RÉPUBLIQUE ALGÉTIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE FARHAT ABBAS – SETIF-1
Faculté des sciences
Département d'Informatique

Approche bio-inspirée pour l'extraction des règles d'association

Thèse présenté par:

Heraguemi Kamel Eddine

En vue de l'obtention
du diplôme de *Doctorat 3*$^{\text{ème}}$ *Cycle en Informatique*

Devant le jury :

Moussaoui	Abdelouhab	Président	Prof.	UFA - Sétif-1
Drias	Habiba	Directeur	Prof.	USTHB- Alger
Kamel	Nadjet	Co-directeur	Prof.	UFA - Sétif-1
Baatouche	Mohamed	Examinateur	Prof.	UAM - Constantine-2
Refoufi	Allaoua	Examinateur	MCA	UFA - Sétif-1
Alti	Adel	Examinateur	MCA	UFA – Sétif-1
Mokhtari	Aicha	Invitée	Prof.	USTHB- Alger

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
FARHET ABBAS UNIVERSITY OF SETIF
Faculty of Science
Department of Computer Science

A bio-inspired approach for Association Rules Mining

A Thesis presented by

Heraguemi Kamel Eddine

As a requirement to aim for the degree of:
3^{rd} cycle Doctorate in Computer science

ADVISORS
Pr. HABIBA DRIAS USTHB, ALGIES
Pr. NADJET KAMEL UNIV-SETIF1, SETIF

March 2017

Declaration of Authorship

I, Heraguemi Kamel Eddine, declare that this thesis titled, 'A bio-inspired approach for Association Rules Mining ' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

"A person who never made a mistake never tried anything new."

- Albert Einstein

Abstract

Association Rule Mining (ARM) can be considered as a combinatorial problem with the purpose of extracting the correlations between items in sizable data sets. The numerous polynomial exact algorithms already proposed for ARM become maladjusted for large databases and especially for those existing on the web due to the large number of generated rules and time requirement. Whereas, users cannot exploit all rules obtained by the search process in exhaustive algorithms and can not loss long time when the database increase. Thus, the creation of new frameworks and tools for ARM issue become indispensable. Therefore, this Doctoral dissertation proposes several algorithms for mining association rule from stored data based on swarm intelligence which is a modern and efficient mechanism for solving problems in computer science, engineering, economics and optimization, it is a collective behavior of decentralization and self-organization system. The focal point of this work is devoted to Bat algorithm which is a swarm based algorithm, developed recently. BA already proved its efficiency and superiority in many application fields. Here, we propose three advanced proposals based on bat algorithm to discover the best set of interesting association rules from data, Bat algorithm for mining association rule mining, Multi-swarm Cooperative Bat Algorithm for ARM and multi objective bat-algorithm for ARM. The later, deals with the ARM issue as a multi-objective optimization problem due to the several quality measures of rules that can be maximized at the same time. Whereas, the others deal with ARM as a mono-objective optimization problem that use support and confidence framework to evaluate the extracted rules. All the proposed algorithms in this thesis are evaluated using a proper series of experiments, utilizing several well-known data sets in ARM field and the performance of each proposed approach is evaluated and compared against those of other recently published methods. The results show a clear superiority of our proposals against other approaches in terms of time and rule quality.

Résumé

L’extraction des règles d’association (ARM) peut être considérée comme un problème combinatoire Dans le but d'extraire les corrélations entre les éléments dans des ensembles de données considérables. Les nombreux algorithmes exacts polynômes déjà proposés pour ARM deviennent inadapté aux grandes bases de données et surtout à ceux qui existent sur le Web en raison au grand nombre de règles générées et aux exigences de temps. Attendu que les utilisateurs ne peut pas exploiter toutes les règles obtenues par le processus de recherche dans des algorithmes exhaustifs et ne peut pas perdre beaucoup de temps lorsque la base de données augmente. Ainsi, la création de de nouveaux cadres et outils pour la problématique ARM deviennent indispensables. Par conséquent, cette thèse de doctorat propose plusieurs algorithmes pour la règle d'association minière à partir de données stockées basées sur l'intelligence de l'essaim qui est un mécanisme moderne et efficace pour résoudre des problèmes d'informatique, d'ingénierie, d'économie et d'optimisation, c'est un comportement collectif de décentralisation et de système d'auto-organisation. Le point focal de ce travail est consacré à l'algorithme Bat qui est développé récemment. BA a déjà prouvé son efficacité et sa supériorité dans plusieurs champs d'application. Ici, nous proposons trois propositions avancées basées sur l’algorithme bat pour découvrir le meilleur ensemble de règles d'association à partir des données, L’algorithme Bat pour l'exploitation minière des règles d'association minière, Algorithme coopératif multi-essaim pour ARM et algorithme Bat multi-objectifs pour l'ARM.

Tous les algorithmes proposés dans cette thèse sont évalués à l'aide d'une série appropriée d'expériences, en utilisant plusieurs ensembles de données bien connus dans le domaine d’ARM et la performance de chaque approche proposée est évalué et comparé à ceux d'autres méthodes récemment publiées. Les résultats montrent une nette supériorité de nos propositions contre d'autres approches en termes du temps et de la qualité des règles.

الملخص

التنقيب على قواعد العلاقات يمكن اعتباره كمشكل معقد لأجل استخراج العلاقات ما بين مكونات قاعدة البيانات. الطرق التي تم تطويرها الي حد الان هي طرق يمكنها ايجاد كل الحلول الممكنة لكن تبقي غير عملية نظرا لكميات المعلومات الموجودة وكذلك المستعمل لهذه الطرق يجد نفسه امام كم هائل من القواعد ويصبح التعامل معها شيء صعب لهذا أصبح تطوير طرق جديدة شيء حتمي.هذه الاطروحة تتطرق الي اقتراح العديد من الطرق الجديدة التي ستساعد المستعملين في هذا الميدان بناءا على فكرة الذكاء الجماعي في الطبيعة الذ اثبت جدارته في عديد الميادين نثل الهندسة، علوم الحاسوب، والاقتصاد وغيرها. ان اساس هذه الاطروحة هو خوارزمية الخفافيش التي اثبتت جدارتها في حل العديد من المشاكل المصنفة كمشاكل جد صعبة. في هذا العمل نتطرق الي كيفية استعمال خوارزمية الخفافيش في استخراج العلاقات ما بين مكونات قاعدة البيانات. حيث اقترحنا ثلاث طرق مبنية على خوارزمية الخفافيش، أولا خوارزمية الخفافيش من اجل التنقيب على قواعد العلاقات، ثانيا خوارزمية الخفافيش متعددة الاسراب بثلاث طرق تعاون مختلفة، وفي الأخير اقترحنا خوارزمية للتنقيب على القواعد اعتمادا على الأمثلة متعددة الأهداف. النتائج المحصل عليها اثبتت نجاعة الطرق المطورة التي ترتكز على خوارزمية الخفافيش مقارنتا مع الطرق التي تم تطويرها في السابق.

Acknowledgements

Following this work, I thank, first of all, **God** for giving me the strength and courage during the elaboration of this modest work.

I would like to express my sincere gratitude to my advisor *Prof. Habiba Drias* for the continuous support of my Doctorate study and related research, for her patience, motivation, and immense knowledge. Her guidance helped me in all the time of research and writing of this thesis.

A special thanks goes to my co-advisor, *Prof. Nadjet Kamel*, who is most responsible for helping me to complete the writing of this dissertation as well as the challenging research that lies behind it. She taught me how to write well organized academic papers. Without her encouragement and constant guidance, I could not have finished this dissertation.

I thank the jury members for agreeing to evaluate and judge my present work.

Last but not the least, I would like to thank my family: my parents and to my brothers for supporting me spiritually throughout writing this thesis and my life in general.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

KDD	**K**nowledge **D**iscovery in **D**ata
DM	**D**ata **M**ining
AR	**A**ssociation **R**ule
ARM	**A**ssociation **R**ule **M**ining
CI	**C**omputational **I**ntelligence
AI	**A**rtificial **I**ntelligence
SC	**S**oft **C**omputing
ANN	**A**rtificial **N**eural **N**etworks
EA	**E**volutionary **A**lgorithm
GA	**G**enetic **A**lgorithm
ES	**E**volution **S**ystem
EP	**E**volutionary **P**rograming
ACS	**A**nt **C**olony **S**ystem
PSO	**P**article **S**warm **O**ptimization
ABC	**A**rtificial **B**ee **C**olony
BSO	**B**ee **S**warm **O**ptimization
FA	**F**irefly **A**lgorithm
CS	**C**uckoo **S**earch
BA	**B**at **A**lgorithm
MSBA	**M**ulti-**S**warm **B**at **A**lgorithm
MOOP	**M**ulti-**O**bjective **O**ptimization Problem
VEGA	**V**ector **E**valuation **G**enetic **A**lgorithm
MOBA	**M**ulti-**O**bjective **B**at **A**lgorithm

FLBA	**F**uzzy **L**ogic **B**at **A**lgorithm
CBA	**C**haotic **B**at **A**lgorithm
BBA	**B**inary **B**at **A**lgorithm
DLBA	**D**ifferential Operator and **L**évy flights **B**at **A**lgorithm
PLBA	**P**arameter-free **B**at **A**lgorithm
IBA	**I**mproved **B**at **A**lgorithm
DABA	**D**irected **A**rtificial **B**at **A**lgorithm
HBA-PR	**H**ybrid **B**at **A**lgorithm with **P**ath **R**elinking
MBBA	**M**ulti-**O**bjective **B**inary **B**at **A**lgorithm
FLANN	**F**unctional **L**ink **A**rtificial **N**eural **N**etwork
SVM	**S**upport **V**ector **M**achine
PSS	**P**ower **S**ystem **S**tabilizers
DIC	**D**ynamic **I**temset **C**ounting
G3P	**G**rammar **G**uided **G**enetic **P**rogramming
QARGA	**Q**uantitative **A**ssociation **R**ules by **G**enetic **A**lgorithm
CPU	**C**omputing **P**rocess **U**nit
GPU	**G**raphics **P**rocessing **U**nit
BAT-ARM	**BAT** algorithm for **ARM**
MSB-ARM	**M**ulti-**S**warm Cooperative **B**at for **ARM**
MOB-ARM	**M**ulti-**O**bjective **B**at algorithm for **ARM**
ACO$_R$	**A**nt **C**olony **O**ptimization based Algorithm for A**R**M

To my parents, and all my familly

To all my teachers

To my freinds

Introduction

Nowadays, stored data become huge due to the great development of the INTERNET and the unlimited connected devices. Global INTERNET active users in the world grows up about 14.3% between 2011, 2013 and now represents over 3.4 billion people[154]. In 2012, Google received over 2 million queries per minute, whereas two years later that number was doubled up to 4 million queries per minute[132]. It is estimated that GOOGLE is now about 10 Exabytes of data which is the largest stored data in the world. All these stored data are used by several users and companies to generate useful information to be effective for decision making. Therefore, these quantities of data make the development of new frameworks and methods of processing indispensable and hard task.

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. The KDD process is commonly defined in three successive stages: Pre-processing; Data Mining; and finally Post-Processing. Data Mining is the main phase in the whole KDD process, being responsible for mining non-trivial information hidden in data [53]. It contains several techniques used commonly in data processing such as: Classification, Clustering, Regression analysis and Association rules.

Association rule mining (ARM), one of the most important technique topics in data mining, was firstly innovated by *Agrawal et al.* [3] in early 90s. This task is able to generate a set of relationships among several attributes, also known as items, in a transactional database. It was originally designed 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. This spice of rules would allow shop-keepers to exploit this specific relationship by classify their products or to create a good marketing plan. However, in last decade, ARs become very utilized in different application domains such as: medical diagnosis, biomedical literature, protein sequences, logistic regression and fraud detection in web, etc. Many algorithms have been developed to solve ARM issues. The initial methods for rule extraction are exhaustive search proposals based on APRIORI algorithm proposed by *R. Agrawal and R. Srikant* in 1994 [5]. APRIORI is most popular and used algorithm for ARM due to its efficiency in generating all rules. It uses a recursive process to extract rules starting from the smallest rule that contains 2 items till the largest one. Generally, APRIORI process contains two main steps: (1) obtaining all the existing frequent item-sets in the data and (2) extracting all the ARs from the item-sets obtained in the previous step. Nevertheless, with stored data explosion, exhaustive methods suffer from memory requirements and the computational time consumption. Therefore, these algorithms failed to handle big databases.

To deal with huge databases within reasonable time, studies use intelligent heuristic algorithms, which are already successfully used to solve many NP-Complete problems, to discover association rules such as genetic algorithm [66, 41], particle swarm optimization [91, 147] and Bee swarm algorithm [36]. Generally, intelligent algorithms make individuals moving in an n-dimensional space to search the optimum solutions for an n-variable function optimization problem. Accordingly, to extract association rules from data, a database may be considered as a large search space and a mining algorithm as an exploration strategy. The datasets are sample space to search and each attribute is a dimension for the miner which aims to maximize/minimize a fitness function that evaluates the quality of extracted rules according to rule measures. Moreover, ARM was seen in many works as a multi-objective optimization problem according to rule measures which can be utilized as objectives to optimize [57, 111]. It is noted that not only time consumption and high-performance are principals in the field of ARM, but quality of generated rules is essence too. This condition can generate a useful and interesting rule to the final user which can be guaranteed by the right choice of quality measures considered for maximization.

1.1 Goals and Objectives

The general goal of this PhD Thesis is to develop new proposals to deal with Association Rule Mining, facing up to existing problems in the field. This issue, with the voluminous stored data, becomes significant which makes tools and framework creation indispensable. The main problems in ARM are computational time consummation, memory requirement and the large number of generated rules especially with exhaustive search algorithms which makes them useless for large databases. Therefore, we proposed different approaches to overcome the cited types of issues.

Firstly, we conduct a depth historical review of ARM, including all the basics used in this field, the existing exhaustive search methodologies, and discuss about their advantages and drawbacks. Also, we debate the solutions provided to overcome these difficulties. This study will help readers to understand the real issue and its difficulties. One of the most known drawbacks with exhaustive search algorithms are computational and memory requirements, since the search space could be so large that make it hardly maintainable. Thereby, the door was opened to a new series of methods based on intelligent computational discipline that reduces ARs mining complexity.

Secondly, we propose three different approaches to deal with several kinds of attributes that can be found in ARM including categorical and numerical ones. All our proposals are based on Bat algorithm that is a well known and recent efficient nature-inspired algorithm proposed by Yang in 2010 [177]. Firstly, we develop a redefinition for all the principals of bat algorithm to fit with ARM issue, Also, we introduce a new search strategy that makes our proposal able to extract valid rules. As well, we present another variant of our proposal by proposing a cooperative bat algorithm approach by developing three main collaborative strategies, that maintain a good trade-off between diversification and intensification which are the most important criteria for success in optimization algorithm. Finally, due to the several measures exists to quantify the quality of rules, we also propose a multi-objective model that aims to optimize several measures at the same time. To validate these approaches, many experiments are carried out on several well known datasets in the field of ARM for each proposal. The results prove the effectiveness of our approaches against other approaches in terms of computation time and extracted rules quality.

To sum up, in this PhD Thesis we are looking forward to reaching the following objectives:

- Analyze the state-of-the-art in ARM then review the most representative algorithms, including exhaustive search proposals and their major advantages and drawbacks.
- State a review about computational intelligence which is the successor of artificial intelligence, and its paradigms specially swarm intelligence algorithms and their applications to mine association rules.
- Provide an adaptation of the bat algorithm to extract association rules
- Propose a multi-swarm cooperative version of the bat algorithm to mine association rule that makes a good trade-off between exploration and exploitation.
- Present ARM issue as multi-objective perspective, and propose a version of our algorithm to optimize different quality measures at the same time.

1.2 Thesis organization

The remainder of this thesis is organized as follows: the next chapter provides the state-of-the-art about association rule mining and its existing algorithms, including exhaustive search algorithms developed in this field.
Chapter 3 presents a review about computational intelligence and its paradigms, where we focus on swarm intelligence and we describe a few popular swarm algorithms as examples. Also, presents an overview of the applications of CI algorithms to extract association rules.
Chapter 4 describes a detailed overview about Bat algorithm which the essence of our proposals.

Chapter 5 describes the utilization of bat algorithm for ARM. As our first contribution, we propose a modification for original Bat algorithm, called BAT-ARM, concepts to fit with ARM issue. Our proposal prove its efficiency in term of time, memory usage and quality of generated rules in-face-of FPgrowth algorithm.

Also, another version is provided in chapter 6. It is a multi-swarm cooperative bat algorithm for mining association rules within a transactional database, named MSB-ARM. In this proposal, the cooperation can be applied with three different topologies to maintain the best trade-off among exploration and exploitation to extract best rules. The experimental results show the superiority of this method against BAT-ARM and other approaches existing in the literature.

In the last chapter, we propose a multi-objective Bat algorithm for association rules mining, known as MOB-ARM, with the aim of extracting more useful and understandable rules to the final user. Also, it overcomes some drawbacks that appear in the previous contributions.

Finally, we conclude this thesis by general review, some remarks and we present our perspectives for a future work.

Background on Association Rule Mining

2.1 Introduction

Association rule mining (ARM)[3] identifies practical and interesting dependencies between items in a transactional database to help for decision making. Association rules have proven to be very useful in an enterprise as it strives to improve its competitiveness and profitability. In general terms, the extracted relationships can be shown as *IF-THEN* statements, *IF* $<$ *some conditions are satisfied*$>$ *THEN* $<$ *some values of other attributes*$>$. Conditions are in *IF* statement called *Antecedent* and those within the *THEN* clause are *Consequences*. In this chapter we start bya brief presentation of Knowledge Discovery in Databases. The second part is dedicated to association rule mining, definitions and notations. It also includes the most important quality measures of ARs. Finally, the last part states a review about the exhaustive search approaches proposed in the literature for mining association rules.

2.2 Knowledge discovery in databases and Data mining

Recently, Data mining is treated as an alternative to knowledge discovery in data (KDD), even though most studies in this field consider Data mining as a part of a whole process called KDD. KDD takes its emergency our days because of the massive amount of data stored in the world. As innovation in the last decade, there were only a few examples of knowledge discovery in real data whereas nowadays more domains benefit from the utilization of KDD techniques, such as medicine, finance, agriculture, social work, marketing, military, and many others. Consequently, KDD becomes a multidisciplinary domain which is integrated with big variety of fields in computer science such as: artificial intelligence, machine learning, neural networks, data bases, information retrieval and data visualization.

Generally, KDD aims at developing methodologies and tools which can automate the data analysis process and extract useful information and knowledge from data to help in decision making. Knowledge discovery in databases is firstly defined by Frawley et al. in 1992[53] as: *the nontrivial extraction of implicit, previously unknown, and potentially useful information from data.* After that this definition was revised by Fayyad et al. in 1996[46] as: *non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.*. This definition presents KDD as a complex process that contains several steps in which Data mining is one of them. Moreover, KDD is characterized as being a *non-trivial* process because it can decide whether the results are interesting enough to the user.

As presented in the definition given by Fayyad et al. four essential characteristics needs to be available in each pattern detected by the whole process: *validity, novelty, usefulness and comprehension by users.* For more clarification, generated patterns need to be **valid** for any new data with a certainty specified by interesting measures of any technique. Secondly, they need to be **novel** compared to the previous detected knowledge. Next, the patterns should be **significant and help** managers to make correct and profitable decisions for the companies. Finally, the patterns should be **understandable** to the analyzer which usually means the simplicity of knowledge. Mainly, KDD process includes three main steps to extract patterns from data: *Data Pre-Processing, Data mining* and *Post-mining.* **Figure. 2.1** shows main KDD steps[46].

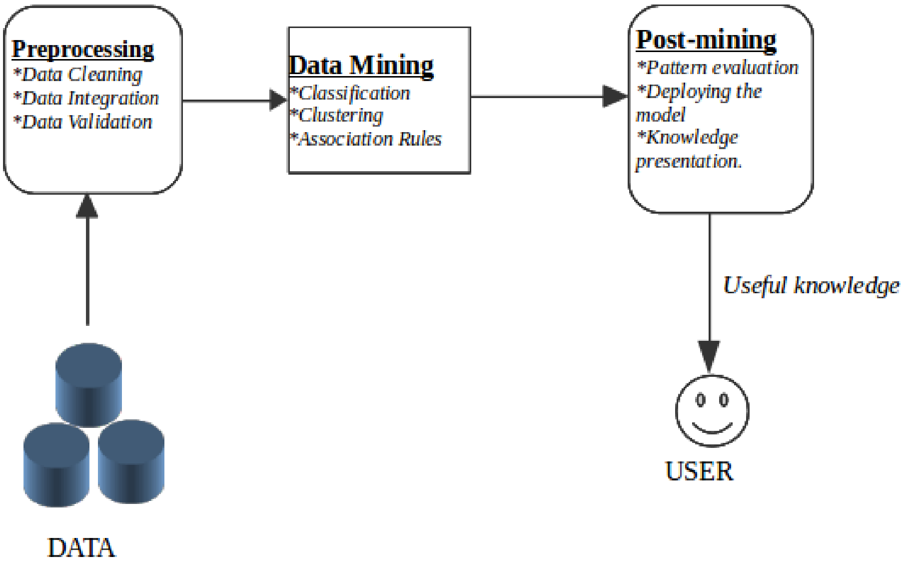


Figure 2.1: An overview of the steps comprising KDD process

2.2.1 Data Preprocessing

Data preprocessing is the first phase in the whole complicate KDD process. It consists of three main tasks:

- **DATA CLEANING**

  Data in real world are never 100% correct and always contains noise, missing values and additional information by errors which make data inconsistent. If KDD process is applied on this data as it is, it will generate bad trustworthiness knowledge. Data cleaning phase is included to overcome all similar issues of real life data. Therefore, data cleaning consists of detecting, correcting or removing inaccurate and inconsistent data from the database. For instance: if a value of an attribute is missed, this step will compute it based on statistical methods. In other case, some values are inserted by error, and data cleaning phase detects and removes it. Generally, this step uses automated systems based on statistical methods. These steps require considerable effort often as much as 70%, or more, of the total data mining effort.

