Mining Interesting Association Rules with a Modified Genetic Algorithm

Abir Derouiche\textsuperscript{1(ES)}, Abdesslem Layeb\textsuperscript{2}, and Zineb Habbas\textsuperscript{3}

\textsuperscript{1} MISC Laboratory, Abdelhamid Mehri, Constantine 2 University, Constantine, Algeria
\texttt{abir.derouiche@univ-constantine2.dz}

\textsuperscript{2} Abdelhamid Mehri, Constantine 2 University, Constantine, Algeria
\texttt{abdesslem.layeb@univ-constantine2.dz}

\textsuperscript{3} University of Lorraine, Metz, France
\texttt{zineb.habbas@univ-lorraine.fr}

Abstract. Association Rules Mining is an important data mining task that has many applications. Association rules mining is considered as an optimization problem; thus several metaheuristics have been developed to solve it since they have been proven to be faster than the exact algorithms. However, most of them generates a lot of redundant rules. In this work, we proposed a modified genetic algorithm for mining interesting non-redundant association rules. Different experiments have been carried out on several well-known benchmarks. Moreover, the algorithm was compared with those of other published works and the results found proved the efficiency of our proposal.

Keywords: Data mining · Association rules mining · Multi-objective association rules mining · Metaheuristics · Genetic algorithm · Quality measures · Redundancy

1 Introduction

Association Rules Mining (ARM) has become one of the main topics in data mining. It attracts a lot of attention because of its wide applicability in different area such as in web mining [24], document analysis [17], telecommunication alarm diagnosis [19], network intrusion detection [5], and bioinformatics applications [27].

Most of the classical association rules mining algorithms, such as the Apriori algorithm [2] and the FP-growth algorithm [13] work in two phases, the first phase is the frequent itemsets generation and the second phase is the rules generation.

Generating the frequent itemsets might be considered as a straightforward task. However, it is a time-consuming task when the number of items is large. For example a dataset comprising \( n \) items contains \( 2^n - 1 \) different itemsets,
whereas the number of itemsets of size \( k \) is equal to \( \binom{n}{k} \) for any \( k \leq n \). Thus, given the amount of computations needed for each candidate rule is \( O(k) \) the overall complexity is \( O\left(\sum_{k=1}^{n} k \times \binom{n}{k}\right) = O(2^{n-1} \times n) \).

So the complexity of finding itemsets is in exponential order, and this complexity is even higher when the frequency of each itemset is calculated\[10\].

For any dataset comprising \( n \) items and \( m \) different transactions, the complexity to compute the frequencies of all the itemsets within the dataset is equal to \( O(2^{n-1} \times n \times m) \) [28].

On the other hand, metaheuristics are increasingly considered as a more promising alternative approach. They have been proven beneficial as they directly generate association rules, skipping the frequent itemsets generation phase by maximizing the support and the confidence of the rules. The quality of an association rule is not limited to its support and confidence, there are many other metrics available to measure the quality of an association rule such as coverage, comprehensibility, leverage, interestingness, lift and conviction. Therefore, the problem of ARM can be considered also as a multi-objective optimization problem [11].

Although ARM metaheuristics are proved to be faster than ARM exact algorithms, most of them suffer from accuracy. They do not generate all the rules and not necessarily those of good quality. Besides, a large number of association rules returned to the user are redundant, so it is time-consuming to analyses the results, the user has to handle a large proportion of redundant rules, most of these rules are not interesting for the application at hand.

This article proposed a Modified Genetic Algorithm for Mining interesting Association Rule (MGA-ARM), different contributions have been embedded in the proposed algorithm. First, the random initial population is replaced by a special initial population to enhance both the CPU time and the solution quality. Secondly, we consider different measures beside support and confidence in the objective function in order to extract better association rules and finally, we propose a method to handle and prune the non-significant redundant rules. Different experiments have been carried out on several well-known benchmarks. The results were compared with those of other published methods and proved the efficiency of the proposal.

