Enhancing the scalability of a genetic algorithm to discover quantitative association rules in large-scale datasets

María Martínez-Ballesteros, Jaume Bacardit, Alicia Troncoso and José C. Riquelme

Abstract. Association rule mining is a well-known methodology to discover significant and apparently hidden relations among attributes in a subspace of instances from datasets. Genetic algorithms have been extensively used to find interesting association rules. However, the rule-matching task of such techniques usually requires high computational and memory requirements. The use of efficient computational techniques has become a task of the utmost importance due to the high volume of generated data nowadays. Hence, this paper aims at improving the scalability of quantitative association rule mining techniques based on genetic algorithms to handle large-scale datasets without quality loss in the results obtained. For this purpose, a new representation of the individuals, new genetic operators and a windowing-based learning scheme are proposed to achieve successfully such challenging task. Specifically, the proposed techniques are integrated into the multi-objective evolutionary algorithm named QARGA-M to assess their performances. Both the standard version and the enhanced one of QARGA-M have been tested in several datasets that present different number of attributes and instances. Furthermore, the proposed methodologies have been integrated into other existing techniques based in genetic algorithms to discover quantitative association rules. The comparative analysis performed shows significant improvements of QARGA-M and other existing genetic algorithms in terms of computational costs without losing quality in the results when the proposed techniques are applied.

Keywords: Data mining, genetic algorithms, multi-objective optimization, quantitative association rules, large scale datasets

1. Introduction

The use of computational processing techniques for massive data analysis and management has revolutionized the scientific research in the last years. Namely, the knowledge extraction process is becoming more difficult and complex, often causing scalability problems, due to the high volume of data that can be stored nowadays. Thus, the use of efficient computationally techniques is a task of the utmost importance.

In the field of data mining, the learning of association rules (AR) – and particularly of quantitative association rules (QAR) in this work – is a popular and well-known methodology to discover significant and apparently hidden relations among variables in large datasets [7,50]. The AR extraction process consists in using a non-supervised strategy to explore the data properties instead of predicting the class of new data. AR are widely used in many application areas such as intrusion detection, Web usage mining, the healthcare environment to identify risk factor in the onset or diseases [23]. Formally, AR were first defined by Agrawal et al. in [6]. In this context, QAR is a relationship established between continuous attributes using interval

of membership values for each attribute involved in the rule [48].

Note that it is important measuring the quality of the rules in order to evaluate the results obtained by any algorithm and select the best rules. There are several probability-based measures proposed in the literature to evaluate the generality and reliability of AR obtained in the mining process. A detailed interpretation of these measures is provided in [33,49].

Different strategies can be found in the literature to find AR. Mainly, there exist some approaches based on classical methods such as Apriori [7] and a wide range of methodologies based on soft computing techniques [1,60] such as swarm intelligence [51,59] or evolutionary computation [48].

Evolutionary algorithms (EA), and particularly genetic algorithms (GA), have been extensively used for the optimization and adjustment of models in data mining tasks. They are global search algorithms that have been used successfully in many complex and difficult optimization problems due to their flexibility and robust behavior [23].

In the era of Big Data, it's necessary to process huge amount of data within a reasonable time. Hence, it is of utmost importance to improve the scalability of existing EA-based methodologies to discover rules in continuous domains for large-scale datasets.

The development of techniques to reduce the computational costs has been a relevant research topic for many years [3,55,57]. In the context of GA, several mechanisms such as parallel processing [2,4,5,37,54], precomputing the possible matches of the individuals [34] and incremental learning [9] have been addressed. Nevertheless, most of the existing techniques are focused on supervised learning problems such as classification tasks [10], which are well known as genetic based machine learning (GBML) techniques [29]. In contrast, most of the EA-based methods devoted to unsupervised learning problems, include parallel mechanisms that require special hardware to handle large scale datasets [18].

From the best of our knowledge, the development of methods to improve the scalability of AR mining by non-parallel EA has been poorly studied. Moreover, most of these methods are devoted to find AR in categorical or discrete data [6,7], although the domain of most of the real-world data applications is continuous. On the other hand, many existing algorithms to discover AR and QAR are unable to handle large-scale datasets.

Therefore, the main motivation of this work is to propose a preliminary solution to the growing demand for large-scale data mining and applications avoiding the use of special or parallel hardware. In particular, our aim is focused on QAR mining successfully in large-scale datasets, reducing the computational costs and memory resources and without quality loss in the results.

To fulfill this goal, general-purpose mechanisms are proposed in order to achieve run-time reductions in QAR mining according to the number of instances and the dimensionality of variables of the datasets. Namely, the proposed mechanisms are based on a new representation of the individuals, new genetic operators and a windowing-based learning scheme to discover high quality QAR in large datasets. A multi-objective EA to discover QAR, called QARGA-M [50], is used to assess the performance of the techniques proposed here.

The remainder of the paper is as follows. Section 2 presents an overview of several methods focused on large-scale dataset mining. In Section 3, the general-purpose mechanisms developed to enhance the efficiency of EA-based methods to extract QAR in large-scale datasets are presented. In addition, the main features of QARGA-M before applying the techniques proposed in this work are summarized. Section 4 provides a description of the datasets. It also includes the setup of the parameters involved in the process. Furthermore, the results obtained by QARGA-M and other existing EA-based techniques after including the proposed mechanism are compared and discussed to those obtained of the standard version. Finally, Section 5 summarizes the conclusions drawn from the analysis conducted.

2. Related work

This Section provides the most relevant techniques published devoted to improve the scalability of EA, specifically GBML, for large-scale datasets.

EA, specifically GA, have been used in many real-world problems, such as images [52], 3D modeling [17], building structures [30,58], traffic signal coordination [53] or monitoring [16] among others. In the last years, the hybridization with fuzzy logic [56] or neural networks [38] has also been a common strategy in evolutionary computation.

The rule-matching process is the most costly phase in EA-based systems in terms of execution time [28]. The problem is especially emphasized when EA have to handle large-scale datasets. A dataset becomes a large-scale dataset, especially, when the number of in-

Table 1
Type of efficiency enhancement techniques for GA

Category	Subcategory	References
(1) Software Solution	(1.a) Windowing mechanisms	[11,20,24,29,31,62]
	(1.b) Exploiting regularities in the data	[10,34]
	(1.c) Hybrid methods	[36,43,44]
	(1.d) Fitness surrogates	[45,65]
(2) Hardware acceleration techniques	(2.a) Vectorial Instructions	[43]
	(2.b) GPU	[18,19,26,28,35]
(3) Parallelization models		[13,21,25,41]
(4) Data-intensive computing		[40,63]

stances of the dataset is extremely large but the number of variables of the problem should be also considered. For instance, hundreds or even thousands of variables characterize microarray gene expression datasets [14].

EA are good candidates for large-scale data mining mainly due to its inherent principle of evolution. Usually, EA need to process expensive fitness functions a high number of times, and therefore, its parallelization capacity has been widely studied to improve their efficiency for many years [14]. In summary, the techniques applied to improve the adaptability of EA, and particularly GBML, to discover rules in large-scale datasets can be organized in the following not mutually exclusive categories: software solutions, hardware acceleration techniques, parallelization models and data-intensive computing. Table 1 summarizes the four categories, the subcategories in which each category is divided and some references of techniques found in the literature. Note that the aforementioned four categories are based on the taxonomy recently published in [14].

Our interest is focused on the study of methods classified within the software solutions category that includes techniques able to modify data mining methods with the aim at improving their efficiency without including a special or parallel hardware. This category is divided into the following four groups: windowing mechanisms, exploiting regularity in the data, hybrid methods and fitness surrogates.

In particular, the methods in which the fitness function evaluates only a subset of examples from the training set are considered as windowing mechanisms. Alex Freitas [29] proposed a good taxonomy by defining three types of methods according to the strategy for selecting the training subsets, the frequency of changing them, etc. When a static subset of examples is selected before the learning process of GBML methods is referred to as prototype selection [31]. Other methods are focused on changing the subset of the training examples for each generation of the evolutionary process. A windowing method that divides the training dataset into non-overlapping strata preserving the same class distribution, named ILAS, was presented in [11]. Note that other types of techniques such as subgroup discovery [20], frequent patterns mining in data streams [24] or regression models [62] have been also tackled by windowing mechanisms.

The second subcategory within the software solutions category, which is referred to as exploiting regularities in the data, is devoted to reduce computational costs by precomputing some parts of the evaluation process or avoiding computations on irrelevant parts of the data. Specifically, some methods are able to precompute instances by grouping the examples that share the same value for the attributes of the problem and by building an efficient tree structure [34]. Other family of approaches exploits regularities in the data considering that not all attributes of the problem have the same relevance [10]. As a consequence [10] presents a sublinear complexity with respect to the problem dimensionality.

The methods that use smart or directed exploration mechanisms, such as estimation distribution algorithms (EDA), memetic algorithms and messy GA [36], are included in the group of hybrid methods. The compact classifier system proposed in [43] and its extension [44] are an integration of EDA in which the main goal was determined the minimum set of rules that creates a maximally general solution. These methods are mainly focused on tackling large-scale datasets with a huge search space.

Finally, those methods generating a cheap estimation of the fitness function are considered within the fitness surrogates group. The principle of fitness surrogates applied to GBML was presented in [45]. In [65], the authors proposed a multi-age EA that calculates the fitness function of several individuals when reading an instance from the dataset, instead of processing the whole dataset for each individual.

Regarding the hardware acceleration techniques in EA, the use of vectorial instructions to perform the

match operations [42] or the computation of fitness function by the recent and popular technology based on graphics processing units (GPU) can be high-lighted in this category [19,28]. Focused on the discovery of AR, two efficient Apriori implementations using GPU were described in [26]. A novel methodology to evaluate AR on GPU with the purpose of reducing the computational time was recently presented in [18]. A method to extract AR from large and dense datasets with a huge amount of attributes using parallel processing of genetic network programming was introduced in [35].

