm for association rule mining (BAT-ARM). Section 3 leads with summary of the existing ARM algorithms. Section 4, presents formally our approach. Section 5, reports on the experimental results for our approach and the comparison with other ARM existing algorithms. Finally, we conclude with our prospective for a future work.

## 2 Preliminaries

### 2.1 Multi-objective Optimization Problems

A general Multi-Objective Optimization problems (MOOP) includes a set of  $n$  parameters (decision variables), a set of  $k$  objective functions, and a set of  $m$

constraints. Objectives and constraints are functions of the decision variables. Generally, The form of a MOOP can be described as minimizing/maximizing a set of objective functions,  $f(x) = (f_1(x), \dots, f_k(x))^T$  subject to  $G_i(x) \leq 0, i = 1, 2, \dots, n$ , by finding the vector  $X = (x_1, x_2, \dots, x_n)^T$ . It is noted that  $G_i(x)$  are the constraints that must be satisfied while minimizing/maximizing the objective functions. With the presence of several objective functions, the notion of “optimum” has changed to “Pareto optimum” [5] because MOOP aims to find a vector of solutions rather than a single solution. In general, it is not possible to find an exact PF for complex MOOPs, and in such cases the goal is to determine a Pareto optimal set that approximates the exact PF as close as possible by generating a diverse range of solutions.

## 2.2 Association Rule Mining

Formally, association rule [1] problem is defined as follow: Let  $I = \{i_1, i_2, \dots, i_n\}$  be a set of literals called items, let  $D$  be a transactional database where each transaction  $T$  contains a set of items. An association rule is implication like  $X \implies Y$  where  $X, Y \in I$  and  $X \cap Y = \emptyset$ . The item-sets  $X, Y$  are named antecedent and consequent, respectively. Mainly, two principal measures are used to detect the interesting and useful association rules: Support and Confidence. They are defined as follows:

**Support:** written  $\text{supp}(X)$ , it is the proportion of transactions in  $D$  that contains  $X$ , to the total of records in database. Support is calculated using the following equation

$$\text{supp}(X) = |\{(y, Xy) \in D / X \subseteq Xy\}| / |D| \quad (1)$$

The support of an association rule  $X \rightarrow Y$  is the support of  $X \cup Y$ .

**Confidence:** written  $\text{conf}(X \rightarrow Y)$ , it is the proportion of transactions covering  $X$  and  $Y$ , to the total of records containing  $X$ , when the percentage exceeds threshold of confidence an interesting association rule can be generated. The confidence of a rule is calculated as:

$$\text{conf}(X \rightarrow Y) = \text{supp}(X \cup Y) / \text{supp}(X) \quad (2)$$

In another word, support denotes the frequency of occurring patterns while confidence expresses the strength of implication [1].

## 2.3 ARM Based on Bat Algorithm (BAT-ARM)

In [11], we proposed a new algorithm for ARM inspired from bat behavior, which aims to generate the best rules in defined dataset starting from minimum support and confidence with reasonable execution time. In association rule mining, the rule is accepted if its support and confidence satisfy user minimum support and confidence threshold. Based on this definition we describe a simple objective function based on the support and the confidence to evaluate the solution and

never generate invalid rules. Based on the definition of bat algorithm in [18] new formal description for bat motion is described related to association rule mining bases, where the frequency, velocity and position are defined as follow:

- **Frequency**  $f_i$ : presents how many items can be changed in the actual rule, where the maximum frequency  $f_{max}$  is the number of attributes in the dataset and the minimum frequency  $f_{min}$  is 0.
- **Velocity**  $v_i$ : indicates where the changes will be started.
- **Position**  $x_i$ : it is the new generated rule based on new frequency, velocity and the loudness.