- **DATA INTEGRATION**
  Data sources in the real life are several and with different schema. Data integration is used to combine data residing in different sources and provide users with a unified view of these data. This step becomes significant in the process of KDD in a variety of fields in our-days, especially in commercial domain when two or more companies share or merge their stored data. Moreover, data merging can produce some incoherent data and this task proposes solutions for this case. A most frequently used example in this kind of problems is redundancy. If an attribute A can be determined from another attribute B, then A is redundant compared to B.

- **DATA VALIDATION**
  Once the two previous phases are done, a new database is created which makes the process easier next. Data validation step is needed to verify that data is correctly cleaned in the first step and integrated in the second one, else a new re-cleaning or re-integration is processed. Afterword, The generated data is transformed and reduced into appropriate forms to facilitate the application of data mining techniques.

2.2.2 Data mining

The term data mining is proposed in recent decades. It is essential phase in the whole KDD process as demonstrated in Figure.2.1. 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, marketing and factory assembly lines, etc. An acceptable definition for data mining is proposed by Fayyad et al. in [46].

Definition 2.1. ***Data Mining*** *is a step in the KDD process consisting of applying data analysis and discovery algorithms that, under acceptable computational efficiency limitations, produces a particular enumeration of patterns over the data*

In general, four main classes of tasks have been developed in the literature in order to extract interesting patterns. These tasks can be classified in two main categories: *descriptive data mining* and *predictive data mining.*[22, 141], as shown in Figure 2.2. 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:

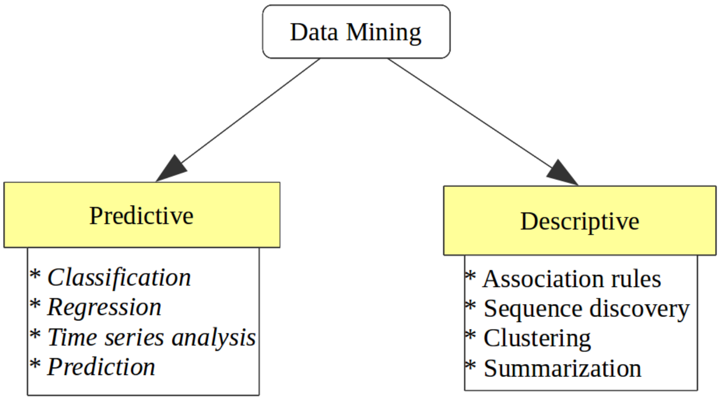


FIGURE 2.2: Data mining tasks.

- **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, modelling 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).*

- **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[52] 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*[16].

- **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*[3] 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 efficient 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.

2.2.3 Post-mining phase in KDD

It is the final step of the whole KDD process which follow the data mining step, in which many patterns are discovered via the algorithms applied on the database with different levels of confidence. Thus, this step verifies the produced information and detects which patterns can be presented as knowledge to the user. Post mining phase is essential to visualize the extracted knowledge in such way the user can understand them. It includes: *Knowledge Filtering, Evaluation and Information Visualization.* In the first, two main methods are usually used, rule truncation and post pruning, in decision tree and rules. *Evaluation* technique is essential to evaluate the system performance. This technique has a number of methods such as Classification accuracy, error, computational complexity etc. Which are specified for classifier model, Confusion Matrix for both classification and clustering, etc. Finally, *Visualization* is the most important technique in this step. It visualizes the knowledge discovered in data mining phase in such form the user can interpret them simply for better decision making[138]. Various methods are developed for knowledge visualization such as Scatter Plot Matrix Technique, Parallel Co-ordinates, and Pixel Oriented Technique, etc.

Generally, the post mining phase is a user driven step, in which the user becomes responsible to decide about the interestingness of some knowledge. Therefore, several methods are developed to guide the user in this step. For example, the over-fitting issue is frequently happened with classification algorithms, in which some patterns in the training set do not exist in general dataset, and it needs user intervention to eliminate them.

2.3 Association rule mining

Association rule mining (ARM) is one of the most significant tasks in data mining in our days. As mentioned above, It is a descriptive data mining technique. It has received wide attention in research since its innovation by *Agrawal et al.*[3] in the early 90s. ARM is developed to detect frequent, interesting and strong relationships among a database. As first innovation, ARM is used in market basket analyses to understand client behavior in shopping and use this information to make marketing decisions. Whereas, nowadays the huge quantity of data stored in various fields makes ARM more useful and important such as: business intelligence [79], Medical[55] and natural language processing [82].Etc. So, there is a clear need to evolve new automated tools and frameworks to extract useful and comprehensive rules for the final user. In this section, we introduce the main basics on the ARM issue and later we present a state-of-the-art the various exhaustive search methods to solve this problem.

2.3.1 Basics of association rule mining

Association rule is a relationship between two non-empty item-sets which make an implication of the form $Antecedent \Rightarrow Consequence$, where both $Antecedence$ and $Consequence$ have no common items. As previously noted, ARM was firstly used in basket market analysis but how such informations can help in decision making for the market owner?

Terminology and Annotation

Market basket analysis[3] is a process that analyses customer buying habits by finding associations between the different items that customers place in their "shopping baskets". The discovery of such associations can help retailers to develop marketing strategies by gaining insight into which items are frequently purchased together by customers. 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? Such information can lead to increased sales by helping retailers to do selective marketing and plan their shelf space. For example, placing milk and bread within proximity may further encourage the sale of these items together within a single visit to the store.

In this section, the basic concepts that appear in the frequent pattern and association rules are defined. **Table.2.1** illustrates five of "shopping baskets" in a supermarket, each one of them contains a number of articles among { *Milk, Bread, Sugar, Eggs*}.

	Milk	Bread	Suggar	Eggs
Basket 1	X	X		
Basket 2	X		X	
Basket 3		X		
Basket 4	X		X	X
Basket 5	X	X		

Table 2.1: Shopping baskets of some customers

Definition 2.2. Item: All objects, articles, attributes, literals that belong to a set of discrete elements.

Example 2.1. *In Market basket analysis all the articles in shop are items. In our example items are* ***Milk, Bread, Sugar, Eggs****.*

Definition 2.3. Itemset: is all subsequence of items in **I** and all itemset that contains **k** items are called k-itemset.

Example 2.2. ***{Milk, Bread}*** *is 2-itemset,* ***{Milk, Sugar, Eggs}*** *is 3-itemset and* ***{Milk, Bread, Sugar, Eggs}*** *is 4-itemset.*

Definition 2.4. Transaction : All itemset associated to a unique identifier called T_{id}, a set of transactions identifiers is presented by **T**. A set of couples that contain a unique identifier and transaction is known as transactional database.

$$D = \{(T_{id}, X)/T_{id} \in T, X \in I\}$$

Example 2.3. *In our example :*
$I = \{\ Milk,\ Bread,\ Sugar,\ Eggs\}$
$T = \{\ 1,\ 2,\ 3,\ 4{,}5\}$
$D = \{\ (1{,}\{Milk{,}Bread\}){,}(2{,}\{Milk{,}Sugar\}){,}(3{,}\{Bread\}){,}(4{,}\{Milk{,}Sugar{,}Eggs\}){,}(5{,}\{Bread\})\}$

Definition 2.5. Transaction database representation: transactional database can be represented in horizontal, vertical and bitmap representation[192].

- *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(see table 2.2).

T_{id}	Items
1	A,B,C
2	B,C
3	C,D
4	B,C,D

Table 2.2: 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 2.3 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}.
- *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 2.4 shows an illustration of a bitmap layout.

Items	T_{id}
A	1
B	1,2,4
C	1,2,3,4
D	3,4

Table 2.3: Vertical representation of the transactional database

T_{id}	A	B	C	D
1	1	1	1	0
2	0	1	1	0
3	1	1	1	1
4	0	1	1	1

Table 2.4: Bitmap representation of the transactional database

Definition 2.6. Association rules: Let $I = \{i_1,\ i_2, \ldots, i_n\}$ be a set of literals called items. Let D be a set of database transactions where each transaction T contains a non-empty set of items. First, Association rule is defined as an implication of the form $A \longrightarrow B$, where A and B satisfy the following constraints: $A \subset I$, $B \subset I$, $A \cap B = \emptyset$, under support and confidence above a minimum threshold[3].

Later, the definition is extended to an implication of the form $A \longrightarrow B$, where A and B are itemsets and $A \cap B = \emptyset$. The former, A, is called the antecedent of the rule, and the latter, B , is called the consequent of the rule[5]. Moreover, an association rules is expressing the fact of if/then statement. For example, *an association rule would be "If a customer buys bread, he is 66% likely to also purchase milk."*

With the incremental number of rule mining applications new types of rule are developed related to several factors in real life. Generally, there are four main types of association rules that can be detected from stored data:

Quantitative association rules. In our-days, real data contain numerical attributes such as; temperature, age and size, etc. Thus, it becomes necessary to mine this type of data which produces a new type of ARs called: Quantitative association rules or numerical association rules[145, 9, 111]. In general, the quantitative data presents a wide range of values which makes the mining in this data almost impossible with all values as in categorical data, in which the biggest issue in quantitative rules is situated. So, the challenge is to divide these numerical values into domains that satisfies a search criteria during the mining process[67]. Therefore, A QAR is an AR, where an implication of the form $A \Rightarrow C$, have the particularity that $A \subset I_{\mathbb{R}}$, $C \subset I_{\mathbb{R}}$, and $A \cup C = \emptyset$, where $\mathbb{R}$ is a set of real numbers.

Negative association rules. In some cases, it is necessary to know the uncorrelated items, which mains the presence of such items implied the absence of another. For example *"customers that buy juice do not buy bottled water"* which provide valuable information that can help in decision making. Thus, negative association rule is a rule that contains a negation of an item (i.e a rule for which its antecedent or its consequent can be formed by a conjunction of presence or absence of terms)[189]. Some examples of such association would be as follows: $\neg A \Rightarrow B, A \Rightarrow \neg C$ or $A \wedge \neg B \wedge \neg C \vee D \Rightarrow \neg F \wedge E$

Class association rules. Additionally, normal ARM does not have any target which means any item in dataset can appear even in the condition or in the consequence. However, in some applications users can be interested to some targets. Therefore, a new sort of ARs is defined and called Class association rule. In other words, CAR[98] is a subset of association rule which has specified targets or classes as their consequent. Formally, CAR can be defined as follows: Let be T a set of n transactions. Each transaction is labeled by a class y. The set of all items in T is labeled by I and the set of class labels is labeled by Y where $I \cup Y = \emptyset$. A class association rule (CAR) is an implication of the form: $A \Rightarrow B$, where $A \subset I$ and $B \subset y$.

Rare association rules. Finally, ARM is mainly a search for the frequent correlations and relationships between items in database to discover significant knowledge. Whereas, there is some important information that can be hidden behind non-frequent relationships, unusual, or exceptional rules, The aim of RAR is to detect low-raked itemsets and generate meaningful rule from them[140]. There are several approaches to discover rare association rules. The simplest way is to directly apply the Apriori algorithm by setting the minimum support threshold to a lower value which generates a wide number of rules including the rare rules.

2.3.2 Association rules measures

As mentioned above, the objective of KDD process is extracting patterns of data that should be valid and compressible to the user. Generally, users do not know which knowledge will be generated from such data. Therefore, the interesting patterns should be quantified by some metrics that define its importance to the user[59]. Based on user interaction with the KDD process interestingness measures in ARM could be divided in two main categories: *subjective and objective measures*[20, 158]. The first takes into account both user's goals (objectives) and data measures, whereas, the second is based only on the data information. In the following we detailed the significant objective measures in ARM field, which do not require any user's background.

As previously stated in definition 2.6, due to the wide number of patterns extracted from a large transactional database, a detected rule is accepted as association rule if its support and confidence satisfy a minimum threshold, specified by a final user, and rejected otherwise. Thus, *Support* and *Confidence* are two measures that aim to determine rules quality which are defined as follows:

Definition 2.7. Support is the proportion of transactions in D that contains X, to the total of records in database. Support is calculated using the following equation:

$$support(X) = \frac{\text{Number of transactions containing } X}{\text{Total Number of transactions}} \tag{2.1}$$

The support of an association rule $X \Rightarrow Y$ is the support of $X \cup Y$.

Definition 2.8. 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. The rule confidence is calculated as follows:

$$confidence(X \Rightarrow Y) = \frac{support(X \cup Y)}{support(X)} \tag{2.2}$$

The confidence quantifies extracted rules strength. An association rule $X \to Y$ with a confidence of 80% means that 80% of the transactions that contain X also contain Y. Based on these two measures (Support and Confidence), ARM aims at extracting all rules that satisfy both user-specified minimum support and minimum confidence thresholds[67]. Nerveless, these measures could not evaluate the real interest and quality of extracted rules, consequently, a great variety of proposals define other quality measures to deal with weakness of the support-confidence framework such as: Lift, Leverage and Conviction, etc. in the following we detailed and define the most frequently used objective measures in ARM issue.

Definition 2.9. Lift measure was proposed to determine the correlation between condition and consequence of the rule by *Brin et al.*[23], it is defined as follows:

$$lift(A \Rightarrow C) = \frac{Confidence(A \Rightarrow C)}{Support(C)} \tag{2.3}$$

In other words, the lift measure quantifies the degree of dependence among rule antecedent and consequence. More specifically, if the lift is equal to 1 then antecedent and consequent are independent; else, if it is greater than 1, antecedent and consequent are positively dependent. Otherwise, they are negatively dependent. Therefore, most proposals search for the positive dependence between the two parts of the rule.

Definition 2.10. Leverage (novelty) measure quantifies how different is the co-occurrence of the antecedent and consequent [93]. The rule leverage is calculated as follows:

$$leverage(A \Rightarrow C) = support(A \Rightarrow C) – (support(A) \times support(C)) \tag{2.4}$$

Generally, leverage takes its values in the range of $[-0.25, 0.25]$, where a strongly positive value indicates a strong association between A and C, whereas, negative values indicates a low association between A and B and strong association between $\neg A$ and C. In addition to these, a variety of measures are proposed[157] to deal with other weakness of support-confidence measure.

Definition 2.11. Conviction was firstly proposed by *Brain et al.* in 1997[24]. It is inspired from the logical definition of implication and represents the degree of implication of the rule. Conviction ranges along the values 0.5,.., ∞, when its values are far from 1, this indicates the interesting of the rule. Unlike other measures, it is sensitive to rule direction, means $conv(A \Rightarrow C) \neq conv(C \Rightarrow A)$. It is defined as:

$$Conviction(A \Rightarrow C) = \frac{1 - support(A)}{1 - support(A \Rightarrow C)} \tag{2.5}$$

In association rule, when the rule contains a huge number of attributes, this makes the rule more difficult to comprehend. If generated rules are not comprehensible to the user, they will be useless. Therefore, another measure is defined to overcome this issue known as *Comprehensibility*.

Definition 2.12. Comprehensibility[163] measure attempts to calculate the simplicity of the generated rule and its understandability to the user. It can be calculated as:

$$Comprehensibility(A \Rightarrow C) = \frac{\log(1+|C|)}{\log(1+|A \cup C|)} \tag{2.6}$$

Where, $|C|$ and $|A \cup C|$ are items number in the Consequence part and the total rule respectively. The comprehensibility increases and the rules are more understandable whenever items number in the antecedent part are smaller.

Moreover, it is important to extract those rules that have a low support in the whole database but such rare rules can be interesting to decision-makers. Thus, another measure is defined for this problem called *Interestingness*.

Definition 2.13. Interestingness[163] 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 by :

$$Interesting(A \Rightarrow C) = \frac{Supp(A \cup C)}{Supp(A)} \times \frac{Supp(A \cup C)}{Supp(C)} \times \frac{(1 - Supp(A \cup C)}{N} \tag{2.7}$$

Where A, C and N are the antecedence, consequence and transactions number in the whole database, respectively.

Example 2.4. *let consider the transaction database modeled in table 2.5, that contains 5 transactions t1,t2,t3,t4,t5 and 5 items A,B,C,D,E. 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 2/5. Thereafter, the confidence of rule* $A \Rightarrow B$ *is equal to proportion of the support(*$A \Rightarrow B$*) to the support(A), that was* $\frac{2/5}{3/5} = 2/3$. *If we consider the Minsup and Minconf equal to 2/5 and 2/3, respectively, then the rule* $A \Rightarrow B$ *is accepted.*

t1	A	B	C
t2	A	B	
t3	C	D	
t4	E	D	
t5	C	A	

TABLE 2.5: An example of transactional database

Moreover, the other measures of the rule $A \Rightarrow B$ *are calculated as follows:*

- *Lift(*$A \Rightarrow B$*) = 5/3.*
- *Leverage(*$A \Rightarrow B$*)= 4/25.*
- *Conviction(*$A \Rightarrow B$*) = 9/5.*
- *Comprehensibility(*$A \Rightarrow B$*)* $= \frac{log(2)}{log(3)}$ *= 0,63.*
- *Interesting(*$A \Rightarrow B$*)* $= \frac{2/5}{3/5} \times \frac{2/5}{2/5} \times \frac{1 - 2/5}{5}$ *= 2/25.*

2.3.3 Applications of AR

Since ARM was innovated the earlier in 90s by *Agrawal et al.*[3], this task has attracted researchers attention and it was broadly used in the business field to help managers in decision making. However, business, which needs and utilizes extracted patterns of stored data, is not the only challenge domain for data mining tasks. Also, ARM was applied to wide large of significant domains in the last decade such as: Medical diagnosis, biomedical literature, protein sequences, census data, logistic regression and fraud detection in web, etc.

As first innovation ARM was used for market basket analysis[3] which is a typical and widely-used example of association rule utilization. In this application, the data are recorded in the supermarket using a bar-code scanners. Such databases contain a wide number of transactions. Managers could be interested to which products are purchased together. This information can be used in several locations in decision making in the supermarket. For example, they could use to improve the marketing strategies by new plans which increase sales and profit, 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. In the last years, basket analysis went over web stores, who are mostly using ARM to suggest combinations of products for web-user based on the history of sells in that web-store.

Nowadays, one of the most important domains is health care where medical diagnosis is an interesting issue that can be used to assist physicians to cure patients. Practically, diagnosis is a hard task due to the unreliable diagnostic tests. This may fact on the prediction that obstructs on the medical process. ARM was applied successfully in this field by *Serbin et al.* By proposing a technique based on relational association rules to identify the probability of illness in a certain disease by determining the relationships among symptoms[151]. Also, in [122], the authors implemented efficient search for diagnosis of heart disease comparing association rules with decision trees. Recently, association rule mining was used to deal with sick and healthy factors which contribute to heart disease for males and females[118]. Another study used ARM to identify risk patterns for type 2 diabetes incidence[135]. This study showed that the ARM is a useful approach in determining which combinations of variables or predictors occur together frequently, in people who will develop diabetes.

In addition, Proteins studies and analysis are an important field where ARM is applied frequently. Proteins are the principal constituents of cellular machinery of an organism. They are sequences made up of 20 types of amino acids. Each protein has a unique 3-dimensional structure, which depends on the amino-acid sequence; slight changes in the sequence may change the protein function. In[64] association rule is used to decrypt the nature of associations between different amino-acids that are present in a protein which help to understand the protein composition. Moreover, in the last decade, significant efforts have been made to study the principles of protein–DNA bindings. Also, association rule mining is successfully applied to discover protein–DNA bindings[95, 170]. Recently, association rule mining was used to predict users web page access behaviour[58].

Many other important domain applications in which association rules are applied successfully and ARs add benefits to them such as: Communication failure detection[134], analysis of interesting rare patterns in telecommunication networks [146] and educational field[142], etc. Furthermore, association rule mining is used to fine reveal correlations between pages accessed together during a server session. Such rules determine the possible relations among sets of pages that are viewed together, and can indicate the association between groups of users with specific interests. Thus, such observation can guide the website restructuring or produces a powerful recommendation system[114, 100, 172].

2.4 Exhaustive Search algorithms for ARM

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[3] as a first method proposed by *Agrawal et al.* for mining frequent pattern in the database. Next, Apriori algorithm[5] was presented to mine ARs. This algorithm uses prior knowledge of frequent item-set properties to extract ARs. Within this section we will spotlight on the most known classical algorithms in ARM field.

TID	ITEMS
T100	I1,I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3

(A)

Items	Counting
I1	3
T2	4
I3	2
I4	2
I5	1

(B)

Items
I1
I2
I3
I4

(C)

Items	Counting
I1,I2	2
T1,I3	1
I1,I4	1
I2,I3	1
I2,I4	2
I3,I4	0

(D)

Items
I1,I2
I2,I4

(E)

Items	Counting
I1,I2,I4	1

(F)

Table 2.6: AIS mining process

AIS[3] (Agrawal, Imielinski, Swami) is the fist algorithm proposed to solve the association rule problem. In this proposal just one consequent rules are generated, that means all generated rules are like $X \cap Y \to Z$.

