Mining quantitative association rules based on evolutionary computation and its application to atmospheric pollution

M. Martínez-Ballesteros[a], A. Troncoso[b,*], F. Martínez-Álvarez[b] and J. C. Riquelme[a] [a]*Department of Computer Science, University of Seville, Seville, Spain*
[b]*Area of Computer Science, Pablo de Olavide University of Seville, Seville, Spain*

Abstract. This research presents the mining of quantitative association rules based on evolutionary computation techniques. First, a real-coded genetic algorithm that extends the well-known binary-coded CHC algorithm has been projected to determine the intervals that define the rules without needing to discretize the attributes. The proposed algorithm is evaluated in synthetic datasets under different levels of noise in order to test its performance and the reported results are then compared to that of a multi-objective differential evolution algorithm, recently published. Furthermore, rules from real-world time series such as temperature, humidity, wind speed and direction of the wind, ozone, nitrogen monoxide and sulfur dioxide have been discovered with the objective of finding all existing relations between atmospheric pollution and climatological conditions.

Keywords: Data mining, evolutionary algorithms, quantitative association rules

1. Introduction

Predicting a chronological sequence of observations on a variable, commonly known as *time series forecasting*, has been traditionally performed by the application of statistical methods [8]. The results obtained from such methods for synthetic data are usually satisfactory. Furthermore, the inherent simplicity shown by statistical-based methods makes their use popular and widespread. However, when dealing with real-world time series the accuracy of the predictions are not as expected since these datasets often present non-linear features that the classical Box-Jenkins approaches are unable to model.

The temporary evolution of most variables is usually influenced by the changes occurring in other time series. In other words, the correlation between different time series is a frequent phenomenon. For instance, when a rainfall forecast is required, the analysis of other variables such as temperature, humidity or atmospheric pressure is mandatory. Consequently, a diligent analysis of the correlated variables may lead to the discovery of how the variable in question may behave in the near future.

The goal of the association rules (AR) extraction process precisely consists of discovering the presence of pair conjunctions (attribute (A) – value (v)) that appear in a dataset with a certain frequency in order to formulate the rules that outline the existing relationship among attributes. Formally, an association rule is a relationship between attributes in a database such that $C_1 \Rightarrow C_2$, where C_1 and C_2 are pair conjunctions such as $A = v$ if $A \in \mathbb{Z}$ or $A \in [v_1, v_2]$ if $A \in \mathbb{R}$. Generally, the antecedent C_1 is formed by a conjunction of multiple pairs and the consequent C_2 is usually a single pair.

The main motivation of this research is to develop a genetic algorithm (GA) capable of finding quantitative association rules in databases with continuous attributes avoiding the discretization as a prior step of the

*Corresponding author: A. Troncoso, Pablo de Olavide University of Seville, Ctra. Utrera, Km.1, 41013, Sevilla, Spain. Tel.: +34 95 4977522; Fax: +34 95 4348377; E-mail: ali@upo.es.

process. Thus, a real-coded genetic algorithm (RCGA) that expands the general scheme of the CHC binary-coded evolutionary algorithm [15] is proposed in this work. The approach provides numeric association rules establishing relationships among all attributes of the datasets.

For evaluating the performance of the RCGA, two different kind of datasets are analyzed. On one hand, its application over synthetic datasets is reported. On the other hand, an attempt to forecast real-world time series is made by means of the extracted quantitative association rules.

With regard to the real-world time series, three environmental agents responsible for pollution are evaluated: ozone (O_3), sulfur dioxide (SO_2) and nitrogen monoxide (NO). The tropospheric ozone is an atmospheric particle typically identified as a pollutant when it overlaps some threshold. The variation in concentration of this agent in the air is continuously studied, as the noxious effects caused in all living beings is well known [30]. Both sulfur dioxide and nitrogen monoxide are usually formed in various industrial processes, and its concentration in the air has dramatically increased during the last decade. Higher concentrations may cause what experts usually call *acid rain*, which causes damage to living beings and infrastructures [17].

The search of AR in ozone time series must not be mistaken with the Subgroup Discovery (SD) issue [13]. The AR are a non-supervised learning tool, while the SD performs supervised learning. Both AR and SD search for rules but SD searches for conditions of a single attribute. Nevertheless, AR can deal with multiple attributes in the antecedent and in the consequent. Moreover, the AR do not preset the range to which the attributes of the consequent can vary.

The rest of the paper is divided as follows: Section 2 describes the state of the art. Section 3 provides the methodology used in this work. The results of the approach applied to synthetic data are discussed in Section 4. Section 5 refers to the results obtained for the atmospheric datasets. Finally, Section 6 discusses the resulting conclusions.

2. State of the art

There are many efficient algorithms that find AR. Genetic algorithms have been used profusely to generate rules in many learning problems [2,9,24]. Also, genetic algorithms are used as a tool in many real-world problems, such as scheduling [14], forecasting [35], design [26] or classification [10]. Finally, hybridization with fuzzy logic [31], neural networks [20] or simulation [11] are common strategies in evolutionary computation.

However, many researchers focus on databases with discrete attributes while most real-world databases essentially contain continuous attributes, as in the case with time series analysis [6]. Moreover, the majority of the tools said to work in the continuous domain just discretize the attributes using a specific strategy and later, handle these attributes as if they were discrete [1, 33].

A review of recently published literature reveals that the amount of works providing metaheuristics and search algorithms relating to AR with continuous attributes is scarce. Thus, the authors of [25] proposed an evolutionary algorithm to discover numeric association rules, dividing the process in two phases. The first one determined the frequent itemsets, that is, the set of features appearing with a certain frequency within a dataset. In the second phase, the rules were extracted from the itemsets previously calculated.

The work presented in [32] studied the conflict between minimum support and confidence problems. They proposed a method to find quantitative AR by clustering the transactions of a database. Afterwards, such groupings were projected into the domains of the attributes in order to create meaningful intervals which could be overlapped.

Hydrological time series were studied in [36]. First, the numeric attributes were transformed into intervals by means of clustering techniques. Then, the AR were generated making use of the well-known Apriori algorithm [1].

A classifier system was presented in [28] with the purpose of extracting quantitative AR over unlabeled (both numerical and categorical) data streams. The main novelty of this approach was the efficiency and adaptability to data gathered on-line.

A metaheuristic optimization based on rough particle swarm techniques was presented in [3]. In this case, the singularity was the obtention of the values that determine the intervals for the AR instead of frequent itemsets. In synthetic data, several new operators such as rounding, repairing and filtering were evaluated and tested.

MODENAR is a multi-objective pareto-based genetic algorithm that was presented in [4]. The fitness function was composed of four different objectives: Support, confidence, comprehensibility of the rule (to be maximized) and the amplitude of the intervals that constitutes the rule (to be minimized).

The work published in [38] exhibited a new approach based on three novel algorithms: value-interval clustering, interval-interval clustering and matrix-interval clustering. Their application was found especially useful when mining complex information.

Another GA was used in [37] in order to obtain numeric AR. However, the unique objective to be optimized in the fitness function was the confidence. To fulfill this goal, the authors avoided specifying the actual minimum support, which is the main contribution to this work.

The use of AR in bioinformatics is also widely spread. Hence, the work in [16] analyzed microarray data using quantitative AR. For this purpose, they chose a variant of the algorithm introduced in [29] based on half-spaces or linear combinations of bounded variables against a constant. Moreover, Gupta et al. mined quantitative AR for protein sequences [19] and for this reason they proposed a new algorithm with four steps to follow. They first equi-depth partitioned the attributes; second, the partitions were mapped on consecutive integers, thus representing the intervals; third, they found the support of all intervals; and, finally, they used the frequent itemsets to generate AR. On the other hand, the authors in [27] proposed a novel temporal association rule mining method based on the Apriori algorithm. Hence, they identified temporary dependencies from gene-related time series.

The AR had been applied in fuzzy sets by various authors. Thus, Kaya and Alhajj first proposed a GA-based framework for mining fuzzy AR in [21]. To be precise, they presented a clustering method for adjusting the centroids of the clusters and then, they provided a different approach based on the well known CURE [18] clustering algorithm to generate membership functions. Later, they introduced a GA to optimize membership functions for fuzzy weighted AR mining in [22]. Their proposal automatically adjusted these sets to provide maximum support and confidence. To fulfill this goal, the base values of the membership functions for each quantitative attribute were refined by maximizing two different evaluation functions: the number of large itemsets and the confidence interval average of the generated rules. Alternatively, Alcalá-Fdez et al. [5] presented a new algorithm for extracting fuzzy AR and membership functions by means of evolutionary learning based on the 2-tuples representation model.

