NICGAR: a Niching Genetic Algorithm to Mine a Diverse Set of Interesting Quantitative Association Rules

D. Martín[a], J. Alcalá-Fdez[b,*], A. Rosete[a], F. Herrera[b]

[a] *Dept. Artificial Intelligence and Infrastructure of Informatic Systems, Higher Polytechnic Institute José Antonio Echeverría, Cujae, 19390 La Habana, Cuba*

[b] *Department of Computer Science and Artificial Intelligence, University of Granada, CITIC-UGR, 18071 Granada, Spain*

Abstract

Evolutionary algorithms are normally applied to mine association rules on quantitative data but most of them obtain enough similar rules due to that the usual behavior of these algorithms is to converge on the best solution of the problem. To overthrow this issue, in this paper we present NICGAR, a new Niching Genetic Algorithm to obtain a reduce set of different positive and negative quantitative association rules with a low runtime. To do that, we extract the rules based on the existence of a pool of external solutions that contains the best rule of each niche found in the search process according to several quality measures, we penalize similar rules by means of a process based on fitness sharing, and we restart the algorithm leading to a diverse population. Moreover, the user can tune the trade-off between the quality and diversity of the mined rules making use of two thresholds. Finally, a new measure have also been presented to assess the similarity between rules based on involved attributes and covered examples by the rules. The quality of our proposal is analyzed using statistical analysis and comparing with classical, mono-objective evolutionary, and multi-objective evolutionary approaches for mining association rules.

Keywords: Data Mining, Positive and Negative Quantitative Association Rules, Niching Genetic Algorithms

1. Introduction

In recent times, the amount of data collected from different application areas has continuously increased. This growth has led to a situation in which the extraction of interesting knowledge from large datasets is a very attractive and challenging task. Data Mining (DM) techniques [19] are used to build efficient predictive or descriptive models from a large amount of data that not only best fits or explains it, but is also able to generalize to new data. Discovering association rules is one of the DM techniques described in the literature, the aim of which is to detect and to represent interesting relationships between the different items of a dataset [49]. These are represented as $X \rightarrow Y$, where X and Y are item sets and $X \cap Y = \oslash$. Many studies have been used to extract association rules from datasets with quantitative data [12, 42], but most of them only mine positive quantitative association rules (QARs) without paying particular attention to negative QARs. Recently, different studies have been proposed to mine positive and negative QARs (PNQARs) from datasets according to negative rules, such as $X \rightarrow \neg Y$, which may also be interesting as they relate the presence of certain items to the absence of others [2, 31, 44].

Over the last few years, many researchers have proposed Evolutionary Algorithms (EAs) [13] to mine QARs from datasets [2, 4, 21, 29, 31, 32, 33, 46]. EAs, and particularly Genetic Algorithms (GAs) [17], are one of the most successful search and optimization techniques for complex problems. These algorithms optimize a population of solutions in order to obtain the best quality solutions for the problem. However, the natural tendency of GAs is always to converge on the best solution and as a consequence they usually present a poor diversity in the final set of

[*] Corresponding author. Tel +34-958-240429

Email addresses: `dmartin@ceis.cujae.edu.cu` (D. Martín), `jalcala@decsai.ugr.es` (J. Alcalá-Fdez), `rosete@ceis.cujae.edu.cu` (A. Rosete), `herrera@decsai.ugr.es` (F. Herrera)

solutions. Thus, finding and maintaining multiple solutions in the population is a challenge for the use of GAs with multi-modal problems, i.e., problems with several global optimals.

The extraction of association rules is a highly multi-modal problem in which all the best quality rules (the optimal solutions of the problem) should be recovered due to the fact that they provide the most interesting and diverse knowledge of the dataset. Niching Genetic Algorithms (NGAs) [10, 34] have proved to be an interesting method with which to approach highly multi-modal problems, as these algorithms allow us to locate and to maintain multiple global optimals, avoiding the convergence on only one solution. For this reason, the use of NGAs to mine association rules could be an interesting way of working, in which the seed of each niche obtained represents an interesting association rule that provides different knowledge to that of the rest of the rules.

In this paper, we propose a new NGA for mining with a low runtime a diverse and reduced set of PNQARs that are easy to understand, interesting and which provide a good coverage of the dataset, called NICGAR. To do that, we extract the rules based on the existence of an external population (EP) that contains the best rule of each niche found in the search process according to several quality measures, we penalize similar rules by means of a process based on fitness sharing [18], and we restart the algorithm leading to a diverse population. Moreover, the user can tune the trade-off between the quality and diversity of the mined rules making use of two thresholds. Finally, a new measure have also been presented to assess the similarity between rules based on involved attributes and covered examples by the rules.

In order to evaluate the performance of NICGAR, an experimental study has been performed using 27 real world datasets, in which the number of examples is within the interval [40, 100,968] and the number of attributes is within the interval [4, 91]. We have firstly studied the influence of two parameters on NICGAR that allow the user to adjust the trade-off between quality, diversity of rules and coverage of the datasets. Second, we have compared the performance of our proposal with two NGAs (Clearing [36, 35] and ASCGA [25]), which were extended to extract PNQARs. Third, the performance of NICGAR has been compared with four mono-objective evolutionary approaches (EARMGA [46], GAR [21], GENAR [33] and the approach proposed by Alatas et al. in [2], which in this paper will be called Alatasetal) and two Multi-Objective Evolutionary Algorithms (MOEAs) (QAR-CIP-NSGA-II [32] and MOPNAR [31]), in which Alatasetal and MOPNAR mine PNQARs and the rest of the algorithms mine positive QARs. Fourth, we have compared two other classical algorithms (Eclat [48] and Apriori [42]) that mine positive association rules with our approach. In all these studies, we have made use of some nonparametric statistical tests for the pairwise and multiple comparison [15] of the performance of these methods over the 27 real-world datasets. Fifth, we study the diversity of the set of rules obtained by some evolutionary approaches. Finally, we have analyzed the complexity and scalability of our proposal. Moreover, a web page associated with this paper (i.e., http://sci2s.ugr.es/NICGAR/) contains material complementary to this study.

This paper is arranged as follows. Section 2 provides a brief study of the existing NGAs, some basic descriptions of PNQARs and some quality measures. Section 3 presents our proposal to obtain a diverse set of interesting PNQARs. Section 4 analyzes and discusses the obtained results over 27 real-world datasets. Finally, Section 5 presents some concluding remarks.

2. PRELIMINARIES

In this section, we first show a brief study of NGAs. Then we present the basic descriptions of PNQARs and some quality measures.

2.1. Niching Genetic Algorithms

EAs [13] simultaneously deal with a set of possible solutions which enables them to find several optimal solutions in a single run of the algorithm. However, finding and maintaining of multiple solutions in the population is a challenge for the use of GAs with multi-modal problems, i.e., problems with several global optimals, because of the above mentioned natural tendency of GAs to always converge on the best solution. In order to deal with this, the NGAs extend the GA to locate and maintain multiple optimal solutions in the population for this type of problem [10, 34].

One of the first studies of the preservation of diversity was Cavicchio's dissertation in 1970 [8]. In this study, different preselection schemes for replacing an offspring with one of its parents were proposed. Later, De Jong generalized the preselection schemes with the *crowding* method [11]. The use of these methods in multimodal functions may not be successful because they can preserve several representatives of the same optimum due to substitution errors [30]. To overcome this problem many versions of the crowding method were developed, with one of the most representative being the Deterministic Crowding method [30].

In 1987, Goldberg and Richardson proposed the *sharing* method in which the fitness of individuals decreases in accordance with the number of similar individuals in the population [18]. This method enables genetic algorithms to simultaneously work on multiple relative optima in multimodal optimization problems. However, later studies have shown it to contain certain limitations owing to the sharing parameter and the high computational complexity [38]. In the following years, many algorithms were designed in order to resolve both drawbacks. Most of them focus on a preliminary distribution of individuals in niches by the clustering algorithm or calculate shared fitness from fixed samples of the population [9, 26, 38, 47].

Later, Petrowski presented the *clearing* procedure [36, 35] based on the same concept as the sharing method. However, unlike the sharing method, only the best individuals of each niche survive, the fitness of the rest of them is reset to zero. This process is applied after the evaluation process and before the selection. Following this idea, several methods have been presented which also modify the quality of the solutions according to distance [22, 23].

Recently, Li has presented a different niching approach, called *species* conservation [24]. This proposal divides the population into several species according to their similarity and each of these species is built around a dominating individual, commonly referred to as a species seed. This technique has been proved to be effective to obtain multiple solutions of multimodal problems.

A study of the different NGAs that have been proposed in the literature can be found in [10, 34].

2.2. Positive and Negative Quantitative Association Rules

QARs are association rules extracted from datasets with quantitative values whereby each item is usually represented by a pair attribute-interval [42]. These rules may include positive or negative items in their representation. For instance, the rule $Age \in [10, 200] \rightarrow Weight \in [25, 200]$ involves only positive items, indicating that when the Age is equal to or higher than 10 then the Weight is between 25 and 200; the rule $Age \in [0, 5] \rightarrow Weight \in \neg[25, 200]$ involves positive and negative items indicating that when the Age is between 0 and 5 then the Weight is lower than 25. The use of negative items in the extraction of QARs is very useful since a single negative item can represent knowledge that would require several positive items to represent it.

To be able to mine PNQARs from datasets with quantitative values, many algorithms need to make an a priori partition of the attribute's domains, but the given intervals may determine the final set of rules. Because of this, a number of studies have been proposed to also include a learning or tuning of the partitions of the attribute's domains when we mine QARs [2, 21, 33, 46].

The most common measures to evaluate association rules are support and confidence. These measures are based on the support of an itemset I, which will be called $SUP(I)$. The $SUP(I)$ is calculated by dividing the number of covered examples between the whole number of examples. These measures can be defined for a rule $A \rightarrow B$ as follows:

$$support(A \rightarrow B) = SUP(AB) \tag{1}$$

$$confidence(A \rightarrow B) = \frac{SUP(AB)}{SUP(A)} \tag{2}$$

Many proposals have used these measures to extract QARs, however it has been pointed out that they present several problems [6, 7]. On the one hand, if the consequent support of the rule is very high any antecedent can seem to be a good predictor. On the other hand, negative dependence is not detected by the confidence measure due to the fact that it does not consider the consequent support. For this reason, several authors have proposed other measures to overcome these problems and to select and rank examples on the basis of their potential interest to the user [16]. Table 1 shows a brief description of some of these measures, which allows us to find out independence, positive dependence and negative dependence. Notice that the range of the conviction and lift measures is not bounded above.

Table 1: Summary of some interestingness measures

Measures	*Equation*	*Description*	*Domain*
$conviction(A \to B)$[7]	$\frac{SUP(A)SUP(\neg B)}{SUP(A\neg B)}$	Dependence between A and $\neg B$ Value < 1: negative dependence Value = 1: independence Value > 1: positive dependence	[0,∞)
$lift(A \to B)$ [37]	$\frac{SUP(AB)}{SUP(A)SUP(B)}$	Ratio between the confidence and the expected confidence of the rule Value < 1: negative dependence Value = 1: independence Value > 1: positive dependence	[0,∞)
$netconf(A \to B)$[1]	$\frac{SUP(AB) - SUP(A)SUP(B)}{SUP(A)(1 - SUP(A))}$	Estimate the strength of the rules. Value < 0: negative dependence Value = 0: independence Value > 0: positive dependence	[-1,1]
$yule'sQ(A \to B)$[43]	$\frac{SUP(AB)SUP(\neg A\neg B) - SUP(A\neg B)SUP(\neg AB)}{SUP(AB)SUP(\neg A\neg B) + SUP(A\neg B)SUP(\neg AB)}$	Correlation between two possibly related dichotomous events Value < 0: negative dependence Value = 0: independence Value > 0: positive dependence	[-1,1]
Certain Factor CF (A → B)[41]	*if confidence(A → B) > SUP(B):* $\frac{confidence(A \to B) - SUP(B)}{1 - SUP(B)}$ *if confidence(A → B) < SUP(B):* $\frac{confidence(A \to B) - SUP(B)}{SUP(B)}$ *Otherwise is* 0	Variation of the probability that B is in a example considering only those where A is present. Value < 0: negative dependence Value > 0: positive dependence Value = 0: independence	[-1,1]

3. A new NGA for mining positive and negative quantitative association rules: NICGAR

In what follows, we will present our proposal to obtain a diverse set of PNQARs while ensuring a good balance between the quality and variety of the knowledge obtained. This proposal performs an evolutionary learning of the rules, combining an EP, a punishment mechanism and a restarting process to preserve multiple optimal solutions in the population and to introduce diversity into the search process. All their characteristics are presented in detail in the following subsections.

