Improving a multi-objective evolutionary algorithm to discover quantitative association rules

M. Martínez-Ballesteros, A. Troncoso,
F. Martínez-Álvarez, J. C. Riquelme

Abstract This work aims at correcting flaws existing in multi-objective evolutionary schemes to discover quantitative association rules, specifically those based on the well-known non-dominated sorting genetic algorithm-II (NSGA-II). In particular, a methodology is proposed to find the most suitable configurations based on the set of objectives to optimize and distance measures to rank the non-dominated solutions. First, several quality measures are analyzed to select the best set of them to be optimized. Furthermore, different strate-gies are applied to replace the crowding distance used by NSGA-II to sort the solutions for each Pareto-front since such distance is not suitable for handling many-objective problems. The proposed enhancements have been integrated into the multi-objective algorithm called MOQAR. Several experiments have been carried out to assess the algorithm's performance by using different configuration settings, and the best ones have been compared to other existing algorithms. The results obtained show a remarkable performance of MOQAR in terms of quality measures.

Keywords Association rules · Data mining · Evolutionary computation · Pareto-optimization

1 Introduction

The use of computational processing techniques has increased in the last years owing to the massive amount of data generated nowadays. In the field of data mining, the learning of association rules (AR) is a popular method used to discover interesting and apparently hidden relations among variables in large databases [2,4].

There exist many algorithms focused on the discovery of AR in datasets with discrete values such as AIS [1] or Apriori [2]. However, the domain of most of the real-world data applications is continuous [25], and in that case, the AR are known as quantitative association rules (QAR). A common strategy to deal with continuous values is to discretize the attributes, but this loss of information may lead to poor results in some contexts.

Evolutionary algorithms (EA) [28] and genetic programming [19] have been widely used to discover AR and QAR since they offer several advantages for knowledge extraction and, specifically, for rule induction processes. EA were originally designed to solve single-objective optimization problems. Nevertheless, the use of EA to solve multi-objective real-world optimization problems has increased recently. Multi-objective optimization searches for a set of non-dominated solutions (rules in this context), called the Pareto-optimal set [38], whereas the single-objective optimization seeks to obtain the best solution for a single objective.

The mining process of AR can be considered as a multi-objective problem in which the measures used to evaluate a rule are different objectives in conflict that require simultaneous optimization [12]. In this context, two goals are pursued: to discover the rules that will form the Pareto-optimal set, and moreover, these rules have to be as diverse as possible. The most common multi-objective evolutionary algorithms (MOEA, from now on) based on a Pareto-optimal set are strength Pareto-evolutionary algorithm (SPEA-II) [37] and non-dominated sorting genetic algorithm (NSGA-II) [10].

Most of the MOEA are based on a combination of weighted objectives in a fitness function. Although a weighted scheme could be solved by a single-objective optimization, several disadvantages are identified such as the best trade-off for all the objectives could not be found, all non-dominated solutions could not be identified or the solutions depend on the weight assignments. Hence, the methods based on the Pareto-optimal set improve the AR optimization process since neither weights nor previous information on the problem under analysis is required.

This work extends previous proposals which optimize a sum of weighted objectives into a multi-objective approach based on the well-known NSGA-II algorithm. In particular, a non-dominated MOEA to discover QAR, hereafter called MOQAR, is here described. MOQAR is able to find QAR in datasets with continuous attributes avoiding the discretization. This task is accomplished by using adaptive intervals and by not requiring that attributes belong to either the antecedent or the consequent of the rules.

On the other hand, a review of the literature reveals that there is no a general agreement on the selection of the best set of quality measures to evaluate QAR, that is, it does not exist a defined group of AR quality measures to be optimized. Therefore, the study proposed in [24] is used as first step to discover the most adequate objectives to be included in the fitness function in order to simultaneously optimize as many measures as possible.

Moreover, the effects on the performance of different strategies to rerank solutions without modifying other components of the evolutionary process in MOQAR are also analyzed. For this purpose, two substitute distance assignment methods based on those proposed in [17] and one new distance here proposed have been applied. These distances use different strategies to recompute the secondary ranking of the solutions.

In summary, the main contribution of this work is to introduce a new methodology to obtain the most suitable configuration settings to discover high-quality QAR by means of MOEA-based approaches. The methodology proposed attempts to provide the most suitable set of objectives to guide the search of the rules and distance assignment measures to rank the non-dominated solutions.

MOQAR has been tested on several real-world datasets of different nature to assess its performance according to the proposed techniques. Specifically, datasets from the public Bilkent University Function Approximation (BUFA) repository [16] have been used. The best configuration settings of objectives and distance assignments have been compared to those obtained by other existing MOEA to discover QAR, specifically, the MOEA proposed in [5,15] and [21], henceforth named MODENAR, MOEA_Gosh and QAR-CIP-NSGAII, respectively. Nonparametric statistical tests have been used in the performance study.

The remainder of the paper is structured as follows. Section 2 presents a summary of several methods to discover QAR mainly focused on MOEA. Section 3 provides a brief preliminary on AR. Section 4 widely describes the main features of MOQAR, the statistical analysis carried out to select the best objectives to be optimized and the distances used in the secondary ranking of MOQAR. The results achieved by the different configuration settings of MOQAR according to the selected objectives and the substitute distance assignments are discussed in Sect. 5. This section also presents a comparative study to assess MOQAR's performance versus other MOEA approaches. Finally, Sect. 6 summarizes the most relevant conclusions.

2 Related work

This section explores the different approaches recently published in the literature related to the extraction of QAR. A large amount of methods have been proposed to address the discovery of QAR by optimizing a fitness function composed of weighted objectives [22]. The authors in [26] proposed an EA-based approach, called GENAR, capable of obtaining a variable number of quantitative attributes in the antecedent of the rule. GENAR optimized a fitness function based on a sum of weighted measures such as the support and confidence measures and the number of recovered instances. The same quality measures plus the comprehensibility and the amplitude of the intervals forming the rule were included in the fitness function of the GAR-plus algorithm [28].

In [4], an EA is proposed as a search strategy to mine both positive and negative QAR within databases. The discovery of QAR was optimized by a sum weighted of objectives composed of support, confidence, number of attributes and amplitude. Another EA known as EARMGA was used in [35] in order to obtain QAR. However, the confidence was the only objective included in the fitness function. To fulfill this goal, the authors avoided the specification of the actual minimum support, which is the main contribution of this work indeed. A weighted support based on the individual weight of the items according to their importance in the dataset was calculated in [29].

Nevertheless, the majority of the approaches are focused on Pareto-based MOEA with the purpose of finding the best trade-off between two or more conflicting objectives. A thorough review of the state of the art of MOEA can be found in [36]. The QAR mining based on a multi-objective genetic algorithm (MOGA) was introduced in [15]. In that work, the comprehensibility, interest and confidence (or predictive accuracy) were the objectives to be optimized. These measures were also considered in a Pareto-based MOEA rule mining method, which used as model the GA presented in [34].

The authors in [11] proposed a fast and scalable MOGA for mining QAR from large datasets using parallel processing and a homogeneous network of workstations. The confidence, comprehensibility and interest of QAR were the target objectives.

A multi-objective Pareto-based EA called MODENAR was presented in [5]. The fitness function was composed of four different objectives: support, confidence, comprehensibility of the rule (aimed to be maximized) and the amplitude of the intervals that forms the rule (intended to be minimized). An approach for mining QAR by means of NSGA-II was proposed in [7]. Different measures were used in the evaluation, such as interestingness, comprehensibility, support, confidence and the number of attributes.

Another multi-objective EA to discover QAR is proposed in [31]. This algorithm does not take into account the minimum support and confidence and applies the FP-tree algorithm. The fitness function maximizes the correlation between support and confidence. The authors in [21] proposed a MOEA based on the NSGA-II scheme which optimizes the product between the support and confidence, the interest of the rules and the comprehensibility. Furthermore, the amplitude of the rules was also evaluated.

Finally, two approaches based on NSGA-II and SPEA-II using a grammar-guided genetic programming were presented in [20]. Both proposals obtained very frequent and reliable rules achieving an optimal trade-off in terms of support, confidence and interest.

As can be appreciated, most existing multi-objective approaches based on Pareto-optimal set extend the NSGA-II algorithm to optimize the discovery of QAR, as MOQAR does. Recently, new substitute distance arrangement mechanisms have been proposed in [17] to replace the crowding distance in NSGA-II due to its poor performance in multi-objective optimization problems. However, all the existing proposals based on NSGA-II to discover QAR typically use the crowding distance.

Table 1 summarizes the measures used for both evaluation and optimization in the works reviewed in this section. It can be noted that most of the works reviewed in this section use the support and the confidence to evaluate and optimize the quality of the rules obtained. However, the optimization of such measures is not usually enough. In fact, very general QAR could be obtained and the amplitude of the intervals could be enlarged until reaching the whole domain of each attribute if only the support is optimized. Regarding the confidence, the support of the consequent of the rule is not considered; therefore, it is not able to detect negative dependence among attributes. Other existing measures to assess the QAR quality are presented in Sect. 3.

