REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOHAMED BOUDIAF - M'SILA
FACULTE DES MATHEMATIQUES ET DE L'INFORMATIQUE

DEPARTEMENT D'INFORMATIQUE

MEMOIRE de fin d'étude

Présenté pour l'obtention du diplôme de MASTER

Domaine : Mathématiques et Informatique

Filière : Informatique

Spécialité : Informatique Décisionnel et Optimisation

Par : amira zabi –kadiri houria

SUJET

Whale optimization algorithm for solving association rule mining

Soutenu publiquement le : / /2020 devant le jury composé de :

Dr. ……	**Université de M'sila**
Examinateur	
Dr. Kamel eddin heraguemi	**Université de M'sila**
Rapporteur	
Dr. ……	**Université de M'sila**
Examinateur	

2019/2020

Dedications

To our dear parents.

Acknowledgments

First of all, Praise be to God, who gave us strength and patience to carry out these work. We would like to thank **Dr. KamelEddine Heraguemi**, for the honor he gave us in supervising us, and for the advises given to us while completing this work.

Also, We thank the jury members for agreeing to take care of us and evaluate this work.

We keep the last of my very warm thanks to our parents and family for their constant presence and support.

Finally, We want to thank all our teachers and our friends who provided their assistance.

Table of Contents

List of Figures

List of table

List of algorithms

Introduction general

INTRODUCTION GENERAL:

Nowadays, stored data have grown due to the enormous development of the INTERNET and unlimited connected devices. All these stored data are used by several users and companies to generate useful information to help for decision making. Therefore, data processing turned into a real challenging issue, which imposes the development of new frameworks and methods with low processing time and low memory usage. The most frequently used method to process data in the last decade is Knowledge Discovery in Databases (KDD), which aims to extract interesting patterns from stored data, commonly contains three successive stages: Pre-processing; Data Mining; and finally Post-Processing. Within KDD, the primary process is data mining that has a goal to extract non-trivial information hidden in data. It contains several techniques used commonly in data processing such as Classification, Clustering, Regression analysis, and Association rules [1].

Association rule (AR) has attracted researchers' attention since its first release by Agrawal et al. in the early 90s [2] . It refers to relationships that exist between items in a real-world database. It was designed initially for market basket analysis to obtain relations between products, like *milk* $\Rightarrow$ *bread*, which means that someone bought milk, also get bread with high probability. These rules would allow managers to plan their marketing strategy to increase benefits. In the last decade, ARs become very utilized in different application domains such as medical diagnosis, biomedical literature, protein sequences, logistic regression, and fraud detection on the web, etc. Mining association rules is the process that generates relationships among items in a data-set that generally given as If-THEN statements. Restrictions are in If statement, and those inside THEN clause are Outcomes. Many traditional algorithms have been developed to solve ARM issues, such as Apriori[3], FP-growth [4], Etc. These algorithms created to extract all the relations that exist in the dataset. However, they suffer, our days, from the considerable quantity of data stored in databases that affect their execution time and memory usage. In order to overcome exact algorithms drawbacks, researchers apply intelligent meta-heuristic, which are previously employed to solve numerous NP-Complete problems, In which ARM problems can be classed. As an NP-complete problem, many works proposed to use evolutionary algorithms and swarm-inspired algorithms to solve Arm to pick the optimal rules. Firstly, genetic algorithm[5] has been successfully applied and given promising results.. Few years after swarm intelligence was employed with ARM using various well-known algorithms such as Particle Swarm Optimization [6], Bees swarm Optimization (BSO) [7], Bat algorithm (BA) [8]..Etc. Formally, The datasets regarded as sample search

space, in which the algorithm tries to maximize/minimize an objective mathematical function that compute the selected rules quality according to several measurements.

The whale optimization algorithm is newly swarm-based algorithms produced by Mirjalili et al.[9]. It simulates the whales' humpback hunting behavior. Usually, humpback whales hunt fishes near sea surface by moving around the victim and produce bubbles circle or 9-formed path. This method is a specific hunting technique for humpback whales called the bubble-network feeding method. Many pieces of research have been conducted on WOA in the last three years. Those works applied WOA in various real-world optimization problems utilizing many ways, such as improvements, hybridization, and proposing new variants for the algorithm[10]. WOA confirmed its competitivity in-the-face of other swarm inspired metaheuristics such as PSO, BA, and BSO in terms of exploitation by exploration. Which make WOA one of the most utilized in numerous domain as: Electrical Engineering[11], Classification[12], Clustering[13], Image Processing[14], and many other problems. Highly motivated by the success of WOA, this memory suggests a new whale optimization algorithm to deal with the association rule mining problem named WO-ARM, to extract high-quality rule that can be useful for the final user. This proposal investigates the advantages of the whale algorithm. Firstly, with its simplicity and low complexity, it will utilize lower computing power and less memory. Also, WOA needs a low number of parameters that make it suitable to use for final users. In order to judge the stated WO-ARM method, profound experimental tests are carried out on various datasets benchmarks with different sizes. Our initial outcome are encouraging. Also, the proposed algorithm demonstrates its effectiveness compared to similar methods in the Association Rule Mining field according to runtime consumption and rules quality.

The remainder of this memory designed as follows : in the next chapter of writing will are talking a general about the rule mining problem and its principals presented and measurements, and mentioned the exact algorithms for association rule mining and know its in real life applications . In the second chapter are talking the state-of-art regarding novel work in the association rule mining field and the different evolutionary algorithm applied to solve this mentioned problem . Furthermore , the chapter will introduce the original whale optimization algorithm . After , third chapter outlined our proposal , and our experimentation and results interpreted , and our application graphical interface has been recognized . Eventually , we will draw some conclusions.

Chapter 1: Association rule mining

1. Introduction :

Our-days , with the significant number of connected devices, data stored has grown significantly. The exploitation of the information stored in it for decision-making has become indispensable in the most prominent companies. Association rule mining draws in consideration of researchers to extract relationships between items in data. Nevertheless , of finding association rules it is a daunting task but if successful it is very rewarding.

Association Rule Mining often uses to analyze sales transactions to find correlation between different objects in a group or to discover frequent, interesting and strong relationships between a database, in order to find out which items customers buy frequently together by creating a set of rules called union rules . In simple words it gives you outputs as rules in shape if this is then Clients can use these rules for many marketing strategies .

2. Datamining :

The term data mining is proposed in recent decades , it is one of the fastest growing domain in computer science due to the large data stored last years. Data mining is essential phase in the whole Knowledge discovery in databases (KDD) process . With the aim to detect the hidden knowledge from databases, data mining methodologies are inspired from multiple fields including mathematics and computer science such as: machine learning, artificial intelligence, and statistics, etc. On the other hand, data mining techniques serve a lot of benefits to several fields as: business, medicine, market ing and factory assembly lines etc [15].

2.1 Definition :

Data mining is a process of discovering and extracting interesting knowledge from large amounts of data stored in databases, data warehouses, or other information repositories, and summarizing it into useful information [16]by using software and tools to look for discrimination and expressive patterns . Data mining can be categorized into tasks, according to different goals of a data mining practitioner. The two "high level" primary goals of data mining in practice, are descriptive data mining and predictive data mining [17] ,as shown in Figure (1) ,The first is used to find correlations and subgroups in data with the aim of studying and focusing on the characteristics of data. Whereas, the later is used to predict the future outcome based on the current behaviour, for example, predicting the future sales for a customer depends on his/her historical data such as: age, gender and purchase items. In the following, we present the four main data mining task classes:

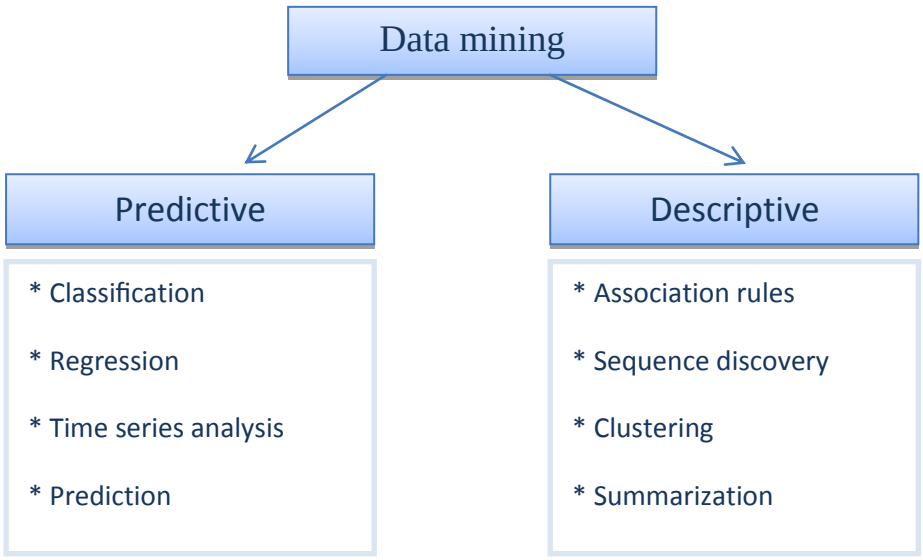


Figure (1) : Data mining tasks.

2.2 Data mining steps :

- **Classification :**

Classification is a predictive data mining task. It aims to assign an object to a certain predefined class based on its similarity to training sets in the databases. It aims at building a model from different data attributes where one of these attributes is a target class. Therefore, the model can be able to predict the class of new cases. Generally, classification task is decomposed into two steps: learning and classification, the former aims to describe a set of classification rules, whereas, the later verify and validate the rules generated in the first step.

Classification technique is used in customer segmentation, modeling businesses, credit analysis, and many other applications. For example, Classifying prospects as good or bad customers based on their historical transactions. The most frequently used techniques in classification are Non-parametric (K nearest neighbor), Mathematical models(neural networks) and Rule based models(decision trees).[18]

- **Regression Analysis :**

Regression is another predictive task, it is the oldest statistical technique known in data mining. Basically, it fits data to a mathematical formula. In order to specify the correct

formula, it is necessary to know the forms of correlations for the data. Moreover, regression can be seen as a special case of the classification task when the target attribute is numeric. For example, it uses the linear equation y = ax + b and determines values for a and b with a given value of x to predict the future value of y. The advantage of such methods is the possibility to gain from the equation, some qualitative knowledge about input-output relationships. In the literature numerous techniques[19] are proposed for regression such as linear regression, logistic regression and elastic net regression, etc.

- **Clustering :**

Clustering is one of the most significant descriptive techniques in data mining, in which data are partitioned into several meaningful groups. It is the processes of finding a set of classes called clusters, in database, such as these classes contain items that shared similar characteristics. Hence, it is mainly used to find hidden classes in datasets. This technique generates classes in such a way that the similarities are maximized in intra-classes and minimized in inter-classes. Next, the data are classed into these classes. Many methods are developed for clustering. Among these methods, the most known method K-means[20].

- **Association rules :**

Association rule mining is another descriptive technique in data mining. It is the discovery of relationships and correlations between items in a defined dataset, where, an association rule is expressed as an IF-THEN statement between two item-set X and Y; which represents the condition and consequence respectively. Association rules[21] is useful in many domains such as business, medicine, etc. For example a supermarket can judge the clients behaviour based on the historical transactions and use these rules, which shows products frequently bought together, to design catalogs or take some marketing decisions. Much research has been performed recently on association rule mining with efecient algorithms, including several types of rules such as: level-wise Apriori search, mining multiple-level, multidimensional associations, mining associations for numerical and categorical data, etc., Moreover, there are other data mining tasks, such as sequence discovery, outliers analysis and summarization, etc.[18]

3. Association rule mining :

Association rule mining problem has been largely studied since it first appears in 1993 by Agrawal and co-workers [21]. A surprising number of studies can be found in the literary

study, that can be separated into two principal classes: exact and optimization methods. The first class aims to extract all the relationships between items exists in all the database, whereas the other has a goal to generate the primary and useful rules to the final user. It was designed initially for market basket analysis to obtain relations between products, like milk ⇒ bread , which means that someone bought milk, also get bread with high probability. These rules would allow managers to plan their marketing strategy to increase benefits.

3.1 Market basket problem :

Association Rule Mining is sometimes referred to as "Market Basket Analysis", as it was the first application area of association mining. The aim is to discover associations of items occurring together more often than you'd expect from randomly sampling all the possibilities, For instance, if customers are buying milk, how likely are they also buy bread (and what kind of bread) on the same trip to the supermarket?

data are collected using bar-code scanners in supermarkets. Such 'market basket' databases consist of a large number of transaction records. Each record lists all items bought by a customer on a single purchase transaction. Managers would be interested to know if certain groups of items are consistently purchased together. They could use this data for adjusting store layouts (placing items optimally with respect to each other), for cross-selling, for promotions, for catalog design and to identify customer segments based on buying patterns [22] .

3.2 Definition ARM :

In 1993, agrawal et al. introduced association rule mining problem[21], to extract typical business decisions for helping supermarket managers to design coupons, place products on shelves in order to maximize the profits. These decisions are taken based on the relationships generated from a large amount of transaction history collected over time from sells.

An example of association rule problem is defined as follow: Let I = {i1 , i2 , ..., in } be a set of literals called items, let D{t1 , t2 , ..., tm } be a transactional database where each transaction t contains a set of items. An association rule is implication like $X \Rightarrow Y$ where $X, Y \in I$ and $X \cap Y = \emptyset$. The item-sets X, Y are named antecedent and consequent, respectively. In order to evaluate the generated association rules from any data-set.

3.3 Transactional data base representations :

Transactional database refers to the collection of transaction records, in most cases they are sales records. With the popularity of computer and e-commerce, massive transactional databases are available now. Data mining on transactional database focuses on the mining of association rules, finding the correlation between items in the transaction records[23]. transactional database can be represented in horizontal, vertical and Bitmap representation [24] .

