International Journal of Swarm Intelligence Research
Volume 12 • Issue 4

A Discrete Crow Search Algorithm for Mining Quantitative Association Rules

Makhlouf Ledmi, Abbes Laghrour University of Khenchela, Algeria & Batna 2 University, Algeria
Hamouma Moumen, Batna 2 University, Algeria
Abderrahim Siam, Abbes Laghrour University of Khenchela, Algeria
Hichem Haouassi, Abbes Laghrour University of Khenchela, Algeria
Nabil Azizi, Abbes Laghrour University of Khenchela, Algeria

ABSTRACT

Association rules are the specific data mining methods aiming to discover explicit relations between the different attributes in a large dataset. However, in reality, several datasets may contain both numeric and categorical attributes. Recently, many meta-heuristic algorithms that mimic the nature are developed for solving continuous problems. This article proposes a new algorithm, DCSA-QAR, for mining quantitative association rules based on crow search algorithm (CSA). To accomplish this, new operators are defined to increase the ability to explore the searching space and ensure the transition from the continuous to the discrete version of CSA. Moreover, a new discretization algorithm is adopted for numerical attributes taking into account dependencies probably that exist between attributes. Finally, to evaluate the performance, DCSA-QAR is compared with particle swarm optimization and mono and multi-objective evolutionary approaches for mining association rules. The results obtained over real-world datasets show the outstanding performance of DCSA-QAR in terms of quality measures.

KEYWORDS

Association Rules Mining, Crow Search Optimization, Data Mining, Meta-Heuristic, Particle Swarm Optimization, Quantitative Association Rules

INTRODUCTION

Association rule mining is an important data mining task that aims to extract explicit relationships between the different attributes or items in a large dataset. This mining process can be divided into two essential phases: first, all of the frequent itemsets whose support exceeds a certain threshold are extracted, and second, association rules are generated from the frequent itemsets.

In practice, enumerating all frequent itemsets in a large dataset, especially when dealing with dense datasets or low support threshold values, is very costly (Alves, Rodríguez-Baena, & Aguilar-Ruiz, 2010). Besides, many datasets contain both numerical and categorical attributes that need to introduce more specific techniques to deal with these cases and take into account the different types of attributes and the high-dimensional feature space.

DOI: 10.4018/IJSIR.2021100106

In the recent years, many researchers utilized several meta-heuristic algorithms such as Genetic Algorithms (Kabir, Xu, Kang, & Zhao, 2017) and (Martín, Alcalá-Fdez, Rosete, & Herrera, 2016), Ant Colony Optimization (Manju and C. Kant, 2015) and Particle Swarm Optimization (Yan, Zhao, Lin, & Bai, 2019) to generate association rules sets with different performances by adopting search algorithms to obtain the best quality solutions (rules) in relation to the candidate solutions.

Recently, many hybrid algorithms are proposed for solving optimization problems to take full benefit of the advantages of both GA and PSO algorithm to improve the searchability and increase the exploration of the solution space ((Garg, 2019); (Narang, Patwal, & Garg, 2017) ; (Garg, 2016)). Many others bio-inspired algorithms like Grey Wolf Optimization (GWO), Wind Driven Optimization (WDO) and Whale Optimization Algorithm (WOA) are embedded with principal component analysis (PCA) into the deep neural network (DNN) to choose optimal parameters for training or extract relevant dimensions ((Gadekallu, et al., 2020); (Gadekallu, et al., 2020); (Iwendi, et al., 2020)).

It is worth noting that there is an overlap between works in this field and other areas. For instance, (Lakshman, Kaluri, Gadekallu, Nagaraja, & Subramanian, 2016) where the authors proposed an enhanced algorithm for discovering the most recurrently occurring patterns in biological sequences.

However, according to the NFL Theorem (No Free Lunch) (Wolpert & Macready, 1995), any meta-heuristic algorithm performs only as well as the knowledge concerning the cost function; hence it cannot be capable of dealing with all optimization problems, which would provide perspectives for proposing a new algorithm or improving existing algorithms.

Crow search algorithm (CSA) is one of the recently developed meta-heuristic algorithms successfully used to solve continuous problems by returning interesting results (Askarzadeh, 2016). Due to its simplicity and efficiency, CSA has been used to solve different problems such as feature selection problem (Gupta, et al., 2018), image segmentation (Oliva, et al., 2017), diagnosis of diseases (Gupta, Sundaram, Khanna, Ella Hassanien, & Albuquerque, 2018), electromagnetic optimization (dos Santos Coelho, Richter, Mariani, & Askarzadeh, 2016), economic load dispatch problem (Mohammadi & Abdi, 2018), and Scheduling Problems (Huang, Girsang, Wu, & Chuang, 2019).

However, CSA presents some disadvantages. The first is that the diversity of solutions can be deteriorated since CSA employs unidirectional search that can lead to local minima trapping. The second is that CSA doesn't use any inefficient search strategy regarding the promising region, which leads to the search without improvement in the quality of the solution and decreases the convergence rate of CSA ((Shekhawat & Saxena, 2020); (Hassanien, R. M., & Elhoseny, 2018)).

Motivating to remove these shortcomings and take full benefit of simplicity and efficiency of CSA to adapt it to dealing with quantitative association rules, the present paper proposes an algorithm for mining quantitative association rules (DCSA-QAR). The main contributions of this paper are summarized as follows: first, a new discretization method for numerical attributes is proposed. Then, the transition from continuous to a discrete version of CSA is done by adopting the crow position encoding to association rules and defining new operators to ensure any position update in the search area and increases the ability to explore the searching space.

In order to evaluate the performance of DSCA-QAR, we have performed an experimental study using ten real-world datasets. We have firstly compared our discretization algorithm with two classical approaches (fixed equal-width and entropy-based discretization). Second, we have compared the performance of our proposal with four mono and multi-objective evolutionary approaches. Finally, we have compared the results obtained through the application of particle swarm optimization approaches (Alatas & Akin, 2008), (Kuo, Gosumolo, & Zulvia, 2017) and (Heraguemi, Kamel, & Drias, 2016) with our approach result.

The remainder of this paper is organized as follows. In Section 2, a summary of the benchmark algorithms for mining quantitative association rules is presented. Section 3 provides a brief study of the standard version of the crow search algorithm (CSA). Moreover, this section introduces a basic definition of quantitative association rules (QARs) and some quality measures. Section 4 details the description of our discrete DCSA version for mining QARs. Section 5 reports and discusses the obtained results over ten real-world datasets. Finally, Section 6 presents some relevant conclusions and future works.

BACKGROUND

The problem of mining quantitative association rule was firstly introduced in (Srikant & Agrawal, 1996). They proposed a data discretization approach to discover frequent rules by partitioning numerical attributes values into fine intervals and then combining adjacent intervals as necessary before performing the Apriori algorithm to generate association rules. The FI-QAR proposed in (Zhang, Pedrycz, & Huang, 2018) employed a clustering approach for discretization and an improved Apriori algorithm for mining quantitative association rules. The fuzzy inference is realized by combing the TS fuzzy model with the mined quantitative association fuzzy rules. In (Brussel, Müller, & Goethals, 2016), the authors focused on overlapping quantitative association rules by proposing a method for selecting relevant attributes that show a dependency with an interval from a different dimension, detect relationships of dense intervals in these attributes, and finally combine them into quantitative association rules.

In order to perform a global search in place of the candidate generation, many evolutionary algorithms are proposed for mining QARs. Alataş & Akin (2006) proposed a genetic algorithm for mining both positive and negative quantitative association rules. Each chromosome consists of genes that represent the items and intervals. Each gene is characterized by four parts that represent the antecedent or consequent of the rule, the positive or negative ARs, the lower and the upper bounds of the item interval. A uniform operator is proposed to ensure genetic diversity, and an adjusted fitness function is used for mining all interesting ARs from the last population. In (Kabir, Xu, Kang, & Zhao, 2017), multiple seeds based genetic algorithm (MSGA) is proposed for discovering boolean association rules in order to increase the diversity in the initial population. In (Martín, Alcalá-Fdez, Rosete, & Herrera, 2016), a new Niching Genetic Algorithm (NICGAR) is proposed to obtain a reduced set of different QARs based on the existence of an external solution pool that contains the best rule of each niche found in the search process according to several quality measures.

In addition to genetic algorithms, many meta-heuristic optimization methods based on swarm intelligence and other bio-inspired approaches are proposed to extract association rules. In (Alatas & Akin, 2008), the rough particle swarm optimization algorithm (RPSOA) is proposed. This algorithm is based on the notion of rough patterns that use rough values defined with upper and lower intervals that represent a range of values. In (Heraguemi, Kamel, & Drias, 2016), the authors proposed a multi-swarm cooperative bat algorithm for mining association rules MSB-ARM inspired from bat behaviors. Each rule is represented in integer encoding format and evaluated with a simple objective function based on support and confidence. Yan, Zhao, Lin, & Bai (2019) proposed the use of a parallel PSO algorithm for mining QAR and take the association rules mining as a multi-objective optimization problem by using four measures to evaluate the quality of each particle including support, confidence, comprehensibility, and interestingness. Three parts encode each attribute in the particle: the marked field that indicates which part of the rule the current attribute belongs to: the antecedent part, the consequent part, or none of them, the lower bound and the upper bound values of that attribute in the dataset.