AIS is a multi-pass algorithm, the main idea in this method is scanning the database and count the support of each item set, and generate the new candidate. During the algorithm process to extract the frequent itemsets, the first pass over the transactions, each individual item support was counted. Next all items that not satisfied the minimum support are deleted from the result, let considered minimum support in table 2.6b equal to 2. So, I5 is eliminated because its support is less than the specified minimum support. Table 2.6c presents the generated frequent 1-itemset. Hereafter, 1-itemset is joined with itself to generate a candidate 2-itemset, during the second pass on the database support of those 2-itemsets are accumulated and the itemsets that not satisfied the minimum support are eliminated. Similarly, the candidate k-Itemsets are generated from extending the (k-1)-itemset by joining with themselves as showing with 1-itemset, new pass on the database is required for each new candidate k-itemset to count the support and eliminate the k-itemset that not satisfied the *min-supp*. This process iterates until one of those candidates is empty. To make this algorithm more efficient, a support estimation of these candidates is used to guide whether these candidates need to be extended further to produce more candidates[3]. The main disadvantage of this algorithm is the generation of too many candidates and multi-passes on the whole database that makes more complexity in time and space.

Apriori algorithm is the most improved algorithm for ARM in the history, it was proposed by *Agrawal et al*[5]. The authors are trying to solve the disadvantages of AIS. AIS is a simple algorithm that generates a lot of candidates and require a multi-passes on the database to count the support of each candidate while the most of them are not frequent and will never be. The novel idea proposed by Apriori was adding a new step in the generation of the candidate. First, candidates itemsets are generated, as in AIS, by joining the previous large itemsets. Next, a pruning step was applied on these candidates related to the Apriori Propriety[5]; *The Apriori propriety says that every sub (k-1)-itemset of frequent k-itemset must be frequent.* Afterward, the association rules are extracted based on the frequent patterns detected in the first step.

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. As AIS algorithm, 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.

Example 2.5. *For example as showing in the table 2.7, I1 was extended with I2,I3,I5 the result is (I1,I2),(I1,I3),(I1,I5), I2 was extended with I3,I5 the results are (I2,I3);(I2,I5), and it's not extended with I1 because (I1,I2) was generated in the extension of I1 with I2, I3 extended with I5 and gives (I3,I5).*

Thus, the same process of joining is used to generate the k-itemset from the (k-1)-itemset, but after any joining process a new pruning step is executed to eliminate the itemsets that can't be frequent related to Apriori propriety. For instance, the 3-itemset in our example is generated, the result of joining is *(I1,I2,I5); (I1,I2,I3);*

TID	ITEMS
T100	I1,I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1,I2, I3,I5
T900	I1, I2, I3
T000	I1, I2, I5, I6

(A) Sample dataset

Items	Counting
I1	7
T2	8
I3	6
I4	2
I5	3
I6	1

(B) C_1, 1-itemset candidate

Items
I1
I2
I3
I5

(C) L_1, Frequent 1-itemset

Items	Counting
I1,I2	5
I1,I3	4
I1,I5	3
I2,I3	4
I2,I5	3
I3,I5	1

(D) C_2, 2-itemset candidate

Items
I1,I2
I1,I5
I2,I5
I2,I3
T I1,I3

(E) L_2, Frequent 2-itemset

Items	Counting
I1,I2,I5	3
I1,I2,I3	2

(F) C_3, 3-itemset candidate

Items
I1,I2,I5

(G) L_3, Frequent 3-itemset

$\emptyset$

(H) End

TABLE 2.7: Apriori process to generate the frequent itemsets

(I2,I3,I5); (I1,I3,I5), those items are checked using the Apriori propriety and we eliminate *(I2,I3,I5); (I1,I3,I5)* because *(I3,I5)* are not frequent and do not appear into the 2-itemset. 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.

Therefore, we can say that Apriori is an improved AIS algorithm, where the main difference between them is that Apriori ignore the support counting of itemsets that can not be frequent based on Apriori propriety applied after each new candidate generation, that will make the candidates much smaller than those into AIS, which reduce the time of computation and memory consumption.

The second step of the Apriori algorithm is rule generation. This step aims to

Algorithm 1 Frequent itemset generation in an Apriori algorithm

```
Input:
D: transaction database;
Min_sup: the minimum support threshold
Output: frequent itemsets
$L_1$= find_frequent_1-itemsets(DB);
for k=2; $L_{k-1} = \varphi; k++$ do
  $C_k$= Apriori_gen($L_{k-1}$);
  for each transaction $t \in DB$ do
    $C_t$ = subset($C_k, t$);
    for each candidate $c \in C_t$ do
      $c.count++$;
    end for
  end for
  $L_k = \{c \in C_k | c.count \geq min\_sup\}$
end for
return $L = \bigcup_k L_k$;
Procedure Apriori gen($L_{k-1}$: frequent($k-1$)-itemsets)
```

create association rules from the frequent itemsets generated in the first step. The algorithm of this step is presented in **Algorithm** 2. However, With greater efficiency of Apriori, this algorithm is still having the major drawback of AIS, it's scan of the whole database takes too many times.

Since the Apriori was proposed, many algorithms are developed based on it and try to improve it. Generally two big approaches are used to improve Apriori efficiency: one is to reduce the number of passes on the whole database or replacing the database by a part of it. Other approaches used new and many kinds of pruning approaches. e.g : AprioriID and aprioriHybrid[5], DHP[125], partition technique[148], Sampling approach[161], parallel and distributed mining[4, 126] Dynamic itemset counting [24], Based matrix improvement method[102], IAA adopts a new count-based method to prune candidate itemsets and uses generation record to reduce total data scan amount[171]. In the following we will present some of these algorithms:

AprioriTid[5]: It has the same process of Apriori with two steps candidate generation and support counting. The main difference between the classical Apriori and aprioriTid is that the later does not use the whole database to count supports after the first pass. An encoding for(k-1)-itemset (TID) is used in the new pass

Algorithm 2 Rule generation step in Apriori algorithm

```
Input:
Set of itemsets $l$
Min_sup: the minimum support threshold
Min_conf: the minimum confidence threshold
Output: Set of association rules Rules
for all itemsets $l_k, k \geq 2$ do
    call generate-rules($l_k, l_k$)
end for
Procedure generate-rules($l_k : k - itemset,\ a_m : m - itemset$)
$A = \{$(m-1)-itemsets $a_{m-1} \mid a_{m-1} \subset a_m\}$
for all $a_{m-1} \subset A$ do
    $conf = support(l_k)/support(a_{m-1})$
    if $conf \geq Min\_conf$ then
        $R = a_{m-1} \Rightarrow (l_k - a_{m-1})$
        if m-1 $\geq 1$ then
            call generate-rules($l_k, a_{m-1}$)
            Rules = Rules $\cup$ $R$
        end if
    end if
end for
return  Rules
```

over the database. The encoding number will be less than transactions number in the whole database. Hybrid Apriori uses Apriori in the initial passes and moves to AprioriTid for the rest of iterations.

Partitioning technique[148]: This approach requires two passes over the database to find the frequent itemsets. It works in two steps: firstly, it divides the database to disjoint partitions, each one contains a number of transactions from the database. After that, a local frequent pattern of each partition is found related to its local $Minsupp(Local_{m}insupp = minsupp \times number_{o}f_{t}ransactions_{i}n_{t}he_{p}artition)$. Next, new scan for the whole database is required to find the general frequent pattern based on the local frequent patterns found in the previous step.

Sampling approach[160]: This proposal reduces scan number over the database to one in the best case and two in the worst one. The main idea is mining the frequent itemset in a subset of the given data. This algorithm picks a random sample S from the database D and searches for the frequent itemsets in this sample using Apriori. When S is in D that means the algorithm search in frequent patterns in

D, if the found patterns in S are all the patterns in D, so just one scan is enough, if not an another scan of the database is required to find the missed patterns.

Dynamic Itemset Counting[24]: The authors of DIC reduces the number of scans in Apriori to just two scans.. The principal idea is to generate and calculate the support earlier. Here the database is viewed as intervals, in the first one 1-itemsets are generated, in the second the 2-itemsets are detected, etc. Generally, in the kth interval the k-itemsets which are potentially frequent are generated. A second scan of the database is required to count the itemsets that are not fully counted in the first scan.

Apriori Based Matrix[102]: This approach introduced the matrix concept to reduce the number of scans over the whole database. The main idea is to create a binary sequence for each item, where 1 designs the item's existence in the transaction T and 0 presents the absence of that item in the transaction T.

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.

FP-growth is an algorithm proposed by Han et Al. in [68] where the authors try to solve the bottlenecks of Apriori. FP-growth algorithm mines the frequent itemsets 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.

For example, let table 2.8a be a database which contains 8 transactions to create the Fp-tree, we fixed the minimum support threshold to 3. After the first scan for the database we get the results shown in table 2.8b which contains the existing items and their frequencies. Related to Fp-tree algorithm, these results must be sorted in descending order as shown in table 2.8c. Finally, the algorithm generates a transformed data with sorting items in each transaction based on counted support in the previous step.

Figure 2.3 presents the steps to insert transactions one by one in the FP-tree. Initially, FP-tree have *'null'* as root node. FPgrowth algorithm attaches items in first transaction one by one respectively to the root node and writes their occurrences. The tree resultant is shown in figure 2.8a. Next, for each item we call the function *Insert([p |P], T)* which determines if the node p exists in the tree or not. If it exists, the occurrence in that node increments by one, otherwise a new node containing p, with occurrence equal to 1, is created and attached to the parent node. Figure 2.8b shows the tree after completing the insertion of all the second transaction items.

The frequent patterns are generated by FP-growth[68] algorithm based on the FP-tree and head table. Here, the full process to generate the frequent patterns concerns **c:3**, where 3 is the occurrence of c. Firstly, we find all paths that finishe with

TID	List Items
1	EADB
2	DACEB
3	CABE
4	BAD
5	D
6	DB
7	ADE
8	BC

(A)

Item	Frequency
A	5
B	6
C	3
D	6
E	4

(B)

Item	Frequency
B	6
D	6
A	5
E	4
C	3

(C)

Item	Frequency
1	BDAE
2	BDAEC
3	BAEC
4	BDA
5	D
6	Bd
7	DAE
8	BC

(D)

Table 2.8: Head table creation

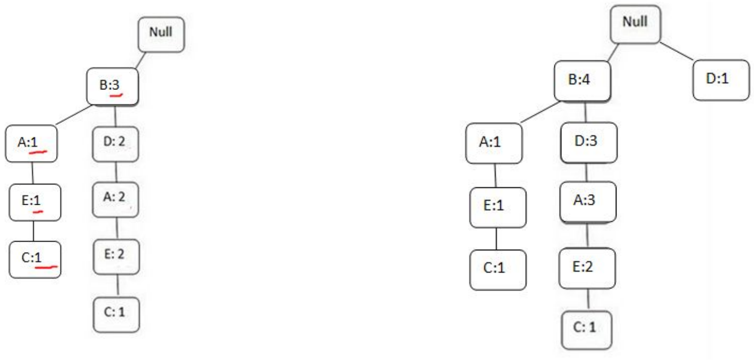
c in the FP-tree. So, conditional pattern based on *c* are *(B):1,(B,A,E):1,(B,D,A,E):1*. Next, the supports of all items in the based patterns are accumulated such as: *B:3,D:1, A:2,E:2*, just *B:3* satisfy the minimum support count. The conditional FP-tree of *c* is *(B):3*. Now we can say that *C:3, BC:3* are the frequent patterns based on *c*. This process iterates until all items in head table are visited.

FP-growth is more efficient than Apriori for three reasons. First, just two scans of the whole database are required to generate all the frequent patterns. Secondly, the Fp-tree is compressed representation of the original database. Thirdly, FP-Tree uses a divide and conquer method that considerably reduces the size of subsequent conditional FP-Tree. Moreover, longer frequent patterns are generated by adding a suffix to shorter ones.

Every algorithm has its limitations. FP-Tree algorithm is difficult to be used in an interactive mining system. Generally, during an interactive mining process, users may change support threshold according to extracted rules and their exigencies. However, FP-Tree changing support threshold may lead to repetition of the whole

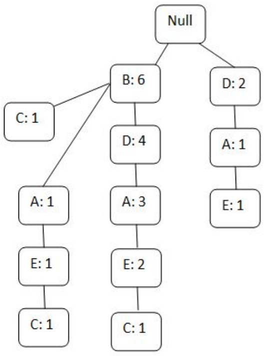
(A) Adding Row 1 to Fp-tree

(B) Adding Row 2 to Fp-tree



(C) Adding Row 3 to Fp-tree

(D) Connect D to null node



(E) Final FP tree

FIGURE 2.3: Process of Fp-tree creation

mining process. Furthermore, FP-Tree is not suitable for incremental mining, since as time goes on databases keep changing, new transactions may be inserted into the database, those insertions may also lead to a repetition of the whole process. On another side, support count step is an expensive step in time because it scans the whole database for each new candidate itemset. This step can be improved if the database is indexed in such way. Thus, the time to accumulate the support can be reduced.

Eclat algorithm proposed in [190] realizes the previous idea and to accumulate the support it uses Tidsets. The basic idea is to use a vertical layout for transactional database, because this representation makes the counting easier than in horizontal layout.

Example 2.6. *let an item X present in the transactions T100, T350, T800, T1240, the support count of supp(X) is the number of transactions in which X appears,* $supp(X) = |\{T100, T350, T800, T1240\}| = 4$.

Generally, The support of a candidate itemset can be computed by intersecting the Tidsets. In general, support of the itemset containing X and Y is the number of transactions in the intersection between the Tidset of X and Y.

Example 2.7. *let Tidsets for A and C be t(A) = 1,2,3,4,6 and t(C) = 1,3,4, respectively, then we can determinate* $t(AC) = t(A) \bigcap t(C) = \{1, 3, 4\}$ *and the support count* $supp(AC) = |\{1, 2, 3\}| = 3$.

Thus, whenever generating a candidate itemset, we also generate its Tidset by intersecting the Tidsets of its parent. By the way, the support of that candidate is the size of its Tidset, so that support counting becomes trivial and it is done simultaneously with candidate generating.

Eclat[190] uses the depth-first-search, and it starts with prefix {}. To divide the initial search tree, it picks the prefix of first item in the database, then it generates the corresponding equivalence class and adds frequent itemset mining in the sub tree of all itemsets containing the selected item. This sub tree is divided further into two sub trees: the first sub tree consists of all itemset containing the prefix with second item in the database as new prefix, whereas the other consists of all itemsets containing the selected prefix but not the second item.This process is recursively iterated until all itemsets in the initial search tree are visited.

Zaki and Gouda [191] proposed a new vertical data representation and try to improve Eclat, called Diffset, and introduced dEclat, an Eclat-based algorithm using Diffset. Instead of using Tidsets, they use the difference of tidsets (called diffsets). Using diffsets has reduced drastically the set size representing itemsets and thus operations on sets are much faster. dEclat had been shown to achieve significant improvements in time consumption performance as well as memory usage over Eclat. Especially on dense databases [191]. However, when the dataset is sparse, Diffset loses its advantage over Tidset. Therefore, Zaki and Gouda suggested to start using Tidset format for sparse databases and then switching to Diffset format, later, when a switching condition is met.

2.5 Conclusion

In this chapter, we described a general background on Knowledge discovery in database process. More particularly, we focused on the field of association rule mining. Also, we detailed the most important basics and notations in this field. Additionally, the most significant quality measures for ARs are presented. Next, we presented a wide number of cases where association rules are successfully applied. Finally, we described the existing exhaustive search methods for mining association rules which proportionally reduces the complexity of association rule extraction problem. For example, the complexity of Apriori algorithm in the worst case is $O(n^2 \times m)$, with the increasing number of items and transactions such algorithm become slow and useless. Thus, they can be suitable for small data or users who can allow a long time processing. However, to overcome this drawback many researchers in the last decade use computational intelligence methods to solve the ARM issue as an optimization problem which will be detailed in the next chapter of this thesis.

Computational intelligence

3.1 Introduction

The nature become a large inspiration source for researchers to solve many complex problems in computer science. Specially, optimization problems, including mono-objective and multi-objective issues. Evolutionary algorithms are inspired from evolution theory, animals organization and movement give birth to swarm intelligence to solve optimization problems. These two categories of methods are included in a large research domain called ***Computational intelligence (CI)***. CI proves its efficiency in many real applications, that motivates researchers to use intelligent heuristic algorithms to discover association rules such as genetic algorithm [66, 41], particle swarm optimization [91, 147], and Bee swarm algorithm [36]. In this chapter, we present a general overview on the field of Computational Intelligence and in particularity swarm intelligence methods such as: Ant Colony, Bee-swarm and Particle Swarm, etc. Next, a review about intelligent methods applied to extract association rules is stated.

3.2 Artificial or Computational intelligence and soft computing

Artificial intelligence (AI) aims to transmit human intelligence to machines to make this last capable to solve complex problems which need human intelligence. IA is a big domain in the computer science field that includes: reasoning, machine learning, classification and prediction, etc. The classical AI problems are usually formulated as state-space search problems [143]. The term state describes and refers to one instance of the whole problem. Whereas, space, describes the collection of instances in the presented problem. Generally, one or different states are given as an initial case to start from and achieves the goal after a set of movements. The traditional AI methods are very effective in inductive but inefficient to realize supervised learning[131]. Except heuristic search algorithms, traditional AI methods are not able to handle with real world optimization issues. In addition, AI was incompetent to serve the increasing demand of search, optimization and machine learning in various fields our-days. Thus, the successive failures of AI traditional methods open the door to new discipline known as ***Computational intelligence (CI)*** to be applied in various engineering applications.

Many definitions are found in the literature for computational intelligence depending on the development of this new discipline such as: the study that design intelligent agents. An agent is something that acts in an environment and does something. Agents include animals, humans and society. An intelligent agent is a system that acts intelligently. In another word, CI is a set of calculation methods inspired by nature used to solve complex real-world problems. The most accepted definition of CI can be found in [44] as follows:

Definition 3.1. ***Computational intelligence*** is a methodology involving computing that provides a system with an ability to learn and/or to deal with new situations, such that the system is perceived to possess one or more attributes of reason, such as generalization, discovery, association, and abstraction.

For deep understood Prof. James Bezdek, the father of fuzzy pattern recognition theory, defined the computationally intelligent systems as follows[17]:

Definition 3.2. A system is ***computationally intelligent*** when it: deals with only numerical data, has pattern recognition components, does not use knowledge in the AI sense; and additionally when it exhibits:

1. Computational adaptivity,
2. Computational fault tolerance,
3. Speed approaching human-like turnaround,
4. Error rates that approximate human performance.

This definition states that any computational intelligence system should be represented by four main characteristics which are the capability of computational adaptation, fault tolerance, high speed and fewer errors. Adaptation means that parameters can be changed automatically depending on the actual system state following some optimization criteria. In fact, Fault tolerance used mostly for parallel and distributed systems. Generally in parallel or distributed systems, most of computational resources can be replicated at the distributed units of the system; So, if some resources are lost, this can not damage the whole computing process. The most important features in the above definition are computational speed and fewer error rates, which ravel with human error rate. But in fact and generally a high computational speed may yield to poor accuracy in the results.

After few years of the birth of computational intelligence in 90s, new studies went to the synergistic behavior between the computational tools that named this discipline ***Soft Computing***. Generally, when single computational tool is used to solve the problem, soft computing is same as computational intelligence. Whereas, when there is a fusion of two or more computational tools/techniques Soft computing become different from CI. In [127], Pedrycz defined the soft computing as follows:

Definition 3.3. ***Soft Computing*** is a consortium of methodologies which, either singly or in combination, serve to provide effective tools for the development of intelligent systems.

The difference between traditional AI and computational intelligence is that AI is based on hard computing, which is conventional computing, and requires a precisely stated analytic model, whereas CI is based on soft computing[188]. Table 3.1 summarizes the differences between soft and hard computing.

Hard Computing	Soft Computing
Conventional computing requires a precisely stated analytical model.	Soft computing is tolerant of imprecision.
Often requires a lot of computation time.	Can solve some real world problems in reasonably less time.
Not suited for real world problems for which ideal model is not present.	Suitable for real world problems.
It requires full truth	Can work with partial truth
It is precise and accurate	Imprecise.
High cost for solution	Low cost for solution

TABLE 3.1: Difference between Soft and Hard computing

3.3 Principal paradigms of computational intelligence

Generally, CI systems usually include several paradigms such as artificial neural networks, fuzzy systems, evolutionary computation systems, swarm intelligence, and human reasoning. Neural networks are based on their biological counterparts in the human nervous system. Similarly, evolutionary computing draws heavily on the principles of Darwinian evolution observed in nature. Swarm intelligence, in turn, is modeled on the social behavior of insects and the choreography of birds flocking. Finally, human reasoning using imprecise, or fuzzy, linguistic terms is approximated by fuzzy systems. Figure 3.1 shows these four primary branches of CI.