- **Horizontal representation :** it is the most used layout in association rule mining algorithms. Each transaction is represented by an itemset which contains the items bought in that transaction [40] , see Table (1).

T	items
1	A,B,C
2	A,C
3	B,C

Table (1) : Horizontal representation of the transactional database .

- **Vertical representation :** In this layout, the transactional database is represented by a set of items. However, each item is defined by a set of transactions that include it. This structure is known as: Tidlists. Table (2) shows a vertical layout for a transactional database. For instance, item B belongs to transactions 1,2 and 4, so its Tidlists is {1,1,0,1}.[40]

items	T
A	1,2
B	1,3
C	1,2,3

Table (2): Vertical representation of the transactional database

- **Bitmap representation:** is a matrix of bits called Bitmap. However, each line represent one transaction which is defined by the items that includes. Such as, Bitmap[i][j]=1 if the

item j is in the transaction i, otherwise, it contains 0. This layout is usually used in parallel computing. Table (3) shows an illustration of a bitmap layout.[40]

T	A	B	C
1	1	1	1
2	1	0	1
3	0	1	1

Table (3) : Bitmap representation of the transactional database

3.4 Association rule mining measurements :

To detect the most impressive states to the final user, many measurements are invented in the literature, divided into two main groups: objectively and subjectively [25]. The first one involves statistical analysis of the data, whereas the others more oriented towards the user requirements.

In this work, the rule measurements used to evaluate the generated rules. Due to the vast number of patterns extracted from a sizable transnational database, a detected rule is accepted as association rule if its support and confidence are equal or superior to the minimum threshold, specified by a final user, and rejected otherwise. Support and confidence are two measures that aim to determine rules quality which is defined as follows:

Definition 1 : Support is the proportion of transactions in D that contains X, to the total of records in database. Support of item X is calculated using equation Error : Reference source not found and The support of an association rule $X \rightarrow Y$ is the support of $X \cup Y$ [26] .

$$support(X) = \frac{Numberoftransactionscontaining X}{TotalNumberoftransactions} \quad (1)$$

Definition 2 : Confidence is the proportion of transactions covering X and Y, to the total of records containing X. When the percentage exceeds a threshold of confidence, an interesting association rule can be generated [26]. An association rule $X \rightarrow Y$ with a confidence of 80 % means that 80 % of the transactions that contain X also contain Y . The rule confidence is calculated as follows:

$$confidence(X \Rightarrow Y)$$
$$= \frac{support(X \cup Y)}{support(X)} \qquad \text{(Erreur ! Signet non défini.)}$$

Definition 3 : lift is a measure of the performance of a targeting model (association rule) at predicting or classifying cases as having an enhanced response of the rule by Brin [27], it is defined as follows:

$$lift\ (X \rightarrow Y) = \frac{Confidence(X \rightarrow Y)}{support(y)} \qquad (3)$$

Definition 4 : Leverage measures the difference of X and Y appearing together in the data set and what would be expected if X and Y where statistically dependent. The rational in a sales setting is to find out how many more units (items X and Y together) are sold than expected from the independent sells.[28]

$$leverage(X \rightarrow Y) = support(X \rightarrow Y) - (support(X) * support(Y)) \qquad (4)$$

Definition 5 : Conviction was developed as an alternative to confidence which was found to not capture direction of associations adequately. Conviction compares the probability that X appears without Y if they were dependent with the actual frequency of the appearance of X without Y. In that respect it is similar to lift , however, it contrast to lift it is a directed measure since it also uses the information of the absence of the consequent. An interesting fact is that conviction is monotone in confidence and lift[28].

$$conviction(X \rightarrow Y) = \frac{1-supp(X)}{1-supp(X \rightarrow Y)} \qquad (5)$$

Definition 6 : Comprehensibility measure attempts to calculate the simplicity of the generated rule and its understand-ability to the user [30]. It can be calculated as:

$$Comprehensibility(X \rightarrow Y) = \frac{log(1+|Y|)}{log(1+|X \cup Y|)} \qquad (6)$$

Where, |Y| and |X∪Y| are items number in the Consequence part and the total rule respectively. The comprehensibility increases and the rules are more under-standable whenever items number in the antecedent part are smaller.

Definition 7 : Interestingness of a rule is used to quantify how much rule is surprising for users. As the most important point of rule mining is to find some hidden information, it should discover those rules having comparatively less occurrences in the database. Interestingness measure is defined [31] by :

$$Interesting(X \rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)} * \frac{supp(X \cup Y)}{supp(Y)} * \frac{(1 - supp(X \cup Y))}{N} \qquad (7)$$

Example 1 :

Let's look at the transaction database in the Table (4) , that contains 4 transactions t1,t2,t3,t4 and 3 items A,B,C. For example, to compute the support of the 2-itemset A,B, we need to calculate the number of transactions which includes both A and B, that equal to 2, thus, the support of A,B is 2/4. Thereafter, the confidence of rule A ⇒ B is equal to proportion of the support (A ⇒ B) to the support(A), that was $\frac{2/4}{3/4}$ = 2/3. If we consider the Minsup and Minconf equal to 2/4 and 2/3, respectively, then the rule A ⇒ B is accepted.

T	items
t1	A,B,C
t2	A,C
t3	B,C
t4	A,B

Table (4): An example of transactional database.

In addition, the other measurements of rule A ⇒ B are calculated as follows:

- Lift(A ⇒ B) = 8/9.
- Leverage(A ⇒ B)= -1/16 .
- Conviction(A ⇒ B) = 1/2.
- Comprehensibility(A ⇒ B) = log(4)/ log(3)= 1.26.
- Interesting(A ⇒ B) = 2/4/ 3/4 × 2/4/ 3/4 × (1−2/4) /4 = 1/18.

3.5 Exact algorithms for association rule mining (Appriori and Fp-growth) :

Association rule mining is a significant research area and different exact approaches are introduced to solve this issue since the beginning of 90s. AIS algorithm was proposed in 1993[21] as a first method proposed by Agrawal and al. for mining frequent pattern in the

database. Next, Apriori algorithm[41] was presented to mine ARs. This algorithm uses prior knowledge of frequent item-set properties to extract ARs.

In Apriori algorithm, two processes are required to find the frequent itemsets in a large database. First candidate itemsets are generated, and next, the database is scanned to accumulate the support count for each one of these candidates . firstly , the support of each item is calculated and the items that have a support count less than the predefined threshold are eliminated, using the 1-itemset generated in the first pass Apriori can generate 2-itemset by joining it with itself [40]These two processes are iterated until one of the candidate or frequent itemset become empty. Algorithm 1 presents the pseudo code of the whole process to generate frequent itemsets in Apriori .The second step of the Apriori algorithm is rule generation. This step aims to create association rules from the frequent itemsets generated in the first step.The algorithm of this step is presented in Algorithm 2.

Even so, Apriori algorithm and its new improvement techniques still generate numerous candidate itemsets which needs large time and memory consumption when the database become large. This is the most expensive step in all algorithms. The Pattern Growth algorithms have been introduced to eliminate the need for candidate generation and thus reduce the algorithms execution time [40].

Algorithm 1. Frequent itemset generation in an Apriori algorithm

```
Input:
D: transaction database;
Min sup: the minimum support threshold
Output: frequent itemsets
 L1= find frequent 1-itemsets(DB);
for k=2; Lk−1 = φ; k + + do
        Ck= Apriori gen(Lk−1);
         for each transaction t ∈ DB do
              Ct = subset (Ck , t);
             for each candidate c ∈ Ct do
                 c.count + +;
              end for
         end for
        Lk = {c ∈ Ck|c.count ≥ min sup}
 end for
return L =∪ k Lk;
Procedure Apriori gen(Lk−1: frequent(k−1)-itemsets)
```

Algorithm 2. Rule generation step in Apriori algorithm

```
1: Input:
2: Set of itemsets l
3: Min sup: the minimum support threshold
4: Min conf: the minimum confidence threshold
5: Output: Set of association rules Rules
6: for all itemsets lk , k ≥ 2 do
7:     call generate-rules(lk,lk)
8: end for
9: Procedure generate-rules(lk : k−itemset , a m : m−itemset)
10: A = {(m-1)-itemsets a m−1 | a m−1 ⊂ am}
11: for all a m−1 ⊂ A do
12:     conf = support(lk)/support(am−1)
13:     if conf ≥ Min conf then
14:         R = a m−1 ⇒ (lk –a m−1)
15:         if m-1 ≥ 1 then
16:             call generate-rules(lk,am−1)
17:             Rules = Rules ∪ R
18:         end if
19:     end if
20: end for
21: return Rules
```

FP-growth is an algorithm proposed by Han et Al. in [42] where the authors try to solve the bottlenecks of Apriori. FP-growth algorithm mines the frequent item sets without any candidate generation and needs just two scans for the database. With the avoiding of candidate generation and the less number of passes over the database, FP-tree becomes faster than Apriori. Generally, FP-Growth algorithm includes two principal steps: constructing the FP-tree and generating frequent patterns based on the Tree constructed in the first step. The general processing of this proposal can be summarized in three essential points:

1. In the first scan, as in Apriori, all supports of 1-itemsets are calculated, the algorithm sorted these itemsets in descending relative to their supports counts, also head table is created.

2. Start create the FP-tree with head table, another scan of the database is required. For each transaction the items, that not satisfied the minimum support threshold, are deleted

and the others are resorted relative to their supports in the first scan. Also, the root node is created and labeled with Root.

3. For each transaction in the transformed data, items are inserted to the tree by calling the Insert([p |P],T), where T is the Fp-tree, p is the item that will be inserted and P is the rest of the transaction items. Insert function is processed as follows : if the tree contains the node p then the count of p is increased by 1, otherwise a new node p is created and inserted with a support count of 1.

3.6 Association rule mining real life application :

- **Market Basket Analysis:**

This is the most typical example of association mining [43] . Data is collected using bar-code scanners in most supermarkets. This database, known as the "market basket" database [41] , consists of a large number of records on past transactions. A single record lists all the items bought by a customer in one sale. Knowing which groups are inclined towards which set of items gives these shops the freedom to adjust the store layout and the store catalog to place the optimally concerning one another.

- **Medical diagnosis:**

Applying association rules in medical diagnosis can be used for assisting physicians to cure patients. The general problem of the induction of reliable diagnostic rules is hard because theoretically no induction process by itself can guarantee the correctness of induced hypotheses [32]. Practically diagnosis is not an easy process as it involves unreliable diagnosis tests and the presence of noise in training examples. This may result in hypotheses with unsatisfactory prediction accuracy which is too unreliable for critical medical applications [33]. Serban [32] has proposed a technique based on relational association rules and supervised learning methods. It helps to identify the probability of illness in a certain disease. This interface can be simply extended by adding new symptoms types for the given disease, and by defining new relations between these symptoms.

- **Protein sequences:**

Proteins are important constituents of cellular machinery of any organism. recombination DNA technologies have provided tools for the rapid determination of DNA sequences and, by inference, the amino acid sequences of proteins from structural genes [34]. Proteins are sequences made up of 20 types of amino acids. Each protein has a unique 3-dimensional structure, which depends on amino-acid sequence; slight change in sequence may change the

functioning of protein. The heavy dependence of protein functioning on its amino acid sequence has been a subject of great anxiety.

Lot of research has gone into understanding the composition and nature of proteins; still many things remain to be understood satisfactorily. It is now generally believed that amino acid sequences of proteins are not random. Nitin Gupta, Nitin Mangal, Kamal Tiwari, and Pabitra Mitra [22] have deciphered the nature of associations between different amino acids that are present in a protein. Such association rules are desirable for enhancing our understanding of protein composition and hold the potential to give clues regarding the global interactions amongst some particular sets of amino acids occurring in proteins. Knowledge of these association rules or constraints is highly desirable for synthesis of artificial proteins.

- **Census data:**

Censuses make a huge variety of general statistical information on society available to both researchers and the general public [35]. The information related to population and economic census can be forecaster in planning public services(education, health, transport, funds) as well as in public business(for setup new factories, shopping malls or banks and even marketing particular products). The application of data mining techniques to census data and more generally to official data, has great potential in supporting good public policy and in underpinning the effective functioning of a democratic society [36] . On the other hand, it is not undemanding and requires exigent methodological study, which is still in the preliminary stages.

- **CRM of credit card business:**

Customer Relationship Management (CRM), through which, banks hope to identify the preference of different customer groups, products and services tailored to their liking to enhance the cohesion between credit card customers and the bank, has become a topic of great interest [37]. Shaw [38] mainly describes how to incorporate data mining into the framework of marketing knowledge management.

The collective application of association rule techniques reinforces the knowledge management process and allows marketing personnel to know their customers well to provide better quality services. Song [39] proposed a method to illustrate change of customer behavior at different time snapshots from customer profiles and sales data.

The basic idea is to discover changes from two datasets and generate rules from each dataset to carry out rule matching.

4. Conclusion :

This chapter focused on the association rule mining . It is all about to find some kind of pattern or relationship between various items in datasets. In the future, association rule mining will include more complex data types. Research in association rule mining will generate new methods to determine the most interesting characteristics in the data. As many models are developed and implemented, ARM has become a research area with increasing importance.

Chapter2: Bio-insired algorithms for association rule mining

1. Introduction:

In our daily life, we face various problems, which led researchers to suggest ways to solve them and make great efforts to improve their performance from nature, which is a great source of inspiration for researchers to solve problems in terms of the required computing time and / or the quality of the proposed solution. Over years several solutions have been proposed to many problems of various complexities. Consequently, a large variety of different principles and strategies were distinguished.

Evolutionary algorithms are inspired by the theory of evolution, Animals motion and their organization gives birth of swarm intelligence to solve optimization problems . In this chapter, we provide an overview about swarm intelligence methods such as: ants colony, bee swarm, and particle swarms ... etc., then a review on smart methods Applied to extract association rules.