3.1. Niche management within NICGAR

We consider the use of an NGA to learn the association rules, which extend the GAs to locate and maintain multiple optimal solutions in the population and to avoid the convergence on only one solution. In order to manage the niches and to ensure diversity in the population, we have introduced an EP, a punishment mechanism and a restarting process in the search process. The following sections provide detailed descriptions of the three components to manage the niches and the process of identifying whether two solutions belong to the same niche in NICGAR. Notice that these components are complementary and cannot be used separately without affecting the evolutionary process.

3.1.1. External Population

The EP allows us to avoid the loss of global solutions, keeping the best solution of each niche found in the search process independently of the number of niches in the problem. Notice that our proposal follows a dataset independent approach and the size of the EP is not bounded, which results in a greater number of rules regardless of the population's size, and allows us to reduce the size of the population and to use a fixed size. The EP is initially empty and it will be updated when the current population percentage of new solutions is less than $\alpha\%$ of solutions of the population, i.e, when the evolution has achieved a stable quality in the population in order to avoid the addition of low quality rules.

To update the EP, first it is necessary to identify the niches from the current population. To do this, the algorithm proposed by Li [24] is applied to determine the seed of each niche. For this, the current population is arranged in decreasing order of fitness (see subsection 3.4), with the first individual the seed of the first identified niche. From here onwards the process becomes iterative, the next individual in the arrangement is taken and if it does not belong to the same niche as any previously identified seeds this individual will be considered as the seed of a new niche (see subsection 3.1.4 for more information on when two solutions belong to the same niche). Then, the rest of the individuals of the population are classified into the different identified niches. The EP will be updated with the best solution of each niche found in the population that has a fitness higher than a threshold value, called $Ev_{Min}\%$, which represents a percentage of the average fitness of the individuals from the EP (in our case, it has been fixed to 85%, i.e., 0.85). Notice that the Ev_{Min} threshold allows the expert to remove low quality solutions, which correspond to local optimas of the problem. To avoid the redundant rules in the EP, we check if there are rules in the EP that belong to the same niche, in which case we keep only the best rule of the niche.

3.1.2. Punishment mechanism

The punishment mechanism based on the fitness sharing method [18] is applied to promote the search for different optimal solutions from those that are in the EP. This mechanism will penalize the individuals that belong to the same niche of any of the solutions of the EP and that have a fitness value lower than the corresponding rule of the EP. Thereby, we encourage the development of niches in other areas of the search space in which optimal solutions have not been found. The fitness value of a solution that is penalized is modified as follows:

$$fitness'(C) = fitness(C) - ((1 - dist_{OC}(C, Sol_{EP})) * (fitness(C) * \chi\%)) \qquad (3)$$

where $fitness(C)$ is the fitness value of the solution (see subsection 3.4), Sol_{EP} is the corresponding solution of the EP and $dist_{OC}$ is the distance value between the solution and the corresponding solution of the EP (see equation 7 in subsection 3.1.4). This function penalizes the fitness value of a solution with a maximum value of $\chi\%$ of its fitness value. In our case, χ has been fixed to 20%, i.e, 0.2.

3.1.3. Restarting Process

The restarting process is applied to move away from local optima and to restart the search in other areas of the search space, allowing us to find niches in other areas in which we have not found beforehand. The population will be restarted when the current population percentage of new solutions is under $\alpha\%$ of solutions of the population. In this case, the EP is updated and the process of initialization is again applied from uncovered examples by the rules of the EP (see subsection 3.2). Finally, the punishment mechanism is applied to the new individuals of the population.

3.1.4. Identifying niches and a new similarity measure

In order to identify whether two solutions belong to the same niche, two properties of the rules are analyzed: the overlap ratio of the common attributes and the ratio of examples covered by the two rules. The overlap ratio represents the average overlap of the common attributes of the rules. If this ratio is higher than the threshold value $Nich_{Min}$, then these rules would appear to provide us with similar information of the search space. However, this information could be provided from different examples depending on the rest of the attributes of the rules. Because of this, the covered examples ratio is also calculated, which represents the maximum percentage of covered examples by the two rules regarding the examples covered for each rule. If this ratio is also higher than the threshold $Nich_{Min}$, the rules will be grouped in the same niche. Notice that the $Nich_{Min}$ threshold allows the expert to determine the degree of diversity of the generated rules.

The overlap of two intervals of the same attribute, A_{I1} and A_{I2}, is defined as the highest value obtained from dividing the length of the common part of the two intervals between the length of each interval. If any interval involved is negative then the overlap is calculated considering its positive equivalent interval. Thus, the overlap of two attributes A_{I1} and A_{I2} is defined as:

$$overlap\,(A_{I1}, A_{I2}) = Max\left(\frac{Lenght_{overlap}}{Lenght_{A_{I1}}}, \frac{Lenght_{overlap}}{Lenght_{A_{I2}}}\right) \qquad (4)$$

where values close to 0 reveal that the intervals are very different and values close to 1 that the intervals are very similar. Notice that the overlap for categorical attributes will be 1 if the value of the attributes is the same, and 0 if otherwise. Thus, the overlap ratio of the rules R_1 and R_2 is defined as:

$$ratio_{overlap}(R_1, R_2) = \frac{\sum overlap(A_{i_{R_1}}, A_{i_{R_2}})}{\mid CA \mid} \qquad (5)$$

Table 2: Six examples in this example

ID	X_1	X_2	X_3
ID1	20	1.5	7
ID2	19.5	1	6
ID3	20	2.1	6
ID4	22	4.3	7
ID5	30	8.4	13
ID6	20	2.4	2

where CA are the common attributes of the two rules and $\mid CA \mid$ is the number of common attributes of the two rules. Notice that the overlap ratio will be 0 if all the attributes of the rules are different since they provide different information about the problem.

The ratio of examples covered takes values in the range [0,1], where values close to 0 show that the rules cover few common examples and values close to 1 that the rules cover almost the same examples. This is defined as:

$$ratio_{cover}(R_1, R_2) = Max\left(\frac{cov(R_1R_2)}{cov(R_1)}, \frac{cov(R_1R_2)}{cov(R_2)}\right) \tag{6}$$

where $cov(R_1R_2)$ represents the number of common examples covered by both rules R_1 and R_2, and $cov(R_1)$ and $cov(R_2)$ represent the number of examples covered by R_1 and R_2, respectively. Notice that, for rules that are specializations of other rules, this measure will obtain its maximum value.

For instance, let us consider a simple dataset with three attributes X_1, X_2 and X_3, 6 training examples and $Nich_{Min} = 0.5$. Table 2 shows the values of the 6 examples of the dataset. Let us suppose that we have the rules: R_1: $X_1 \in [20, 20]$ and $X_2 \in [1.5, 3.5] \to X_3 \in [1, 8]$; and R_2: $X_1 \in [18, 25] \to X_3 \in [5, 14]$. To determine whether these two rules belong to the same niche, we first calculate the overlap ratio between the rules as follows:

$$overlap\left(X_{1_{R_1}}, X_{1_{R_2}}\right) = 1$$

$$overlap\left(X_{3_{R_1}}, X_{3_{R_2}}\right) = Max\left(\frac{3}{7}, \frac{3}{9}\right) = 0.42$$

$$ratio_{overlap}(R_1, R_2) = \frac{1 + 0.42}{2} = 0.71$$

As the ratio of overlap of the common attributes is higher than $Nich_{Min}$, we also calculate the ratio of examples covered by the two rules. Table 2 shows how R_1 covers the examples ID1, ID3 and ID6, and R_2 covers ID1, ID2, ID3 and ID4, with the common examples being: ID1 and ID3. Take into account that in this situation, the ratio of examples covered is calculated as follows:

$$ratio_{cover}(R_1, R_2) = Max\left(\frac{2}{3}, \frac{2}{4}\right) = 0.66$$

Since the ratio of examples covered is also higher than the $Nich_{Min}$ threshold, the rules R_1 and R_2 will be grouped in the same niche.

Based on these two properties we have also proposed a new similarity measure between rules that allows us to measure how peculiar the rules are [16]. This measure takes values in the interval [0,1], where values close to 0 represent similar rules and values close to 1 represent different rules, and is defined as:

$$dist_{OC}(R_1, R_2) = 1 - \frac{ratio_{overlap}(R_1, R_2) + ratio_{cover}(R_1, R_2)}{2} \tag{7}$$

3.2. Coding scheme and initial gene pool

In this paper, each chromosome is a vector of genes that represent the attributes and intervals of the rule. We have used a positional encoding, where the i-th attribute is encoded in the i-th gene used. To combine a condition selection with the learning of the intervals, each gene consists of four parts: t indicates the utility of the gene, if it is part of the antecedent (0), consequent (1) or is not involved in the rule (-1); s indicates the sign of an interval, i.e., whether it is positive (1) or negative (0); l and u represents the lower and upper bound of the interval of the attribute respectively. Notice that l and u are equal when we represent the values of a nominal attribute. So, a chromosome C has the following form:

$$C = G_1 G_2 \ldots G_n$$
$$G_i = (t_i, s_i, l_i, u_i), \quad i = 1, \ldots, n$$

where n is the number of attributes. We have defined *Amplitude* to limit increasing the intervals until they span the total domain. So that the positive intervals cannot have a size greater than *Amplitude* and the negative intervals cannot be smaller than *Amplitude*.

$$Amplitude_i = (Maximum_i - Minimum_i)/\rho \tag{8}$$

where ρ is defined by the user, denoting the trade-off between the specificity and generalization of the rules, and $Maximum_i$ and $Minimum_i$ represent the maximum and minimum value of the attribute i, respectively.

The population is consisted initially of a rule set in which the consequent only have one item (although this coding scheme allows many items to be handled in the consequent). Moreover, these rules will present a good coverage of the dataset to obtain information from the whole dataset. To accomplish this, first the attributes involved in the consequent and antecedent of the rule and whether their intervals will be positives or negatives are selected at random. Then an example from the dataset is randomly selected in order to generate the intervals of each attribute of the rule. To create an interval, the bounds of the intervals are generated with the value of the example selected fixed at the center of the interval. This interval will have a length equal to the $Amplitude_i$ / 2. Notice that if the lower or upper bound of the interval outdoes the domain of the attribute it will be replaced by the bound of the domain. Then, we mark the examples from the dataset that have been covered by this rule in order to generate the following rules based on unmarked examples. In this manner we assure the obtaining of an initial population with a good coverage of the dataset. We iteratively apply this mechanism in order to completely generate the initial pool of individuals. Moreover, if all the examples are marked and we are not completed the initial population, then we unmark all the examples and apply the process again until the initial population is complete.