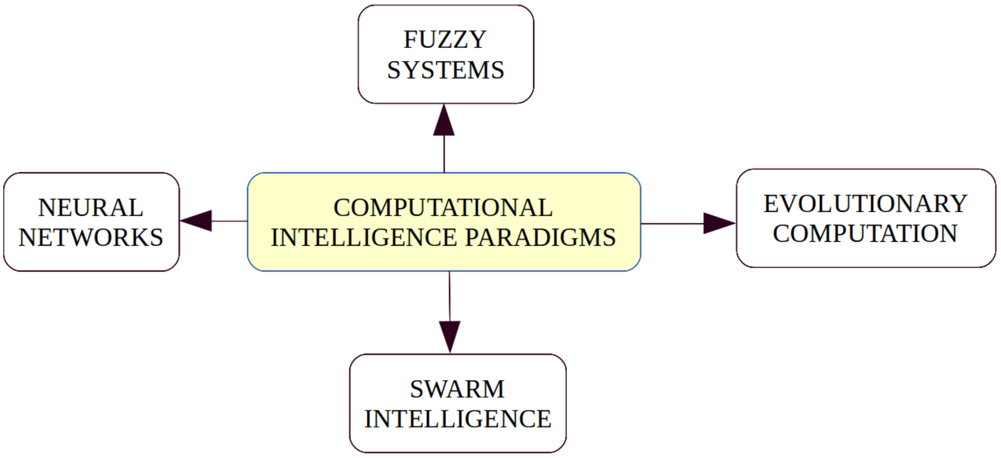


Figure 3.1: Computational Intelligence Paradigms

3.3.1 Neural Networks

The neural network is a computing model which is formerly inspired from the human biological nervous system. This later,is made up of real biological neurons that are connected or related functionally to the nervous system. In neuroscience, they are often identified as groups of neurons that perform a function in specific physiological laboratory analysis. In same, artificial neural networks (ANN) consist of interconnected artificial neurons. The ANN can be used to gain an understanding of biological neural networks, or to solve artificial intelligence problems without necessarily creating a model of a real biological system[156]. It is composed of numerous highly interconnected processing elements (neurons) working in unison to solve specific problems. In general, ANN can learn like a human and it is configured for a specific application, such as pattern recognition or data classification, through a learning process. There are many types of neural networks: The feed-forward neural network is the simplest one, where the information moves in just one direction. Competitive neural networks, which set the different neurons against each other, hoping that the ”winner” will be close to the answer. Etc.

The main advantages[162] in Neural Networks are:

- Learning capacities;
- Generalization capacities;

- Robustness in relation to disturbances.

Whereas, its disadvantages are :

- Impossible interpretation of the functionality;
- Difficulty in determining the number of layers and number of neurons needed to solve the problem.

3.3.2 Fuzzy Systems

In our real world, some concepts have an uncertainty,for instance, let us consider the set of *"people who are near to the conference room"*, in this case it is quite imprecisely defined and difficult to identify peoples closer to the room. Intuitively, we define a distance to the conference room and a threshold to decide who is near to the room. In order to overcome such problems and handle uncertainty in such problems, L.A. Zadeh in [186] introduced the Fuzzy Set Theory. After that, he proposed approximate reasoning based on fuzzy set theory[187].

Fuzzy systems are systems based on fuzzy set theory and/or fuzzy logic. Where, fuzzy set is objects that belong to the set with a certain degree known as membership value which is a continue ranging number that takes its values in [0,1][186].Fuzzy logic aims to formalize the "approximate reasoning" that human beings use in everyday life, providing a formal method that exploits the fuzzy set theory. Fuzzy logic can be viewed as an extension of classical logic, where bi-valued logic (true or false) is extended to a multi-valued logic (degree of truth between true and false). Hence, in fuzzy logic, truth values can assume any value on the interval [0, 1][187]. Generally, The term fuzzy system is used to identify expert systems which aim to apply a more human-like way of thinking in the programming of computers.

The main advantages[162] in fuzzy systems are:

- Capacity to represent inherent uncertainties of the human knowledge with linguistic variables;

- Simple interaction of the expert of the domain with engineer designer of the system;
- Easy interpretation of the results, because of the natural rules representation;
- Easy extension of the base of knowledge through the addition of new rules;
- Robustness in relation of the possible disturbances in the system

Whereas, its disadvantages are:

- Incapable to generalize, or either, it only answers to what is written in its rule base;
- Not robust in relation the topological changes of the system, such changes would demand alterations in the rule base;
- Depends on the existence of an expert to determine the inference logical rules;

3.3.3 Evolutionary computing

In computer science, evolutionary computation is a sub field of artificial intelligence (more particularly soft computing) that can be defined by its algorithms, called evolutionary algorithms, which are based on adopting Darwinian principles. They represent a class of meta-heuristics inspired from the nature evolution. Its principal idea is to evolve an initial population, that contains a set of candidate solutions for the considered problem, using a predefined operations (Selection, crossover and mutation). These algorithms aim to converge the population to an optimal solution based on some criteria, where the degree of convergence is calculated based on the objective function which contains all the variables needed to be optimized (maximized/minimized).

In earlier 70^{th}, the first works in this field have concerned three main types of algorithms: *Genetic algorithm(GA)*[74], *Evolution strategy(ES)*[136] and *Evolutionary programming(EP)*[51]. These three types of algorithms use a global common principle because they are all inspired by the same principles of Darwinism. Afterward, another type of evolutionary algorithms is proposed by Koza in 1992 known as *Genetic programming (GP)*[90] that is an extension of genetic algorithms.

Genetic algorithms(GA) are developed, first, by Holland and its team at Michigan university [74] where their researches focused on the adaptive systems. These algorithms take their importance over the years for its large acceptance to solve a wide number of classical problems including optimization, search and learning issues. Generally, GA have four main components:

- ⋆ *Population of individuals*: it contains a set of candidate solutions considered as initial solution to be evoved during all algorithm iterations,
- ⋆ *Selection function*: This function decide what are the best individuals need to be chosen from the actual population for next generation process,
- ⋆ *Crossover and Mutation functions*: These functions, called genetic operations, which explore new regions from the search space. The first combines two parents to form children for the next generation and the later applies random changes to individual parents to form children.
- ⋆ *Fitness function*: it is an evaluation function that shows if an individual is a good solution or not.

Evolution strategy(ES) similar to genetic algorithms, ES is an optimization technique based on adaptation and evolution. It is a subclass of the evolutionary algorithms which use a population contains a set of candidate solution and evolute it to solve such problem. It was developed in the 1970s by Rechenberg and its co-workers in Germany [136] as a method to solve practical optimization issues. Evolution strategies most commonly address the problem of continuous black-box optimization [69]. Mainly, the objective of evolution strategies is to maximize the ability of a collection of candidate solutions as part of an objective function from a domain. The main advantage of evolution strategy is the self adaptation of the parameters.

Evolutionary programming (EP) originally conceived by Lawrence J. Fogel in 1960 [51], it is a stochastic optimization method similar to Genetic algorithms and an instance of Evolutionary Algorithm from the field of Evolutionary Computation. It is inspired by the theory of evolution by means of natural selection. This approach aims to maximize the suitability of candidate solutions collection within the framework of an objective function. This goal is pursued by using an adaptive model with surrogates for the processes of evolution, specifically hereditary (reproduction with variation) under competition [25].

Genetic programming (GP) is an extension of genetic algorithm that evolve computer programs. It is proposed by Koza and its co-workers in 1992 [90]. The main differences between GP systems and other GA systems are: In a GP system individuals are generally executable structures such as computer programs, while in other GA systems individuals take various forms (typically bit or character strings). Also, in a GP system fitness is normally assessed by executing the individuals, while in GA systems fitness is assessed in ways that depend on the structure of individuals and the problem being solved. The Genetic Programming algorithms aim to use induction to devise a computer program. This is achieved by using evolutionary operators on candidate programs with a tree structure to improve the adaptive fit between the population of candidate programs and an objective function [25].

3.3.4 Swarm intelligence

Some researchers separate swarm intelligence algorithms from EAs whereas others consider them as subclass within the field of evolutionary computation inside the artificial intelligence discipline. For instance, Shi Yuhui, one of the pioneers of optimization by particle swarms, refers to it as an evolutionary algorithm [152].

Swarm Intelligence that's become increasingly popular during the last decade, is a set of algorithms inspired by the social behavior of different individuals in the nature such as ants, termites, bees, and wasps, birds or schools of fish [21]. This approach is a population base method inspired by the collective behavior of the self-organization systems in the nature. Within these algorithms, individuals can interact with each other via simple rules to decide what are the best solutions for such problem. This kind of algorithms was developed after the observable success of swarms to solve complex real problems in nature. In biological swarms individuals are not intelligent, but they are simply agents that have a limited information and easy communication rules. Whereas, the whole swarm gives an intelligent behavior and produces an effective solution to the defined problem. The advantage of these approaches over traditional techniques is their robustness and flexibility which makes swarm intelligence a successful paradigm that deal with increasingly complex problems. A good introduction to Swarm Intelligence with explanations of several natural collective behaviors can be found at[27].

In [21], the authors give an acceptable definition for terms **Swarm**, **Swarm intelligence** and **Self-organization** which is the most important concept in swarm intelligence algorithms, as follows:

Definition 3.4. Swarm is a structured collection of interacting organisms (or agents).

Definition 3.5. Swarm Intelligence is the emergent collective intelligence of groups of simple agents.

Definition 3.6. Self-organization is a set of dynamical mechanisms whereby structures appear at the global level of a system from interactions of its lower-level components.

Mark Millonas, one of the development pioneers of this kind of swarm methods for applications in real life, has presented five principles for swarm intelligence [110]:

- ⋆ *Proximity*: The population should be able to carry out simple space and time computations.
- ⋆ *Quality*: The population should be able to respond to quality factors in the environment.
- ⋆ *Diverse response*: The population should not commit its activity along excessively narrow channels.
- ⋆ *Stability*: The population should not change its mode of behavior every time the environment changes.

- ⋆ *Adaptability*: The population must be able to change behavior mode when it's worth the computational price.

In the last decade, several swarms based algorithms are developed and applied to solve different theoretical and real life problems. For example: Particle Swarm Optimization Algorithm [43, 78], Ant Colony System Algorithm [40], Bee-swarm Optimization Algorithm [42], Firefly Algorithm [175, 176], Cuckoo Search Algorithm [182] and Bat algorithm [177], Etc. In the newt section, we will detail all these algorithms.

3.4 Swarm intelligence algorithms instances

3.4.1 Ant Colony System Algorithm

Ants are insects live in colonies which create a social collaboration between themselves. Ant colonies that range in size from a few dozen living in small natural cavities to large and organized colonies that contains millions of individuals and occupy a large territory. The colonies are described as super-organisms because the ants appear to operate as a unified entity, collectively working together to support the colony. Real ants are capable of finding the shortest path from a food source to their nest by exploiting pheromone information that ants deposit on their paths. Generally, ants (individuals) have a limited physical and cognitive capacities that never produce problems for them. Indeed, they beat this weakness by their collaboration to solve complex issues.

The above ant behavior has inspired Ant colony system, an algorithm based on swarm intelligence. It has introduced in the early 90s by colorni, Dorigo et Maniezzo [29, 40]. The basic idea of this algorithm is to simulate the collective behavior of ants when they search the shortest path from their nest to the food source. With the objective to find the shortest path to source food, ants (individuals) communicate indirectly by changing their environment by adding pheromones. Firstly, ants moves randomly over different paths and each one put its pheromone on the path to keep track of the nest. The pheromone contains information about the path quality. Afterward, individuals follow the path that has the largest quantity of pheromone which is the shortest path. Figure 3.2 shows a way ants exploit pheromone to find a shortest path between two points F and N.

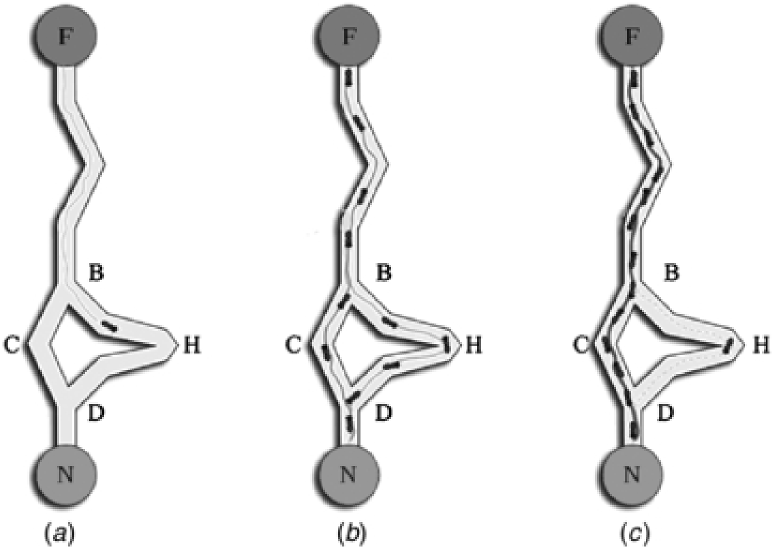


FIGURE 3.2: A way ants exploit pheromone to find the shortest path between two points F and N

In computer science, ants represent solutions for the problem considered to be solved, whereas pheromones are values associated with each solution, and shows its quality. Ant system, an algorithm in which a set of artificial ants cooperates in the solution of a problem by exchanging information via pheromone deposited on graph edges. Ant system is firstly applied for traveling salesman problem [29]. Afterword, many updates are introduced to this algorithm such as: MAX-MIN Ant System (MMAS) [155], ACS [40] and Rank-based Ant System (ASRank)[26]. For instance, **Algorithm 3** shows the pseudo code of ACS algorithm [39].

Algorithm 3 The ACS algorithm

```
Initialize
repeat
    Each ant is positioned on starting node
    repeat
        Each ant applies a state transition rule to incrementally build a solution
        and a local pheromone updating rule
    until All ants have built a complete solution
until END CONDITION
```

Moreover, many successful applications of ant system are provided in several fields such as: data mining [104, 124], vehicle routing problem [15] and robot motion planning [116], Etc.

3.4.2 Particle Swarm Optimization Algorithm

PSO is a well-known nature inspired algorithm. This method was originally proposed by Kennedy et Eberhart [43, 78] in 1995. The algorithm is a stochastic population based search method inspired by the collective behavior of birds flocking and fish schooling. 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 area being searched. The birds do not know where the food is. But they know bow 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 fly through the problem space by following neighbored with the best solutions [75].

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:

$$V'_{id} = \omega V_{id} + c_1 \times rand() \times (P_{id} - X_{id}) + c_2 \times rand() \times (P_{gd} - X_{id}) \tag{3.1}$$

$$X'_{id} = X_{id} + V'_{id} \tag{3.2}$$

Where X'_{id} and X_{id} represent the current and previous position of the id^{th} particle, V'_{id} and V_{id} the current and previous velocity of the id^{th} particle, P_{id} and P_{gd} 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, c_1 and c_2 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[75].

Algorithm 4 The PSO algorithm

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

Indeed, PSO is a simple algorithm that proves its efficiency and shows great results with several and real difficult problems and applications. Nowadays, many variants of PSO are developed according to the new posed problems such as: Hybrid of Genetic Algorithm and PSO, Composite PSO (C-PSO), niching PSO, constrained PSO, multi-objective PSO and discrete PSO, etc [31]. Most modifications to the basic PSO are going towards improving convergence of the PSO and increasing the diversity of the swarm. Since 1995, PSO algorithm has been successfully applied in many fields. For example: Artificial neural network training, scheduling problems, medical diagnosis and data mining [75], Etc.

3.4.3 Bees 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 sophisticated navigation system which helps them to detect new hives accomplish their various biological tasks such as foraging and honey production. Bees collect nectar from flower patches as a food source for the hive. The hive sends out scout's that locate patches of flowers, who then return to the hive and inform other bees about the fitness and the location of a food source via a waggle dance. The scout returns to the flower patch with follower bees. A few scouts continue to search for new patches, while bees returning from flower patches continue to communicate the quality of the patch[25].

Many researchers have been inspired by the collective behavior of bees in the nature in order to create an artificial simulation of this behavior which can be used to solve complex optimization issues. In [84], the authors presented a survey of the algorithms described based on the intelligence in bee swarms and their applications. and his co-workers[33] have divided bees inspired algorithms in two main classes: **Mating-Based Optimization Algorithms** and **Foraging Behavior Based Approaches**. The former draws their inspiration from the mating behavior of honey bee queens such as: *Honeybee Mating Optimization Algorithm* and the latter take inspiration from the mechanisms underlying the foraging process in honey bees. For instance:*The Artificial Bee Colony Algorithm (ABC), Bees Algorithm (BA), etc.*. Below, we will describe one of bees inspired algorithms, as an example, called Bee swarm optimization algorithm (BSO)[42].

The BSO is based on a population of artificial bees cooperating together to solve an instance of an optimization problem. It was proposed by Drias in 2005 [42] and inspired by the foraging bees behavior. Firstly, a set of solutions generated of the search space by using a certain strategy from an initial solution $Sref$. Then, every bee will consider a solution from search area as its starting point. Afterward, every bee shared the best visited solution with all its neighbors through a table named dance. During the next iteration, the reference solution, that is stored every time in a taboo list, will be selected from the dance table according to the quality criterion. The algorithm stops when the optimal solution is found or the maximum number of iterations is reached[36]. **Algorithm 5** shows the pseudo code of Bee swarm optimization algorithm.

Algorithm 5 The BSO algorithm

```
Define the reference solution Sref
while maximum iterations or minimum criteria is not attained do
    Insert Sref in taboo list.
    Search-Area(Sref).
    Assign a solution from SearchArea to each bee.
    for each $bee_k$ do
        Built-Search-Area($bee_k$ ).
        Store the result in the table Dance.
        Communication between bees
    end for
    Choose the new reference solution Sref.
end while
```

3.4.4 Firefly Algorithm

Firefly algorithm (FA) is a new meta heuristic inspired by the flashing patterns and behavior of fireflies, it is proposed by Xin-She Yang in 2008 at Cambridge University[175, 176]. Naturally, fireflies have the ability in emitting light in order to attract other individuals for intermarriage purposes and play a big role to attract potential prey. Also, flashes serve as a protective mechanism. Based on these natural behaviors, Xin-She Yang formulated this firefly algorithm by assuming the following three idealized rules:

- Fireflies are unisex so that one firefly will be attracted to other fireflies regardless of their sex.
- Attractiveness is proportional to their brightness, and for any two fireflies, the less bright one will be attracted by (and thus move towards) the brighter one; however, the intensity (apparent brightness) decrease as their mutual distance increases.
- The brightness of a firefly is determined by the landscape of the objective function.

Based on the above rules, Firefly Algorithm can be summarized as the following pseudocode (**Algorithm 6**)[175]

Algorithm 6 The FA algorithm

```
Generate initial population of fireflies
determine light intensity I for each firefly at its position
Define light absorption coefficient
while not reach maximum iterations or minimum criteria is not attained do
    for each firefly$_i$ do
        for each firefly$_j$ do
            if $I_j > I_i$ then
                Move firefly$_i$ towards j in d-dimension;
            end if
            Attractiveness varies with distance.
            Evaluate new solutions and update light intensity.
        end for
    end for
    Rank the fireflies and find the current best.
end while
Post-process results and visualization.
```

Basically, The movement of firefly toward another more attractive is determined by the following equation:

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2} (x_j^t - x_i^t) + \alpha_t \epsilon_i^t \tag{3.3}$$

Where the second term is due to the attraction while the third term is randomization with α being the randomization parameter and ϵ_i^t is a vector of random numbers drawn from a Gaussian distribution or uniform distribution at time t.[175]. Firefly algorithm is much more based on local optima. Thus the exploration rate in this algorithm is very limited. Therefore, Many variants focuses on this weakness to improve the firefly algorithm such as Fuzzy firefly algorithm[70] which improves the global search of FA by using a fuzzy variable. Some other works have been investigated FA for multi-objective optimization problems[179]. Furthermore, other studies have tried to hybridize FA with another algorithm which enhance on the performance of firefly algorithm[45]. On the one hand applications, FA has been successfully applied in many fields such as:digital image compression, feature selection, scheduling problems, Classifications and clustering[185].

3.4.5 Cuckoo search

Cuckoo search (CS) is one of the latest nature inspired algorithms. It was developed by Xin-she Yang of Cambridge University and Suash Deb of C.V. Raman College of Engineering in 2009[181]. It was inspired by obligate brood parasitism of some cuckoo species by laying their eggs in the nests of other host birds (of other species). In addition, this algorithm is enhanced by the Lévy flights rather than by simple isotropic random walks which make CS more efficient that PSO and genetic algorithms[180]. As doing with FA algorithm, Yang has defined three idealized rules to describe the standard CS algorithms where each egg represents a solution as follows:

- Each cuckoo lays one egg at a time and dumps it in a randomly chosen nest.
- The best nests with high-quality eggs will be carried over to the next generations.
- The number of available host nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability $p_a \in (0, 1)$. In this case, the host bird can either get rid of the egg or simply abandon the nest and build a completely new nest.

Regarding the mentioned rules, Cuckoo Search can be summarized as the following pseudocode **Algorithm 7**[181]. The algorithm uses a balance between random local and global walk which make a good balance between exploration and exploitation, controlled by a probabilistic parameter p_a. The global walk is given by lévy flights and the local random walk can be described as:

$$x_i^{t+1} = x_i^t + \alpha s \otimes H(p_a - \epsilon) \otimes (x_j^t - x_k^t) \tag{3.4}$$

Where x_j^t, x_k^t are two different solutions selected randomly by random permutation, $H(u)$ is a Heaviside function, ϵ is a random number drawn from a uniform distribution, s is the step size. And $\otimes$ means the entry-wise product of two vectors.