From the review of the literature, it can be concluded that an analysis of the quality measures is needed to select the most adequate objective function. In a previous work [24], the authors proposed a methodology to automatically select a subset of measures whose optimization leads to the optimization of the entire set of measures. This work was focused on finding relations among different quality measures in order to determine which measures must be included in the fitness function. To fulfill this goal, a principal component analysis (PCA) was used to select the best subset of measures.

In the light of the above considerations, this paper attempts to overcome limitations of the existing MOEA to optimize the QAR extraction process. Moreover, new features have been included in this proposal, which will be described in the following sections.

Table 1 QAR algorithms and quality measures used in the literature

Algorithms	Quality measures considered					
	Support	Confidence	Recovered	#Attributes	Amplitude	Interest
Multi-objective algorithms using a sum of weighted objectives						
GENAR [26]	√	√	√			
Alatas and Akin [4]	√	√		√	√	
EARMGA [35]		√				
GAR-plus [28]	√	√	√	√	√	
Pears et al. [29]	√					
Pareto-based multi-objective algorithms						
Ghosh et al. [15]	√			√		√
Dehuri et al. [11]		√		√		√
Wakabi et al. [34]	√			√		√
Anand et al. [7]	√	√			√	√
MODENAR [5]	√	√		√	√	
Quodmanan et al. [31]	√	√				
NSGA-G3PARM [20]	√	√				√
SPEA-G3PARM [20]	√	√				√
QAR-CIP-NSGAII [21]	√	√		√	√	√

Table 2 Quality measures for quantitative association rules

Measures	Equation	Description	Range
$Sup(X)$	$n(X)/N$	Coverage of X	[0, 1]
$Sup(X \Longrightarrow Y)$	$n(X \cap Y)/N$	Generality of the rule	[0, 1]
$Conf(X \Longrightarrow Y)$	$sup(X \Longrightarrow Y)/sup(X)$	Reliability of the rule	[0, 1]
$Lift(X \Longrightarrow Y)$ [8]	$sup(X \Longrightarrow Y)/(sup(X) \cdot sup(Y))$	Interest of the rule	
		Value $<$ 1: X and Y negatively dependent	$[0, +\infty)$
		Value $=$ 1: X and Y independent	
		Value $>$ 1: X and Y positively dependent	
$Conviction(X \Longrightarrow Y)$ [9]	$(1 - sup(Y))/(1 - conf(X \Longrightarrow Y))$	Implication of the rule	
		Value $<$ 1: X and Y negatively dependent	$(0, +\infty)$
		Value $=$ 1: X and Y independent	
		Value $>$ 1: X and Y positively dependent	
$Gain(X \Longrightarrow Y)$ [14]	$conf(X \Longrightarrow Y) - sup(Y)$	Added value or change of support	$[-0.5, 1]$
$Certainty\ Factor(X \Longrightarrow Y)$ [32]	If $conf(X \Longrightarrow Y) > sup(Y)$:	Gain normalized, strength of the rule	
	$(conf(X \Longrightarrow Y) - sup(Y))/(1 - sup(Y))$	Value $<$ 0: X and Y negatively dependent	[-1, 1]
	If $conf(X \Longrightarrow Y) <= sup(Y)$:	Value $=$ 0: X and Y independent	
	$(conf(X \Longrightarrow Y) - sup(Y))/sup(Y)$	Value $>$ 0: X and Y positively dependent	
$Leverage(X \Longrightarrow Y)$ [30]	$sup(X \Longrightarrow Y) - sup(X)sup(Y)$	Strength of the rule	
		Value $<$ 0: X and Y negatively dependent	$[-0.25, 0.25]$
		Value $=$ 0: X and Y independent	
		Value $>$ 0: X and Y positively dependent	
$Accuracy(X \Longrightarrow Y)$ [14]	$sup(X \Longrightarrow Y) + sup(\neg X \Longrightarrow \neg Y)$	Veracity of the rule	[0, 1]

3 Association rule mining: definition and measures

This section provides a brief preliminary on AR including some definitions, in addition to some interestingness quality measures described in the literature.

The AR mining process is considered as a non-supervised learning technique to describe and explore data properties instead of predicting the class of new data. The main purpose of AR mining is to discover groups of attributes appearing frequently together in a dataset and provide comprehensive rules able to represent the existing relationships among the attributes.

Formally, AR were first defined by Agrawal et al. [1]. Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of n items or attributes, and $D = \{tr_1, tr_2, \ldots, tr_N\}$ a set of N instances in a dataset, where each tr_j contains a subset of items or attributes. Thus, a rule can be defined as $X \Rightarrow Y$, where $X, Y \subseteq I$ and $X \cap Y = \emptyset$, X and Y are known as antecedent (or left side of the rule) and consequent (or right side of the rule), respectively.

When the domain is continuous, the AR are known as QAR. In this context, let $F = \{F_1, \ldots, F_n\}$ be a set of features, with values in $\mathbb{R}$. Let A and C be two disjunct subsets of F, that is, $A \subset F, C \subset F$, and $A \cap C = \emptyset$. A QAR is a rule $X \Rightarrow Y$, in which features in A belong to the antecedent X, and features in C belong to the consequent Y, such that X and Y are formed by a conjunction of multiple Boolean expressions of the form $F_i \in [v_1, v_2]$ (with $v_1, v_2 \in \mathbb{R}$). Thus, in a QAR the features or attributes of the antecedent are related with the features of the consequent, establishing an interval of membership values for each attribute involved in the rule. The consequent Y is usually a single expression.

Alternatively, it is needed to measure the quality of the rules in order to select the best ones and evaluate the results obtained by any algorithm. Support and confidence are the most common measures to evaluate AR. The support of the rule $X \longrightarrow Y$ is the percentage of instances in the dataset that satisfy X and Y simultaneously. It is used to evaluate the generality of the rule. The confidence is the probability that instances satisfying X, also satisfy Y. In other words, it is the support of the rule divided by the support of the antecedent. Table 2 details the most popular quality measures proposed in the literature to assess the AR quality. Support and confidence measures are denoted as *sup* and *conf*, respectively, in Table 2. Furthermore, the mathematical definition, the description of the evaluated properties and the range of variability of the measures are provided. Note that $n(X)$ is the number of occurrences of the itemset X in the dataset and N is the total number of instances in the dataset.

4 Description of the algorithm

MOQAR is a non-dominated MOEA that discovers QAR for datasets with continuous attributes without a previous discretization. Adaptive intervals instead of fixed ranges are used to identify subgroups of samples whose features share certain sets of values in a continuous domain. MOQAR performs the search for the most adequate intervals, which are adjusted to find high-quality QAR. It should be noted that it is not necessary to set what variables must belong to the antecedent or to the consequent of the rules.

In the population of MOQAR, each individual constitutes a rule and each gene of an individual represents the interval bounds and the membership type of each attribute to the rule. Specifically, the representation of an individual consists of two data structures where the first structure includes all the attributes of the dataset, and the second structure indicates the membership of an attribute to the rule represented by an individual. The individual representation is described in [23]. The individuals are subject to an evolutionary process, in which specific mutation and crossover operators are applied. An iterative rule learning (IRL) [33] is performed to penalize instances already covered by rules found by MOQAR, in order to boost the covering of instances still not covered.

MOQAR reuses the individual representation and genetic operators of QARGA [23] (crossover operator and mutation operators focused on the type of the attributes or the interval of the attributes). However, MOQAR includes new features to improve the mining of QAR. The main achievements of MOQAR is that it solves the inherent drawbacks of the use of fitness functions based on a weighted objective scheme to obtain QAR as described in Sect. 1. A multi-objective approach is the best way to tackle such problems since the best trade-off among all the measures could be obtained. MOQAR is inspired by NSGA-II [10], which is a computationally fast and elitist MOEA based on a non-dominated sorting approach.

NSGA-II assigns primary and secondary rankings to solutions using two different procedures. The primary ranking is given by a non-dominated sorting and the secondary ranking by the crowding distance of the solutions [3]. The primary ranking is used to sort the solutions, and the secondary ranking is applied when solutions have equals primary rankings. Its main purpose is the evolution of the population according to the sort of the solutions in dominance fronts. The first non-dominated front is composed of the non-dominated solutions of the population (called Pareto-front), the second one is composed of the solutions dominated by one solution, the third one of solutions is dominated by two, and so forth.

The multi-objective evolutionary process of MOQAR based on the NSGA-II algorithm is thoroughly described in Sect. 4.1. Although MOQAR is a MOEA based on NSGA-II, new features have been added to improve it. These enhancements are detailed in the following sections:

- Section 4.2 describes the generation of the initial population based on the instances previously covered by rules.
- Section 4.3 presents the genetic operators of MOQAR. In particular, crossover and mutation operators focused on the type and intervals of the attribute have been applied.
- Section 4.4 explains the selection of the most representative quality measures based on a principal component analysis (PCA) in order to optimize the discovery of QAR.
- Finally, Sect. 4.5 exposes different distance criteria used in the secondary ranking of MOQAR as an alternative option to the crowding distance [10]. These distances are used in the selection of the best individual of all the non-dominated solutions of the Pareto-front at the end of the evolutionary process.