Recently, some researchers have modeled the mining process of QAR as a multi-objective problem. The authors in (Ghosh & Nath, 2004) proposed a Pareto based genetic algorithm to extract QARs using support, comprehensibility and interestingness measures as the objectives of rule mining problem. In (Alatas, Akin, & Karci, 2008), the authors presented a multi-objective Pareto-based evolutionary algorithm formulated as a four objective optimization problem that maximizes support, confidence value and the comprehensibility of the rule, and minimizes the amplitude of the intervals. Finally, Martín, Rosete Suárez, Alcala-Fdez, & Herrera (2014) proposed a multi-objective evolutionary algorithm for mining both positive and negative QARs (MOPNAR) maximizing three objectives: comprehensibility, interestingness and performance.

Lately, a novel metaheuristic technique known as crow search algorithm (CSA) was proposed in (Askarzadeh, 2016). Based on the intelligent behaviors of crows, CSA provides very promising results in comparison with other metaheuristic techniques. Offering a high degree of diversification in combination with ease of implementation, CSA can lead us to obtain interesting association rules in terms of reliability and diversity. Consequently, we have seen the usefulness of extending the CSA to mining quantitative association rules in order to improve the performance of the obtained rules.

PRELIMINARIES

This section provides a brief study of the standard version of the crow search algorithm (CSA). Then a basic description of quantitative association rules (QAR) including some definitions and quality measures is presented.

Crow Search Algorithm (CSA)

CSA was firstly proposed in their continuous version by (Askarzadeh, 2016). It is a new nature-inspired meta-heuristic algorithm mimicking the crow search mechanism for hiding their food. Crows are part of the family of corvids, birds known for their exceptional intelligence. Crows are also characterized by a social behavior similar to that of humans. It can store food in anticipation of future periods of scarcity, instead of eating at once. Crows are called intelligent thieves because they take additional safety measures to hide their food and stealing other's food. Based on the crow's behavior (Askarzadeh, 2016), there are four principles of CSA:

- They live in the flock of several crows;
- They store their excess food in hiding places;
- They follow each other to steal other's food;
- They protect their cache's food from being pilfered stochastically.

The CSA evolutionary process emulates the mechanism conducted by the flock of N crows, when each crow hiding their extra food and searching the hiding food of other crows. CSA algorithm is computed in several iterations (it_{max}). In each iteration, the position of a crow *i* represents a possible solution of the optimization problem. For a *d*-dimensional problem, the position vector of i^{th} crow is presented by the equation (1):

$$X_i^{it} = \left[x_1^{it}, x_2^{it}, \ldots, x_d^{it}\right] \tag{1}$$

where $it = 1, 2, \ldots, it_{\max}$, $i = 1, 2, \ldots, N$, and d is the number of dimensions.

Each crow is equipped with a memory stocking its best-visited position. This fact is presented via the equation (2):

$$M_i^{it} = \left[m_1^{it}, m_2^{it}, \ldots, m_d^{it}\right] \tag{2}$$

Crows move in the search space to detect the better food sources (hiding places). In each iteration it, the crow i decides to follow the crow *j* (randomly selected from the flock) to access to their best hiding place (M_i^{it}). The current position of the crow *i* is updated according to two behaviors: pursuit and evasion.

In the case of pursuit, crow *j* does not identify that crow *i* follows it. So, the crow *i* will come close to the hiding place of crow *j*. For the evasion case, the crow *j* knows that it is being followed by the crow *i*. So, the crow *j* will fool crow *i* by moving to a new random position in the search space. Using these cases, the position of the crow i is updated according to the equation (3):

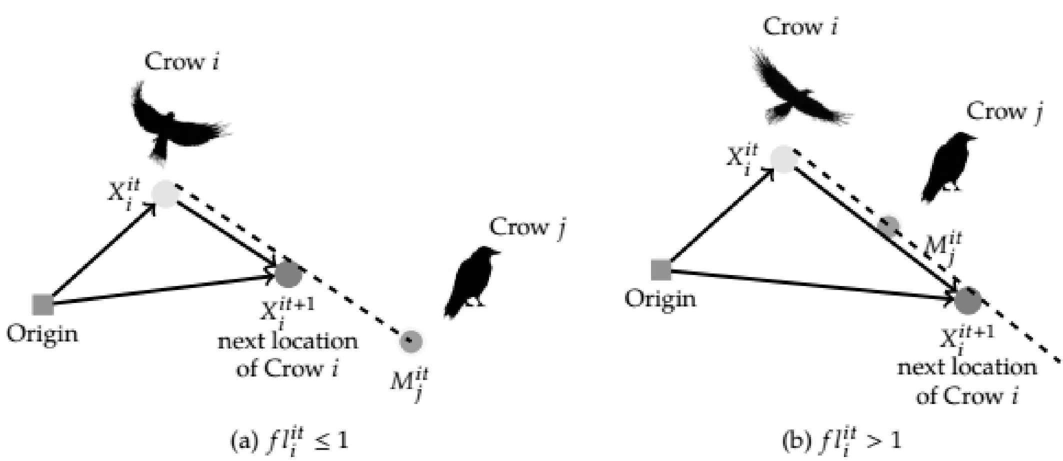
$$X_i^{(it+1)} = \begin{cases} X_i^{it} + r_i \times fl_i^{it} \times \left(M_j^{it} - X_i^{it}\right), if \ r_j \geq AP_j^{it} \\ a \ random \ position, \ Otherwise \end{cases} \tag{3}$$

where:

- r_i and r_j are random numbers with uniform distribution between 0 and 1.
- fl_i^{it} is the flight length of the crow *i* at the iteration *it*, it indicates the magnitude of movement from the current position of the crow *i* towards the best memory position of the crow *j*. By selecting a small value of fl_i^{it}, the algorithm performs a local search around their current position. On the other hand, by selecting large values, the algorithm performs a global search (far from their current position). Figure 1 shows the effect of the parameter f li on the search capability of crow *i*.
- The parameter AP_j^{it} denotes the awareness probability of crow *j* at the iteration *it*, this parameter plays an important role in the search mechanism. Small values of *AP* increase intensification while large values of *AP* increase diversification.

In CSA, the positions of all crows are randomly initialized. Once the crow's position is updated, the new position is evaluated using an objective function to update the memory vector, as reflected in equation (4):

Figure 1. The effect of the parameter fl_i^{it} on the search capability of crow i



$$M_i^{(it+1)} = \begin{cases} X_i^{(it+1)}, \ if \ F\left(X_i^{(it+1)}\right) is better than F\left(M_i^{it}\right) \\ M_i^{it}, \ Otherwise \end{cases} \tag{4}$$

where, $F\left(X\right)$ represents the objective function to be optimized.

Quantitative Association Rules

Notations

In this paper, we used the notations which are stated as below:

- $I = \left\{i_1, i_2, \ldots, i_n\right\}$ represents a set of items.
- $i_k \left(k \in \left[1, n\right]\right)$ represents each item in *I*.
- $D = \left\{t_1, t_2, \ldots, t_N\right\}$ represents a transaction database.
- $\left|D\right|$ represents the total number of transactions in *D*.
- $t_i \left(i \in \left[1, N\right]\right) \subseteq I$ represents each transaction in *D*.
- Each subset of *I* is called an itemset.
- *A* and *C* represent two itemsets.

Association Rules and Quantitative Association Rules

Definition 1: Firstly introduced by Agrawal, Imieliński, & Swami (1993), an association rule can be defined as $A \Rightarrow C$ where $A, C \subset I$ and $A \cap C \neq \varnothing$. *A* is called antecedent or left side of the rule and *C* is called consequent or right side of the rule.

Definition 2: Quantitative association rule (QAR) refers to a special type of association rules in the form of A $\Rightarrow C$, where *A* and *C* are a set of numerical and/or categorical attributes.

In contrast to general association rules where the antecedent and the consequent of the rule should be categorical (nominal or discrete) attributes, at least one attribute of the quantitative association rule (antecedent or conclusion) must include a numerical attribute (Zhu, 2009). In this context, each item is a pair attribute-interval, and *A* and *C* are formed by a conjunction of multiple boolean expressions of the form $t_i \in \left[v_1, v_2\right]$. For example, a QAR could be:

$$Age \in \left[30, 45\right] and Salary \in \left[4K, 5K\right] \Rightarrow NumTVs \in \left[2, 3\right]$$

Support and Quality Measures

Definition 3: The support of an itemset *l* refers to the number of transactions that contain *l* divided by the total number of transactions in *D*, denoted as *Sup(l),* It's defined as:

$$Sup\left(l\right) = \left|t \ / t \in D and l \subseteq t\right| \big/ \left|D\right|$$

Based on the support of an itemset, many measures are proposed to assess the quality of rules in order to evaluate the results obtained by any method for mining association rules. Table 1 shows many popular quality measures used to evaluate QAR (Martínez-Ballesteros, Martínez-Álvarez, Troncoso, & Riquelme, 2014). Each measure is described by its mathematical definition, a short description, and its range of variability.