The category of parallelization models comprises classic methods that execute data mining experiments using multiple computing nodes. There are several examples of GBML methods that implement classic paradigms of parallel GA [13]. Note that most of the existing techniques including mechanisms to enhance the efficiency and scalability of AR mining in large-scale datasets are based in parallelization models. The algorithm Apriori is parallelized in [41] following three different strategies: partitioning the datasets, partitioning both the datasets and the candidate itemsets or replicating the candidate itemsets instead of partition-and-exchanging the dataset transactions. Recently, the authors in [21] presented an extended version of the well-known FP-Growth algorithm combining a parallel algorithm to improve its efficiency and suitability in large-scale datasets. A sequential algorithm for mining AR and four parallel approaches based in this algorithm are proposed in [25].

Finally, the data-intensive computing category includes parallel and distributed scenarios where the number of computing cores is higher than tens of thousands such as the MapReduce data analysis methodology used by Google [63]. Regarding the AR mining, three algorithms based on the MapReduce framework were introduced to analyze several effective implementations of the Apriori algorithm in [40].

In the light of the published literature, it can be concluded that there exist many kind of mechanisms to reduce the run-time costs of EA-based methods for classification problems, whereas the existing algorithms to discover AR have been mainly focused on parallel models. Hence, this paper attempts to provide mechanisms to overcome the flaws related to the high computational costs and memory resources of the existing EA-based methods to discover AR, namely QAR, without using a parallel or special hardware.

3. Proposed methodology

This section details the enhancements performed to improve the scalability of EA-based methods to mine QAR in large-scale datasets. Firstly, a brief description of the QARGA-M algorithm used to assess the performance of the proposed techniques is presented.

3.1. Description of QARGA-M

The authors of this paper have previously published several approaches focused on the discovery of AR and specifically QAR. The first works [48] aimed at providing a real-value coded EA [39] to find QAR in continuous datasets avoiding the discretization step of the variables. Adaptive intervals instead of fixed ranges were used to group samples whose features share certain sets of values in continuous domains.

In order to have a defined notation for QAR, let $A = \{a_1, \ldots, a_n\}$ be a set of features or attributes, with values in. Let S and T be two disjoint subsets of A, that is, S A, T A, and S T $=$ Ø. A QAR is a rule X Y, in which features in S belong to the antecedent X, and features in T belong to the consequent Y, such that X and Y are formed by a conjunction of multiple boolean expressions of the form a_i [l, u], (with l, u).

With the aim at achieving the best trade-off between diversity and convergence of individuals of the population, the authors proposed an improved version [47] based on the well-known Cross-generational elitist selection, Heterogeneous recombination and Cataclysmic mutation (CHC) scheme. Particularly, it comprises an elitist strategy for selecting the population, mechanisms of incest prevention to include a strong diversity and a reinicialization process when the population diversity is poor.

Finally, a multi-objective EA based on the well-known NSGA-II algorithm [22] was presented in [50]. This algorithm, called QARGA-M, extends the main features of previous proposals and improves the QAR mining task by performing the best trade-off among all the measures. QARGA-M evolves the population based on the non-dominated sort of the solutions in fronts of dominance. The first front is composed of the non-dominated solutions of the population; the second one is composed of the solutions dominated by one solution, and so on.

In the population, each individual constitutes a rule. Each rule is represented by a particular codification of the individuals such that it is not necessary to set which variables belong to the antecedent or consequent. Nevertheless, the representation of the individuals includes all the attributes appearing in the dataset, even though only a subset of them is usually expressed in the rule.

Thus, the run-time costs and memory resources are highly increased in large-scale datasets. Note that other existing EA-based methods proposed by other authors to discover QAR also include all the attributes of the dataset in the codification of the individuals [46].

The fitness function is computed for each individual of the population by processing all the instances of the training dataset. As previously stated in Section 2, this phase is the most costly phase of an EA in terms of run-time. Finally, the evolutionary process of QARGA-M ends when the number of generations is achieved.

3.2. Improving the scalability of QARGA-M

This Section depicts an improved representation for continuous attributes in addition to an efficient incremental learning scheme. Namely, a new representation of the individuals has been proposed to avoid irrelevant computations since only a small subset of attributes usually appears in the rules when real world datasets with a large number of attributes are handled. In addition, to reduce the computational cost focused on the number of attributes of the dataset, an incremental learning scheme based on windowing techniques has been implemented. Furthermore, new genetic operators have been designed to deal with the new representation of the individuals. The new representation of the individuals and the new genetic operators are described in Sections 3.2.1 and 3.2.2, respectively. Finally, the incremental windowing-based learning scheme is presented in Section 3.2.3.

3.2.1. New representation of individuals with varying length

The existing methods typically use a rule representation to deal with QAR based on the definition of a hyper-rectangle where an interval, which is delimited by a lower bound and an upper bound, is associated for each dimension. However, all the attributes of the datasets are usually coded in the individuals. Thus, all the attributes of the dataset are processed for each instance of the dataset for each individual in the fitness function computation. This factor highly increases the computational cost and the memory resources when addressing large-scale dataset. The problem can be tackled through a previous feature selection, but interesting information might be lost.

Therefore, we have focused on reducing the size of the individuals and a new representation of the individuals has been proposed in which only the expressed attributes in the rules are coded instead of coding all

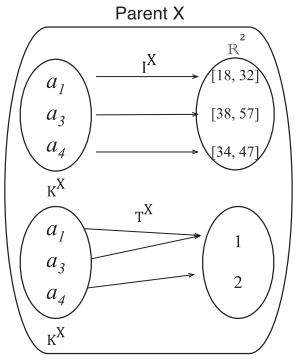

Fig. 1. Example of an individual R representing a rule.

the attributes appearing in the dataset. The individuals are represented by a real coding. The lower and upper bounds of the intervals for each attribute expressed in the rule are represented by an array of variable length less or equal than n, where n is the number of attributes belonging to the dataset. The new coding of an individual, henceforth named R, is defined as a set of identifiers of the attributes expressed in the rule R. In addition, two functions containing the bounds of the intervals and the type of membership for each attribute belonging to the rule are included in the new codification. Note that the membership type describes if an attribute belongs to the antecedent or the consequent of the rule. The main parts of the new representation are described as follows:

- Let R be an individual of the population which represents a rule, let K^R be the subset of attributes of the dataset, $K^R \subset A$, which are expressed in the rule R and let a be an attribute, $a \in K^R$.
- Let I^R be a function, I^R: $K^R \rightarrow \Upsilon^2$, which defines the relation between the attributes in K^R and the bounds of the intervals for such attributes. Thus, $I^R(a) = [l_a^R, u_a^R]$ represents the lower l_a^R and upper u_a^R bounds for the attribute a, which belong to the rule R.
- Let T^R be a function, $T^R : K^R \rightarrow \{1, 2\}$, which defines the relation between the attributes belonging to K^R and the membership type of the attributes. Therefore, $T^R(a)$ represents the membership type of the attribute a in the rule R, that is, if a belongs to the antecedent or the consequent of R. Thus, $T^R(a) = 1$ if a belongs to the antecedent of the rule R or $T^R(a) = 2$ if a belongs to the consequent of the rule.

The evaluation of individuals is more efficient if only the attributes that belong to the rule are repre-

sented, because only these attributes are evaluated instead of processing all attributes of the dataset. An illustrative example of the codification of an individual with the new representation is depicted in Fig. 1. We suppose that the input dataset has 4 attributes $A = \{a_1, a_2, a_3, a_4\}$. In particular, the rule R defined as $a_1 \in [18, 32] \wedge a_3 \in [38, 57] \Rightarrow a_4 \in [34, 47]$ is represented. Note that attributes a_1 and a_3 belong to the antecedent, a_4 to the consequent and a_2 is not involved in the rule. Therefore, $T^R(a_1) = T^R(a_3) = 1$ and $T^R(a_4) = 2$.

3.2.2. New genetic operators

New genetic operators have been defined to handle the new proposed representation. Specifically, crossover and mutation operators are able to modify the bounds of the intervals for the attributes in the rules in addition to edit the set of expressed attributes in the rules. The main features of the genetic operators are described as follows.

3.2.3. Crossover operator

Two parent individuals X and Y, chosen by means of the tournament selection, are used to generate a new individual Z. Note that only the expressed attributes in the rule are represented in an individual. Hence, the attributes belonging to both parents are not necessarily equal. The crossover operator works as follows:

1. If an attribute a is expressed for both parents, that is, $a \in K^X \cap K^Y$, two cases could occur:
   - If $T^X(a) = T^Y(a)$, that is, the membership type of the attribute a is equals in both parents, then the same type is assigned to the offspring and the interval bounds are obtained by generating two random numbers among the interval bounds of both parents.
     The set K^Z and the functions I^Z and T^Z are defined as follows:
     $K^Z = K^Z \cup a$
     $I^Z(a) =$ [*random*(), *random*()]
     $T^Z(a) = T^X(a)$
   - If $T^X(a) \neq T^X(a)$, that is, the membership type of the attribute a is different in both parents, then one of them is randomly chosen without modifying the intervals of the attribute.
     The set K^Z and the functions I^Z and T^Z are defined as follows:
     $K^Z = K^Z \cup a$
     Let $t =$ *random* $(T^X(a), T^Y(a))$, *then*:
     $I^Z(a) = I^X(a)$, if $t = T^X(a)$

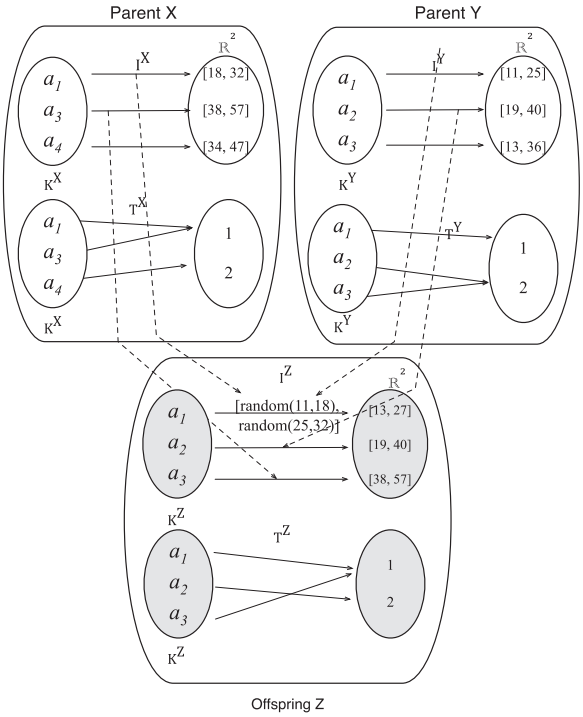

Fig. 2. Crossover for the individuals X and Y.