4.1 Multi-objective evolutionary model

This section describes the MOQAR's evolutionary process. The pseudocode of the algorithm is shown in Fig. 1.

The algorithm starts by creating an initial population of parents, $P_{t=0}$, with N individuals (line 4) focused on instances covered by not many rules. The individuals are evaluated by means of the measures selected as objectives (line 5). Then, the population evolves through $nGen$ generations (line 6). Because the MOQAR algorithm aims at finding the rule that presents the best trade-off among all the objectives in each iteration, the rule that belongs to the first Pareto-front F_1 with higher value in terms of the distance is selected (line 7). A solution a is better than other one b if a is located in a region with a greater crowding distance. Note that the extremal points of the Pareto-set are not considered to be selected since their crowding distances are always assigned with an infinite value. MOQAR performs an IRL process to boost the covering of instances still not covered. The instances covered by the best rule found are penalized (line 8) for this purpose. The samples covered by few

```
Algorithm MOQAR
Input 1: Maximum number of rules ($nRules$)
Input 2: Maximum number of generations ($nGen$)
Input 3: Number of individuals ($N$)
Output: Set of the best rules found ($R$)

 1: procedure 1 MOQAR($nRules, nGen, N$)
 2:     $R \leftarrow \emptyset$
 3:     for $r \leftarrow 1, nRules$ do
 4:         $P_0 \leftarrow$ initialize-population ($N$)
 5:         evaluate-objectives ($P_0$)
 6:         $P_{nGen} \leftarrow$ EA-generations ($P_0$, $N$, $nGen$)
 7:         $br \leftarrow$ best-rule($P_{nGen}$)
 8:         penalize-instances($br$)
 9:         $R \leftarrow R \cup br$
10:     end for
11:     return $R$
12: end procedure

13: procedure 2 EA-GENERATIONS($P_0$, $N$, $nGen$)
14:     for $t \leftarrow 0, nGen - 1$ do
15:         $Q_{tc} \leftarrow$ crossover ($P_t$)
16:         $Q_{tm} \leftarrow$ mutation ($Q_{tc}$)
17:         $Q_t \leftarrow Q_{tc} \cup Q_{tm}$
18:         $R_t \leftarrow P_t \cup Q_t$
19:         $\mathcal{F}= (\mathcal{F}_1, \mathcal{F}_2, ...) \leftarrow$ fast-non-dominated-sort($R_t$)
20:         $P_{t+1} \leftarrow$ make-next-population($\mathcal{F}$, $N$)
21:     end for
22:     return $P_{nGen}$
23: end procedure

24: procedure 3 MAKE-NEXT-POPULATION($\mathcal{F}$, $N$)
25:     i $\leftarrow$ 1
26:     $P_{t+1} \leftarrow \emptyset$
27:     while $|P_{t+1}| + |\mathcal{F}_i| <= N$ do
28:         $P_{t+1} \leftarrow P_{t+1} \cup \mathcal{F}_i$
29:         $i \leftarrow i + 1$
30:     end while
31:     if $|P_{t+1}| < N$ then
32:         crowding-distance-assignement($\mathcal{F}_i$)
33:         sort($F_i$, $\prec_n$)
34:         $P_{t+1} \leftarrow P_{t+1} \cup \mathcal{F}_i[1 : (N - |P_{t+1}|)]$
35:     end if
36:     return $P_{t+1}$
37: end procedure
```

Fig. 1 Pseudocode of the MOQAR algorithm

rules have a higher priority to be selected for the initial population. The whole evolutionary process is repeated until the desired number of rules is reached and the best rules found are returned by MOQAR (line 11).

During the evolutionary process (procedure 2), an offspring population Q_t, of same size as P_t, is generated for each generation (line 17) by applying the crossover (line 15) and mutation (line 16) operators to a selection of individuals from P_t. Note that the individuals are selected using binary tournament-based selection method [27]. Later, the individuals of P_t and Q_t are merged into R_t (line 18) and the population R_t is ranked using the fast non-dominated sort method (line 19), which consists in sorting the individuals of a population in different Pareto-fronts $F = (F_1, F_2, \ldots)$ according to the level of non-domination (primary ranking). Therefore, each solution must be previously compared to every other solution in the population to check if it is dominated or not. The next population P_{t+1} (line 20) consists of the N best individuals of R_t. The evolutionary process ends when the desired number of generations is reached (line 22).

The building process of the population for the next generation (procedure 3) works until P_{t+1} (lines 27–30) is completed with N individuals as follows: the individuals of the ith Pareto-front F_i are added to the population P_{t+1} if the number of individuals of population P_{t+1} is less or equal to N individuals when the individuals of the ith Pareto-front F_i are added (line 28). In other case, the crowding distance assignment (line 32) is used to sort the population of F_i (line 33), and the $N - |P_{t+1}|$ best individuals are selected to fill the population of next generation P_{t+1} (line 34) (secondary ranking).

4.2 Generation of the initial population based on ranking

The initial population in MOQAR is generated at the beginning of each evolutionary process and is performed such that at least one sample of the dataset is covered. The samples of the dataset are selected based on their level of hierarchy. The hierarchy is organized according to the number of rules which cover a sample. Thus, the records are sorted by the number of rules they satisfy and the samples covered by few rules have the highest priority.

A sample is selected according to the inverse of the number of rules which cover the sample. Intuitively, the process is similar to the roulette selection method where the parents are selected depending on their fitness function. In this selection method, a sample is represented by a portion of roulette inversely proportional to the number of rules that cover the sample. Therefore, the samples covered by few rules have a wider portion of roulette and the probability of selection is higher. All samples are equally probable to be selected when generating the initial population for the first time.

The process that generates the initial population is listed below:

1. For all instances of the dataset, the cumulative sum *totalSum* of the inverse of the number of rules that cover every instance is computed.
2. A random number between 0 and *totalSum* is generated.
3. For each instance of the dataset, if the *totalSum* is greater or equal than the random number, then the current example is selected.

The selected example is used to generate a new individual of the initial population as follows:

- First, the number of attributes which belong to the rule is randomly selected according to a maximum and minimum number of attributes given by the user.
- Taking into account that number, the number of attributes in the antecedent and consequent of the rule are randomly selected limited by a maximum and minimum number of attributes given by the user.
- Then, the attributes to appear in the antecedent and the consequent are randomly selected from those that belong to the dataset considering the structure of the rule. The user can define whether there are attributes fixed or not fixed in the consequent of the rule.
- Finally, an upper and lower bound are randomly generated to create the interval for each attribute selected taking into account the instance selected in the above-described process.

This process is repeated until completing the number of individuals that compose the initial population.

4.3 Genetic operators

An elitist strategy replicating the individuals with the best fitness and a binary tournament-based selection method [27] for the remaining individuals of the population is applied in order to select the individuals to be parents.

The genetic operators implemented in MOQAR are the crossover and the mutation described in [23]. The mutation process consists in modifying the genes of the randomly selected individuals according to a probability. The mutation of a gene can be focused on the type of the attribute or the interval of the attribute. The choice between the mutation of the type or the interval depends on a given probability. The limits of the intervals and types of the attributes of the offspring are checked to ensure those represent meaningful rules. If any attribute does not fulfill the required constraints related to the type of attributes, the individual is discarded and a new mutation is applied to the same original individual.

In addition, the directional mutation operator, focused on the type of the attribute, has been added. If the selected attribute belongs to the antecedent of the rule, i.e., the attribute membership type is 1, then the type of the attribute changes to belong to the consequent of the rule and the new type is 2. On the contrary, if the selected attribute belongs to the consequent, i.e., the attribute membership type is 2, then the type of such attribute changes to the antecedent and now the type is 1.

4.4 Principal component analysis

A statistical analysis for the quality measures described in Sect. 3 has been applied to select the best subset of measures to be optimized by MOQAR, according to the study carried out in [24]. Following with that study, QARGA [23] has been applied to 32 datasets from the BUFA repository described in Sect. 5.1. In particular, three sets of experiments have been performed to obtain diversity in the input data of PCA. The first set of experiments has used a minimum support threshold equal to 0 to obtain all the QAR found by QARGA. The second and third sets have used a minimum support threshold equal to 0.05 and 0.1, respectively, to obtain only those QAR with a support greater or equal to the minimum support.

All configurations of QARGA have been executed by modifying the weight w_a, corresponding to the attribute number measure, in order to include or not this measure in the fitness function of QARGA. QARGA was run 6 times (3×2) for each dataset, that is, 192 times in total. Note that a maximum number of 100 QAR has been obtained for each configuration and dataset. Thus, a matrix composed of 15,487 rows (192 × the number of QAR obtained by each configuration that satisfy the minimum threshold), and one column for each quality measure considered has been generated as PCA's input. Once PCA has been applied to the input data, one measure representing each component is selected. Thus, all selected measures will be the objectives used by MOQAR as fitness function. Figure 2 illustrates the complete process including the application of QARGA and PCA to obtain the objectives to optimize as first step before executing MOQAR. The results obtained after applying PCA are presented and discussed in Sect. 5.3.

