



Multi-objective particle swarm optimization algorithm using adaptive archive grid for numerical association rule mining

R. J. Kuo¹ · Monalisa Gosumolo² · Ferani E. Zulvia³

Received: 7 February 2017 / Accepted: 13 November 2017 / Published online: 6 December 2017
© The Natural Computing Applications Forum 2017

Abstract

The most challenging issues in association rule mining are dealing with numerical attributes and accommodating several criteria to discover optimal rules without any preprocessing activities or predefined parameter values. In order to deal with these problems, this paper proposes a multi-objective particle swarm optimization using an adaptive archive grid based on Pareto optimal strategy for numerical association rule mining. The proposed method aims to optimize confidence, comprehensibility and interestingness for rule discovery. By implementing this method, the numerical association rule does not require any major preprocessing activities such as discretization. Moreover, minimum support and confidence are not prerequisites. The proposed method is evaluated using three benchmark datasets containing numerical attributes. Furthermore, it is applied to a real case dataset taken from a weight loss application in order to discover association rules in terms of the behavior of customer page usage.

Keywords Multi-objective optimization · Association rules · Particle swarm optimization

1 Introduction

With the rapid growth of technology, data collection has become a relatively easy task. A huge number of data can be easily collected and transformed into valuable information. Data mining is a method of retrieving and transforming raw data into meaningful information [1]. One well-known data mining technique is association rule mining. Association rule mining (ARM) is an approach to discover a valuable relationship among a set of items within a given set of data. It was first introduced by Agrawal et al. [2]. ARM produces a list of rules, made up of antecedents and consequents.

In ARM, dealing with a dataset containing categorical and quantitative attributes is a challenging problem. Most ARM methodology discretizes attributes and treats them as categorical attributes [3]. However, quantitative attributes usually have a wider range of continuous values, and a complex process is required to discretize all the attributes [4, 5]. In order to deal with numerical dataset, a discretization scheme is required to transform numerical data into categorical data. This transformation requires a predefined range or limitation, which is subjectively defined. Thus, if the range is not suitable, the result is unlikely to be accurate. In order to overcome this problem, some methods have included the discretization in their algorithm [6, 7]. This paper also proposes an association rule algorithm which can automatically categorize numerical data. The proposed algorithm therefore does not require data discretization, since it can automatically categorize each variable.

Measuring the quality of rules discovered by a data mining algorithm is a non-trivial problem. It involves several criteria, some of which are quite subjective. Ideally, the rules discovered by a data mining algorithm should be accurate, comprehensible (simple) and interesting (novel, surprising, useful) [8]. For example, a rule

✉ Ferani E. Zulvia
feranieva.z@universitaspertamina.ac.id

¹ Department of Industrial Management, National Taiwan University of Science and Technology, No. 43, Section 4, Kee-Lung Road, Taipei 106, Taiwan

² Data Science Division, Midtrans Corp., Kav. 10-11, Jend. Sudirman Road, Jakarta 10220, Indonesia

³ Department of Logistics Engineering, Pertamina University, Teuku Nyak Arief Road, Simprug, Kebayoran Lama, Jakarta 12220, Indonesia

{Pregnant} \rightarrow {Female}. This rule has a very high accuracy and can be considered as a simple and comprehensible rule. However, it is entirely uninteresting since it contains a very obvious relationship. In order to cover these objectives, it has been proposed that ARM considers multi-objective problems rather than single-objective problems.

Solving ARM as a multi-objective problem has been proposed in some previous studies. Ghosh and Nath [3] and Qodmanan et al. [9] proposed a multi-objective ARM considering three common objectives: confidence, comprehensibility and interestingness. These studies solved ARM using a multi-objective genetic algorithm. Other metaheuristic-based methods for solving multi-objective ARM have been proposed by Alatas et al. [6] and Beiranvand et al. [4]. Alatas et al. [6] considered the confidence, comprehensibility and amplitude of interval as the objectives, and solved the problem using a differential evolution-based algorithm. Recently, Beiranvand et al. [4] applied particle swarm optimization to solving multi-objective ARM.

This paper proposes a rule discovery method that uses an adaptive archive grid multi-objective particle swarm optimization algorithm (MOPSO) to find the effective rules. It considers three objectives, namely confidence, comprehensibility and interestingness. In addition, this paper is also focuses on numerical data.

The remainder of this paper is arranged as follows. Section 2 describes the basic concepts of association rules, multi-objective optimization and PSO. The proposed MOPSO for numerical association rule mining is described in Sect. 3. Section 4 presents a thorough discussion on computational experiences. Section 5 presents the case study. Finally, concluding remarks are given in Sect. 6.

2 Literature study

This section briefly reviews some theories related to association rule mining and particle swarm optimization as applied in this paper.

2.1 Association rules

Association rule mining (ARM) was first introduced by Agrawal et al. in 1993. It aims to find interesting relations between variables within a dataset. An association rule is an implication expression of the $X \rightarrow Y$, where X and Y are disjoint item sets. X is referred to as the antecedent, while Y is called the consequent. Usually, association rules are applied to datasets with Boolean attributes. Although Boolean ARM can extract meaningful information from

the data, many real data consist of categorical (e.g., sex or brand) and quantitative (e.g., age, salary or heat) data.

2.2 Multi-objective optimization

The main challenge in the real-world optimization problem is that multiple solutions may exist, and it is difficult to compare one solution with another. Thus, how to simultaneously optimize a multi-objective problem is practically relevant. This paper applies a Pareto method to optimize more than a single objective. In the Pareto method, a candidate solution that is better than all other candidates for each objective is said to dominate other candidates. None of the solutions included in the POF are better than the other solutions in the same POF for all the objectives being optimized; hence, all of them are equally acceptable.

In the maximization problem, let $F = \{f_1, f_2, \dots, f_n\}$ be a set of objective functions. A solution s belongs to the POF if there is no other solution s' that dominates it. A solution s' dominates s if and only if the following two conditions in Eq. (1) are satisfied:

$$\begin{aligned} \forall_i \in \{1, 2, \dots, n\}, f_i(s') \geq f_i(s), \text{ and } \exists_j \\ \in \{1, 2, \dots, n\}, f_j(s') > f_j(s) \end{aligned} \quad (1)$$

Figure 1 shows an example of a Pareto optimal front with two objectives f_1 and f_2 , which must be maximized simultaneously. The shaded region represents the feasible solutions. s_i represents solution i in the objective space. Solutions $s_1, s_3, s_6, s_9, s_{11}, s_{12}$ are not dominated by any other solutions among the available feasible solutions.