$I^Z(a) = I^Y(a)$, if $t = T^Y(a)$
$T^X(a) = t$

2. If an attribute a is only expressed in one of the parents, that is, $a \in \{K^X - K^Y\} \cup \{K^Y - K^X\}$, then two cases are possible: a is added or is not added to K^Z.
   - If a is randomly selected to be added in the new offspring Z, the same type and the same intervals of a are assigned to the offspring.
     The set K^Z and the functions I^Z and T^Z are defined as follows:
     $K^Z = K^Z \cup a$
     If $a \in K^X$, then:
     $T^Z(a) = T^X(a)$
   - In other case, a does not belong to the new offspring.

Note that the attributes of the new offspring are sorted in the same order in which they are defined in the dataset. An example of the crossover process is depicted in Fig. 2.

3.2.4. Mutation

The mutation process consists in modifying the genes/attributes of the individuals randomly selected according to a probability and it is applied to the offspring population after the crossover. The mutation op-

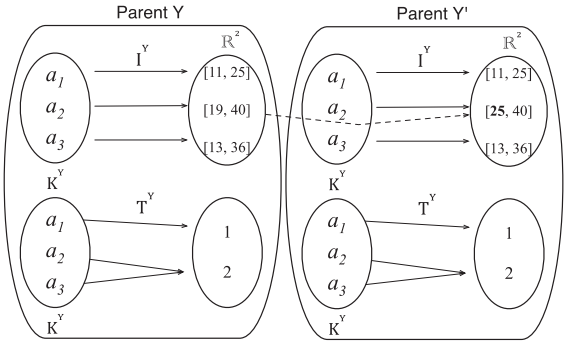

Fig. 3. Lower bound interval mutation.

erator can be focused on the membership type or the intervals of the attributes, which is defined as follows:

1. *Mutation operator focused on the attribute membership type* (T^R). Adding or removing expressed attributes to/from the rule R:
   - *Generalizing mutation.* One expressed attribute a is randomly selected with uniform probability and it is removed from the rule R. An example of this operator is shown in Fig. 4.
   - *Specializing mutation*. One attribute a is not expressed in the rule R and is randomly selected, then it is added to R. Then, the membership type of a is established to belong to the antecedent ($T^R(a) = 1$) or the consequent ($T^R(a) = 2$) of R according to the same probability. The interval bounds of a are randomly generated taking into account the domain of the attribute. An example of this operator is visualized in Fig. 4.
   - *Directional mutation.* One attribute a is expressed in the rule R and is randomly selected. The membership of a is swapped between antecedent and consequent. If the type of a is antecedent ($T^R(a) = 1$), $T^R(a)$ is modified to belong to the consequent ($T^R(a) = 2$). In other case, if the type of a is consequent ($T^R(a) = 2$), $T^R(a)$ is mutated to belong to the antecedent ($T^R(a) = 1$).
2. *Mutation operator focused on the attribute interval bounds* (I^R). Modifying the intervals of the expressed attributes in the rule. Three equiprobable cases are possible for one expressed attribute a randomly selected:
   - Lower bound. A random value is added or subtracted to the lower bound of the interval of the selected attribute a.
   - Upper bound. A random value is added or subtracted to the upper bound of the interval of the selected attribute a.
   - Lower and upper bounds. A random value is added or subtracted to both bounds of the interval of the selected attribute a.

   For all the three cases, the random value is generated between 0 and a percentage (usually 10%) of the amplitude of the interval and it will be added or subtracted according to a certain probability. An example for the lower bound interval mutation is shown in Fig. 3.

The choice between the mutation operators focused on the membership type or the interval bounds, respectively, depends on a given probability. The new offspring is checked to ensure that represent meaningful rules. In particular, the lower bound of the interval has to be less than the upper bound of the interval; the antecedent and the consequent cannot be empty sets and have to be disjoint sets.

3.2.5. Applying the incremental learning with alternating strata windowing scheme

In this section we describe the improvements carried out to reduce the run-time costs of EA-based methods regarding the number of examples of the training dataset. In particular, we propose to modify the learning scheme of QARGA-M to incrementally discover QAR from subsets of the training instances.

The incremental learning performed in QARGA-M is based on the ILAS mechanism due to the similar evolutionary features between QARGA-M and GBML systems. However, different features have been included since ILAS was designed for supervised learning approaches. The windowing-based learning scheme splits the training set into a defined number of non-overlapped subsets or strata of equal size by a methodology similar to stratified n-fold cross validation. In the evolutionary process, each generation or iteration alternatively uses a different stratum to compute the fitness function by a round-robin policy as can be followed in Fig. 5. This mechanism provides more general rules due to the good solutions need to survive in multiple strata [15]. At the beginning of each iteration of the IRL process, the training dataset is randomly reordered and split into a defined number of strata with the same size as can be observed in Fig. 6.

The incremental windowing-based learning scheme implemented within the multi-objective evolutionary process performed by QARGA-M is depicted in Fig. 7. First, the size of each stratum is computed based on

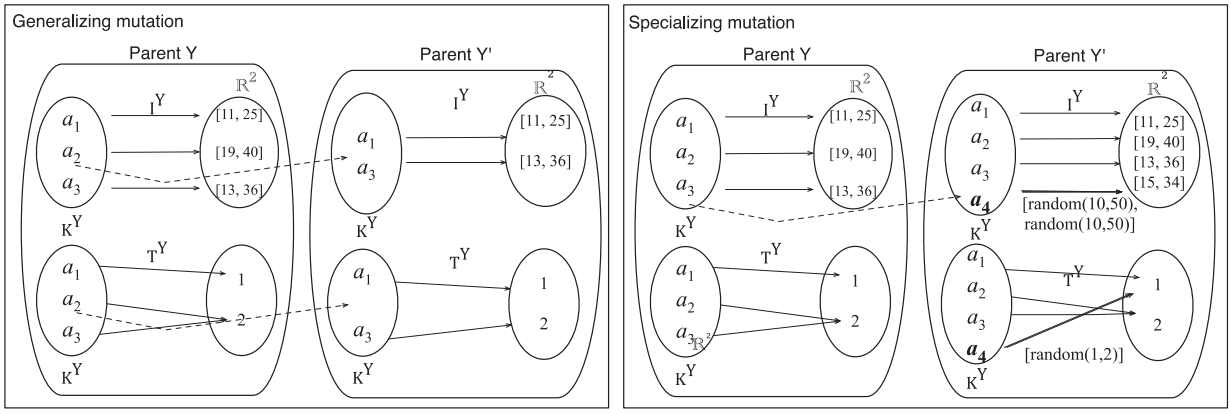

Fig. 4. Generalizing and Specializing mutation.

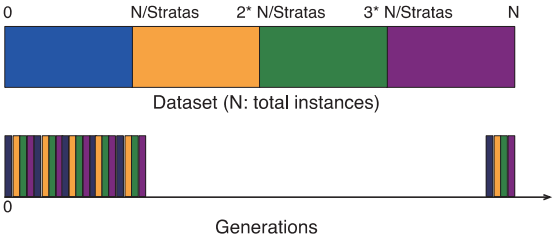

Fig. 5. Incremental windowing-based learning scheme.

the number of instances of the dataset divided by the number of strata considered (line 1). Then, the parent population is initialized focused on the instances previously covered by not many rules to ensure the diversity of the rules (line 2).

In addition, an empty elitist population is created in which the best individual found for each stratum will be stored (line 3). Afterwards, each stratum is sequentially selected in each generation of the multi-objective evolutionary process as follows (line 4.a): the number of current stratum is calculated through the remainder of the division between the number of current generation and the number of the strata in which the dataset is split.

As can be observed, the population is evolved and evaluated using the selected stratum instead of the whole training dataset (except the last generation). In the last generation, the population is evolved and evaluated using all the strata (line 4.b), that is, all the instances of the dataset with the aim at selecting the individual, which best represents the whole dataset. Thereafter, an offspring population of the same size as the parent population is generated after applying the crossover and mutation operators described in Section 3.2 (line 4.c).

The elitism of the population has been redefined to consider the best individual of the generation where the current stratum was used for the last time. A solution A is better than another solution B if A is dominated by a number of individuals lesser than that of B. If both solutions A and B are dominated by the same number of individuals, then A is better than B if the crowding distance of A is greater than that of B. Note that the extreme points of the Pareto-set are not considered to be selected since their crowding distances are always assigned with an infinite value.

Then, the best individual for each generation is found in the least crowded region of the first Pareto front (individual with the highest crowding distance excluding the extreme points) according to the current stratum used.

Hence, an elitist population with the same size as the number of strata and composed of individuals selected as the best rule for each processed stratum. Note that the elitist population is empty at the beginning of each evolutionary process. The new set of individuals to generate the next population is created through the elitist population above described, in addition to the current population and the offspring obtained (line 4.d).