4.5 Substitute distance assignments

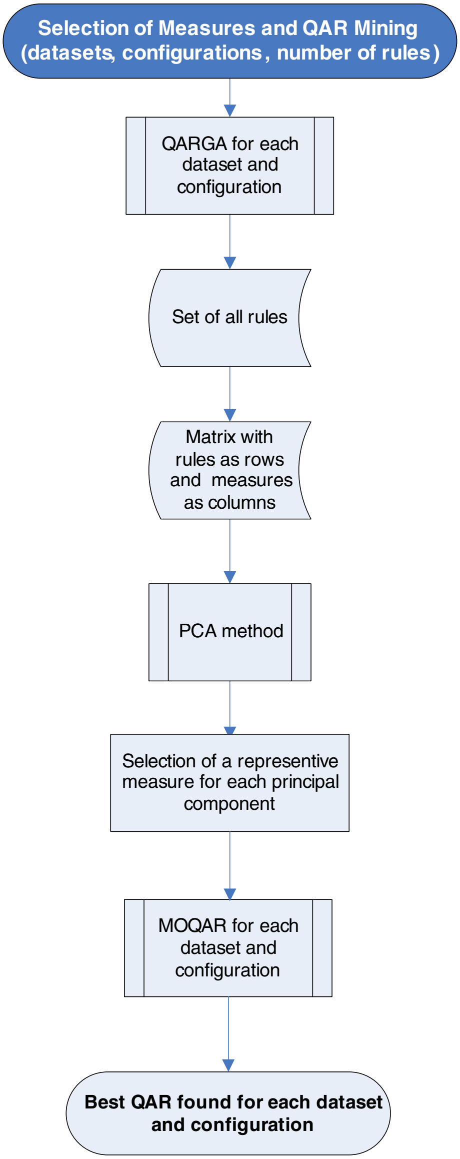
As defined in Sect. 4.1, a two-stage ranking is performed to define an order over the individuals of the population in the Pareto-dominance fronts of the NSGA-II algorithm [10].

The primary ranking considers that two individuals are equal if they both belong to the same Pareto-front, that is, if the individuals are dominated by the same number of solutions. In the case of a maximization problem, an individual x dominates another different individual y (denoted $x < y$) if the values of all objectives for x are greater than those for y. That is,

$$x < y \iff f_i(x) \geq f_i(y), \quad \forall i \in \{1, 2, \ldots, M\} \tag{1}$$

where f_i is the ith objective and M is the maximum number of objectives to be maximized.

Fig. 2 General process of MOQAR including the application of QARGA and PCA to obtain the objectives to be optimized


Selection of Measures and QAR Mining (datasets, configurations, number of rules)
QARGA for each dataset and configuration
Set of all rules
Matrix with rules as rows and measures as columns
PCA method
Selection of a representive measure for each principal component
MOQAR for each dataset and configuration
Best QAR found for each dataset and configuration

When the individuals belong to the same Pareto-front, then the secondary ranking establishes differences among them by using a distance [36]. Individuals with low distance values are desirable. NSGA-II applies the crowded-comparison operator $\prec_n$ to guide the selection of the individuals which will belong to the Pareto-optimal front [10]. The order $\prec_n$ between two individuals x and y of the population, taking into account the result obtained by the primary ranking x_r and the crowding distance x_{dist}, can be defined as follows:

$$x \prec_n y \iff (x_r < y_r) \quad \text{or} \quad (x_r = y_r \text{ and } x_{dist} < y_{dist}) \tag{2}$$

However, the dominance among solutions becomes more and more unlikely when the number of objectives increases [17]. This fact is due to the decrease in the selective pressure in NSGA-II because the proportion of non-dominated solutions in the population is increased. Therefore, several substitute distance assignment measures have been proposed in [17] to be used instead of crowding distance to improve the NSGA-II convergence. These distance measures are focused on several criteria according to the degree in which a solution A is nearly dominated by a solution B. For instance, the number of objectives with smaller or larger values, the difference between the values of different objectives or a multi-criterion based on the former ones can be considered.

Several distances are now described to replace the crowding distance for the secondary ranking in NSGA-II and therefore in MOQAR. Specifically, the subvector dominance assignment (henceforth svDominance) and the epsilon dominance assignment (henceforth eDominance), proposed in [17], assess the highest degree in which a solution is nearly Pareto-dominated by any other solution. On the other hand, the individuals dominance assignment (henceforth indDominance) is a novel measure, proposed in this work for the first time based on the number of individuals dominated for each individual. All the distances described in this section have been included in MOQAR, as well as the crowding distance. The experimental results for all distances considered are reported in Sect. 5.4.

4.5.1 Subvector dominance assignment (svDominance)

Let A and B be two solutions of a set S of solutions, the subvector dominance distance counts the number of objectives of a solution B that are better than the corresponding objectives of A. It is applied to every individual $B \neq A$ that belongs to S, and the largest value is designated as the distance of the solution A. The svDominance distance is computed as follows [3]:

$$svd(A, B) = count\{f_i(B) > f_i(A), \quad i = 1, 2, \ldots, M\} \tag{3}$$

where f_i is the ith objective and M is the maximum number of objectives to be optimized. The secondary ranking d_A^S of A is given by:

$$d_A^S = max\{svd(A, B) \quad \forall B \in S \quad B \neq A\} \tag{4}$$

The smaller the distance value is, the better the solution is, and therefore, a smaller d_A^S implies a better secondary ranking.

4.5.2 Epsilon dominance assignment (eDominance)

Let A and B be two solutions of a set S of solutions, and the epsilon dominance distance considers all the objectives of B that are worse than the corresponding objectives of A. Then, it calculates the necessary largest magnitude $epsilon$ for solution B to become a solution Pareto-dominating of A. It is applied to all the individuals $B \neq A$ which belong to S, and the smallest value is designated as the distance of the solution A. The eDominance distance is computed as follows [3]:

$$mepsd(A, B) = max\{f_i(A) - f_i(B), i = 1, 2, \ldots, M\} \tag{5}$$

where f_i is the ith objective and M is the maximum number of objectives to be optimized. The secondary ranking d_A^S of A is given by:

$$d_A^S = min\{mepsd(A, B) \quad \forall B \in S \quad B \neq A\} \tag{6}$$

The larger the distance value is, the better the solution is, and the higher the difficulty is to make the other solutions Pareto-dominating of A. Therefore, a larger d_A^S implies a better secondary ranking.

4.5.3 Individuals dominance assignment (indDominance)

Let A and B be two solutions of a set S of solutions, and the individuals dominance distance calculates the difference between the number of solutions dominated by a solution A and the number of solutions dominated by a solution B, taking into account the dominance definition previously described in this section. It is applied to all the individuals $B \neq A$ which belong to S and the largest value is designated as the distance of the solution A. The indDominance distance is computed as follows:

$$ind(A, B) = \{|P_A| - |P_B|\} \tag{7}$$

where P_A and P_B are the set of solutions which belong to the population of individuals dominated by the solution A and B, respectively. The secondary ranking d_A^S of A is given by:

$$d_A^S = max\{ind(A, B) \quad \forall B \in S \quad B \neq A\} \tag{8}$$

The larger the distance is, the better the solution is, and therefore, a larger d_A^S implies a better secondary ranking.

5 Experimentation

This section presents and discusses the experiments carried out to assess the MOQAR's performance for the datasets described in Sect. 5.1. Specifically, different configuration settings have been executed: first, by using the distance assignments specified in Sect. 4.5 as substitutes of the crowding distance in the secondary ranking of the multi-objective evolutionary process, and second, by optimizing different sets of objectives according to the statistical study detailed in Sect. 4.4. Furthermore, a comparative analysis is performed to compare MOQAR to other similar techniques.

This section is divided as follows. Section 5.1 describes the datasets from the public BUFA repository, in which MOQAR has been tested. Then, a summary of the main parameter settings of MOQAR and the MOEA (MODENAR [5], the MOEA proposed in [15] henceforth named MOEA_Ghosh, and the MOEA described in [21] hereinafter called QAR-CIP-NSGAII) analyzed in the comparative study can be found in Sect. 5.2. The set of measures to be optimized according to the results provided by the PCA is presented in Sect. 5.3. Section 5.4 discusses the experimental results obtained by MOQAR with different configuration settings related to distances and objectives. Finally, the performance of the best configurations of MOQAR are compared to MODENAR, MOEA_Ghosh and QAR-CIP-NSGAII in Sect. 5.5.

5.1 Datasets description

This section summarizes the datasets from BUFA repository [16] used for the experiments. The number of records and attributes of the thirty-two datasets is summarized in Table 3.