The generation of new positions (rules) is extracted based on the frequency, velocity of each virtual bat which are updated at each iteration by Eqs. 3 and 4

$$f_i^t = 1 + (f_{max})\beta, \quad (3)$$

$$v_i^t = f_{max} - f_i^t - v_i^{t-1}, \quad (4)$$

Generally, BAT-ARM provides a great performance in term of CPU-time and memory usage in the face of FP-growth algorithm, thanks to the echolocation concept of the bat algorithm that can determine which part of the best rule have changed to get a better position (rule) for the actual bat. However, the ARM by means of single objective optimization methods also has a few limits that are listed as follows. Firstly, in order to solve association rule mining issues, these methods explore the maximum of search space, which generates many rules having a high fitness value. This process can generate many redundant rules. Secondly, as this method focuses on covering the search space and generating the maximum of rules, it neglects the comprehensibility and usability of rules that are meaningful for the end users. To overcome these main drawbacks we investigate with a new approach based on multi-objective bat algorithm.

### 3 Literature Review

This section presents a literature review on the existing evolutionary algorithms that deal with ARM issue. In the literature, there are many other bio-inspired approaches are proposed to extract association rules, In [17] G3PARM algorithm is developed, it is based on genetic programming. The authors used grammar guided genetic programming (G3P) to avoid invalid individuals found by Genetic Programming (GP) process. Also, G3PARM permits multiple variants of data by using a context free grammar. In [6] the authors developed a new approach inspired from bees behavior and based on bee swarm optimization algorithm called BSO-ARM. The results of this approach show that BSO-ARM performs better than all genetic algorithms. As extension to their work, the authors present an amelioration to BSO-ARM in [7], where three strategies to determine the search area of each bee are proposed (modulo, next, syntactic). In our earlier work [11] we present an adaptation of bat algorithm to association

rule mining issue known as BAT-ARM. We present a new mathematical definition for the virtual bats motion related to ARM problem basics. The outcomes show a high performance in solution quality and CPU-time consumption thanks to the echolocation concept of bat algorithm. Later, within [12] we propose a multi-population bat algorithm to extract association rules within transactional database which is based on the search process developed in BAT-ARM. Furthermore, sub-populations use master-slave plan to cooperate among themselves. The results outperform those of BAT-ARM in both quality and time execution. This later was improved by new cooperation strategies in [13]. All these motioned methods are single objective approaches which stay suffer from several drawbacks mainly the huge number of generated rules and the extraction of useless ones.

These disadvantages open the door to dealt with association rule mining as a multi-objective optimization problem where different measures are used in the same algorithm. In [15], a multi-objective genetic algorithm approach to mine association rules for numerical data was proposed, where confidence, interestingness and comprehensibility are used to define the fitness function. Results showed that the generated rules are more appropriate than similar approaches. In [16] the authors proposed a multi-objective genetic algorithm for generating interesting association rules with multiple criteria i.e. support, confidence and simplicity (comprehensibility). Their method can identify the interesting rules without having the user-specified thresholds of minimum support and minimum confidence. Another study presented in [4] discussed multi-objective particle swarm optimization algorithm for numerical ARM named MOPAR. This method uses confidence, comprehensibility, and interestingness to evaluate the extracted rules. In [8], three multi-objective techniques proposed for mining association rules without specifying neither support nor confidence by optimizing several quality measures. The methods are Multi-objective Binary Particle Swarm Optimization (MO-BPSO), a Multi-objective Binary Firefly optimization and Threshold Accepting (MO-BFFO-TA), and a Multi-objective Binary Particle Swarm optimization and Threshold Accepting (MO-BPSO-TA). More recently, a new multi-objective evolutionary algorithm, MBAREA, for mining useful Boolean association rules with low computational cost is proposed in [14].

## 4 Multi-objective Bat Algorithm for ARM

### 4.1 Rule Encoding

In our method we use Michigan Approaches. Where each solution  $X$  represents a rule that contains  $k$  items. Therefore, the solution  $X$  is represented with a vector  $S$  which contains  $k + 1$  positions where:

1.  $S[0]$  separates between the antecedent and the consequent of the rule,
2.  $S[i] = j$  where  $i > 0$  If the  $j^{th}$  item in the database is in the rule, else the position contains 0.