2.3 Metaheuristic method

Metaheuristics provide acceptable solutions in a reasonable time for hard and complex problems in science and engineering [10, 11]. Recently, metaheuristic algorithms have been applied in association rule mining [12, 13]. Table 1

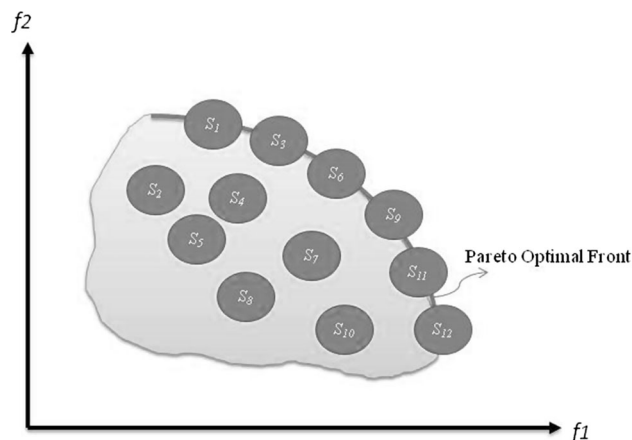


Fig. 1 Pareto front concept

Table 1 Studies related to metaheuristics in numerical ARM

Method	Basic method	Objective function	Concept
Genetic algorithm association rules (GAR) [7]	Genetic algorithm	Covered, marked and amplitude	GAR to find the frequent item sets in numerical databases and intervals of each attribute that conforms a frequent item sets
Multi-objective using genetic algorithm for negative and positive quantitative association rule mining [25]	Genetic algorithm	Covered, amplitude	GA is proposed as a search strategy for not only positive but also negative quantitative association rule (AR) mining within databases
Rough particle swarm optimization (RPSOA) [24]	Particle swarm optimization	Covered and amplitude	RPSOA based on the notion of rough patterns that use rough values defined with upper and lower intervals that represent a range or set of values
Multi-objective differential evolution algorithm (MODENAR) [6]	Differential evolution algorithm	Confidence, comprehensibility and interestingness	DE as a search strategy based on Pareto front concept for mining accurate and comprehensible numerical association rules
Chaotic particle swarm optimization algorithm [26]	Particle swarm optimization	Predictive accuracy and comprehensibility	Classification rule mining has been modeled as a multi-objective optimization problem with predictive accuracy and comprehensibility objectives
Association Rules Multi-objective genetic algorithm (ARMGA) [9]	Genetic algorithm and FP tree	Confidence, comprehensibility and interestingness	Genetic algorithm and FP tree algorithm to improve the efficiency of multi-objective association rules

summarizes some previous studies related to the application of metaheuristic algorithms in multi-objective association rule mining for numerical datasets.

2.4 Multi-objective particle swarm optimization using adaptive archive grid (MOPSO)

Particle swarm optimization (PSO) algorithm is a nature-inspired metaheuristic proposed by Eberhart and Kennedy [14]. Each particle represents a potential solution. In finding the optimal solution, PSO algorithm maintains the searching directions of all particles. Let $x_i(t)$ denote the position of particle i in the search space at time step t [15]. Each particle iteratively moves across the search space to find the best solution (fitness). Besides its current position, each particle also records its own best position, as well as the fitness value. This position is called *pbest*. Another position, called *gbest*, is the best position among all particles' positions. During the exploration process, the particles update their velocity (v_i), *pbest* and *gbest*. The velocity is updated using Eq. (2):

$$\vec{v}_i(t + 1) = w\vec{v}_i(t) + r_1c_1(\vec{pbest} - \vec{x}_i(t)) + r_2c_2(\vec{gbest} - \vec{x}_i(t)), \tag{2}$$

where w is an inertia weight to control the exploration and exploitation abilities of the particle. r_1 and r_2 are two random numbers uniformly distributed in the range of [0, 1], while c_1 and c_2 are two acceleration constants which usually lie between [1, 4].

Multi-objective optimization (MOO) differs from single-objective optimization. This study applies a multi-objective optimization inspired by Coello et al. [16]. The main objective of an archive in multi-objective optimization algorithms is to track all of the non-dominated solutions found so far. The MOPSO is proposed based on the Pareto archive strategy. It uses an adaptive grid concept, in which the objective space is separated into a number of hypercubes. The edge length of these hypercubes, l_k , is calculated in Eqs. (3)–(5):

$$l_k = \frac{\alpha(f_{\max,k} - f_{\min,k})}{nGrid}, \tag{3}$$

where

$$f_{\max,k} = \max_{a=1,\dots,A.n_s} \{f_k(x_a) | x_a \in A\}, \tag{4}$$

$$f_{\min,k} = \min_{a=1,\dots,A.n_s} \{f_k(x_a) | x_a \in A\}, \tag{5}$$

where f_k is the value of objective function k , $f_{\max,k}$ and $f_{\min,k}$ are the maximum and minimum values of objective function k , $\alpha \in (0, 1)$ is the parameter controlling the ratio of objectives' value and grid size, $nGrid$ is the grid size and A denotes the archive.

A truncated archive is used to store non-dominated solutions. For each iteration, if the archive is not yet full, a new particle position representing a non-dominated solution is added to the archive. However, because of the size limit of the archive priority is given to new non-dominated solutions located in less populated areas. In the case that

members of the archive must be deleted, members in densely populated areas have the highest probability of deletion. More densely populated hypercubes have a lower score. Roulette wheel selection is then used to select a hypercube, H_h , based on the selection fitness values. The global guide for particle i is selected randomly from among the members of hypercube H_h . Particles will therefore have different global guides.

3 Methodology

This paper proposes a multi-objective particle swarm optimization algorithm using an adaptive archive grid for numerical association rule mining. The original MOPSO algorithm was proposed by Coello et al. [16]. They proposed a multiple objective particle swarm optimization algorithm integrating the concept of Pareto dominance and adopted an archive controller. The integration of these two approaches decides and stores the membership of new non-dominated solutions found in each iteration. The proposed MOPSO algorithm considers three objectives: confidence, comprehensibility and interestingness.

(a) Confidence

Confidence is a standard measurement for association rules. Confidence is very important to measure how often Y appears in transactions t that contain X . The confidence value, $CONF$, can be obtained by Eq. (6):

$$CONF = \frac{\sigma(X \cup Y)}{\sigma(X)}, \tag{6}$$

where $\sigma(X \cup Y)$ is the total number of transitions that contain both X and Y , and $\sigma(X)$ is the total number of transactions that contain X only [17].

(b) Comprehensibility

Knowledge comprehensibility is a kind of subjective concept—a rule that is incomprehensible to one user may be very comprehensible to another. Nevertheless, to avoid difficult subjective issues, the data mining literatures often use an objective measure of rule comprehensibility. In this paper, the fewer the number of rules in the antecedent, the more the comprehensible it is, according to Ghosh and Nath [3]. Equation (7) is used to quantify the comprehensibility of the rules which may contain more than one attribute in the consequent:

$$\text{Comprehensibility} = \frac{\log(1 + |C|)}{\log(1 + |A \cup C|)}, \tag{7}$$

where $|C|$ and $|A \cup C|$ show the number of attributes in the consequent and the whole rule (both antecedent and consequent of the rule), respectively [5].