Algorithm 7 Cuckoo Search via Lévy Flights

```
Generate initial population of $n$ host nests $x_i$
while not each maximum iterations or stop criteria is not attained do
    Get a cuckoo randomly
    Generate a solution by Lévy flights
    Evaluate its solution quality or objective value $f_i$
    Choose a nest among n(say, j) randomly
    if $f_i < f_j$ then
        Replace $j$ by the new solution $i$
    end if
    A fraction ($p_a$) of worse nests are abandoned
    New nests/solutions are built/generated by 3.4
    Keep best solutions (or nests with quality solutions)
    Rank the solutions and find the current best
end while
Post-process results and visualization.
```

As mentioned above CS search is an efficient algorithm and satisfy the global convergence requirements. This efficiency comes from many advantages that characterize this algorithm. Firstly, cuckoo search has two search capabilities: local search and global search, controlled by a switching/discovery probability. This later makes the search space exploration much more effective on the large scale. In addition, utilization of Lévy flight rather than standard random walks makes CS explore the search space more efficiently than standard algorithms [183].

In the last few years, CS has attracted great interest due to its simplicity, efficiency and flexibility. As result, many variants of CS are created in order to deal with different optimization problems in real life. The original version of CS was developed for continuous optimization problem thus one of the most important variants of CS if the discrete version [123] which deals with discrete optimization issues. Another variant based quantum computing was developed by Layeb known as Quantum inspired CS[94]. Recently, a multi-objective version of CS is presented by Yang[179]. Similar to the previous swarm inspired algorithms, CS have been applied to several applications in various fields such as: Wireless sensor networks[32],Business optimization[184] and Data clustering [144],Etc. A good and detailed review about Cuckoo search algorithm can be found in [48].

3.5 Computational intelligence algorithms for ARM

In this section, a state-of-the-Art for association rule mining algorithms based on CI is described in depth. The proposals presented in this section are mainly grouped into two main categories: *evolutionary* and *swarm intelligence*, where each one includes: *Mono-objective* and *multi-objective* approaches. Figure 3.3 depict a general classification of existing algorithms in ARM field.

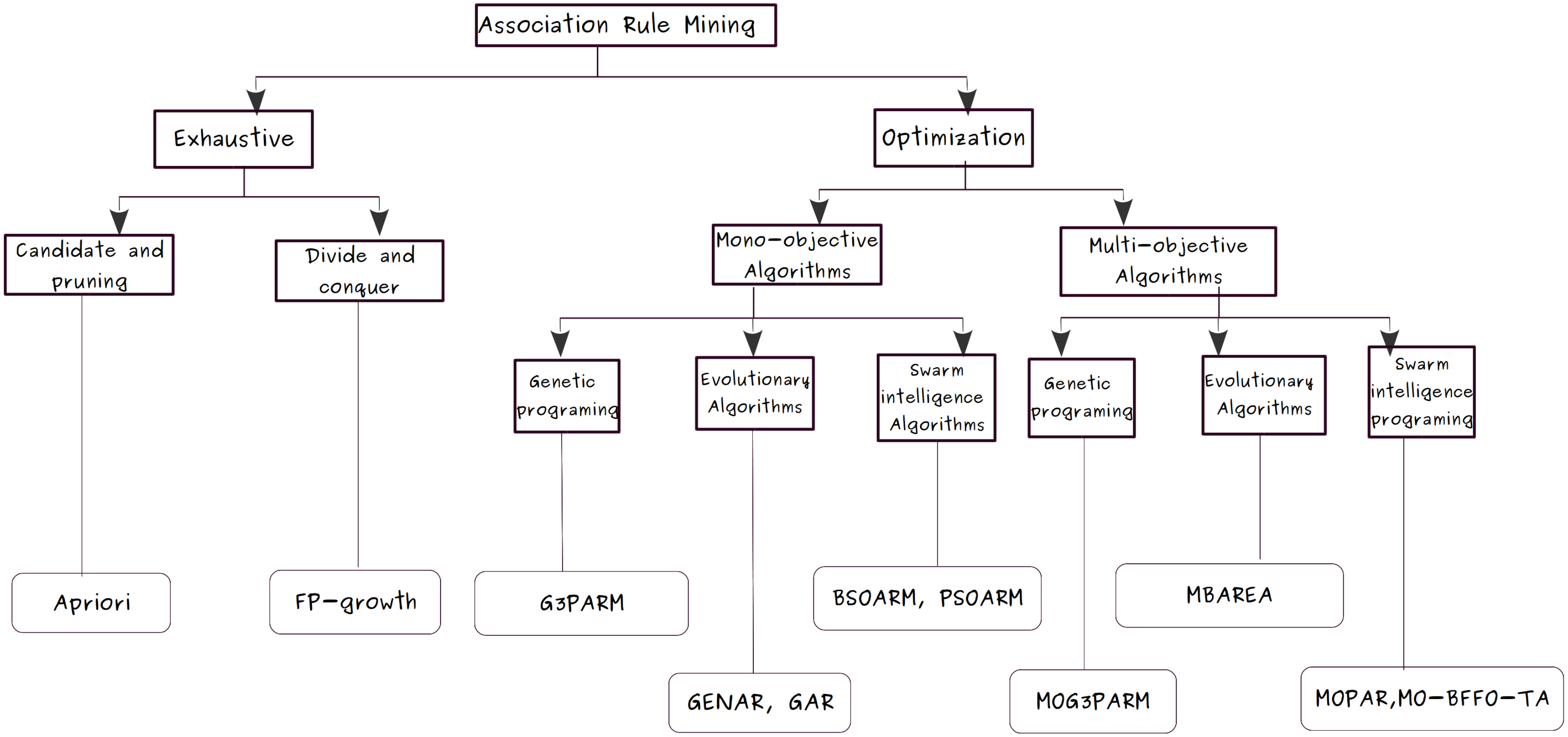


FIGURE 3.3: Classification of existing algorithms in ARM field

3.5.1 Evolutionary algorithms for ARM

ARM process and the relationships among attributes of data sets become indispensable. So, there is a clear need to evolve new automated tools and frameworks to extract useful and comprehensive rules to the final user. Besides, with the growing trend of data web size, ARM becomes a really difficult task. Furthermore, traditional algorithms, previously discussed, need a huge processing time to handle big databases. To overcome this drawback and deal with huge databases within reasonable time, evolutionary algorithms(EA) have been deeply used in ARM [30, 41]. Most of the existing evolutionary approaches are based on genetic algorithms(GA).

Meta et al. proposed Genar [108] and GAR [109] as fist application of genetic algorithm in the field of association rule mining. These algorithms used a bad rule encoding that effect 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 [174], 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 [145] 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.

Another work presented by *Wang et al.* [169] was proposed to mine association rules using genetic algorithm. This proposal, called AGA, is using a new crossover and mutation operations which are calculated according to the evaluation of actual population. This strategy can speed up the optimal solution achievement.

Within [101], *Liu et al.* proposed in improved genetic algorithm based on simulated annealing and quantum computing strategy for mining association rules. This algorithm was implemented to enhance the capability of exploring valuable information from databases with continuous values. The algorithm combines with quantum-inspired genetic algorithm and simulated annealing to find interesting association rules. However, this algorithm favorites the diversification in-face-of intensification that leads to less exploration for the whole search space. In [121], G3PARM algorithm is developed, it is based on genetic programming. The authors used grammar guided genetic programming (G3P) to avoid invalid individuals found by Genetic Programming(GP) process. Also, G3PARM permits multiple variants of data by using a context free grammar. In [41], the authors designed a genetic algorithm called IARMGA and a mimetic algorithm known as IARMMA for association rule mining, which used two approaches that avoid non admissible rules by developing a new strategy called delete and decomposition strategy.

QARGA [107] (Quantitative Association Rules by Genetic Algorithm) is another proposal to extract quantitative association rules based on genetic algorithm. This algorithm is real genetic algorithm implementation which handles with numerical attributes without discretization over all the mining process. In QARGA, each individual is represented by an array of fixed length that is the number of available continuous attributes in data. Each gene represents the upper and lower limit of the interval of each continuous attribute. Additionally, each gene comprises a value that represents the type of attribute. Thus, each attribute could not belong to the individual, belong to the antecedent, or belong to the consequent. In [106], the authors aimed at improving the scalability of quantitative association rule mining techniques based on genetic algorithms to handle large-scale datasets without quality losing. To do this, a new representation of the individuals, new genetic operators and a windowing-based learning schema are proposed to achieve successfully such challenging task. A significant work presented in [76] which analyzed the performance of genetic algorithms in mining association rules based on the variations and modification of GA parameters.

Also, GAs are successfully applied to mine negative association rule in much research. In [8], *Alataş et al.* proposed a genetic algorithm for automated mining of both positive and negative quantitative association rules. The rules considered including, at least, one negative condition. In this GA, each chromosome consists of genes that represent the attributes and intervals. Thus, the ith gene encodes the ith attribute. Each gene has a value to determine whether the attribute will be in the antecedent, in the consequent or not be involved in the rule. Moreover, each gene has a value to determine whether the attribute will be positive or negative, and two values to indicate the lower and upper bounds of the interval. Recently, in [105], *Martín et al.* presented a new Niching Genetic Algorithm to obtain a reduced set of different positive and negative quantitative association rules with a low runtime. To do that, the algorithm extracts rules based on the existence of a pool of external solutions that contains the best rule of each niche found in the search process according to several quality measures, after that, it penalizes similar rules by means of a process based on fitness sharing, and we restart the algorithm leading to a diverse population. Moreover, the user can tune the trade-off between the quality and diversity of the mined rules making use of two thresholds.

Most recently, research has gone over hybridization to improve both quality and time consumption in ARM field. Recently, in [77], the authors present a hybrid method combining both genetic algorithm and particle swarm optimization called hybrid GA/PSO (GPSO). The population is split and sorted based on fitness function value, and both algorithms were run separately on respective sub-population, where the upper half of individuals are fed in GA to create new individuals by crossover and mutation operations. The lower half of the sorted population is fed in PSO for evolution based on personal and global best positions. Hybridization aims to combine the features of both Genetic and Particle swarm optimization algorithms.

3.5.2 Swarm intelligence algorithms for ARM

In addition to evolutionary algorithms, the studies use swarm intelligence algorithms, which are already used to solve many NP-Complete problems, to discover association rules such as particle swarm optimization [91, 147], Bee swarm algorithm [36] and recently, Bat algorithm [71, 72]. Generally, Intelligent algorithms make individuals moving in an n-dimensional space to search the optimum solutions for n-variable function optimization problem. Accordingly, to extract association rules from data, a database may be considered as a large search space and a mining algorithm as an exploration strategy. The datasets are sample space to search and each attribute is a dimension for the miner which aims to maximize/minimize a fitness function that evaluates the quality of extracted rules according to rule measures.

In [91], *Kuo et al.* proposed a particle swarm algorithm to detect association rules, called PSOARM. This proposal defines 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.* [147] 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 [159] Binary Particle Swarm Optimization (BPSO) was used to extract a fuzzy association rule from a transactional database. In [2] 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 [13], we found a review on application of Particle Swarm Optimization in Association Rule Mining.

The authors of [38] 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 [36], 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[35] 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 [37] 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[34].

Many other meta-heuristics are applied to ARM field, ACO_R [124] is an algorithm based on ant colony optimization (ACO) for association rule mining. The major drawback of this algorithm is its high time consuming to search for only one solution. Another work [85] proposed two algorithms known as ARMBGSA and ARMGSA, which are based on the binary and continuous gravitational search algorithm for mining interesting and understandable association rules. In a confrontation with other evolutionary algorithm, these two algorithms generate just a few rules. *Poonam et al.*, in[128], proposed an efficient firefly algorithm (ARMFA) to discover association rules. In this approach each firefly is considered as a rule and fitness value is calculated for each firefly with the aim of discovering high frequency association rules. In [60], the authors propose a new approach based on the penguins search optimization algorithm known as Pe-ARM. This proposal ensures a good diversification over the whole solutions space. Its results reveal that the proposed approach outperforms the well-known ARM algorithms in both execution time and solution quality.

3.5.3 Multi-objective Proposals for ARM

Several measures are introduced to extract the most useful and understandable rules for the user who is the target from using automated tools to get the rules.

Based on this, several efforts dealt with association rule mining as a multi-objective optimization problem where different measures are used in the same algorithm. The literature states show that big efforts are made to develop ARM multi-objective algorithms that use a Pareto Optimization theory. In [111], a multi-objective genetic algorithm approach to mine association rules for numerical data was proposed. In this work, confidence, interestingness and comprehensibility were used to define the fitness function. The results show that the generated rules by this method are more appropriate than similar approaches. In [130], the authors also propose a multi objective association rule mining using genetic algorithm. The proposed method is based on genetic algorithm without taking the minimum support and confidence. Experimental results show that the method extracts the best rules having the best correlation between support and confidence.

Wakabi-Waiswa et al. presented in [164] a new method called (MOGAMAR) to generate high quality association rules with five rule quality metrics: confidence, support, interestingness, lift and JMeasure. Experimental results show that the algorithm produces high quality rules in good computational times. Another approach for optimizing association rules using Multi-objective feature of Genetic Algorithm with multiple quality measures: Support, confidence, comprehensibility and interestingness are presented in [63]. This method effectively reduces the number of generated rules. In [115] the authors proposed a multi-objective genetic algorithm for generating interesting association rules with multiple criteria i.e. support, confidence and simplicity (comprehensibility). Their method can identify the interesting rules without having the user-specified thresholds of minimum support and minimum confidence. ARMGAAM [80] was proposed. This proposal was a new evolutionary algorithm, which generates a reduced set of association rules and optimizes several measures that are present in different degrees based on the datasets are used. This method extends the existing ARMGA model for performing an evolutionary learning, while introducing an initialization process along with an adaptive mutation method. Moreover, this approach maximizes conditional probability, lift, net confidence and performance in order to obtain a set of rules which are interesting, useful and easy to comprehend. In [105] NICGAR was presented. It is a new Niching Genetic Algorithm that aims to obtain a reduced set of different positive and negative quantitative association rules with a low runtime. More recently, a new multi-objective evolutionary algorithm, MBAREA, for mining useful Boolean association rules with low computational cost is proposed in [81]. To accomplish this, the authors proposed a method that extends a recent multi-objective evolutionary algorithm based on a decomposition technique to perform evolutionary learning of a fitness value of each rule. Moreover, this approach maximizes two objectives such as performance and interestingness for getting rules which are useful, easy to understand and interesting.

Another study presented in [14] discussed multi-objective particle swarm optimization algorithm called MOPAR for numerical ARM that discovers numerical association rules. This method uses confidence, comprehensibility, and interestingness to evaluate the extracted rules. We can find a review on multi-objective rule mining in [153]. Recently, in [57] three multi-objective techniques were proposed for mining association rules without specifying minimum support and confidence by optimizing several quality measures. The methods are Multi-objective Binary Particle Swarm Optimization (MO-BPSO), a Multi-objective Binary Firefly optimization and Threshold Accepting (MO-BFFO-TA), and a Multi-objective Binary Particle Swarm optimization and Threshold Accepting (MO-BPSO-TA). Another work was presented in [83] where a meta-heuristic algorithm is used, called MO-CANAR. It is based on multi-objective cuckoo search algorithm which extracts high quality association rules from numeric datasets. The support, confidence, interestingness and comprehensibility are the objectives that have been considered optimization.

3.6 Conclusion

In this chapter, we have seen a general overview on computational intelligence which is the successor of classical artificial intelligence. CI is a set of calculation methods inspired by nature and approaches used to solve complex real-world problems. In addition, we have introduced the different paradigms of CI including: Neural Networks, Fuzzy Systems, Evolutionary computing and Swarm intelligence. This latter is one of the most important paradigms these days, it takes its importance due to its simplicity, adaptability and flexibility. Finally, we have stated a set of swarm based algorithms as examples. Also, this chapter presented a historical review about CI algorithms, including evolutionary and swarm based algorithms, applied to the ARM field in depth. Our work in this thesis is based on bat algorithm(BA) one of the newest and most efficient swarm based algorithms which will be described in details within chapter 4.

Bat Algorithm: An Overview

4.1 Introduction

As previously presented in the previous chapter, swarm intelligence has taken great interest these last years, where numerous algorithms inspired by the behavior of animal communities have been created. For instance: PSO, BSO and FA, etc. This kind of methods got a lot attention from researchers due to their efficiency to solve many real life issues in a reasonable time, as well as their simplicity and flexibility. Bat algorithm (BA) is a new nature-inspired algorithm developed by Xin-She Yang[177] in 2010 and BA has been found to be very efficient.

In this chapter, we describe an overview of bat algorithm which is the essence of our work in this thesis. Before the bat algorithm is introduced, a short review of the natural and biological behavior of bats is described. Then, the original bat algorithm and its several variants are described. Finally, a review of BA applications is presented.

4.2 Natural Bat behavior

Bats, also named blind mice in English, are the second largest mammals after the rodents and they are the only mammals that fly . 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 fly 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 crevices in the dark, As shown in **Figure 4.1**. 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 predators in darker waters. Toothed whales use it for locating food and for navigation underwater, and Shrew use echolocation only for investigating their habitats rather than additionally for pinpointing food, etc.

Echolocation system can be defined 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, microbats can emit about 10 to 20 such 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.

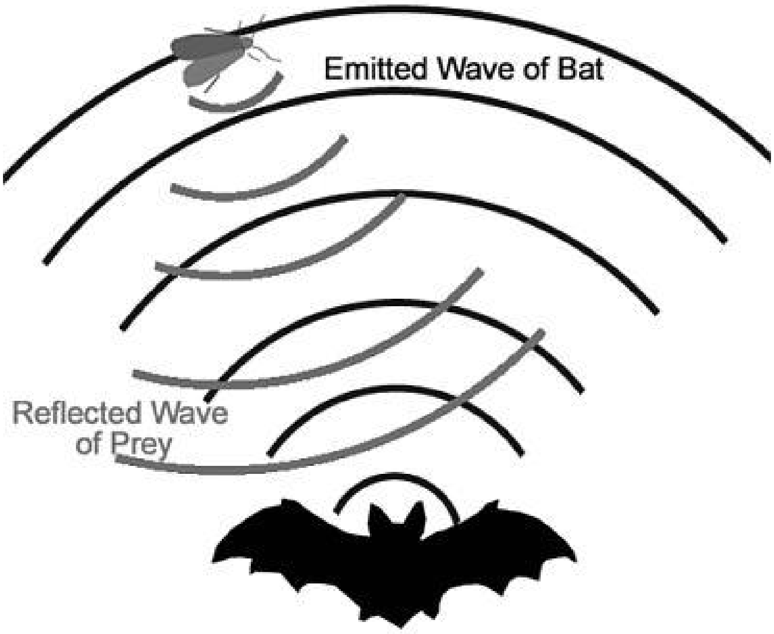


FIGURE 4.1: A bat use echolocation to determine prey

4.3 Original Bat algorithm

Yang[177] 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[177]:

- All bats use echolocation to sense distance, and they also 'know' the difference between food/prey and background barriers in some magical way;
- Bats fly randomly with velocity $v_i$at position x_i with a fixed frequency f_{min}, varying wavelength λ and loudness A_0 to search for prey. They can automatically adjust the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission $r \in [0, 1]$, 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) A_0 to a minimum constant value A_{min}.

Algorithm 8 Original Bat Algorithm

```
Objective Function $f(x), x = (x_1, x_2, ..., x_n)^T$
Initialize the bat population $x_i, i = (1, 2, ..., n)$ and $v_i$
Define pulse frequency $f_i$ at $x_i$
Initialize pulse rates $r_i$ and the loudness $A_i$
while t < Max number of iterations do
    Generate new solutions by adjusting frequency, and updating velocities and locations/solutions [equations 4.1 to 4.3]
    if rand $> r_i$ then
        Select a solution among the best solutions
        Generate a local solution around the selected best solution
    end if
    Generate a new solution by flying randomly
    if rand $< A_i$ and $f(x_i) < f(x_*)$ then
        Accept the new solutions
        Increase $r_i$ and reduce $A_i$
    end if
    Rank the bats and find the current best $x_*$
end while
Post-process results and visualization.
```

Regarding the mentioned rules, BA can be summarized as presented in (**Algorithm 8**)[177]. At the beginning, bat population is randomly initialized with x_i, v_i, f_i for each bat b_i. Let T be the number of iterations. As mentioned in [177], motion of virtual bats, or new solution generation, is performed by updating their frequency, velocity and position according to the following equations:

$$fi = f_{min} + (f_{max} - f_{min})\beta, \tag{4.1}$$

$$v_i^t = v_i^{t-1} + [v_i^{t-1} - x*]fi, \tag{4.2}$$

$$x_i^t = x_i^{t-1} + v_i^t. \tag{4.3}$$

Where $\beta \in [0, 1]$ 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 [177] 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:

$$x_{new} = x_{old} + \epsilon A^t \tag{4.4}$$

Where $\epsilon \in [-1, 1]$ 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:

$$A_i^t = \alpha A_i^{t-1} \tag{4.5}$$

$$r_i^{t+1} = r_i^0[1 - exp(-\gamma t)] \tag{4.6}$$

Where α and γ are constants. At the initialization step of the algorithm, each bat has a different random loudness A_0 which is in [1,2] and random rate r_0 which is in [0,1] as mentioned in [177].