For example, let  $I = \{i_1, i_2, \dots, i_{10}\}$  be a set of items:

- $X1 = \{3, 1, 5, 0, 6, 2, 0, 0, 7, 0, 0\}$  represents the rule  $i_1, i_5 \Rightarrow i_6, i_2, i_7$ ,

## 4.2 Objective Functions

As mentioned above single objective evolutionary algorithms use generally only one measure i.e., *Support*, *Confidence*, *etc.* to evaluate extracted rules quality. These measures assess rules depending on number of occurrence in database. Nevertheless, these algorithms do not give any importance to other rule quality measures like i.e., comprehensibility and interestingness.

In our work we use *comprehensibility*, *interestingness* measures in addition to confidence and support which are used as fitness function in [11], and as objectives in our method (MOB-ARM). The confidence criterion evaluates the quality of each rule based on occurrences number in the whole dataset. When the rule has more occurrences in the database, this means the rule has a better quality. We define the first objective for our method using the support and confidence, shown in Eq. 5.

$$Obj_1(R) = \alpha conf(R) + \beta supp(R) / \alpha + \beta \quad (5)$$

When the rule contains a huge number of attributes, this makes the rule more difficult to comprehend. If generated rules are not comprehensible for the user, they will be useless. This is why we introduce the comprehensibility measure, which can be modeled as shown in Eq. 6.

$$Comprehensibility(X) = \frac{\log(1 + |Conseq|)}{\log(1 + |Antec \cup Conseq|)} \quad (6)$$

Where,  $|Conseq|$  and  $|Antec \cup Conseq|$  are items number in the Consequence part and the total rule respectively. The comprehensibility increases and the rules are more understandable whenever items number in the antecedent part was smaller. Moreover, interestingness of a rule is used to quantify how much rule is surprising for users. As the most important point of rule mining is to find some hidden information, it should discover those rules having comparatively less occurrence in the database. Interestingness measure is defined by Eq. 7.

$$Interesting(X) = \frac{Supp(A \cup C)}{Supp(A)} \times \frac{Supp(A \cup C)}{Supp(C)} \times \frac{(1 - Supp(A \cup C))}{N} \quad (7)$$

Where A, C and N are the antecedence, consequence and transactions number in the whole database, respectively. We define the second objective for our algorithm based on *Comprehensibility*, *Interestingness* using Eq. 8.

$$Obj_2(R) = \gamma Comp(R) + \delta Inter(R) / \gamma + \delta \quad (8)$$

Where  $\alpha, \beta, \gamma$  and  $\delta$  are empirical parameters which are chosen relative to the importance of support, confidence, Comprehensibility and Interestingness to final user.

**Algorithm 1.** MOB-ARM Algorithm pseudo code

---

```

objective functions  $f_1(x), \dots, f_k(x)$ .
Initialize the bat population  $x_i$  and  $v_i$ ;
Initialize pulse rates  $r_i$  and the loudness  $A_i$ ;
for  $j = 1$  to  $N$  (points on Pareto fronts) do
    Generate  $K$  weights  $w_k \geq 0$  so that  $\sum_{k=1}^k w_k = 1$ ;
    Form a single objective  $\sum_{k=1}^k w_k f_k$ ;
    while ( $t < \text{Max number of iterations}$ ) do
        Generate new solutions by adjusting frequency  $f_i$ ;
        and updating velocities and locations/solutions [11];
        Generate a new solution  $x_i$  [11];
        if ( $\text{rand} > r_i$ ) then
            Generate a local solution around the selected best solution by changing only
            one item in the rule;
        end if
        if ( $f(x_i) > f(x_{i*})$ ) then
            Accept the new solutions;
             $x_{i*} = x_i$ ;
            Increase  $r_i$  and reduce  $A_i$ 
        end if
        Rank the bats according to the best solution;
    end while
    Record  $x_{i*}$  as non-dominate solution;
end for
Post-process results and visualize the best detected rules.