(c) Interestingness

Rule interestingness measures the potential for rules generated by ARM to be surprising for the user. The basic intuition about this approach is to find the potentially interesting rules over the whole dataset. For instance, $\{\text{Salary} = \text{High}\} \rightarrow \{\text{Credit} = \text{good}\}$. This rule tends not to be very surprising for the user—even though it might be an overall good rule, in the sense of being accurate and comprehensible. In contrast, a similar rule $\{\text{Salary} = \text{High}\} \rightarrow \{\text{Credit} = \text{bad}\}$ will be surprising and so potentially interesting since its prediction is the opposite of what the user would expected, given the occurrence of the condition in the rule antecedent [18]. The interestingness can be calculated by Eq. (8):

$$\text{Interestingness} = \left[\frac{SUP(A \cup C)}{SUP(A)} \right] \times \left[\frac{SUP(A \cup C)}{SUP(C)} \right] \times \left(1 - \frac{SUP(A \cup C)}{SUP(D)} \right). \tag{8}$$

The equation contains three parts. The first expression describes the probability of generating the rule based on the antecedent. The second part shows the probability based on the consequent, while the last one describes the probability of not generating the rule based on the whole dataset [5].

These objectives are computed and considered as a maximization problem based on the Pareto optimal strategy. The results of this method are measured by several measurement factors for evaluation purposes. There are at least three factors that can be evaluated in order to ensure the quality of the generated rules, namely support, amplitude and size.

(a) Support

This criterion measures the quality of a rule based on the number of occurrences of a rule in the whole dataset. A rule that has very low support may occur simply by chance. A rule with more occurrences in the dataset is considered to be of better quality. This support is evaluated over all the rules generated, using Eq. (9):

$$SUP = \frac{\sigma(X \cup Y)}{n} \tag{9}$$

(b) Amplitude

Amplitude is obtained to evaluate the mean size of the intervals extracted for each attribute in the rules; smaller amplitude is considered more interesting. The amplitude value is calculated by Eq. (10):

$$\text{Amplitude} = \frac{1}{m} \sum_{i=1}^m \frac{u_i - l_i}{\max(A_i) - \min(A_i)} \tag{10}$$

where u_i and l_i are the upper and lower bounds of interval i , respectively, A_i is attribute i and m is the number of attributes.

(c) Size

The size shows the mean number of attributes appearing in the rule result. Based on the comprehensibility objective, the fewer the attributes in the antecedent, the more understandable the rule! Thus, this factor is measured by the number of attributes involved in each rule. A smaller size indicates that the attributes involved are not too numerous.

3.1 MOPSO algorithm

The proposed MOPSO for numerical ARM consists of three parts: initialization, adaptive archive grid and particle swarm optimization (PSO) searching. Figure 2 shows the flowchart of the MOPSO algorithm.

3.1.1 Solution representation

In this MOPSO algorithm, each particle represents a rule solution. Each attribute in the particle consists of three parts. The first part of an attribute in the particle specifies which part of the rule it belongs to. If $0 \leq AC_i < 0.33$, then the attribute belongs to the antecedent. If $0.33 \leq AC_i < 0.66$, then the attribute belongs to the consequent. If $0.66 \leq AC_i \leq 1$, then the attribute belongs to neither. The second and third parts represent the upper and lower bound values of that attribute in the dataset. The upper bound is always greater than the lower bound generated in each particle. Table 2 illustrates the particle representation.

For example, suppose a dataset consists of four attributes (A, B, C, D). MOPSO represents a solution for this dataset as follows:

$$A(0.1)(0.2)(0.5) \ B(0.45)(0.8)(0.9) \ C(0.9)(0.1)(0.3) \\ \times \ D(0.36)(0.7)(0.8)$$

The expression above represents a generated rule $A \rightarrow BD$. Attribute C is not included in the rule, since the value of AC is equal to 0.9. The rule contains attribute A from 0.2 until 0.5, attribute B from 0.1 until 0.3 and attribute D from 0.7 until 0.8.

3.1.2 Initialization

The initial particles and velocities are randomly generated based on the size of the population and the aforementioned particle structure. The initial particle generation is based on [19]. The random particles are evaluated to find the potential solutions based on their objectives.

3.1.3 Adaptive archive grid

In this step, all the particle solutions are compared to obtain the non-dominated solution based on the Pareto optimality approach. According to Coello et al. [16], there are two components to maintain the repository as follows:

(a) The archive controller

This decides whether a certain solution is eligible or not. The non-dominated vectors found at each iteration in the primary population are compared (on a one-to-one basis) with respect to the contents of the external repository, which will be empty at beginning of the search. If the external archive is empty, then the current solution is accepted (Fig. 3 Case 1). If this solution is dominated by an individual within the external archive/repository, then such a solution is automatically discarded (Fig. 3 Case 2). Otherwise, if none of the elements contained in the external population dominate the solution wishing to enter, then such a solution is stored in the external archive (Fig. 3 Cases 3 and 4). Finally, if the external population has reached its maximum allowable capacity, then the adaptive grid procedure is invoked (Fig. 3 Case 5).

(b) The grid

In order to produce well-distributed Pareto fronts, this approach uses a variation of the adaptive grid proposed by Knowles and Corne [20]. The basic idea is to use an external archive to store all the solutions that are non-dominated with respect to the contents of the archive. In the archive, the objective function space is divided into regions, as shown in Fig. 4. Note that if the individual inserted into the external population lies outside the current bounds of the grid, then the grid must be recalculated, and each individual within it must be relocated, as shown in Fig. 5.

The adaptive archive grid is a space formed by hypercube. The detailed steps for the adaptive archive grid are as follows:

- Step 1* Obtain and store all non-dominated particles in the archive.
- Step 2* The archive controller determines which particles must be inserted into the repository or archived, based on the rules in Fig. 3. The archive controller will allow all particle values to be added into the archive, since it is an empty archive, namely *rep*.
- Step 3* Generate a hypercube of the search space explored so far, and locate the particles using these hypercube as a coordinate system, where each particle's coordinates are defined according to the values of its objective functions.