Table 3 Public datasets from BUFA repository

Dataset	Records	Attributes
Ailerons (AI)	7154	41
Baseball (BA)	337	17
Basketball (BK)	96	5
Bodyfat (FA)	252	18
Bolts (BL)	40	8
Buying (BU)	100	40
Computer Activity (CA)	8192	22
Country (CN)	122	21
College (CO)	236	21
Education (ED)	1500	44
Elevators (EV)	16,599	19
House_16H (HH)	22,784	17
Longley (LO)	16	7
Mortgage (MO)	1049	17
Normal Body Temperature (NT)	130	3
Plastic (PL)	1650	3
Pw Linear (PW)	200	11
Pollution (PO)	60	16
Pole Telecomm (PT)	9065	49
Pyramidines (PY)	74	28
Quake (QU)	2178	4
Read (RE)	681	26
School (SC)	62	20
Sleep (SL)	57	8
Stock price (SP)	950	10
Televisions (TV)	40	5
Treasury (TR)	1049	17
Triazines (TZ)	186	61
Usnews College (US)	1269	32
Vineyard (VY)	52	4
Weather Ankara (WA)	1609	11
Weather Izmir (WI)	1641	11

Note that Buying, College, Education, Read and Usnews Colleged have been preprocessed in order to deal with missing values. Namely, the K-means imputation method, proposed in [18] and available in the KEEL tool [6], has been applied.

5.2 Parameters setup of MOQAR and the algorithms used in the comparative study

Table 4 summarizes the main parameter values of MOQAR and each MOEA used in the comparative study. MOQAR is compared to MODENAR [5] (*Multi-objective differential evolution algorithm for mining numeric association rules*), MOEA_Ghosh [15] (*Multi-objective*

Table 4 Parameters setting of the algorithms

MODENAR	MOEA_Ghosh	QAR-CIP-NSGAII	MOQAR
#Executions = 50	#Executions = 50	#Executions = 50	#Executions = 50
Population size = 100	Population size = 100	Population size = 100	Population size = 100
#Evaluations = 50000	#Evaluations = 50000	#Evaluations = 50000	#Generations = 100
Amplitude factor = 2	Amplitude factor = 2	Amplitude factor = 2	#Evaluations = 10,000 per rule
Threshold = 60	Prob. mutation = 0.02	Prob. mutation = 0.1	#Rules = 100
Crossover rate = 0.3	Prob. crossover = 0.8	Difference threshold = 5	Amplitude factor = 2
$W_{sup} = 0.8$	Point crossover = 2		Prob. mutation = 0.1
$W_{conf} = 0.2$			
$W_{comp} = 0.1$			
$W_{ampl} = 0.4$			

Table 5 Matrix of rotated components obtained by PCA from BUFA datasets

Measure	Comp. 1	Comp. 2	Comp. 3
Accuracy	**0.997**	0.036	0.047
Consequent Support	−**0.982**	0.114	0.03
Gain	**0.823**	0.526	0.033
Confidence	−0.07	**0.963**	0.092
Certainty Factor	0.255	**0.886**	0.137
Leverage	0.089	0.08	**0.916**
Rule Support	−0.462	−0.042	**0.785**
Lift	0.406	0.135	−0.286
Antecedent Support	−0.214	−0.646	0.484
Amplitude	−0.008	0.019	0.284
Attributes	0.496	0.207	−0.207

rule mining using genetic algorithms) and QAR-CIP-NSGAII [21] algorithms, which are available in KEEL software. It is noteworthy that the parameter settings of the studied algorithms are those appearing as default parameter values in the KEEL tool, having followed the kind suggestions of the authors.

5.3 Selecting the measures to be optimized by MOQAR

This section details the statistical analysis performed by the analysis of PCA. This analysis has been applied to the following quality measures: support of the rule, support of the antecedent, support of the consequent, confidence, leverage, accuracy, lift, gain, certainty factor, amplitude and finally, number of attributes of the rule. Note that the conviction measure has been excluded because a positive infinitive value for this measure is obtained for most of the rules since the confidence of them is 1. Therefore, PCA cannot be computed when infinitive values are presented in the input data.

Table 5 shows the principal components extracted after the application of PCA to the thirty-two datasets as detailed in Sect. 4.4. It can be noticed that three principal components, each of them corresponding to an independent group of measures, have been obtained. Accuracy, gain and support of the consequent belong to the first group because they are the most correlated measures in the first component. Note that the second component is composed of the certainty factor and confidence measures. Finally, support of the rule and leverage belong to the third group due to the high correlation that theses measures present in the third component.

The next step is to select a representative measure for each component. The accuracy, confidence and leverage measures are selected to be optimized by MOQAR since these measures present the highest correlation in each group (0.997, 0.963 and 0.916, respectively). However, support of the antecedent has a similar correlation in the second and third component and lift does not present a high correlation in any component. Hence, these measures should be also considered as objectives since they do not belong to a specific component. Note that amplitude of the intervals and number of attributes are not considered as objectives since optimal values depend on the type of desired rules.

From the previous analysis, two set of objectives are considered to be maximized to assess the quality of the QAR obtained by MOQAR. Specifically, the first set of objectives is composed of accuracy, confidence and leverage, and the second one is composed of the lift and support of the antecedent in addition to the aforementioned three measures.

5.4 Performance of MOQAR using different distances and objectives

This section presents a comparative analysis for different MOQAR's configuration settings and for the two sets of objectives obtained in the previous section. Additionally, the crowding distance defined to sort the individuals in the secondary ranking of NSGA-II scheme has been compared to the distances described in Sect. 4.5.

The average results obtained for the fifty executions for each set of objectives selected in Sect. 5.3 and for each distance assignment described in Sect. 4.5 are summarized in Table 6. Percentage of records, support of the rule, support of the antecedent, support of the consequent, lift, confidence, accuracy, leverage, certainty factor and gain have been the measures used in order to evaluate the QAR. Note that the best configuration setting for each quality measure has been emboldened.

When the number of objectives is five (lift and support of the antecedent are included), the percentage of records decreases in all distance assignments except indDominance. A similar behavior was found for the support of the rules, support of the antecedent and support of the consequent. Lift and accuracy values are incremented when crowDistance and eDominance are applied. On the other hand, confidence, leverage, certainty factor and gain get worse results when the set of five objectives is optimized.

In general terms, the substitute distance assignments provide more specific, reliable and accurate rules when three objectives are optimized by MOQAR. It can observed that crowDistance and svDominance are the distances that present the worst performance when five objectives are maximized. Alternatively, the proposed distance assignment indDominance and eDominance are the best ones when three objectives are optimized.

A statistical analysis has been conducted to assess the performance of MOQAR according to different objectives to be optimized and distances following the nonparametric procedures discussed in [13]. In particular, the analysis is devoted to determine the best configuration setting for each measure separately and for the normalized average of all the measures.

Table 6 Average results obtained by MOQAR for each configuration setting

Obj.	Distance Sup. (%)	Rule Sup. (%)	Ant. Sup. (%)	Cons.	Lift Sup. (%)	Confidence Sup. (%)	Accuracy	Leverage	CF	Gain
3	crowDistance	19.19	20.86	26.11	35.43	85.22	91.40	0.094	0.82	0.59
	eDominance	**21.44**	22.63	**28.09**	29.33	93.61	92.16	**0.105**	0.90	0.66
	indDominance	9.09	9.36	10.48	**362.87**	**96.80**	**98.35**	0.035	**0.95**	**0.86**
	svDominance	16.34	18.19	22.93	36.53	80.75	91.55	0.072	0.76	0.58
5	crowDistance	7.43	12.59	9.50	224.99	63.61	92.78	0.025	0.59	0.54
	eDominance	7.53	10.01	10.35	78.80	71.11	94.71	0.040	0.68	0.61
	indDominance	14.47	16.00	15.94	306.62	92.28	96.99	0.033	0.85	0.76
	svDominance	12.03	**47.72**	22.01	7.74	31.31	54.26	0.020	0.19	0.09

5.4.1 Statistical tests for each quality measure separately

Friedman and Iman-Davenport tests have been applied to assess whether there are global differences in the configuration settings of MOQAR for all the measures considered. The ranking obtained for each measure for the level of significance $\alpha = 0.05$ is summarized in Table 7. The average ranking of all measures is shown in the rightmost. The best setting for each quality measure has been emboldened. It can be appreciated that the best average ranking values are obtained for the indDominance and eDominance distances when optimizing three objectives.

The statistics obtained by Friedman and Iman-Davenport tests have been greater than the critical values associated with each measure, according to the χ^2 and $F - Snedecor$ distributions. Therefore, the null hypothesis is rejected, that is, there are significant differences among the configuration settings of MOQAR for each quality measure. Hence, a post hoc statistical analysis has been performed in order to find significant differences among the results obtained for each one of the different configuration settings with respect to the best configuration setting for every quality measure.