3.3. Genetic Operators

The crossover operator interchanges genes from a pair of parents at random in order to generate two new solutions. Moreover, the Parent Centric BLX-α (PCBLX-α) operator [28] (an operator that is based on BLX-α) is applied to the boundaries of the intervals of the quantitative attributes. This operator is based on the concept of neighborhood that allows the offspring boundaries to be generated around the boundaries of one parent (see Fig. 1).

Let us assume that $X = [l_x, u_x]$ and $Y = [l_y, u_y]$, where $l_x, l_y, u_x, u_y \in [a_i, b_i] \subset \Re$, are two intervals of the quantitative attribute i which are going to be crossed. The offspring boundaries $[O_1^l, O_1^u]$ and $[O_2^l, O_2^u]$, are generated as follows:

- O_1^l is a randomly (uniformly) chosen value from the interval $[O^l_{1_{min}}, O^l_{1_{max}}]$, with $O^l_{1_{min}} = \max\{a_i, l_x - I^l \cdot \alpha\}$, $O^l_{1_{max}} = \min\{b_i, l_x + I^l \cdot \alpha\}$, and $I^l = | l_x - l_y |$. In our case, α has been fixed to 0.5.

- O_2^l is a randomly (uniformly) chosen value from the interval $[O^l_{2_{min}}, O^l_{2_{max}}]$, with $O^l_{2_{min}} = \max\{a_i, l_y - I^l \cdot \alpha\}$, $O^l_{2_{max}} = \min\{b_i, l_y + I^l \cdot \alpha\}$.

- O_1^u is a randomly (uniformly) chosen value from the interval $[O^u_{1_{min}}, O^u_{1_{max}}]$, with $O^u_{1_{min}} = \max\{a_i, u_x - I^u \cdot \alpha\}$, $O^u_{1_{max}} = \min\{b_i, u_x + I^u \cdot \alpha\}$, and $I^u = | u_x - u_y |$.

- O_2^u is a randomly (uniformly) chosen value from the interval $[O^u_{2_{min}}, O^u_{2_{max}}]$, with $O^u_{2_{min}} = \max\{a_i, u_y - I^u \cdot \alpha\}$, $O^u_{2_{max}} = \min\{b_i, u_y + I^u \cdot \alpha\}$.

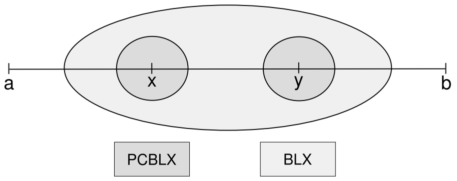
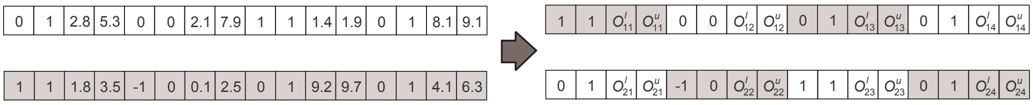


Figure 1: Scheme of the performance of BLX and PCBLX operators based on environments



0	1	2.8	5.3	0	0	2.1	7.9	1	1	1.4	1.9	0	1	8.1	9.1
1	1	1.8	3.5	-1	0	0.1	2.5	0	1	9.2	9.7	0	1	4.1	6.3

1	1	O^l_{11}	O^u_{11}	0	0	O^l_{12}	O^u_{12}	0	1	O^l_{13}	O^u_{13}	0	1	O^l_{14}	O^u_{14}
0	1	O^l_{21}	O^u_{21}	-1	0	O^l_{22}	O^u_{22}	1	1	O^l_{23}	O^u_{23}	0	1	O^l_{24}	O^u_{24}

Figure 2: An example of the crossover operator

A simple example of the crossover operator is shown in Fig. 2.The mutation operator modifies the components of a randomly selected gene. Firstly, the values of *t* and *s* are randomly modified. Later, a boundary is selected at random and its value is increased or decreased randomly, considering the same constraints as the initialization process.

The rules that present problems after applying the crossover and mutation operators are fixed by means of the repairing operator. One attribute is selected at random from the attributes not involved if the rule doesn't have any attribute in the part of the antecedent or consequent. If the consequent of a rule consists of more than one attribute, one attribute is randomly selected as consequent and the rest of the attributes

The rules that present problems after applying the crossover and mutation operators are fixed by means of the repairing operator. If the rule doesn't have an antecedent and/or consequent, these are selected at random from the attributes not involved in the rule. If the consequent of a rule consists of more than one attribute, one attribute is randomly selected as consequent and the rest of the attributes are moved to the antecedent.

Finally, in order to obtain simpler rules the size of the positive intervals is decreased while the covered examples are the same as those covered by the original intervals. For negative intervals, the size is increased, reducing the domain that each one covers.

3.4. Chromosome Evaluation

In order to evaluate a chromosome, the fitness function jointly maximizes three complementary metrics of the rules. This measure takes values between -1 and 2.5, and is defined as:

$$Fitness(C) = metric_1 + metric_2 + metric_3 \tag{9}$$

The first metric combines the classical measure support and the interestingness measure lift (see subsection 2.2) to measure the coverage of the rules while penalizing those rules of low interest. This metric provides interesting rules with a good trade-off between the specific and global. This metric takes values in the range [0,1] and is defined as:

$$metric_1 = \left(1 - \frac{1}{2^{10*support(A\to B)}}\right) * \left(1 - \frac{1}{lift(A \to B)}\right) \tag{10}$$

This expression ensures that there are no large differences between the obtained values for this metric when rules reach significant values for measures of support and lift. Notice that, the lift measure is not bounded above (where values higher than 3 are very significant for this measure) and we have used this arithmetic expression for the support measure (tuned empirically) in order to obtain a good balance between both measures (see subsection 2.2).

The second one measures how interesting the rule is for the user. To accomplish this, we use the interestingness measure netconf (see Table 1), which can distinguish between positive dependencies, negative dependencies or independence between items, and takes values in the range [-1,1].

$$metric_2 = netconf = \frac{SUP(AB) - SUP(A)SUP(B)}{SUP(A)(1 - SUP(A))} \tag{11}$$

Finally, the third one measures the number of items that contains the rule. The greater the number of items, the less comprehensible and useful will the rule be to the user. Their values range from 0 to 1, and are defined as follows, where $Attr_{A\to B}$ represents how many attributes are involved in A since we only take into account rules with one attribute in B.

$$metric_3 = \frac{1}{Attr_{A\to B} * 2} \tag{12}$$

For instance, let us consider the simple dataset with three attributes X_1, X_2 and X_3, and 6 training examples (see Table 2) and the rule $R2$ ($X_1 \in [18, 25] \to X_3 \in [5, 14]$) considered in subsection 3.1.4. The fitness of the rule $R2$ will be calculated as follows:

$$\begin{aligned} metric_1 &= \left(1 - \frac{1}{2^{10*support(A\to B)}}\right) * \left(1 - \frac{1}{lift(A \to B)}\right) \\ &= \left(1 - \frac{1}{2^{10*\frac{5}{6}}}\right) * \left(1 - \frac{1}{0.96}\right) = 0.04 \end{aligned}$$

$$metric_2 = \frac{SUP(AB) - SUP(A)SUP(B)}{SUP(A)(1 - SUP(A))} = \frac{\frac{4}{6} - \frac{5}{6}\frac{5}{6}}{\frac{5}{6}(1 - \frac{5}{6})} = 0.25$$

$$metric_3 = \frac{1}{Attr_{A\to B} * 2} = 0.5$$

$$fitness(R2) = 0.04 + 0.25 + 0.5 = 0.79$$

The rule $R2$ has a high support in this dataset (higher than 80%) and few involved attributes. However, this rule obtains a low value for fitness because its values for the measures lift (0.96 represents negative dependence) and netconf are poor.

It should be noted that we are concerned only with obtaining very strong rules [6], which indicates a positive dependence between items and resolves the problem of the support measure. For this reason, we will assign the worst fitness value to the rules that are not strong in order to remove them from the population.

3.5. Evolutionary model

The evolutionary model of our proposal is as follows. The initial gene pool is firstly obtained using the proposed data covering procedure. Then an offspring population is generated from the current population by selection, crossover and mutation. The punishment mechanism is applied to each new individual, penalizing the fitness value of an individual if it belongs to a niche of the solutions of the EP and is not better than the corresponding solution of the EP according to the fitness function. Then, the next population is constructed from the best individuals between the current and offspring population. Finally, when the current population percentage of new solutions is under $\alpha\%$ of solutions of the population (where α is usually 5%), we update the EP and the initialization process is again employed. To update the EP, first the seed of each niche in the current population is identified and then the rest of the individuals are classified according to the different identified niches (see subsection 3.1.1). The EP will be updated with the best solution of each niche, which should have a fitness value higher than $Ev_{Min}\%$ of the average fitness of the individuals of the EP. Finally, we check if any of the solutions of the EP belong to the same niche, in which case we keep only the best solution. This procedure is repeated until a stopping condition is fulfilled.

4. Experimental results

In order to evaluate the usefulness of the proposed approach, many experiments have been performed in this paper, which are organized in this section as follows:

- In subsection 4.1, we describe the experimental setup and a brief description of the methods considered in this study.
- In subsection 4.2, we analyze the influence of the $Nich_{Min}$ threshold and Ev_{Min} threshold on NICGAR.

Table 3: Datasets used for the experiments

Names	*Attributes(R/I/N)*	*Examples*	*Names*	*Attributes(R/I/N)*	*Examples*
Balance Scale (ba)	5 (5/0/0)	625	Satimage (sa)	37 (0/37/0)	6,435
Basketball (bas)	5 (3/2/0)	96	Segment (se)	20 (19/1/0)	2,310
Bolts (bo)	8 (2/6/0)	40	Sonar (so)	61 (60/0/1)	208
Coil2000 (co)	86 (0/86/0)	9,822	Spambase (sp)	58 (57/1/0)	4,597
House_16H (hh)	17(10/7/0)	22,784	Spectfheart (spe)	45 (0/45/0)	267
Fars (fa)	29 (5/0/24)	100,968	Stock Price (st)	10 (10/0/0)	950
Ionosphere (io)	34 (32/1/1)	351	Stulong (stu)	5(5/0/0)	1,419
Letter (le)	16 (0/16/0)	20,000	Texture (te)	41 (40/1/0)	5,500
Magic (ma)	11(10/0/1)	19,020	Thyroid (th)	22 (6/16/0)	7,200
Movement Libras (mo)	91 (90/0/1)	360	Vehicle (ve)	19 (0/18/1)	846
Optdigits (op)	65 (0/65/0)	5,620	Wdbc (wd)	31 (30/0/1)	569
Penbased (pe)	16 (0/16/0)	10,992	Wine (wi)	14 (13/1/0)	178
Pollution (po)	16 (16/0/0)	60	Vowel (vo)	14 (10/4/0)	990
Quake (qu)	4 (3/1/0)	2,178			

- In subsection 4.3, we compare the performance of our approach with two NGAs extended to extract PNQARs (Clearing [36, 35] and ASCGA [25]).
- In subsection 4.4, the usefulness of NICGAR is compared with four mono-objective evolutionary approaches (EARMGA [46], GAR [21], GENAR [33] and Alatasetal [2]), where Alatasetal mine PNQARs and the rest of the algorithms mine positive QARs.
- In subsection 4.5, two classical algorithms (Apriori [42] and Eclat [48]) for extracting positive association rules are compared with our proposal.
- In subsection 4.6, the performance of our approach is compared to two MOEAs: QAR-CIP-NSGA-II [32] for mining positive QARs and MOPNAR [32] for mining PNQARs.
- In subsection 4.7, we study the diversity of the sets of rules obtained by our proposal.
- In subsection 4.8, we analyze the scalability of the proposed approach.