```

---

**4.3 The Algorithm Flow**

Algorithm 1 illustrates the pseudo code of MOB-ARM. The main computational steps of the proposal are described as follows:

- **Initialization step:** firstly, all the bats are initialized with random frequency and velocity. The values are taken in the intervals  $[0, \text{items number}]$  and  $[0, \text{items number}+1]$  respectively. A randomly generated position/solution (rule) is affected to each bat, and an initial rate and loudness is affected to each bat randomly.
- **Search the non-dominate solution for the Pareto point:** For each Pareto point, a new single global objective function is generated based on weights  $w_k$  in which their sum is equal to 1 ( $\sum_{k=1}^k w_k = 1$ ). The global objective function is generally presented by:  $\sum_{k=1}^k w_k f_k$ , where  $k$  is objective functions number used for the mining problem. In our case, we have only two objective functions presented in Eqs. 5 and 8. So the global function is defined as follows:

$$\text{Obj}(R) = w_1.\text{Obj}_1(R) + w_2.\text{Obj}_2(R); \quad (9)$$

- **Search the best solution (Rule) for each bat at the Pareto point:** At each iteration, a new rule is generated based on BAT-ARM described in [11]

by adjusting frequency, updating velocity and location. If the new fitness is better than the previous one, then the rule will be accepted, the loudness  $A_i$  reduced, and the rate  $r_i$  increased according to updating equations in [18].

## 5 Experimental Results

In order to perform experimentations, several well-known and frequently used real world datasets in data mining, such as Frequent and mining dataset Repository [9], Bilkent University Function Approximation Repository [10], are used in this section for several tests. This section describes the used benchmarks. After-that, comparative study with BAT-ARM and MPB-ARM, which are two mono-objective versions of bat algorithm updated to association rules mining, is given. Also, we present comparison of our approach to three other multi-objective methods recently published. All algorithms are written in Java and executed on Intel core I5 machine with 4GB of memory running under Linux Ubuntu. We examined our approach on seven well known datasets with different sizes of transactions, items and average size per transaction. For instance, Chess dataset has 3196 transactions with 75 items when the average per transaction is 37, unlike mushroom dataset which has much more transactions and items when it has just 23 items per transaction. Table 1 presents different datasets used in our experiments.

**Table 1.** Description of experimental benchmark

Dataset	Transactions size	Item size
Basketball	96	5
Bodyfat	252	15
Quake	2178	4
IBM Quest Standard	1000	20
Chess	3196	37
Mushroom	8124	23

### 5.1 Comparative Study to Single Objective Approaches

In this section, we propose a study that compares our new approach to single objective versions of bat algorithm designed for mining association rules (BAT-ARM, MPB-ARM). This experiment was cried on three datasets with medium transactions size (*IBM Quest Standard*, *Chess* and *Mushroom*). The default parameters of the BAT-ARM, MPB-ARM and MOB-ARM are defined to make the comparison completely fair where support and confidence thresholds are fixed to 0.2 and 0.5 respectively.

Table 2 presents the average results of thirty executions on three algorithms (BAT-ARM, MPB-ARM and MOB-ARM). In our comparison, three axes are



**Table 2.** Comparison of results to mono-objective methods in terms of number of generated rules, support and confidence

	Algorithms	Datasets		
		IBM-standard	Chess	Mashroom
No. of rules	MOB-ARM	215	293	26
	BAT-ARM	485	1870	341
	MPB-ARM	850	739	791
Support (%)	MOB-ARM	26	51	34
	BAT-ARM	25	38	23
	MPB-ARM	23	46	23
Confidance (%)	MOB-ARM	54	83	87
	BAT-ARM	52	72	54
	MPB-ARM	59	79	78.5

taken into account: average support, confidence and number of generated rules. Outcomes shows that the proposed algorithms extract less number of rules for all the datasets. This is because new criteria of selection are introduced as objectives (Comprehensibility and interestingness), so MOB-ARM generates only useful and understandable rules for the user. On contrary, mono-objective approaches generate the maximum number of rules that satisfy support and confidence thresholds. In terms of support and confidence we note that MOB-ARM is more robust than BAT-ARM and MPB-ARM because of the small number of extracted rules and dominance conditions applied when mining association rules.