DCSA-QAR: NEW DISCRETE CROW SEARCH ALGORITHM FOR MINING QUANTITATIVE ASSOCIATION RULES

In this section, we present a new discrete version of CSA as a search algorithm for mining quantitative association rules, called DCSA-QAR. The next sub-sections show a detailed description of our discrete DCSA version of the standard CSA algorithm. The pseudo-code of DCSA-QAR is given in Algorithm 3 (Figure 7).

Discretization Step

Traditionally, the discretization of numerical attributes can be achieved through two approaches (Zhu, 2009):

- **Equal-width discretization:** Each numerical attribute range is divided into *N* intervals of equal width.

Table 1. Some quality measures for quantitative association rules

Measures	Equation	Description	Range
$Sup(A)$	$n(A)/N$	Coverage of A.	[0,1]
$Sup(A \Rightarrow C)$	$n(A \cap C)/N$	Generality of the rule.	[0,1]
$Conf(A \Rightarrow C)$	$Sup(A \Rightarrow C)/Sup(A)$	Reliability of the rule	[0,1]
$Lift(A \Rightarrow C)$	$Sup(A \Rightarrow C)/Sup(A) \times Sup(C)$	Interest of the rule (Dependence between A and C): • Negative dependence if Value < 1. • Independence if Value = 1. • Positive dependence if Value > 1.	[0, +∞ [
$Conviction(A \Rightarrow C)$	$1 - Sup(C)/1 - Conf(A \Rightarrow C)$	Implication of the rule (Dependence between A and ¬C): • Negative dependence if Value < 1. • Independence if Value = 1. • Positive dependence if Value > 1.	[0, +∞ [
$Gain(A \Rightarrow C)$	$Conf(A \Rightarrow C) - Sup(C)$	Change of support or added value	[– 0.5,1]
$Leverage(A \Rightarrow C)$	$Conf(A \Rightarrow C) - (Sup(A) \times Sup(C))$	Change of support or added value	[– 0.25,0.25]

- **Equal-depth discretization:** Each numerical attribute range is divided into N intervals such that each interval equally contains $1/N$ of the total examples.

These methods can be easily implemented, but there is a drawback since the resulting intervals may be incoherent or unbalanced. Also, they discretize each attribute separately ignoring any dependency or correlation between attributes (Lud & Widmer, 2000).

In this paper, we propose a new algorithm C-BUDA (Confidence Based Unsupervised Discretization Algorithm) inspired from some research in the field of classification that attempts to create the best split for a numerical attribute interval that maximizes the prediction of the class (Fayyad & Irani, 1993). The pseudo-code of C-BUDA is given in Algorithm 1 (Figure 2).

Given a target numerical attribute to discretize and a predefined maximum interval number threshold Nb_{max}. First, we apply an Equal-width discretization for all other attributes with $Nb_{Int}=2$ to obtain, in a way, a class attribute. Then, for each possible point split ($Nb=3$ *to* Nb_{max}), we also apply an Equal-width discretization for the target attribute with ($Nb_{Int}=Nb$), and we calculate the confidence's average Avg_{Conf} of all rules $A \Rightarrow C$, where A is the target attribute and C is the class attribute. Finally, we retain the number of intervals that maximize the confidence average Avg_{Conf}.

Crow Position Encoding

DCSA-QAR uses the discrete-based encoding to represent the association rules. With this schema, the position X_i of each crow will simply be a vector that provides discrete numbers representing the one association rule. In this paper, we represent the flock as a set of P crows. Each crow is determined with its current and memory positions. Each crow position X_i and crow memory M_i $(i = 1,2,\ldots,P)$, characterized by $2\times N$ elements, where N is the number of features in the dataset.

In DCSA-QAR, each particle consists of two vectors that represent the control and parametric attributes. Control attributes C_a can have three values: *0* when the feature does not belong to the rule, and *1* or *-1* if it belongs to the antecedent or the consequent of the rule, respectively. In contrast, the

Figure 2. C-BUDA: Confidence Based Unsupervised Discretization Algorithm

```
Input: M is a dataset, t is a target attribute
       Nb_max maximum interval number threshold
Output: Nb is the best splitting interval number;
Description:
for ( Each attribute a in M and a ≠ t ) do
  | Discretize(a,M,2);
end
Best = 0;
Max_C = 0;
for ( Nb = 3; Nb ≤ Nb_max; Nb + + ) do
  | Discretize(t,M,Nb);
  | for ( Each attribute a in M and a ≠ t ) do
  |   | Calculate Avg_C the Confidence average of rules (t ⇒ a);
  |   | Sum = Sum + Avg_C;
  | end
  | Avg = Sum/dim(M);
  | if (Max_C < Avg then
  |   | Best = Nb;
  | end
end
return Best;
```

value of each parametric attribute P_a is an integer representing the value of their corresponding feature. A representation of the crow position and memory are respectively shown in Figure 3 and Figure 4.

Flock Initialization

With the aim of initializing the population, a random uniform initialization is used, in which the crows of the initial flock are randomly scattered in the search space. After generating the initial crow's positions of flock, the initial memory position of each crow was assigned to its current position.

The process that generates the initial population is listed below:

- First, the indexes of attributes which can belong to the rule are randomly selected according to a maximum number of attributes given by the user.
- For each selected index, the control attribute is randomly selected from values 1, −1 and 0.
- Then, the parametric attributes are randomly generated for each selected attribute.
- Finally, the new position is accepted if it is fully compliant with the user requirements (minimum number of attributes in the antecedent and consequent of the rule, minimum support of the rule).

This process is repeated until the reach of the number of positions that compose the initial population.

Fitness Evaluation

In association rules mining, the support and the confidence measures are widely used to evaluate the quality of the obtained rules. However, the optimization of only one measure is not sufficient in most cases, and an adequate combination of some of them is required. In particular, the support of the rule, confidence, gain and accuracy may be the measures that can summarize all the considered measures (Martínez-Ballesteros, Martínez-Álvarez, Troncoso, & Riquelme, 2014).

In DCSA-QAR, the fitness function to be maximized is given by the following equation:

$$F\left(Rule\right) = w_s \times sup + w_c \times conf + w_g \times gain \tag{5}$$

Figure 3. Crow position structure with N=9

$\mathbf{X_i^{it}}$	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
C_a	0	−1	1	0	−1	1	1	−1	1
P_a	5	2	3	8	4	2	1	6	3

Figure 4. Crow memory structure with N=9

$\mathbf{M_i^{it}}$	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
C_a	1	−1	1	0	−1	1	−1	1	0
P_a	6	1	2	4	2	5	1	3	4

where, *sup* is the support of the rule, *conf* is the confidence of the rule, *gain* is the gain measure of the rule, and w_s, w_c and w_r are weights given by user, and $w_s + w_c + w_r = 1$.

Searching a New Position

In the continuous version of the CSA, the new position is calculated as has been shown in Equation (3). In our proposed algorithm, each crow in the flock represents a possible solution in the form of an association rule.

Regarding the parameters of the classic CSA, which are r_i and fl_i^{it}, where, fl_i^{it} denotes the flight length of crow *i* at the iteration *it*. The philosophy of the fl_i^{it} parameter is that the small values of fl leading to the local search (at the vicinity of x_i) and large values results in global search. Additionally, the crow *i* search for a step of moving to the new position.

The equation (3) is modified as shown in equations (6) and (7):

$$X_i^{it+1} = X_i^{it} + S_i^{it} \tag{6}$$

where:

$$S_i^{it} = fl_i^{it} \times \left(M_j^{it} - X_i^{it}\right) \tag{7}$$

Intending to adapt the algorithm as accurately as possible, we have considered appropriate to relate with the displacement step of the crow *i* to the hiding position of the crow *j*. In this case, we can deduct from the formula 6 and 7 that the new position of a crow *i* depends on its current position and the displacement step to displace to the hiding position of a crow *j*. However, this formula cannot be applied directly to the discrete optimization problem. For this reason, we have introduced a new replacement operator, the equations (6) and (7) are modified as shown in equations (8) and (9):

$$X_i^{it+1} = \rhd\lhd \left(X_i^{it}, M_j^{it}, \mathrm{int}\left(S_i^{it}\right)\right) \tag{8}$$

where, $\rhd\lhd$ means that the crow *i* execute $int\left(S_i^{it}\right)$ times the *randomly* replacement operation (presented in the Algorithm (2)) between the vectors X_i^{it}, M_j^{it}. The function $int\left(a\right)$ calculates the integer value of a given real *a*. The number S_i^{it} is calculated as in equation (9):

$$S_i^{it} = fl_i^{it} \times nonZHD\left(M_j^{it}, X_i^{it}\right) \tag{9}$$

where, the $nonZHD\left(M_j^{it}, X_i^{it}\right)$ function calculate the non-zero Hamming distance between two vectors M_j^{it} and X_i^{it}, and fl_i^{it} is the flight length in [0,1].