Finally, Ayubi et al. [7] proposed an algorithm that mined general rules whose applicability ranged from discrete attributes to quantitative discretized ones. Thus, they stored general itemsets in a tree structure in order for it to be recursively computed. They equally addressed the association rules in tabular form allowing a set of different operators.

3. Description of the algorithm

In this work a real-coded [23] genetic algorithm (hereafter called RCGA) has been used to obtain AR from quantitative datasets. The proposed RCGA follows the general scheme of the CHC binary-coded evolutionary algorithm proposed by Eshelman in 1991 [15]. The original CHC presents an elitist strategy for selecting the population that will make up the next generation and includes strong diversity in the evolutionary process through mechanisms of incest prevention and a specific operator of crossover called Half Uniform (HUX). Furthermore, the population is reinitialized when its diversity is poor. Details of these main features of the CHC algorithm are outlined in the following points.

- Elitist selection: This kind of strategy guarantees the survival of the best individuals. Thus, the current population and its offspring are joined and the best individuals (according to the fitness function) are chosen to compose the population of the next generation.
- The HUX crossover operator: This operator swaps exactly half of the nonmatching genes of the parents. Therefore, the Hamming distance divided by two is the number of genes to be swaped. This crossover is highly destructive and introduces some diversity in the population preventing premature convergence.
- Incest prevention: In the CHC algorithm the crossover among siblings is forbidden. Therefore, in order to prevent this, the following function is applied: Two individuals are only crossed if their Hamming distance divided by two is greater than a certain threshold which is set to the length of the individual, i.e. the number of bits, divided by four. Consequently, only highly dissimilar parents are crossed. When there are no parents to be crossed due to their Hamming distance divided by two is less than the predetermined threshold, the threshold is decremented by one unit. As such, the key idea is to avoid the application of the crossover operator among similar individuals.

– Reinitialization: When the evolutionary process converges, the individuals are usually similar and if the iterated threshold becomes negative, the population is restarted in order to provide diversity to the population. Generally, the population is reinitialized with the best individual of the population and mutations of the best individual that usually implies flipping 35% of the genes with some probability.

The proposed RCGA approach for discovering AR from datasets with real values extends the CHC algorithm detailed below. However, it adopts a more conservative reinitialization strategy and a less disruptive crossover operator than the HUX crossover scheme. The pseudocode of the CHC algorithm is as follows:

Input: Maximum number of generations ($MaxNumGen$) and threshold for preventing incest ($MinDist$)
Output: Population of the last generation
CHC()

$numGen \leftarrow 0$
Initialize $P(numGen)$
Initialize $MinDist$
while $(numGen <= MaxNumGen)$
  Evaluate $P(numGen)$
  $C(numGen) \leftarrow Crossover(P(numGen))$
  Evaluate $C(numGen)$
  $P(numGen + 1) \leftarrow SelectBest(P(numGen) \cup C(numGen))$
  if$(P(numGen + 1)$ **equals** $P(numGen))$
    $MinDist \leftarrow MinDist - 1$
    if $(MinDist < 0)$
      Initialize $P(numGen + 1)$
      Initialize $MinDist$
    end if
  end if
  $numGen \leftarrow numGen + 1$
end while
return $P(numGen)$

In a continuous domain, it is necessary to group certain sets of values that share the same features and, as a consequence, it becomes necessary to be able to express the membership of the values in each group. Intervals have been chosen to represent the membership of such values in this work.

The search of the most appropriate intervals is carried out by means of the proposed RCGA. Thus, the intervals are adjusted to find the AR with high values for both support and confidence as well as other measures used in order to quantify the quality of the rule.

Within the population, each individual constitutes a rule. These rules are then subjected to an evolutionary process in which both crossover operator with incest prevention and reinitialization of the population are applied and, at the end of the process, the fittest individual is designated as the best rule. Moreover, the fitness function has been provided with a set of parameters so that the user can drive the search process depending on the desired rules. The punishment of the covered instances allows the subsequent rules found by the RCGA to try to cover those instances that were still not covered, by means of Iterative Rule Learning (IRL) [34].

Table 1
Representation of an individual of the population

$i_1\ s_1$	$i_2\ s_2$	...	$i_n\ s_n$
t_1	t_2	...	t_n

The following subsections describe the representation of the individuals, the fitness function, the genetic operators and how the population is restarted.

3.1. Codification of the individuals

Each gene of an individual represents the upper and lower limit of the intervals of each attribute. The individuals are represented by an array of fixed length n, where n is the number of attributes belonging to the database. Furthermore, the elements are real-valued since the values of the attributes are continuous.

Two structures are available for representing an individual, as is shown in Table 1. Note that all attributes included in the database are depicted in the first row. The limits of the intervals of each attribute are stored in this row, where i_j is the inferior limit of the interval and s_j the superior one.

Nevertheless, not all attributes will be present in the rules that describe an individual. A second row indicating the type of each attribute (shown in the second row of Table 1) has been developed to improve the efficiency. Note that t_i can have three different values: 0 when the attribute does not belong to any individual, 1 when the attribute belongs to the antecedent and 2 when it belongs to the consequent. Therefore, if an attribute is retrieved for a specific rule, it can be achieved by modifying the value equal to 0 of the type by a value equal to 1 or 2. Analogously, an attribute that appears in a rule may be removed by changing the type of the attribute from values 1 or 2 to 0.

3.2. Generation of the initial population

The number of attributes is randomly generated for each individual taking into consideration the desired structure for the rules, the maximum and minimum number of allowed antecedents and consequents and the maximum and minimum number of attributes forming an individual.

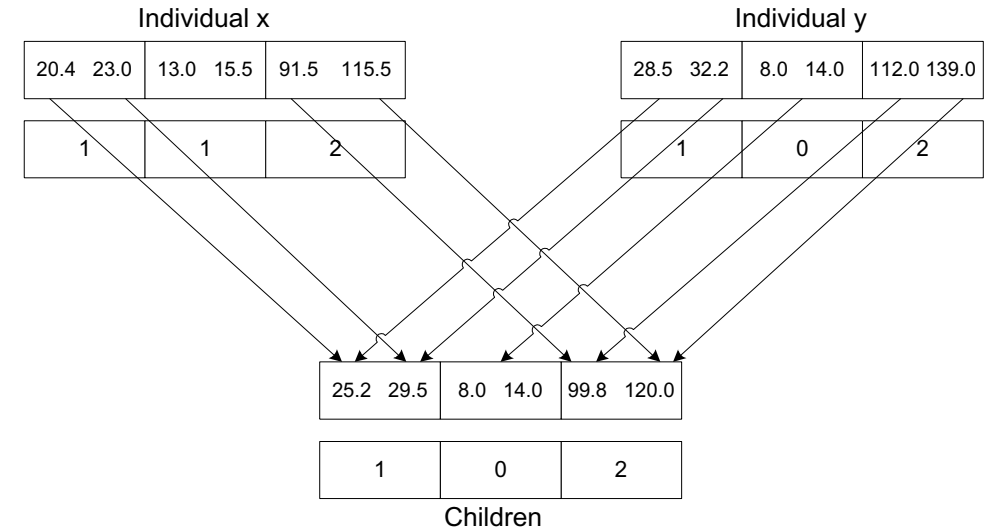

Fig. 1. Crossover and distance for the individuals x and y (attribute ranges: $a_1 \in [15, 35]$, $a_2 \in [5, 20]$ and $a_3 \in [85, 150]$).

It is important to remark that the generation of the interval limits is not arbitrary. On the contrary, it is so that at least one sample of the dataset is covered and that the size of the intervals is less than a given maximum amplitude.

3.3. Crossover operator without incest

Two parent individuals, chosen by means of the roulette selection, are combined to generate a new individual. However, not all parents are crossed, only those parents who differ sufficiently. Thus, the distance between two possible parents is calculated and the parents are only crossed if the distance is greater than $D_{max}/4$ [12] where D_{max} is the maximum possible distance between two individuals and it is defined by:

$$D_{max} = 2 \cdot \sum_{i=1}^{n} (MAX_i - MIN_i) \qquad (1)$$

where MAX_i and MIN_i are the maximum and minimum of the range in which the attribute i varies and n is the number of attributes in the dataset. When there are no individuals or potential parents to be crossed due to a distance less than $D_{max}/4$, this threshold is decremented by a percentage of its initial value [5] (10% in this work).