2 Association Rules Mining

2.1 Definitions

The formal definition of an association rule was initiated by Agrawal in [1]. Let \( I = \{i_1, i_2, \ldots, i_n\} \) be a set of \( n \) items, \( T = \{t_1, t_2, \ldots, t_m\} \) a set of \( m \) transactions where each transaction \( t_i \) is a set of items such that \( t_i \subseteq I \). An association rule is a pattern of the form \( X \rightarrow Y \) meaning that there is an association between the presence of the itemsets \( X \) and \( Y \) in transactions. \( X \) is called antecedent or left-hand side of the rule, \( Y \) is called consequence or right-hand side of the rule. The association rules have two significant basic measures: support and confidence. Given an association rule \( (X \rightarrow Y) \):
The support denoted support \((X \rightarrow Y)\) is the ratio between the number of transactions containing \(X \cup Y\) and the number of transactions in the database. It determines how often the rule is applicable to a given dataset.

\[ \text{Support}(X \rightarrow Y) = \frac{\text{Support}(X \cup Y)}{\text{Total number of transactions}} \quad (1) \]

The confidence denoted confidence \((X \rightarrow Y)\) is the ratio between the number of transactions containing \(X \cup Y\) and the number of transactions containing \(X\). It determines how frequently items in \(Y\) appear in transactions that contain \(X\).

\[ \text{Confidence}(X \rightarrow Y) = \frac{\text{Support}(X \cup Y)}{\text{Support}(X)} \quad (2) \]

**Association rules mining task** is formally stated as follows: given a set of transactions \(T= \{ T_1, T_2, ..., T_m \}\), the objective is to find all valid rules i.e. rules having support and confidence greater than the user-specified minimum support (mnsup) and minimum confidence (minconf).

### 2.2 Association Rules Mining as an Optimization Problem

Association rules mining task is not limited to the previous definition, it is considered as an optimization problem that consists to find the best rules \(r \in R\) where \(R\) is the set of all possible rules, while maximizing the values of the support and confidence [8].

There are many other metrics available to measure the quality of an association rule such as coverage, comprehensibility, leverage, interestingness, lift and conviction [11]. Therefore, the problem of ARM can be considered also as a multi-objective optimization problem rather than a single objective one, where the goal is to find association rules while optimizing several objective measures simultaneously.

Whether we use just support and confidence or we use more measures, metaheuristic always generate duplicated and redundant rules and this effects the quality of the results presented to the user.

### 2.3 Redundant Rules

**Definition 1.** Let \(X \rightarrow Y\) and \(X' \rightarrow Y'\) be two rules with confidence \(cf\) and \(cf'\), respectively. \(X \rightarrow Y\) is said a redundant rule to \(X' \rightarrow Y'\) if \(X\) belongs to \(X'\); \(Y\) belongs to \(Y'\) and \(cf \leq cf'\) [29].

**Definition 2.** Let \(R1\) be the rule \(X \rightarrow Y\) and \(R2\) the rule \(X' \rightarrow Y'\). We say that the rule \(R1\) is more general than the rule \(R2\), denoted \(R1 \leq\leq R2\), if \(R2\) can be generated by adding additional items to either the antecedent or consequent of \(R1\) [30].
Formally let \( R = R_1, \ldots, R_n \) be a set of rules, such that all their supports and confidences are equal. For all \( i, 1 \leq i \leq n \) if \( X \subseteq X' \) and \( Y \subseteq Y' \), then we say that a rule \( R_j \) is redundant if there exists some rule \( R_i \), such that \( R_i \leq R_j \).

In other words, since all the rules in the collection \( R \) have the same support and confidence, the simplest rules in the collection should suffice to represent the whole set. Thus the non-redundant rules in the collection \( R \) are those that are most general, i.e., those having minimal antecedents and consequent, in terms of subset relation [30]. This second definition is considered in our work.