4.4 Variants of Bat algorithm

4.4.1 Multi-swarm bat algorithm

For swarm inspired algorithms, there must exist a balance between exploitation (intensification) and exploration (diversification) of the search space in order to arm the algorithm with robustness and efficiency. Exploiting promising regions yields better local solutions whereas exploration aims at ensuring a more desirable browsing of the whole search space. Recently, multi-swarm (multi-population) technology [97, 96] has been suggested in number of swarm algorithms to maintain more balance between exploitation and exploration, such as ABC[19], genetic algorithms[129], and PSO[194], so the best solution can be obtained and the algorithms can be globally optimal. Multi-swarm approaches are found on the use of numerous sub-populations instead of just one population. The main idea is to make each sub-swarm responsible for determining an effective solution from a specific region of the search space.

Two studies were suggested on multi-swarm bat algorithm[6, 167]. The bat algorithm already performs a balance between global and local search by adjusting the loudness and pulse rate. However, there are so much loudness and pulse rate combinations which making it difficult to choose the most adequate one. In[167], the authors propose a multi-swarm bat algorithm for the global search problem, where an immigration operator is used to exchange information through the swarms, which can make a good trade-off between global and local search. The best individuals of every swarm is put into the elite swarm through a selection operator. **Figure** 4.2 shows the Flowchart of multi-swarm bat algorithm, where the original bat algorithm is utilized in each swarm.

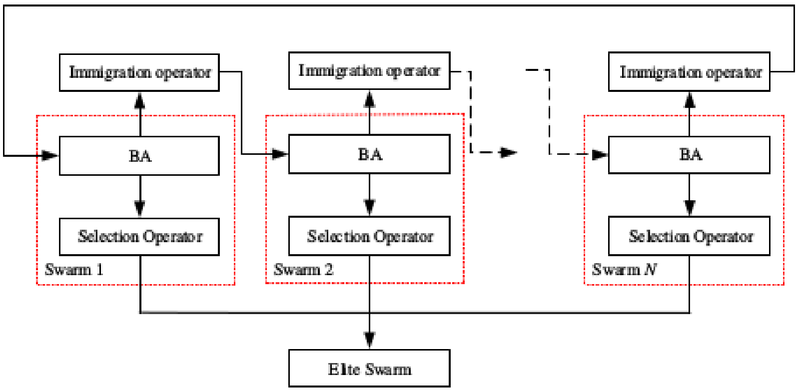


FIGURE 4.2: Flowchart of multi-swarm bat algorithm[167].

In [6], another multi-swarm bat algorithm, called MSBA, is addressed to cope with the premature convergence, which usually occurs in bat algorithm. In the latter case, the evolution process converges to a non-optimal solution, due to an insufficiency of diversity. Within MSBA, when the quality of solution does not improve after a predefined number of iterations, the swarm is split into smaller sub-swarms. The reported results of all these studies demonstrate the superior performance of the multi-swarms optimization algorithms in face of a single-swarm methods.

4.4.2 Bat Algorithm for Multi-objective Optimization

Multi-objective optimization problem (MOOP) is the process to find the vector of solutions to minimize/maximize K objective functions simultaneously where

$K > 1$. This means that multi-objective optimization problems are more complicated than single objective ones where one solution is simple to determine. Mathematically, A general Multi-Objective Optimization problems (MOOP) includes a set of $\boldsymbol{n}$ parameters(decision variables), a set of $\boldsymbol{k}$ objective functions, and a set of $\boldsymbol{m}$ constraints. Objectives and constraints are functions of the decision variables. Generally, The form of a MOOP can be described as minimizing/maximizing a set of objective functions, $f(x) = (f_1(x), ..., f_k(x))^T$ subject to $G_i(x) <= 0, i = 1, 2, ..., n$, by finding the vector $X = (x_1, x_2, ..., x_n)^T$. It is noted that $G_i(x)$ are the constraints that must be satisfied while minimizing/maximizing the objective functions.

In the presence of several objective functions, the notion of "optimum" has changed because MOOP aims to find a vector of solutions rather than a single solution. The notion of "Optimum" has been adopted several propositions related to MOOP but the most commonly accepted term is " Pareto optimum " [28]. In many cases the main goal of multi-objective optimization algorithms is to extract solutions in Pareto optimal set, especially for combinatorial optimization problems where proving the exact solution optimality is almost impossible. Therefore, many methods aim to approximate the Pareto optimal set as much as possible.

In the literature, a huge number of approaches are found to solve MOOPs [28]. The principal three are: aggregating, population-based Pareto and population-based non-Pareto techniques. In aggregating techniques, all the objective functions are combined in one global function. Here the most commonly used technique for the combination is weighted sum method, where each objective function gets a weight as: $G(x) = \sum_{k=1}^{k=n} w_k f_k(x)$. In Pareto based approaches one of the well-known techniques is the Vector Evaluated Genetic Algorithm [149] (VEGA) where the global population is divided into sub-populations equal to the number of objective functions, Each sub-population is used to optimize one of the functions.

In Yang [178], the authors proposed an extended version of the bat algorithm (BA) for multi-objective optimization. Based on the definitions of Pareto front and Pareto optimality. They called this advanced version multi-objective bat algorithm (MOBA). This latter was tested and validated against a subset of multi-objective

Algorithm 9 MOBA Algorithm pseudo code

```
objective functions $f_1(x), ..., f_k(x)$.
Initialize the bat population $x_i$ and $v_i$;
Initialize pulse rates $r_i$ and the loudness $A_i$;
for $j = 1$ to $N$ (points on Pareto fronts) do
    Generate $K$ weights $w_k \geq 0$ so that $\sum_{k=1}^{k} w_k = 1$;
    Form a single objective $\sum_{k=1}^{k} w_k f_k$;
    while ($t <$ Max number of iterations) do
        Generate new solutions and update by [equations 4.1 to 4.3]
        if ($rand > r_i$) then
            Random walk around a selected best solution
        end ifGenerate a new solution by flying randomly
        if rand $< A_i$ and $f(x_i) < f(x_*)$ then
            Accept the new solutions
            Increase $r_i$ and reduce $A_i$
        end if
        Rank the bats according to the best solution;
    end while
    Record $x_i*$ as non-dominate solution;
end for
Post-process results and visualize the best detected rules.
```

test functions. Simulation results showed its efficiency. MOBA that is illustrated in **Algorithm 9**, uses a weighted sum in order to combine all objectives into a single objective. MOBA can be shortly described in few steps as follows:

1. Initialization of the bat population,
2. Combine all the objectives in a single objective.
3. Run the original bat algorithm for the general objective function obtained in the second step.
4. At each point of the Pareto front, the best non-dominated solution is recorded.

4.4.3 Other variants of bat algorithm

The standard BA has many advantages which make it efficient for many applications like data mining, engineering en others when a less time consumption is needed. However, a main weakness can appear when we allow the algorithm to switch to exploitation stage too quickly by varying loudness and pulse rates. Thus, a stagnation will happen after several iterations. To overcome this disadvantage and apply BA with new applications several researchers have created many methods and strategies have been attempted to increase the diversity of the solution and thus to enhance the performance. From the literature we can state the following variants:

- **Fuzzy logic Bat algorithm (FLBA)**[86]: In this paper the authors introduced fuzzy logic in bat algorithms.
- **Chaotic Bat set of rules (CBA)**[99]: The authors presented a chaotic bat algorithm using Lévy flights and chaotic maps to carry out parameter estimation in dynamic biological systems.
- **Binary bat algorithm (BBA)** [112]: In this proposal a binary version of this algorithm is proposed. It deals with binary problems directly by changing their positions from "0" to "1" and vice versa.
- **Differential Operator and Lévy flights Bat algorithm (DLBA)** [173]: A differential operator was introduced to speed up the convergence of BA. Also, the Lévy flight trajectory was incorporated to ensure the diversity of the population against premature convergence.
- **Parameter-free Bat Algorithm(PLBA)**[50]: The original bat algorithm works with five parameters that create possible issues to the users whose not know how to specify the best parameters. Thus, this variant proposed a new parameter less of BA that eliminates this weakness. This technique automated the parameters settings.
- **Improved bat algorithm (IBA)**[168]: An improved version of Bat Algorithm (BA) in combination with a Differential Evolution that is required to select the best individual in bat population. The experimental results proved the performance of IBA in the face of the basic BA model.
- **Hybridized BA with Harmony Search**[165]: In this proposal an improvement was applied to mutate between bats using Harmony Search during the process of bats updating. In fact, new harmonies are generated based

on new bat position after updating. Thus, this new method speeds up the global convergence rate without losing the strong robustness of the basic BA.

- **BA with Mutation**[166]: The original BA was modified by introducing a mutation between bats during the process of new solution generation. That accelerates the global convergence while maintaining the strong robustness of the basic BA.
- **Directed Artificial BA(DABA)**[139]: The algorithm simulates the bat echo emitting and the reflection capturing to be able to create a figure from the surrounding entities. This behavior is used for prey catching. DABA is an artificial algorithm, so it gives the computational meaning/mapping of real bat's hunting concepts.
- **Hybrid BA with Path Relinking(HBA-PR)**[195]: This algorithm was constructed based on the framework of continuous BA, the greedy randomized adaptive search procedure and path relinking were effectively integrated into the standard BA.
- **Multi-objective Binary Bat Algorithm**[11]: This work proposed a multi-objective binary bat algorithm (MBBA) for multi-objective optimization in binary search space. The algorithm uses a modified bat position updating strategy to handle with binary problems. Moreover, a mutation operator is introduced to improve BA local search ability.

Day after day, new research and improvements are presented for Bat algorithm which makes it more efficient and solve their weakness. Also, other variants are proposed according to some new application fields. The next section will review on bat algorithm applications till these days.

4.5 Bat algorithm applications

Bat algorithm has already proven its efficiency since its first application in continuing optimization[177]. That makes researchers uses a bat algorithm to solve several problems in many fields and disciplines, such as: optimization, image processing, scheduling, data mining and others, as illustrated in **table 4.1**. In the following, we detailed the most works that use bat algorithm.

BAT Algorithm applications		
Optimization	**Data Mining**	**Engineering**
Continues Optimization[177]	Classification[113]	Image Processing[150]
Constraint Optimization[56]	Clustering [1]	Spam filtering[120]
Multi-objective Optimization[178]	Feature selection[119]	Industrial design[137]
Unconstrained optimization[61]	Association rule mining[73]	Scheduling[133]
		Sports [47]
		Electronics[137]

Table 4.1: Review of Bat algorithm applications

4.5.1 Optimization

Optimization is the biggest discipline that uses nature inspired algorithms to find the optimal solution for each problem. It has four main types: Continues, Constrained, Unconstrained and Multi-objective optimization, where bat algorithm has been applied with success. In[177], Yang has put the original BA in application for continues optimization with several benchmark functions. This work proves the improvement of convergence rate against GA and PSO. This rate has been more improved in another work. In which BA was hybridized with differential evolution strategies[49]. Moreover, in [165] a hybridization with harmony search is made which speed up the global convergence rate without losing the strong robustness of the basic BA. Xie et al. [173] proposed a new BA with a differential operator and Lévy flights to cope with the slow convergence rate and low accuracy of BA. The experiments showed that the proposed algorithm was feasible and effective, and also had superior approximation capabilities within high-dimensional spaces.

Gandomi et al.[56] have used BA for a constrained optimization, The authors declare, after testing with classical benchmarks and real world constrained problems, that BA is very efficient and overcome, in many cases, the existing algorithms.

Also, BA has a better dynamic control of exploration and exploitation. On the other hand, Goel et al. [61] tested the performance of BA for unconstrained optimization using five standard test functions. The results prove that for bat algorithm to work efficiently for unconstrained optimization problems the dimension of the algorithm should be kept small whereas the population size should be large. In addition, BA has been successfully applied within multi objective optimization problems. As mentioned above, Yang[178] have proposed a version of BA to deal with multi-objective optimization problems which proved its efficiency against other methods. Other work[11] proposed a multi-objective version that deals optimization problems in binary search space.

4.5.2 Data mining

Data mining is one of the fastest growing domain in computer science due to the large data stored last years. Many efficient algorithms are developed for data mining where nature inspired algorithms are some of them. In the last five years bat algorithm has been applied to many times for many problems in data mining, such as: classification, clustering, feature selection and association rule mining, etc.

In[113] a model has been developed for classification using a bat algorithm to update the weights of a Functional Link Artificial Neural Network (FLANN) classifier. The results show the superiority of bat algorithm against FLANN and PSO-FLANN. In [87], the authors proposed a comparison of BA, GA, PSO, BP and LM for training feed forward neural networks in e-Learning context.

Moreover, Bat algorithm has been applied for clustering. Khan et al. [86] presented a method called fuzzy bat clustering for ergonomic screening of office workplaces. The results of the experiments showed that the method is fast, efficient and provides better performance in-face-of fuzzy c-means clustering algorithm. The same authors presented a study of clustering problems using bat algorithm and its extension as a bi-sonar optimization variant with good results [88]. Komarasamy[89] studied K-means clustering using bat algorithm and they concluded that the combination of both K-means and BA can achieve higher efficiency and thus performs better than other algorithms. Recently, Aboubi et al [1] have applied BA for Clustering large data sets called BAT-CLARA. That is based on BA and k-medoids partitioning. The computational results revealed that BAT-CLARA outperforms all compared algorithms to solve the clustering problems.

Also, a new version of BA called the Binary bat algorithm (BBA) was developed for feature selection which is a technique aims to find important information from a set of features. As well, the experiments performed on five standard datasets prove the superiority of BBA in-face-of PSO, FFA and GSA. In[92] a combination of Bat Algorithm with a support vector machine (SVM) classifier for simultaneous feature and optimal SVM parameters selection, to reduce data dimensionality and to improve IDS detection.

4.5.3 Engineering

Engineering is a crucial field for nature inspired algorithm applications. Thus, BA becomes important technology applied for solving issues in engineering practice. In image processing, Senthilnath et al.[150] proposed to use BA to classify Multi-spectral Satellite Image. In this proposal, a partitional clustering used to extract information in the form of cluster centers, then the extracted clusters are validated. The performance of the Bat Algorithm is compared with the traditional K-means, GA and PSO, which prove BA can be successfully applied to solve crop type classification problems. Other work can be found within the field of image processing in [193]. This later presented a new bat algorithm with mutation, known as BAM, between bats during the generation of new solutions for image matching issues. The results illustrated that this method was more efficient than differential evolution and studied genetic algorithm models. Within the same field, Abdel-Rahman et al. [7] proposed to use bat algorithm for articulated human motion tracking from multi-view video data. Results showed that BA performs better than Particle Filter, Annealed Particle Filter and Particle Swarm Optimization algorithms.

Scheduling problem is one of the biggest issues in engineering where BA has been applied successfully. Musikapun and Pongcharoen[117] solved multi-stage, multi-machine, multi-product scheduling problems using bat algorithm, and they solved a class of non-deterministic polynomial time (NP) hard problems with a detailed parametric study. In addition, BA was used to solve multistage hybrid flow shop scheduling problems which are known as strongly NP-hard problem. The computational results show that the proposed BA is an efficient approach for solving the HFS scheduling problems[103]. Recently, In [133], the authors used binary bat algorithm for mapping of tasks and resources is done using this method in a cloud, where the optimal resources are selected such that the overall cost of the workflow is minimal.

In [10] optimal design of Power System Stabilizers (PSSs) in a multi-machine environment parameter tuning problem is converted to an optimization problem which is solved by BAT algorithm. Within this work results demonstrated the robustness of the proposed algorithm over the GA and conventional one. Another work uses BA to obtain the optimal solution of economic load dispatch[18]. Experimentation showed that BA has good convergence property and better in quality of solution than PSO.

4.6 Conclusion

In this chapter, we presented a general review on Bat algorithm which is the essence algorithm in our work proposed in the thesis. We have described bats behavior in the nature that is the source of inspiration for the algorithm. Afterward, the original bat algorithm and its variants were presented in details. Finally, we stated several application domains where BA have been applied successfully. At the time when writing this thesis more works uses a bat algorithm in many fields and disciplines can be found on Google scholar and other databases, which proves the popularity of BA and its efficiency against other methods.

Bat algorithm for ARM

5.1 Introduction

Data mining is the process of extracting useful knowledge from a large database by using software and tools to look for discrimination and expressive patterns. This process helps companies to focus on important information in their historical databases to make decisions. Association rule mining is one of the most important domains in data mining. It aims to extract correlations, frequent pattern and associations between the items in databases.

In this chapter, we propose a bat-based algorithm (BA) for association rule mining (BAT-ARM). Our 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 our proposed algorithm is tested on several generic datasets with different number of transactions and items.

5.2 The BAT-ARM algorithm

5.2.1 Encoding

In the literature, several representations of the rule to mine the ARM using genetic algorithms or meta-heuristic algorithms can be found. In ARMGA algorithm [174], the rule is represented as a chromosome of *k+1* length, where the first position is the cut point that separates the antecedent and the consequent of the rule. Positions 1 to k are the index of the items, Let the rule $X \Longrightarrow Y$, $X = \{A_1, ..., A_j\}$ and $Y = \{A_{j+1}, ..., A_k\}$. Another representation is considered by [30] where all items appearing in the chromosome. So, if we have n item, then we get *2n* length chromosome. In this representation, we have four codes that can be assigned to each item:

- **00**: if the referred item is in the antecedent of the rule,
- **01** or **10**: the referred item is not included in the rule,
- **11**: if the referred item is in the consequent of the rule.

In our approach, we use the first representation where each solution X represents a rule and contains k item. Therefore, the solution$X$ contains *k+1* positions. Assume $I = \{I_1, I_2, ..., I_n\}$ the set of all items in the database. The rule $X \Longrightarrow Y$ is encoded as follows:

j	I_1	....	I_j	I_{j+1}	....	I_n

Where the j separates between the antecedent and the consequent of the rule, if item i^{th} item in I is in the rule then the position k contains i else the position contains 0 where $0 < k <= n + 1$;

5.2.2 Fitness Function

As mentioned above, in association rule mining, the rule is accepted if its support and confidence satisfy user support and confidence threshold. Each work on evolutionary algorithms for mining association rule uses its fitness function for the individuals. Our fitness function is described as follows:

$$f(R) = \begin{cases} \alpha conf(R) + \beta supp(R)/\alpha + \beta & \textit{if accepted rule } R \\ -1 & \textit{otherwise} \end{cases}$$

Where $R = X \Longrightarrow Y$, and let α and β be two empirical parameters.

5.2.3 Virtual Bat Motion

In[177] the authors present a mathematical simulation of the natural bat movement, where the frequency f_i, the velocity v_i and the position (solutions) x_i for each virtual bat are described, and these values are generated based on three equations presented in 4.1, 4.2, and 4.3. In our approach we propose a new description for bat motion related to association rule mining. The same concepts are defined: frequency, velocity and position.

- **Frequency** f_i: Presents how much this bat is strong to change its position. In other words, it means how many items can be changed in the actual rule, where the maximum frequency f_{max} is the number of attributes in the dataset and the minimum frequency f_{min} is 0.
- **Velocity** v_i: Presents where the changes will be started.
- **Position** x_i: It is the new generated rule based on new frequency, velocity and the loudness.

The new positions (rules) are extracted based on frequency and velocity of each virtual bat and they are updated at each iteration using Equations 5.1 and 5.2.

$$f_i^t = 1 + (f_{max})\beta, \tag{5.1}$$

Algorithm 10 Generate new solution in BAT-ARM

```
input: Rule $x_{i-1}$, Frequency $f_i$, velocity $v_i$, loudness $A_i$
output: new rule $x_i$
while ($v_i <$ Frequency $f_i + v_i$) do
    if ($rand > A_i$) then
        Item at ($v_i$) $\longleftarrow$ Item at ($v_i$) + 1
    else
        Item at ($v_i$) $\longleftarrow$ Item at ($v_i$) - 1
    end if
    if (Item at ($v_i$) less or equal to 0 or greater than number of attributes) then
        Item at ($v_i$) $\longleftarrow$ 0
    end if
    if The new rule contains duplicated items then
        Keep one randomly and delete the others.
    end if
    increment ($v_i$)
end while
```

$$v_i^t = f_{max} - f_i^t - v_i^{t-1}, \tag{5.2}$$

The new position (rule) x_i computed by algorithm 10, such that, if the loudness is less than a random value *rand*, the value of current bit increases, otherwise it decreases. Also, If the value is out of the interval [1..N], the value is replaced by 0. The new generated rule may contain duplicated bits (items). In this case, the algorithm keeps a random bit from the duplicated ones and deletes the others. Hence, the algorithm will never generate an invalid rule. This operation is repeated starting from the bit with current velocity to achieve the actual frequency. The modified bat algorithm (BAT-ARM) pseudo code is presented in **Algorithm 11**

5.2.4 Complexity of BAT-ARM

Association rule mining is one of the most attractive problem in ***NP***-class[12]. 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), with the use of **Algorithm 10**. So the number of modified bits is the $frequency_i - velocity_i$. In the worst case the frequency will be equal to F_{max} and the velocity is 0, then the items will be changed. Here the complexity is $O(n \times Maxiterations \times k)$

Algorithm 11 BAT-ARM Algorithm

```
input: a database of transactions
output: association rules
objective function(fitness function)
Initialize the bat population $x_i$ and $v_i$;
Define pulse frequency $f_i$ at $x_i$;
calculate the fitness of each initial position $f_i$ at $x_i$;
Initialize pulse rates $r_i$ and the loudness $A_i$;
while ($t <$ Max number of iterations) do
    for all Bats in the population do
      A* {
        Adjust frequency $f_i$ and update velocity $v_i$ using Eq.5.1,5.2;
        Generate a new solution $x_i$ ;
        if ($rand > r_i$) then
            Generate a local solution $x_i$ around the best solution $x_i*$ by changing
            just one item in the rule
        end if;
        if ($f(x_i) > f(x_i*)$) then
            Accept the new solution $x_i$;
            $x_i* = x_i$;
            Increase $r_i$ and reduce $A_i$ using Eq.4.5,4.6
        end if;
        Determine the global best solution $x*$ amoung $x_i*$;
      }
    end for
end while
```

5.3 Analysis and comparative study

To evaluate our algorithm we tested it on a several synthetic common standard databases [62] with different number of transactions and items. The programs are written in Java and run on Intel core I5 machine with 4Go of memory running on Linux Ubuntu. Results are described in the following section.