Therefore, the parents to be crossed have to be at least 25% different in order to prevent incest. The distance between two individuals x and y is defined as follows:

$$\textit{distance}(x, y) = \sum_{j \epsilon S} |i_j^x - i_j^y| + |s_j^x - s_j^y| \qquad (2)$$

where i_j^x, s_j^x, i_j^y and s_j^y are the inferior and superior limits of the interval of the attribute j which is associated to the individual x and y, respectively. S is the set of attributes and the type for both parents, t_j^x and t_j^i, may or may not coincide as one of them is 1 and the other is 0, or viceversa.

When the type of the attribute t_j, is zero for an individual, the attribute does not form part of the rule represented by the individual and the interval considered to calculate the distance is the range in which the attribute varies. This process is depicted in Fig. 1. However, when the same attribute is an antecedent for one individual and consequent for another one, these two individuals are considered different enough to be crossed and it is not necessary to calculate the distance between them.

Once the distance between the parents has been calculated and it is greater than $D_{max}/4$ the parents are crossed as follows. First, all the attributes associated to each parent are analyzed in order to discover their type. Then, if the same attribute in both parents belonged to the same type of attribute, this type of attribute would be assigned to the descendent and the interval would be

obtained generating two random numbers among the limits of the intervals of both parents. Thus, the lower interval would be generated by a random number that belonged to the interval formed by both lower intervals of the parents; the upper interval is analogously calculated. Otherwise, one of the two types would be randomly chosen between both parents, without modifying the intervals of such attribute. Formally, x and y are the two individuals to be crossed and $[l_i^x, s_i^x]$ and $[l_i^y, s_i^y]$ are the intervals in which the attributes vary, respectively. t_i^x is the type of the attribute a_i that belongs to the individual x and finally z is the offspring obtained by the crossover between x and y. Then,

$$[l_i^z, s_i^z] = [random(l_i^x, l_i^y), random(s_i^x, s_i^y)] \quad \text{if } t_i^x = t_i^y \tag{3}$$

If $t_i^x \neq t_i^y$, then t_i^z is randomly selected to be equal to t_i^x or t_i^y and the intervals will be equal to that of any parent:

$$[l_i^z, s_i^z] = [l_i^x, s_i^x], \text{if } t_i^z = t_i^x \tag{4}$$

$$[l_i^z, s_i^z] = [l_i^y, s_i^y], \text{if } t_i^z = t_i^y \tag{5}$$

3.4. Reinitialization of the population

The population is restarted when the threshold is set to a negative value in order to introduce diversity in the population and avoid the well-known premature convergence of genetic algorithms. In this work, the population is reinitialized with 35% of the best individuals of the population and mutations of the best individual.

The mutation consists in varying one gene of the individual. The mutation is focused on the intervals, in which three different cases are possible: equiprobable of the upper limit or of the lower limit or of both limits of the interval. To this regard, a random value between 0 and a percentage (10% usually) of the amplitude in which the attribute varies is generated and is added or subtracted to the limit of the interval randomly selected.

3.5. The fitness function

The fitness of each individual allows for determining which are the best candidates to remain in subsequent generations. In order to make this decision, it is preferable to have high support since this fact implies that more samples from the database are covered. Nevertheless, to only take into consideration the support is not enough to calculate the fitness because the algorithm would attempt to enlarge the amplitude of the intervals until the whole domain of each attribute was completed. For this reason, it is necessary to include a measure to limit the growth of the intervals during the evolutionary process. The chosen fitness function to be maximized is:

$$f(i) = w_s \cdot sup + w_c \cdot conf - w_r \cdot recov + w_n \cdot nAttrib - w_a \cdot ampl \tag{6}$$

where sup is the support, $conf$ is the confidence, $recov$ is the number of recovered instances, $nAttrib$ is the number of attributes appearing in the rule, $ampl$ is the average size of attribute intervals that compose the rule and w_s, w_c, w_r, w_n and w_a are weights in order to drive the search depending on the required rules.

The support prefers the rules with a high value of support, that is, rules fulfilled by many instances and the weight w_s can increase or decrease its effect.

The confidence together with the support are the most widely used measures to evaluate the quality of the AR. The confidence is the reliability grade of the rule. High values of w_c may be used when rules without error are desired, and viceversa.

The number of recovered instances is used to indicate that a sample has already been covered by a previous rule. Thus, rules covering different regions of the search space are preferred. The process of penalizing the covered instances is now described. Every time the evolutive process ends and the best individual is chosen as the best rule, the database is processed in order to find those instances already covered by the rule. Hence, each instance has a counter that increases its value by one every time a rule covers it.

Rules with a high number of attributes provide more information but also, in many cases, it is difficult to find rules where a high number of attributes appear. The number of attributes of a rule can be adjusted by means of the weight w_n.

Finally, the amplitude controls the size of the intervals of the attributes that compose the rules and those individuals with large intervals are penalized by means of the factor w_a, which allow the rules to be more or less permissive regarding the amplitude of the intervals.

3.6. The IRL approach

The proposed algorithm is based on the Iterative Rule Learning (IRL) process, whose general scheme is illustrated in Fig. 2.

In each iteration, the CHC takes place. Thus, in each evolutionary process, the individual that represents the best rule is chosen. If the maximum number of rules

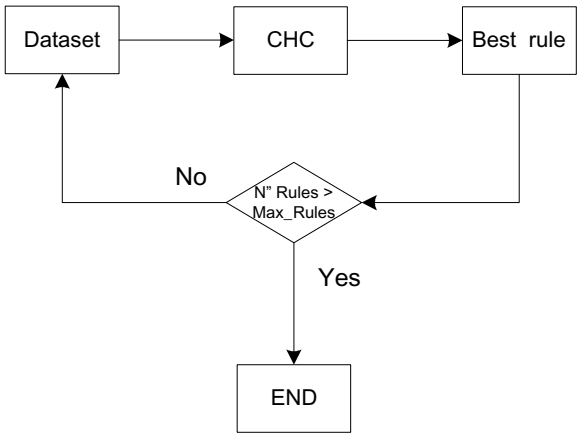

Fig. 2. Scheme of IRL.

to be found is not reached, the samples that have been covered are checked.

The goal of this process is to penalize the instances covered by the best rule in order to cover the remaining instances in subsequent iterations. Subsequently, coverage of search space regions is attempted and the set of rules covers all the domain of the consequent. The iterative process ends when the maximum number of desired rules are found.

Figure 3 illustrates how the proposed algorithm works with the CHC inserted as a crucial step of the IRL process.

First, the population is initialized and the crossover threshold, *MinDist*, is set to prevent incest. In each iteration of the CHC, the population is evaluated and the fitness of each individual is calculated according to (6). Then, the crossover operator without incest is applied to a maximum number of parents (equal to half the population), as described in Section 3.3. Those parents that overlap *MinDist* are crossed to prevent the incest and generate new offspring. Thus, a maximum distance between offspring and parents is guaranteed. Later, the elitist selection takes place. The N best individuals are chosen from the current generation and from the offspring. If no new individuals are created in the current generation, *MinDist* is decremented. In case the threshold was less than zero, the population and the threshold are reinitialized. Finally, the process has to be carried out as many times as the maximum number of generations indicates.

4. Application to synthetic datasets

The proposed algorithm has been applied to the same synthetic dataset used in [4] with the aim of determining if it is possible to find AR with the precise values for the numeric intervals to which each attribute of the rule belongs to.

Table 2
Synthetic sets

A1 $\in$ [1, 10] $\wedge$ A2 $\in$ [15, 30]
A1 $\in$ [15, 45] $\wedge$ A3 $\in$ [60, 75]
A2 $\in$ [65, 90] $\wedge$ A4 $\in$ [15, 45]
A3 $\in$ [80, 100] $\wedge$ A4 $\in$ [80, 100]

The synthetic dataset is composed of 1000 instances with four numeric attributes each. The selected interval is $[0, 100]$ and all values are uniformly distributed according to Table 2. Note that the amplitude of the intervals is different for each attribute. Furthermore, the datasets have been generated so that the support is 25% and the confidence is 100%. The generation of values out of such datasets are carried out so that no rules better than the ones provided by themselves can exist.