Table 8 summarizes the results obtained by Friedman and Iman-Davenport tests and the Holm and Finner post hoc statistical tests for a level of significance $\alpha = 0.05$. The column *Measure* describes the quality measure under study. They are grouped according to the algorithm that finds the best solution. The *PCA component* column shows the number of the principal component obtained by PCA in Sect. 5.3 for each measure specified in the first column. The *Control algorithm* column details the configuration setting determined as control algorithm, that is, the best configuration setting for each measure. It is worth noting that all the measures belonging to the same PCA component share the same control configuration setting obtained for the Friedman and Iman-Davenport tests. Consequent support achieves the best values with eDominance and three objectives, but it can be observed that is negatively correlated with the first PCA component. Therefore, the optimization of the set of three objectives using the indDominance distance should lead to the optimization of all the measures belonging to the first and second PCA components in addition to lift. On the contrary, if MOQAR uses the eDominance distance with the set of three objectives, it should maximize the third PCA component. Alternatively, if MOQAR optimizes the set of five objectives and uses the svDominance distance, it should optimize the support of the antecedent, which does not belong to a predetermined group.

Finally, the fourth column specifies the configuration settings in which the null hypothesis is accepted with respect to the control algorithm for each measure. That is, there are not significant differences between the specified configuration settings and the best one (*Control algorithm* column).

5.4.2 Statistical tests for the normalized average of all the quality measures

The average of the results retrieved for all the measures considered for each configuration setting of MOQAR has been calculated. Note that it is necessary to normalize (taking into account the minimum and maximum values obtained in the experimentation) all measures in order to compute the average. The normalized value of each measure for each configuration setting and dataset is computed considering their minimum and maximum values, which are calculated according to that obtained by all the configuration settings and all datasets studied in the experimentation.

Table 7 Average ranking obtained by Friedman of each quality measure for each configuration setting of MOQAR

Configuration	Rule Sup.	Ant. Sup.	Cons. Sup.	Lift	Confidence	Accuracy	Leverage	CF	Gain	Average ranking
indDominance-3	6.31	6.84	6.53	**1.66**	**1.12**	**1.31**	5.46	**1.22**	**1.38**	**3.61**
eDominance-3	**2.16**	3.22	**2.31**	6.28	2.59	5.19	**1.53**	2.56	3.97	**3.67**
crowDistance-3	2.78	3.69	2.84	5.34	4.12	5.00	2.12	3.78	4.56	4.00
indDominance-5	5.44	5.75	5.53	2.44	2.81	2.63	5.56	3.47	3.00	4.04
svDominance-3	3.63	4.28	3.34	5.28	5.03	5.56	3.43	4.84	5.16	4.58
eDominance-5	5.81	5.81	5.69	4.63	5.78	3.72	5.12	5.63	4.44	5.10
crowDistance-5	5.69	5.19	5.88	3.09	6.75	4.59	6.12	6.56	5.53	5.22
svDominance-5	4.19	**1.22**	3.88	7.28	7.78	8.00	6.62	7.94	7.97	5.78

Table 8 Post hoc comparison table ($\alpha = 0.05$) of each quality measure for each configuration setting of MOQAR

Measure	PCA component	Control algorithm	Holm, Finner H_0 accepted
Gain	1	indDominance-3	indDominance-5
Accuracy	1	indDominance-3	indDominance-5, eDominance-5
Lift	1	indDominance-3	indDominance-5
Confidence	2	indDominance-3	eDominance-3, indDominance-5
CF	2	indDominance-3	eDominance-3, indDominance-5
Consequent Sup.	1	eDominance-3	crowDistance-3, svDominance-3
Leverage	3	eDominance-3	crowDistance-3
Rule Sup.	3	eDominance-3	crowDistance-3
Antecedent Sup.	2, 3	svDominance-5	eDominance-3, crowDistance-3

Friedman and Iman-Davenport (ID) tests have been applied to determine which configuration setting presents the best performance for the mean of all measures since there is no single algorithm that has obtained the best performance for all the considered measures.

The results obtained by both tests for the level of significance $\alpha = 0.05$ have shown that there exist significant differences in the results obtained for each different configuration settings of MOQAR as the p-values obtained from both tests are lower than the level of significance considered. Therefore, a post hoc statistical analysis is required.

The average ranking obtained for the average of the normalized measures considering the level of significance $\alpha = 0.05$ is summarized in Table 9. The ranking obtained agrees with that of Table 7 since the top two is composed of eDominance and indDominance using three objectives.

The Holm and Finner tests have been applied to compare separately the control configuration setting, that is, eDominance using three objectives against the other configuration settings. Table 10 shows the sorted p-values obtained for each configuration setting for the level of significance $\alpha = 0.05$. These tests allow to conclude that eDominance with three objectives has better performance in six out of seven configurations for such level of significance. The post hoc statistical tests reject the hypotheses that have a p-value less or equal to 0.05 in the Holm and Finner tests. It can be noted that indDominance using three objectives is not significantly better than eDominance using three objectives, but is significantly better than the other six settings according to the p-values. In short, eDominance and indDominance distances present a similar performance for the average of all the measures when considering three objectives. However, each one of these two configurations could be applied depending on the desired type of rules: eDominance can be used to obtain general rules, that is, QAR with a high support; and indDominance to obtain specific rules with high confidence and accuracy. Finally, the svDominance presents the worst performance regardless of the number of objectives.

5.5 Comparison to other algorithms

This section details a comparative study between the best configuration settings of MOQAR, MODENAR [5], MOEA_Ghosh [15] and QAR-CIP-NSGAII [21]. As a result of the previous section, eDominance and indDominance have been the distances used in the secondary rank-

Table 9 Average ranking obtained by Friedman for the normalized average of quality measures for MOQAR

Configuration settings of MOQAR	Ranking
eDominance-3	1.69
indDominance-3	2.59
crowDistance-3	3.28
indDominance-5	3.56
svDominance-3	4.56
eDominance-5	5.68
crowDistance-5	6.66
svDominance-5	7.97

Table 10 Post hoc comparison table ($\alpha = 0.05$) for the normalized average of quality measures for each configuration setting of MOQAR (Control algorithm: eDominance-3)

i	Configuration	p	z	Holm	Finner	H_0 Reject
1	**indDominance-3**	0.1389	1.4799	0.0500	0.0500	
2	crowDistance-3	0.0093	2.6026	0.0250	0.0430	√
3	indDominance-5	0.0022	3.0619	0.0167	0.0360	√
4	svDominance-3	0.0000	4.6949	0.0125	0.0289	√
5	eDominance-5	0.0000	6.5320	0.0100	0.0217	√
6	crowDistance-5	0.0000	8.1139	0.0083	0.0145	√
7	svDominance-5	0.0000	10.2572	0.0071	0.0073	√

Table 11 Average measures of the rules obtained by MODENAR, MOEA_Ghosh, QAR-CIP-NSGAII and MOQAR algorithms

Measure	MODENAR	QAR-CIP-NSGAII	MOEA_Ghosh	MOQAR-e	MOQAR-i
Rule support (%)	34.59	14.04	**44.23**	21.44	9.09
Lift	33.84	208.58	73.55	29.33	**362.87**
Confidence (%)	87.51	95.42	76.07	93.61	**96.80**
Accuracy (%)	72.52	95.44	81.88	92.16	**98.35**
Leverage	0.035	0.068	0.045	**0.105**	0.035
CF	0.57	0.94	0.44	0.90	**0.95**
Gain	0.29	0.77	0.27	0.66	**0.86**
#Rules	43.69	159.44	**376.94**	93.37	73.37
Records (%)	67.87	96.61	66.04	**99.18**	82.98
Runtime (s)	18.96	**15.21**	37.30	661.74∗	618.70∗

*MOQAR-e and MOQAR-i perform 10,000 evaluations per rule (1,000,000 in total) whereas the rest of proposals are executed with 50,000 evaluations in total

ing and three objectives, confidence, leverage and accuracy, to be maximized by MOQAR have been considered.

The average results obtained by MODENAR, MOEA_Ghosh, QAR-CIP-NSGAII and MOQAR are summarized in Table 11. Note that MOQAR-e and MOQAR-i denote MOQAR

Table 12 Example of QAR obtained by MOQAR-e (ID 1 and ID 2) and MOQAR-i (ID 3 and ID 4) using AI dataset

ID	Rule	Acc. (%)	CF	Conf. (%)	Gain	Lift	Sup. (%)	Lev.
1	IF SeTime5 ∈ [0.021, 0.029] ⟹ SeTime4 ∈ [0.021,0.029] ∧ diffSeTime8 ∈ [0.24, 0.320]	100	1	100	0.71	3.5	28.8	0.21
2	IF SeTime9 ∈ [0.021, 0.029] ⟹ SeTime8 ∈ [0.031, 0.051]	100	1	100	0.90	10.4	9.5	0.09
3	IF SeTime10 ∈ [0.026, 0.036] ∧ SeTime14 ∈ [0.026, 0.036] ⟹ SeTime12 ∈ [0.026, 0.036] ∧ Se ∈ [0.026, 0.036]	100	1	100	0.87	7.6	13	0.11
4	IF SeTime8 ∈ [0.036, 0.054] ⟹ SeTime9 ∈ [0.036, 0.054] ∧ diffSeTime4 ∈ [0,0] ∧ SeTime3 ∈ [0.032, 0.054]	100	1	100	0.95	18.8	5.3	0.06

when using the eDominance and indDominance distances for the secondary ranking, respectively. The best value for each quality measure and each algorithm is emboldened.