## 5.2 Comparative Study to Multi-objective Approaches

In this study, effectiveness of the proposed algorithm is compared with three similar algorithms. All of these methods are based on a multi-objective evolutionary approach and designed for association rule mining. The three algorithms are: MODENAR [2], MOGAR [15] and MOPAR [4].

Table 3 compares the outcomes obtained by MOB-ARM to previous similar methods that deal with association rule mining as multi-objective optimization problem in terms of average support. The results show that our proposed method yields a competitive support of the extracted rule. However, in case of Bodyfat datasets the overage support is less than other methods (MODENAR, MOGAR), this is caused by the fact of the strict application of dominance conditions.

In addition, we compute the average confidence to evaluate the strength of extracted rules. Table 3 shows that our suggested method gives acceptable results. Our results can be improved and give better confidence average because we use a minimum confidence threshold that can be changed by the user according to his exigencies. To make the study more comprehensive, we calculate the average number of extracted rules for each dataset and the results are presented in Table 3. From the outcomes, we observe that our method have a stable behavior and it is competitive to the previous methods.

**Table 3.** Comparison of results to mono-objective methods in terms of number of generated rules, support and confidence

	Algorithms	Datasets		
		Basketball	Bodyfat	Quake
Support (%)	MOPAR	30.76	22.95	31.97
	MODENAR	37.20	65.22	39.86
	MOGAR	50.85	57.22	30.12
	MOB-ARM	37.5	23	41
Confidance (%)	MOPAR	95	81.8	89.32
	MODENAR	61	62	63
	MOGAR	83	85	79
	MOB-ARM	79	83	88
No. of rules	MOPAR	69	70	54
	MODENAR	48	52	55
	MOGAR	50	84	44
	MOB-ARM	63	51	50

## 6 Conclusion

In this paper, we proposed a new multi-objective meta-heuristic to deal with association rule mining based on bat algorithm and called MOB-ARM. The proposal uses four quality measures, (Support, Confidence, Comprehensibility and Interestingness) to extract the best rules that help the user in decision making process and can be understood. Our approach is based on vertical dataset representation that reduces the computation time of computing and avoids the repeated scans of the whole datasets in each rule evaluation. The performance of MOB-ARM has been compared to two single objective algorithms based bat algorithm for mining association rules and three other methods dealing with multi-objective miners. The experimental results prove the effectiveness of our proposed method. For the near future, we aim to develop a new version that deals with quantitative association rules without requiring a discretization step. We think also about parallelism the algorithm and implement it on a GPU to improve both solution quality and running time.

## References

1. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: ACM SIGMOD Record, vol. 22, pp. 207–216. ACM (1993)
2. Alatas, B., Akin, E., Karci, A.: Modenar: multi-objective differential evolution algorithm for mining numeric association rules. Appl. Soft Comput. **8**(1), 646–656 (2008)