Moreover, in order to provide additional material to the paper's content, we have developed an associated web page that can be found at http://sci2s.ugr.es/NICGAR/. This web page includes a brief summary of the analyzed algorithms, their set-up and the results they obtained in all datasets.

4.1. Experimental Set-Up

In the following, the main characteristics of the datasets used to analyze the proposed approach are shown. Then, we provide a brief description of the analyzed algorithms and their configurations (determining all the parameters used). And finally, we describe the statistical analysis that is used in this study.

4.1.1. Datasets

We evaluate the performance of our proposal considering 27 real-world datasets, in which the number of examples is within the interval [40, 100,968] and the number of attributes is within the interval [4, 91]. Table 3 summarizes the main characteristics of the 27 datasets, where "Attributes(R/I/N)" is the number of attributes (Real/Integer/Nominal) in the data and "Examples" is the number of examples. The datasets are available in the repository KEEL-dataset [3] where they can be downloaded (Available at `http://sci2s.ugr.es/keel/datasets.php`). To develop the different experiments, we consider the average results of 5 runs for each dataset. The results shown in the tables represent the average of the mean values of the measures obtained by the algorithms in all datasets.

Table 4: Parameters considered for the comparison

Algorithms	*Parameters*
Clearing	$PopSize = 100$, $N_{eval}=100000$, $P_{mut}= 0.1$, $\rho=3$, $Nich_{Min} = 0.5$
ASCGA	$PopSize = 100$, $N_{eval}=100000$, $P_{mut}= 0.1$, $\rho=3$
Alatasetal	$N_{eval}=100000$, $nInitialRandomChromo=12$, $r = 3$, $TournamentSize = 10$, $P_{sel}= 0.25$, $P_{cro} = 0.7$, $P_{mut\_min} = 0.05$, $P_{mut\_max} = 0.9$, $W_{sup} = 5$, $W_{conf} = 20$, $W_{amplRule} = 0.05$, $W_{amplInterv} = 0.02$, $W_{covered} = 0.01$
EARMGA	$PopSize = 100$, $N_{eval} = 100000$, $k = 2$, $P_{sel} = 0.75$, $P_{cro} = 0.7$, $P_{mut} = 0.1$, $\alpha = 0.01$
GAR	$PopSize = 100$, $nItemset = 100$, $N_{eval} = 100000$, $P_{sel}= 0.25$, $P_{cro} = 0.7$, $P_{mut} = 0.1$, $\omega = 0.4$, $\Psi= 0.7$, $\mu= 0.5$, $minSup = 0.1$, $minConf = 0.8$
GENAR	$PopSize = 100$, $N_{eval} = 100000$, $P_{sel}= 0.25$, $P_{cro} = 0.7$, $P_{mut} = 0.1$, $nRules = 30$, $FP = 0.7$, $AF = 0.2$
QAR-CIP-NSGA-II	$PopSize = 100$, $N_{eval}=50000$, $P_{mut}= 0.1$, $\delta=2$, $\alpha = 5\%$
MOPNAR	$N_{eval}=100000$, $H=13$, $m=3$, $PopSize=N^{m-1}_{H+m-1}$, $T=10$, $\delta=0.9$, $\eta_r=2$, $\gamma=2$, $P_{mut}= 0.1$, $\alpha = 5\%$
Apriori	$minSup = 0.1$, $minConf = 0.8$
Eclat	$minSup = 0.1$, $minConf = 0.8$
NICGAR	$PopSize = 100$, $N_{eval}=100000$, $P_{mut}= 0.1$, $\rho=3$, $Nich_{Min} = 0.5$, $Ev_{Min} = 0.85$, $\alpha = 5\%$

4.1.2. Algorithms and parameters considered for comparison

In these experiments, we compare the proposed approach with ten other algorithms: Clearing [36, 35] and ASCGA [25] are classical NGAs (see subsection 2.1), which we have extended to mine PNQARs considering the same fitness function, initial gene pool, code scheme, genetic operators and the process to determine when two individuals will belong to the same niche (see subsection 3.1) as in our proposal; EARMGA [46], GENAR [33] and Alatasetal are mono-objective evolutionary algorithms, in which Alatasetal enables the obtention of PNQARs and the other two algorithms mine positive QARs; GAR [21] searches frequent itemsets and it is necessary to run an additional procedure to generate positive QARs; Apriori [42] and Eclat [48] extract positive rules whose support and confidence are greater than a minimum support (minSup) and minimum confidence (minConf) given by the user; and QAR-CIP-NSGA-II [32] and MOPNAR [31] are MOEAs to mine positive QARs and PNQARs, respectively. Notice that all of these algorithms are available from the KEEL software tool [5] and a brief description of them can be found at http://sci2s.ugr.es/NICGAR/.

The parameters of the analyzed algorithms are presented in Table 4. We have selected values for our proposal that work well for the majority of the datasets and for the remaining methods the parameters were selected following the instructions of the authors of each approach. Notice that Apriori, Eclat and GAR need a minSup and a minConf to extract association rules. In order to facilitate comparisons, we have selected 0.1 and 0.8 for minSup and minConf, respectively, which are standard common values that work well in most cases, instead of searching for specific values for each one.

4.1.3. Statistical Analysis

In order to compare the obtained results, we have used nonparametric tests for multiple comparison to find the best approach (for a detailed description of these tests, see http://sci2s.ugr.es/sicidm/). We have applied statistical tests [15, 40] to the average results that the analyzed algorithms obtained for the interestingness measures CF, netconf, yulesQ and lift, and the diversity measure (these results can be found on the web page associated with the paper at http://sci2s.ugr.es/NICGAR/). Since the algorithms reach infinity in the majority of the cases, the conviction metric has been not considered in this study. The mean values of the interestingness metrics are computed as a *MeanS* function as follows:

- *For the measures CF, netconf and yulesQ:*

$$MeanS = \begin{cases} (\mid meanValue \mid \ / \ 2) & if\ meanValue \leq 0 \\ (meanValue \ / \ 2) + 0.5 & otherwise \end{cases}$$

- *For the measure lift:*

$$MeanS = \begin{cases} 1 - (0.5 \ / \ meanValue) & if\ meanValue > 1 \\ 0.5 - (meanValue \ / \ 2) & if\ 0 \leq meanValue \leq 1 \end{cases}$$

Table 5: Analysis of the performance depending on $Nich_{Min}$ threshold with Ev_{Min} = 0.85

$Nich_{Min}$	#R	$Av_{Sup}(\sigma)$	$Av_{Conf}(\sigma)$	$Av_{Lift}(\sigma)$	Av_{Conv}	$Av_{CF}(\sigma)$	$Av_{NetConf}(\sigma)$	$Av_{YulesQ}(\sigma)$	$Av_{Amp}(\sigma)$	$Av_{Div}(\sigma)$	$\%Tran(\sigma)$
0.4	18.37	0.22 (0.04)	**0.93** (0.01)	8.76 (3.02)	∞	**0.89** (0.03)	**0.84** (0.02)	**0.98** (0.01)	**2.08** (0.02)	**0.86** (0.02)	92.91 (4.56)
0.5	19.79	0.22 (0.03)	**0.93** (0.01)	**8.81** (3.12)	∞	**0.89** (0.03)	**0.84** (0.02)	**0.98** (0.01)	2.09 (0.03)	0.85 (0.02)	95.57 (3.01)
0.6	**20.81**	**0.23** (0.04)	**0.93** (0.02)	7.56 (3.02)	∞	0.88 (0.03)	0.83 (0.03)	0.97 (0.01)	2.09 (0.02)	0.83 (0.02)	**95.87** (4.58)

Table 6: Analysis of the performance depending on Ev_{Min} threshold with $Nich_{Min}$ = 0.5

Ev_{Min}	#R	$Av_{Sup}(\sigma)$	$Av_{Conf}(\sigma)$	$Av_{Lift}(\sigma)$	Av_{Conv}	$Av_{CF}(\sigma)$	$Av_{NetConf}(\sigma)$	$Av_{YulesQ}(\sigma)$	$Av_{Amp}(\sigma)$	$Av_{Div}(\sigma)$	$\%Tran(\sigma)$
0.8	**22.24**	**0.22** (0.03)	0.93 (0.01)	**9.01** (3.91)	∞	0.87 (0.03)	0.82 (0.02)	0.97 (0.01)	2.11 (0.03)	0.84 (0.02)	**96.57** (2.42)
0.85	19.79	**0.22** (0.03)	0.93 (0.01)	8.81 (3.12)	∞	0.89 (0.03)	0.84 (0.02)	**0.98** (0.01)	2.09 (0.03)	**0.85** (0.02)	95.57 (3.01)
0.9	16	**0.22** (0.02)	**0.94** (0.02)	7.98 (2.73)	∞	**0.9** (0.02)	**0.85** (0.02)	**0.98** (0.01)	**2.08** (0.02)	**0.85** (0.04)	89.64 (4.82)

where $meanValue$ is the corresponding metric average. $MeanS$ ensures well-defined differences on the used metrics since it provides values in [0,1], where independence is represented by zero, positive dependences are represented by values higher than 0.5, and negative dependences are represented by values less than or equal to 0.5. Notice that low values of negative dependence are represented by low values of $MeanS$, close to the value for independence, as in the case of the interestingness measures (see subsection 2.2).

To analyze the results obtained in the comparison with the NGAs we have used a Wilcoxon's Signed-Ranks test [45] with a level of significance of 0.05. To compare the obtained results with the four mono-objective evolutionary algorithms and the two MOEAs to extract QARs we have used the Friedman [14] test in order to find out whether significant differences exist among all the mean values. Once Friedman's test rejects the null hypothesis, we can proceed with a post-hoc test in order to find the concrete pairwise comparisons which produce differences. In the comparison with the four mono-objective evolutionary algorithms we apply the Holm's test [20] to compare the control algorithm (the one that obtains the best ranking) with the remaining methods. Moreover, we apply the Shaffer's test [39] to perform all pairwise comparisons with the two MOEAs to extract QARs. We will compute the adjusted p-value (APV) associated with each comparison, which represents the lowest level of significance of a hypothesis that results in a rejection. This value differs from the standard p-value in the sense that it determines univocally whether the null hypothesis of equality is rejected at a significant level. This facilitates the comparison of the algorithms, as it is no longer necessary to contrast each one with the α/i value of a standard statistical results table. Finally, we have also used Wilcoxon's Signed-Ranks test to compare statistically the results obtained with the classical algorithms.

4.2. Analysis of the influence of the $Nich_{Min}$ threshold and Ev_{Min} threshold on NICGAR

In this subsection, we have been carried out several experiments to analyze the performance of our proposal depending on the $Nich_{Min}$ threshold and Ev_{Min} threshold. In order to make this analysis easier to interpret, we have used three different values for the $Nich_{Min}$ threshold (0.4, 0.5 and 0.6) and Ev_{Min} threshold (0.8, 0.85 and 0.9). Table 5 and Table 6 show the average and the standard deviation (shown in parentheses) of the results obtained, where #R represents the number of the association rules obtained, Av_{Sup}, Av_{Conf}, Av_{Conv}, Av_{CF}, $Av_{NetConf}$ and $Av_{Yule'sQ}$ are, respectively, the average value for the measures support, confidence, lift, CF, netconf, and yule'sQ of the set of rules generated. Av_{Amp} represents the average of the number of attributes involved in the rules, Av_{Div} is the average value of the diversity measure of the rules obtained, and $\%Tran$ is the percentage of examples of the database that are covered by the rules generated. The value for the diversity measure represents the average of the mean value of the dist_{OC} measure (see equation 7) of each rule with the rest of the generated rules. Notice that the value ∞ stands for the maximum value for some measures (see subsection 2.2).