2. Generalities Optimization Problems:

2.1 Optimization Problems

Optimization problems An optimization problem consists in maximizing or minimizing some function relative to some set, representing a range of choices available in a certain situation. The function allows comparison of the different choices for determining which might be best. More formally we define the optimization problem as

$$\text{Optimize } f(x) x \in S$$

where optimize stands for min or max $f: R^n \to R$ denotes the objective function, that we assume throughout at least continuously differentiable, and $S \subseteq R^n$ is the feasible set, namely the set of all admissible choices for x. In the following we will refer to minimization problems. Indeed the optimal solution of a maximization problem

$$max f(x) \quad x \in S$$

coincide with the optimal solutions of the minimization problem

$$min - f(x) \quad x \in S$$

and we have: $\max_{x \in S} f(x) = -\min_{x \in S} \big(-f(x)\big)$

The feasible set S is a subset of$R^n$and hence $x = (x_1, x_2, \dots x_n)^T$ is the vector of variables of dimension n and f is a function of n real values f)[44]

2.2 Solving optimization problems:

Three steps to solve optimization problems[44] :

- **Identify the problem**

1. Draw a picture.

2. Determine your Objective Equation. The Objective Equation is the equation that illustrates the object of the problem. If asked to maximize area, an equation representing the total area is your objective equation. If asked to minimize cost, an equation representing the total cost is your objective equation.

3. Determine your Constraint Equation. The Constraint Equation is an equation representing any constraints that you are given in the problem. Note: There may not always be a constraint in the problem. This may imply that the objective equation is already in one variable.

4. Make sure that the objective equation is in terms of one variable. You will probably be able to use the constraint equation in some way to complete this step.

5. Take the first derivative of the objective equation, set it equal to zero, and solve for your variable.

6. Go back and make sure you have solved for all variables in the problem.[45]

- **Modeling the problem**

is simplification of reality. The quality of the solution depends on the quality of the model. The founded solution is for the abstract model and not the original There are two common approaches:

  - Simplified model, exact approach

Figure 2: Simplified model, exact approach.

- Precise model, approximate approach

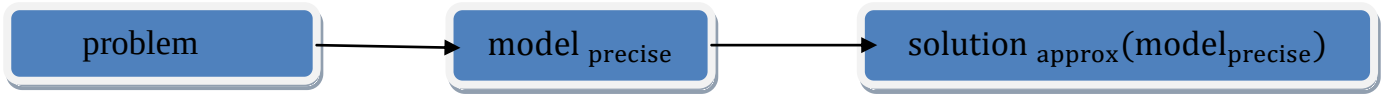


Figure 3: Precise model, approximate approach.

- **Solving models**

After the modeling of the original problem, the model can be solved by some kind of algorithm (usually an optimization algorithm). An algorithm is a procedure (a finite set of well-defined instructions) for accomplishing some task. an algorithm starts in an initial state and terminates in a defined end-state. The goal of an algorithm to find a solution (either

specific values for the decision variables or one specific decision alternative) with minimal or maximal evaluation value. [44]

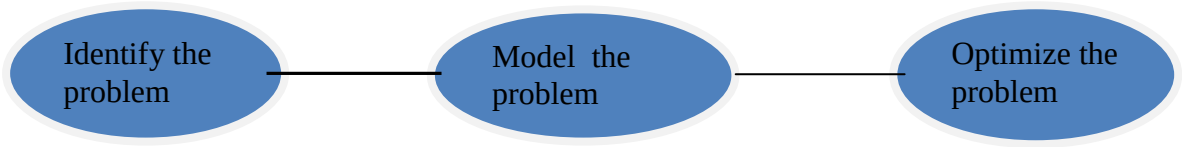


Figure 4: solving model.

2.3 Formal definition of optimization problem:

An instance of a combinatorial optimization problem is a pair (S , f) where :

- S is the set of feasible solution (solution space, search space) [44]
- $f : S \rightarrow \Re$ is the objective function to optimize
- S is defined by input parameters and constraints

In discrete (or combinatorial) optimization we concentrate on optimization problems, where for every instance I = (S',f) the set S of feasible solutions is **discrete**, i.e., F is finite or countably infinite. The main goal in solving an optimization problem is finding the global optimum (or one of them if there are multiple global optimal) [44]

2.3.1 Global optimum*:*

- A solution $s^* \in S$ is a global optimum if
  - ❖ $\forall s' \in S, f(s^*) \leq f(s')$ (minimization problem)
  - ❖ $\forall s' \in S, f(s^*) \geq f(s')$ (maximization problem) [25]

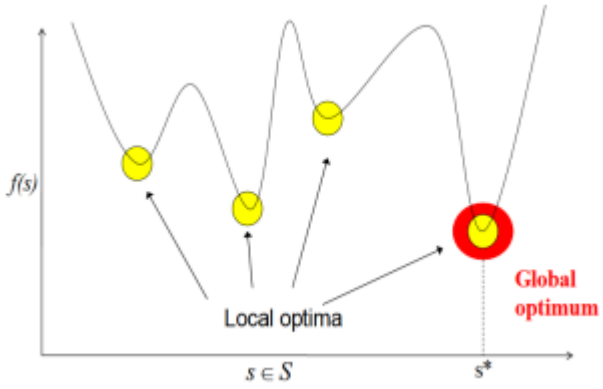


Figure 5 : global optimum and local optimum

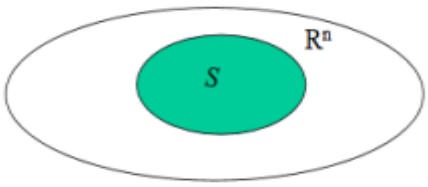
2.3.2 Solution space:

S is solution space for an optimization problem

- ✓ S is empty that means no solution exists and the problem is too constrained
- ✓ S is non-empty that means one or more (even finite) optimal solution can exist (with the same value of f)

$S = \{ x \in \Re^n$ such that x fulfils the constraint $\}$

$S \subseteq \Re^n$ [44]



A combinatorial optimization problem can actually be attacked in three different versions which are:

I. Search version: given an instance (S , ,f), find an optimal solution, that is an element

Sopt ∈ Sopt .

II. Evaluation version: given an instance (S , ,f), find the optimal objective function value

opt f (Sopt)

III. Decision version: given an instance (S, f) and a bound L, decide whether there is a

feasible solution $S' \in S$ with f (S') ≤L. [44]

3.1 Genetic algorithm (GA)

Genetic algorithm (GA) is a meta-heuristic inspired by the process of natural selection that belongs to the larger class of evolutionary algorithms (EA). Genetic algorithms are commonly used to generate high-quality solutions to optimization and search problems by relying on biologically inspired operators such as mutation, crossover and selection.[46] John Holland introduced genetic algorithms in 1960 based on the concept of Darwin's theory of evolution, and his student David E. Goldberg further extended GA in 1989.[46]

Standard GA apply genetic operators such selection, crossover and mutation on an initially random population in order to compute a whole generation of new strings. GA runs to generate solutions for successive generations. The probability of an individual reproducing is proportional to the goodness of the solution it represents. Hence the quality of the solutions in successive generations improves. The process is terminated when an acceptable or optimum solution is found. GA is appropriate for problems which require optimization, with respect to some computationcriterion. The functions of genetic operators are as follows[47]:

1) Selection: Selection deals with the probabilistic survival of the fittest, in that, more fit chromosomes are chosen to survive. Where fitness is a comparable measure of how well a chromosome solves the problem at hand.

2) Crossover: This operation is performed by selecting a random gene along the length of the chromosomes and swapping all the genes after that point.

3) Mutation: Alters the new solutions so as to add stochasticity in the search for better solutions. This is the chance that a bit within a chromosome will be flipped (0 becomes 1, 1 becomes 0) [47]

Genetic algorithm is based on the idea of survival of the fittest and the greedy approach and performs very well global search with less time. The GA works as follows:

- ✓ An initial population is created. A Population is a group of individuals (Chromosomes) and represents a candidate solution. A Chromosome is a string of genes.
- ✓ Select chromosomes with higher fitness.
- ✓ Crossover between the selected chromosomes to produce new offspring with better higher fitness
- ✓ Mutate the new chromosomes if needed.
- ✓ Terminate when an optimum solution is found. This generational process is repeated until a termination condition has been reached. Common terminating conditions are:
  - o A solution is found that satisfies minimum criteria
  - o Fixed number of generations reached
  - o Allocated budget (computation time/money) reached
  - o The highest ranking solution's fitness is reaching or has reached a plateau such that successive iterations no longer produce better results
  - o Manual inspection
  - o Combinations of the above Ghosh et al. proposed an alg[47]

Algorithm 3. Algorithm of genetic algorithm [75]

1. [Start] Generate random population of n chromosomes (suitable solutions for the problem)
2. [Fitness] Evaluate the fitness f(x) of each chromosome x in the population
3. [New population] Create a new population by repeating following steps until the new population is complete
   - 3.1 [Selection] Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected)
   - 3.2 [Crossover] With a crossover probability cross over the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents.
   - 3.3 [Mutation] With a mutation probability mutate new offspring at each locus (position in chromosome).
   - 3.4 [Accepting] Place new offspring in a new population
4. [Replace] Use new generated population for a further run of algorithm
5. [Test] If the end condition (for example number of populations or improvement of the best solution) is satisfied, stop, and return the best solution in current population

1. [Loop] Go to step 2

3.2 Particle swarm optimization algorithm

PSO is a well-known nature inspired algorithm. This method was originally proposed by Kennedy et Eberhart [48, 50] in 1995. The algorithm is a stochastic population based search method inspired by the collective behavior of birds . It is a simple optimization algorithm, effective and comprehensive that can solve new real life problems. Therefore, it is the most used optimization algorithms in the recent years. To understand the whole idea behind this algorithm which is supposed to be simulated in PSO algorithm, assume the following scenario: a group of birds is randomly searching food in an area. There is only one piece of food in the search area . The birds do not know where the food is. But they know how far the food is, and they peers' positions. So what's the best strategy to find the food? An effective strategy is to follow the bird which is nearest to the food. PSO learns from the scenario and uses it to solve the optimization problems. In PSO,each single solution is like a "bird" in the search space, which is called " particle ". All particles have fitness values which are evaluated by the fitness function to be optimized, and have velocities which direct the flying of particles. The particles through the problem space by following neighbored with the best solutions [49].

The algorithm starts a search process with a random population, a set of candidate solutions, where each solution was known as " particle ". This latter is represented by velocity value and position in the search space which are updated as follows:

$$V'id = \omega Vid + c1 \_ rand(\,) * (Pid - Xid) + c2 * rand(\,) * (Pgd - Xid) \quad (3.1)$$

$$X'id = Xid + V'id \quad (3.2)$$

Where X'id and Xid represent the current and previous position of the id^{th} particle, V'id and Vid the current and previous velocity of the id^{th} particle, Pid and Pgd are the individual best position and the global best solution found by the swarm respectively. $0 \leq \omega < 1$ is an inertia weight which determines how much the previous velocity is preserved. Finally, c1 and c2 are acceleration constants. Within the search process, particles move based on their velocity, actual position,

the best position found in the above iterations and the global best solution found by the swarm. This movement makes the particles update their velocity and position at each iteration related to equations 3.1, 3.2. Algorithm 4 shows the pseudo code of Particle swarm optimization algorithm[49].

Algorithm 4. The PSO algorithm[49]

```
1: Initialize the particle population randomly
2: while maximum iterations or minimum criteria is not attained do
3: Calculate fitness values of each particle
4: Update local best if the current fitness value is better than actual best
5: Determine best neighbor solution for each particle: choose the particle with
       the best fitness value of all the neighbor's
6: for each particle do
7: Calculate particle velocity according to 3.1
8: Update particle position according to 3.2
9: end for
10: end while
```

3.3 Bat algorithm

Bats, also named blind mice in English,They represent 20% of all classified mammal species in worldwide. Bats usually live in colonies. They are neither birds nor mice. The world's largest urban bat colony resides in Austin Texas. These animals use a very advanced biological system that utilizes a decentralized decision making and coordinated motion in order to y toward another point in the space as well as search prey. Bat colonies are migratory, they migrate when the food supply becomes depleted in their current environment. However, some species of bats choose to hibernate when the food supply runs low. Microbats are one of the hundred's species of bats. They use echolocation as a perceptual system to detect prey, avoid obstacles, and locate their roosting crevice sin the dark, As shown in Figure 6. Indeed, they are not the only animal that use echolocation to navigate and survive in the nature. There are also many other creatures that uses it. For instance: Dolphins use sound production and reception for navigation, communication, hunting and defense against predator sin darker waters.

Echolocation system can be defineed as an ultrasonic sound emitted by bats. The nervous system of bat can produce a model of his surrounding surface by comparing the outgoing pulse with returning ones echoes, as well as, discrimination different kind of insects in its environment. Typically, micro-bats can emit about 10 to 20such sound bursts every second, and the rate of pulse emission can be sped up to about 200 pulses per second when homing on their prey..[8].