Fig. 2 Multi-objective particle swarm optimization algorithm using adaptive archive grid

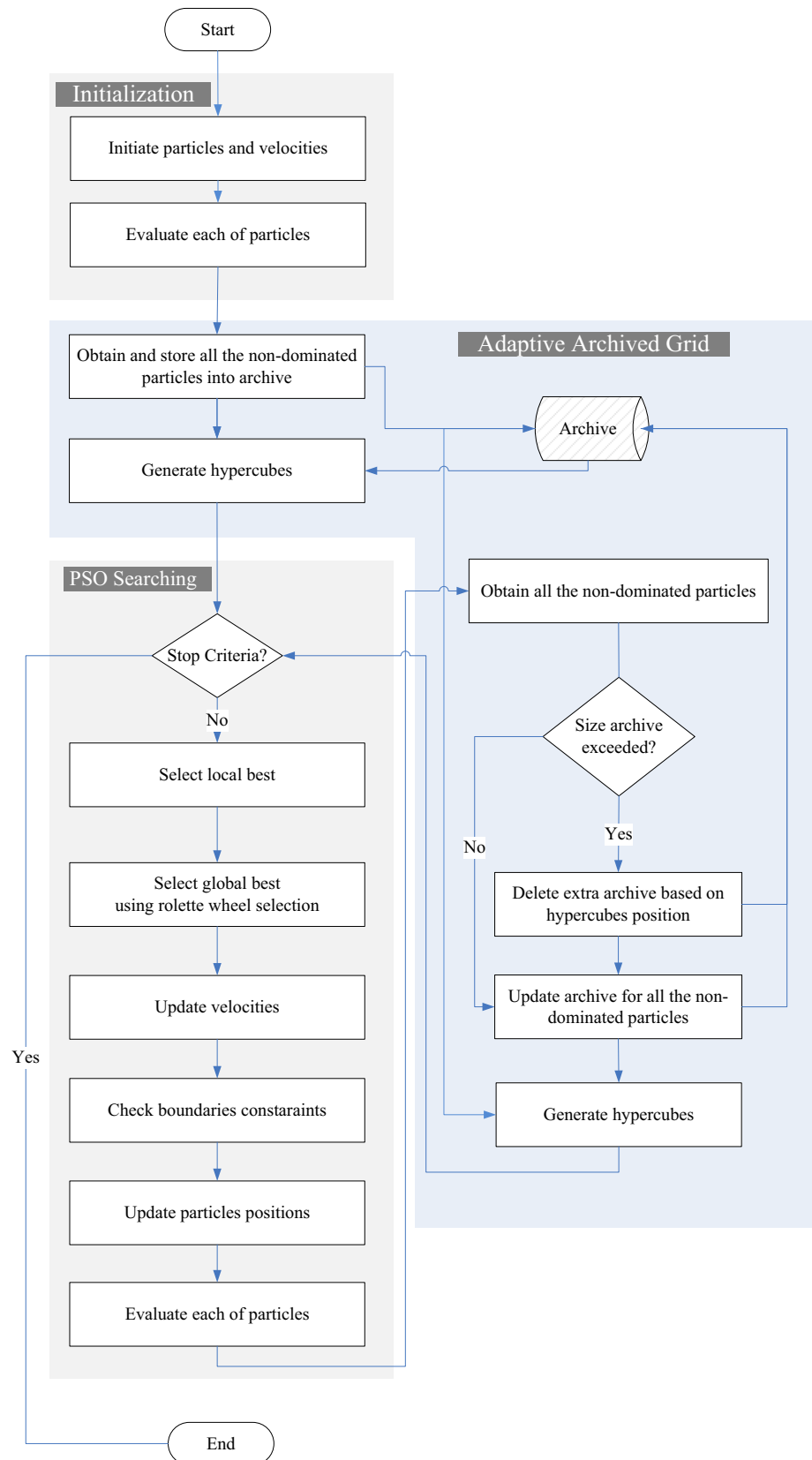


Table 2 Solution representation

Attribute 1			Attribute 2			...			Attribute n		
AAC ₁	LLB ₁	UUB ₁	AAC ₂	LLB ₂	UUB ₂	AAC _n	LLB _n	UUB _n

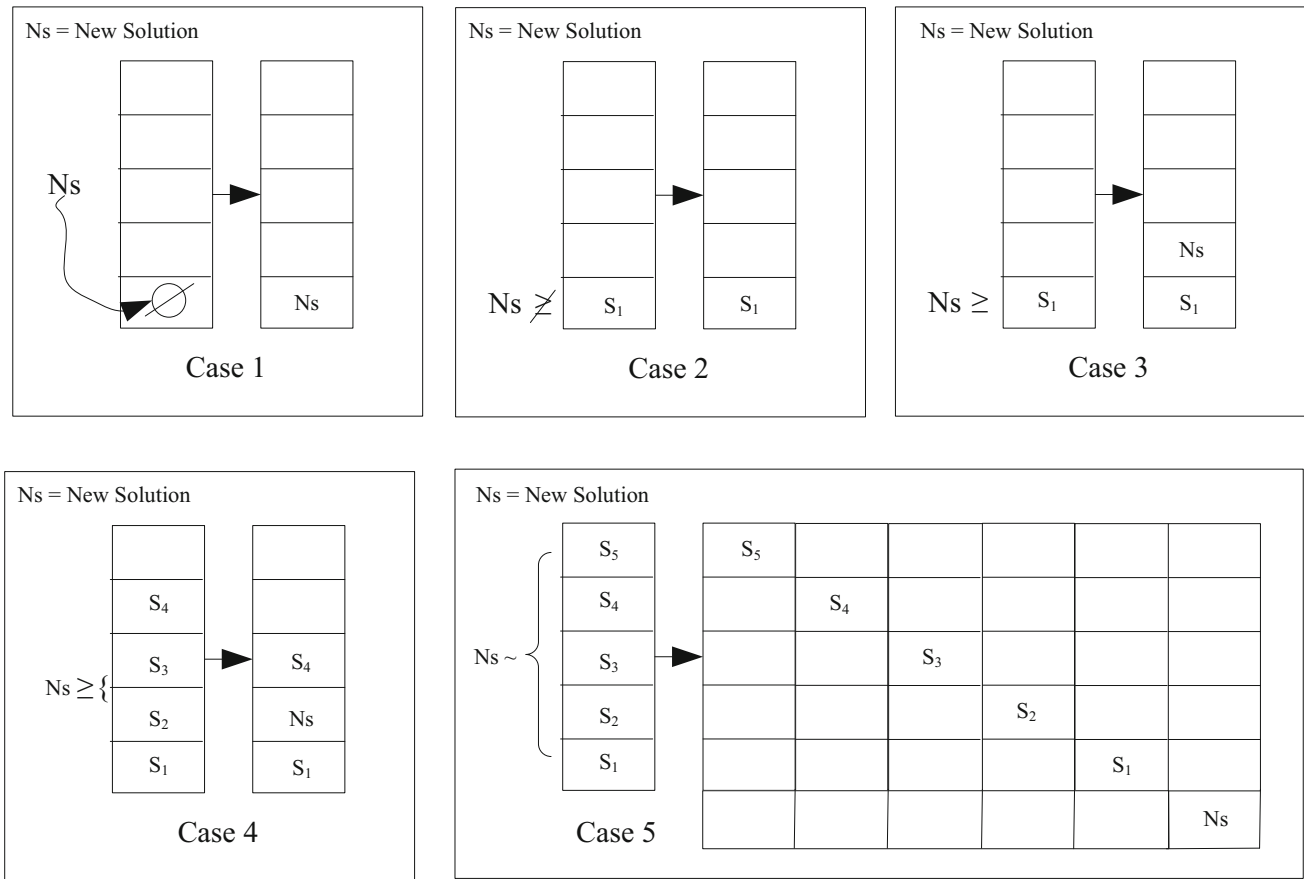


Fig. 3 Possible cases for archive controller

Step 4 Obtain all non-dominated particles. The non-dominated particles are obtained after the particles move to different positions in the search space. These new non-dominated particles are ready to be inserted into *rep*. However, the size of these archives must be checked first, and if it exceeds the predefined size, the excess particles must be deleted based on the hypercube position. Otherwise, it will just be updating *rep*.