Taking into account the results shown in Table 5 we can highlight how the number of rules decreases when the $Nich_{Min}$ threshold decreases because a higher number of rules can reach the $Nich_{Min}$ threshold and are included in the niches created. This fact complicates the formation of new niches and, consequently, reduces the number of rules generated. The knowledge obtained from the dataset will be very different since the rules must be very different to belong to different niches. However, the coverage of the dataset decreases due to the formation of niches in specific areas of the search space being more difficult.

On the other hand, the results shown in Table 6 highlight how the mean values of the interestingness measures increase when the value of the Ev_{Min} threshold increases because the rules must present a higher quality to be added

Table 7: Results of the average value of the measures for all datasets in the comparison with mono-objective evolutionary approaches

$Algorithm$	$\#R$	$Av_{Sup}(\sigma)$	$Av_{Conf}(\sigma)$	$Av_{Lift}(\sigma)$	Av_{Conv}	$Av_{CF}(\sigma)$	$Av_{NetConf}(\sigma)$	$Av_{YulesQ}(\sigma)$	$Av_{Amp}(\sigma)$	$Av_{Div}(\sigma)$	$\%Tran(\sigma)$
Clearing	**25.92**	0.17 (0.07)	0.91 (0.04)	**25.57** (16.21)	∞	**0.83** (0.08)	0.80 (0.06)	0.90 (0.05)	2.1 (0.09)	0.83 (0.05)	92.72 (9.06)
ASCGA	15.93	**0.22** (0.05)	0.82 (0.05)	7.27 (6.45)	∞	0.69 (0.09)	0.62 (0.09)	0.82 (0.07)	2.36 (0.03)	0.74 (0.08)	85.60 (10.93)
NICGAR	19.79	**0.22** (0.03)	**0.93** (0.01)	8.81 (3.12)	∞	**0.83** (0.02)	**0.84** (0.02)	**0.97** (0.01)	**2.09** (0.03)	**0.85** (0.02)	**95.57** (3.01)

Table 8: Wilcoxon's test ($\alpha = 0.05$) on the different measures for the NGAs

$Measure$	$Algorithms$	R^+	R^-	$Hypothesis$	$p-value$
CF	NICGAR vs. Clearing	291	60	**Rejected**	0.00241
	NICGAR vs. ASCGA	371.5	6.5	**Rejected**	<0.001
netconf	NICGAR vs. Clearing	251.5	99.5	**Rejected**	0.05
	NICGAR vs. ASCGA	373.5	4.5	**Rejected**	<0.001
yulesQ	NICGAR vs. Clearing	342.5	35.5	**Rejected**	<0.001
	NICGAR vs. ASCGA	375.5	2.5	**Rejected**	<0.001
lift	NICGAR vs. Clearing	38	340	**Rejected for Clearing**	<0.001
	NICGAR vs. ASCGA	254	124	**Not Rejected**	0.12
diversity	NICGAR vs. Clearing	244.5	133.5	**Not Rejected**	0.18
	NICGAR vs. ASCGA	376	2	**Rejected**	<0.001

to the EP. However, the coverage of the dataset usually decreases due to the fact that, often we cannot mine very high quality knowledge from all of the search space.

Therefore, we will use 0.5 for the $Nich_{Min}$ threshold and 0.85 for the Ev_{Min} threshold for the rest of our experiments because these values allow us to maintain a good trade-off between the number of rules, quality and coverage of the dataset.

4.3. Comparison with other NGAs: Clearing and ASCGA

This section compares the performance of our algorithm with two NGAs: Clearing [36, 35] and ASCGA [25]. In order to use these two algorithms in this way we have extended them to mine PNQARs considering the same coding scheme, initial gene pool, fitness function and genetic operators as in our proposal. Furthermore, we have used the same process to determine when two individuals will belong to the same niche (see subsection 3.1). Table 7 shows the obtained results by the analyzed algorithms (the description of this table can be found in subsection 4.2). To statistically compare the obtained results of the interesting measures, we have used a Wilcoxon's Signed-Ranks test [45] with a level of significance of 0.05. Table 8 shows the results obtained by this test.

Table 7 shows that all the NGAs analyzed allow a diverse and reduced set of association rules to be obtained. However, our proposal obtains rules with a higher average coverage of the datasets than the rest of the analyzed algorithms (in almost all the datasets more than 90%, see the associated web page at http://sci2s.ugr.es/NICGAR/), providing us with interesting knowledge of the whole of the datasets. Moreover, the results of the statistical test show that the null hypothesis is rejected for all the interestingness measures except for the lift measure, because this measure is not bounded above and the analyzed methods obtain very high values for lift in some datasets. Finally, the null hypothesis for the diversity measure is rejected with ASCGA but not with Clearing, as Clearing is an NGA that encourages a high diversity in the search process, where only the best individuals of each niche will survive, and as we mentioned before, it uses the same process as our proposal to determine when two individuals belong to the same niche. Even so, NICGAR achieves a better ranking for this measure than Clearing.

Fig. 3 and Fig. 4 show the boxplot graphics for the results of the interestingness measures CF and netconf for the rules obtained by NICGAR, respectively. We can see how the obtained rules present values for the interestingness measures near to each other, obtaining low values for standard deviation in these measures. Moreover, we can see how only 25% of the rules were unable to achieve values higher than 0.6 for CF and netconf measures (except for stulong) and how all of them obtained positive values for these measures, which represents positive dependence between the items of the rules.

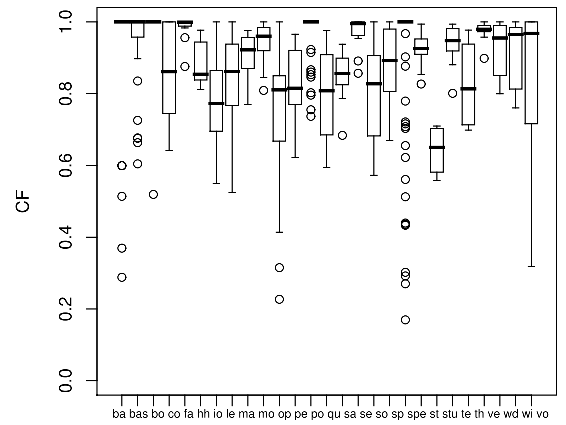


Figure 3: Obtained boxplots on all datasets considering the CF measure.

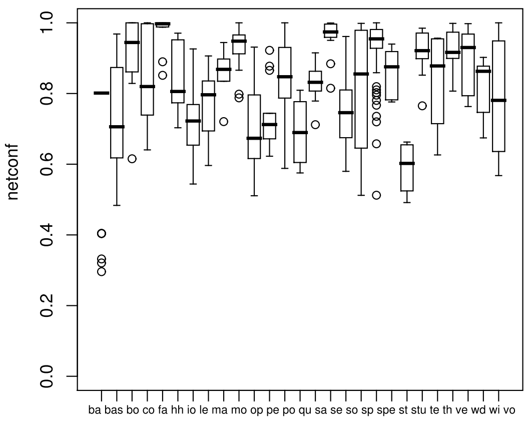


Figure 4: Obtained boxplots on all datasets considering the netconf measure.

Table 9: Results of the average value of the measures for all datasets in the comparison with mono-objective evolutionary approaches

Algorithm	*#R*	$Av_{Sup}(\sigma)$	$Av_{Conf}(\sigma)$	$Av_{Lift}(\sigma)$	Av_{Conv}	$Av_{CF}(\sigma)$	$Av_{NetConf}(\sigma)$	$Av_{YulesQ}(\sigma)$	$Av_{Amp}(\sigma)$	$Av_{Div}(\sigma)$	$\%Tran(\sigma)$
EARMGA	**94.97**	0.35 (0.1)	**1** (0)	1.02 (0.03)	∞	0.05 (0.06)	0.01 (0.01)	0.03 (0.03)	**2.01** (0.02)	0.58 (0.06)	**99.31** (1.55)
GAR	59.51	**0.58** (0.03)	0.88 (0.01)	1.4 (0.12)	∞	0.4 (0.03)	0.33 (0.03)	0.57 (0.04)	2.03 (0.03)	0.23 (0.05)	84.32 (3.58)
GENAR	29.3	0.24 (0)	0.84 (0.01)	3.38 (0.07)	∞	0.59 (0.01)	0.4 (0.01)	0.68 (0.01)	28.93 (0)	0.33 (0.01)	64.25 (1.3)
Alatasetal	24.95	0.27 (0.09)	0.63 (0)	**10.85** (17.37)	∞	0.33 (0.09)	0.14 (0.1)	0.24 (0.12)	3.66 (0.61)	0.15 (0.05)	47.36 (9.89)
NICGAR	19.79	0.22 (0.03)	0.93 (0.01)	8.81 (3.12)	∞	**0.89** (0.03)	**0.84** (0.02)	**0.98** (0.01)	2.09 (0.03)	**0.85** (0.02)	95.57 (3.01)

Table 10: Friedman's test ($\alpha = 0.05$) on the different measures for the mono-objective evolutionary algorithms

	CF	*netconf*	*yulesQ*	*lift*	*diversity*
Critical Value	11.07	11.07	11.07	11.07	11.07
Statistic (X_F^2)	**56.57**	**80.21**	**82.10**	**54.30**	**82.46**
p **value**	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001

4.4. Comparison with mono-objective evolutionary approaches for mining association rules

We have carried out several experiments to analyzes the usefulness of the proposed approach together with the mono-objective evolutionary algorithms: EARMGA [46], GAR [21], GENAR [33] and Alatasetal [2], where Alatasetal mines PNQARs and the rest of the algorithms mine positive QARs. Table 9 shows the obtained results by the analyzed algorithms (the description of this table can be found in subsection 4.2). The results presented in this table show that our proposal obtains reduced sets of short rules (almost 20 rules on average) with a good average support, presenting a good trade-off between specific and general rules, and avoiding the extraction of redundant rules that provide us with similar information to the dataset. Moreover, these rules allow us to have information of the whole search space, presenting a high coverage of the dataset. Notice that only EARMGA obtains a better average coverage than our proposal but this method should be the worst when we compare the results obtained in the interestingness measures.

Table 10 shows the Friedman statistics, and it relates them to the corresponding critical values for each distribution by using the level of significance $\alpha = 0.05$. The p-value obtained is also reported for this test. Given that the level of significance is clearly greater than their associated critical values, there are significant differences among the observed results with a level of significance $\alpha \leq 0.05$.