It was noted that the parameter r_i has not been taken into account in the equation (7) directly, but it is interpreted as a random choice of attribute index to be replaced.

Based on the hamming distance [29], we define the non-Zero Hamming Distance between two position or memory vectors *X* and *Y*, denoted by $nonZHD\left(X, Y\right)$, as the number of positions where the symbols x_i and y_i are different and their control attributes $C_a\left(x_i\right)$ and $C_a\left(y_i\right)$ are not nulls at the same time. That is:

$$nonZHD\left(X,Y\right)=\sum_{i=1}^{d}\delta\left(x_i,y_i\right) \tag{10}$$

where:

$$\delta\left(x_i,y_i\right)=\begin{cases}0,\ if\left(x_i=y_i\right)\vee\left(C_a\left(x_i\right)=0\wedge C_a\left(y_i\right)=0\right)\\ 1,\ otherwise\end{cases} \tag{11}$$

An illustrative example of the replacement operator is given in Figure. 6, and its pseudo code is given in Algorithm 2 (Figure 5).

At each generation, each crow executes $int\left(S_i^{it}\right)$ times the replacement of randomly two elements with the same index between its current position vector and the hiding position vector of a selected crow *j* for generating a vector between these two vectors. Thereby, S_i^{it} is a real number calculated by the equation (9).

Then, the position's update equation of each crow is modified by equation (12):

$$X_i^{it+1}=\begin{cases}\triangleright\triangleleft\left(X_i^{it},M_j^{it},int\left(S_i^{it}\right)\right),\ if\left(\ r_j\geq AP_j^{it}\right)\\ a\,random\,position,\ otherwise\end{cases} \tag{12}$$

where, X_i^{it+1} indicates the updated position of the crow *i* in iteration *it + 1*, M_j^{it} denotes the memorized position of the crow *j* in iteration *it*, S_i^{it} as calculated in equation (9), r_j is a random number in $\left[0,1\right]$, and AP_j^{it} is the awareness probability of a crow *j* at iteration *it*.

PERFORMANCE STUDIES

In this section, we present and discuss the experiments carried out to evaluate the performance of our proposal. First, we describe the datasets used for testing algorithms. Then, we compare the

Figure 5. Replacement Operator Algorithm

```
Input: X, M are vectors
       S is an integer
Output: Y is a vector;
Description:
Y=X;
for (i = 1; i ≤ S ; i + +) do
    j = rand(1; N);
    Y[1, j] = M[1, j];
    Y[2, j] = M[2, j];
end
return Y;
```

Figure 6. An illustrative example of performing replacement operator (⋈)

$Fl_i^{it} = 0.6$

$nonZHD(X_i^{it}, M_j^{it}) = 7$

$\mathbf{X_i^{it}}$	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
C_a	0	1	1	0	−1	1	1	−1	1
P_a	5	2	3	8	4	2	1	6	3

$S_i^{it} = 0.6 * 7 = 4.2$
$int(S_i^{it}) = 4$

Indexes chosen randomly

$\mathbf{M_j^{it}}$	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
C_a	1	−1	1	0	−1	1	−1	1	0
P_a	6	1	2	4	4	5	1	3	4

After a number of $int(S_i^{it})$ replacements for indexes (1, 3, 6 and 9) chosen randomly, the new position X_i^{it+1} is obtained:

$\mathbf{X_i^{it+1}}$	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
C_a	1	1	1	0	−1	1	1	−1	0
P_a	6	2	2	8	4	5	1	6	4

C-BUDA approach with two basic methods: fixed equal-width and entropy-based discretization. Finally, we compare the performance of DCSA-QAR to other state-of-art algorithms: mono and multi-objective evolutionary approaches (MOPNAR (Martín, Rosete-Suárez, Alcala-Fdez, & Herrera, 2014), MODENAR (Alatas, Akin, & Karci, 2008), MOEA-Ghosh (Ghosh & Nath, 2004) and NICGAR (Martín, Alcalá-Fdez, Rosete, & Herrera, 2016)), and particle swarm optimization approaches (MOPSO (Kuo, Gosumolo, & Zulvia, 2017), RPSOA (Alatas & Akin, Rough particle swarm optimization and its applications in data mining, 2008) and MSB-ARM (Heraguemi, Kamel, & Drias, 2016).

Datasets

We have considered the public real datasets from BUFA repository (Altay Guvenir & Uysal, 2000) to evaluate the performance of our proposed algorithms, in which the number of examples varies between 40 and 40; 768, and the number of attributes varies between 5 and 41. The abbreviation, The number of attributes and examples of the ten datasets are summarized in Table 2. On the one hand, the first ten datasets are used for comparison between DSCA-QAR and evolutionary approaches. On the other hand, Basketball, Bodyfat and Quak datasets are only considered for comparison with particle swarm optimization approaches.

Performance of DSCA-QAR Using Different Discretization Approaches

In this sub-section, we study the performance of C-BUDA against the classical approaches (fixed equal-width and entropy-based discretization) in order to analyze the performance of the new discretization method introduced through the DCSAQAR algorithm.

Figure 7. DCSA-QAR Algorithm

```
Input: P, the population size
       AP, the awareness probability
       fl, the flight length
       t_max is the number of iterations
Output: M the vector of solutions (set of association rules);
Description:
Initialize the crow's positions randomly;
Evaluate the fitness function of all crow positions;
Initialize the memory position M of all crows as these initial positions;
for t = 1 to t_max do
    for i = 1 to P do
        Choose a random index j;
        Generate a random number R in [0,1];
        if (R ≥ AP) then
            S = fl × nonZHD(X_i^t, M_j^t);
            X_i^{t+1} =⋈ (X_i^t, M_j^t, S);
        else
            X_i^{t+1} =a random position from the search space;
        end
        Check the feasibility of the solution X_i^(t + 1);
        Evaluate the new position using the fitness function F(X_i^{t+1});
        if (F(X_i^{t+1}) > F(M_i^t) then
            Updates the crow's memory M_i^{t+1} = X_i^{t+1};
        end
    end
end
return M;
```

In the entropy-based discretization, the number of intervals of each attribute is determined in the same way as C-BUDA by replacing the confidence measure with the information gain that calculates the expected reduction in entropy for each attribute interval splitting. Whereas in the fixed equal-width approach, the number of attribute intervals is fixed for all attributes.

The average results obtained during the ten executions regarding the three approaches are summarized in Tables 3 with DCSA-QAR parameters: $Pop_{Size} = 100$, $N_{Eval} = 100$, $P_{awar} = 0.2$, $Flight_l = 2$, $Supp_{min} = 0.1$, $Conf_{min} = 0.3$, $W_{Supp} = 0.1$, $W_{Conf} = 0.8$, $W_{Gain} = 0.1$ and $Nb_{IntMax} = 15$ for three approaches. Table3 shows the average of covered records, support, confidence, gain, lift, and the average number of attributes for the rules obtained by the different approaches for all datasets. It can be noted that C-BUDA outperforms fixed equal-width and entropy-based approaches in all measures expect the diversity and rule size measures.

Finally, the Friedman test (Siegel & Castellan, 1988) has been applied to detect if there are significant differences in the measures obtained by C-BUDA and other discretization approaches. Table 4 shows the average rankings for each quality measure obtained by the Friedman test. In contrast, Table 5 demonstrates the Holm (Holm, 1979) and Finner (Finner, 1993) post hoc statistical tests performed to find significant differences among the results obtained for each approach. It can be observed that C-BUDA is the control approach returning the best average ranking in most cases, except for the number of attributes and diversity measures. For the rule size, there is no significant difference between C-BUDA and fixed Equal-W approach.