3 Related Works

Most metaheuristics were investigated to solve the ARM problem, whether evolutionary or swarm intelligence algorithms. Saggar et al. [25] were the first authors to propose the use of evolutionary algorithms for ARM, their idea consisted in optimizing association rules extracted with the Apriori algorithm by using a genetic algorithm, after that genetic programming was also used for ARM in [23]. An interesting survey about the use of evolutionary computation for frequent pattern mining with particular emphasis on genetic algorithms can be found in [28]. Particle swarm optimization was largely used to mine association rules a survey of its applications is proposed in [4], we can cite for example the work published in [20] and the work proposed by Sarath and Ravi in [26], after that in [21] a Modified Binary Cuckoo search (MBCS-ARM) was proposed. In [14] the authors present an adaptation of bat algorithm to ARM issue known as BAT-ARM. Later, they proposed a multi-population bat algorithm in [15].

In [22], the authors used Ant Colony Optimization for continuous domains (ACO\(_R\)), this algorithm mines numeric association rules without any need to specify minimum support and minimum confidence. Another recent works propose an algorithm based on animal migration optimization [6] and chemical reaction optimization metaheuristic [7].

Hybrid approaches was also proposed such as the work in [9] where the authors proposed an algorithm based on bees swarm algorithms and tabu-search and in the work [18] where the authors combine both genetic algorithm and particle swarm optimisation in an algorithm called (GPSO).

The following works considered the association rules mining as a multi-objective problem rather than a single-objective one, in [12] the authors proposed a pareto based genetic algorithm where they used three measures: comprehensibility, interestingness and confidence as objective functions.

Alatas et al. [3] proposed pareto-based multi-objective differential evolution (DE) for extracting association rules. They formulated the association rule mining problem as a four-objective optimization problem, where, support, confidence and comprehensibility of rules are maximized, while the amplitude of the intervals, which conforms the item set and rule is minimized.

In [11], the Multi-objective Binary Particle Swarm Optimization (MO-BPSO), the Multi-objective Binary Firefly optimization and Threshold Accepting (MO-BFFO-TA), and the Multi-objective Binary Particle Swarm optimization and Threshold Accepting (MO-BPSO-TA) were used to extract association
rules without specifying support and confident threshold’s. Recently a multi-objective bat algorithm known as MOB-ARM is proposed in [16].

4 The Proposed Modified Genetic Algorithm for ARM

To use metaheuristics, the objective function, the representation of the solution and the different operators need to be defined and adapted to the problem in hand.

4.1 Objective Function and Rule Representation

For the rule encoding, the solution $S$ is represented by an integer vector defined as follows:

- $S[i] = 0$ if the item $i$ is not present in the rule $S$.
- $S[i] = 1$ if the item $i$ belongs to the antecedent part of the rule $S$.
- $S[i] = 2$ if the item $i$ belongs to the consequent part of the rule $S$.

Example: Given a set of five items \{1, 2, 3, 4, 5\}, the rule \{1, 4 $\rightarrow$ 2, 5\} is represented by $S = \{1, 2, 0, 1, 2\}$.

The Objective Function: Four quality measures of the rule was maximized in the objective function, we used the comprehensibility and the interestingness measure in addition to the support and the confidence.

$$ F(S) = \text{Max}(\text{Support}(S) \times \text{Confidence}(S) \times \text{Comprehensibility}(S) \times \text{Interestingness}(S)) $$ (3)

where according to [12]

- The Comprehensibility of an association rule quantify how much the rule is comprehensible

$$ \text{Comprehensibility}(X \rightarrow Y) = \frac{\log(1 + |X|)}{\log(1 + |X \cup Y|)} $$ (4)

- Interestingness: a rule is interesting when the individual support count values are greater than the collective support $(X \rightarrow Y)$ values

$$ \text{Interestingness}(X \rightarrow Y) = \frac{\text{support}(X \cup Y)}{\text{support}(X)} \times \frac{\text{support}(X \cup Y)}{\text{support}(Y)} \times \frac{1 - \text{support}(X \cup Y)}{|N|} $$ (5)
4.2 Algorithm and Operators

In this section we present the Modified Genetic Algorithm for ARM (MGA-ARM), we explain the different steps of the algorithm and the proposed method to prune the non-significant redundant rules.