Table 11 shows the rankings (which are computed with the use of a Friedman test) of the different methods that are considered in this study and the adjusted p-value obtained by the Holm's post-hoc procedure. Notice that our algorithm obtains the best ranking for all the measures. Moreover, we found very low adjusted p-values, pointing out that significant difference can be detected between our proposal and the rest of the algorithms. Hence, NICGAR is the

Table 11: Average Friedman rankings and adjusted p-*values* using Holm's test, with NICGAR as the control algorithm

Measure	*Algorithms*	*Friedman ranking*	*Adjusted p − value*
CF	EARMGA	4.42	0
	Alatasetal	3.62	0.000001
	GAR	3.05	0.000256
	GENAR	2.48	0.012563
	NICGAR	1.40	-
netconf	EARMGA	4.55	0
	Alatasetal	4.03	0
	GAR	2.68	0.000362
	GENAR	2.64	0.000362
	NICGAR	1.07	-
yule'sQ	EARMGA	4.59	0
	Alatasetal	4.12	0
	GAR	2.79	0.00043
	GENAR	2.27	0.012563
	NICGAR	1.20	-
lift	EARMGA	4.48	0
	Alatasetal	3.46	0.000019
	GAR	3.16	0.000256
	GENAR	2.37	0.047757
	NICGAR	1.51	-
diversity	Alatasetal	4.35	0
	GAR	4.01	0
	GENAR	3.40	0
	EARMGA	2.22	0.004509
	NICGAR	1	-

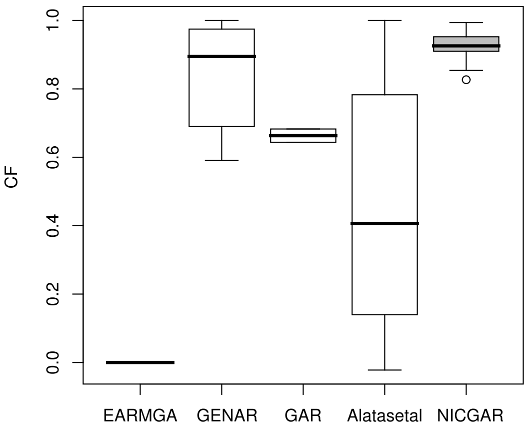


Figure 5: Obtained boxplots by the mono-objective evolutionary algorithms on the dataset stock considering the CF measure

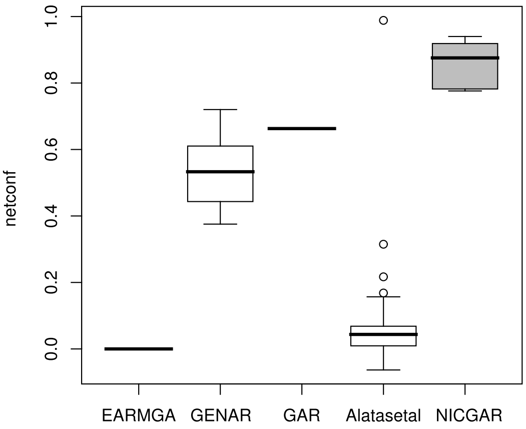


Figure 6: Obtained boxplots by the mono-objective evolutionary algorithms on the dataset stock considering the netconf measure

best performing method in all the measures when compared with the remaining methods analyzed in this study. Notice that NICGAR presents significant differences with Alatasetal for the measure lift even though Alatasetal obtains a better average value than NICGAR (see table 9) because, as we mentioned before, this measure is not bounded above and Alatasetal obtains very high values for this measure in some datasets (obtaining high values for the standard deviation).

The results of the interestingness measures CF and netconf for the obtained rules by the analyzed algorithms on the dataset stock are presented in the boxplot graphics shown in Fig. 5 and Fig. 6, respectively. These boxplots show that NICGAR obtains values of CF and netconf higher than the rest of the algorithms, and most of the values of the rules are close to the best that this measure can achieve. Moreover, some of the rules obtained by EARMGA and Alatasetal denote negative dependence or independence between the items considering these measures.

Table 12: Results of the average value of the measures for all datasets in the comparison with classical algorithms

Algorithm	*#R*	$Av_{Sup}(\sigma)$	$Av_{Conf}(\sigma)$	$Av_{Lift}(\sigma)$	Av_{Conv}	$Av_{CF}(\sigma)$	$Av_{NetConf}(\sigma)$	$Av_{YulesQ}(\sigma)$	$Av_{Amp}(\sigma)$	$Av_{Div}(\sigma)$	$\%Tran(\sigma)$
Apriori	**8345221.46**	0.16 (0)	**0.93** (0)	4.09 (0)	∞	0.81 (0)	0.64 (0)	0.84 (0)	5.13 (0)	0.51 (0)	90.43 (0)
Eclat	**8345221.46**	0.16 (0)	**0.93** (0)	4.09 (0)	∞	0.81 (0)	0.64 (0)	0.84 (0)	5.13 (0)	0.51 (0)	90.43 (0)
NICGAR	19.79	**0.22** (0.03)	**0.93** (0.01)	**8.81** (3.12)	∞	**0.89** (0.03)	**0.84** (0.02)	**0.98** (0.01)	**2.09** (0.03)	**0.85** (0.02)	**95.57** (3.01)

Table 13: Wilcoxon's test ($\alpha = 0.05$) on the different measures for the classical algorithms

Measure	*Algorithms*	R^+	R^-	*Hypothesis*	$p - value$
CF	NICGAR vs. Apriori	86	19	**Rejected**	0.003
	NICGAR vs. Eclat	86	19	**Rejected**	0.003
netconf	NICGAR vs. Apriori	100.5	19.5	**Rejected**	0.019
	NICGAR vs. Eclat	100.5	19.5	**Rejected**	0.019
yule'sQ	NICGAR vs. Apriori	95	25	**Rejected**	0.04
	NICGAR vs. Eclat	95	25	**Rejected**	0.04
lift	NICGAR vs. Apriori	74	46	**Not Rejected**	≥ 0.2
	NICGAR vs. Eclat	74	46	**Not Rejected**	≥ 0.2
diversity	NICGAR vs. Apriori	117	3	**Rejected**	<0.001
	NICGAR vs. Eclat	117	3	**Rejected**	<0.001

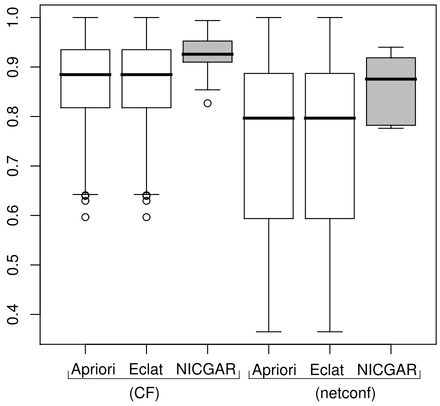


Figure 7: Obtained boxplots by classical algorithms on the dataset stock considering the CF and netconf measures.

4.5. Comparison with classical algorithms: Apriori and Eclat

We analyze the usefulness of NICGAR together with two classical algorithms for mining positive association rules, Apriori [42] and Eclat [48]. In order to compare with the classical methods Apriori and Eclat, the domains of the quantitative attribute have to be partitioned. In this case, we do not have extra information to use algorithms based on information theory or other concepts, hence, an usual discretization algorithm has been applied to partition the domain in to 4 uniformly distributed intervals [27].

Table 12 shows the obtained results by the classical algorithms (the description of this table can be found in subsection 4.2). This table shows only the average results obtained by Apriori and Eclat on 15 datasets due to the fact that they cannot be run in all the datasets due to scalability problems. Notice that the values of the standard deviation obtained by Apriori and Eclat are 0 because they are deterministic, performing only one run. In this case, we have used a Wilcoxon's Signed-Ranks test [45] with a level of significance of 0.05 to statistically compare the obtained results with the analyzed methods for the interesting measures. Table 13 shows the results obtained by this test.

Analyzing the results presented in Table 12 we can see that NICGAR obtains the smallest rule sets and with the best average coverage of the datasets. On the other hand, the results of the statistical test show that the null hypothesis is rejected for all the measures analyzed except for the lift measure, even though NICGAR obtains better average lift than the rest of the method. As we have previously commented, this measure is not bounded above and the classical methods obtain very high values for lift in some datasets.

Fig. 7 presents a boxplot that represents the results of the interestingness measures CF and netconf on the dataset stock for the obtained rules from Apriori, Eclat and NICGAR. We can see how NICGAR presents better values of CF

Table 14: Results of the average value of the measures for all datasets in the comparison with MOEAs algorithms to mine QARs

Algorithm	*#R*	$Av_{Sup}(\sigma)$	$Av_{Conf}(\sigma)$	$Av_{Lift}(\sigma)$	Av_{Conv}	$Av_{CF}(\sigma)$	$Av_{NetConf}(\sigma)$	$Av_{YulesQ}(\sigma)$	$Av_{Amp}(\sigma)$	$Av_{Div}(\sigma)$	$\%Tran(\sigma)$
QAR-CIP-NSGA-II	**149.36**	0.14 (0.02)	**0.94** (0.01)	**450.06** (83.99)	∞	**0.92** (0.01)	0.79 (0.03)	0.91 (0.02)	2.91 (0.14)	0.81 (0.02)	93.07 (2.53)
MOPNAR	89.63	0.3 (0.04)	0.93 (0.03)	14.11 (3.55)	∞	0.9 (0.03)	0.74 (0.04)	**0.98** (0.01)	2.88 (0.19)	0.58 (0.06)	**99.82** (0.31)
NICGAR	19.79	**0.22** (0.03)	0.93 (0.01)	8.81 (3.12)	∞	0.89 (0.03)	**0.84** (0.02)	**0.98** (0.01)	**2.09** (0.03)	**0.85** (0.02)	95.57 (3.01)

Table 15: Friedman's test (α = 0.05) on the different measures for the MOEAs

	Friedman Test				
	CF	*netconf*	*yulesQ*	*lift*	*diversity*
Critical Value	7.81	7.81	7.81	7.81	7.81
Statistic (X_F^2)	**15.01**	**14.74**	**6.68**	**45.85**	**41.68**
***p* value**	<0.0001	<0.0001	0.035	<0.0001	<0.0001

Table 16: Average Friedman rankings on the different measures for the MOEAs

Algorithms	**Rankings**				
	CF	**netconf**	**yule'sQ**	**lift**	**diversity**
MOPNAR	2.11	2.55	**1.64**	2.18	3
QAR-CIP-NSGA-II	**1.42**	1.92	2.35	**1**	1.64
NICGAR	2.46	**1.51**	2	2.81	**1.35**

Table 17: Adjusted p-values obtained by the Shaffer Test

Measure	*i*	Algorithms	*Adjusted p − value*
CF	1	QAR-CIP-NSGAII vs NICGAR	0.000416
	2	QAR-CIP-NSGAII vs MOPNAR	0.011818
	3	MOPNAR vs NICGAR	0.196085
netconf	1	NICGAR vs MOPNAR	0.000416
	2	QAR-CIP-NSGAII vs MOPNAR	0.0207
	3	NICGAR vs QAR-CIP-NSGAII	0.134417
yule'sQ	1	MOPNAR vs QAR-CIP-NSGAII	0.029165
	2	NICGAR vs QAR-CIP-NSGAII	0.196085
	3	MOPNAR vs NICGAR	0.196085
lift	1	QAR-CIP-NSGAII vs NICGAR	0
	2	QAR-CIP-NSGAII vs MOPNAR	0.000013
	3	MOPNAR vs NICGAR	0.0207
diversity	1	NICGAR vs MOPNAR	0
	2	QAR-CIP-NSGAII vs MOPNAR	0.000001
	3	NICGAR vs QAR-CIP-NSGAII	0.276303

and netconf than the classical algorithms, and values close to the best that these measures can achieve.