Subsequently, the new set of individuals is evaluated by the measures selected as objectives to be maximized by QARGA-M. In particular, confidence (Eq. (2)), accuracy (Eq. (3)) and leverage (Eq. (4)) measures have been selected following the proposed study in [49]. Thereafter, the fast non-dominated sorting is performed to sort the new individuals in fronts of dominance (line 4.e) [22] and the best individuals are selected to generate the next population (line 4.f). First, the individuals are sorted taking into account the Pareto front in which they belong, and second, by the crowding distance. Note that the size of the next popu-

lation has to be equal to the size of the current population.

Once each generation is completed, the best individual found for the current stratum is compared to the best individual previously obtained for this stratum. As mentioned above, the best individual in the current generation is located in the least crowded region of the first Pareto front excluding the extreme points of the Pareto-set. If the current stratum is processed for the first time (line 4.g), the best individual found in the current generation is added to the elitist population. If the current stratum has been previously processed (line 4.h), the best individual selected is compared to the previous one stored in the elitist population according to the dominance concept [22]. As a consequence of the comparison, the best individual for the current stratum is updated.

The best rule of the whole evolutionary process is selected by following the described method above to select the best individual of the generation (line 5). Finally, the instances covered by the best rule found are penalized to boost the covering of instances still not covered and prevent similar rules (line 6). Only a rule is selected for each iteration of the evolutionary process to perform the IRL process. The whole evolutionary process is repeated until the desired number of rules is reached.

3.3. Quality measures optimized by QARGA-M

Probability-based measures [33] have been selected as objectives to be optimized with the aim of selecting the best rules following the proposed study in [49]. Specifically, leverage, confidence and accuracy measure are selected to be optimized, respectively, to obtain general and reliable rules. In addition, support measure has been considered as a threshold to filter the set of resulting rules. The description and the mathematical definition of these measures are described as follows:

Support($X \Rightarrow Y$): The support of the rule $X \Rightarrow Y$ is the percentage of instances in the dataset that satisfy X and Y simultaneously.

$$Sup(X \Rightarrow Y) = n(X \cap Y)/N \tag{1}$$

where $n(X \cap Y)$ is the number of instances that satisfy the conditions for the antecedent X and consequent Y simultaneously.

Confidence($X \Rightarrow Y$): The confidence is the probability that instances satisfying X, also satisfy Y and

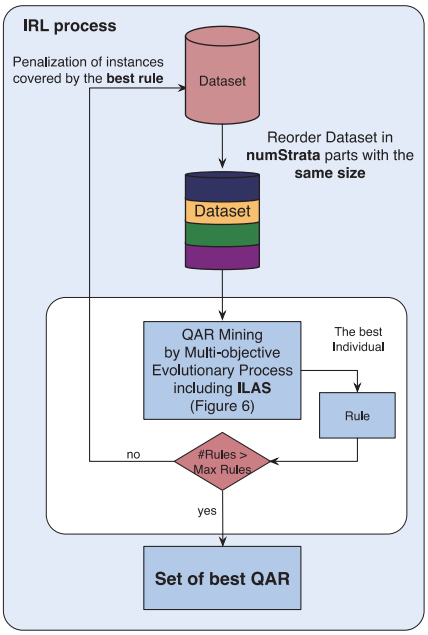

Fig. 6. Scheme of IRL process including the incremental windowing-based learning scheme.

measures the reliability of the rules.

$$Conf(X \Rightarrow Y) = Sup(X \Rightarrow Y)/Sup(X) \tag{2}$$

Accuracy($X \Rightarrow Y$): Accuracy measures the degree of veracity of the rules, {i.e.}, the matching degree between the obtained values and the actual data.

$$Acc(X \Rightarrow Y) = Sup(X \Rightarrow Y) + Sup(\neg X \Rightarrow \neg Y) \tag{3}$$

where $\neg$ means negation, therefore $Sup(\neg X \Rightarrow \neg Y)$ is the percentage of instances in the dataset that do not satisfy X and Y simultaneously.

Leverage($X \Rightarrow Y$): Leverage measures the proportion of additional cases covered by both X and Y above those expected if X and Y were independent of each other.

$$Lev(X \Rightarrow Y) = Sup(X \Rightarrow Y) - (Sup(X)Sup(Y)) \tag{4}$$

4. Experimental results and discussion

Several experiments have been carried out to assess the performance of QARGA-M handling large-scale datasets. The results obtained by the application of QARGA-M with both techniques previously detailed to the datasets specified in Section 4.1 are presented.

Inputs: Maximum number of generations (*MaxNumGen*), number of strata (*NumStrata*), dataset split in *NumStrata* strata (*DataSet*)
Output: Best rule found

QAR Mining by Multi-objective Evolutionary Process including ILAS (*MaxNumGen, NumStrata, DataSet*)

1. **Calculate the size of each stratum (number of instances of *Dataset* divided by *NumStrata*).**
2. Initialize the parent population and evaluate and rank the population by the Fast non dominated Sort [26] using *DataSet.*
3. Initialize the empty elitist population.
4. Repeat while the *MaxNumGen* is not reached. // Generations of an EA loop
   - **(a) If the current generation is not the last one, then:**
     ***TrainingSet* is only composed of the subset of instances of *DataSet* belonging to the *Current Strata*.**
   - **(b) In other case**
     ***TrainingSet* is composed of all the instances contained in *DataSet*.**
   - (c) Generate the offspring population of same size as the parent population by the genetic operators.
   - (d) Generate the new set of individuals by merging the current population, the offspring population and the **elitist population composed of the best individual found for each stratum.**
   - (e) Evaluate and rank the new set of individuals by the Fast non dominated Sort using the ***TrainingSet*.**
   - (f) Select the best individuals to generate the next population.
   - **(g) Select the best individual found for the *Current* Strata. If *Current Strata* is processed for the first time:**
     Add the best individual found for the *Current Strata* in the elitist population.
   - **(h) In other case, if *Current Strata* was previously processed, then:**
     Select from the elitist population the previous individual found for *Current Strata* and replace it by the new best individual if the latter one is better than the former one according to the dominance concept.
5. Return the best individual found, the rule in the first Pareto front that is located in the least crowded region.
6. Penalize the instances covered by the best rule found.

Fig. 7. Multi-objective evolutionary process to discover QAR including ILAS.

The goal of this experimentation is to show that the computational costs of discovering QAR in large-scale datasets could be reduced without quality loss in the obtained rules.

First, Section 4.1 provides a description of all used datasets. A summary of the parameter settings of QARGA-M can be observed in Section 4.2. The results obtained by QARGA-M and other existing approaches are presented and discussed in Section 4.3.

4.1. Datasets description

Four large-scale bioinformatics datasets have been used to assess the performance of the new representation and the incremental windowing-based learning included in QARGA-M. The four datasets belong to the Protein Structure Prediction (PSP) family of problems that can be found at ICOS PSP benchmarks repository [12,61]. This site contains an adjustable real-world family of benchmarks suitable for testing the scalability of machine learning methods. The four datasets are different versions of the same dataset with varying number of attributes, corresponding to different sizes of the neighbourhood around amino acids. The four datasets have 60 attributes (n_1), 100 attributes (n_2), 140 attributes (n_3) and 180 attributes (n_4). All datasets have 234638 instances.

Alternatively, eight different public datasets from the BUFA repository have been also used to evaluate the performance of the proposed mechanisms. Relevant information about these datasets is summarized as follows: Ailerons (AI) dataset has 13750 instances and 41 attributes; Computer Activity (CA) dataset has 8192 instances and 22 attributes; Elevators (EV) dataset has 16599 instances and 19 attributes; Fried (FR) dataset has 40768 instances and 11 attributes; House 16H (HH) dataset has 22784 instances and 17 attributes. Kinematics (KI) dataset has 8192 instances and 9 attributes. 2Dplanes (PN) dataset has 40768 instances and 11 attributes. Pole Telecomm (PT) has 9065 instances and 49 attributes.

4.2. Parameters configuration

The values for the main parameters of QARGA-M are described in this section. It is noteworthy that these values have been used for each test case carried out to assess the performance of QARGA-M. The QARGA-M algorithm including the new features to enhance its scalability in large-scale datasets has been executed ten times for both non-windowed learning and each number of strata used in the windowing-based learning scheme, and each dataset in which QARGA-M has been applied. The main parameters of QARGA-M algorithm are: 100 for the number of the rules, 100 for the size of the population, 100 for the number of generations, 0.1 for the mutation probability of the *generalizing mutation* operator, *specializing mutation* oper-

ator and *directional mutation* operator, and 0.3 for the *interval mutation* operator. The minimum amplitude of each gene in the individual is 2.5% over the full domain of the corresponding attribute.

Following the proposed study in [49], confidence (Eq. (2)), accuracy (Eq. (3)) and leverage (Eq. (4)) measures are the selected measures to be optimized by QARGA-M with the aim at obtaining the rules with high quality, reliability and strong dependence between antecedent and consequent. Minimum thresholds for support and confidence measures have been considered with the aim at reducing the number of rules to be obtained by QARGA-M and comparing the number of rules with good quality to the number strata used in the incremental learning. Specifically, 0.4 is the minimum threshold for confidence measure and 0.0025 is the minimum value for support measure.

It is noteworthy that the values of the main parameter are standard values usually used in other works. The values of the rest of parameters have been experimentally obtained after performing several executions.

4.3. Performance of QARGA-M scalability handling large-scale datasets

This section details the experimentation carried out to assess the scalability of QARGA-M in large-scale datasets. The techniques detailed in Section 3 has been integrated into QARGA-M to reduce run-time costs. Hence, the new representation of the individuals, where only the expressed attributes are coded in the rule, is compared to the previous representation that codes all the attributes appearing in the dataset in Section 4.3.1. Alternatively, the windowing-based learning scheme is compared to the standard non-windowed system in Section 4.3.2. Section 4.3.3 provides a comparative study when the proposed mechanisms are integrated into other existing EA-based techniques.