In our experiment, we change the two essential parameters in our algorithm: the number of bats n and the number of iterations t, and we fixed α and β to 1. As mentioned in [177], α and γ in **Equa. 4.5,4.6** are $\alpha = \gamma = 0.9$.

Table 5.1 presents the results of running our proposed algorithm on synthetic common database of 1000 transactions and 40 items were created with the IBM dataset generator[5]. The initial positions are generated randomly, and the minimum support and the minimum confidence are fixed to 0.2 ad 0.5 respectively.

Bats	iteration	execution time (s)	best fitness	Memory usage (Megs)	FPgrowth run-time(s)	Memory usage (Megs)
10	25	0.56	0.3795	1.31	0.73	16.97
	50	1.06	0.4039			
	100	2.03	0.4087			
	150	2,9	0.4087			
	200	3	0.4118			
25	25	1	0.3815	2.45	0.73	16.97
	50	2	0.4138			
	100	5	0.4137			
	150	7	0.4162			
	200	9	0.4087			
50	25	2	0.4087	2.74	0.73	16.97
	50	5	0.4087			
	100	10	0.4087			
	150	15	0.4118			
	200	19	0.4138			

Table 5.1: ARM Bat results with common database (1000 transactions and 40 item)

The results show that our algorithm yields a good performance in term of time and memory usage with max time 19 seconds and 2.74 Megs of memory usage. The number of iterations is 200 and the number of bats is 50. On the other hand, FPgrowth algorithm is more efficient in term of time with 0.73 second with higher memory usage.

Table 5.2 shows the results of running our tests on CHESS[62] database that contains 3,196 transactions and 75 items. The results show that our algorithm gives a good runtime with a good quality of rules generated and less memory usage. Our algorithm goes to maximize the fitness function till 0.9295 with support 0.92 and confidence 0.93 and maximum memory usage 48.63. For FPgrowth algorithm the runtime grows up because of the number of transactions in the CHESS dataset.

Table 5.3 shows the results of our tests on mushroom[62] database that contains 8124 transactions and 119 items. Again, the results confirm the efficiency of our algorithm with the growth of the number of transactions and items in database in term of execution time and memory storage. The maximum usage of memory was

Bats	iteration	execution time (s)	best fitness	Memory usage (Megs)	FPgrowth run-time(s)	Memory usage (Megs)
10	25	2	0	42.51	523	104.55
	50	9	0.7034			
	100	13	0.8044			
	150	22	0.8209			
	200	28	0.8152			
25	25	12	0	45.95	523	104.55
	50	38	0.632			
	100	72	0.6464			
	150	60	0.8728			
	200	73	0.8152			
50	25	34	0.5795	48.63	523	104.55
	50	38	0.8837			
	100	67	0.8565			
	150	125	0.9044			
	200	141	0.9295			

Table 5.2: ARM Bat results with Chess database (3,196 transactions and 75 items)

170 Mo when the parameters are 200 iterations and 50 bats. On the other hand, FPgrowth algorithm goes till 291 Megs of memory and runtime of 1165 seconds.

Based on the conducted experiments, we observe that our algorithm provides a great performance in term of CPU-time and memory usage. Thanks to the echolocation concept of the bat algorithm that can determine which part of the best rule have changed to get a better position (rule) for the actual bat.

5.4 Conclusion

In this chapter, we presented a new application of bat algorithm for association rule mining. The proposed approach is inspired from bats behavior and based on an echolocation concept to generate new positions (rules). Compared to FP-growth algorithm, our proposal proved its efficiency in term of time, memory usage and quality of generated rules with maximizing the fitness function. The main drawback in our algorithm is the lack of communication between bats that

Bats	iteration	execution time (s)	best fitness	Memory usage (Megs)	FPgrowth run-time(s)	Memory usage (Megs)
10	25	6	0	128.67	1165	291.1
	50	10	0.6063			
	100	26	0.6329			
	150	41	0.6927			
	200	68	0.6927			
25	25	18	0	156.69	1165	291.1
	50	37	0.5622			
	100	86	0.6329			
	150	115	0.6927			
	200	199	0.7070			
50	25	16	0.6758	170.29	1165	291.1
	50	61	0.6464			
	100	151	0.6927			
	150	253	0.7198			
	200	341	0.7376			

TABLE 5.3: ARM Bat results with Mushroom database(8124 transactions and 119 items)

can generate redundancies in the positions while searching a new one. In the next chapter, we propose a cooperative bat algorithm for mining association rules to solve the weakness of BAT-ARM

5.5 Publications Associated with this Chapter

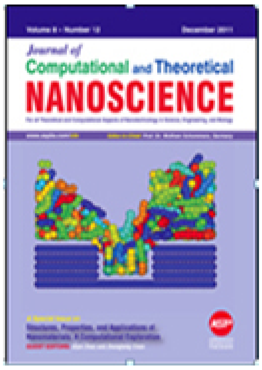
- **International Journal:**

  TITLE: Association rule mining based on bat algorithm

  AUTHORS: HERAGUEMI Kamel Eddine, KAMEL Nadjet, and DRIAS Habiba

  DOI: 10.1166/jctn.2015.3873

  PUBLICATION DATE: **July 1, 2015**



  JOURNAL OF COMPUTATIONAL AND THEORETICAL NANOSCIENCE

  IMPACT FACTOR: **1.42**

  H-INDEX: **34**

- **International Conference:**

  HERAGUEMI Kamel Eddine, KAMEL Nadjet, and DRIAS Habiba. Association rule mining based on bat algorithm. *In : Bio-Inspired Computing-Theories and Applications.* Springer Berlin Heidelberg, 2014. p. 182-186.

Multi-swarm Cooperative Bat Algorithm for ARM

6.1 Introduction

In general, organizations attempt to achieve their objectives through collaboration with other corporations rather than concurrence with them. Also, in population based algorithms, competition between solutions usually enhances performance, whereas better improvements can be obtained through collaboration. In this chapter, we propose a multi-swarm cooperative bat algorithm for mining association rules within a transactional database named MSB-ARM. This proposal supposes to utilize several collaboration topologies to maintain the best trade-off among exploration and exploitation to extract best rules. In the following, we describe the dataset representation and the rule encoding used for the algorithm. After that, we explain the objective function that is considered for maximization and rule quality assessment. Finally, the proposed three collaborative strategies(Ring, Master-Slave and Hybrid) will be introduced.

6.2 The MSB-BAT algorithm

6.2.1 Dataset representation and Rule encoding

Dissimilar to BAT-ARM, which scans the whole dataset to calculate the support of each item-set, in this approach, we apply vertical layout database. In this way, the support of an item-set can be easily computed by a simple intersecting any two subsets [148].

In association rule mining, there are two main layout techniques for representing rules to mine, known as binary encoding and integer encoding. In the first, each rule (solution) is represented by a table S of n elements, where $S[i] = 1$ if the item i is in the rule and 0 otherwise. However, in integer encoding, the solution X is represented by a vector S, which contains *k+1* positions, where $S[0]$ separates between the antecedent and consequent of the rule. If i^{th} item in the database is in the rule then position k contains i and 0 otherwise, where $0 < k <= n + 1$. In this proposal, we opt with integer encoding.

Example 6.1. *For instance, let* $I = \{i_1, i_2, ..., i_{10}\}$ *be a set of items:*

- $X1 = \{3, 1, 5, 0, 6, 2, 0, 0, 7, 0, 0\}$ *represents the rule* $i_1, i_5 \Rightarrow i_6, i_2, i_7$*,*
- $X2 = \{5, 0, 1, 3, 6, 2, 0, 0, 0, 0, 7\}$ *represents the rule* $i_1, i_3, i_6, i_2 \Rightarrow i_7$*,*
- $X3 = \{2, 2, 0, 0, 0, 0, 3, 0, 0, 8, 0\}$ *represents the rule* $i_2 \Rightarrow i_3, i_8$*,*

6.2.2 Cooperative strategies in multi-swarm bat algorithm for ARM

Different cooperation plans are suggested for use in our proposal with the goal of finding the optimal solutions and best rules based on BAT-ARM generation process. The collaboration topologies proposed in this section are: Ring, Master-Slave and Hybrid strategy.

6.2.2.1 Ring strategy

Ring cooperative strategy is one of the most famous strategies and basic schema used in cooperative algorithms. Within this strategy, a number of sub-swarms are evolved in parallel to found the best local solution. Also, each swarm interchanges its best solutions (*lbest*) with its neighbor, if there is no improvement in the solution quality after a predefined number of iterations. This circular communication of *lbest* is performed in the manner of directed ring, which is illustrated schematically in **Figure** 6.1. **Algorithm** 12 illustrates the pseudo-code of MSB-ARM using the ring strategy.

This strategy requires fewer communications relative to the other plans, where all the swarms communicate between themselves. It must be then more efficient in term of time consuming.

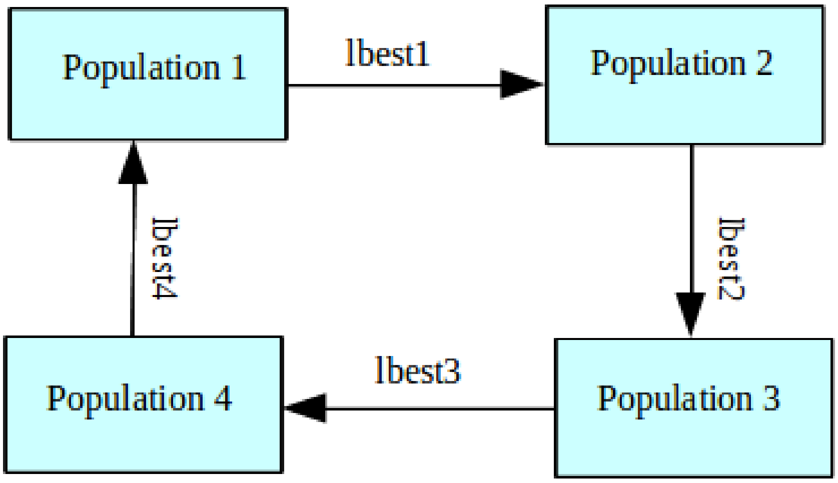


FIGURE 6.1: Ring cooperative strategy for MSB-ARM

6.2.2.2 Master-Slave strategy

A multi-population collaborative strategy known as Master-Slave is designed, besides the previous one. One swarm plays the role of master swarm while others are slaves and all swarms utilize BAT-ARM generation process. Workers are evolved

Algorithm 12 Ring cooperative strategy for MSB-ARM

```
Input: Number of Sub-populations, Number of bats in each Sub-population
Output: Association Rules
Initialize the bat algorithm parameters;
calculate the fitness of each initial position;
while ($t <$ Max number of iterations) do
    for all Subpop do
        for all Bat in population do
            Apply part $A^*$ of algorithm 11 ;
        end for
    end for
    if after a predefined number of iterations, there is no improvement in solution quality then
        communicate Lbest of each sub-population to its neighbor;
    end if
    Increase the iteration;
end while
Post-process results and visualization the best detected rules;
```

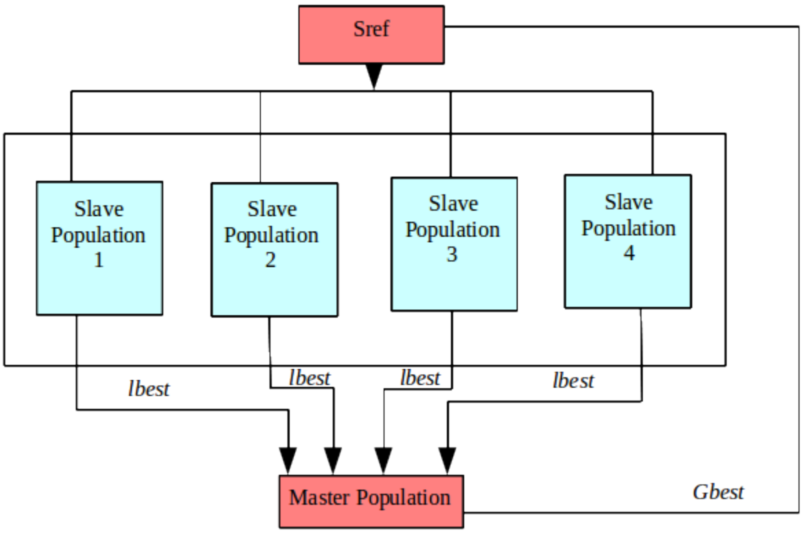


FIGURE 6.2: Master-Slave cooperative strategy for MSB-ARM

in parallel to reach their best local *lbest* solutions, which are sent to the Master population. This latter determines the global best solution *gbest* among all local best solutions transmitted by the slaves. Thereafter, all the slaves are notified by the master about the new global solution *gbest*, which will be a new reference (Sref) for the next iteration.

This topology is illustrated schematically in **Figure 6.2**. The relationship between master population and slave populations maintains the right balance of exploitation and exploration, which is necessary for successful optimization process. **Algorithm 13** illustrates the pseudo-code of Master-Slave strategy.

Algorithm 13 Master-Slave cooperative strategy for MSB-ARM

```
Input: Number of Slave populations, Number of bats in each slave population
Output: Association Rules
Initialize the bat algorithm parameters;
calculate the fitness of each initial position;
while ($t <$ Max number of iterations) do
    for all Slave-population do
        for all Bat in population do
            Apply part $A^*$ of algorithm 11 related to the global best;
        end for
        send the best solution of each population to master population;
    end for
    for all Bat in Master-population do
        Apply $A^*$ in algorithm 11 to generate the global best solution gbest;
    end for
    Notify All slave populations with the new gbest;
    Clean out the master population;
    Increase the iteration;
end while
Post-process results and visualization the best detected rules;
```

6.2.2.3 Hybrid strategy

In this section, a new hybrid strategy is innovated, which is a combination between the both previous strategies: Ring and Master-Slave. The main idea in this hybridization is the injection of ring cooperation between the slave populations into the master slave strategy. The schematic view of this strategy is depicted in **Figure 6.3**.

In this strategy, a copy of the best solution of each slave(*lbest*) is delivered to the master population in each iteration. Moreover, if after a predefined number of iterations there is no improvement in the solution quality, each slave swarm shares its best rule information with its neighbor. Finally, the general best solution is

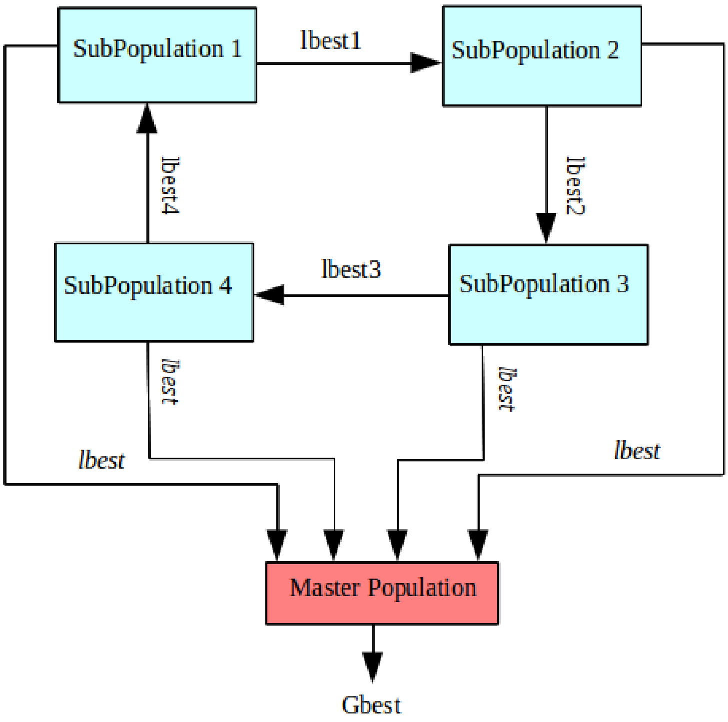


FIGURE 6.3: Hybrid cooperative strategy for MSB-ARM

detected by the master swarm. **Algorithm 14** illustrates the pseudo code of this strategy.

This strategy aims at maximizing the chance of finding the optimal solution because it includes a diversification of search. On the other hand, as the swarms are working independently and in real parallelism, the overall algorithm must be more effective than the previous ones.

6.3 Experimental Study

6.3.1 Datasets description

In order to demonstrate the forcefulness of above described strategies, a series of experiments were carried out on nine datasets from various and well known repositories in the domain of association rules, such as Frequent and mining dataset Repository[62], Bilkent University Function Approximation Repository [65]. Table

Algorithm 14 Hybrid cooperative strategy for MSB-ARM

```
Input: Number of Slave populations, Number of bats in each slave population
Output: Association Rules
Initialize the bat algorithm parameters
calculate the fitness of each initial position
while (t < Max number of iterations) do
    for all Slave-population do
        for all Bat in population do
            Apply part A* of algorithm 11 ;
        end for
    end for
    if at more than a predefined number of iterations there is no improvement then
        send Lbest of each sub-population to its neighbor;
    end if
    send the best solution of each population to master population;
    for all Bat in Master-population do
        Apply part A* of algorithm 11 to generate the global best solution gbest;
    end for
    Clean out the mater population;
    Increase the iteration;
end while
Post-process results and visualization the best detected rules;
```

6.1 sheds light on the description of different datasets used in the experimentations. Also, from the table, it can be observed, that the datasets differ from each other on the number of transactions and items. For instance, Chess dataset has 3196 records with 75 items whereas mushroom has much more transactions and items.

DATASET NAME	*Transactions size*	*Item size*
Basketball	96	5
Bodyfat	252	15
IBM-Quest-standard	1,000	40
Quak	2,178	4
Chess	3,196	75
Mushroom	8,124	119
Pumbs star	40,385	7,116
BMS-WebView-1	59,602	497
Connect	100,000	999

Table 6.1: Datasets description

In this section, datasets and experiments set-up are described. Next, the results obtained by the different collaborative strategies will be presented. Finally, we will expose a comparative study with several approaches including single objective and multi-objective optimization methods recently developed in the field of ARM.

6.3.2 Experimental Set-up

All the algorithms were scripted in Java and executed on intel core I5 machine with 4GB of memory running under Linux Ubuntu.

6.3.3 Analysis and comparison

6.3.3.1 Stability analysis

In these tests we aim to determine the stability of each cooperative strategy in dealing with the fitness function in terms of iterations. On the other hand, the CPU time is reported and compared to BAT-ARM outcomes. In this study, we utilize three datasets (*IBM Quest Standard, Chess and Mushroom*). We execute each strategy 20 times with fixed thresholds of support and confidence to 0.2 and 0.5 respectively. The number of bats fixed to 50 divided in 10 swarms and the average results are considered to be analysed. The outcomes obtained in these tests are shown in **Figure** 6.4 and 6.5. **Figure** 6.4 depicts the CPU time consumption for the three cooperative strategies and BAT-ARM, run on the three tested datasets. It shows that Ring strategy outperforms the other strategies in terms of execution time. This can be explained by the total parallelism of the sub-swarms execution.

In addition to reasonable runtime, MSB-ARM needs to yield rules with high quality. **Figure** 6.5 provides a schematic view of the fitness function for the three cooperative strategies and BAT-ARM, run on three tested datasets. It is observed that all strategies achieve the stabilization point almost at 200 iterations. Besides, all the strategies outperform BAT-ARM algorithm in terms of fitness, except with mushroom dataset, where the ring strategy gives less performance. These promising results are due mainly to the communication that exists between sub-swarms and that give rise to more balance among the intensification and diversification in MSB-ARM proposal.

6.3.3.2 Comparative study to similar approaches

In this section, we compare the results of our proposal and each one of its cooperative strategies to similar methods recently developed in the field of rule mining. The parameters are set as follows: Iterations equal to 200, Number of swarms equal to 10 with 5 bats in each sub-swarm. The experimental results were compared against the following five well-known algorithms:

- Bees swarm optimization algorithm for ARM, i.e., BSO-ARM[36]
- Hybrid Bees swarm optimization with tabu-search algorithm for ARM, i.e., HBSO-TS[35]
- Ant colony optimization based Algorithm for ARM, i.e.,ACO_R[124]
- Grammar guided genetic programming for ARM, i.e., G3PARM[121]
- Binary and continues gravitational search algorithm for ARM, i.e., ARM-BGSA [85]

The obtained results express the average of 20 executions. Table 6.2 summarizes the outcomes of MSB-ARM with its different cooperative strategies (Ring, Master-Slave and hybrid), HBSO-TS, ACO_R, G3PARM and ARMBGSA in terms of the number of generated rules from five different datasets. It can be noticed that all the cooperative strategies outperform the other algorithms. Furthermore, master-slave strategy shows its superiority on the other strategies which can be argued by the great balance between the global diverse exploration and local intensive exploitation in this strategy.