Table 2. Public datasets used for the experiments

Names	Abbreviation	#Attributes	#Examples
Ailerons	AI	41	7154
Basketball	BA	5	96
Bodyfat	FA	18	252
Bolts	BO	8	40
Computer activity	CA	22	8192
College	CO	21	236
Elevators	EV	19	16,599
Fried	FR	11	40,768
Pollution	PO	16	60
Treasury	TR	17	1049
Weather Ankara	WA	11	1609
Quak	QU	4	2178

Table 3. Results of the average value of the measures for all datasets obtained by discretization approaches

Measure	Discretization approach		
	Fixed Equal-W	Entropy-based	C-BUDA
Diversity	**0.997**	0.892	0.977
Coverage	0.276	0.327	**0.964**
Support	0.034	0.046	**0.134**
Confidence	0.091	0.142	**0.516**
Gain	0.046	0.075	**0.193**
Lift	2.627	3.354	**8.635**
Rule size	**4.384**	3.761	3.868

Table 4. Average rankings of the discretization approaches for each quality measure obtained by Friedman test

Measure	Discretization approach		
	Fixed Equal-W	Entropy-based	C-BUDA
Support	2.75	2.25	**1.00**
Confidence	2.65	2.35	**1.00**
Coverage	2.85	2.10	**1.05**
Gain	2.55	2.25	**1.20**
Lift	2.55	1.85	**1.60**
Size	**1.45**	2.05	2.50
Diversity	**1.70**	2.00	2.30

Table 5. Post Hoc comparison Table for $\alpha = 0.05$ for each quality measure for all discretization approaches

Measure	*i*	Approach	*z*	*p*	Holm	Finnner	H_0 Rejected
Confidence	*Control app.: **C-BUDA***						
	1	Fixed Equal-W	3.6895	0.0002	0.0250	0.0253	√
	2	Entropy-based	3.0187	0.0025	0.0500	0.0500	√
Support	*Control app: **C-BUDA***						
	1	Fixed Equal-W	3.9131	0.0001	0.0250	0.0253	√
	2	Entropy-based	2.7951	0.0052	0.0500	0.0500	√
Diversity	*Control app: **Fixed Equal-W***						
	1	Entropy-based	0.6708	0.5023	0.0500	0.0500	√
	2	*C-BUDA*	1.3416	0.1797	0.0250	0.0253	√
Coverage	*Control app: **C-BUDA***						
	1	Fixed Equal-W	4.0249	0.0001	0.0250	0.0253	√
	2	Entropy-based	2.3479	0.0188	0.0500	0.0500	√
Gain	*Control app: **C-BUDA***						
	1	Fixed Equal-W	3.0187	0.0025	0.0250	0.0253	√
	2	Entropy-based	2.3479	0.0189	0.0500	0.0500	√
Lift	*Control app: **C-BUDA***						
	1	Fixed Equal-W	2.1243	0.0337	0.0250	0.0253	√
	2	Entropy-based	0.5590	0.5761	0.0500	0.0500	√
Size	*Control app: **Fixed Equal-W***						
	1	C-BUDA	2.3479	0.0189	0.0250	0.0253	
	2	Entropy-based	1.3416	0.1797	0.0500	0.0500	√

Parameters of DSCA-QAR and Other Algorithms Considered for Comparison

Table 6 presents the main parameter values of DSCA-QAR and each algorithm considered for comparison. DSCA-QAR is compared to NICGAR (Martín, Alcalá-Fdez, Rosete, & Herrera, 2016), MOPNAR (Martín, Rosete-Suárez, Alcala-Fdez, & Herrera, 2014), MODENAR (Alatas, Akin, & Karci, 2008), and MOEA-Ghosh (Ghosh & Nath, 2004). It is noted that all of these algorithms are available in the KEEL software tool (Alcala-Fdez, et al., 2009), and the parameter settings of these algorithms appear as default parameter values according to the instructions of the authors of each algorithm. The parameters of DSCA-QAR are based on the suggestions in the corresponding article (Askarzadeh, 2016).

Comparison With Mono and Multi-Objective Evolutionary Approaches

In this sub-section, we analyze the performance of DCSA-QAR in comparison with four mono and multi-objective evolutionary approaches: NICGAR and MOPNAR for mining positive and negative QARs, as well as MODENAR and MOEA-Ghosh for mining positive QARs. The results of the average value of the measures obtained for all datasets are summarized in Table 7. The average rankings of the analyzed algorithms for each quality measure obtained by Friedman test are shown in Table 8. The results obtained by the Holm and Finner post hoc statistical tests for each quality measure are summarized in Table 9. The best value for each quality measure and each algorithm is highlighted

Table 6. Parameters used for the comparison between algorithms

Algorithm	Parameter values
MOEA-Ghosh	$Pop_{Size} = 100$, $N_{Eval} = 50000$, $N_{Obj} = 3$, $Cross_{Pt} = 2$, $P_{Mut} = 0.02$, $Amp_f = 2$
NICGAR	$Pop_{Size} = 100$, $N_{Eval} = 50000$, $Ampl_f = 2$, $P_{Mut} = 0.1$, $Nich_{min} = 0.5$, $Quality_{min} = 0.85$, $Update = 5.0$
MODENAR	$Pop_{Size} = 100$, $N_{Eval} = 50000$, $Cross_R = 0.3$, $Sol_{min} = 60$, $Ampl_f = 2$, $W_{Supp} = 0.8$, $W_{Conf} = 0.2$, $W_{Comp} = 0.1$, $W_{Amp} = 0.4$
MOPNAR	$N_{Eval} = 50000$, $N_{Obj} = 3$, $W_{neighbor} = 10$, $Diff_{min} = 5.0$, $P_{Cross} = 0.8$, $Max_{child} = 2$, $P_{Mut} = 0.1$, $Amp_f = 2$, $P_{parent} = 0.9$, $H_{control} = 13$
DCSA-QAR	$Pop_{Size} = 100$, $N_{Eval} = 100$, $P_{awar} = 0.2$, $Flight_l = 2$, $Supp_{min} = 0.1$, $Conf_{min} = 0.3$, $W_{Supp} = 0.1$, $W_{Conf} = 0.8$, $W_{Gain} = 0.1$

Table 7. Results of the average value of the measures for all datasets

Algorithms	$\# R$	Av_{Sup}	Av_{Conf}	Av_{Lift}	Av_{Conv}	Av_{Size}	$Av_{Coverage}$
MOPNAR	82.98	0.275	0.946	18.220	∞	2.599	**0.998**
MOEA-Ghosh	**315.8**	**0.435**	0.772	77.993	∞	**11.866**	0.701
MODENAR	45.3	0.394	0.838	**157.327**	∞	8.942	0.800
NICGAR	19.6	0.190	**0.952**	5.537	∞	2.028	0.924
DCSA-QAR	74.9	0.376	0.938	2.651	∞	2.790	0.969

Table 8. Average rankings of the algorithms for each quality measure obtained by Friedman test

Algorithms	Support	Confidence	Coverage	Gain	Lift	Size
DCSA-QAR	2.8	**1.9**	2.20	3.3	4.4	3.2
NICGAR	4.0	2.5	3.05	**1.1**	3.3	5.0
MODENAR	**2.3**	4.0	3.80	4.5	3.3	2.1
MOPNAR	3.2	2.5	**1.75**	2.1	2.1	3.6
MOEA-Ghosh	2.7	4.1	4.20	4.0	**1.9**	**1.1**

Table 9. Post Hoc comparison Table ($\alpha = 0.05$) for each quality measure for all algorithms

Measures	*i*	Algorithm	*z*	*p*	Holm	Finnner	H_0 Rejected
Support	Control alg.: **MODENAR**						
	1	MOEA-Ghosh	0.5657	0.5716	0.05	0.05	
	2	DCSA-QAR	0.7071	0.4795	0.025	0.0377	
	3	MOPNAR	1.2728	0.2031	0.0167	0.0253	
	4	NICGAR	2.4042	0.0162	0.0125	0.0127	√
Confidence	Control alg.: **DCSA-QAR**						
	1	MOPNAR	0.8485	0.3961	0.05	0.05	
	2	NICGAR	0.8485	0.3961	0.025	0.0377	√
	3	MODENAR	2.9698	0.003	0.01667	0.0253	√
	4	MOEA-Ghosh	3.1113	0.0019	0.0125	0.0127	√
Coverage	Control alg.: **MOPNAR**						
	1	DCSA-QAR	0.6364	0.5245	0.05	0.05	
	2	NICGAR	1.8385	0.066	0.025	0.0377	√
	3	MODENAR	2.8991	0.0037	0.01667	0.0253	√
	4	MOEA-Ghosh	3.4648	0.0005	0.0125	0.0127	√
Gain	Control alg.: **MODENAR**						
	1	MOPNAR	1.4142	0.1573	0.05	0.05	
	2	DCSA-QAR	3.1113	0.0019	0.025	0.0377	√
	3	MOEA-Ghosh	4.1012	0	0.0167	0.0253	√
	4	MODENAR	4.8083	0	0.0125	0.01274	√
Lift	Control alg.: **MOEA-Ghosh**						
	1	MOPNAR	0.2828	0.7773	0.05	0.05	
	2	NICGAR	1.9799	0.0477	0.025	0.0377	
	3	MODENAR	1.9799	0.0477	0.01667	0.0253	√
	4	DCSA-QAR	3.5355	0.0004	0.0125	0.0127	√
Diversity	Control alg.: **MOPNAR**						
	1	DCSA-QAR	0.8485	0.3961	0.05	0.05	
	2	MODENAR	2.2627	0.0236	0.025	0.0377	√
	3	MOEA-Ghosh	2.4042	0.0162	0.0167	0.0253	√
	4	NICGAR	4.3841	0	0.0125	0.0127	√

in bold. Notice that we have not considered the conviction measure because the algorithms reach infinity in most of the datasets.

Based on the analysis of the results shown in showcased tables, the following conclusions emerge:

- MOEA-Ghosh obtains the largest rule sets with the worst values regarding confidence measure and the lowest average coverage of the datasets.
- NICGAR provides the reduced sets (less than 20 rules on average) of PNQARs with the lowest average number of attributes and also an interesting average coverage of the datasets (higher

than 92% on average). Furthermore, the obtained rules show the best value for the confidence measure (higher than 95% on average).