3.1.4 PSO searching

The PSO searching as a part of the proposed MOPSO algorithm follows the following steps:

Step 1 Stopping criteria

The stopping criterion in this MOPSO procedure is the number of iterations.

Step 2 Local best selection (*pBest*)

Local best is the memory of each particle that serves as a guide to travel through the search space. If the current particle dominates the local best, then the local best will be updated by the current particle.

Step 3 Global best selection (*rep(h)*)

The selection of global best (*rep(h)*) is made using roulette wheel selection from *rep*. The index *h* is selected in the following way: Those hypercubes containing more than one particle are assigned fitness equal to the result of dividing any number *x* (*x* > 1, in this paper, *x* = 10 [16]) by the number of particles they contain.

Step 4 Velocities update

The speed of each particle is computed using the velocity by Eq. (2).

Step 5 Boundary constraints

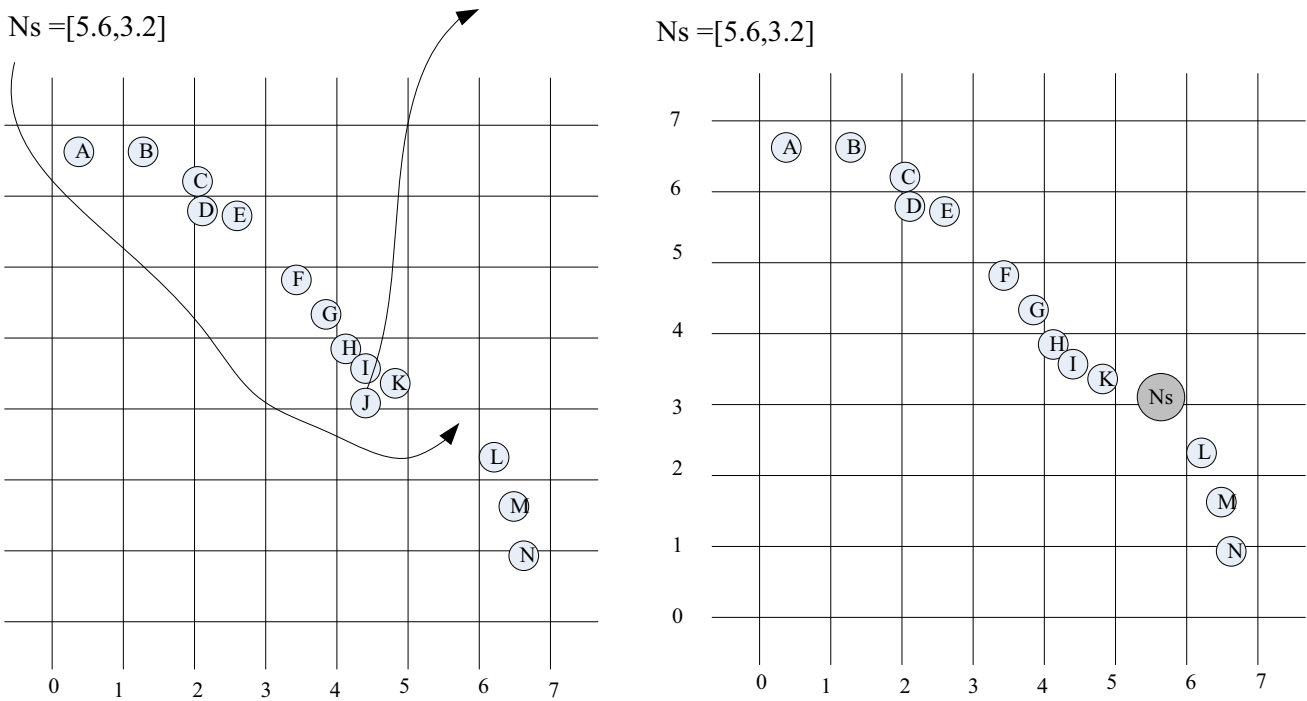


Fig. 4 Adaptive grid when the solution lies within the current boundaries of the grid

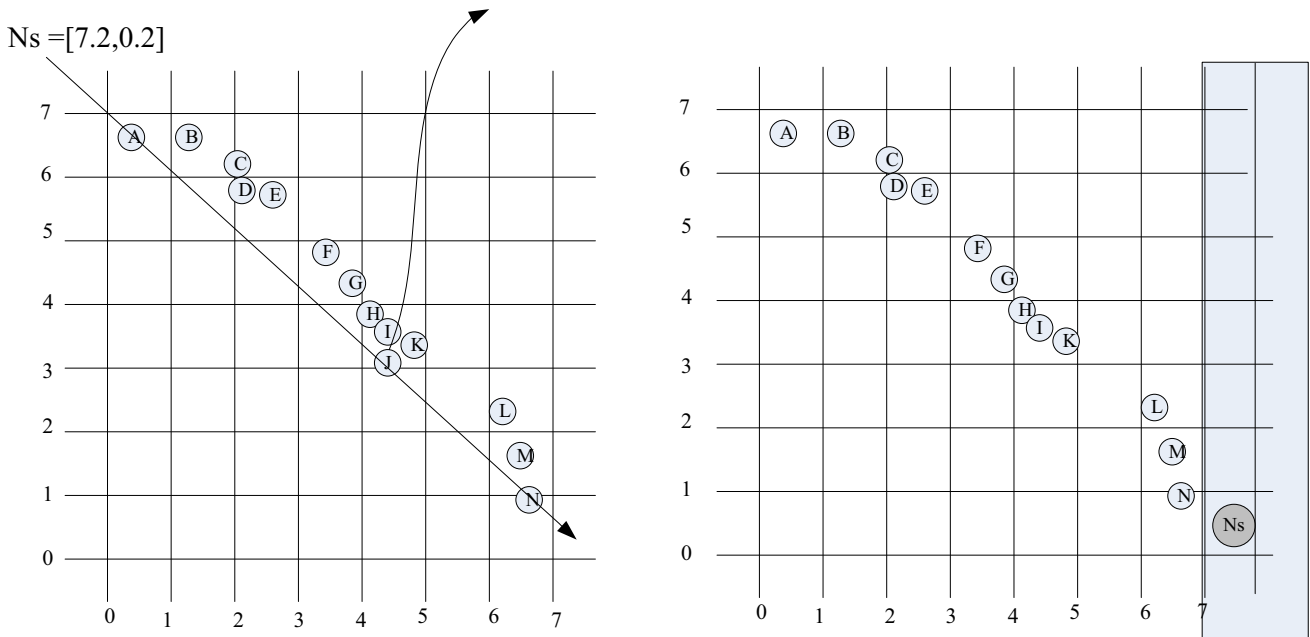


Fig. 5 Adaptive grid when the solution lies outside the previous boundaries of the grid

The boundary constraints restrict the velocity and prevent particles moving beyond the feasible search space. In this study, the particle limit is between 0 and 1. The velocity limit is from -0.5 to 5 , based on [16].