- Search space: we believe that the search space can be easily pruned using the Apriori principle and some other propriety. The following example shows that the infrequent items are useless because they always lead to infrequent rules. Let $ab \rightarrow c$ be a given frequent rule and $d$ an infrequent item, that means $sup(d) < minsup$. Consequently $sup(abcd) < minsup$, therefore, all rules generated from the itemset \{abcd\} will be infrequent. In our proposal, the search space contains a set of frequent items instead of considering all the items [8].

- Initial population: most the metaheuristic use a random initialization and exploit all the set of items to create the population where infrequent items generate infrequent rules and this is useless for the search procedure and it is a waste of time as explained previously. For MGA-ARM, the random population is replaced by a set of valid rules of size two. Algorithm 1 resumes the different steps of the procedure \textbf{InitPop} used for the generation of the initial population.

\begin{algorithm}
\textbf{Algorithm 1.} \textbf{InitPop}($T$, $ValidAR.2$ )
\begin{algorithmic}[1]
\State \textbf{Input:} $T$ a transactional dataset
\State \textbf{Outputs:} $ValidAR.2$ a set of valid Association Rules of size 2,
\Function{InitPop}{\State \textbf{Begin}}
\State $FIS.1 \leftarrow \textbf{BestFrequentItemset.1}(T)$
\State $AR.2 \leftarrow \textbf{GenerateRules} (FIS.1, T)$
\State $ValidAR.2 \leftarrow \textbf{SelectValidRules} (AR.2)$
\EndFunction
\end{algorithmic}
\end{algorithm}

\textbf{InitPop} starts by extracting a set of frequent items $FIS.1$ from the dataset $T$, then based on this set, it generates a set of rules that contains two frequent items one at the antecedent and the other one at the consequence of the rule. Finally, it selects the best rules within the generated rules and saves it as $ValidAR.2$ to be the special initial population exploited by our algorithm [8].

- Crossover: in our algorithm, the proposed crossover takes an item that does not exist in parent1 and add it to parent2, and add a new item to parent1 from parent2 that does not exist beforehand. This operator ensures that the new solutions are feasible since the parents were feasible solutions. Figure 1 shows an example of the crossover operator. First, we copy parent1 to child1 and parent2 to child2, then we move an item that does not exists in parent2 and exists in parent1 to child2 and vice versa respectively.
Fig. 1. Example of the crossover operator

- Mutation: the exchange operator is applied between two items one at the antecedent and one at the consequent of the rule. Figure 2 shows an example of applying the mutation operator on the rule $1 \rightarrow 5$.
- Selection: the elitism selection is used.

Fig. 2. Example of the mutation operator

- Stopping criteria: it can be reached if the objective function value is less than a predefined threshold, or a maximum number of iterations performed without improvements. In our case, we use the maximum number of iterations.
- Filter rules procedure introduced at the final step of MGA-ARM is used to eliminates three types of useless rules. First, it eliminates the duplicated rules. The second type of discarded rules are the rules constructed from the same itemset thus they have the same support however their confidence and fitness are different since they have different antecedent and consequence. The third type is the redundant rules, defined in Sect. 2.3, they are eliminated by grouping the rules that have the same values of support and confidence and their itemsets are included in each other. We keep then the general rule that has the smallest number of items and we delete the other ones.