- MOPNAR and DCSA-QAR provide reduced sets of rules (almost a third of the rules amount obtained by MOEA-Ghosh), with the best average coverage (higher than 97% on average). Moreover, the obtained rules show good values for the confidence measure that is very close to the best value obtained by NICGAR algorithm.

It can be observed that DCSA-QAR is the control algorithm for the confidence measure (reflecting the best ranking for the confidence measure). For other measures such as coverage of records and diversity, DCSA-QAR does not present significant differences in comparison with the control algorithm MOPNAR. According to the support measure, DCSA-QAR does not present important differences with the control algorithm MODENAR.

Finally, we can deduce that DSCA-QAR provides a reduced number of easily understandable and reliable rules (best ranking for the confidence measure, higher than 93%) with good coverage of the dataset (higher than 97%).

Comparison With Particle Swarm Optimization Approaches

In this sub-section, we analyze the performance of DCSA-QAR in comparison with three particle swarm optimization approaches: RPSOA (Alatas & Akin, 2008) and MOPSO (Kuo, Gosumolo, & Zulvia, 2017) using rough and multi-objective particle swarm optimization for mining numerical association rules respectively, and MSB-ARM (Heraguemi, Kamel, & Drias, 2016) using multi-swarm bat algorithm for mining QARs.

Since the algorithms are not available, we have reported the results obtained by each algorithm for the common dataset only, namely Basketball (BK), Bodyfat (FA), and Quak (QU) datasets. The comparative results summarized in Table 10 confirm the conclusions previously confirmed in sub-section 5.4 that DSCA-QAR provides a reduced set of rules with high confidence value (greater than 93 on average) and few attributes (less than 3 attributes per rule on average).

Table 10. Results of the average value of the measures for Basketball, Bodyfat and Quak datasets

Dataset	Algorithm	$\# R$	Av_{Sup}	Av_{Conf}	Av_{Size}
BK	MOPSO	41.0	**0.39**	0.89	**3.35**
	RPSOA	34.2	0.36	0.60	3.21
	MSB-ARM	**369.0**	0.34	**0.92**	NA
	DCSA-QAR	27.0	0.18	0.91	2.29
FA	MOPSO	65.0	0.37	0.92	5.18
	RPSOA	46.4	**0.65**	0.61	**6.94**
	MSB-ARM	NA	0.23	0.88	NA
	DCSA-QAR	**96.4**	0.26	**0.98**	3.16
QU	MOPSO	29.0	0.42	**0.93**	**2.8**
	RPSOA	46.4	0.39	0.63	2.22
	MSB-ARM	**93.0**	0.31	0.84	NA
	DCSA-QAR	12.0	**0.43**	0.91	2.06

Analysis of Scalability and Complexity

Several experiments have been carried out to analyze the scalability of the algorithms in the datasets Ailerons (AI) and Fried (FR), which present the highest number of attributes and examples, respectively, of the datasets used for the experiments (see Table 2). All experiments were performed on an Intel Core i5-4300U, 1.90 GHz CPU with 8 Gb of memory and running Windows operating system.

Table 11 shows the average runtime expended by the analyzed algorithms when the number of attributes increases on the dataset (AI). In contrast, Table 12 shows the average runtime spent by them when the number of examples increases on the dataset (FR). Moreover, Figures 8. and 9. show the relationship between the runtime expended by algorithms and the number of attributes and examples, respectively.

Analyzing the results shown in Tables 11 and 12, and Figures 8 and 9, we can see that the runtime of all algorithms scale almost linearly when the number of examples or the number of attributes in the dataset increases. Notice that DCSA-QAR expends a large amount of time for mining association rules because it needs an additional process to perform discretization step for each attribute.

CONCLUSION

In this paper, we have proposed DCSA-QAR, a new discrete crow search algorithm for mining quantitative association rules. We have extended the continuous version of CSA to mine QAR by using a discrete-based encoding to represent the association rules and adopting new operators to calculate the update position and increase the ability to explore the searching space.

The results obtained over ten real-world datasets have shown how DCSA-QAR provides reduced rule sets with an interesting trade-off between the different objectives, getting association rules with

Table 11. Average expended runtime (seconds) by algorithms on the dataset Ailerons with the increase of the number of attributes

Algorithms	Number of attributes				
	8	16	24	32	41
MOPNAR	**11.6**	**15.6**	**22.0**	**26.6**	**33.8**
MOEA-Ghosh	14.6	22.4	32.8	42.8	46.6
MODENAR	24.4	29.6	33.0	35.8	38.8
NICGAR	18.4	29.6	44.0	69.4	88.4
DCSA-QAR	12.6	19.6	23.2	32.0	35.2

Table 12. Average expended runtime (seconds) by algorithms on the dataset Fried with the increase of the number of examples

Algorithms	Number of examples				
	20%	40%	60%	80%	100%
MOPNAR	20.1	37.2	74.7	77.8	117.4
MOEA-Ghosh	**16.9**	**31.2**	**48.2**	**57.3**	**94.8**
MODENAR	38.0	69.4	139.5	146.3	221.1
NICGAR	51.7	108.4	166.5	219.1	331.0
DCSA-QAR	48.8	88.4	132.8	140.8	179.4

Figure 8. Relationship between the expended runtime by algorithms and the number of attributes on the dataset Ailerons

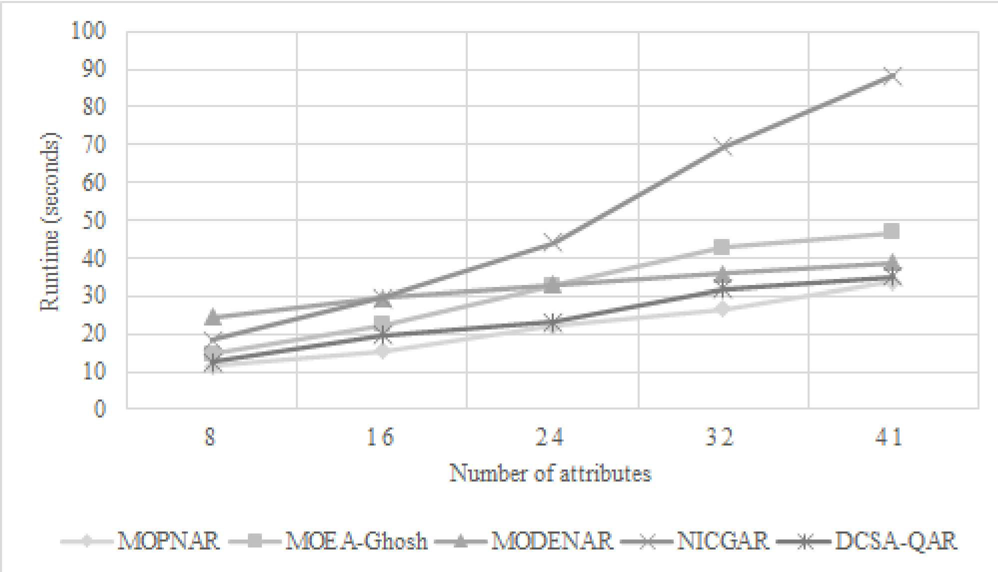
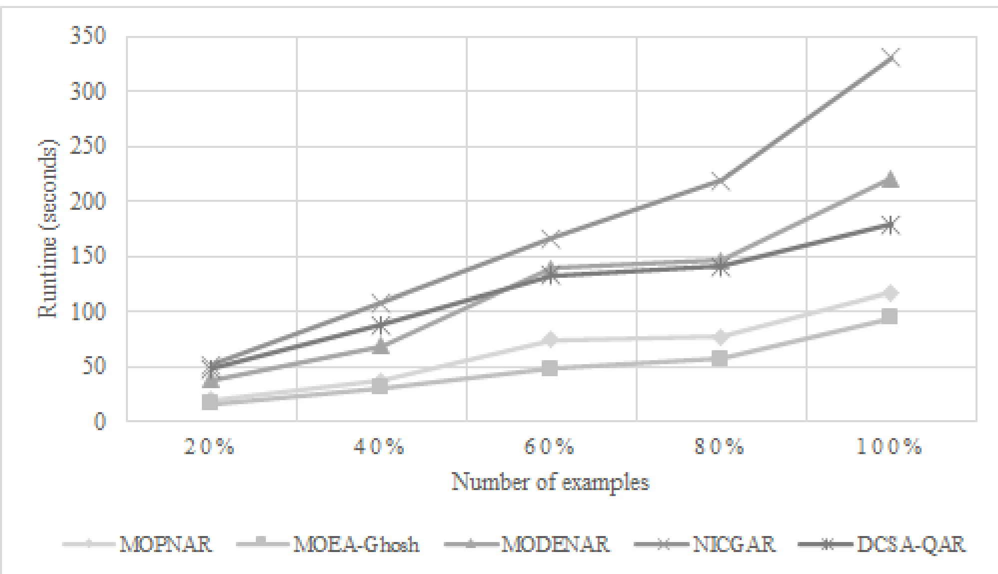


Figure 9. Relationship between the expended runtime by algorithms and the number of examples on the dataset Fried



better values for reliability measure, high coverage of the dataset and few attributes, making the rules relatively easy to understand.