4.6. *Comparison with multi-objective evolutionary approaches for mining association rules*

We illustrate the behavior of NICGAR with respect to two recent MOEAs: QAR-CIP-NSGA-II [32] for mining positive QARs and MOPNAR [32] for mining PNQARs. Table 14 shows the obtained results by the analyzed algorithms (the description of this table can be found in subsection 4.2). Table 15 shows the statistical results obtained by the Friedman test (the description of this table can be found in subsection 4.4), which show the significant differences obtained among the observed results with a level of significance $\alpha \leq 0.05$. Moreover, Table 16 shows the rankings (which are computed with the use of a Friedman test) of the different methods that are considered in this study and Table 17 the adjusted p-value obtained by Shaffer's post-hoc procedure. Analyzing the results shown in these tables, we can draw the following conclusions:

- QAR-CIP-NSGA-II allows more specific rule sets to be mined than the rest of algorithms (with 0.14 on average support) with very high values for the interestingness measures, presenting the best ranking of the Friedman test for half of the quality measures analyzed in comparison with the rest of the algorithms. However, this algorithm

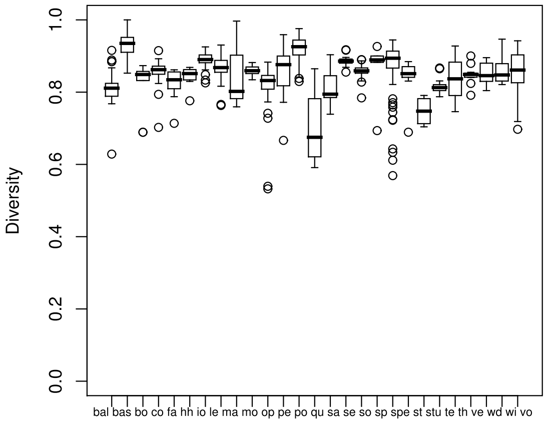


Figure 8: Obtained boxplots on all the datasets considering the diversity measure.

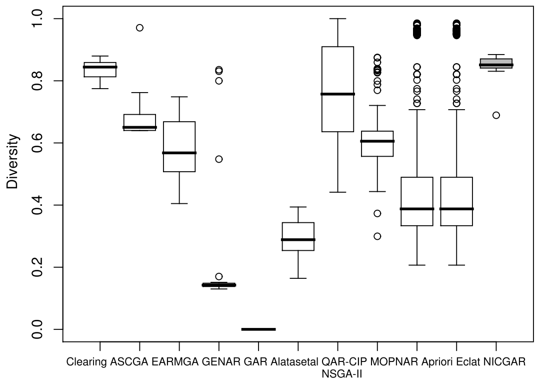


Figure 9: Obtained boxplots by all the algorithms on the dataset stock considering the diversity measure.

obtains the largest rule sets and with the lowest average coverage of the datasets due to that most of its rules are more specific.

- MOPNAR allows us to obtain reduced sets of PNQARs (almost half the amount of rules obtained by QAR-CIP-NSGA-II), with the best average coverage of the datasets and low values of the standard deviation. Moreover, the obtained rules by MOPNAR obtain good values for the interestingness measures on all the datasets that are close to the obtained rules by the analyzed algorithms.

- NICGAR provides reduced sets of PNQARs (less than 20 rules on average) with the lowest average number of attributes and a good average coverage of the datasets (higher than 95% on average). On the other hand, the obtained rules show high values for the interestingness measures, obtaining the best average values for netconf and yule'sQ and average values for CF and lift similar to the obtained rules by the analyzed algorithms (as we mentioned before, lift measure is not bounded above and values higher than 3 are very significant for this measure). Finally, NICGAR obtains the best ranking for the diversity measure, mining rule sets which provide us with diverse information of the dataset.

The three analyzed algorithms allow us to obtain interestingness association rules, which will be used according to the specific necessity of each user. The users can base their selection on the potentialities that each method provides, with the most significant being the following:

- QAR-CIP-NSGA-II: more specific association rules with a very high quality.

- MOPNAR: reduced sets of interestingness PNQARs with the best coverage of the datasets.

- NICGAR: PNQARs sets with a high diversity, quality and a good coverage of the datasets.

Finally, we can highlight that NICGAR allows us to obtain reduced and diverse sets of rules that are easy to understand, interesting and with a good coverage of the dataset.

4.7. The diversity of the set of rules obtained by some evolutionary approaches

In this section we analyze the diversity of the rule sets obtained by the analyzed methods. Fig. 8 shows a boxplot that presents the values of the diversity measure (see subsection 4.2) for the rules obtained by NICGAR on all the datasets. These figures show that all the rule sets obtained present a high level of diversity with values higher than 0.7 for more than 75% of rules on all datasets (except for the dataset quake). Fig. 9 shows a boxplot that presents the values of the diversity measure for the rules obtained by the analyzed methods on the dataset stock. Moreover, the obtained rules by NGAs provide more diverse knowledge than the rest of the methods. Notice that the classical

Table 18: Obtained rules by some evolutionary approaches from stock dataset

Algorithm	*RuleSet*	*Sup*	*CF*	*Netconf*	*Diversity*
NICGAR	R1: If **Company4** is not [47.37, 59.87] then **Company7** is not [74, 85.87]	0.27	0.95	0.93	0.87
	R2: If **Company1** is not [31.89, 61.5] then **Company2** is [49.0, 55.75]	0.33	0.97	0.93	0.84
	R3: If **Company2** is [22.12, 36] then **Company5** is [30.12551.87]	0.24	0.99	0.91	0.86
Alatasetal	R1: If **Company5** is [90.37, 93] then **Company1** is not [31.89, 61.5]	0.02	1	0.66	0
	R2: If **Company5** is [93, 93] then **Company1** is not [31.89, 61.5]	0.001	1	0.65	0
	R3: If **Company5** is [90.375, 93.0] and **Company4** is [44.37, 45.87] then **Company1** is not [31.89, 61.5]	0.02	1	0.66	0
GAR	R1: If **Company3** is [19.24, 22.37] then **Company2** is [49.34, 59.14]	0.39	0.69	0.67	0
	R2: If **Company2** is [49.34, 59.14] then **Company3** is [19.24, 22.37]	0.39	0.64	0.66	0

methods Apriori and Eclat present a wide range of values of diversity (0.2 to 1) due to many of the extracted rules being redundant.

Table 18 shows some of the rules obtained by our proposal from one of the 5 runs performed in the dataset stock, where *RuleSet* is the generated rule, *Sup* is the support of the rules, *CF* is the value for the CF measure of the rules, *Netconf* is the value for the netconf measure of the rules and *Diversity* is the value of the diversity of the rule. We can see that the rules provide us with different information about the problem with values higher than 0.85 for almost all the rules. This table also shows some of the rules obtained by two of the analyzed methods: Alatasetal and GAR. Analyzing these rules, we can find some problems:

- The rules obtained by Alatasetal show how $R2$ is subsumed by the rule $R1$, where both of them involve the same attributes, $R1$ covers all the examples covered by $R2$, and moreover $R1$ achieves better values for the quality measures. In cases such as these, rules like $R2$ should be removed.
- The rules $R1$ and $R3$ obtained by Alatasetal show how adding more attributes to the antecedent of the rule does not affect the prediction of the consequent. Rules like $R3$ do not improve the quality of the information provided and are more difficult to understand for the user, since they involve more attributes in the antecedent. Thus, this kind of rule should be removed in order to reduce the number of rules provided to the user.
- The rules $R1$ and $R2$ obtained by GAR are the same rule but with the antecedent and consequent exchanged. In this case, if there are significant differences for the interestingness measures between the rules then only the best rule should be provided to the user.

4.8. Analysis of scalability and complexity

We present some experiments that have been carried out to analyze the scalability of the algorithms in the datasets Movement Libras and Fars, which present the highest number of attributes and examples, respectively, of the datasets used for the experiments (see Table 3). In order to perform the experimental study, we have used an Intel Core i7, 2.80 GHz CPU with 12 Gb of memory and running Linux. The average runtime expended by the analyzed algorithms when the number of attributes increases on the dataset Movement Libras is shown in Table 19 and the average runtime expended by them when the number of examples increases on the dataset Fars is shown in Table 20. Fig. 10 and Fig. 11 show the relationship between the runtime and the number of attributes and examples, respectively. Moreover, the average runtime expended by the analyzed algorithms on all the datasets is shown in Table 21. Because of the scalability problems presented by Apriori and Eclat, these tables show only the runtime expended on the datasets where they could be run and the figures do not show these results due to the fact that the values exceeds more than 94000 seconds.

Analyzing the results shown in these tables and figures, we can see that most of the methods increase their runtime almost linearly in proportion to the increase in the number of attributes and examples, excepting Apriori, Eclat and GAR which increase exponentially. It should be noted that GAR obtains runtime values far greater than the rest of algorithms as it requires an additional process to mine the positive QARs. Fig. 11 shows few results pertaining to GAR, since in all cases the runtime is higher than 8000 seconds. Furthermore, we can highlight how the increase in the

Table 19: Expended runtime (seconds) by all the algorithms on the dataset Movement Libras with the increase of the number of attributes

	Number of Attributes				
Algorithms	18	36	54	72	91
Clearing	4	5	6	7	10
ASCGA	13	7	7	9	13
EARMGA	6	4	3	3	4
GAR	81	31	46	51	67
GENAR	**2**	**1**	**2**	**2**	**2**
Alatasetal	14	4	4	3	3
Apriori	94200	106482	128324	-	-
Eclat	94200	99940	119427	-	-
QAR-CIP-NSGA-II	3	4	6	7	9
MOPNAR	3	4	6	7	9
NICGAR	10	8	9	19	14

Table 20: Expended runtime (seconds) by all the algorithms on the dataset Fars with the increase of the number of examples

	Number of Examples				
Algorithms	20%	40%	60%	80%	100%
Clearing	224	386	565	958	1178
ASCGA	263	395	462	1240	1663
EARMGA	270	553	624	1078	1438
GAR	6872	17621	23619	36597	45986
GENAR	**64**	**139**	**131**	**174**	**289**
Alatasetal	119	345	634	672	796
QAR-CIP-NSGA-II	254	358	549	749	1163
MOPNAR	215	398	799	833	1262
Apriori	-	-	-	-	-
Eclat	-	-	-	-	-
NICGAR	259	543	834	1097	1657

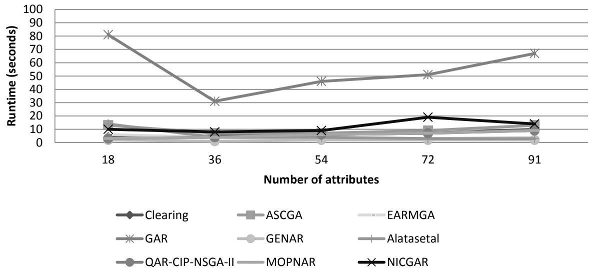


Figure 10: Relationship between the expended runtime by the evolutionary algorithms and the number of attributes on the dataset Movement Libras

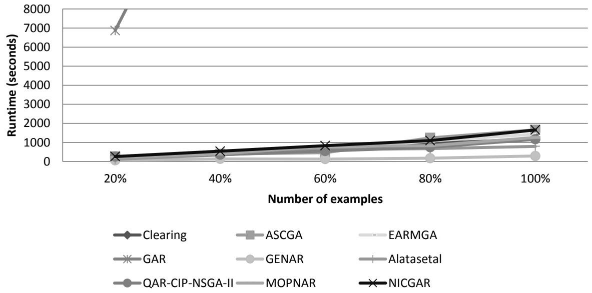


Figure 11: Relationship between the expended runtime by the evolutionary algorithms and the number of examples on the dataset Fars

Table 21: Expended runtime (seconds) on all the datasets

Datasets	Clearing	ASCGA	Alatasetal	EARMGA	GAR	GENAR	QAR-CIP-NSGA-II	MOPNAR	Apriori	Eclat	NICGAR
ba	3	6	4	8	46	1	2	3	28	23	7
bas	1	1	1	1	14	0	0	1	5	4	3
bo	1	1	1	1	8	0	0	0	5	4	2
co	156	355	28	110	3094	48	189	384	-	-	171
hh	132	553	326	208	4858	119	129	226	12652	8065	173
fa	3195	2864	553	1772	30901	160	710	899	-	-	1036
io	5	10	2	5	77	2	3	6	165432	110001	8
le	123	175	70	231	4208	76	195	202	3486	3032	167
ma	100	296	75	199	3446	68	77	129	28360	15072	133
mo	9	20	3	5	71	2	7	15	-	-	15
op	82	210	14	64	1370	18	95	164	-	-	125
pe	72	198	37	142	1407	32	130	108	511	473	114
po	1	2	1	1	14	0	1	1	181	178	3
qu	9	22	9	24	225	6	8	11	0	0	12
sa	54	162	13	63	1498	63	90	114	-	-	78
se	15	34	11	28	315	7	18	26	-	-	24
so	4	9	2	3	44	1	4	7	-	-	8
sp	41	157	13	45	477	56	98	128	-	-	57
spe	4	9	1	2	73	1	9	7	-	-	8
st	4	13	4	9	52	2	4	6	23	20	6
stu	6	13	7	14	305	4	6	7	102	93	10
te	51	107	15	54	684	27	65	103	-	-	64
th	50	169	147	72	1624	37	79	91	-	-	49
ve	6	10	2	5	140	2	10	8	198338	172034	11
wd	7	12	2	6	139	1	5	9	-	-	9
wi	2	3	3	2	25	0	1	2	10	9	4
wo	6	13	9	11	92	1	5	8	34	23	10

number of examples and attributes affects the evolutionary algorithms less than classical association rules extraction algorithms. Finally, notice how it is the number of examples rather than the number of attributes that most affects the runtime expanded by all the algorithms.