The details of the proposed algorithm are given by Algorithm 2.
Algorithm 2. MGA-ARM

**Input:** $T$ a transactional database  
**Output:** $IR$ a set of interesting Association Rules

1: Initialize the parameters $\text{pop.size, iter, max.iter, cross.rate, mutate.rate}$;  
2: $\text{pop} \leftarrow \text{InitPop()}$  
3: Evaluate_fitness($\text{pop}$)  
4: **while** $\text{iter} \leq \text{max.iter}$ **do**  
5: **for** $i \leftarrow 1$ to $\text{pop.size}$ **do**  
6: Select(parent1, parent2)  
7: **if** $\text{rand}(0, 1) \leq \text{cross.rate}$ **then**  
8: Child $\leftarrow$ Crossover(parent1, parent2)  
9: **end if**  
10: **if** $\text{rand}(0, 1) \leq \text{mutate.rate}$ **then**  
11: Child $\leftarrow$ Mutation()  
12: **end if**  
13: **end for**  
14: $\text{iter} \leftarrow \text{iter} + 1$  
15: Update pop  
16: **end while**  
17: $S \leftarrow$ best chromosomes  
18: $\text{IR} \leftarrow$ Filter_rules($S$)

5 Experiments and Results

The experiments were done in two steps, in the first step a study that compares the proposed approach to other approaches in terms of quality of solutions and CPU run time is presented. Then in the second step, a study about the effect of pruning redundancy on the final solution is explained.

5.1 Experimental Environment and Datasets

The experiments were done under Windows 8 using a desktop computer with Intel Core-i3 processor, 1.8 GHz and 6 GB memory, all the implementations have been achieved using Matlab. The tests have been conducted on well-known scientific datasets, frequently used by the data mining community: Chess database that contains 3196 transactions and 75 items. Mushroom dataset which has much more transactions 8124 and 119 items. IBM Quest Standard dataset with 1000 transaction and 40 items. Finally, two small data set Book and Food with also 1000 transaction and 11 items.

The dataset can be found found in (http://fimi.uantwerpen.be/data/) and (http://funapp.cs.bilkent.edu.tr/DataSets/) and (https://www.solver.com/xlminer) and (www.ibm.com/software/analytics/spss/) respectively.

5.2 A Comparative Study with Other Approaches

In this section, we compared our proposed MGA-ARM to single and multi-objective algorithms. We used three different versions of the bat algorithm...
designed for mining association rules BAT-ARM [14], MPB-ARM [15], MOB-ARM [16]. The experiment was carried out on three datasets within the dataset stated before. The size of the population was fixed to 50 and the maximum number of iteration to 100.

Tables 1 and 2 presents the average results of support and confidence respectively in thirty executions.

Table 1. Comparing our approaches with other works in terms of support

| Dataset      | Algorithm | MGA-ARM | MOB-ARM | BAT-ARM | MPB-ARM |
|--------------|-----------|---------|---------|---------|---------|
| IBM-Quest    | 0.275     | 0.26    | 0.25    | 0.23    |
| Chess        | 0.935     | 0.51    | 0.38    | 0.46    |
| Mushroom     | 0.371     | 0.34    | 0.23    | 0.23    |

Table 2. Comparing our approaches with other works in terms of confidence

| Dataset      | Algorithm | MGA-ARM | MOB-ARM | BAT-ARM | MPB-ARM |
|--------------|-----------|---------|---------|---------|---------|
| IBM-Quest    | 0.594     | 0.54    | 0.52    | 0.59    |
| Chess        | 0.970     | 0.83    | 0.72    | 0.79    |
| Mushroom     | 0.653     | 0.87    | 0.54    | 78.5    |

In terms of support and confidence, the results show that our proposed algorithm MGA-ARM outperforms MOB-ARM, BAT-ARM and MPB-ARM in most cases.

Table 3 presents the CPU run times for the different algorithms in 100 runs the result are given in seconds.

Table 3. Comparing our approaches with other works in terms of CPU run time

| Dataset      | Algorithm | MGA-ARM | BAT-ARM | MPB-ARM |
|--------------|-----------|---------|---------|---------|
| IBM-Quest    | 9.1       | 10      | 7       |
| Chess        | 47        | 67      | 70      |
| Mushroom     | 119       | 141     | 130     |

Our proposed algorithm is faster than the other algorithms in most cases thanks to the special population proposed that avoid a lot of useless exploitation of the search space.
5.3 Study on the Effect of Pruning Redundancy

In this section, we will see the effect of applying Filter rules procedure on the results to eliminates the duplicated and redundant rules.