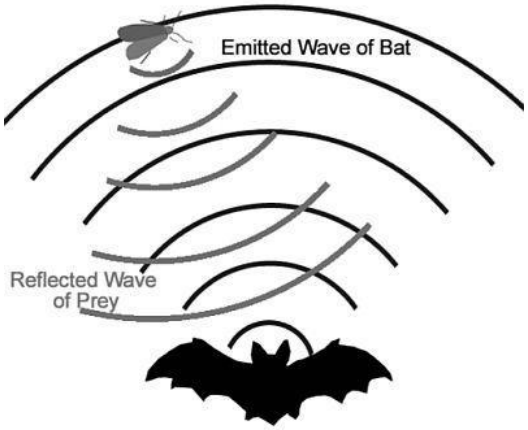


Figure 6: A bat use echolocation to determine prey

Original Bat algorithm

Yang[51] proposed a new and interesting meta-heuristic optimization technique called Bat Algorithm. This technique, that creates a powerful algorithm which could be applied to almost all areas of optimization, was proposed to behave as a band of bats tracking prey/foods using their echolocation. To model this algorithm, Yang proposed few idealized rules to summarize microbats behavior as follows[51]:

- All bats use echolocation to sense distance, and they also `know' the diference between food/prey and background barriers in some magical way;
- Bats y randomly with velocity $v_i at$ position x_i with a fixed frequency f_{min} , varying wavelength λ and loudness A0 to search for prey. They can auto matically adjust the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission r ε [0,1] , depending on the proximity of their target;
- Although the loudness can vary in many ways, we assume that the loudness varies from a large (positive) A0 to a minimum constant value Amin.

Algorithm 5 Original Bat Algorithm

1: Objective Function f(x); x = (x1; x2; :::; xn)T
2: Initialize the bat population xi; i = (1; 2; :::; n) and vi
3: De_ne pulse frequency fi at xi
4: Initialize pulse rates ri and the loudness Ai
5: while t < Max number of iterations do
6: Generate new solutions by adjusting frequency, and updating velocities and locations/solutions [equations 4.1 to 4.3]
7: if rand > ri then
8: Select a solution among the best solutions
9: Generate a local solution around the selected best solution
10: end if
11: Generate a new solution by ying randomly
12: if rand < Ai and f(xi) < f(x_) then
13: Accept the new solutions
14: Increase ri and reduce Ai
15: end if
16: Rank the bats and _nd the current best x_
17: end while
18: Post-process results and visualization.

Regarding the mentioned rules, BA can be summarized as presented in (Algorithm 5)[51]. At the beginning, bat population is randomly initialized with xi; vi; fi for each bat b_i. Let T be the number of iterations. As mentioned in [51], motion of virtual bats, or new solution generation, is performed by updating their frequency, velocity and position according to the following equations:

$$f_i = f_{min} + (f_{max} - f_{\min}\ )\beta \quad , \qquad (4.1)$$

$$v_i^t = v_i^{t-1} + [v_i^{t-1} - x^*]f_i, \qquad (4.2)$$

$$x_i^t = x_i^{t-1} + v_i^t \qquad (4.3)$$

Where β ε [0; 1] is a random generated number drawn from a Gaussian distribution with zero mean and a standard deviation of one and x* is the current best solution which is located after comparing all the solutions among all the bats. For local search, Yang [51] uses a random walk to generate a new solution for each bat b_i. First, a solution is selected among the current

best solutions, then the random walk is applied on the bats that have their rates smaller than the random rate $rate$ as follows:

$$x_{new} = x_{old} + \epsilon A^t \qquad (4.4)$$

Where ε ∈ [-1; 1] is a random number and A^t is the average loudness of all the bats at time t. At each iteration of the algorithm, the loudness A_i is reduced and the rate r_i is increased as follows:

$$A_i^t = \alpha A_i^{t-1} \qquad (4.5)$$
$$r_i^{t+1} = r_i^0[1 - \exp(-\Upsilon t] \qquad (4.6)$$

Where α and Υ are constants. At the initialization step of the algorithm, each bat has a di_erent random loudness A_0 which is in [1,2] and random rate r_0 which is in [0,1] as mentioned in [51].

3.4 bee swarm optimization algorithm

Bees swarm is one of the most important swarms in the nature. These swarms use collective intelligence to adapt to its environment changes. Bees have a so- phisticated navigation system which helps them to detect new hives accomplish their various biological tasks such as foraging and honey production. Bees collect nectar from ower patches as a food source for the hive. The hive sends out scout's that locate patches of owers, who then return to the hive and inform other bees about the _tness and the location of a food source via a waggle dance. The scout returns to the ower patch with follower bees. A few scouts continue to search for new patches, while bees returning from ower patches continue to communicate the quality of the patch[81].

3.5.Penguins search optimization algorithm

Penguins search optimisation algorithm (PeSOA) is a new swarm based meta-heuristic algorithm which was proposed in [76]. The PeSOA algorithm has been used to solve combinatorial problems such as automotive safety integrity levels allocation [77], capitated vehicle routing problem [78] and optimal spaced seed finding [79]. The dietary behaviour of penguins may be explained by economic reasoning: it comes to a profitable food search activity when the gain of energy is greater than the expenditure required to obtain this gain. Penguins, behaving along the line of foraging predators, must extract information about the time and cost to get food and the energy content of prey in order to choose the course for making their next dive. PeSOA is inspired by the penguins' hunting behavior.

Algorithme 6 :PeSOA

1 Generate random population of P solutions (penguins) in groups ;
2 Initialize the probability of existence of fish in the holes and levels;
3 For i=1 to number of generations;
4 For each individual i ∈ P d o
5 While oxygen reserves are not depleted do
5.1 Take a random step.
5.2 Improve the penguin position using Eqs. (1)
5.3 Update quantities of fish eaten for this penguin.
6 End
7 End
7.1 Update quantities of eaten fish in the holes, levels and the best group.
7.2 Redistributes the probabilities of penguins in holes and levels (these
8 probabilities are calculated based on the number of fish eaten).
8.1 Update best-solution
9 End

All Penguins (i) represent a solution (Xi)are distributed in groups, and each group search food in defined holes (Hj) with differences levels (Lk). In this process penguins sorted in order to their groups and start search in a specific hole and level according to food disponibility probability (Pjk).In each cycle, Accordingly, the position of the penguin with each new solution is Adjusted as follows:

$$D_{new} = D_{LastLast} + rand() |X_{LocalBest} - X_{localLast} \quad (1)$$

Where Rand() is a random number for distribution; and we have three solution, the best local solution ,the last solution and the new solution. the calculations in update solution equation (equation 1) are repeated for each penguins in each group, and after several plunged , penguins communicate to each other the best solution witch represented by the number of eaten fish, and we calculate the new distribution probability of holes and levels.

3.6 Whale Optimization algorithms:

In 2016, Mirjalili et al. proposed a new swarm-based nature meta-heuristic inspired by hunting behavior of whales humpback, which considered as the biggest mammals in the world, termed Whale Optimization Algorithm [9]. More precisely, the algorithm mimics the bubbles-net feeding in the foraging behavior of the humpback whales. The bubbles-net formed when the whale swims in a 9-shipped path. Fig.7 shows the Bubble-net feeding behavior.

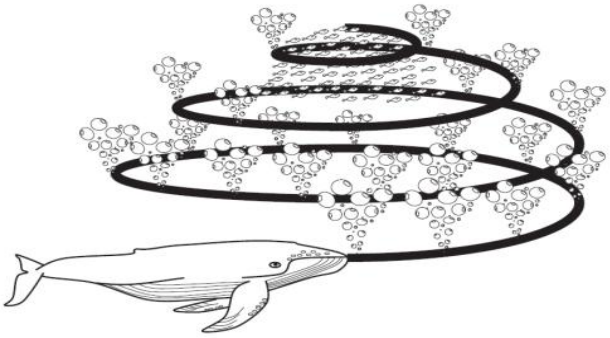


Figure. 7: Bubble-net behavior of humpback whales

As described in Mirjalili et al. paper [9], the algorithm has mainly three phases, Encircling prey, Bubble-net attacking method and Search for prey. These three phases award a good trade-off between exploitation and exploration in the algorithm. Therefore, WOA proves its efficiency in face of other swarm inspired meta-heuristics. The mathematical model of WOA is described as follows:

To hunt a prey, humpback whales first encircle it. Eqs. 1 and 2 can be used to mathematically model this behavior .

$$\vec{D} = \left|\vec{C}.\overrightarrow{X'}(t) - \vec{X}(t)\right| \qquad (1)$$

$$\vec{X}(t+1) = \overrightarrow{X'}(t) - \overrightarrow{(A)}.\vec{D} \qquad (2)$$

where *t* indicates the current iteration, *X'* represents the best solution obtained so far, *X* is the position vector, In addition, *A* and *C* are coefficient vectors that are calculated as in Eqs. 3 and 4.

$$\vec{A} = 2\vec{a}.\vec{r} - \vec{a} \qquad (3)$$

$$\vec{C} = 2.\vec{r} \qquad (4)$$

where $\vec{a}$ decreases linearly from 2 to 0 over the course of iterations (in both exploration and exploitation phases) and $\vec{r}$is a random vector generated with uniform distribution in the interval of [0,1]. Search agents update their positions based on the best known solution. The solution location is controlled by the adjustments of $\vec{a}$and $\vec{r}$ values.

The hum-pack hunting method is based on shrinking encircling mechanism and a spiral-shaped path toward the prey. The shrinking behavior is formulated as shown in Eq 5.

$$a=2-t\frac{2}{\text{MaxIter}} \qquad (5)$$

where t is the iteration number and MaxIter is the maximum number of allowed iterations. The spiral-shaped path is calculated by the distance between the actual solution and the best position by Eq 6,

$$\vec{X}(t+1)=D'e^{bl}.\cos(2\Pi l)+\overrightarrow{X'}(t) \qquad (6)$$

Where $D'=\left|\overrightarrow{X'}(t)-\vec{X}(t)\right|$ describe the distance of i^{th} whale from the prey(The best solution obtained so far). A random coefficient *p* between 0 and 1 is used to choose between the two mechanisms (shrinking encircling mechanism and the spiral-shaped path) with probability of 50% during the optimization process. So that if p <0,5 the shrinking encircling is used to update the position, else the spiral-shaped path will be used.

Whales also have a certain probability of searching for prey when they are constructing bubble-network. Mathematically, searching a prey enhance WOA exploration, This phase is based on the change of *A* coefficient. If *A* exceeds the range of [−1, 1], the distance $\vec{D}$ is updated randomly. At this time, whales will deviate from the original optimal fitness, so that the algorithm has a certain global search-ability, which is formulated as follow:

$$\vec{D}=\left|\vec{C}.\overrightarrow{X_{\text{rand}}}-\vec{X}\right| \qquad (7)$$

$$\vec{X}(t+1)=\overrightarrow{X_{\text{rand}}}-\vec{A}.\vec{D} \qquad (8)$$

Where, $\overrightarrow{X_{rand}}$ is random location information of a whale selected from this iteration. The flowchart of WOA technique is depicted in Algorithm 7.

Algorithm 7. Whale Optimization Algorithm

```
Input Number of MaxIter and Population etc
Initialize the whales population Xi (i = 1, 2, ..., n)
Initialize a, A, C, l and p
Calculate the fitness of each search agent
X*= the best search agent
while (it < MaxIter)
   for each search agent
      if (p < 0.5)
         if (|A| < 1)
            Update the position of the current search agent by the equation (eq 4)
         else if (|A| ≥ 1)
            Select a random search agent (X_rand)
            Update the position of the current search agent by the equation (eq 10)
          End
      else if (p ≥ 0.5)
         Update the position of the current search by the by the equation (6 8)
      End
   End
   Calculate the fitness of each search agent
   Update X* if there is a better solution
   it=it+1
   Update a, A, C, l and p
end while
return X*
```

4 Bio-inspired algorithms for ARM

4.1 Genetic algorithms for ARM

Genetic algorithm based on the concept of strength of implication of rules was presented by Zhou et al. [53]. The properties of independence and correlation of descriptions in rules are taken up for fitness calculation. Genxiang et al. [54] introduced dynamic immune evolution, and biometric mechanism in Engineering immune computing namely immune recognition, immune memory and immune regulation to GA for mining association rules.

Gonzales. E et al. [55] introduced the Genetic Relation Algorithm (GRA) based on evaluating the distances between rules. The distance is calculated using both matching criteria namely complete match and partial match. Genetic algorithm easily leads to premature convergence or takes too much time to converge during evolution process. Hong Lei et al. [56] propose GA where the fitness function is based on predictive accuracy, comprehensibility and interestingness factor. The selection method is based on elitist recombination.

proposed Genar [57] and GAR [58] as first application of genetic algorithm in the field of association rule mining. These algorithms used a bad rule encoding that efect on computation in evaluation step. However, individuals are represented as a list of genes grouped in threes. In each group, the first gene represents the attribute, whereas the remaining genes indicate the minimum and maximum limits of the interval.

Afterwards, several applications of genetic algorithm are presented with better rule encoding. Yan et al. developed a genetic algorithm, called ARMGA [59], to identify association rules without specifying minimum support. Within this algorithm each individual represents a rule. Moreover, it used separator to distinct between rule antecedent and consequence. To generate new rules ARMGA used a simple crossover and mutation operations. The main inconvenience with this algorithm is that generates invalid chromosomes and produces many rules. Quantminer [60] is a genetic algorithm for mining quantitative association rules.

In this work, individuals represent rules and the algorithm evolves to search for the best solution.

4.2 Particle Swarm Optimization for ARM

In [61], Kuo et al. proposed a particle swarm algorithm to detect association rules, called PSOARM. This proposal de fitnes through two main parts: preprocessing and mining. The first part calculates the fitness of particle swarm, whereas in the mining part, PSO is used to mine the rules. In this algorithm the neighbors space is determined by moving the particles in front and rear. This approach gives better results in-face-of AGA algorithm which uses the same strategy to generate neighbors. Sarath et al. [62] provided binary particle swarm optimization based association rule miner called (BPSO). This algorithm generates the best M association rules without specifying the minimum support and confidence, where M is a performance threshold. The quality of extracted rules is measured by a fitness function defined as the product of support and confidence. Moreover, in [63] Binary Particle Swarm Optimization (BPSO) was used to extract a fuzzy association rule from a transactional database. In [64] association rules are optimized by using improved particle swarm optimization algorithm (PSO Algorithm) which is classical PSO algorithm with additional operator in the forms of mutation of genetic algorithm. This operator is used after the initialization phase of PSO algorithm. Firstly, different frequent item sets are generated by the standard Apriori algorithm, then improved PSO algorithm is applied on these generated

association rules for optimizing them. In [65], we found a review on application of Particle Swarm Optimization in Association Rule Mining.