In the future, we will attempt to extend this proposal to deal with gene expression data in order to infer gene regulatory networks. We also focus on the use of the continuous version of CSA to mine QARs without performing a previous discretization. In addition, we will propose a new hybrid optimization approach based on CSA with Genetic algorithms (GA) to improve the searchability and increase the exploration of the solution space.

REFERENCES

Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining Association Rules Between Sets of Items in Large Databases. *SIGMOD Record*, *22*, 207–216. doi:10.1145/170036.170072

Ai, D., Pan, H., Li, X., Gao, Y., & He, D. (2018). Association rule mining algorithms on high-dimensional datasets. *Artificial Life and Robotics*, *23*(3), 420–427. doi:10.1007/s10015-018-0437-y

Alataş, B., & Akin, E. (2006). An efficient genetic algorithm for automated mining of both positive and negative quantitative association rules. *Soft Computing*, *10*(3), 230–237. doi:10.1007/s00500-005-0476-x

Alatas, B., & Akin, E. (2008). Rough particle swarm optimization and its applications in data mining. *Soft Computing*, *12*(12), 1205–1218. doi:10.1007/s00500-008-0284-1

Alatas, B., Akin, E., & Karci, A. (2008). MODENAR: Multi-objective differential evolution algorithm for mining numeric association rules. *Applied Soft Computing*, *8*(1), 646–656. doi:10.1016/j.asoc.2007.05.003

Alcala-Fdez, J., Sanchez, L., Garcia, S., Del Jesus, M. J., Ventura, S., Garrell, J.-M., Otero, J., Romero, C., Bacardit, J., Rivas, V. M., Fernández, J. C., & Herrera, F. (2009). KEEL: A Software Tool to Assess Evolutionary Algorithms for Data Mining Problems. *Soft Computing*, *13*(3), 307–318. doi:10.1007/s00500-008-0323-y

Altay Guvenir, H., & Uysal, I. (2000). *Bilkent University Function Approximation Repository*. Retrieved from http://funapp.cs.bilkent.edu.tr

Alves, R., Rodríguez-Baena, D. S., & Aguilar-Ruiz, J. S. (2010). Gene association analysis: A survey of frequent pattern mining from gene expression data. *Briefings in Bioinformatics*, *11*(2), 210–224. doi:10.1093/bib/bbp042 PMID:19815645

Ankita, S., Shikha, A., Jitendra, A., & Sanjeev, S. (2013). A Review on Application of Particle Swarm Optimization in Association Rule Mining. In S. C. Satapathy, S. K. Udgata, & B. N. Biswal (Eds.), *Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA)* (*Vol. 199*, pp. 405–414). Berlin: Springer Berlin Heidelberg. 46 doi:10.1007/978-3-642-35314-7_46

Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Computers & Structures*, *169*, 1–12. doi:10.1016/j.compstruc.2016.03.001

Brussel, T., Müller, E., & Goethals, B. (2016). Discovering Overlapping Quantitative Associations by Density-Based Mining of Relevant Attributes. *Proceedings of the 9th International Symposium on Foundations of Information and Knowledge Systems* - Volume 9616 (pp. 131–148). New York, NY: Springer-Verlag New York, Inc. doi:10.1007/978-3-319-30024-5_8

dos Santos Coelho, L., Richter, C., Mariani, V. C., & Askarzadeh, A. (2016, 11). Modified crow search approach applied to electromagnetic optimization. *2016 IEEE Conference on Electromagnetic Field Computation (CEFC)*. doi:10.1109/CEFC.2016.7815927

Fayyad, U. M., & Irani, K. B. (1993). Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. *IJCAI (United States)*, 1022–1029. http://dblp.uni-trier.de/db/conf/ijcai/ijcai93.html#FayyadI93

Finner, H. (1993). On a Monotonicity Problem in Step-Down Multiple Test Procedures. *Journal of the American Statistical Association*, *88*(423), 920–923. doi:10.1080/01621459.1993.10476358

Gadekallu, T. R., Khare, N., Bhattacharya, S., Singh, S., Maddikunta, P. K., Ra, I.-H., & Alazab, M. (2020). Early Detection of Diabetic Retinopathy Using PCA-Firefly Based Deep Learning Model. *Electronics (Basel)*, *9*(2), 274. Advance online publication. doi:10.3390/electronics9020274

Gadekallu, T. R., Khare, N., Bhattacharya, S., Singh, S., Maddikunta, P. K., & Srivastava, G. (2020). Deep neural networks to predict diabetic retinopathy. *Journal of Ambient Intelligence and Humanized Computing*. Advance online publication. doi:10.1007/s12652-020-01963-7

Garg, H. (2016). A hybrid PSO-GA algorithm for constrained optimization problems. *Applied Mathematics and Computation*, *274*, 292–305. doi:10.1016/j.amc.2015.11.001

Garg, H. (2019). A hybrid GSA-GA algorithm for constrained optimization problems. *Information Sciences*, *478*, 499–523. Advance online publication. doi:10.1016/j.ins.2018.11.041

Ghosh, A., & Nath, B. (2004). Multi-objective Rule Mining Using Genetic Algorithms. *Inf. Sci.*, *163*(1-3), 123–133. doi:10.1016/j.ins.2003.03.021

Gupta, D., Rodrigues, J., Sundaram, S., Khanna, A., Korotaev, V., & Albuquerque, V. H. (2018). Usability feature extraction using modified crow search algorithm: A novel approach. *Neural Computing & Applications*. Advance online publication. doi:10.1007/s00521-018-3688-6

Gupta, D., Sundaram, S., Khanna, A., Ella Hassanien, A., & Albuquerque, V. H. (2018). Improved diagnosis of Parkinson's disease using optimized crow search algorithm. *Computers & Electrical Engineering*, *68*, 412–424. doi:10.1016/j.compeleceng.2018.04.014

Hassanien, A. E., Rizk-Allah, R. M., & Elhoseny, M. (2018, June 25). A hybrid crow search algorithm based on rough searching scheme for solving engineering optimization problems. *Journal of Ambient Intelligence and Humanized Computing*. Advance online publication. doi:10.1007/s12652-018-0924-y

Heraguemi, K. E., Kamel, N., & Drias, H. (2016). Multi-swarm bat algorithm for association rule mining using multiple cooperative strategies. *Applied Intelligence*, *45*(4), 1021–1033. doi:10.1007/s10489-016-0806-y

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

Huang, K.-W., Girsang, A. S., Wu, Z.-X., & Chuang, Y.-W. (2019). A Hybrid Crow Search Algorithm for Solving Permutation Flow Shop Scheduling Problems. *Applied Sciences (Basel, Switzerland)*, *9*(7), 1353. Advance online publication. doi:10.3390/app9071353

Iwendi, C., Maddikunta, P. K., Gadekallu, T. R., Lakshmanna, K., Bashir, A. K., & Piran, M. J. (2020). A metaheuristic optimization approach for energy efficiency in the IoT networks. *Software, Practice & Experience*, spe.2797. Advance online publication. doi:10.1002/spe.2797

Kabir, M. M., Xu, S., Kang, B. H., & Zhao, Z. (2017). A new multiple seeds based genetic algorithm for discovering a set of interesting Boolean association rules. *Expert Systems with Applications*, *74*, 55–69. doi:10.1016/j.eswa.2017.01.001

Kowalski, P. A., Franus, K., & پاukasik, S. (2019). Crow Search Algorithm for Continuous Optimization Tasks. *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, 7-12. 10.1109/CoDIT.2019.8820600

Kuo, R. J., Gosumolo, M., & Zulvia, F. E. (2017). Multi-objective particle swarm optimization algorithm using adaptive archive grid for numerical association rule mining. *Neural Computing & Applications*, *163*, 123. doi:10.1007/s00521-017-3278-z

Lakshman, K., Kaluri, R., Gadekallu, T., Nagaraja, G., & Subramanian, D. V. (2016). *An enhanced algorithm for frequent pattern mining from biological sequences.* Academic Press.