5. Conclusion

In this paper, we have proposed NICGAR, a new NGA for mining with low runtime a diverse set of interesting PNQARs. To do so, NICGAR performs a condition selection and an evolutionary learning of the intervals of attributes of the rules considering a punishment mechanism, an EP and restarting process to locate and preserve multiple global solutions in the population. This proposal also includes two threshold values that allow the user to adjust the diversity and quality of the obtained PNQARs. Moreover, we have proposed a new similarity measure between rules based on two of their properties: the examples covered and the common attributes of the rules.

The results obtained over 27 real-world datasets have shown how NICGAR enables us to extract rule sets with a good trade off between the quality and diversity of the knowledge obtained of the whole of the datasets, presenting a high percentage of coverage of the datasets. Moreover, since we usually obtain few attributes in the rule sets derived from the proposed approach, the obtained rules should be simpler to comprehend. It should be noted that the proposed approach expends a reasonable amount of time in all the datasets and presents good scalability when the size of the problem increases.

6. Acknowledgements

This work was supported in part by the Spanish Ministry of Economy and Competitiveness under Grant TIN2014-57251-P, the Andalusian Government under Grant P11-TIC-7765, and the GENIL program of the CEI BioTIC GRANADA under Grant PYR-2014-2.

References

[1] K.-I. Ahn, J.-Y. Kim, Efficient mining of frequent itemsets and a measure of interest for association rule mining, Journal of Information & Knowledge Management 3 (3) (2004) 245–257.

[2] B. Alatas, E. Akin, An efficient genetic algorithm for automated mining of both positive and negative quantitative association rules, Soft Computing 10 (3) (2006) 230–237.

[3] J. Alcala-Fdez, A. Fernandez, J. Luego, J. Derrac, S. Garcia, L. Sanchez, F. Herrera, Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework., Journal of Multiple-Valued Logic and Soft Computing 17 (2-3) (2011) 255–287.

[4] J. Alcala-Fdez, N. Flugy-Pape, A. Bonarini, F. Herrera, Analysis of the effectiveness of the genetic algorithms based on extraction of association rules, Fundamenta Informaticae 98 (1) (2010) 1–14.

[5] J. Alcala-Fdez, L. Sanchez, S. Garcia, M. del Jesus, S. Ventura, J. O. J. Garrell, C. Romero, J. Bacardit, V. Rivas, J. Fernandez, F. Herrera, Keel: A software tool to assess evolutionary algorithms to data mining problems, Soft Computing 13 (3) (2009) 307–318.

[6] F. Berzal, I. Blanco, D. Sanchez, M. Vila, Measuring the accuracy and interest of association rules: A new framework, Intelligent Data Analysis 6 (3) (2002) 221–235.

[7] S. Brin, R. Motwani, J. Ullman, S. Tsur, Dynamic itemset counting and implication rules for market basket data, ACM SIGMOD Record 26 (2) (1997) 255–264.

[8] D. Cavicchio, Adapting search using simulated evolution, Ph.D. thesis, Univ. Michigan, Ann Arbor, MI, USA (1970).

[9] D. E. G. Christopher K. Oei, S.-J. Shang, Tournament selection, niching, and the preservation of diversity, Tech. rep., Univ. of Illinois at Urbana-Champaign, Urbana, IL, USA (1991).

[10] S. Das, S. Maity, B.-Y. Qu, P. N. Suganthan, Real-parameter evolutionary multimodal optimization - a survey of the state-of-the-art, Swarm and Evolutionary Computation 1 (2) (2011) 71–88.

[11] K. A. De Jong, An analysis of the behavior of a class of genetic adaptive systems., Ph.D. thesis, Univ. Michigan, Ann Arbor, MI, USA (1975).

[12] Q. Ding, Q. Ding, W. Perrizo, Parm - An efficient algorithm to mine association rules from spatial data, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 38 (6) (2008) 1513–1524.

[13] A. Eiben, J. Smith, Introduction to Evolutionary Computing, Springer-Verlag, Berlin, Germany, 2003.

[14] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, Journal of the American Statistical Association 32 (200) (1937) 675–701.

[15] S. Garcia, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms behaviour: A case study on the cec2005 special session on real parameteroptimization, Journal Heuristics 15 (2009) 617–644.

[16] L. Geng, H. J. Hamilton, Interestingness measures for data mining: A survey, ACM Computing Surveys 38 (3) (2006) 1–32.

[17] D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley/ Longman, USA, 1989.

[18] D. E. Goldberg, J. Richardson, Genetic algorithms with sharing for multimodal function optimization, in: Proc. 2dn International Conference Genetic Algorithms on Genetic Algorithms and their Application, Nueva York, USA, 1987, pp. 41–49.

[19] J. Han, M. Kamber, Data Mining: Concepts and Techniques, 2nd ed., Morgan Kaufmann, USA, 2006.

[20] S. Holm, A simple sequentially rejective multiple test procedure, Scandinavian Journal of Statistics 6 (1979) 65–70.

[21] J. J.Mata, J.Riquelme, An evolutionary algorithm to discover numeric association rules, in: ACM Symposium on Applied Computing, Madrid, Spain, 2002.

[22] J. Kim, D. Cho, H. Jung, C. Lee, Niching genetic algorithm adopting restricted competition selection combined with pattern search method, IEEE Transactions on Magnetic 38 (2) (2002) 1001 – 1004.

[23] C. Lee, D. Cho, H. Jung, Niching genetic algorithm with restricted competition selection for multimodal function optimization, IEEE Transactions on Magnetic 35 (3) (1999) 1722 – 1725.

[24] J. P. Li, M. E. Balazs, G. T. Parks, P. J. Clarkson, A species conserving genetic algorithm for multimodal function optimization., Evolutionary Computation 10 (3) (2002) 207–234.

[25] J.-P. Li, A. Wood, An adaptive species conservation genetic algorithm for multimodal optimization, International Journal for Numerical Methods in Engineering 79 (13) (2009) 1633–1661.

[26] C.-Y. Lin, W.-H. Wu, Niche identification techniques in multimodal genetic search with sharing scheme, Advances in Engineering Software 33 (11–12) (2002) 779 – 791.

[27] H. Liu, F. Hussain, C. Tan, M. Dash, Discretization: An enabling technique, Data Mining and Knowledge Discovery 6 (4) (2002) 393–423.

[28] M. Lozano, F. Herrera, N. Krasnogor, D. Molina, Real-coded memetic algorithms with crossover hill-climbing, Evolutionary Computation 12 (3) (2004) 273–302.

[29] J. M. Luna, J. R. Romero, S. Ventura, Grammar-based multi-objective algorithms for mining association rules, Data Knowledge and Engineering 86 (2013) 19–37.

[30] S. W. Mahfoud, Crowding and preselection revisited, in: 2nd International Conference on Parallel Problem Solving from Nature (PPSN II), North-Holland, 1992, pp. 27–36.

[31] D. Martin, A. Rosete, J. Alcala-Fdez, F. Herrera, A new multiobjective evolutionary algorithm for mining a reduced set of interesting positive and negative quantitative association rules, IEEE Transactions Evolutionary Computation 18 (1) (2014) 54–69.

[32] D. Martin, A. Rosete, J. Alcala-Fdez, F. Herrera, QAR-CIP-NSGA-II: A new multi-objective evolutionary algorithm to mine quantitative association rules, Information Science 258 (2014) 1–28.

[33] J. Mata, J. Alvarez, J. Riquelme, Mining numeric association rules with genetic algorithms, in: 5th International Conference on Artificial Neural Networks and Genetic Algorithms, Taipei, Taiwan, 2001.

[34] E. Pérez, M. Posada, F. Herrera, Analysis of new niching genetic algorithms for finding multiple solutions in the job shop scheduling, Journal of Intelligent Manufacturing 23 (3) (2012) 341–356.

[35] A. Petrowski, A clearing procedure as a niching method for genetic algorithms., in: International Conference on Evolutionary Computation, Nagoya, Japan, 1996, pp. 798–803.

[36] A. Petrowski, A new selection operator dedicated to speciation., in: 7th International Conference on Genetic Algorithms (ICGA), Michigan, USA, 1997, pp. 144–151.

[37] S. Ramaswamy, S. Mahajan, A. Silberschatz, On the discovery of interesting patterns in association rules, in: 24rd International Conference on Very Large Data Bases, California, USA, 1998, pp. 368–379.

[38] B. Sareni, L. Krähenbühl, Fitness Sharing and Niching Methods Revisited, IEEE Transactions Evolutionary Computation 2 (3) (1998) 97–106.

[39] J. Shaffer, Modified sequentially rejective multiple test procedures, Journal of the American Statistical Association 81 (395) (1986) 826–831.

[40] D. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures, Chapman & Hall/CRC, London, U.K., 2003.

[41] E. Shortliffe, B. Buchanan, A model of inexact reasoning in medicine, Mathematical Biosciences 23 (1975) 351–379.

[42] R. Srikant, R. Agrawal, Mining quantitative association rules in large relational tables, in: ACM SIGMOD International Conference on Management of data (SIGMOD96), 1996, pp. 1–12.

[43] P. Tan, V. Kumar, J. Srivastava, Selecting the right interestingness measure for association patterns, in: 8th International Conference on Knowledge Discovery and Data Mining (KDD 2002), Edmonton, Canada, 2002, pp. 32–41.

[44] D. Taniar, W. Rahayu, O. Daly, Mining hierarchical negative association rules, International Journal of Computational Intelligence Systems 5 (3) (2012) 434–451.

[45] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics 1 (6) (1945) 80–83.

[46] X. Yan, C. Zhang, S. Zhang, Genetic algorithm-based strategy for identifying association rules without specifying actual minimum support, Expert Systems with Applications 36 (2) (2009) 3066–3076.

[47] X. Yin, N. Germay, A fast genetic algorithm with sharing scheme using cluster analysis methods in multi-modal function optimization, in: Proceedings of the International Conference on Artificial Neural Nets and Genetic Algorithms, Innsbruck, Austria, 1993, pp. 450–457.

[48] M. Zaki, Scalable algorithms for association mining, IEEE Transactions on Knowledge and Data Engineering 12 (3) (2000) 372–390.

[49] C. Zhang, S. Zhang, Association Rule Mining: Models and Algorithms Series, Lecture Notes in Computer Science, LNAI 2307, Springer-Verlag, Berlin, 2002.