| Dataset     | Rules pruned | Support | Confidence |
|-------------|--------------|---------|------------|
|             | %            | Before  | After      | Before  | After      |
| Book        | [40 44]      | 0.1641  | 0.157      | 0.5432  | 0.525      |
| Food        | [40 50]      | 0.1053  | 0.1036     | 0.3571  | 0.3545     |
| IBM-Quest   | [10 16]      | 0.2778  | 0.2768     | 0.5366  | 0.5335     |
| Chess       | [06 14]      | 0.9008  | 0.8842     | 0.9537  | 0.9517     |
| Mushroom    | [04 10]      | 0.4520  | 0.4382     | 0.7437  | 0.7289     |

Table 4 shows the changes in support and confidence values before and after applying the Filter rules procedure. The interval in the first columns presents the percentage of rule deleted, it is different from dataset to another. It is clear that for the small datasets book and food, the percentage of rule pruned is large, in the best cases 40% of rules are deleted in the worst-case half of the rules are duplicated and redundant. For the larger data set, we notice that the percentage of rule deleted is smaller 16% in worst cases and 4% in best cases.

On the other hand, for the quality measures, we noticed that there is not a big difference in the average of the support and confidence in different runs. For the different data sets, the values in the worst-case decrease with 0.02 and in the best cases with 0.001 this is not important compared to the number of useless rules pruned.

The Filter rules procedure indeed decreases considerably the number of rules but it ensures to present to the user a set of interesting non-redundant association rules. It should be noted that when applying the filter procedure, it is recommended to select a large number of valid rules because after applying the Filter rules procedure the number of rules will decrease.

6 Conclusion

In this work, we proposed a modified genetic algorithm for mining interesting association rules by considering four different quality measures: support, confidence, comprehensibility, and interestingness. Moreover, we propose a new technique to build the initial population of GA, and we handle duplicated and redundant rules by proposing the Filter rules procedure. Different experiments have been carried out on several well-known benchmarks. The results compared with those of other published methods proved the efficiency of the proposal. We emphasize to optimize our algorithm by testing different crossover and mutation, and to test it on larger datasets. Moreover, we can use more quality measurements to define which are really better for ARM problem.
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. Agrawal, R., Srikant, R., et al.: Fast algorithms for mining association rules. In: Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, vol. 1215, pp. 487–499 (1994)
3. 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)
4. Ankita, S., Shikha, A., Jitendra, A., Sanjeev, S.: A review on application of particle swarm optimization in association rule mining. In: Satapathy, S.C., Udgata, S.K., Biswal, B.N. (eds.) Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA). AISC, vol. 199, pp. 405–414. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-35314-7_46
5. Barbará, D., Couto, J., Jajodia, S., Wu, N.: ADAM: a testbed for exploring the use of data mining in intrusion detection. ACM SIGMOD Rec. 30(4), 15–24 (2001)
6. Chiclana, F., et al.: ARM-AMO: an efficient association rule mining algorithm based on animal migration optimization. Knowl.-Based Syst. 154, 68–80 (2018)
7. Derouiche, A., Layeb, A., Habbas, Z.: Chemical reaction optimization metaheuristic for solving association rule mining problem. In: 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA), pp. 1011–1018, October 2017
8. Derouiche, A., Layeb, A., Habbas, Z.: Metaheuristics guided by the apriori principle for association rule mining: Case study-CRO metaheuristic. Int. J. Organ. Collective Intell. (IJOCI) 10(3), 14–37 (2020)
9. Djenouri, Y., Drias, H., Chemchen, A.: A hybrid bees swarm optimization and tabu search algorithm for association rule mining. In: 2013 World Congress on Nature and Biologically Inspired Computing (NaBIC), pp. 120–125. IEEE (2013)
10. Freitas, A.A.: Data Mining and Knowledge Discovery with Evolutionary Algorithms. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-662-04923-5
11. 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). https://doi.org/10.1007/978-3-319-13365-2_4
12. Ghosh, A., Nath, B.: Multi-objective rule mining using genetic algorithms. Inf. Sci. 163(1), 123–133 (2004)
13. Han, J., Pei, J., Yin, Y., Mao, R.: Mining frequent patterns without candidate generation: a frequent-pattern tree approach. Data Min. Knowl. Disc. 8(1), 53–87 (2004)
14. Heraguemi, K.E., Kamel, N., Drias, H.: Association rule mining based on bat algorithm. In: Pan, L., Päun, G., Pérez-Jiménez, M.J., Song, T. (eds.) BIC-TA 2014. CCIS, vol. 472, pp. 182–186. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45049-9_29
15. Heraguemi, K.E., Kamel, N., Drias, H.: Multi-population cooperative bat algorithm for association rule mining. In: Núñez, M., Nguyen, N.T., Camacho, D., Trawinski, B. (eds.) ICCCI 2015. LNCS (LNAI), vol. 9329, pp. 265–274. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24069-5_25
16. Heraguemi, K.E., Kamel, N., Drias, H.: Multi-objective bat algorithm for mining interesting association rules. In: Prasath, R., Gelbukh, A. (eds.) MIKE 2016. LNCS (LNAI), vol. 10089, pp. 13–23. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-58130-9_2