2.4.3 Bat algorithm for association rule mining :

Bat-based algorithm (BA) for association rule mining (BAT-ARM). this algorithm aims to maximize the fitness function to generate the best rules in the defined dataset starting from specific minimum support and minimum confidence. The efficiency of the proposed algorithm is tested on several generic datasets with different number of transactions and items.[8]

Association rule mining is one of the most attractive problem in NP-class[66]. In the proposed algorithm at each iteration, each bat i of the n bats generates a new solution starting from rule that contains k bits (items), So the number of modifed bits is the frequency ,velocity. In the worst case the frequency will be equal to Fmax and the velocity is 0, then the items will be changed.

2.4.4 Bee swarm optimization algorithms for ARM

The authors of [67] developed a new approach inspired from bees behavior and based on bee swarm optimization algorithm called BSO-ARM. The results of this approach show that BSO-ARM performs better than all genetic algorithms. As an extension of their work, the authors present an amelioration to BSO-ARM in, where three strategies to determine the search area of each bee are proposed (modulo, next, syntactic). These improvements yield quality to the rules extracted by BSO-ARM. The results highlighted that this approach outperforms some other existing algorithms in terms of fitness criterion. But unfortunately, the algorithm takes more CPU time. The same authors present another Hybrid approach called (HBSO-TS) in for mining association rules based on Bees swarm and Tabu search algorithms. Where, BSO will browse the search space in such a way to cover most of its regions and the local exploration of each bee is computed by tabu search. The results show that HBSO-TS extracts useful rules in reasonable time. An improvement for HBSO-TS was proposed in where the neighborhood search and three strategies for calculating search area are developed. Recently, bees swarm optimization meta-heuristic for the ARM was implemented in the GPU which improves both quality and time.

2.4.5 Penguins Pe-ARM algorithm

Starts with generating a random population of penguins(each penguin represents a rule). This population is divided into groups, each group contains a variable number of penguins which is updated according to the penguins' health. The division of the initial population is based on the amount of overlapping between population rules. At first, a random penguin (Pr) is selected (will be the center of the first group) and all penguins that have a distance (amount of overlapping) from Pr less than the (Min-distance) will be added to this group. The Min-distance is equal to the average of distances between any two rules in the entire population. A new group is created if all other remaining penguins have a distance from Pr greater than Min-distance.

The diversification generation strategy is used to generate K diversified groups in the initial penguin population. Pe-ARM starts with a population distributed in K groups, and each group is placed in a separate region with a maximum distance from one to another. The purpose is to start the search with a set of diversified initial solutions which have contrasting features benefiting future solution improvement and to control the non visited region in the coming iterations.[80]

5. Conclusion

In this chapter, we saw an overview of optimization problems, and biology-inspired algorithms for optimization problems, including genetics and particle swarm optimization algorithms, bat, bee and whale swarm algorithms.

Also that we presented the latest algorithms applied to the arm field. Our work is based on the whale algorithm , and it is one of the most recent and most effective algorithms , which willbe described in Chapter Three.

Chapter 03 : Whale optimization algorithm for association rule mining

1. Introduction :

In this chapter, we propose a Whale algorithm (WA) for association rule mining (Whale-ARM). Our algorithm aims to minimize the fitness function to generate the best rules in the defined dataset starting from specific minimum support and minimum confidence. The eficiency of our proposed algorithm is tested on several generic datasets with different number of transactions and items.

2. PROPOSED METHOD

Our proposal achieves a noval miner for rule miner based on a whale optimization algorithm named WO-ARM. It aims to extract the most trustworthy association rules in less time and less computational needs. In this section, the used database layout, encoding, and fitness function are highlighted. Furthermore, the modified whale algorithm will be described.

1) Database layout

Database presentation has a real effect on time and resource consumption due to a large volume of sales in such a database Also, the number of scans over the any algorithm execution will influence directly on execution time. Generally, transnational database can be represented in horizontal, vertical and bitmap representation [68]. Therefore, the vertical layout was chosen for our approach, Because item X's support is the tidset dimension, also, To calculate the item-set A{X, Y} support, Tid-set of A needs to be defined as the intersection of X Tid-set and Y Tid-set. Fig.8 shows an instance of layout transformation from horizontal to vertical layout.

2) Rule Encoding

In the literature, several representations of the rule exist to mine association rules using genetic algorithms or meta-heuristic algorithms.

T_id	Item
1	A,B,C
2	A,C
3	B,C

(A) HORIZONTAL LAYOUT

T_id	A	B	C
1	1	1	1
2	1	0	1
3	0	1	1

(B) VERTICAL LAYOUT

Figure. 8: Database representation

The main two, as discussed in [69], are Pittsburgh and Michigan, where the first consider a a set of solutions as one chromosome, whereas, within the second, each chromosome indicates one solution. Our work opts for the last codification. In which we code each rule (solution) X as a vector contains *J* items, whereas, *J* presents the items number in the data-set. Where the victor is coded as follows:

- S[i] = 1 if the i^{th} item exist in the *if statement* (antecedent),
- S[i] = 2 if the i^{th} item exist the *then statement* (consequence),
- S[i] = 0 when the item will not appear in the rule.

For instance: let I = {i_1, i_2, ..., i_{10} } be a set of items : the rule i1 , i5 ⇒ i6 , i2 , i7. is coded as X1 = {1, 2, 0, 0, 1, 2, 2, 0, 0, 0} Fig.9 shows a sample rule coded.

1	2	0	0	1	2	2	0	0	0

Figure.9: Rule Encoding

3) fitness function

As reported in the background section, in the ARM issue, if support and confidence of such rule satisfy user threshold then the rule is accepted. The suggested algorithm aims to maximize the objective function, which supervises uppermost rules extraction. If α and β are two observational parameters utilized for weighting amid the utilized measures inside the objective function, which showed as follow:

$$f(x) = \begin{cases} \alpha.support + \beta.confedence & if acceptedruler \\ -1 & otherwise \end{cases} \quad (9)$$

4) Algorithm description

Algorithm 8 illustrates the pseudo-code of WO-ARM. That has three principal stages. Which are outlined as follows:

- ***Data-set preprocessing:*** as mentioned earlier in this section, our proposal is based on a vertical data-set layout. In which less computational necessaries needed to calculate the support and confidence of each generated rule—also, no whole database scan to complete the computation process. The central fact of this choice is shown in the time response; for these reasons, a preprocessing step in required to convert from horizontal to vertical representation.
- ***Parameter initialization:*** Firstly, all whales are initialized by arbitrary rules, and initial values attributed to all other whale algorithm parameters. After the fitness function for each rule is calculated, and the best rule is chosen and affected to *X**. This phase is entirely randomized.
- ***Rules extraction:*** The same concepts are reused from the original whale optimization algorithm[9], in which the authors use each whale as a candidate solution that will be improved toward the optimal solution. Our proposal assumes that each whale is a nominee rule that includes n elements, where n denotes Items number within the transnational database. The general process of obstetrics a new rule is based on changing items values in each rule based on whale optimization algorithm process (Encircling prey, Bubble-net attacking method, and Search for prey). Next, the rule will be validated to our encoding using a simple way based on odd or even number, such as: if the Item is odd, it belongs to the rule antecedent, else if it is even it belongs to the rule consequence. Otherwise, the Item is not in the rule, which will be 0. Afterword, the algorithm calculates the fitness for each whale and replaces the optimal solution X* by the new one, and this search process will be reiterated until the maximum number of iterations is attained.

5) Algorithm complexity

According to algorithm 8, WO-ARM has a simple structure that is similar to WOA. According to ARM problem which is an NP-hard problem[70]□. There is a little difference in the WO-ARM algorithm complexity by adding a number of Items changed in a rule which represents a solution. Therefore, in worst case the WO-ARM complexity is *O(n* Max_iterations*2*J)*, where *n* is the whales number, Max_iterations is the iterations number and J is the Item number in the transactional database.

3. EXPERIMENTATION AND RESULTS

In order to demonstrate the performance of our method, we did various tests on the recommended algorithm, WO-ARM. This part describes the utilized datasets. Afterword, a comparison recently developed similar approaches is provided.

1) Benchmark and setup description

In order to evaluate our proposal performance, the study utilized various data-sets,which are famous and commonly real-world in data mining, in many tests, taken from Frequent and mining data-set Repository[71]□, Bilkent University Function

Algorithm 8. Whale optimization algorithm for association rule mining

Data-set preprocessing
Input Number of MaxIter and Population, minsupport, minconfedance
Initialize the population Xi (i = 1, 2, ..., n)
Initialize a, A, C, L and p
Compute the fitness function of each search whale
X*= the best rule
While(t<=MaxIter)
    Update a, A, C, L, and p
    For all whales in the population do
        If (p<0.5)
            If (|A|<1)
                For each Item in the solution Xi
                Update Item by using equation (eq 8)
            Else if(|A|=1)
                Select a random Item in Xi
                Update Item using equation (eq 14)
            End if
        Else if (p ≥ 0.5)
            For each Item in the solution Xi
            Update Item by using equation (eq 11)
        End if
        For each Item in the solution Xi
            If the Item is odd, it belongs to the antecedent, Otherwise, it belongs to the consequence
    End for
    Calculate the fitness of each search agent
    Update X* if there is a better solution
    it=it+1
End while
Return X*

Approximation Repository[72]□. Table.5 shows the different datasets included in our tests. The data-sets vary from one to the other in terms of transaction size, item number, and the overage items number per transaction. As example, Chess data-set includes 3196 activities with 75 elements, while each transaction contains an average of 37 items, unlike the mushroom data-set, which more significant in terms of transactions and items, whereas it has just 23 items per transaction.

This section describes the datasets and tests setup. after, the outcomes achieved will be given. The last section will show a comparative repport beside diverse advanced optimization

techniques in the field of ARM. Moreover, a comparative study to exact algorithms in terms of memory consumption will be illustrated.

Note: All algorithms in our study are written in JAVA and all tests were conducted on a machine Intel core I5 with 4Go ram running on Linux Ubuntu.

Data-set	Transactions size	Item size	Average size
Basketball	96	5	8
Bodyfat	252	15	8
IBM - standard	1,000	20	20
Quak	2,178	4	5
Chess	3,196	37	37
Mushroom	8,124	23	23

Table 5: Description of experimental benchmark

2) Stability study

In this section, we focus on the stability of our proposal (WO-ARM). In other words, we study how deals WO-ARM with objective function and CPU time in terms of redundancies. On the other hand, how deals with whales number changing with objective function and CPU time. These tests aim to extract the best parameters (Number of population and Maximum number iteration), that can reach the best results of our algorithm. In this study, we used four datasets with an average size of transactions, which are IBM-standard, Quak, Chess, and Mushroom. We execute whale optimization algorithm for ARM 20 times, and the average results are taken, the support and confidence thresholds fixed to 0.2 and 0.5, respectively.

Table.6 shows the outcomes obtained by our tests. That present the performance of the proposed algorithm WO-ARM in terms of the iterations number, which is changed. Change regularly from 100 to 1000 iteration. These results are obtained with a fixed number of whales to 30 whales. In terms of fitness, we can observe that our proposal achieves its best results at 500 iterations. With different datasets except for IBM-Quest-standard in which the best result was obtained at 100 iterations. On the other hand, we can note that CPU time growing with the iterations increment, which is a natural behavior of each swarm-based algorithm.

#Iter	IBM-STD		Quak		Chess		Mushroom	
	Time	Fitness	Time	Fitness	Time	Fitness	Time	Fitness
100	0,5	1	0,6	0,55	1	0,91	5,2	0,74
200	1,2	1	1	0,72	3	0,83	10	0,72
300	1,6	1	1,7	0,73	4,7	0,89	19	0,87
400	1,9	1	2,6	0,86	6,8	0,89	24	0,86
500	2,2	1	2,9	0,99	7,9	0,88	27	0,97
600	3,4	1	4,5	0,98	11,2	0,92	31	0,98
700	3,7	1	4,7	0,99	12,5	0,96	38	0,98
800	3,9	1	4,9	1	13,9	0,96	42	0,98
900	4,1	1	5,5	1	16,5	0,95	47	0,99
1000	4,4	1	6,8	1	19,3	0,96	63	1

TABLE 6: Performance of the WO-ARM with different numbers of iterations.

On the opposite side, the number of whales in the population also influences the stability of the algorithm. With this in mind, we repeated our tests. Whereas, this time by fixing the maximum iterations number, and change the agents' number in the population, that changed from 10 to 50 regularly. Table. 7 illustrates the results achieved by our algorithm. From the outcome, it is noted that the best fitness in Quack, Chess, and Mushroom datasets is achieved within 30 whales and after this number, almost the same fitness function is obtained. With IBM-Quest-standard dataset the best fitness obtained with ten whales because it has the smallest number of transactions which makes it simpler in exploration than other datasets .

#pop	IBM-STD		Quak		Chess		Mushroom	
	Time	Fitness	Time	Fitness	Time	Fitness	Time	Fitness
10	0,6	1	0,7	0,68	2,8	0,61	7	0,84
20	1,7	1	2,2	0,94	4,2	0,89	20	0,90
30	2,3	1	2,8	0,99	8,9	0,90	27	0,97
40	3,7	1	4,5	0,99	14,6	0,91	39	0,98
50	5,5	1	5,9	1	16,5	0,91	69	1

TABLE 7: Performance of the WOA-ARM with different numbers of whales .

These promising results in terms of rules quality can be explained the good trad-off between exploration and exploitation in the WOA which uses a shrinking encircling mechanism and the spiral-shaped path mechanisms in exploitation and searches the prey for exploration of the search space.