The main parameters of the proposed RCGA are as follows: 100 for the size of the population, 100 for the number of generations and 20 for the number of rules to be obtained. After an experimental study to test the influence of the weights on the rules to be obtained, the weights of the fitness function, 3 for w_s, 1 for w_c, 1.2 for w_r, 0.2 for w_n and 1 for w_a have been chosen.

Table 3 shows the best AR found by the proposed RCGA for the sythentic datasets described previously. The values for support and confidence are also provided, as well as the percentage of covered instances by all rules. It can be noted that the rules have a support of 25% and a confidence of 100%, according to the real values for both measures on the synthetic datasets considered.

These rules have been compared to those shown in Table 4, which have been obtained through a multi-objective differential evolution algorithm (MODENAR) that was recently published in [4]. It can be appreciated that rules obtained by the RCGA share the same support and confidence to those found by MODENAR. Nevertheless, the intervals, to which the numeric attributes belong, determined that RCGA is more precise than MODENAR, since such intervals present the same range and amplitude as those intervals shown in Table 2. In conclusion, it can be stated that the rules found by RCGA are more precise to those found by MODENAR even if the support and confidence are the same.

Different levels of noise have been added to the synthetic datasets in order to validate the efficiency of the RCGA. Thus, values that are not comprised of the in-

Table 3
Association rules found by RCGA

Rule	Support (%)	Confidence (%)	Records (%)
A1 $\in$ [1, 10] $\Longrightarrow$ A2 $\in$ [15, 30]	25	100	100
A1 $\in$ [15, 45] $\Longrightarrow$ A3 $\in$ [60, 75]	25	100	
A3 $\in$ [80, 100] $\Longrightarrow$ A4 $\in$ [80, 100]	25	100	
A2 $\in$ [65, 90] $\Longrightarrow$ A4 $\in$ [15, 45]	25	100	
A2 $\in$ [15, 30] $\Longrightarrow$ A1 $\in$ [1, 10]	25	100	
A3 $\in$ [60, 75] $\Longrightarrow$ A1 $\in$ [15, 45]	25	100	
A4 $\in$ [80, 100] $\Longrightarrow$ A3 $\in$ [80, 100]	25	100	
A4 $\in$ [15, 45] $\Longrightarrow$ A2 $\in$ [65, 90]	25	100	

Table 4
Association rules found by MODENAR

Rule	Support (%)	Confidence (%)	Records (%)
A1 $\in$ [1, 10] $\Longrightarrow$ A2 $\in$ [15, 30]	25	100	100
A1 $\in$ [15, 45] $\Longrightarrow$ A3 $\in$ [60, 75]	25	100	
A3 $\in$ [80, 100] $\Longrightarrow$ A4 $\in$ [80, 98]	25	100	
A2 $\in$ [65, 90] $\Longrightarrow$ A4 $\in$ [15, 43]	25	100	
A2 $\in$ [15, 30] $\Longrightarrow$ A1 $\in$ [1, 10]	25	100	
A3 $\in$ [60, 75] $\Longrightarrow$ A1 $\in$ [15, 45]	25	100	
A4 $\in$ [80, 98] $\Longrightarrow$ A3 $\in$ [80, 100]	25	100	
A4 $\in$ [15, 44] $\Longrightarrow$ A2 $\in$ [65, 89]	25	100	

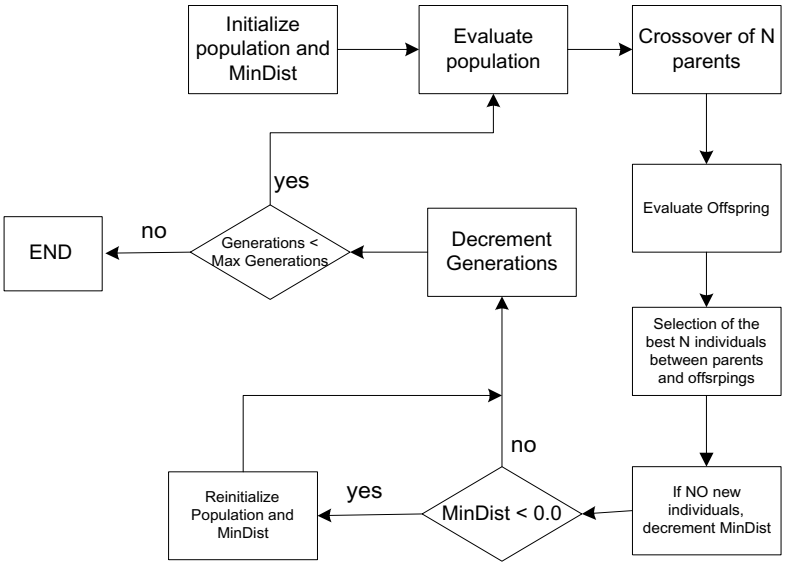

Fig. 3. Scheme of algorithm CHC.

terval of the second item (A_2) of the dataset have been inserted, that is, a percentage r of instances exist whose second item does not belong to the preset interval. The RCGA has been tested with three different levels of noise (4%, 6% and 8% for the value r).

Table 5 shows the rules obtained by applying RCGA to the different synthetic datasets after the noise addition. The support and confidence values are also provided, as well as the percentage of covered instances by all rules. For the three noise levels, all the extracted rules (but one) have exact intervals. Equally remarkable is that for all noise levels the support in most rules coincides with the real support values which are 24%, 23.5% y 23%, for noise levels of 4%, 6% and 8% respectively.

Table 6 shows the AR, the support values, the confidence and the percentage of covered instances obtained by the MODENAR algorithm for different levels of noise in synthetic datasets. Note that for the case where noise level is 4% the range of the intervals are close

Table 5
Rules mined under different noise level (RCGA)

Mined rules	Support (%)	Confidence (%)	Records (%)
r = 4%			
A1 ∈ [1, 10] ⟹ A2 ∈ [15, 30]	24.0	96.0	96.0
A1 ∈ [15, 45] ⟹ A3 ∈ [60, 75]	24.0	96.0	
A3 ∈ [80, 100] ⟹ A4 ∈ [80, 100]	24.0	96.0	
A2 ∈ [65, 90] ⟹ A4 ∈ [15, 46]	24.2	95.0	
A2 ∈ [15, 30] ⟹ A1 ∈ [1, 10]	24.0	100	
A3 ∈ [60, 75] ⟹ A1 ∈ [15, 45]	24.0	100	
A4 ∈ [80, 100] ⟹ A3 ∈ [80, 100]	24.0	98.8	
A4 ∈ [15, 45] ⟹ A2 ∈ [65, 90]	24.0	99.0	
r = 6%			
A1 ∈ [1, 10] ⟹ A2 ∈ [12, 30]	23.7	94.0	94.0
A1 ∈ [15, 45] ⟹ A3 ∈ [60, 75]	23.5	94.0	
A3 ∈ [80, 100] ⟹ A4 ∈ [79, 100]	23.6	92.9	
A2 ∈ [65, 90] ⟹ A4 ∈ [15, 45]	23.5	92.5	
A2 ∈ [15, 30] ⟹ A1 ∈ [1, 10]	23.5	100	
A3 ∈ [60, 75] ⟹ A1 ∈ [15, 45]	23.5	100.0	
A4 ∈ [80, 100] ⟹ A3 ∈ [80, 100]	23.5	97.5	
A4 ∈ [15, 45] ⟹ A2 ∈ [65, 90]	23.5	97.5	
r = 8%			
A1 ∈ [1, 10] ⟹ A2 ∈ [15, 30]	23.0	92.0	92.0
A1 ∈ [15, 45] ⟹ A3 ∈ [60, 75]	23.0	92.0	
A3 ∈ [80, 100] ⟹ A4 ∈ [80, 100]	23.0	89.8	
A2 ∈ [65, 90] ⟹ A4 ∈ [15, 45]	23.0	92.0	
A2 ∈ [15, 30] ⟹ A1 ∈ [1, 10]	23.0	100.0	
A3 ∈ [60, 75] ⟹ A1 ∈ [15, 45]	23.0	100.0	
A4 ∈ [80, 100] ⟹ A3 ∈ [80, 100]	23.0	97.8	
A4 ∈ [15, 45] ⟹ A2 ∈ [65, 90]	23.0	95.4	

Table 6
Rules mined under different noise level (MODENAR)