4.3.1. Quality of the rules obtained by QARGA-M

Both enhancements have been analyzed to study the behavior of QARGA-M in terms of quality of the rules obtained and run-time costs. First, the results of different quality measures are shown for both the old and the new representation of the individuals as well as both the standard non-windowed system and the windowing-based learning scheme. Then, the speedup and the average time per generation measured in second obtained by each representation and each learning scheme are also presented.

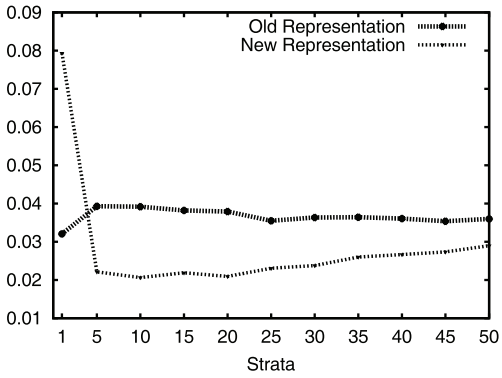

Fig. 8. Average leverage obtained by QARGA-M in all datasets using the old and new representation of the individuals.

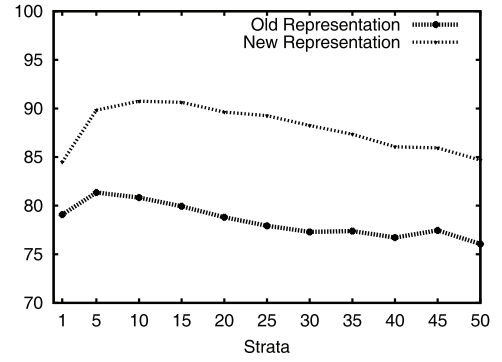

Fig. 9. Average confidence (%) obtained by QARGA-M in all datasets using the old and new representation of the individuals.

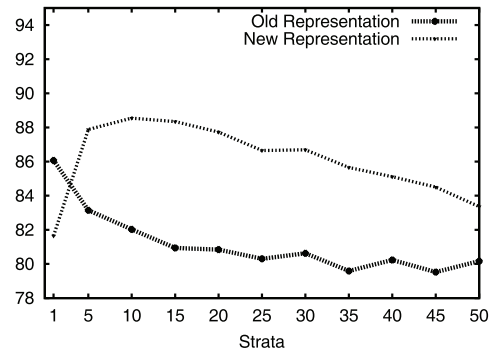

Fig. 10. Average accuracy (%) obtained by QARGA-M in all datasets using the old and new representation of the individuals.

Figures 8–10 summarize the average results for all the datasets considered of the three quality measures optimized by QARGA-M in order to compare the new representation to the old one using the windowing-based learning scheme (varying from 5 to 50 strata with increments of 5 strata) and the standard non-windowed system (1 stratum).

Specifically, each figure shows the average values of leverage, confidence and accuracy, respectively, after applying ten executions. Other quality measures such as the percentage of covered records, gain, certainty factor, leverage, number of rules and number of attributes in the rules have also been studied. Note that only the results obtained for three measures have been displayed since similar results were obtained for the remaining measures.

A non-parametric statistical analysis [32] has been conducted in order to analyze if the improved representation of the individual affects to the quality of the obtained rules or causes information loss in the results. Specifically, the Wilcoxon test has been applied using a level of significance $\alpha = 0.05$ to detect significant differences in the measures of the rules obtained by QARGA-M using the old and new representation of the individuals. Let R^+ be the sum of ranks for the different number of strata for each dataset in which the new representation outperformed the old one, and R^- the sum of the opposite ranks.

The results obtained by the Wilcoxon test have shown a greater value of R^+ in confidence, accuracy, support, percentage of covered instance and number of rules obtained. Both R^+ and R^- values are similar for the lift measure. Only leverage, gain and certainty factor have present a greater value of R^-. However, the resulting *p-values* are greater than the significance level considered for gain, certainty factor and lift measures. The *p-values* obtained by the remaining measures are lower than the significance level. Therefore, the new representation outperforms the old one in confidence, accuracy and support measures in addition to the percentage of covered instances and number of rules obtained. Both representations do not present significant differences in gain, certainty factor and lift measures. The old representation only outperforms the new one in terms of leverage measure.

Alternatively, some differences have been appreciated in the results when the windowing-based learning scheme is applied and the old representation of the individuals is used. The number of quality rules that satisfy the minimum thresholds specified in Section 4.2 decreases when the number of strata is increased. That fact is due to the problem becomes more difficult to be resolved when the number of strata increases. Furthermore, the individuals of the population hardly evolve towards optimal solutions in order to satisfy the maximum number of strata to survive in the following generations. Similar conclusions can be extended to the rest of the measures. The percentage of covered records in the dataset reached by the windowing-based learning scheme and the non-windowed system is 100% for the new representation and greater than 95% for the old representation.

Table 2
Average rankings obtained by Friedman test for each number of strata

Number of strata	Ranking
1 stratum	5.7448
5 strata	4.9063
10 strata	4.9479
15 strata	5.3021
20 strata	6.1406
25 strata	6.1979
30 strata	6.2396
35 strata	6.4063
40 strata	6.6458
45 strata	6.6042
50 strata	6.8646

Nevertheless, an opposite behavior can be observed with respect to the conclusions detailed above when the number of strata considered decreases from 5 strata to only 1, that is, when the standard non-windowed system is applied. It can be noted that the results obtained by the standard non-windowed system are worse compared to the windowing-based learning scheme in many cases. For instance, the support, confidence, leverage, gain and accuracy measures are better when the windowing-based learning scheme is applied. This fact is due to the process of discovering quality and accurate rules becomes more complex when we try to tackle a entire large-scale dataset instead of a subset of instances.

Note that the windowing-based learning scheme not only reduces run-time costs but also overcomes the non-windowed learning in terms of quality rules in many cases. However, slight differences between both learning schemes have been found. Therefore, the application of the windowing-based learning scheme or the standard non-windowed learning should not greatly affect in the quality of the resulting set of rules.

A statistical analysis has been performed to evaluate the significance of the QARGA-M including the windowing-based learning scheme following the non-parametric procedures discussed in [32]. In addition, it could be interesting to find the most suitable number of strata of QARGA-M algorithm to obtain the most optimal results. For this purpose, the support of the rule, confidence, leverage, lift, gain, accuracy, percentage of covered instances and certainty factor measures obtained by QARGA-M using the new representation of the individuals applied to the BUFA and n_1, n_2, n_3 and n_4 datasets respectively, for all the considered number

of strata have been considered. Note that in this statistical analysis, we aim at analyzing the behavior of both learning schemes, whereas the previous statistical analysis has compared the individual representations.

Specifically, Friedman and Iman-Davenport (ID) tests have been applied with level of significance $\alpha = 0.05$ to assess if there are global differences in the measures obtained for all the different number of strata considered. The average ranking obtained by the Friedman test for each number of strata considered in the windowing-based learning scheme of QARGA-M is summarized in Table 2. It can be observed that the lowest value of average ranking is obtained by QARGA-M using 5 strata that can be considered the best number of strata taking to account the measures under study. The worst results are obtained when the number of strata considered is greater than 40.

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 results obtained by the different number of strata. Hence, a post-hoc statistical analysis has been performed. Specifically, Holm test has been applied and it has determined significant differences among the number of strata 5 and 10 with regard to 40, 45 and 50. Hence, it can be concluded that the incremental windowing-based learning has been successfully implemented into QARGA-M and the performance is similar according to the standard non-windowed system.

On the other hand, if the results retrieved from GBML techniques integrating an incremental learning scheme based on windowing techniques such as ILAS are analyzed, interesting conclusions can be achieved. For instance, all the studied problems in [27] lose accuracy with the increase of the number of strata. In contrast, the performance of QARGA-M overcomes the standard windowed system when 5, 10 and 15 strata, respectively, are used. Therefore, it is noteworthy that the robustness of the windowing techniques is higher in the AR context.

4.3.2. Speedup of QARGA-M

The second purpose of this study is to evaluate the run-time differences between the enhanced and the standard techniques integrated in QARGA-M to assess how efficient and faster are both the new representation as the windowing-based learning scheme. All the experiments presented in this section have been carried out using Inter Xeon processors running at 2.00 GHz, Linux operating system and the Java implementation of QARGA-M without any parallelization.

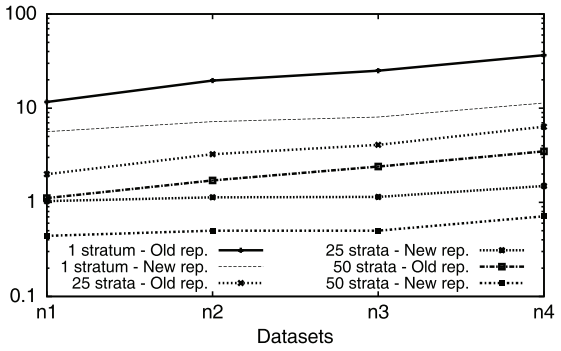

Fig. 11. Average time per generation in seconds obtained by the old and new representations of QARGA-M in PSP datasets.

Figure 11 summarize the average run-time per generation in seconds after applying 10 executions of QARGA-M using the windowing-based learning scheme with 50, 25 and the non-windowed system (1 stratum) respectively, for each representation of the individuals and the n_1, n_2, n_3 and n_4 datasets described in Section 4.1.

We can observe some interesting results. First of all, although computational cost is linearly increased according to the number of attributes of the dataset, the run-time growth of the old representation is higher than the new one.