3) Comparative study to other approaches

To well place our method against other methods designed in the literature for ARM. In this section, we present a comparative study that focuses on CPU time, rules quality, and memory consumption. This comparison divided into two main steps, firstly we compare our method in-face-of single-objective optimization approaches in terms of CPU time and rule quality, and secondly, the WO-ARM compared to exact methods in terms of memory consumption. To make this comparison fair, we use the same machine for all algorithms and use the best parameter for each one. For WO-ARM we fixed the maximum number of iterations to 500 and whales number to 30.

1) In-face-of single-objective optimization techniques

The results of WO-ARM were analysed facing to the following four well-known algorithms:

- Penguins Search Optimization Algorithm for Association Rules Mining (Pe-ARM) [73],
- Bees swarm optimization algorithm for ARM (BSO-ARM) [7],
- Multi-swarm bat algorithm for ARM (MSB-ARM) [74],

- Bat algorithm for Association Rule Mining (BAT-ARM) [8].

Results show the average of 20 execution for each algorithm. Table.8 illustrates the out-comes of each algorithm in terms of time consumption with six datasets with different sizes. It is observed that WO-ARM outperforms the other algorithms with the majority of datasets. Except with mushroom where BSO-ARM has less runtime compared to WO-ARM within 0,9 seconds which can be negligible. These outcomes can be explained by the whale optimization algorithm complexity, which inherited by whale optimization algorithm for association rule mining (WO-ARM).

	Pe-ARM	**BSO-ARM**	**MSB-ARM**	**BAT-ARM**	**WO-ARM**
Basketball	1,5	3,36	4	7	**0,7**
Bodyfat	2,88	5,7	11	14	**1,3**
IBM-standard	1,68	1,92	13	19	**1,2**
Quak	3,35	4,5	40	76	**2,3**
Chess	4,92	5,1	13	141	**4,7**
Mushroom	10,68	**9.1**	144	341	10

TABLE 8: Comparing our approach to existing approaches W.R.T time (sec)

Indeed, the runtime is not enough to judge such a swarm-inspired algorithm. Solution quality is an essential factor that influences on decisions as a good algorithm or not. With this in mind, we compare our proposal in-face-of mentioned above algorithms in terms of fitness function value, and the outcomes are illustrated in Table.9.

Again, WO-ARM proves its superiority against other algorithms with the majority of datasets which can be explained by the excellent trad-off between intensification and diversification in the whale optimization algorithm. Also, thanks to the shrinking encircling mechanism and the spiral-shaped path mechanisms that guide the algorithms to choose the best neighbour in local search.

2) In-face-of exact methods methods

As highlighted in the introduction, one of the most challenging drawbacks in exact algorithms for association rule mining is memory consumption. Especially with the vast stored data our days.

	Pe-ARM	**BSO-ARM**	**MSB-ARM**	**BAT-ARM**	**WO-ARM**
Basketball	1	0,92	1	0,81	1
Bodyfat	1	0,73	1	0,54	1
IBM-standard	0,92	0,93	0,84	0,41	**0,94**
Quak	0,91	1	1	0,52	1
Chess	0,89	0,88	0,97	0,92	**0,99**
Mushroom	0,88	0,75	0,68	0,93	**0,97**

TABLE 3: comparing our approach to exsisting approaches W.R.T fitness

To overcome this drawback, optimization algorithms are investigated for ARM in order to discover WO-ARM memory usage. The algorithm tested with four datasets. Moreover, the outcomes are compared to those from exact algorithms (Apriori, FP-growth) and result obtained from the Multi-swarm bat algorithm for ARM (MSB-ARM), and Bat algorithm for Association Rule Mining (BAT-ARM). The results are summarized in Table. 10. The results presents that WO-ARM has less memory usage compared to other algorithms. These results are related firstly to dataset representation which gives less computation in the phase of rule evaluation. Also, whale optimization algorithm complexity influences memory usage.

	Apriori	**Fp-growth**	**MSB-ARM**	**BAT-ARM**	**WO-ARM**
IBM-standard	26,05	26,22	27,2	13,74	**17,6**
Quak	19,12	25,12	21,6	**16,2**	18,01
Chess	225,33	104,55	58,32	48,63	**31,46**
Mushroom	317,58	291,1	256,7	170,29	**52,62**

TABLE 4: Comparing our approach to exact approaches W.R.T memory usage(MB)

4. CONCLUSION

This study provided a new algorithm for whaling improvement , One of the most important results is that it is the value of fitness and time that is superior to the previously studied algorithms.

The experience provided proved our WOA_arm approach to time is fast in implementation, and so we used the horizontal design of the database where you don't hear the waste of time. Through our experiences, we found that Woa is above Bat in terms of the time of implementation. We can see a 90% increase in lead time.

Application graphical interface:

presentation of interface

We have proposed a whale optimization algorithm updating the mine correlation rules, we are able to develop a simplified program.

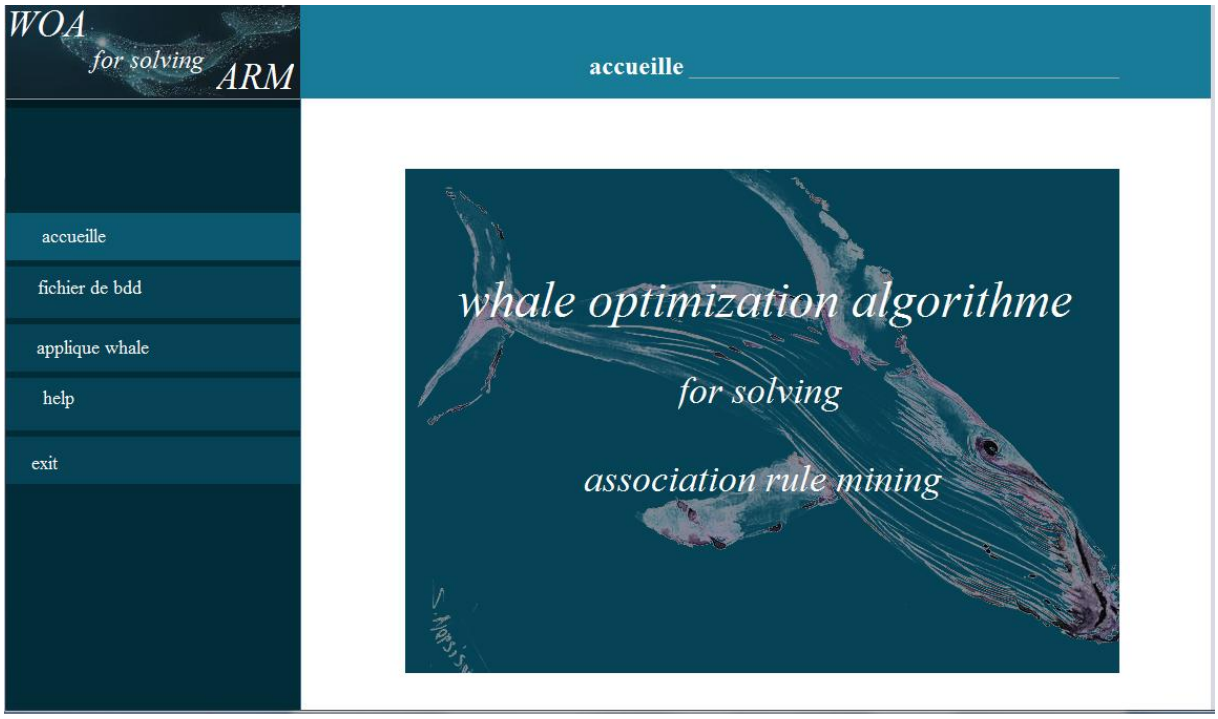


Figure 10: the entry page of the application.

The interface of the program contains two windows, the first window contains extracting the shape in which the figure is studied for the algorithm of the image shown in figure (11) and the second window contains the input of the population information, alpha and beta support and confidence shown in figure (12), as for Figure (10) is the site of the service, and Figure (13) contains the rules of A and the rules of the whale

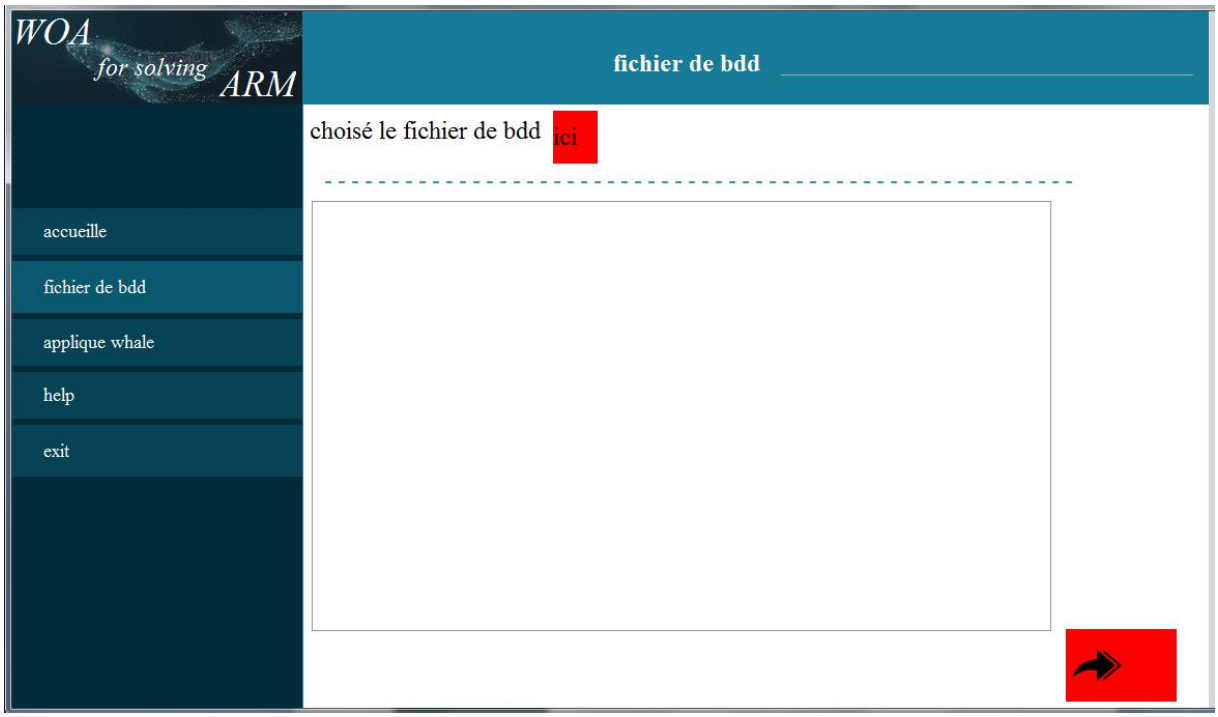


Figure 11: choice of the document for appliqué algorithm.

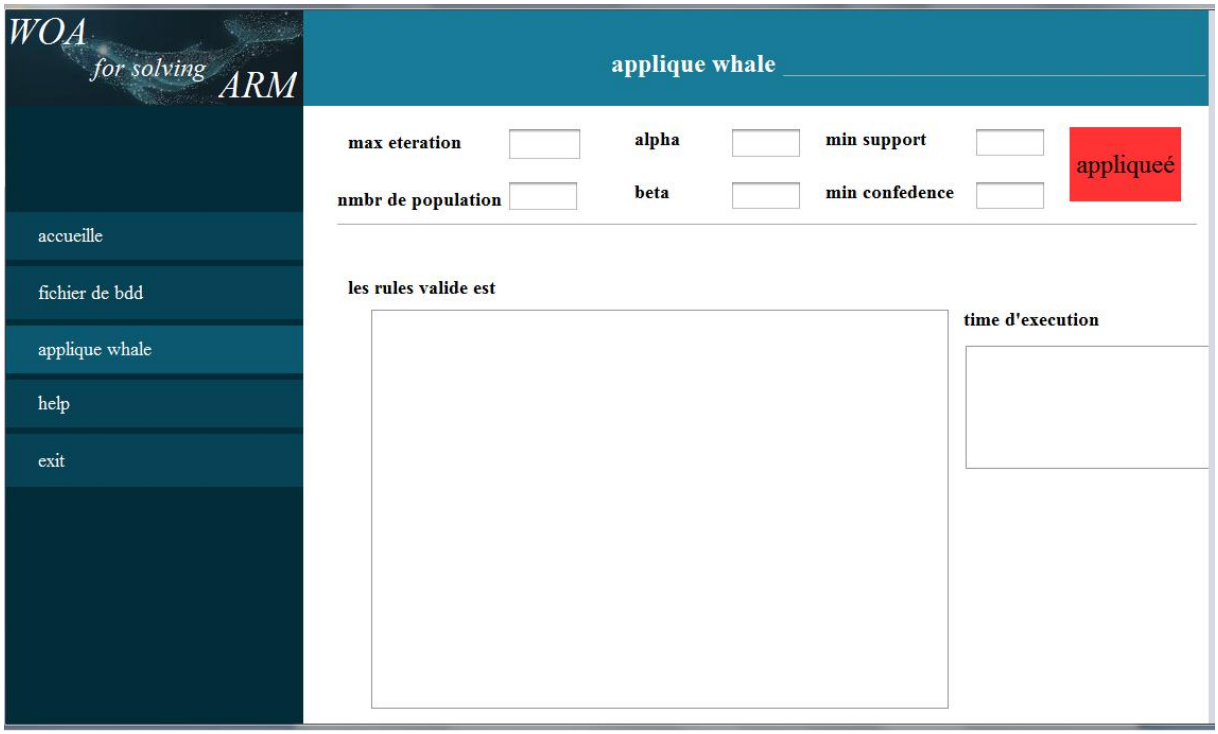


Figure 12: page of application whale.