Mined rules	Support (%)	Confidence (%)	Records (%)
r = 4%			
A1 ∈ [1, 10] ⟹ A2 ∈ [15, 29]	24.1	100.0	96.0
A1 ∈ [15, 45] ⟹ A3 ∈ [60, 73]	24.0	100.0	
A3 ∈ [80, 100] ⟹ A4 ∈ [80, 96]	23.7	96.7	
A2 ∈ [65, 90] ⟹ A4 ∈ [15, 46]	24.2	98.3	
A2 ∈ [15, 29] ⟹ A1 ∈ [1, 10]	24.1	100.0	
A3 ∈ [60, 73] ⟹ A1 ∈ [15, 45]	24.0	100.0	
A4 ∈ [80, 96] ⟹ A3 ∈ [80, 100]	23.7	96.7	
A4 ∈ [15, 46] ⟹ A2 ∈ [65, 89]	24.2	98.3	
r = 6%			
A1 ∈ [1, 11] ⟹ A2 ∈ [14, 31]	23.3	98.9	94.0
A1 ∈ [15, 45] ⟹ A3 ∈ [56, 73]	23.6	99.0	
A3 ∈ [80, 100] ⟹ A4 ∈ [84, 95]	23.3	94.5	
A2 ∈ [65, 89] ⟹ A4 ∈ [14, 49]	23.8	97.8	
A2 ∈ [14, 31] ⟹ A1 ∈ [1, 11]	23.3	98.9	
A3 ∈ [56, 73] ⟹ A1 ∈ [15, 45]	23.6	99.0	
A4 ∈ [84, 95] ⟹ A3 ∈ [80, 100]	23.3	94.5	
A4 ∈ [14, 49] ⟹ A2 ∈ [65, 89]	23.8	97.8	
r = 8%			
A1 ∈ [1, 11] ⟹ A2 ∈ [14, 29]	22.4	97.6	91.8
A1 ∈ [15, 45] ⟹ A3 ∈ [62, 76]	22.9	98.0	
A3 ∈ [79, 100] ⟹ A4 ∈ [82, 98]	22.8	93.4	
A2 ∈ [65, 90] ⟹ A4 ∈ [15, 48]	23.7	95.8	
A2 ∈ [14, 29] ⟹ A1 ∈ [1, 11]	22.4	97.6	
A3 ∈ [62, 76] ⟹ A1 ∈ [15, 45]	22.9	98.0	
A4 ∈ [82, 98] ⟹ A3 ∈ [79, 100]	22.8	93.4	
A4 ∈ [15, 48] ⟹ A2 ∈ [65, 90]	23.7	95.8	

to the real intervals synthetically generated but are not exact. The support in all cases is close to the real value being equal in just two rules. For this level the proposed algorithm provided better rules with more exact intervals than those provided by MODENAR, which implies better support for such rules. However, the confidence values for rules found by the RCGA are slightly lower than those found by MODENAR.

For a noise of 6%, it can be observed that none of the obtained rules by MODENAR has exactly the same intervals as those used to generate the synthetic dataset. Therefore, the support differs from the real value –equal to 23.5%– for this noise level. Likewise, none of the cases reach a confidence of 100%. Nevertheless, it can be observed in Table 5 that in most cases where the RCGA is applied, exact intervals are obtained. This fact entails confidence values of 100% for some rules and a support of 23.5% for most cases, as opposed to MODENAR.

Analogously for a noise level of 8%, if the rules shown in Tables 5 and 6 are compared, it can be concluded that the behavior of the proposed algorithm with noise is similar to that of previous levels. Consequently, the rules obtained for this level of noise have more precise intervals than those obtained by MODENAR. This improvement entails reaching the real value of the support in the majority of cases. Also, the confidence achieved with the RCGA is 100% for two rules, whereas value has never been fully achieved with the rules obtained from MODENAR.

In conclusion, it can be stated that the RCGA satisfactorily extracted rules for synthetic datasets containing noise, since it showed its ability to overcome different levels of noise, even providing an improvement to the rules provided by MODENAR.

5. Application to atmospheric pollution

The proposed algorithm has been applied in order to discover AR between climatological variables –temperature, humidity, wind direction, wind speed–, the hour of the day and day of the week, and three pollutant agents (ozone, nitrogen monoxide and sulfur dioxide). Therefore, these variables are forced to belong to the consequent. However, the intervals are not previously fixed which differentiates from Apriori and the SD issue.

All variables have been retrieved from a meteorological station placed in the outskirts of Seville city (Spain), providing hourly records of them. It is worth mentioning that Seville is a very hot city that frequently reaches temperatures greater than 40 °C during the summer. The following sections detail the rules obtained for each variable.

5.1. Extracting rules for the ozone

AR have been extracted for ozone (O_3) in two different time periods: from July to August in both 2003 and 2004, which leads to a dataset composed of 1688 instances. The selection of such periods is due to the high concentration of ozone present in the aforementioned summers. For prediction purposes, the climatological time series have been forced to belong to the antecedent and the ozone to the consequent. As a result, a prediction of ozone is achieved on the basis of rules extracted from these variables.

Several experiments have been carried out, in which the main parameters of the GA were as follows: 100 for the size of the population and 100 for the number of generations; 20 for the number of rules to be obtained. After an experimental study to test the influence of the weights on the rules to be obtained, the weights of the fitness function, 0.5 for w_s, 2 for w_c, 6 for w_r, 0.2 for w_n and 7 for w_a have been selected.

The most relevant rules are the ones that identify high concentration of ozone. However, this situation is just under 6.5% of the whole dataset and for this reason, the value w_s has been set low and w_a high, since rules with small amplitudes are desirable. Also, w_r has been set with a high value in order to promote rules that cover instances with high ozone concentration.

The experimentation carried out is detailed in following Tables, in which only the most significative rules are represented. Also, it must be noted that the confidence is the percentage of instances covered by the rule in which only the antecedent is covered.

Table 7 outlines the rules obtained when temperature was the antecedent and ozone the consequent, taking into consideration only those rules whose consequent possesses values of high ozone concentration –typically 170 microgrammes per cubic meter, $[\mu g/m^3]$– to which citizens must be informed of such situations. It can be easily concluded that temperature and ozone are directly related, since an increase in temperature involves an increase in the ozone. Another remarkable feature is the perfect division of the temperature ranges regarding ozone as no overlapping is detected. For temperatures ranging from 35°C to 37°C, ozone values were from 157 $\mu g/m^3$ to 175 $\mu g/m^3$ approximately. Likewise, a temperature in the range [38, 40]°C entails ozone

Table 7
Association rules found by RCGA for temperature (°C) and O_3 ($\mu g/m^3$)

Rule	Support (%)	Confidence (%)
temperature $\in$ [34.9, 37.0] $\Longrightarrow$ $O_3 \in$ [157.7, 175.8]	9.7	19.4
temperature $\in$ [38.6, 40.6] $\Longrightarrow$ $O_3 \in$ [180.0, 202.3]	8.3	22.6
temperature $\in$ [42.8, 44.9] $\Longrightarrow$ $O_3 \in$ [205.8, 223.5]	1.4	66.6

Table 8
Association rules found by RCGA for humidity (%) and O_3 ($\mu g/m^3$)

Rule	Support (%)	Confidence (%)
humidity $\in$ [14.0, 20.0] $\Longrightarrow$ $O_3 \in$ [124.2, 163.7]	4.8	77.7
humidity $\in$ [38.6, 40.6] $\Longrightarrow$ $O_3 \in$ [180.0, 202,3]	5.5	19.5

Table 9
Association rules found by RCGA for wind direction (°) and O_3 ($\mu g/m^3$)

Rule	Support (%)	Confidence (%)
direction $\in$ [91.8, 117.1] $\Longrightarrow$ $O_3 \in$ [144.0, 161.7]	2.1	33.3
direction $\in$ [208.6, 233.8] $\Longrightarrow$ $O_3 \in$ [127.8, 145.5]	13.8	20.0

Table 10
Association rules found by RCGA for wind speed (m/s) and O_3 ($\mu g/m^3$)

Rule	Support (%)	Confidence (%)
speed $\in$ [18.1, 20.0] $\Longrightarrow$ $O_3 \in$ [91.2, 160.8]	29.6	89.5

Table 11
Association rules found by RCGA for hour of the day and O_3 ($\mu g/m^3$)

Rule	Support (%)	Confidence (%)
hour $\in$ [11 am, 1:30 pm] $\Longrightarrow$ $O_3 \in$ [123.5, 141.2]	14.4	17.8
hour $\in$ [2 pm, 3:30 pm] $\Longrightarrow$ $O_3 \in$ [137.3, 157.7]	25.5	30.8
hour $\in$ [3 pm, 4:30 pm] $\Longrightarrow$ $O_3 \in$ [160.3, 178.0]	8.9	21.3
hour $\in$ [4 pm, 5:30 pm] $\Longrightarrow$ $O_3 \in$ [130.7, 166.3]	32.4	38.5
hour $\in$ [8 pm, 9:30 pm] $\Longrightarrow$ $O_3 \in$ [135.9, 153.6]	8.2	19.6

levels of 180 $\mu g/m^3$ and 200 $\mu g/m^3$. Finally, when the temperature reaches 42°C, the ozone has values greater than 200 $\mu g/m^3$. The last rule is of the utmost importance since the confidence obtained is 66%.