Step 6 Particle update

The new position of particle i , $pop(i)$, is updated using Eq. (11), where $v(i)$ is the velocity of particle i .

$$pop(i) = pop(i) + v(i) \tag{11}$$

Step 7 Particle evaluation

The new particle position is re-evaluated to find other potential solutions. The particle evaluation procedure is explained under Initialization.

4 Experiment results

This section presents the evaluation of the proposed method. The proposed MOPSO for numerical ARM is applied to some benchmark datasets. The evaluation covers the sensitivity analysis for the parameter setting in the MOPSO and comparison with other metaheuristic methods.

4.1 Datasets

The computational experiments were conducted using three benchmark datasets selected from the Bilkent University Function Approximation Repository. These datasets were introduced by Guvenir and Uysal [21]. Table 3 gives a description of each dataset. The algorithm is implemented in MATLAB, on a PC with an Intel Core i7 @3.40 GHz Processor and 16 GB RAM.

4.2 Parameter settings

Since the proposed algorithm involves some parameter settings, this section aims to analyze the effect of the parameter setting on the results. This paper applies a general factorial design to determine the best combination of tuning parameters. The experimental factors for PSO are inertia weight (w), learning rate 1 (c_1) and learning rate 2 (c_2), while the archive size parameter is fixed at 100. The factors and levels of determination are summarized in Table 4. The parameter settings in this paper are based on previous papers and a preliminary experiment [20, 22, 23]. Table 5 summarizes the results of general factorial design (Table 5).

Analysis of variance is employed to conduct a statistical test of the MOPSO tuning parameters effect model. The statistic results given in Table 6 show that for the Basketball dataset, c_1 and interaction between w and c_1 significantly influence the results. The result shows that a

Table 3 Brief information of benchmark data. Source: <http://funapp.cs.bilkent.edu.tr/DataSets/>

Dataset	No. of records	No. of attribute
Basketball	96	5
Body Fat	252	18
Quake	2178	4

Table 4 Factors and levels

Factors	Level 1	Level 2
w	0.3	0.8
c_1	1	2
c_2	1	2

smaller c_1 and w lead to better results for the Basketball dataset. For the Body Fat dataset, w and interaction between w and c_1 significantly influence the results. Better results are obtained with higher values of w and c_1 . On the other hand, for the Quake dataset, different parameter settings do not significantly affect the results. The best parameter settings for each dataset are shown in Table 7.

4.3 Experimental results and analysis

In order to obtain the best solution, 20 independent runs were conducted using the optimal parameters. The results were then compared with other algorithms. Parameter settings for these algorithms were optimized in previous studies [4, 5, 7, 24]. In this paper, the proposed MOPSO algorithm is compared with multi-objective PSO algorithm for association rules (MOPAR) [4], multi-objective differential evaluation algorithm for mining numerical association rules (MODENAR) [6], genetic algorithm association rules (GAR) [7], multi-objective genetic algorithm (MOGAR) [5] and rough particle swarm optimization (RPSOA) [24]. The first comparison considers the support result. It is then followed by confidence, amplitude, size and the number of rules.

Table 8 shows the average support value for each method over benchmark datasets. It shows that the MOPSO algorithm obtains the best support for the Quake dataset. The best algorithms for the Body Fat and Basketball datasets are the MODENAR and MOGAR algorithms, respectively. In terms of confidence, shown in Table 9, the proposed MOPSO algorithm achieves better results for two of the three tested datasets. These results also show that among the five algorithms, three have relatively higher confidence. These three algorithms are the MOPSO, MOPAR and MOGAR algorithms. This is because they use confidence as one of their objective functions.

Furthermore, the amplitude is presented in Table 10. The proposed MOPSO algorithm only can obtain the best amplitude value for the Body Fat dataset. For the Basketball dataset, the best result is given by the MOPAR algorithm, while for the Quake dataset, the best result is given by MODENAR. For the Basketball and Quake datasets, the MOPSO algorithm is the second best algorithm. Similar results were also obtained for the size measure presented in Table 11. Additional information about the results is summarized in Tables 12 and 13.

Table 5 Average MOPSO solution on different tuning parameters

Tuning parameters		w	0.3			
	c_1	1	1	2	2	
	c_2	1	2	1	2	
Dataset	Basketball	41.02184214	36.86786	40.23458	38.73628	
	Body Fat	37.23948878	31.89661	30.1911	30.06441	
	Quake	40.72394392	44.08425	44.83679	43.28271	
Tuning parameters		w	0.8			
	c_1	1	1	2	2	
	c_2	1	2	1	2	
Dataset	Basketball	40.30381933	40.37121	38.39427	31.796	
	Body Fat	33.08585292	30.15898	42.04037	40.54267	
	Quake	40.54886285	42.79836	43.93529	43.16952	

Table 6 Statistic test for parameter analysis (p value)

Parameter	Basketball	Body Fat	Quake
w	0.237	0.023	0.562
c_1	0.065	0.147	0.099
c_2	0.017	0.170	0.441
w and c_1	0.023	0.000	0.917
w and c_2	0.862	0.884	0.940
c_1 and c_2	0.429	0.355	0.065
w, c_1 and c_2	0.067	0.598	0.656

Table 7 Optimal tuning parameters for MOPSO

Tuning parameters	Datasets		
	Basketball	Body Fat	Quake
w	0.3	0.8	0.3
c_1	1	2	2
c_2	1	1	1

These results prove that the proposed MOPSO algorithm is a promising ARM algorithm for numerical datasets. It can generate a set of rules with high support and confidence. That the generated rules also have smaller

Table 8 Comparison of the support (%) measure in different methods

Dataset	MOPSO	MOPAR	MODENAR	GAR	MOGAR	RPSOA
Basketball	38.79	30.76	37.2	36.69	50.82 ^a	36.44
Body Fat	37.53	22.95	65.22 ^a	65.26	57.22	65.22
Quake	42.21 ^a	31.97	39.86	38.65	30.12	38.74

^aThe best result

Table 9 Comparison of the confidence (%) measure in different methods

Dataset	MOPSO	MOPAR	MODENAR	MOGAR	RPSOA
Basketball	89.27	95 ^a	61	83	60
Body Fat	91.69 ^a	81.8	62	85	61
Quake	93.40 ^a	89.32	63	82	63

^aThe best result

Table 10 Comparison of the amplitude measure in different methods

Dataset	MOPSO	MOPAR	MODENAR	GAR	RPSOA
Basketball	15.84	5 ^a	19	25	19
Body Fat	4.25 ^a	15.84	25	29	25
Quake	19.25	94.15	17 ^a	25	17 ^a

^aThe best result

amplitude values shows that they are interesting rules. Furthermore, the attributes involved in the rules are not too numerous.