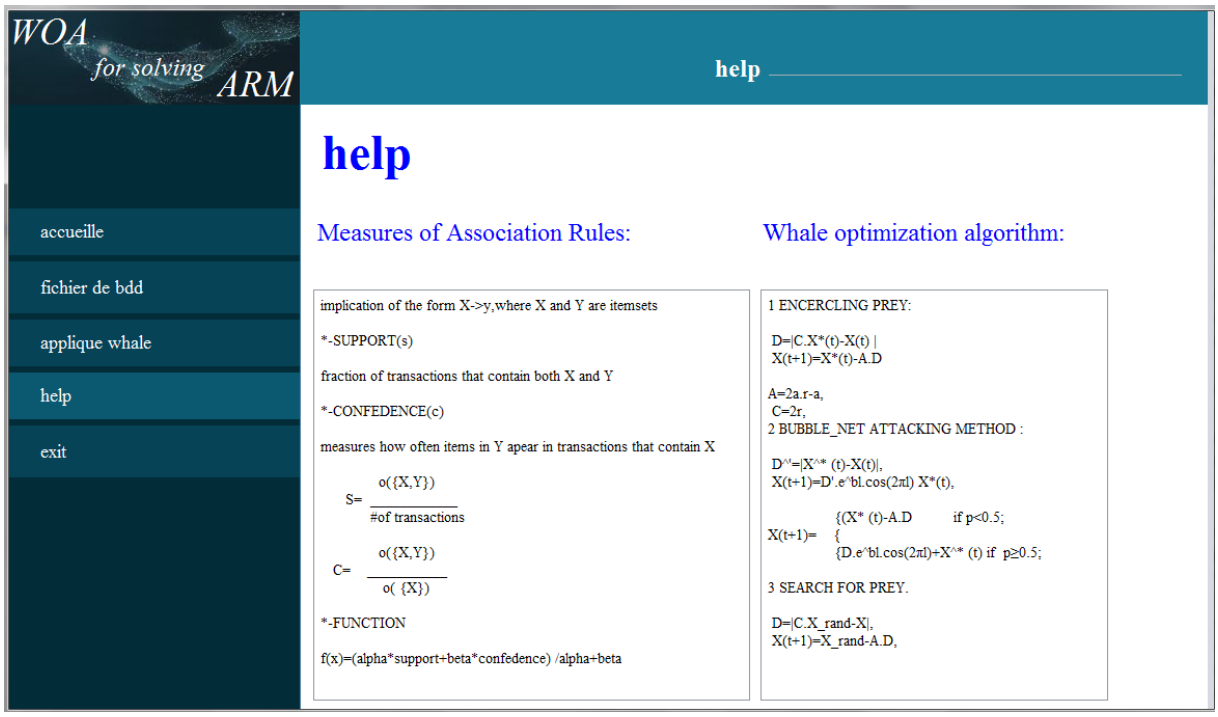


Figure 13 : page of rules.

example for algrithme

The first step is to select the file for the databases, such as what is shown in Figure (14), then fill in the numbers of the empty cells and press the button (arrow) to correct the updated whale algorithm and sample the results of all the valid rules, the best support, the best union, and the maximum fitness such as shown in the figure (15th)

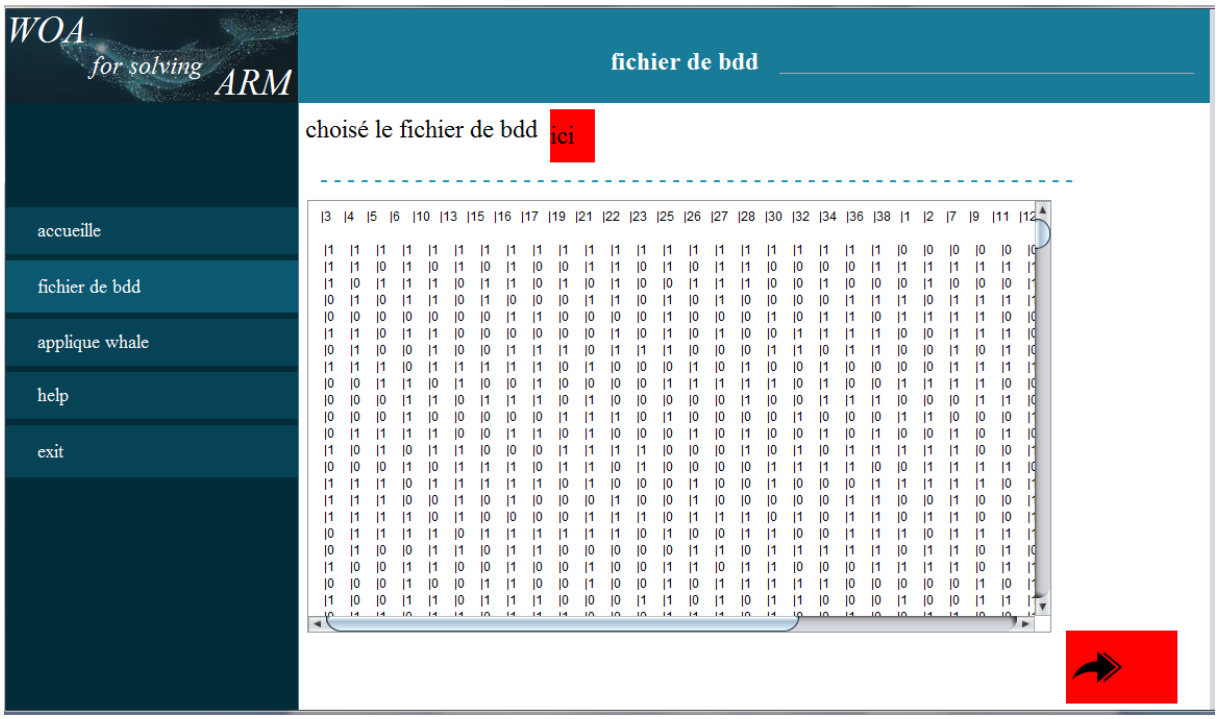


Figure 14:the document of data.

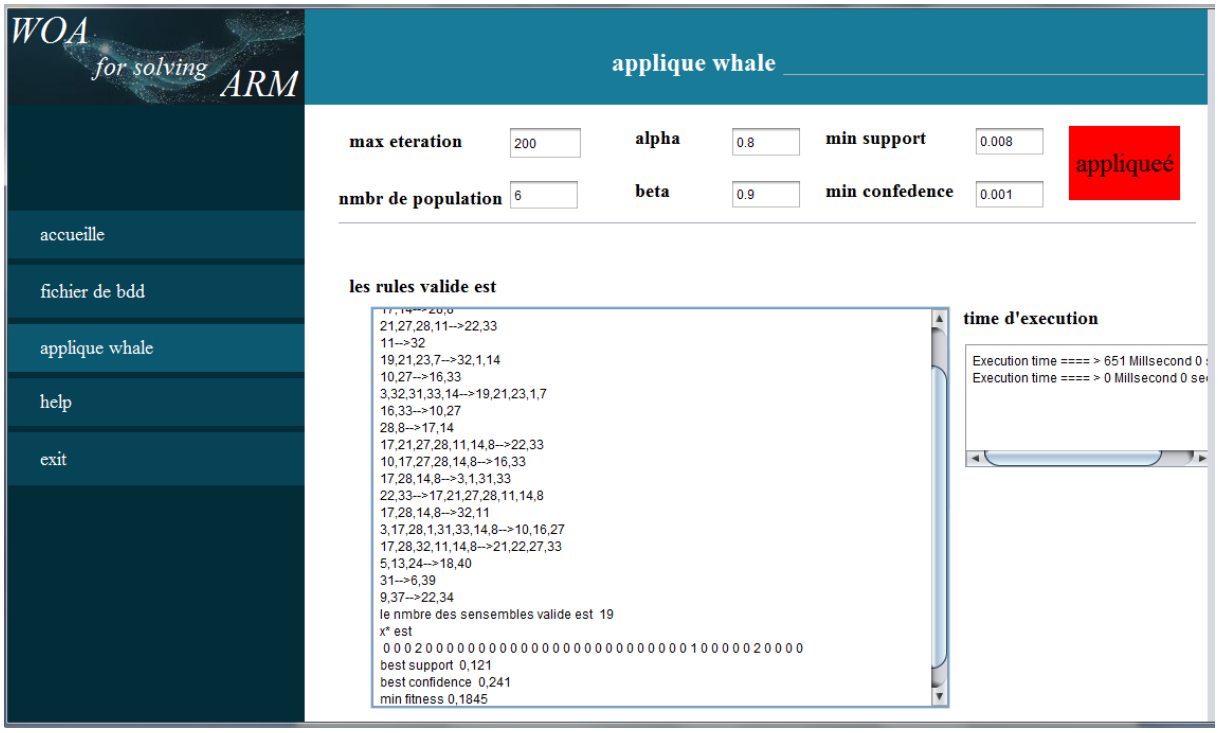


Figure 15:the rules valid and best support and time of execution.

Conclusion general

Conclusion general:

Nowadays, Association rules have widely used to define relationships between items in databases. Nevertheless, association rule mining is an NP-complete problem; time and memory consumption are explosively grown with the number of transactions and items in the database, making rule extraction a challenging problem for exact algorithms. To overcome this challenge, this thesis presented a whale optimization algorithm for association rule mining (WO-ARM). In which, we investigated in the good trad-off between intensification and diversification that distinguished the original whale optimization algorithm based on shrinking encircling mechanism, the spiral-shaped path, and search prey technique. We evaluated the proposed algorithm on six well-known datasets in the field of ARM, and the outcomes are compared to recently developed similar approaches. Results showed the effectiveness of WO-ARM in terms of runtime, quality, and memory consumption. These results are obtained due to the whale optimization algorithm mechanisms. In the near future, we aim to develop our proposition to handle large scale datasets. The improvement will be concertized by the use of parallel execution on Graphical Processing Units (GPU).

REFERENCES

[1] W. J. Frawley, G. Piatetsky-Shapiro, and C. J. Matheus, "Knowledge discovery in databases: An overview," AI Mag., vol. 13, no. 3, p. 57, 1992.

[2] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in ACM SIGMOD Record, 1993, vol. 22, no. 2, pp. 207–216.

[3] R. Agrawal, R. Srikant, and others, "Fast algorithms for mining association rules," in Proc. 20th int. conf. very large data bases, VLDB, 1994, vol. 1215, pp. 487–499.

[4] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in ACM SIGMOD Record, 2000, vol. 29, no. 2, pp. 1–12.

[5] W. Wang and S. M. Bridges, "Genetic Algorithm Optimization of Membership Functions for Mining Fuzzy Association Rules," 2000.

[6] Z. Kou and L. Xi, "Binary Particle Swarm Optimization-Based Association Rule Mining for Discovering Relationships between Machine Capabilities and Product Features," 2018, doi: 10.1155/2018/2456010.

[7] Y. Djenouri, H. Drias, and Z. Habbas, "Bees swarm optimisation using multiple strategies for association rule mining," Int. J. Bio-Inspired Comput., vol. 6, no. 4, pp. 239–249, 2014.

[8] K. E. Heraguemi, N. Kamel, and H. Drias, "Association Rule Mining Based on Bat Algorithm," J. Comput. Theor. Nanosci., vol. 12, no. 7, pp. 1195–1200, 2015.

[9] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," Adv. Eng. Softw., vol. 95, pp. 51–67, May 2016, doi: 10.1016/j.advengsoft.2016.01.008.

[10] F. S. Gharehchopogh and H. Gholizadeh, "A comprehensive survey: Whale Optimization Algorithm and its applications," Swarm Evol. Comput., vol. 48, no. November 2018, pp. 1–24, 2019, doi: 10.1016/j.swevo.2019.03.004.

[11] G. Nalcaci and M. ERMİŞ, "Selective Harmonic Elimination for Three-Phase Voltage Source Inverters Using Whale Optimizer Algorithm," Accessed: Jun. 21, 2020. [Online]. Available: https://avesis.metu.edu.tr/yayin/dd544ccd-4fe9-4180-8ba0-7938df0b3b78/selective-harmonic-elimination-for-three-phase-voltage-source-inverters-using-whale-optimizer-algorithm.

[12] R. K. Saidala and N. R. Devarakonda, "Bubble-net hunting strategy of whales based optimized feature selection for e-mail classification," in 2017 2nd International Conference for Convergence in Technology, I2CT 2017, Dec. 2017, vol. 2017-January, pp. 626–631, doi: 10.1109/I2CT.2017.8226205.

[13] J. Nasiri and F. M. Khiyabani, “A whale optimization algorithm (WOA) approach for clustering,” Cogent Math. Stat., vol. 5, no. 1, Jun. 2018, doi: 10.1080/25742558.2018.1483565.

[14] S. J. Mousavirad and H. Ebrahimpour-Komleh, “Multilevel image thresholding using entropy of histogram and recently developed population-based metaheuristic algorithms,” Evol. Intell., vol. 10, no. 1–2, pp. 45–75, Jul. 2017, doi: 10.1007/s12065-017-0152-y.

[15] Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. From data mining to knowledge discovery in databases. AI magazine 17, 3 (1996), 37.

[16] Zubcoff, J., & Trujillo, J. (2007). A UML 2.0 profile to design Association Rule mining models in the multidimensional conceptual modeling of data warehouses. Data & Knowledge Engineering, 63(1), 44-62.

[17] Srinivas, K., Rani, B. K., & Govrdhan, A. (2010). Applications of data mining techniques in healthcare and prediction of heart attacks. International Journal on Computer Science and Engineering (IJCSE), 2(02), 250-255.

[18] Heraguemi, K. E., Kamel, N., & Drias, H. (2015). Association rule mining based on bat algorithm. Journal of Computational and Theoretical Nanoscience, 12(7), 1195-1200. [4] Branden, C. I., & Tooze, J. (2012). Introduction to protein structure. Garland Science.