Regarding the new representation, no significant differences can be observed in the computational cost according to the dataset used. Furthermore, it could be appreciated that the new representation is more efficient because avoids the processing of useless and irrelevant attributes and reduces the number of wasted iterations. For instance, the new representation is up to two times faster than the old one when the non-windowed learning scheme is applied for the dataset n_1 and it becomes four times faster for the dataset n_4.

With respect to the windowing-based learning scheme, the new representation achieves a remarkable performance in contrast to the old representation. Indeed, it is up to five times faster using 50 strata in the windowing-based learning scheme with a high number of attributes in the dataset. Contrary to the strong dependence between the computational cost of the old representation and the number of attributes of the datasets, the run-time of the new representation of the individuals is not affected by the number of attributes of the datasets.

Figure 12 shows the results obtained by QARGA-M when the BUFA datasets are used. Datasets are sorted

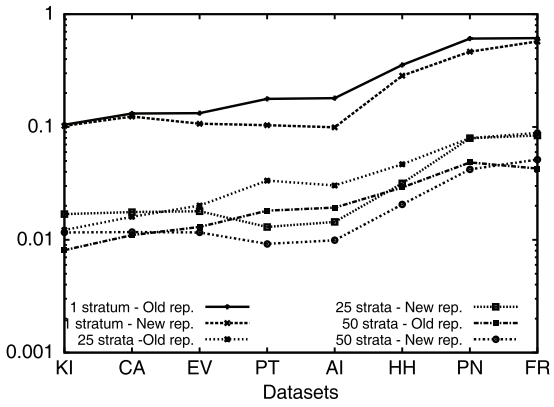

Fig. 12. Average time per generation in seconds obtained by the old and new representations of QARGA-M in BUFA datasets.

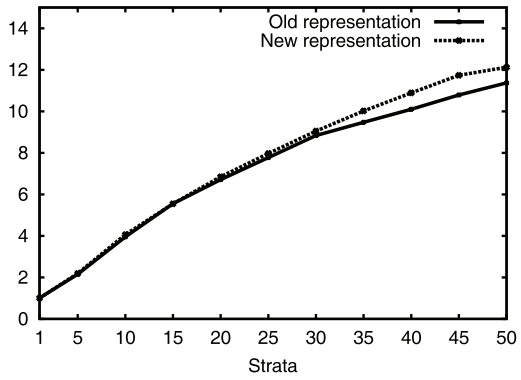

Fig. 13. Speedup of QARGA-M using the old individual representation vs the new individual representation.

by number of instances and attributes. In this case, the datasets have a lower number of instances and computational costs are similar for both representations when the windowing-based learning scheme is applied. When the non-windowed system is considered, the old representation of the individuals gets slower.

In the light of results obtained, it can be concluded that coding only the information of the attributes that are relevant to each rule successfully reduces run-time costs improving the efficiency and performance of QARGA-M algorithm as we expected. The algorithm becomes more efficient when the number of instances and attributes increase.

A standard metric to evaluate the scalability of an algorithm is the speedup factor in order to assess how faster or efficient is the parallel version of an algorithm. In this context, the speedup is a measure of relative performance between the time of the original scheme over time of the windowed scheme using the same number of iterations. To have a better insight in speedup (sp), we compute the speedup of s strata as $sp_s = t_1/t_s$, that is, the run-time obtained by 1 stratum over the time for s strata. In the ideal scenario, the speedup factors will be equivalent to the number of strata used. However, the speedup factor will be below the number of strata for several reasons, such as costs of the algorithm unrelated to the amount of data.

Figure 13 plots the average speedup for all datasets of both individual representations respectively. It can be observed that the speedup achieved by the new representation using the windowing based learning scheme is similar with regard to the speedup factor of the old representation. The speedup factor is greater than 10 when the number of strata is equal or greater than 35 reaching values around 12 when the number of strata is 50. We have observed that the new representation obtain a higher speedup in contrast to the old representation. The windowing scheme based on different number of strata implies a higher generalization of the individuals. This fact involves a less number of conditions and hence, a fewer attributes in the rules. Thus, the individuals are faster to evaluate.

It is noteworthy that the windowing-based learning system in addition to reduce the run-time costs of the QARGA-M algorithm also improves the performance of the system without quality loss in the results in contrast to the standard non-windowed system. Therefore, in the light of the results obtained, we can conclude that the efficiency of QARGA-M handling large-scale datasets has been successfully enhanced by the methodologies proposed in this work.

4.3.3. Performance of other genetics algorithms to discover QAR

A comparative analysis when the proposed methodologies are integrated into other existing methods to discover QAR is described in this section. Specifically, the genetic algorithms QARGA [48], EARMGA [64] and the multi-objective algorithm named NSGAII-CIP-QAR [46] have been used to carry out the comparative study. Both EARMGA and NSGAII-CIP-QAR are available in the KEEL tool [8]. The windowing-based learning scheme has been integrated in QARGA, EARMGA and NSGAII-CIP-QAR algorithms. Furthermore, the new representation of the individuals has been also integrated in QARGA to evaluate the performance of the methodologies proposed over other approaches.

All the algorithms have been executed 10 times for each dataset described in Section 4.1, each number of strata varying from 5 to 50 strata with increments

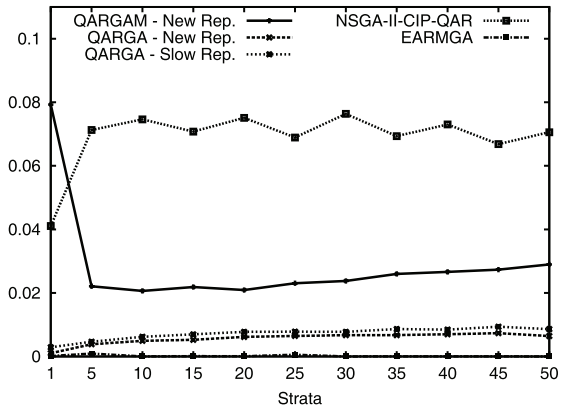

Fig. 14. Average leverage obtained by QARGA-M, QARGA, NSGAII-CIP-QAR and EARMGA algorithms.

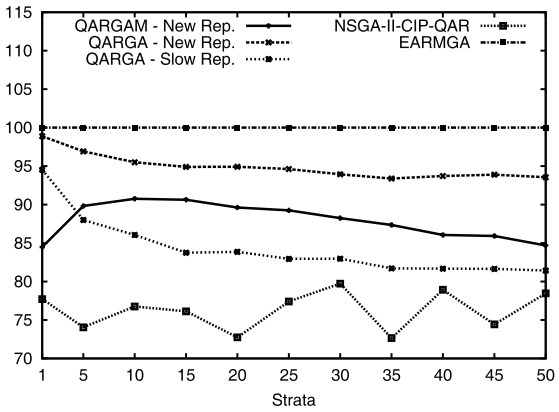

Fig. 15. Average confidence obtained by QARGA-M, QARGA, NSGAII-CIP-QAR and EARMGA algorithms.

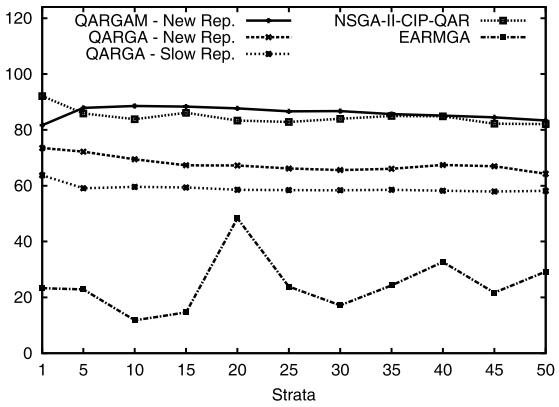

Fig. 16. Average accuracy obtained by QARGA-M, QARGA, NSGAII-CIP-QAR and EARMGA algorithms.

of 5 strata and the standard non-windowed system (1 stratum). The main parameters of QARGA, EARMGA and NSGAII-CIP-QAR algorithms are 100 for the size of the population, 100 for the number of generations and 0.1 for the probability of mutation. As does QARGA-M, QARGA and EARMGA use a minimum support and confidence thresholds of 0.0025 and 0.4, respectively, and 100 for the number of the rules.

The rest of the specific parameter values of QARGA algorithm are 0.5, 0.25, 0.25, 0.25 and 1 for the support, covered instances, amplitude, number of attributes and confidence weights, respectively. On the other hand, parameter values of EARMGA algorithm are 5 for both the number of partitions for numeric attributes and the fixed length of AR. This algorithm uses 0.25 for the probability of selection and 0.7 for the probability of crossover. Finally, parameter values of NSGAII-CIP-QAR are 3 objectives, 2 for the amplitude factor and 5 for the difference threshold. Note that such parameter values for QARGA, EARMGA and NSGAII-CIP-QAR are the default values defined in the KEEL tool.

The average results for all datasets when the windowing-based learning scheme is applied in QARGA, NSGAII-CIP-QAR and EARMGA in addition to QARGA-M including the new representation are summarized in Figs 14–17.

Figure 14 shows the average leverage values obtained by all the algorithms. It can be observed that leverage increases when the number of strata also increases. NSGAII-CIP-QAR algorithm obtains the maximum values when the windowing-based learning scheme is applied whereas QARGA-M achieves the highest value when the standard non-windowed system is considered. Regarding QARGA, similar results are achieved between the new and old representations. EARMGA is the worst algorithm for this measure.

Figure 15 summarizes the average values of confidence measure obtained for each algorithm and each number of strata. In contrast to the previous measure, EARMGA presents the best values for confidence but NSGA-II-CIP-QAR obtains the worst results. Note that EARMGA only optimizes the confidence measure. As QARGA-M does, QARGA achieves the best results when the new representation is considered. In general terms, the confidence values decrease when the number of strata increases. Alternatively, NSGAII-CIP-QAR achieves better values when the number of strata is increased in some cases. Nevertheless, no significant differences are presented among the results obtained by the standard and the windowing-based learning schemes.