17. Holt, J.D., Chung, S.M.: Efficient mining of association rules in text databases. In: Proceedings of the Eighth International Conference on Information and Knowledge Management, pp. 234–242. ACM (1999)

18. Indira, K., Kannmani, S.: Mining association rules using hybrid genetic algorithm and particle swarm optimisation algorithm. Int. J. Data Anal. Tech. Strat. 7(1), 59–76 (2015)

19. Klemettinen, M.: A knowledge discovery methodology for telecommunication network alarm databases (1999)

20. Kuo, R.J., Chao, C.M., Chiu, Y.: Application of particle swarm optimization to association rule mining. Appl. Soft Comput. 11(1), 326–336 (2011)

21. Mlakar, U., Zorman, M., Fister Jr., I., Fister, I.: Modified binary cuckoo search for association rule mining. J. Intell. Fuzzy Syst. 32(6), 4319–4330 (2017)

22. Moslehi, P., Bidgoli, B.M., Nasiri, M., Salajegheh, A.: Multi-objective numeric association rules mining via ant colony optimization for continuous domains without specifying minimum support and minimum confidence. Int. J. Comput. Sci. Iss. (IJCSI) 8(5), 34–41 (2011)

23. Olmo, J.L., Luna, J.M., Romero, J.R., Ventura, S.: Association rule mining using a multi-objective grammar-based ant programming algorithm. In: 11th International Conference on Intelligent Systems Design and Applications (ISDA), pp. 971–977. IEEE (2011)

24. Pei, J., Han, J., Mortazavi-asl, B., Zhu, H.: Mining access patterns efficiently from web logs. In: Terano, T., Liu, H., Chen, A.L.P. (eds.) PAKDD 2000. LNCS (LNAI), vol. 1805, pp. 396–407. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45571-X_47

25. Saggar, M., Agrawal, A.K., Lad, A.: Optimization of association rule mining using improved genetic algorithms. In: 2004 IEEE International Conference on Systems, Man and Cybernetics, vol. 4, pp. 3725–3729. IEEE (2004)

26. Sarath, K., Ravi, V.: Association rule mining using binary particle swarm optimization. Eng. Appl. Artif. Intell. 26(8), 1832–1840 (2013)

27. Satou, K., et al.: Finding association rules on heterogeneous genome data. In: Proceedings of Pacific Symposium on Biocomputing, pp. 397–480. Citeseer (1997)

28. Ventura, S., Luna, J.M.: Pattern Mining with Evolutionary Algorithms. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-33858-3

29. Zaki, M.J.: Generating non-redundant association rules. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 34–43 (2000)

30. Zaki, M.J.: Mining non-redundant association rules. Data Min. Knowl. Disc. 9(3), 223–248 (2004)