[19] Fox, J. Applied regression analysis, linear models, and related methods. Sage Publications, Inc, 1997.

[20] Berkhin, P. A survey of clustering data mining techniques. In Grouping multidimensional data. Springer, 2006, pp. 25–71.

[21] Agrawal, R., Imieli´nski, T., and Swami, A. Mining association rules between sets of items in large databases. In ACM SIGMOD Record (1993), vol. 22, ACM, pp. 207–216.

[22] Rajak, A., & Gupta, M. K. (2008, February). Association rule mining: applications in various areas. In Proceedings of international conference on data management, ghaziabad, india (pp. 3-7).

[23] Zhao, Q., & Bhowmick, S. S. (2003). Association rule mining: A survey. Nanyang Technological University, Singapore, 135

[24] Zaki, M. J., Parthasarathy, S., Ogihara, M., & Li, W. (1997). Parallel algorithms for discovery of association rules. Data mining and knowledge discovery, 1(4), 343-373.

[25] P.-N. Tan, V. Kumar, and J. Srivastava, “Selecting the right objective measure for association analysis,” Inf. Syst., vol. 29, no. 4, pp. 293–313, 2004.

[26] 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.

[27] Brin, S., Motwani, R., & Silverstein, C. (1997, June). Beyond market baskets: Generalizing association rules to correlations. In Proceedings of the 1997 ACM SIGMOD

[28] Piatetsky-Shapiro, G. (1991). Discovery, analysis, and presentation of strong rules. Knowledge discovery in databases, 229-238.

[29] Brin, S., Motwani, R., Ullman, J. D., & Tsur, S. (1997, June). Dynamic itemset counting and implication rules for market basket data. In Proceedings of the 1997 ACM SIGMOD international conference on Management of data (pp. 255-264).

[30] Wakabi-Waiswa, P. P., & Baryamureeba, V. (2008). Extraction of interesting association rules using genetic algorithms. International Journal of Computing and ICT Research, 2(1), 26-33.

[31] Heraguemi, K. E., Kamel, N., & Drias, H. (2018). Multi-objective bat algorithm for mining numerical association rules. International Journal of Bio-Inspired Computation, 11(4), 239-248.

[32] Serban, G., Czibula, I. G., & Campan, A. L. I. N. A. (2006). A programming interface for medical diagnosis prediction. Studia Universitatis" Babes-Bolyai", Informatica, LI (1), pag, 21-30.

[33] Gamberger, D., Lavrac, N., & Jovanoski, V. (1999). High confidence association rules for medical diagnosis pages 42-51.

[34] Branden, C., & Tooze, J. (1991). The structure of spherical viruses. Introduction to protein structure. Garland Publishing, Inc., New York and London, 161-78.

[35] Malerba, D., Esposito, F., Lisi, F. A., & Appice, A. (2003). Mining spatial association rules in census data. Research in Official Statistics. v5 i1, 19-44.

[36] Saporta, G. (2000). Data mining and official statistics. Paper, Chaire de Statistique Appliquée, Conservatoire National des Arts et Métiers, 292.

[37] Wu, R. C., Chen, R. S., & Chen, C. C. (2005, July). Data mining application in customer relationship management of credit card business. In 29th Annual International Computer Software and Applications Conference (COMPSAC'05) (Vol. 2, pp. 39-40). IEEE.

[38] Shaw, M. J., Subramaniam, C., Tan, G. W., & Welge, M. E. (2001). Knowledge management and data mining for marketing. Decision support systems, 31(1), 127-137.

[39] Song, H. S., kyeong Kim, J., & Kim, S. H. (2001). Mining the change of customer behavior in an internet shopping mall. Expert Systems with Applications, 21(3), 157-168.

[40] Eddine, H. K. (2017). A bio-inspired approach for Association Rules Mining (Doctoral dissertation, UNIVERSITY OF SETIF).

[41] Agrawal, R., & Srikant, R. (1994, September). Fast algorithms for mining association rules. In Proc. 20th int. conf. very large data bases, VLDB (Vol. 1215, pp. 487-499).

[42] Han, J., Pei, J., & Kamber, M. (2011). Data mining: concepts and techniques. Elsevier.

[43] Han, J., & Fu, Y. (1999). Mining multiple-level association rules in large databases. IEEE Transactions on knowledge and data engineering, 11(5), 798-805.

[44] : lecture heuristic optimization methods : introduction to optimization university of zagreb https://www.fer.unizg.hr/_download/repository/Lecture_1-2_%5BENG%5D_-_Introduction_to_optimization.pdf .

[45].https://gato-docs.its.txstate.edu/slac/Subject/Math/Word-Problems/Optimization-Word-Problems/Opti

[46]. https://en.wikipedia.org/wiki/Genetic_algorithm

[47]. Ho-Huu, V., Nguyen-Thoi, T., Nguyen-Thoi, M. H., & Le-Anh, L. (2015). An improved constrained differential evolution using discrete variables (D-ICDE) for layout optimization of truss structures. Expert Systems with Applications, 42(20), 7057-7069.

[48] Eberhart, R. C., Kennedy, J., et al. A new optimizer using particle swarm theory. In Proceedings of the sixth international symposium on micro machine and human science (1995), vol. 1, New York, NY, pp. 39{43.

[49] Hu, X., Shi, Y., and Eberhart, R. Recent advances in particle swarm. In Evolutionary Computation, 2004. CEC2004. Congress on (2004), vol. 1, IEEE, pp. 90{97.

[50] James, K., and Russell, E. Particle swarm optimization. In Proceedings of 1995 IEEE International Conference on Neural Networks (1995), pp. 1942{1948.

[51] Yang, X.-S. A new metaheuristic bat-inspired algorithm. In Nature in- spired cooperative strategies for optimization (NICSO 2010). Springer, 2010, pp. 65{74.

[53].Zhou, J., Li, S.-y., Mei, H.-y., Liu, H.-x.: A Method for Finding Implicating Rules Based on the Genetic Algorithm. In: Third International Conference on Natural Computation,, vol. 3, pp. 400–405 (2007)Google Scholar

[54].G. Zhang, H. Chen. : Immune Optimization Based Genetic Algorithm for Incremental Association Rules Mining. In : International Conference on Artificial Intelligence and Computational Intelligence, AICI '09, Volume: 4, Page(s): 341 – 345, 2009.Google Scholar

[55].Gonzales, E., Mabu, S., Taboada, K., Shimada, K., Hirasawa, K.: Mining Multi-class Datasets using Genetic Relation Algorithm for Rule Reduction. In: IEEE Congress on Evolutionary Computation, CEC 2009, pp. 3249–3255 (2009)Google Scholar

[56].Shi, X.-J., Lei, H.: Genetic Algorithm-Based Approach for Classification Rule Discovery. In: International Conference on Information Management, Innovation Management and Industrial Engineering, ICIII 2008, vol. 1, pp. 175–178 (2008)Google Scholar

[57] Mata, J., Alvarez, J., and Riquelme, J. Mining numeric association rules with genetic algorithms. In Arti_cial Neural Nets and Genetic Algorithms (2001), Springer, pp. 264{267.

[58] Mata, J., Alvarez, J.-L., and Riquelme, J.-C. An evolutionary algorithm to discover numeric association rules. In Proceedings of the 2002 ACM symposium on Applied computing (2002), ACM, pp. 590{594.

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

[60] .Salleb-Aouissi, A., Vrain, C., and Nortet, C. Quantminer: A genetic algorithm for mining quantitative association rules. In IJCAI (2007), vol. 7.

[61].Kuo, R. J., Chao, C. M., and Chiu, Y. Application of particle swarm optimization to association rule mining. Applied Soft Computing 11, 1 (2011), 326{336.

[62].Sarath, K., and Ravi, V. Association rule mining using binary particle swarm optimization. Engineering Applications of Arti_cial Intelligence 26, 8 (2013), 1832{1840.

[63].Tayal, K., and Ravi, V. Fuzzy association rule mining using binary particle swarm optimization: Application to cyber fraud analytics. In 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC) (Dec 2015), pp. 1{5.

[64].Agrawal, M., Mishra, M., and Kushwah, S. P. S. Association rules optimization using improved pso algorithm. In 2015 International Conference on Communication Networks (ICCN) (2015), IEEE, pp. 395{398.

[65].Ankita, S., Shikha, A., Jitendra, A., and Sanjeev, S. A review on application of particle swarm optimization in association rule mining. In Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) (2013), Springer, pp. 405{414.

[66].Angiulli, F., Ianni, G., and Palopoli, L. On the complexity of mining association rules. In SEBD (2001), pp. 177{184.

[67].Djenouri, Y., Drias, H., Habbas, Z., and Mosteghanemi, H. Bees swarm optimization for web association rule mining. In Web Intelligence and Intelligent Agent Technology (WI-IAT), 2012 IEEE/WIC/ACM International Conferences on (2012), vol. 3, IEEE, pp. 142{146.

[68] J. Han, J. Pei, and M. Kamber, Data mining: concepts and techniques. Elsevier, 2011.

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

[70] F. Angiulli, G. Ianni, and L. Palopoli, "On the Complexity of Mining Association Rules.," in SEBD, 2001, pp. 177–184.

[71] B. Goethls and M. J. Zaki, "Frequent Itemset Mining Dataset Repository," 2003, [Online]. Available: http://fimi.ua.ac.be/data/.

[72] H. A. Guvenir and I. Uysal, "Bilkent University Function Approximation Repository," 2000, [Online]. Available: http://funapp.cs.bilkent.edu.tr/DataSets/.

[73] Y. Gheraibia, A. Moussaoui, Y. Djenouri, S. Kabir, and P. Y. Yin, "Penguins Search Optimisation Algorithm for Association Rules Mining," CIT. J. Comput. Inf. Technol., vol. 24, no. 2, pp. 165–179, 2016.

[74] K. E. Heraguemi, N. Kamel, and H. Drias, "Multi-swarm bat algorithm for association rule mining using multiple cooperative strategies," Appl. Intell., vol. 45, no. 4, pp. 1021–1033, Dec. 2016, doi: 10.1007/s10489-016-0806-y.

[75].Bajpai, P., & Kumar, M. (2010). Genetic algorithm–an approach to solve global optimization problems. Indian Journal of computer science and engineering, 1(3), 199-206.

[76].Y. Gheraibia and A. Moussaoui, "Penguins search optimization algorithm (pesoa)", in Recent Trends in Applied Artificial Intelligence, pp. 222–231, Springer, 2013. http://dx.doi.org/10.1007/978-3-642-38577-3_23

[77] Y. Gheraibia et al., "Can aquatic flightless birds allocate automotive safety requirements?", in 2015 IEEE Seventh International Conference on Intelligent Computing and

Information Systems (ICICIS), pp. 1–6, IEEE, 2015. http://dx.doi.org/10.1109/IntelCIS.2015.7397214

[78] M. Ammi and S. Chikhi, "Cooperative parallel metaheuristics based penguin optimization search for solving the vehicle routing problem", International Journal of Applied Metaheuristic Computing (IJAMC), vol. 7, no. 1, pp. 1–18, 2016. http://dx.doi.org/10.4018/IJAMC.2016010101

[79] Y. Gheraibia et al., "Penguin search optimisation algorithm for finding optimal spaced seeds", International Journal of Software Science and Computational Intelligence (IJSSCI), vol. 7, no. 2, pp. 85–99, 2015. http://dx.doi.org/10.4018/IJSSCI.2015040105

[80].Gheraibia, Y., Moussaoui, A., Djenouri, Y., Kabir, S., & Yin, P. Y. (2016). Penguins search optimisation algorithm for association rules mining. Journal of computing and information technology, 24(2), 165-179.

[81].Brownlee, J. Clever algorithms. Nature-Inspired Programming Recipes (2011), 436.

abstract

Nowadays with a large number of connected devices, the stored data has grown significantly. The use of the information stored in it to make decisions for the sake of convenience.

Association rule mining directs researchers into extracting relationships between objects in the data.

The aim of this thesis is to propose ways to solve problems related to mining and data mining, in order to extract the largest possible number of valid rules in the shortest possible time.

In this work: We adopted the principle of the whale algorithm and its application in order to solve the association rules of mining

Keywords: *Association rule mining, Whale optimization algorithm, Support,, Confidence, swarm inspired algorithm*

ملخص

في ايامنا هذه مع وجود عدد كبير من الاجهزة المتصلة ، نمت البيانات المخزنة بشكل ملحوظ . اصبح استغلال المعلومات المخزنة فيه لاتخاذ القرار من اجل التسهيل .

يوجه تعدين قواعد الرابطة في الاعتبار الباحثين لاستخراج العلاقات بين العناصر في البيانات .

الهدف من هذه الاطروحة هو اقتراح طرق لحل المشكلات المتعلقة بالتعدين و استخراج البيانات، وذلك من اجل استخراج اكبر عدد ممكن صحيح للقواعدا وفي اقل وقت ممكن .

في هاذا العمل:اعتمدنا على مبدأ خوارزمية الحوت وتطبيقها من اجل حل تعدين قواعد الرابطة

الكلمات المفتاحية : تعدين قواعد الرابطة ، خوارزمية تحسين الحيتان ، الدعم ، الثقة ، مستوحى من السرب

Résume

Aujourd'hui, avec un grand nombre d'appareils connectés, les données stockées ont considérablement augmenté. C'est devenu l'utilisation des informations qui y étaient stockées pour prendre des décisions pour des raisons de commodité.

L'exploration de règles d'association oriente les chercheurs vers l'extraction des relations entre les objets des données.

Le but de cette thèse est de proposer des pistes pour résoudre les problèmes liés au minage et au data mining, afin d'extraire le plus grand nombre possible de règles correctes dans les plus brefs délais.

Dans ce travail: Nous avons adopté le principe de l'algorithme de baleine et son application afin de résoudre Extraction de règles d'association