In addition, the experimental results reveal that PSO-based and DE-based ARM algorithms can give better

Table 11 Comparison of the size measure in different methods

Database	MOPSO	MOPAR	MODENAR	GAR	RPSOA
Basketball	3.35	2.81 ^a	3.21	3.38	3.21
Body Fat	5.18 ^a	7.79	6.87	7.45	6.94
Quake	2.8	2.79	2.03 ^a	2.33	2.22

^aThe best result**Table 12** Comparison of the number of rules in different methods

Dataset	MOPSO	MOPAR	MODENAR	GAR	RPSOA
Basketball	41	69.75	48	50	34.2
Body Fat	65	70.8	52	84	46.4
Quake	29	54.1	55	44.87	46.4

Table 13 MOPSO summary result

Datasets	No. of rules	Support (%)	Confidence	Comprehensibility
Basketball	41	38.8	0.89	0.82
Body Fat	65	37.53	0.92	0.63
Quake	29	42.22	0.93	0.73

Datasets	Interestingness	Amplitude	Size
Basketball	0.33	15.85	3.65
Body Fat	0.28	4.26	8.27
Quake	0.3	19.26	2.8

results than can GA-based ARM algorithms. This might be because, when generating a set of rules, the algorithm needs to conduct deep searches. Both the PSO and DE algorithms only have a high capability to focus on narrow exploitation; however, the GA algorithm has a mutation operator. This operator is very important for avoiding local optima and performing large exploration. However, for problems with fewer local optima, or if the optimal solution is located in a narrow location, the mutation operator may become a disadvantage. Therefore, from these results, PSO-based and DE-based algorithms have relatively better results. Furthermore, the objective function also has a significant effect on the result. For instance, the MOPSO, MOPAR and MOGAR algorithms use confidence, interestingness and comprehensibility as their objective functions. Therefore, they have good results in terms of confidence and support. On the other hand, MODENAR includes amplitude in its objective functions, and its results have low amplitude.

5 Case study

The performance evaluation result presented in Sect. 4 reveals that the proposed MOPSO algorithm can perform better than some other algorithms. Thus, this paper applies the proposed algorithm to a real-world problem. This case study is obtained from a domestic industry that developed a healthy weight loss application (app). This app provides services to help users manage their personal health using a calorie counter, calorie-controlled meals and location information, and other functions for maintaining body weight. In order to identify the behavior of customers using this app, association rule mining is very useful in determining the relations between each function within the app. The app dataset is the click log history consisting of 38 page functions or attributes that would be considered as attribute inputs for the MOPSO. In addition to page attributes, there are several other important attributes.

5.1 Preprocessing

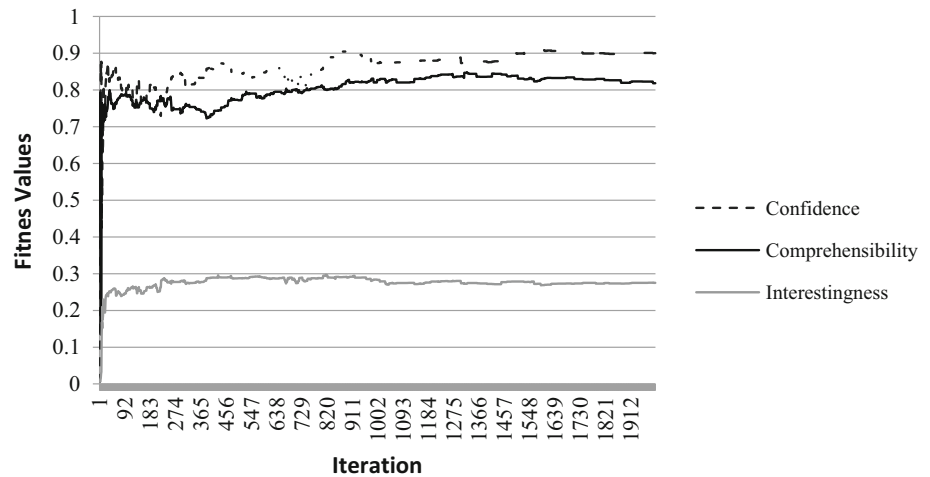
The data preprocessing involves data normalization and attribute selection. Not all attributes are included in the data processing because some attributes have zero value. The attribute selection is conducted based on the following steps:

- Step 1* Calculate the percentages of zero values for each attribute.
- Step 2* Conduct observation for three combinations by eliminating attributes that have more than 90, 80, 70 and 60% zero values.
- Step 3* Compare the results and determine the selection attributes.

Table 13 lists the alternatives of the selected attributes. The total number of attributes in solutions 4 and 5 are similar. The same situation occurs for solutions 3 and 4. However, the difference in number of attributes left between solutions 2 and 3 is significant. In other words, if solution 2 is picked, the overall attributes will contain more attributes with zero value than without zero value, since most zero values lie between 80 and 90%. Therefore, solution 3 is considered the best solution.

5.2 MOPSO implementation

In applying the MOPSO algorithm for the case study, the parameter combination for w , c_1 , c_2 used follows the parameter setting for the Body Fat dataset. The values are 0.8, 2 and 1, respectively. This parameter combination is chosen since the number of attributes in the case study dataset is similar to that of the Body Fat dataset, and the

Fig. 6 APP fitness convergence history**Table 14** Alternatives of selected attributes

Solution	Actions	Total attribute left
1	Keep all the attributes	38
2	Keep attributes having less than 90% zero value	23
3	Keep attributes having less than 80% zero value	14
4	Keep attributes having less than 70% zero value	9
5	Keep attributes having less than 60% zero value	8

statistical analysis result indicates that these parameter combinations have significant impact for the Body Fat dataset. The convergence histories of the MOPSO algorithm for the case study data are shown in Fig. 6.

Table 14 presents the association rule results. It shows that the top rules with confidence equal to 1 reveals that users tend to visit and spend more time on *click_calories_map*. The close relation between *click_calories_map* and *suggest_a_meal* is shown by the highest support value found on rule number 1. Users tend to browse the calories maps provided by restaurants before ordering their meals. These findings are reasonable, since the calorie information in this app is very useful support for their diet or exercise program. Users also tend to use *suggest_a_meal* and *select_a_restaurant* with *exercise_information* together. Furthermore, users tend to click *select_a_restaurant* and then click *exercise_information*, as found in rule #4. There is a close relation between the calories map, weight details and exercise found in rules #6, #7 and #8.