Table 8 shows the rules in which ozone reaches its highest values when the humidity is the antecedent. As it can be noted, the humidity triggers considerably high values of ozone when it reaches values between 14% and 20%. Equally remarkable is the second rule in which the ozone exceeds levels of 180 $\mu g/m^3$ when the humidity lies between 38.6% and 40.6%.

Table 9 describes the rules in which ozone had high values when analyzing the wind direction. Ozone levels start to rise when wind direction varies between 210° and 230°. However, the highest ozone concentration found in the atmosphere is when the wind direction is in the range from 90° to 120°, reaching values around 160 $\mu g/m^3$. The precision of both rules is similar, since confidence verges on 25% for both situations.

The rules that relate wind speed and ozone are found in Table 10. With high accuracy, a confidence of 89.5%, ozone reaches moderate values when wind speed is between 18 m/s and 20 m/s.

Table 11 presents hours of the day (in the antecedent) when higher values of ozone (in the consequent) are detected in the atmosphere. According to the obtained rules, it can be concluded that these hours coincide with hours of heavy traffic, that is, the highest concentrations are found from 2 pm to 4:30 pm and from 8 pm to 9:30 pm. These intervals of time are typically associated with the end of schooltime and the working day in Spain. On the contrary, the lowest levels are detected from 11 am to 1:30 pm, the time in which most people are working or studying. All the rules share values of similar confidence, comprising between 20% and 40%.

Table 12 makes reference to the highest concentrations of ozone distributed throughout the days of the week. It can be appreciated that on the first (Monday) and third day of the week, ozone may reach levels greater than 180 $\mu g/m^3$. In addition, Fridays also produce elevated concentrations of ozone. Applying a

Table 12
Association rules found by RCGA for day of the week and O_3 ($\mu g/m^3$)

Rule	Support (%)	Confidence (%)
day $\in$ [1, 2] $\Longrightarrow$ $O_3 \in$ [168.4, 186.1]	8.2	5.6
day $\in$ [2, 3] $\Longrightarrow$ $O_3 \in$ [130.7, 166.3]	9.6	6.9
day $\in$ [3, 4] $\Longrightarrow$ $O_3 \in$ [171.6, 189.3]	6.9	5.0
day $\in$ [4, 5] $\Longrightarrow$ $O_3 \in$ [136.6, 154.2]	13.8	9.9
day $\in$ [5, 6] $\Longrightarrow$ $O_3 \in$ [154.1, 171.8]	7.6	5.1
day $\in$ [6, 7] $\Longrightarrow$ $O_3 \in$ [132.2, 149.9]	13.8	9.3

Table 13
Association rules found by RCGA for temperature (°C) and NO ($\mu g/m^3$)

Rule	Support (%)	Confidence (%)
temperature $\in$ [35.7, 37.7] $\Longrightarrow$ NO $\in$ [3.0, 8.0]	4.3	88.8
temperature $\in$ [38.3, 40,3] $\Longrightarrow$ NO $\in$ [3.0, 6.9]	3.9	96.6
temperature $\in$ [40.5, 42.6] $\Longrightarrow$ NO $\in$ [3.0, 8.2]	1.1	94.1
temperature $\in$ [42.9, 45.0] $\Longrightarrow$ NO $\in$ [3.0, 6.9]	0.3	100

Table 14
Association rules found by RCGA for humidity (%) and NO ($\mu g/m^3$)

Rule	Support (%)	Confidence (%)
humidity $\in$ [14.0, 19.4] $\Longrightarrow$ NO $\in$ [3.0, 6.9]	0.5	100
humidity $\in$ [36.1, 41.5] $\Longrightarrow$ NO $\in$ [3.0, 7.0]	6.7	73.0

Table 15
Association rules found by RCGA for wind direction (°) and NO ($\mu g/m^3$)

Rule	Support (%)	Confidence (%)
direction $\in$ [88.1, 114.1] $\Longrightarrow$ NO $\in$ [3.0, 6.9]	0.6	81.8
direction $\in$ [208.3, 233.5] $\Longrightarrow$ NO $\in$ [3.0, 7.0]	6.4	93.2

Table 16
Association rules found by RCGA for wind speed (m/s) and NO ($\mu g/m^3$)

Rule	Support (%)	Confidence (%)
speed $\in$ [18.1, 20.0] $\Longrightarrow$ NO $\in$ [3.0, 6.9]	3.6	100

similar rationale to that of Table 11, it can be concluded that the highest values are associated with days with heavy traffic, that is, the first and last working days of the week. A slight decrease of the ozone is detected in the middle of the week as well as over the weekend. All rules present similar levels of confidence, within 5% and 10%.

5.2. Extracting rules for nitrogen monoxide

AR have also been extracted for nitrogen monoxide (NO). This pollutant agent is typically generated by the direct combination of nitrogen and oxygen. The analysis of NO levels in the atmosphere is relevant since it directly contributes to the generation of nitrogen dioxide NO_2, which is an extremely oxidant agent resulting from the oxidation of NO. NO_2 is one of the precursors of photochemical smog and it can easily be recognized in big cities due to the reddish coloration of the air.

To carry out the experimentation, the climatological variables used in the previous section (temperature, humidity, wind direction, wind speed, hour of the day and day of the week) have been considered to belong to the antecedent and the nitrogen monoxide to the consequent. It also needs to be mentioned that the parameters as well as the associated weights to each attribute in the fitness function are the same to the ones used for the ozone experimentation.

Furthermore, in order to perform comparisons with results from ozone, rules with antecedents similar to those of ozone have been chosen, that is, rules in which ozone presented high levels of concentration.

Tables 13, 14, 15, 16, 17 and 18 show the rules discovered for the NO and related with temperature, humidity, wind direction, wind speed, hour of the day and day of the week, respectively.

Table 17
Association rules found by RCGA for hour of the day and $NO\,(\mu g/m^3)$

Rule	Support (%)	Confidence (%)
hour $\in$ [12 pm, 1:30 pm] $\Longrightarrow$ NO $\in$ [3.0, 6,9]	6.9	83.1
hour $\in$ [2 pm, 3:30 pm] $\Longrightarrow$ NO $\in$ [3.0, 6.9]	4.1	100
hour $\in$ [3 pm, 4:30 pm] $\Longrightarrow$ NO $\in$ [3.0, 6.9]	8.3	100
hour $\in$ [4 pm, 5:30 pm] $\Longrightarrow$ NO $\in$ [3.0, 6.9]	8.2	99
hour $\in$ [8 pm, 9:30 pm] $\Longrightarrow$ NO $\in$ [3.0, 6.9]	4.1	100

Table 18
Association rules found by RCGA for day of the week and NO $(\mu g/m^3)$

Rule	Support (%)	Confidence (%)
day $\in$ [1,2] $\Longrightarrow$ NO $\in$ [3.0,7.0]	12.8	88.4
day $\in$ [2,3] $\Longrightarrow$ NO $\in$ [3.0, 6.9]	12.8	88.4
day $\in$ [3,4] $\Longrightarrow$ NO $\in$ [3.0, 6.9]	12.6	87.0
day $\in$ [4,5] $\Longrightarrow$ NO $\in$ [3.0, 6.9]	12.7	87.5
day $\in$ [5,6] $\Longrightarrow$ NO $\in$ [3.0, 6.9]	13.8	95.3
day $\in$ [6,7] $\Longrightarrow$ NO $\in$ [3.0, 6.9]	14.2	98.1

Table 19
Association rules found by RCGA for temperature (°C) and SO_2 $(\mu g/m^3)$

Rule	Support (%)	Confidence (%)
temperature $\in$ [35.9, 38.0] $\Longrightarrow$ $SO_2 \in$ [10.8, 13.0]	1.4	27.5
temperature $\in$ [37.9, 40.0] $\Longrightarrow$ $SO_2 \in$ [7.0, 10.3]	2.2	53.2
temperature $\in$ [42.9, 45.0] $\Longrightarrow$ $SO_2 \in$ [3.7, 7.5]	0.3	100

The analysis of the aforementioned Tables reveals that the values for nitrogen monoxide in each case always varies in the interval comprising of 3 $\mu g/m^3$ and 6 $\mu g/m^3$ with a confidence verging on 100% in all cases. These values are typically considered to be very low and, moreover, it remains invariable with independence of the values of the intervals appearing in the antecedent. This feature allows for concluding that nitrogen monoxide cannot be predicted by means of any of the attributes existing in the dataset. That is, these time series are not correlated enough in regard to NO (coefficient of correlation equals 0.1233 in comparison to ozone which equals 0.3777) and, consequently, no useful information can be extracted from their analysis.