3. Angiulli, F., Ianni, G., Palopoli, L.: On the complexity of mining association rules. In: SEBD, pp. 177–184 (2001)
4. Beiranvand, V., Mobasher-Kashani, M., Bakar, A.A.: Multi-objective PSO algorithm for mining numerical association rules without a priori discretization. *Expert Syst. Appl.* **41**(9), 4259–4273 (2014)
5. Coello, C.A., Coello, D.A., Veldhuizen, V., Lamont, G.B.: *Evolutionary Algorithms for Solving Multi-Objective Problems*, vol. 242. Springer, Heidelberg (2002)
6. Djenouri, Y., Drias, H., Habbas, Z., Mosteghanemi, H.: Bees swarm optimization for web association rule mining. In: 2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT), vol. 3, pp. 142–146. IEEE (2012)
7. Djenouri, Y., Drias, H., Habbas, Z.: Bees swarm optimisation using multiple strategies for association rule mining. *Int. J. Bio-Inspired Comput.* **6**(4), 239–249 (2014)
8. Ganghishetti, P., Vadlamani, R.: Association rule mining via evolutionary multi-objective optimization. In: Murty, M.N., He, X., Chillarige, R.R., Weng, P. (eds.) *MIWAI 2014. LNCS (LNAI)*, vol. 8875, pp. 35–46. Springer, Cham (2014). doi:[10.1007/978-3-319-13365-2\\_4](https://doi.org/10.1007/978-3-319-13365-2_4)
9. Goethls, B., Zaki, M.J.: Frequent itemset mining dataset repository (2003). <http://fimi.ua.ac.be/data/>
10. Guvenir, H.A., Uysal, I.: Bilkent university function approximation repository (2000). <http://funapp.cs.bilkent.edu.tr/DataSets/>
11. Heraguemi, K.E., Kamel, N., Drias, H.: Association rule mining based on bat algorithm. *J. Comput. Theor. Nanosci.* **12**(7), 1195–1200 (2015)
12. Heraguemi, K.E., Kamel, N., Drias, H.: Multi-population cooperative bat algorithm for association rule mining. In: Núñez, M., Nguyen, N.T., Camacho, D., Trawiński, B. (eds.) *ICCCI 2015. LNCS (LNAI)*, vol. 9329, pp. 265–274. Springer, Cham (2015). doi:[10.1007/978-3-319-24069-5\\_25](https://doi.org/10.1007/978-3-319-24069-5_25)
13. Heraguemi, K.E., Kamel, N., Drias, H.: Multi-swarm bat algorithm for association rule mining using multiple cooperative strategies. *Appl. Intell.* **45**(4), 1021–1033 (2016). Springer
14. Kabir, M.M.J., Xu, S., Kang, B.H., Zhao, Z.: A new evolutionary algorithm for extracting a reduced set of interesting association rules. In: Arik, S., Huang, T., Lai, W.K., Liu, Q. (eds.) *ICONIP 2015. LNCS*, vol. 9490, pp. 133–142. Springer, Cham (2015). doi:[10.1007/978-3-319-26535-3\\_16](https://doi.org/10.1007/978-3-319-26535-3_16)
15. Minaei-Bidgoli, B., Barmaki, R., Nasiri, M.: Mining numerical association rules via multi-objective genetic algorithms. *Inf. Sci.* **233**, 15–24 (2013)
16. Al-maqaleh, M.B.: Discovering interesting association rules: a multi-objective genetic algorithm approach. *Int. J. Appl. Inf. Syst.* **5**(3), 47–52 (2013)
17. Olmo, J.L., Luna, J.M., Romero, J.R., Ventura, S.: Association rule mining using a multi-objective grammar-based ant programming algorithm. In: 2011 11th International Conference on Intelligent Systems Design and Applications (ISDA), pp. 971–977, November (2011)
18. Yang, X.-S.: A new metaheuristic bat-inspired algorithm. In: González, J.R., Pelta, D.A., Cruz, C., Terrazas, G., Krasnogor, N. (eds.) *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010). Studies in Computational Intelligence*, vol. 284, pp. 65–74. Springer, Heidelberg (2010). doi:[10.1007/978-3-642-12538-6\\_6](https://doi.org/10.1007/978-3-642-12538-6_6)

Mining Intelligence and Knowledge Exploration  
4th International Conference, MIKE 2016, Mexico City,  
Mexico, November 13 - 19, 2016, Revised Selected  
Papers

Prasath, R.; Gelbukh, A. (Eds.)

2017, XII, 185 p. 50 illus., Softcover

ISBN: 978-3-319-58129-3