Figure 16 visualizes the average values for the accuracy measure. QARGA-M reaches the best accu-

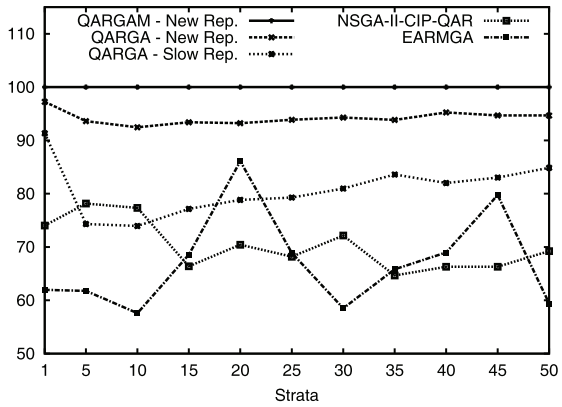

Fig. 17. Covered instances obtained by QARGA-M, QARGA, NSGAII-CIP-QAR and EARMGA algorithms.

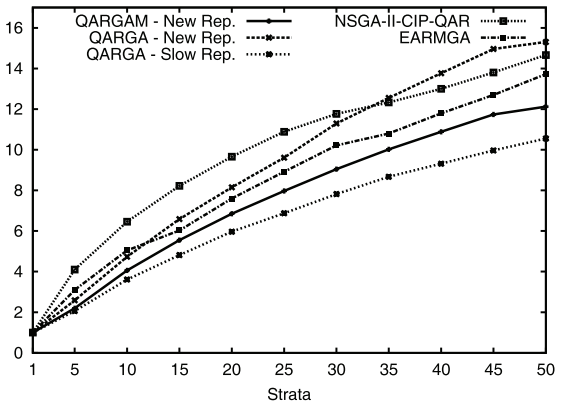

Fig. 18. Speedup obtained by QARGA-M, QARGA, NSGAII–CIP-QAR and EARMGA algorithms.

racy values whereas EARMGA is the worst algorithm. QARGA also obtains better values when the new representation is used.

It can be observed that accuracy values are similar regardless of the number of strata used. In fact, EARMGA achieves better values when the number of strata is above 20.

Finally, Fig. 17 plots the percentage of covered instances for each algorithm and each number of strata. It can be noted that QARGA-M covers the 100% of instances of all datasets whereas the multi-objective NSGAII-CIP-QAR only covers around the 70% of instances of datasets in average terms. QARGA covers more than 90% of instances when the new representation is considered. It can be observed that EARMGA is the most irregular algorithm.

The remaining measures present similar performance when the windowing-based learning scheme is integrated and the number of strata is increased.

Other conclusions can be drawn from the results obtained. For instance, QARGA-M finds more than 70 rules that satisfy the minimum threshold, QARGA achieves between 10 and 30 rules, NSGAII-CIP-QAR discovers more than 50 rules and ERMGA obtains more than 90. Nevertheless, the average support of the consequent of EARMGA achieves values close to 100% that means that the consequent of the rules covers almost all the instances of the datasets. Furthermore, this algorithm achieves the worst values in terms of gain (close to 0), certainty factor and lift due to the high support of the consequent and the weakly dependence among the support of the antecedent and consequent of the rules.

The rest of the algorithms present a stronger dependency between the antecedent and consequent, although EARMGA is better in terms of confidence. Note that this fact should not be considered particularly relevant since the confidence has some drawbacks as discussed previous works [49]. For instance, confidence is not able to find negative dependencies between the antecedent and the consequent.

Regarding the average support, QARGA-M also overcomes QARGA, NSGAII-CIP-QAR and EARMGA. NSGAII-CIP-QAR achieves the better values in terms of lift measure. The results obtained by QARGA-M, QARGA and NSGAII-CIP-QAR for the gain and certainty factor measures are similar.

Figure 18 represents the average speedup obtained by all the algorithms. It can be appreciated that all of them are more than 10 times faster when the number of strata is 50. Therefore, the main goal of this work has been achieved since QARGA-M and the rest of the studied algorithms are able to obtain rules successfully in large-scale datasets when the windowing-based learning scheme is applied without significant loss of quality in the results. Furthermore, the quality of the results is improved when the new representation is considered as shown QARGA and QARGA-M.

5. Conclusions

This paper presents a methodology to improve the efficiency of AR mining techniques based on GA in large-scale datasets. The main drawbacks of these techniques are the high computational costs and memory resources required in the rule-matching process. Each condition belonging to each AR needs to be evaluated for each instance of the training dataset. The improvement performed in this paper aim at reducing run-

time costs during the evaluation process of EA-based techniques to discover QAR. The first improvement has been devoted to avoid irrelevant computation effort coding only the expressed attributes in the rules.

The second one has been focused to integrate a learning scheme to evaluate only a subset of instances of the training dataset in the rule-matching process. Specifically, the inner working of the QARGA-M algorithm has been modified with a new individual representation and windowing-based learning scheme to enhance its scalability in large-scale datasets. The new QARGA-M has been tested in several datasets and compared with the standard QARGA-M in terms of AR quality measures and speedup. Furthermore, a non-parametric statistical test has been applied to detect significant differences between both approaches. The results obtained have shown similar performance between both individual representations of QARGA-M. The windowing-based learning scheme integrated into QARGA-M does not present significant differences in terms of the number of strata used. Regarding the run-time costs focused on the individual representation, the new one does not obtain significant differences according to the dataset used contrary to the old representation.

Furthermore, other existing GA techniques to discover QAR have been also modified to integrate the proposed methodologies showing a remarkable performance in terms of quality measures and speedup. Hence, the methods proposed successfully improve the performance of GA-based methods in terms of efficiency, run-time costs and quality of the rules obtained. As future work, the authors want to study other modern schemes such as NSGA-III, Differential Evolution or Memetic Algorithms to discover efficiently high-quality QAR.

Acknowledgments

The financial support from the Spanish Ministry of Science and Technology, Junta de Andalucía and University Pablo de Olavide under projects TIN2011-28956-C02-02, TIC-7528, P12-TIC-1728 and APPB813097 is acknowledged.

References

[1] H. Adeli and S.L. Hung, Machine learning – neural networks, genetic algorithms, and fuzzy sets, John Wiley and Sons, New York, 1995.

[2] H. Adeli and S. Kumar, Concurrent structural optimization on a massively parallel supercomputer, *Journal of Structural Engineering, ASCE* **121**(11) (1995), 1588–1597.

[3] H. Adeli and S. Kumar, Distributed computer-aided engineering for analysis, design, and visualization, CRC Press, Boca Raton, Florida, 1999.

[4] H. Adeli and S. Kumar, Distributed genetic algorithms for structural optimization, *Journal of Aerospace Engineering* **8**(3) (1995), 156–163.

[5] H. Adeli and K. Sarma, Cost optimization of structures – fuzzy logic, genetic algorithms, and parallel computing, John Wiley and Sons, West Sussex, United Kingdom, 2006.

[6] R. Agrawal, T. Imielinski and A. Swami, Mining association rules between sets of items in large databases, in: *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data* (1993), 207–216.

[7] R. Agrawal and R. Srikant, Fast algorithms for mining associationtion rules in large databases, in: *Proceedings of the International Conference on Very Large Databases* (1994), 478–499.

[8] J. Alcala-Fdez, L. Sánchez, S. García, M.J. del Jesus, S. Ventura, J.M. Garrell, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, J.C. Fernández and F. Herrera, Keel: A software tool to assess evolutionary algorithms for data mining problems, *Soft Computing* **13**(3) (2009), 307–318 (http://sci2s.ugr.es/keel).

[9] J. Bacardit, Pittsburgh genetics-based machine learning in the data mining era: Representations, generalization, and run-time, PhD thesis, Ramon Llull University, Barcelona, Spain, 2004.

[10] J. Bacardit, E.K. Burke and N. Krasnogor, Improving the scalability of rule-based evolutionary learning, *Memetic Computing* **1**(1) (2009), 55–67.

[11] J. Bacardit, D. Goldberg, M. Butz, X. Llorà and J. Garrell, Speeding-up pittsburgh learning classifier systems: Modeling time and accuracy, in: *Parallel Problem Solving from Nature – PPSN VIII volume 3242 of Lecture Notes in Computer Science* Springer Berlin/Heidelberg, (2004), 1021–1031.

[12] J. Bacardit and N. Krasnogor, The infobiotics psp benchmarks repository, 2008. (http://ico2s.org/datasets/psp_ benchmark.html)

[13] J. Bacardit and N. Krasnogor, Performance and efficiency of memetic pittsburgh learning classifier systems, *Evol Comput* **17**(3) (September 2009), 307–342.

[14] J. Bacardit and X. Llora, Large-scale data mining using genetics-based machine learning, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **3**(1) (2013), 37–61.

[15] J. Bacardit, M. Stout, J.D. Hirst, A. Valencia, R.E. Smith and N. Krasnogor, Automated alphabet reduction for protein datasets, *BMC Bioinformatics* **10**(1) (2009), 6.

[16] P. Baraldi, R. Canesi, E. Zio, R. Seraoui and R. Chevalier, Genetic algorithm-based wrapper approach for grouping condition monitoring signals of nuclear power plant components, *Integrated Computer-Aided Engineering* **18**(3) (2011), 221–234.

[17] B.R. Campomanes-Álvareza, O. Cordón and S. Damasa, Evolutionary multi-objective optimization for mesh simplification of 3D open models, *Integrated Computer-Aided Engineering* **20**(4) (2013), 375–390.

[18] A. Cano, J.M. Luna and S. Ventura, High performance evaluation of evolutionary-mined association rules on gpus, *The Journal of Supercomputing* (2013), 1–24.