The following conclusions can be drawn from this result:

- Users of this application tend to use several functions of this app, due to a lot of zero records being found.
- From the most visited functions, it is found that the calorie map provided by restaurants is very useful,

helping users select restaurants or to plan their exercise (Table 15).

6 Conclusions

This paper proposes a multi-objective particle swarm optimization (MOPSO) algorithm for numerical association rule mining. While most ARM algorithms are only applicable for categorical data, the proposed MOPSO algorithm includes a discretization procedure in order to process numerical data. It represents intervals for each variable in a particle to find the best interval for each dataset automatically. Furthermore, in order to obtain interesting and reliable sets of rules, this paper considers three objectives when generating the rules, namely confidence, interestingness and comprehensibility. This algorithm does away with the need for decision-makers to determine the minimum support and confidence in advance.

In order to evaluate the proposed MOPSO algorithm, three benchmark datasets were applied. The results were compared with those of four other ARM algorithms, namely MOPAR, MODENAR, GAR and RPSOA. The computational results showed that the proposed MOPSO algorithm gives better results in terms of confidence and

Table 15 List of association rules over weight loss application dataset in duration

No	Rules	Support	Confidence	Comprehensibility	Interestingness
1	<i>Click_calories_map</i> ∈ (1677.75 s)→ <i>Point_a_meal</i> ∈ (382.66 s)	61.3	1	0.63	0.24
2	<i>Point_a_meal</i> ∈ (25.08 s)→ <i>Click_the_exercise_database</i> ∈ (370.8 s)	54.52	1	0.63	0.25
3	<i>Click_election_restaurant</i> ∈ (369.08 s)→ <i>Click_the_exercise_database</i> ∈ (453.76 s)	50.28	1	0.63	0.25
4	<i>Confirm_new_exercise</i> ∈ (63.24 s)→ <i>Click_weight_details</i> ∈ (126.51 s)	49.15	1	0.63	0.25
5	<i>Click_weight_details</i> ∈ (150.2 s)→ <i>Click_calories_map</i> ∈ (1000.38 s), <i>Point_a_meal</i> ∈ (123.84 s), <i>Confirm_new_exercise</i> ∈ (87.12 s)	46.05	1	0.86	0.26

amplitude, resulting in fewer rules being generated. These results also reveal that PSO-based and DE-based ARM algorithms are more promising than GA-based ARM algorithms. This is because ARM requires more intensive exploitation to get a better set of rules. In addition, the objective functions used by the algorithm also influence the rules. Based on the experimental results, by choosing confidence, interestingness and comprehensibility as the objective functions, the rules generated might have higher support and confidence. This study also applied the proposed MOPSO algorithm to a real-world problem. The case study aims to reveal important information from the user behavior for a weight loss application. Based on the results obtained, further study should consider more objectives, such as amplitude, in order to obtain better rules. Hybrid metaheuristics also should be evaluated to generate better rules.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflicts of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

- Larose DT, Larose CD (2014) Discovering knowledge in data. An introduction to data mining. Wiley, Hoboken
- Agrawal R, Imielinski T, Swami A (1993) Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD international conference on management of data, Washington, D.C.
- Ghosh A, Nath B (2004) Multi-objective rule mining using genetic algorithms. *Inf Sci* 163:123–133
- Beiranvand V, Mobasher-Kashani M, Abu Bakar A (2014) Multi-objective PSO algorithm for mining numerical association rules without a priori discretization. *Expert Syst Appl* 41:4259–4273
- Minaei-Bidgoli B, Barmaki R, Nasiri M (2013) Mining numerical association rules via multi-objective genetic algorithms. *Inf Sci* 233:15–24
- Alatas B, Akin E, Karci A (2008) MODENAR: multi-objective differential evolution algorithm for mining numeric association rules. *Appl Soft Comput* 8:646–656
- Mata J, Alvarez J-L, Riquelme J-C (2002) Discovering numerical association rules via evolutionary algorithm. In: Pacific-Asia conference on knowledge discovery and data mining, Taipei
- Freitas AA (1998) Data mining and knowledge discovery with evolutionary algorithm. Springer, New York
- Qodmanan HR, Nasiri M, Minaei-Bidgoli B (2011) Multi objective association rule mining with genetic algorithm without specifying minimum support and minimum confidence. *Expert Syst Appl* 38:288–298
- Talbi E-G (2009) Metaheuristics from design to implementation. Wiley, New Jersey
- Arqub OA, Abo-Hammour Z (2014) Numerical solution of systems of second-order boundary value problems using continuous genetic algorithm. *Inf Sci* 279:396–415
- Heraguemi KE, Kamel N, Drias H (2016) Multi-swarm bat algorithm for association rule mining using multiple cooperative strategies. *Appl Intell* 45:1021–1033
- Cheng S, Liu B, Ting TO, Qin Q, Shi Y, Huang K (2016) Survey on data science with population-based algorithms. *Big Data Anal* 1:3
- Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science, 1995, MHS '95, pp 39–43
- Engelbrecht AP (2005) Fundamentals of computation swarm intelligence. Wiley, England
- Coello CAC, Pulido GT, Lechuga MS (2004) Handling multiple objectives with particle swarm optimization. *IEEE Trans Evol Comput* 8:256–279
- Adamo J-M (2001) Data mining for association rules and sequential patterns. Springer, New York
- Fidelis MV, Lopes HS, Freitas AA (2002) Discovering comprehensible classification rules with a genetic algorithm. In: Proceedings of the 2000 congress on evolutionary computation. IEEE, California, pp 805–810

19. Eberhart RC, Yuhui S (2001) Particle swarm optimization: developments, applications and resources. In: Proceedings of the 2001 congress on evolutionary computation, Seoul, vol 81 pp 81–86
20. Knowles J, Corne D (2000) Approximating the nondominated front using the pareto archived evolution strategy. *Evol Comput* 8:149–172
21. Guvenir DHA, Uysal I (2000) Function approximation repository. Bilkent University, Ankara, Turkey. <http://funapp.cs.bilkent.edu.tr/DataSets/>
22. Kuo RJ, Zulvia FE, Suryadi K (2012) Hybrid particle swarm optimization with genetic algorithm for solving capacitated vehicle routing problem with fuzzy demand—a case study on garbage collection system. *Appl Math Comput* 219:2574–2588
23. Kuo RJ, Kuo PH, Chen YR, Zulvia FE (2016) Application of metaheuristics-based clustering algorithm to item assignment in a synchronized zone order picking system. *Appl Soft Comput* 46:143–150
24. Alatas B, Akin E (2008) Rough particle swarm optimization and its applications in data mining. *Soft Comput* 12:1205–1218
25. Alataş B, Akin E (2005) An efficient genetic algorithm for automated mining of both positive and negative quantitative association rules. *Soft Comput* 10:230–237
26. Alatas B, Akin E (2009) Multi-objective rule mining using a chaotic particle swarm optimization algorithm. *Knowl-Based Syst* 22:455–460