Fortunately, these results are logical because, on one side, NO oxidizes and creates NO_2 and, on the other, NO_2 is dissociated in particles of NO and atomic oxygen (O) in presence of solar light. Besides, O reacts with molecular environmental oxygen (O_2) and produces ozone (O_3). Therefore, low values of nitrogen monoxide in the atmosphere are strongly ligated to high values of ozone in the intervals of interest.

5.3. Extracting rules for sulfur dioxide

The study of sulfur dioxide in the air is a concerning subject since, apart from being responsible for the generation of sulfuric acid(H_2SO_4), it deeply affects people's health, causing respiratory diseases. The atmospheric SO_2 may oxidize and generate SO_3 and react with humidity (H_2O) by absorption, thus generating thus the molecules of sulfuric acid. These molecules can be dispersed in the air, contributing to the acidification process of the earth and water particles.

Hence, this section describes the experimentation carried out to predict sulfur dioxide (SO_2) from the same climatological time series used in the previous sections. Moreover, all parameters and weights that take part in the fitness function have been set with the same values. The time series are only allowed to appear in the antecedent and sulfur dioxide only in the consequent. Thus, the forecasting is performed on the same basis used for rules discovered in previous sections. Also note that, in order to perform comparisons with results from ozone and nitrogen monoxide, rules with antecedents similar to those of ozone have been chosen, that is, rules in which ozone presented high concentration levels.

Table 19 shows rules relating to the temperature (in the antecedent) and sulfur dioxide (in the consequent), in which no overlapped intervals exist. From its findings, it can be stated that higher the temperature, the less sulfur dioxide there is in the air. This statement may be contradictory since it is reasonable to think that sulfur dioxide increases along with temperature. However, the obtained data are what experts expect since

Table 20
Association rules found by RCGA for humidity (%) and SO_2 ($\mu g/m^3$)

Rule	Support (%)	Confidence (%)
humidity $\in$ [14.1, 19.5] $\Longrightarrow$ $SO_2 \in$ [9.5, 11.6]	0.2	50.0
humidity $\in$ [34.2, 39.5] $\Longrightarrow$ $SO_2 \in$ [3.0, 9.7]	6.2	62.1

Table 21
Association rules found by RCGA for wind direction (°) and SO_2 ($\mu g/m^3$)

Rule	Support (%)	Confidence (%)
direction $\in$ [44.4, 69.6] $\Longrightarrow$ $SO_2 \in$ [3.0, 10.7]	1.3	80.0
direction $\in$ [125.4, 150.6] $\Longrightarrow$ $SO_2 \in$ [3.0, 10.0]	8.4	82.3
direction $\in$ [185.2, 210.4] $\Longrightarrow$ $SO_2 \in$ [3.0, 9.2]	6.1	71.8
direction $\in$ [245.7, 270.9] $\Longrightarrow$ $SO_2 \in$ [3.0, 10.2]	2.8	82.3

Table 22
Association rules found by RCGA for wind speed (m/s) and SO_2 ($\mu g/m^3$)

Rule	Support (%)	Confidence (%)
speed $\in$ [0.0, 1.9] $\Longrightarrow$ $SO_2 \in$ [3.0, 6.9]	8.4	83.4
speed $\in$ [17.5, 19.4] $\Longrightarrow$ $SO_2 \in$ [3.0, 6.9]	2.7	74.5
speed $\in$ [25.7, 27.7] $\Longrightarrow$ $SO_2 \in$ [3.0, 6.9]	0.5	63.6

Table 23
Association rules found by RCGA for hour of the day and SO_2 ($\mu g/m^3$)

Rule	Support (%)	Confidence (%)
hour $\in$ [3 am, 4:30 am] $\Longrightarrow$ $SO_2 \in$ [3.0, 6.3]	2.3	54.8
hour $\in$ [11 am, 12:30 pm] $\Longrightarrow$ $SO_2 \in$ [8.6, 10.8]	1.9	23.4
hour $\in$ [12 pm, 1:30 pm] $\Longrightarrow$ $SO_2 \in$ [11.6, 13.8]	1.4	17.7
hour $\in$ [1 pm, 2:30 pm] $\Longrightarrow$ $SO_2 \in$ [13.6, 15.8]	0.6	14.5
hour $\in$ [4 pm, 5:30 pm] $\Longrightarrow$ $SO_2 \in$ [6.7, 11.8]	2.2	51.6
hour $\in$ [8 pm, 9:30 pm] $\Longrightarrow$ $SO_2 \in$ [11.3, 13.7]	1.3	16.1

the presence of this particle in the air is inversely related to the solar radiation, that is, to the temperature. Therefore, when the temperature increases, the dioxide reacts quicker, generating sulfuric acid and reducing sulfur dioxide concentration. Specifically, when temperature ranges from 35 °C to 38 °C, sulfur dioxide falls in the interval 10-13 μ/m^3 and when temperature reaches 40°C, sulfur dioxide reduces its concentration from 3 $\mu g/m^3$ to 7 $\mu g/m^3$. The last rule is especially reliable due to its high confidence.

Table 20 shows the rules relating to the humidity (in the antecedent) and sulfur dioxide (in the consequent), in which no overlapped intervals exist either. As with temperature, sulfur dioxide is inversely related to humidity. Thus, for humidity between 14% and 19%, sulfur dioxide levels are in the range of 9–11 μ/m^3. Furthermore, when the temperature nears 40 °C, gas concentration is reduced to a level of 3 μ/m^3. The explanation for this phenomenon is similar to that of temperature since the reaction of sulfur dioxide is accelerated by means of humidity absorption, that is, the more humidity, the less sulfur dioxide.

Table 21 presents the rules extracted when using the wind direction as antecedent. In this case, the intervals obtained for sulfur dioxide remain invariable even if the wind direction varies. Consequently, it can be concluded that wind direction does not influence levels of sulfur dioxide in the atmosphere.

Table 22 is devoted to presenting rules extracted when using wind speed as antecedent. As with wind direction, the intervals obtained for the consequent do not vary, independently of the values in the antecedent. Consequently, it can be concluded that the wind speed does not influence levels of sulfur dioxide in the atmosphere.

Table 23 shows the rules for the different hours of a day. As it can be appreciated, high concentrations of sulfur dioxide are concentrated in Spanish rush hours. For instance, when considering the interval from 1 pm to 2:30 pm, the gas reaches values close to 15 $\mu g/m^3$. In comparison, the concentration from 3 am to 4:30 am is no greater than 6 $\mu g/m^3$.

Table 24 describes the rules associated with the day of the week that help sulfur dioxide forecasting. The highest concentrations are on Mondays and Fridays and

Table 24
Association rules found by RCGA for day of the week and SO_2 ($\mu g/m^3$)

Rule	Support (%)	Confidence (%)
day $\in$ [1,2] $\Longrightarrow$ $SO_2 \in$ [14.2, 16.3]	0.9	6.7
day $\in$ [2,3] $\Longrightarrow$ $SO_2 \in$ [9.8, 12.1]	1.6	11.5
day $\in$ [3,4] $\Longrightarrow$ $SO_2 \in$ [12.4, 14.5]	1	15.2
day $\in$ [4,5] $\Longrightarrow$ $SO_2 \in$ [8.9, 11.0]	2.2	15.2
day $\in$ [5,6] $\Longrightarrow$ $SO_2 \in$ [15.7, 17.8]	0.8	5.5
day $\in$ [6,7] $\Longrightarrow$ $SO_2 \in$ [9.4, 11.5]	2.6	18.0

the explanation of this situation is similar to the one provided for the hour of the day. Heavy traffic at the beginning and end of the week causes an increase in the combustion levels, which leads to a higher concentration of this gas in the atmosphere.