[19] A. Cano, A. Zafra and S. Ventura, Speeding up the evaluation

phase of gp classification algorithms on gpus, *Soft Computing* **16**(2) (February 2012), 187–202.

[20] J.R. Cano, S. García and F. Herrera, Subgroup discover in large size data sets preprocessed using stratified instance se-lection for increasing the presence of minority classes, *Pattern Recognition Letters* **29**(16) (2008), 2156–2164.

[21] K. Chen, L. Zhang, S. Li and W. Ke, Research on association rules parallel algorithm based on fp-growth, *Communications in Computer and Information Science* **244** CCIS (PART 2) (2011), 249–256.

[22] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii evolutionary computation, *IEEE Transactions on* **6**(2) (2002), 182–197.

[23] M.J. del Jesús, J.A. Gómez, P. González and J.M. Puerta, On the discovery of association rules by means of evolutionary algorithms, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **1** (2011), 397–415.

[24] M. Deypir, M.H. Sadreddini and S. Hashemi, Towards a variable size sliding window model for frequent itemset mining over data streams, *Computers and Industrial Engineering* **63**(1) (2012), 161–172.

[25] S.M. Fakhrahmad and G.H. Dastghaibyfard, An efficient frequent pattern mining method and its parallelization in transactional databases, *Journal of Information Science and Engineering* **27**(2) (2011), 511–525.

[26] W. Fang, M. Lu, X. Xiao, B. He and Q. Luo, Frequent itemset mining on graphics processors, in: *Proceedings of the Fifth International Workshop on Data Management on New Hardware*, DaMoN '09, New York, NY, USA, ACM (2009), 34–42,.

[27] M. Franco, N. Krasnogor and J. Bacardit, Gassist vs. biohel: Critical assessment of two paradigms of genetics-based machine learning, *Soft Computing* **17**(6) (2013), 953–981.

[28] M.A. Franco, N. Krasnogor and J. Bacardit, Speeding up the evaluation of evolutionary learning systems using gpgpus, in: *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, GECCO '10*, New York, NY, USA, ACM, (2010), 1039–1046.

[29] A.A. Freitas, Data mining and knowledge discovery with evolutionary algorithms, Springer-Verlag, 2002.

[30] C. Fuggini, E. Chatzi, D. Zangani and T.B. Messervey, Combining genetic algorithm with a meso-scale approach for system identification of a smart polymeric textile, *Computer-Aided Civil and Infrastructure Engineering* **28**(3) (2013), 227–245.

[31] S. García, J. Derrac, J.R. Cano and F. Herrera, Prototype selection for nearest neighbor classification: Taxonomy and empirical study, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34** (2012), 417–435.

[32] S. García, A. Fernández, J. Luengo and F. Herrera, A study of statistical techniques and performance asures for genetics based machine learning: Accuracy and interpretability, *Soft Computing* **13**(10) (2009), 959–977.

[33] L. Geng and H.J. Hamilton, Interestingness measures for data mining: A survey, *ACM Comput Surveys* **38**(3) (2006), 9.

[34] R. Giráldez, J.S. Aguilar-Ruiz and J.C.R. Santos, Knowledge-based fast evaluation for evolutionary learning, *IEEE Transactions on Systems, Man, and Cybernetics, Part C* **35**(2) (2005), 254–261.

[35] E. Gonzales, K. Taboada, K. Shimada, S. Mabu, K. Hirasawa and J. Hu, Class association rule mining for large and dense databases with parallel processing of genetic network programming, (2007), 4615–4622.

[36] F.Y. Hsiao, S.S. Wang, W.C. Wang, C.P. Wen and W.D. Yu, Neuro-fuzzy cost estimation model enhanced by fast messy genetic algorithms for semiconductor hookup construction, *Computer-Aided Civil and Infrastructure Engineering* **27**(10) (2012), 764–781.

[37] S.L. Hung and H. Adeli, A parallel genetic/neural network learning algorithm for MIMD shared memory machines, *IEEE Transactions on Neural Networks* **5**(6) (1994), 900–909.

[38] X. Jiang and H. Adeli, Neuro-genetic algorithm for nonlinear active control of highrise buildings, *International Journal for Numerical Methods in Engineering* **75**(8) (2008), 770–786.

[39] H. Kim and H. Adeli, Discrete cost optimization of composite floors using a floating point genetic algorithm, *Engineering Optimization* **33**(4) (2001), 485–501.

[40] M.Y. Lin, P.Y. Lee and S.C. Hsueh, Apriori-based frequent itemset mining algorithms on mapreduce, in: *Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication*, ICUIMC '12, (2012), 1–8.

[41] Y. Liu, W. Liao, A. Choudhary and J. Li, Parallel data mining algorithms for association rules and clustering, CRC Press, LLC, 2007.

[42] X. Llorà and K. Sastry, Fast rule matching for learning classifier systems via vector instructions, in: *GECCO '06: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, ACM Press, (2006), 1513–1520.

[43] X. Llora and K. Sastry and D.E. Goldberg, The compact classifier system: Scalability analysis and first results, *IEEE Congress on Evolutionary Computation* **1** (2005).

[44] X. Llorà, K. Sastry, C.F. Lima, F.G. Lobo and D.E. Goldberg. Linkage learning, rule representation, and the x-ary extended compact classifier sytem, in: *Learning Classifier Systems*, Revised Selected Papers of IWLCS 2006–2007, LNAI 4998, Springer-Verlag (2008), 189–205.

[45] X. Llorà, K. Sastry, T.L. Yu and D.E. Goldberg, Do not match, inherit: fitness surrogates for genetics-based machine learning techniques, in: *GECCO '07: Proceedings of the* 9^{th} *Annual Conference on Genetic and Evolutionary Computation*, ACM (2007), 1798–1805.

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

[47] M. Martínez-Ballesteros, F. Martínez-Álvarez, A. Troncoso and J.C. Riquelme, Mining quantitative association rules based on evoluationary computation and its application to atmospheric pollution, *Integrated Computer-Aided Engineering* **17** (2010), 227–242.

[48] M. Martínez-Ballesteros, F. Martínez-Álvarez, A. Troncoso and J.C. Riquelme, An evolutionary algorithm to discover quantitative association rules in multidimensional time series, *Soft Computing* **15**(10) (2011), 2065–2084.

[49] M. Martínez-Ballesteros, F. Martínez-Álvarez, A. Troncoso and J.C. Riquelme, Selecting the best measures to discover quantitative association rules, *Neurocomputing* **126** (Feb 2014), 3–14.

[50] M. Martínez-Ballesteros, I. Nepomuceno-Chamorro and J.C. Riquelme, Discovering gene association networks by multi-objective evolutionary quantitative association rules, *Journal of Computer and System Sciences* **80**(1) (2014), 118–136.

[51] J.L. Olmo, J.M. Luna, J.R. Romero and S. Ventura, Mining association rules with single and multi-objective grammar guided ant programming, *Integrated Computer-Aided Engineering* **20**(3) (2013), 217–234.

[52] E.C. Pedrino, V.O. Roda, E.R.R. Kato, J.H. Saito, M.L. Tronco, R.H. Tsunaki, O. Morandin and M.C. Nicoletti, A ge-

netic programming based system for the automatic construction of image filters, *Integrated Computer-Aided Engineering* **20**(3) (2013), 275–287.

[53] R. Putha, L. Quadrifoglio and E. Zechman, Comparing ant colony optimization and genetic algorithm approaches for solving traffic signal coordination under oversaturation conditions, *Computer-Aided Civil and Infrastructure Engineering* **27**(1) (2012), 14–28.

[54] K.C. Sarma and H. Adeli, Bi-level parallel genetic algorithms for optimization of large steel structures, *Computer-Aided Civil and Infrastructure Engineering* **16**(5) (2001), 295–304.

[55] K. Sarma and H. Adeli, Fuzzy discrete multicriteria cost optimization of steel structures, *Journal of Structural Engineering, ASCE* **126**(11) (2000), 1339–1347.

[56] K. Sarma and H. Adeli, Fuzzy genetic algorithm for optimization of steel structures, *Journal of Structural Engineering* **126**(5) (2000), 596–604.

[57] K.C. Sarma and H. Adeli, Life-cycle cost optimization of steel structures, *International Journal for Numerical Methods in Engineering* **55**(12) (2002), 1451–1462.

[58] L. Sgambi, K. Gkoumas and F. Bontempi, Genetic algorithms for the dependability assurance in the design of a long span suspension bridge, *Computer-Aided Civil and Infrastructure Engineering* **27**(9) (2012), 655–675.

[59] Y. Shafahi and M. Bagherian, A customized particle swarm method to solve highway alignment optimization problem, *Computer-Aided Civil and Infrastructure Engineering* **28**(1) (2013), 52–67.

[60] N. Siddique and H. Adeli, Computational intelligence – synergies of fuzzy logic, neural networks and evolutionary computing, Wiley, West Sussex, United Kingdom, 2013.

[61] M. Stout, J. Bacardit, J.D. Hirst and N. Krasnogor, Prediction of recursive convex hull class assignments for protein res dues, *Bioinformatics* **24**(7) (2008), 916–923.

[62] K. Veeramachaneni, O. Derby, D. Sherry and U.M. O'Reilly, Learning regression ensembles with genetic programming at scale, in: *Proceeding of the Fifteenth Annual Conference on Genetic and Evolutionary Computation Conference, GECCO '13*, ACM (2013), 1117–1124.

[63] A. Verma, X. Llora, D.E. Goldberg and R.H. Campbell, Scaling genetic algorithms using mapreduce, in: *Intelligent Systems Design and Applications 2009. ISDA '09. Ninth International Conference on*, (2009), 13–18.

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

[65] L. Zhao and L. Wang, Multiage evolutionary algorithm and its application in data mining, *Information* **15**(1) (2012), 347–362.