Li, K., Li, W., Chen, Z., & Liu, Y. (Eds.). (2018). *Computational Intelligence and Intelligent Systems*. Springer Singapore. doi:10.1007/978-981-13-1648-7

Liu, J., Wang, C., Liu, H., Xiao, Y., Hao, S., Zhang, X., . . . Yu, H. (2018). Maize Gene Regulatory Relationship Mining Using Association Rule. In K. Li, W. Li, Z. Chen, & Y. Liu (Eds.), Computational Intelligence and Intelligent Systems (Vol. 873, pp. 249–258). Singapore: Springer Singapore. 21 doi:10.1007/978-981-13-1648-7_21

Liu, L., & Özsu, M. T. (Eds.). (2017). Encyclopedia of Database Systems. New, York, NY: Springer New York. doi:10.1007/978-1-4899-7993-3

Liu, L., & Özsu, M. T. (2018). Encyclopedia of Database Systems. New, York, NY: Springer New York. doi:10.1007/978-1-4614-8265-9

Lud, M.-C., & Widmer, G. (2000). Relative Unsupervised Discretization for Association Rule Mining. In D. A. Zighed, J. Komorowski, & J. Zytkow (Eds.), *Principles of Data Mining and Knowledge Discovery* (pp. 148–158). Springer Berlin Heidelberg. doi:10.1007/3-540-45372-5_15

Manju and C. Kant. (2015). Mining association rules directly using ACO without generating frequent itemsets. *2015 International Conference on Energy Systems and Applications*, 390-395. doi:10.1109/ICESA.2015.7503377

Martín, D., Alcalá-Fdez, J., Rosete, A., & Herrera, F. (2016). NICGAR: A Niching Genetic Algorithm to mine a diverse set of interesting quantitative association rules. *Information Sciences*, *355-356*, 208–228. doi:10.1016/j.ins.2016.03.039

Martín, D., Rosete, A., Alcalá-Fdez, J., & Herrera, F. (2014). QAR-CIP-NSGA-II: A new multi-objective evolutionary algorithm to mine quantitative association rules. *Information Sciences*, *258*, 1–28. doi:10.1016/j.ins.2013.09.009

Martín, D., Rosete-Suárez, A., Alcala-Fdez, J., & Herrera, F. (2014). A New Multiobjective Evolutionary Algorithm for Mining a Reduced Set of Interesting Positive and Negative Quantitative Association Rules. *IEEE Transactions on Evolutionary Computation*, *18*(1), 54–69. doi:10.1109/TEVC.2013.2285016

Martínez Ballesteros, M. a., Martínez-Álvarez, F., Troncoso, A., & Riquelme, J. (2013). A Sensitivity Analysis for Quality Measures of Quantitative Association Rules. doi:10.1007/978-3-642-40846-5_58

Martínez-Ballesteros, M., Martínez-Álvarez, F., Troncoso, A., & Riquelme, J. C. (2014). Selecting the best measures to discover quantitative association rules. *Neurocomputing*, *126*, 3–14. doi:10.1016/j.neucom.2013.01.056

Martínez-Ballesteros, M., Nepomuceno-Chamorro, I. A., & Riquelme, J. C. (2014). Discovering gene association networks by multi-objective evolutionary quantitative association rules. *Journal of Computer and System Sciences*, *80*(1), 118–136. doi:10.1016/j.jcss.2013.03.010

Martínez-Ballesteros, M., Troncoso, A., Martínez-Álvarez, F., & Riquelme, J. C. (2016). Improving a multi-objective evolutionary algorithm to discover quantitative association rules. *Knowledge and Information Systems*, *49*(2), 481–509. doi:10.1007/s10115-015-0911-y

Mohammadi, F., & Abdi, H. (2018). A Modified Crow Search Algorithm (MCSA) for Solving Economic Load Dispatch Problem. *Applied Soft Computing*, *71*, 51–65. doi:10.1016/j.asoc.2018.06.040

Narang, N., Patwal, R., & Garg, H. (2017). A novel TVAC-PSO based mutation strategies algorithm for generation scheduling of pumped storage hydrothermal system incorporating solar units. *Energy*.

Oliva, D., Hinojosa, S., Cuevas, E., Pajares, G., Avalos, O., & Gálvez, J. (2017). Cross entropy based thresholding for magnetic resonance brain images using Crow Search Algorithm. *Expert Systems with Applications*, *79*, 164–180. doi:10.1016/j.eswa.2017.02.042

Priya, S., Bhattacharya, S., Reddy, P., Somayaji, S., Lakshman, K., Kaluri, R., & Gadekallu, T. et al. (2020). Load balancing of energy cloud using wind driven and firefly algorithms in internet of everything. *Journal of Parallel and Distributed Computing*. Advance online publication. doi:10.1016/j.jpdc.2020.02.010

Ruiz, M. D., Gómez-Romero, J., Molina-Solana, M., Ros, M., & Martin-Bautista, M. J. (2017). Information fusion from multiple databases using meta-association rules. *International Journal of Approximate Reasoning*, *80*, 185–198. doi:10.1016/j.ijar.2016.09.006

Satapathy, S. C., Udgata, S. K., & Biswal, B. N. (Eds.). (2013). *Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA).* Berlin: Springer Berlin Heidelberg. doi:10.1007/978-3-642-35314-7

Sayed, G. I., Darwish, A., & Hassanien, A. E. (2017). Chaotic crow search algorithm for engineering and constrained problems. *2017 12th International Conference on Computer Engineering and Systems (ICCES)*, 676-681. doi:10.1109/ICCES.2017.8275390

Shekhawat, S., & Saxena, A. (2020). Development and applications of an intelligent crow search algorithm based on opposition based learning. *ISA Transactions*, *99*, 210–230. doi:10.1016/j.isatra.2019.09.004 PMID:31515097

Siegel, S., & Castellan, N. J. (1988). *Nonparametric statistics for the behavioral sciences* (2nd ed.). McGraw–Hill, Inc.

Srikant, R., & Agrawal, R. (1996). Mining Quantitative Association Rules in Large Relational Tables. *SIGMOD Record*, *25*, 1–12. doi:10.1145/235968.233311

Wang, L., Dong, J.-Y., & Li, S.-L. (2015). Fuzzy Inference Algorithm based on Quantitative Association Rules. *Procedia Computer Science*, *61*, 388–394. doi:10.1016/j.procs.2015.09.166

Wang, L., Meng, J., Xu, P., & Peng, K. (2018). Mining temporal association rules with frequent itemsets tree. *Applied Soft Computing*, *62*, 817–829. doi:10.1016/j.asoc.2017.09.013

Wolpert, D. H., & Macready, W. G. (1995). *No Free Lunch Theorems for Search.* Working Papers, Santa Fe Institute.

Yamada, T. (1990). 2 - Principles of Error Detection and Correction. In H. Imai (Ed.), *Essentials of Error-Control Coding Techniques* (pp. 11–37). Academic Press. doi:10.1016/B978-0-12-370720-8.50006-4

Yan, D., Zhao, X., Lin, R., & Bai, D. (2019). PPQAR: Parallel PSO for quantitative association rule mining. *Peer-to-Peer Networking and Applications*, *8*(5), 53. doi:10.1007/s12083-018-0698-1

Yin, Y., Zhong, Z., & Wang, Y. (2008). Mining Quantitative Association Rules by Interval Clustering. *Journal of Computer Information Systems*, *4*, 609–616.

Zhang, Z., Pedrycz, W., & Huang, J. (2018). Efficient mining product-based fuzzy association rules through central limit theorem. *Applied Soft Computing*, *63*, 235–248. doi:10.1016/j.asoc.2017.11.025

Zhu, X. (2009). Quantitative Association Rules. In L. I. Ling & Ö. Z. M. Tamer (Eds.), *Encyclopedia of Database Systems* (pp. 2240–2244). Springer US., doi:10.1007/978-0-387-39940-9_291

Makhlouf Ledmi received his BS degree in Computer Science from University of Annaba (Algeria) in 1997, and MS degree in Computer Science from University of Khenchela (Algeria) in 2010. He is working as professor in Department of Mathematics and Computer Science in University of Khenchela (Algeria). His research interests include Data Mining, Particle Swarm Optimisation, and Complex Systems.

Hamouma Moumen is an Associate Professor at the University of Batna 2. He is the Dean of the Faculty of Mathematics and informatics, University of Batna 2. His main research interests are the basic principles of distributed computing systems. He has published papers in the IEEE NCA, OPODIS, PODC, and ICDCN international conferences and papers in Journals (Journal of the ACM and Theoretical Computer Science - Journal - Elsevier). He is the recipient of “the Best Paper Award” at ACM PODC 2014.

Abderrahim Siam received his BS degree in Computer Science from University of Batna (Algeria) in 2002, and MS degree in Computer Science from University of Oum El Bouaghi (Algeria) in 2005and his Ph.D in Computer Science from University of Constantine, Algeria. He is working as professor in Department of Mathematics and Computer Science in University of khenchela (Algeria). He is currently the vice rector of University of khenchela (Algeria) and his research interests include software engineering, fuzzy logics, formal methods, multiagent systems and complex systems.

Hichem Haouassi received his Magister in Computer Engineering from Computer science department in Batna university, Algeria And his Ph.D in Computer sciences in the same university. Since 2004, he has been working as Assistant Professor at the Department of Computer sciences, Abbes Laghrour University, Algeria. His research interests are in Artificial intelligence, Swarm-based optimization. Data mining, Feature selection, and classification.

Nabil Azizi received his BS degree in Computer Science from University of Annaba (Algeria) in 1998, and MS degree in Computer Science from University of Khenchela (Algeria) in 2010. He is working as professor in Department of Mathematics and Computer Science in University of Khenchela (Algeria). His research interests include Data Mining, Particle Swarm Optimisation, and Complex Systems.