6. Conclusions

A new algorithm has been proposed in this work in order to discover quantitative AR. The approach is based on the well-known CHC and works diametrally different as most algorithms do, since it does not discretize the attributes as a first step of the process. Moreover, the algorithm has been evaluated over different datasets. On one hand, synthetic data have been mined and the results were compared with those provided by the MODENAR algorithm, reporting better rules in terms of confidence and support. Additionally, the algorithm has been applied to pollutant agents time series and shown to be effective for forecasting purposes. The use of these kind of tools with such data is, to the best of the authors knowledge, unique. Furthermore, the mined rules agreed with chemical processes associated with these agents.

Acknowledgments

The financial support from the Spanish Ministry of Science and Technology, project TIN2007-68084-C-00, and from the Junta de Andalucía, project P07-TIC-02611, is acknowledged. The authors also want to acknowledge the support of the Regional Ministry for the Environment (*Consejería de Medio Ambiente*) of Andalucía (Spain), that has provided all the pollutant agents time series.

References

[1] 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, pages 207–216, 1993.

[2] J.S. Aguilar-Ruiz, R. Giráldez and J.C. Riquelme, Natural encoding for evolutionary supervised learning, *IEEE Transactions on Evolutionary Computation* **11**(4) (2007), 466–479.

[3] B. Alatas and E. Akin, Rough particle swarm optimization and its applications in data mining, *Soft Computing* **12**(12) (2008), 1205–1218.

[4] B. Alatas, E. Akin and A. Karci, MODENAR: Multi-objective differential evolution algorithm for mining numeric association rules, *Applied Soft Computing* **8**(1) (2008), 646–656.

[5] J. Alcalá-Fdez, R. Alcalá, M.J. Gacto and F. Herrera, Learning the membership function contexts forming fuzzy association rules by using genetic algorithms, *Fuzzy Sets and Systems* **160**(7) (2009), 905–921.

[6] Y. Aumann and Y. Lindell, A statistical theory for quantitative association rules, *Journal of Intelligent Information Systems* **20**(3) (2003), 255–283.

[7] S. Ayubi, M.K. Muyeba, A. Baraani and J. Keane, An algorithm to mine general association rules from tabular data, *Information Sciences* **179** (2009), 3520–3539.

[8] G. Box and G. Jenkins, *Time Series Analysis: Forecasting and Control*, John Wiley and Sons, 2008.

[9] L. Carro-Calvo, S. Salcedo-Sanz, R. Gil-Pita, A. Portilla-Figueras and M. Rosa-Zurera, An evolutive multiclass algorithm for automatic classifiøcation of high range resolutio radar targets, *Integrated Computer-Aided Engineering* **16**(1) (2009), 51–60.

[10] L. Carro-Calvo, S. Salcedo-Sanz, R. Gil-Pita, A. Portilla-Figueras and M. Rosa-Zurera, An evolutive multiclass algorithm for automatic classifiøcation of high range resoluti radar targets, *Integrated Computer-Aided Engineering* **16**(1) (2009), 51–60.

[11] T.M. Cheng and R.Z. Yan, Integrating messy genetic algorithms and simulation to optimize resource utilization, *Computer-Aided Civil and Infrastructure Engineering* **24**(6) (2009), 401–415.

[12] O. Cordón, S. Dama and J. Santamara, Feature-based image registration by means of the CHC evolutionary algorithm, *Image and Vision Computing* **24** (2006), 525–533.

[13] M. J. del Jesús, P. González, F. Herrera and M. Mesonero, Evolutionary fuzzy rule induction process for subgroup discovery: A case study in marketing, *IEEE Transactions on Fuzzy Systems* **15**(4) (2007), 578–592.

[14] L. Dridi, M. Parizeau, A. Mailhot and J.P. Villeneuve, Using evolutionary optimisation techniques for scheduling water pipe renewal considering a short planning horizon, *Computer-Aided Civil and Infrastructure Engineering* **28**(8) (2008), 625–635.

[15] L. Eshelman, *The CHC Adaptative search algorithm: How to have Safe Search when Engaging in Nontraditional Genetic Recombination*, Morgan Kaufmann, 1991.

[16] E. Georgii, L. Richter, U. Rckert and S. Kramer, Analyzing microarray data using quantitative association rules, *BMC Bioinformatics* **21**(2) (2005), 123–129.

[17] D.L. Godbold and A. Huttermann, *Effects of Acid Rain on Forest Processes*, John Wiley and Sons, 1994.

[18] S. Guha, R. Rastogiand and K. Shim, *CURE: An Efficient Clustering Algorithm for Large Databases*, In Proceedings of ACM SIGMOD International Conference on Management of Data, pages 73–84, 1998.

[19] N. Gupta, N. Mangal, K. Tiwari and Pabitra Mitra, Mining quantitative association rules in protein sequences, *Lecture Notes in Artificial Intelligen* **3755** (2006), 273–281.

[20] 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.

[21] M. Kaya and R. Alhajj, Genetic algorithm based framework for mining fuzzy association rules, *Fuzzy Sets and Systems* **152**(3) (2005), 587–601.

[22] M. Kaya and R. Alhajj, Utilizing genetic algorithms to optimize membership functions for fuzzy weighted association rules mining, *Applied Intelligence* **24**(1) (2006), 7–152.

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

[24] H. Lee, E. Kim and M. Park, A genetic feature weighting scheme for pattern recognition, *Integrated Computer-Aided Engineering* **12**(2) (2007), 161–171.

[25] J. Mata, J.L. Álvarez and J.C. Riquelme, Discovering numeric association rules via evolutionary algorithm, *Lecture Notes in Artificial Intelligence* **2336** (2002), 40–51.

[26] S. Mathakari, P.P. Gardoni, P.P. Agarwal, A. Raich and T. Haukaas, Reliability-based optimal design of electrical transmission towers using multi-objective genetic algorithms, *Computer-Aided Civil and Infrastructure Engineering* **22**(4) (2007), 282–292.

[27] H. Nam, K. Lee and D. Lee, Identiøcation of temporal association rules from time-series microarray data sets, *BMC Bioinformatics* **10**(3) (2009), 1–9.

[28] A. Orriols-Puig, J. Casillas and E. Bernadó-Mansilla, *First Approach Toward on-line Evolution of Association Rules with Learning Classifier Systems*, In Proceedings of the 2008 GECCO Genetic and Evolutionary Computation Conference, pages 2031–2038, 2008.

[29] U. Ruckert, L. Richter and S. Kramer, *Quantitative Association Rules based on Half-Spaces: An Optimization Approach*, In Proceedings of the IEEE International Conference on Data Mining, pages 507–510, 2004.

[30] S.K. Sahua, S. Yipc and D.M. Hollandb, Improved space-time forecasting of next day ozone concentrations in the eastern US, *Atmospheric Environment* **43**(3) (2009), 494–501.

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

[32] Q. Tong, B. Yan and Y. Zhou, Mining quantitative association rules on overlapped intervals, *Lecture Notes in Artificial Intelligenc* **3584** (2005), 43–50.

[33] M. Vannucci and V. Colla, *Meaningful Discretization of Continuous Features for Association Rules Mining by Means of a Som*, In Proceedings of the European Symposium on Artiøcial Neural Networks, pages 489–494, 2004.

[34] G. Venturini, *SIA: a Supervised Inductive Algorithm with Genetic Search for Learning Attribute Based Concepts*, In Proceedings of the European Conference on Machine Learning, pages 280–296, 1993.

[35] E.I. Vlahogianni, M.G. Karlaftis and J.C. Golias, Spatio-temporal short-term urban traffic flow forecasting using genetically-optimized modular network, *Computer-Aided Civil and Infrastructure Engineering* **22**(5) (2007), 317–325.

[36] D. Wan, Y. Zhang and S. Li, *Discovery Association Rules in Time Series of Hydrology*, In Proceedings of the IEEE International Conference on Integration Technology, pages 653–657, 2007.

[37] 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.

[38] Y. Yin, Z. Zhong and Y. Wang, Mining quantitative association rules by interval clustering, *Journal of Computational Information Systems* **4**(2) (2008), 609–616.