The percentage of covered records and average rule support for the QAR obtained by MODENAR, MOEA_Ghosh, QAR-CIP-NSGAII, MOQAR-e and MOQAR-i point out how general the rules are. It can be noted that the mean percentage of records covered by the rules obtained by MOQAR-e reaches values close to 100 % for all the datasets, whereas the percentage of records covered by MODENAR and MOEA_Ghosh is below 70 %. Specifically, MOQAR-e obtains the 100 % in 29 datasets. QAR-CIP-NSGAII and MOQAR-i obtain the second best values. Regarding the average support, MOEA_Ghosh obtains highly general QAR since average values close to 50 % are obtained, in contrast to the specific rules found by MOQAR-i.

Alternatively, the confidence, certainty factor, leverage, lift and gain measures are devoted to evaluate the reliability of the rules. The QAR found by MOQAR-e, MOQAR-i and QAR-CIP-NSGAII present a confidence value above 95 %. MOEA_Ghosh obtains the worst values regarding confidence measure.

MOQAR-i outperforms all the algorithms considered in the comparative study in terms of lift, accuracy, certainty factor and gain. Additionally, MOQAR-e is the best algorithm with regard to leverage measure. It is noteworthy that MODENAR and MOEA_Ghosh obtain the worst values for gain, certainty factor and leverage because the support of the consequent is close to 50 %. MOEA_Ghosh achieves the higher number of rules although this algorithm is not able to cover the entire dataset. Note that MOQAR-e and MOQAR-i cover most of instances of the datasets with less than 100 rules which lead to conclude that rules are more diverse and less overlapping.

The average runtime in seconds is also summarized in Table 11 for all the approaches. It can be observed that QAR-CIP-NSGAII algorithm achieves the best runtime. Although MOQAR-e and MOQAR-i spend more computational time, it should be considered that the number of evaluations spent by MOQAR-e and MOQAR-i is 20 times higher than the rest of algorithms. This fact is due to the iterative scheme implemented in both proposals since a complete evolutionary process is executed for each rule to obtain. Therefore, if the number of evaluations invested in each iteration is 10,000 and the fixed number of rules to obtain for each iteration is 100, the presented runtime refers to 1 million of evaluations. Hence,

Table 13 Average rankings obtained by Friedman for each quality measure for all algorithms

Algorithms	Rule Sup.	Lift	Conf.	Acc.	Lev.	CF	Gain	Records
MOQAR-i	4.38	**1.41**	**1.71**	**1.42**	3.63	**1.56**	**1.41**	3.16
MOQAR-e	2.66	3.53	3.43	3.34	**1.50**	2.91	2.97	**1.84**
QAR-CIP-NSGAII	3.41	2.06	2.59	2.36	2.31	2.06	2.03	2.53
MOEA_Ghosh	2.31	3.73	3.65	3.41	3.69	4.42	4.28	3.31
MODENAR	**2.25**	4.27	3.59	4.47	3.88	4.05	4.31	4.16

50000 evaluations would be executed in 30 s (on average approximately) by MOQAR-e and MOQAR-i, which would be comparable to other algorithms such as MOEA_Ghosh (Table 11).

It can be concluded that although MOEA_Ghosh presents better results in terms of support of the rules, MOQAR discovers more precise and interesting rules with a strong dependency between the antecedent and consequent. Furthermore, MOQAR reaches higher information gain on the rules regarding the consequent, when the antecedent is also present. In spite of the support of the rules obtained by MOQAR is slightly lower, this fact should not be considered particularly relevant since this measure only determines how general QAR are. Additionally, support of the rules is not able to find negative dependencies between the antecedent and the consequent and high values for that measure do not guarantee quality rules.

In the light of the results obtained, it can be stated that MOQAR presents the best trade-off among all the measures under study. Furthermore, the optimization of the selected measures by the methodology based on PCA leads to the optimization of the whole set of quality measures specified in Sect. 5.3. Representative QAR obtained by both MOQAR-e and MOQAR-i are shown in Table 12. As can be observed, all the measures reach values close to their maximum values, whereas other algorithms usually obtain worse results in the measures not maximized in their fitness function. Alternatively, the use of alternative distances to crowding distance involves better results in terms of quality measures, not only for optimization problems with many objectives but also for problems with three objectives.

Finally, a nonparametric statistical analysis has been carried out to determine whether significant differences are presented between the results obtained by MOQAR and the other approaches. For this purpose, support of the rule, percentage of covered instances, lift, confidence, accuracy, leverage, certainty factor and gain measures obtained from the results obtained by MODENAR, MOEA_Ghosh, QAR-CIP-NSGAII and MOQAR for the thirty-two datasets from BUFA have been calculated. The number of rules have not been included in the statistical analysis since a larger number of them does not necessarily involve that an algorithm is better.

Friedman and Iman-Davenport tests have been applied for a level of significance $\alpha = 0.05$. Note that there exist significant differences for each quality measure for the algorithms under study, since the p-values are lower than the level of significance considered. Therefore, a post hoc statistical analysis has been carried out.

Table 13 shows the average rankings obtained by the Friedman test for the studied algorithms. The best algorithm for each measure is stressed in bold. It can be appreciated that MOQAR is the best algorithm in six out of seven measures, and in particular, MOQAR-i is the best configuration setting in five measures.

Regarding the support of the rules, MODENAR is the best algorithm, although MOEA_Ghosh achieves better average values. It can be noted that such algorithm is bet-

Table 14 Post hoc comparison table ($\alpha = 0.05$) for each quality measure for all algorithms

Measures	i	Algorithms	z	p	Holm	Finner	H_0 Rejected
		(Control alg.: MODENAR)					
Rule support	1	MOEA_Ghosh	0.1581	0.8744	0.0500	0.0500	
	2	MOQAR-e	1.0277	0.3041	0.0250	0.0377	
	3	QAR-CIP-NSGAII	2.9251	0.0034	0.0167	0.0253	√
	4	MOQAR-i	5.3759	0.0000	0.0125	0.0127	√
		(Control alg.: MOQAR-i)					
Lift	1	QAR-CIP-NSGAII	1.6602	0.0969	0.0500	0.0500	
	2	MOQAR-e	5.3759	0.0000	0.0250	0.0377	√
	3	MOEA_Ghosh	5.8897	0.0000	0.0167	0.0253	√
	4	MODENAR	7.2337	0.0000	0.0125	0.0127	√
		(Control alg.: MOQAR-i)					
Confidence	1	QAR-CIP-NSGAII	2.2136	0.0269	0.0500	0.0500	√
	2	MOQAR-e	4.3481	0.0000	0.0250	0.0377	√
	3	MODENAR	4.7434	0.0000	0.0167	0.0253	√
	4	MOEA_Ghosh	4.9015	0.0000	0.0125	0.0127	√
		(Control alg.: MOQAR-i)					
Accuracy	1	QAR-CIP-NSGAII	2.3717	0.0177	0.0500	0.0500	√
	2	MOQAR-e	4.8620	0.0000	0.0250	0.0377	√
	3	MOEA_Ghosh	5.0201	0.0000	0.0167	0.0253	√
	4	MODENAR	7.7081	0.0000	0.0125	0.0127	√
		(Control alg.: MOQAR-e)					
Leverage	1	QAR-CIP-NSGAII	2.0555	0.0398	0.0500	0.0500	√
	2	MOQAR-i	5.3759	0.0000	0.0250	0.0377	√
	3	MOEA_Ghosh	5.5340	0.0000	0.0167	0.0253	√
	4	MODENAR	6.0083	0.0000	0.0125	0.0127	√
		(Control alg.: MOQAR-i)					
CF	1	QAR-CIP-NSGAII	1.2649	0.2059	0.0500	0.0500	
	2	MOQAR-e	3.3994	0.0007	0.0250	0.0377	√
	3	MODENAR	6.2850	0.0000	0.0167	0.0253	√
	4	MOEA_Ghosh	7.2337	0.0000	0.0125	0.0127	√
		(Control alg.: MOQAR-i)					
Gain	1	QAR-CIP-NSGAII	1.5811	0.1138	0.0500	0.0500	
	2	MOQAR-e	3.9528	0.0001	0.0250	0.0377	√
	3	MOEA_Ghosh	7.2732	0.0000	0.0167	0.0253	√
	4	MODENAR	7.3523	0.0000	0.0125	0.0127	√
		(Control alg.: MOQAR-e)					
Covered Records	1	QAR-CIP-NSGAII	1.7393	0.082	0.05	0.05	
	2	MOQAR-i	3.3204	0.0009	0.025	0.0377	√
	3	MOEA_Ghosh	3.7157	0.0002	0.0167	0.0253	√
	4	MODENAR	5.8502	0	0.0125	0.0127	√

ter in terms of support, but the rules obtained cover around the 67 % of instances of datasets (see Table 11). On the contrary, both MODENAR and MOEA_Ghosh are the worst algorithms in the ranking for the remaining measures.

In short, MOQAR-i is the best algorithm for confidence, certainty factor, lift, accuracy and gain measures, MOQAR-e for the leverage measure and percentage of covered records and MODENAR for support of the rule.

Finally, Table 14 summarizes the results obtained by the Holm and Finner post hoc statistical tests for each quality measure. It can be observed that MOQAR-i is the control algorithm for the most of the measures, excluding the support, percentage of covered records and leverage measures in which MODENAR and MOQAR-e are the control algorithms, respectively. As for the support, MOQAR-e does not present significant differences with MODENAR. According to lift, CF and gain, QAR-CIP-NSGAII does not present significant differences with MOQAR-i.

6 Conclusions

An algorithm, MOQAR, based on the NSGA-II scheme, has been improved to discover high-quality QAR. First, MOQAR has found the best trade-off among all the measures by maximizing the objectives selected by a principal component analysis. Namely, accuracy, confidence and leverage have been the set of objectives that better summarize all the considered measures. Second, several distances have been proposed to replace the well-known crowding distance used to obtain the secondary ranking in the Pareto-fronts achieved by the evolutionary process of MOQAR. In particular, the iDominance and eDominance distances have outperformed the quality of the QAR obtained by MOQAR when comparing to those found by using the crowding distance. Finally, thirty-two datasets from BUFA repository have been considered to compare the best configuration settings of MOQAR to other existing evolutionary algorithms such as MODENAR, MOEA_Ghosh and QAR-CIP-NSGAII. From the statistical tests, it can be concluded that MOQAR outperformed the remaining algorithms for all measures except for the rule support.

As future work, the authors intend to consider other modern approaches based on decomposition or differential evolution as a base learning to discover efficiently high-quality QAR.

Acknowledgments The authors would like to thank Spanish Ministry of Science and Technology, Junta de Andalucia and University Pablo de Olavide for the support under Projects TIN2011-28956-C02, TIN2014-55894-C2-R, P12-TIC-1728 and APPB813097, respectively.

References

1. Agrawal R, Imielinski T, Swami A (1993) Mining association rules between sets of items in large databases. In: Proceedings of the ACM SIGMOD international conference on management of data, pp 207–216
2. Agrawal R, Srikant R (1994) Fast algorithms for mining association rules in large databases. In: Proceedings of the international conference on very large databases, pp 478–499
3. Aguirre H, Tanaka K (2009) Space partitioning with adaptive ranking and substitute distance assignments: a comparative study on many-objective mnk-landscapes. In: Proceedings of the annual conference on genetic and evolutionary computation, pp 547–554
4. Alatas B, Akin E (2006) An efficient genetic algorithm for automated mining of both positive and negative quantitative association rules. Soft Comput 10(3):230–237
5. Alatas B, Akin E, Karci A (2008) MODENAR: multi-objective differential evolution algorithm for mining numeric association rules. Appl Soft Comput 8(1):646–656

6. Alcalá-Fdez J, Sánchez L, García S, del Jesús MJ, Ventura S, Garrell JM, Otero J, Romero C, Bacardit J, Rivas VM, Fernández JC, Herrera F (2009) Keel: a software tool to assess evolutionary algorithms for data mining problems. Soft Comput 13(3):307–318
7. Anand R, Vaid A, Singh PK (2009) Association rule mining using multi-objective evolutionary algorithms: strengths and challenges. In: Proceedings of the IEEE world congress on nature biologically inspired computing, pp 385–390
8. Brin S, Motwani R, Silverstein C (1997) Beyond market baskets, generalizing association rules to correlations. In: Proceedings of the ACM SIGMOD, pp 265–276
9. Brin S, Motwani R, Ullman JD, Tsur S (1997) Dynamic itemset counting and implication rules for market basket data. In: Proceedings of the ACM SIGMOD, pp 265–276
10. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evolut Comput 6(2):182–197
11. Dehuri S, Jagadev AK, Ghosh A, Mall R (2006) Multi-objective genetic algorithm for association rule mining using a homogeneous dedicated cluster of workstations. Am J Appl Sci 3(11):2086–2095
12. del Jesús MJ, Gámez JA, González P, Puerta JM (2011) On the discovery of association rules by means of evolutionary algorithms. Wiley Interdiscip Rev Data Min Knowl Discov 1(5):397–415
13. García S, Fernández A, Luengo J, Herrera F (2009) A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability. Soft Comput 13(10):959–977
14. Geng L, Hamilton HJ (2006) Interestingness measures for data mining: a survey. ACM Comput Surv 38(3):1–42
15. Ghosh A, Nath B (2004) Multi-objective rule mining using genetic algorithms. Inf Sci 163:123–133
16. Guvenir HA, Uysal I (2000) Bilkent university function approximation repository. http://funapp.cs.bilkent.edu.tr
17. Köppen M, Yoshida K (2007) Substitute distance assignments in NSGA-II for handling many-objective optimization problems. In: Evolutionary multi-criterion optimization, volume 4403 of Lecture Notes in Computer Science. Springer, Berlin, pp 727–741
18. Li D, Deogun J, Spaulding W, Shuart B (2004) Towards missing data imputation: a study of fuzzy k-means clustering method. In: Rough sets and current trends in computing, volume 3066 of Lecture Notes on Computer Science, pp 573–579
19. Luna JM, Romero JR, Ventura S (2012) Design and behavior study of a grammar-guided genetic programming algorithm for mining association rules. Knowl Inf Syst 32(1):53–76
20. Luna JM, Romero JR, Ventura S (2013) Grammar-based multi-objective algorithms for mining association rules. Data Knowl Eng 86:19–37
21. Martín D, Rosete A, Alcalá-Fdez J, Herrera F (2014) QAR-CIP-NSGA-II: a new multi-objective evolutionary algorithm to mine quantitative association rules. Inf Sci 258:1–28
22. Martínez-Ballesteros M, Martínez-Álvarez F, Troncoso A, Riquelme JC (2009) Quantitative association rules applied to climatological time series forecasting. In: Proceedings of the international conference on intelligent data engineering and automated learning, volume 5788 of Lecture Notes in Computer Science, pp 284–291
23. Martínez-Ballesteros M, Martínez-Álvarez F, Troncoso A, Riquelme JC (2011) An evolutionary algorithm to discover quantitative association rules in multidimensional time series. Soft Comput 15(10):2065–2084
24. Martínez-Ballesteros M, Martínez-Álvarez F, Troncoso A, Riquelme JC (2014) Selecting the best measures to discover quantitative association rules. Neurocomputing 126:3–14
25. Martínez-Ballesteros M, Salcedo-Sanz S, Riquelme JC, Casanova-Mateo C, Camacho JL (2011) Evolutionary association rules for total ozone content modeling from satellite observations. Chemom Intell Lab Syst 109(2):217–227
26. Mata J, Álvarez J, Riquelme JC (2001) Mining numeric association rules with genetic algorithms. In: Proceedings of the international conference on adaptive and natural computing algorithms, pp 264–267
27. Miller BL, Goldberg DE (1995) Genetic algorithms, tournament selection, and the effects of noise. Complex Syst 9(3):193–212
28. Pachón Álvarez V, Vázquez JM (2012) An evolutionary algorithm to discover quantitative association rules from huge databases without the need for an a priori discretization. Expert Syst Appl 39(1):585–593
29. Pears R, Koh YS, Dobbie G, Yeap W (2013) Weighted association rule mining via a graph based connectivity model. Inf Sci 218:61–84
30. Piatetsky-Shapiro G (1991) Discovery, analysis and presentation of strong rules. In: Proceedings of knowledge discovery in databases. AAAI Press, pp 229–248
31. 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(1):288–298

32. Shortliffe E, Buchanan B (1975) A model of inexact reasoning in medicine. Math Biosci 23:351–379
33. Venturini G (1993) SIA: a supervised inductive algorithm with genetic search for learning attribute based concepts. In: Proceedings of the European conference on machine learning, pp 280–296
34. Wakabi-Waiswa PP, Baryamureeba V (2008) Extraction of interesting association rules using genetic algorithms. Int J Comput ICT Res 2(1):26–33
35. Yan X, Zhang C, Zhang S (2009) Genetic algorithm-based strategy for identifying association rules without specifying actual minimum support. Expert Syst Appl 36(2):3066–3076
36. Zhou A, Qu B-Y, Li H, Zhao S-Z, Suganthan PN, Zhang Q (2011) Multiobjective evolutionary algorithms: a survey of the state of the art. Swarm Evol Comput 1(1):32–49
37. Zitzler E, Laumanns M, Thiele L (2001) SPEA2: improving the strength pareto evolutionary algorithm. EUROGEN 3242(103):95–100
38. Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Trans Evolut Comput 3(4):257–271