With evolutionary algorithms, the number of rules generated is not enough to judge the algorithm, but the quality of extracted rules must be reviewed to assess

DATASET	*MSB-ARM*			*HBSO*	*ACO*$_R$	*G3PARM*	*ARMBGSA*
	RING	*Master-Slave*	*Hybrid*				
Basketball	109	369	131	91	37	32	28
Quak	82	93	81	77	58	82	39
IBM-Quest-std	420	850	527	407	67	328	68
Chess	751	802	713	356	85	395	88
Mushroom	755	791	784	758	92	829	115

Table 6.2: Comparing our approach to existing approaches w.r.t number of rules

the performance. Table 6.3 shows all the results obtained from each one of the algorithms including our collaborative strategies in terms of fitness function. The aim is to maximize this function, which ensures the quality of extracted rules. We observe that BSO-ARM, ACO$_R$, G3PARM and ARMBGSA provide less performance compared to HBSO-TS and our proposed algorithms, which can be explained by the carefully chosen intensification and diversification strategies. As well, we can notice the superiority of master-slave strategy using this measure too, with the small datasets that do not exceed 10000 transactions. In the case of large datasets which have more than 40000 transactions, the hybrid strategy shows its efficiency and outperforms all the other methods including the master-slave strategy.

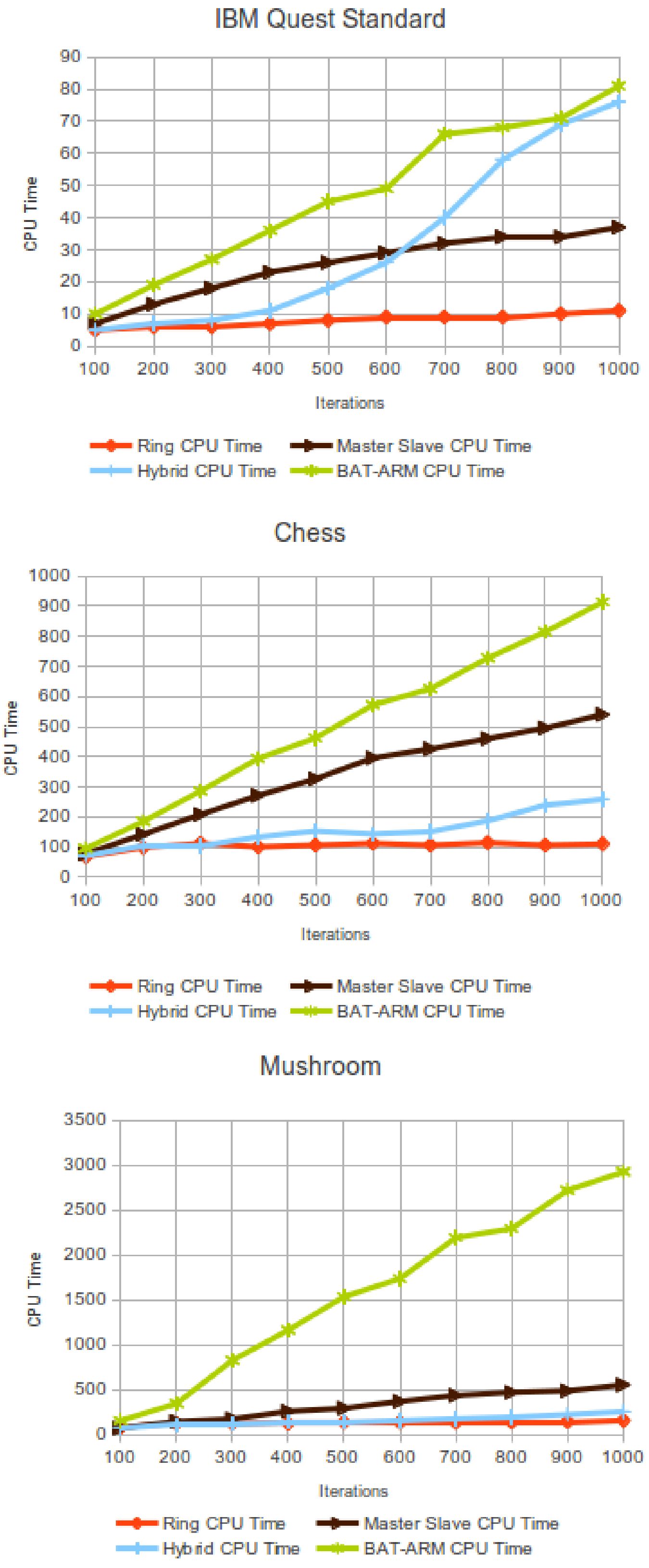


FIGURE 6.4: Evaluation of CPU Time in terms of the number of iterations.

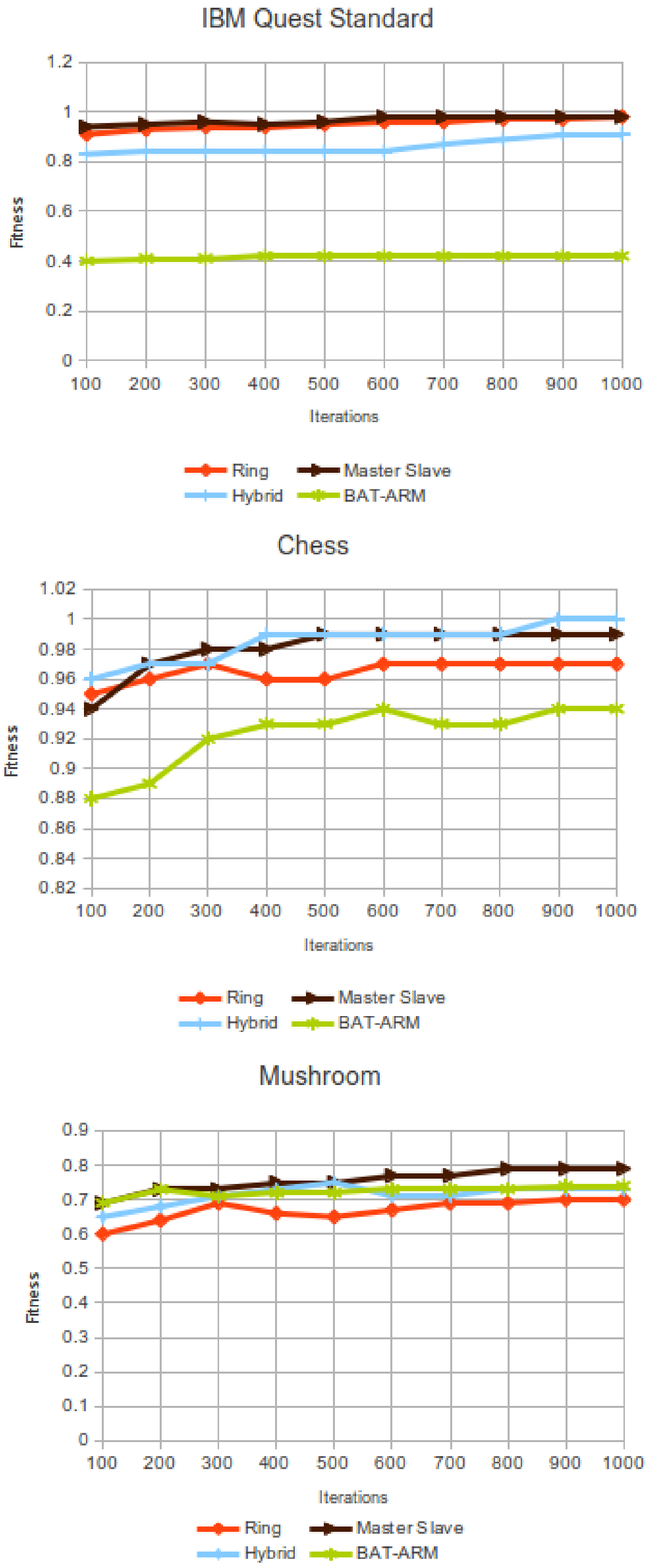


FIGURE 6.5: Evaluation of fitness in terms of the number of iterations.

DATASET	*MSB-ARM*			*HBSO*	*BSO*	ACO_R	*G3PARM*	*ARMBGSA*
	RING	*Master-Slave*	*Hybrid*					
Basketball	1.0	1.0	1.0	0.93	0.97	0.61	0.93	0.45
Quak	1.0	1.0	1.0	1.0	1.0	0.73	0.90	0.39
IBM-Quest-std	0.93	0.95	0.84	0.93	0.94	0.45	0.88	0.40
Chess	0.96	0.97	0.97	0.90	0.88	0.3	0.86	0.38
Mushroom	0.63	0.73	0.68	0.74	0.52	0.1	0.85	0.35
pumbs_star	0.53	0.50	0.59	0.72	0.40	Not tested	Not tested	Not tested
BMS-WebView-1	0.55	0.51	0.56	0.55	0.35	Not tested	Not tested	Not tested
Connect	0.53	0.55	0.64	0.50	0.25	Not tested	Not tested	Not tested

Table 6.3: Comparing our approach to existing approaches w.r.t fitness

DATASET	*MSB-ARM*			*HBSO*	*BSO*	*Apriori*	*FPGrowth*	*G3PARM*	*ARMBGSA*
	RING	*Master-Slave*	*Hybrid*						
pumbs_star	171	170	238	200	150	500	600	250	160
BMS-WebView-1	331	346	296	350	370	1000	800	380	250
Connect	1118	971	971	980	950	2600	2900	910	880

Table 6.4: Comparing our approach to existing approaches w.r.t CPU Time

Table 6.4 shows how the runtime varies w.r.t to different datasets. Again, results prove the efficiency of our approach against other methods including HBSO-TS, BSO-ARM and G3PARM. In addition, our strategies surpass the exact algorithms Apriori and FPGrowth in terms of CPU runtime. These interesting results are reached thanks to the fast successful search in BAT-ARM algorithm, and the parallelism applied between the sub-swarms in each cooperative strategy. Nevertheless, ARMBGSA outperforms our proposal with all its strategies in terms of CPU time. This can be explained by the time taken when swarms interchange the best solution informations among themselves. In addition, results in table 6.4 shows that hybrid strategy surpass the other strategies in time consuming with large datasets such as: *BMS-WebView-1* and *Connect*.

6.3.3.3 Comparative study to multi-objective optimization approaches

In multi-objective optimization algorithms several measures are introduced to extract the most useful and understandable rules for the user. These new measures lead the process to extract the rules with high overage support and confidence, unlike the single objective algorithms that extract a considerable number of rules. Based on this remark, we compared our proposal to three novel multi-objectives algorithms in terms of support and confidence overages of the extracted rules. The three multi-objective algorithms are:

- Multi-Objective Differential Evolution algorithm for mining Numeric Association Rules, i.e., MODENAR [9]
- Multi-Objective Genetic Algorithm for Association Rule mining, i.e., MOGAR [111]
- Multi-objective PSO algorithm for mining numerical association rules, i.e., MOPAR [14]

The results of this comparison are summarized in Tables 6.5 and 6.6.

DATASET	*MSB-ARM*			*MOPAR*	*MODENAR*	*MOGAR*
	RING	*Master-Slave*	*Hybrid*			
Basketball	26	33.3	34.4	30	37	50
Bodyfat	23.4	22.75	22.76	22	65	57
Quak	26.3	25.3	30.6	31	39	30

Table 6.5: Comparing our approach to multi-objective optimization approaches w.r.t overage support(%)

DATASET	*MSB-ARM*			*MOPAR*	*MODENAR*	*MOGAR*
	RING	*Master-Slave*	*Hybrid*			
Basketball	84.1	85.2	92.3	95	61	83
Bodyfat	72	78.3	88.4	81	62	85
Quak	83.3	82.6	84.5	89.32	63	82

Table 6.6: Comparing our approach to multi-objective optimization approaches w.r.t overage confidence(%)

Table.6.5 compares the outcomes obtained by MSB-ARM and its strategies with previous described methods that deal with multi-objective optimization of association rule mining in terms of overage support. The results show that the proposed method yields a competitive support of the extracted rules. Table.6.6 shows that our suggested strategies give acceptable results. Our results can be improved and give better confidence overage related to threshold that can be changed by the user according to his exigencies. These promising results can be argued by best balance between the global exploration and the local exploitation reached by the nominated cooperation strategies, in addition to the intelligent search process in BAT-ARM algorithm. Furthermore, we can note clearly the superiority of hybrid cooperative strategy on the two other strategies which can be explained by the merging advantages of the both other strategies(Ring and Master-Slave).

6.4 Conclusion

In this chapter, we have proposed a multi-swarm version of the bat algorithm with three cooperative strategies: Ring, Master-Slave and Hybrid. Extensive experiments were performed to assess the performance of developed methods. Results prove the effectiveness and efficiency of the multi-swarm-bat (MSB-ARM) algorithm that outperforms BAT-ARM. We have seen a decrease in execution time with all the cooperative strategies. As well, we have noticed the superiority of hybrid plan against the other plans with big datasets. These encouraging results are reached mainly thanks to the vertical representation of database and parallelism in sub-swarms generation process. Furthermore, MSB-ARM algorithm outperforms the similar bio-inspired methods in terms of extracted rules quality. Also, it gives an acceptable and competitive outcomes in-face-of multi-objective approaches for mining association rules in terms of overage support and confidence of rules.

However, MSB-ARM stay suffering from rule redundancy and the large number of generated rules. To overcome this drawback, we propose, in the next chapter, a multi-objective version of BAT-ARM which use several interesting measures to evaluate generated rules. That extracts less number and more useful rules for the final user.

6.5 Publications Associated with this Chapter

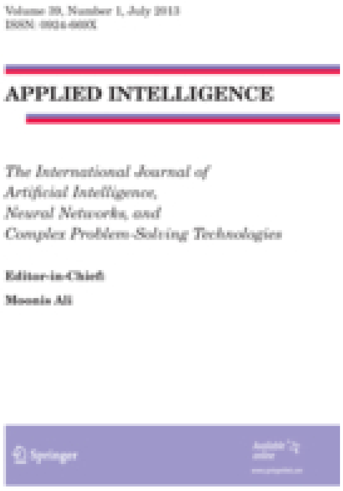
- **International Journal:**

  TITLE: Multi-swarm bat algorithm for association rule mining using multiple cooperative strategies

  AUTHORS: HERAGUEMI Kamel Eddine, KAMEL Nadjet, and DRIAS Habiba

  DOI: 10.1007/s10489-016-0806-y

  PUBLICATION DATE: **June 25, 2016**



APPLIED INTELLIGENCE

The International Journal of
Artificial Intelligence,
Neural Networks, and
Complex Problem-Solving Technologies

Editor-in-Chief:
Moonis Ali

INTERNATIONAL JOURNAL OF ARTIFICIAL INTELLIGENCE, NEURAL NETWORKS, AND COMPLEX PROBLEM-SOLVING TECHNOLOGIES (APPLIED INTELLIGENCE)

IMPACT FACTOR: **1.215**

H-INDEX: **43**

- **International Conference:**

  HERAGUEMI, Kamel Eddine, KAMEL, Nadjet, et DRIAS, Habiba. Multi-population Cooperative Bat Algorithm for Association Rule Mining. *In : Computational Collective Intelligence.* Springer International Publishing, 2015. p. 265-274.

Multi-objective bat algorithm for ARM

7.1 Introduction

Association rule mining problem attracts the attention of researchers inasmuch to its importance and applications in our world with the fast growth of the stored data. ARM process is computationally very expensive because rules number grows exponentially as items number in the database increases. However, ARM is more complex when we introduce the quality criteria and usefulness to the user. This chapter deals with association rule mining issues, we propose Multi-Objective Bat algorithm for association rules mining known as MOB-ARM. With the aim of extracting more useful and understandable rules, we introduce four quality measures of association rules: Support, Confidence, Comprehensibility, and Interestingness, in two objective functions considered for maximization. A series of experiments are carried out on several well-known benchmarks in ARM field and the performance of our proposal are evaluated and compared with those of other recently published methods including mono-objective and multi-objective approaches. The outcomes show a clear superiority of our proposal in-face-of mono objective methods in terms generated rules number and rule quality. Also, The analysis also shows a competitive outcomes in terms of quality against multi-objective optimization methods.

7.2 The MOB-ARM algorithm

7.2.1 Databases layout and rule representation

Association rules attributes can contain both categorical and quantitative values(e.g.: heat, age, time, sex ...). For the purpose of homogeneity and normalization, we use a discretization method to transform the continuous values to categorical ones, such that the values are grouped in intervals where each interval has its *id* value as presented in **Figure 7.1**.

Value	Interval	id
0.01	[0,0.1]	1
0.25	]0.2,0.3]	2
0.332	]0.3,0.4]	3
...	...	...
0.708	]0.7,0.8]	7

FIGURE 7.1: Discretization method for the continuous attributes

To count the different candidate rule measures, a database scan is needed. Since the datasets are very large, the scans need more time and resources. Hence, databases representation is important to facilitate the counting process. There are two layouts that algorithms usually used to represent transaction databases: horizontal and vertical layouts.

In the horizontal layout, each transaction T_i is presented as $T_i = (T_{id}, I)$ where the T_{id} is the transaction identification and I is set of items that occur in the transaction T_i. In the vertical layout, each item i_k in the item-set is represented as $i_k : \{i_k, t(i_k)\}$ where $t(i_k)$ is the transaction that contains i_k.

To count the support of an item-set X in the horizontal layout, we need to scan the whole database and count the transactions that contain X. With the database growth, this operation needs excessive time and resources. In our approach, we choose to use the vertical representation because the support of an item X is the size of its tidset. To count the support of item-set A, firstly, the tidset of A is the intersection of X and Y such that $X \cup Y =$ so $t(X) \cap t(Y) = t(A)$. This could be easily deduced from tidset definition. Then, the support of X is the size of its tidset.

7.2.2 Rule encoding

In the literature, there are two methods of representation for rules to mine ARMs with intelligent algorithms as discussed in[54] Pittsburgh and Michigan Approaches. In the first, each solution represents a set of rules. Whereas, in the second method, each solution encodes only one rule.

In our proposal we use the second approach where each solution X represents a rule that contains k items. Therefore, the solution X is represented with a vector S which contains *k+1* positions, where S[0] separates between rule antecedent and consequent. If the i^{th} item in the database is in the rule, then the position k contains i, else it takes 0, where $0 < k <= n + 1$. **Figure 7.2** illustrates the solution structure X (position of the bat) where n is items number in the database.

j	I_1	....	I_j	I_{j+1}	....	I_n

FIGURE 7.2: General solution representation

Example 7.1. *For example, let* $I = \{i_1, i_2, ..., i_{10}\}$ *be a set of items :*

- $X1 = \{3, 1, 5, 0, 6, 2, 0, 0, 7, 0, 0\}$ *represents the rule* $i_1, i_5 \Rightarrow i_6, i_2, i_7$*,*
- $X2 = \{5, 0, 1, 3, 6, 2, 0, 0, 0, 0, 7\}$ *represents the rule* $i_1, i_3, i_6, i_2 \Rightarrow i_7$*,*
- $X3 = \{2, 2, 0, 0, 0, 0, 3, 0, 0, 8, 0\}$ *represents the rule* $i_2 \Rightarrow i_3, i_8$*,*

7.2.3 Objective Functions

As mentioned above single objective evolutionary algorithms use generally only one measure i.e., *Support, Confidence, etc* to evaluate extracted rules quality. These measures assess rules depending on number of occurrence in database. Nevertheless, these algorithms do not give any importance to other rule quality measures like i.e., comprehensibility and interestingness...

In our work we use *comprehensibility, interestingness* measures in addition to confidence and support which are used as fitness function in [71], and as objectives in our method (MOB-ARM). The confidence criterion evaluates the quality of each rule based on occurrences number in the whole dataset. When the rule has more occurrences in the database, this means the rule has a better quality. We define the first objective for our method using the support and confidence, shown in equation 7.1.

$$Obj_1(R) = \alpha conf(R) + \beta supp(R)/\alpha + \beta \tag{7.1}$$

When the rule contains a huge number of attributes, this makes the rule more difficult to comprehend. If generated rules are not comprehensible for the user, they will be useless. This is why we introduce the comprehensibility measure, which can be modeled as shown in **Equa.7.2**.

$$Comprehensibility(X) = \frac{\log(1+|Conseq|)}{\log(1+|Antec \cup Conseq|)} \tag{7.2}$$

Where, $|Conseq|$ and $|Antec \cup Conseq|$ are items number in the Consequence part and the total rule respectively. The comprehensibility increases and the rules are more understandable whenever items number in the antecedent part was smaller. Moreover, 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 occurrence in the database. Interestingness measure is defined by **Equa.7.3**, Where A, C and N are the antecedence, consequence and transactions number in the whole database, respectively.

$$Interesting(X) = \frac{Supp(A \cup C)}{Supp(A)} \times \frac{Supp(A \cup C)}{Supp(C)} \times \frac{(1 - Supp(A \cup C)}{N} \tag{7.3}$$

We define the second objective for our algorithm based on *Comprehensibility, Interestingness* using Formula 7.4.

$$Obj_2(R) = \gamma Comp(R) + \delta Inter(R)/\gamma + \delta \tag{7.4}$$

Where α, β, γ and δ are empirical parameters which are chosen relative to the importance of support, confidence,Comprehensibility and Interestingness to final user,

7.2.4 The algorithm flow

Algorithm 15 illustrates the pseudo code of MOB-ARM. The main computational steps of the proposal are described as follows:

- ***Initialization step***: firstly, all the bats are initialized with random frequency and velocity. The values are taken in the intervals [0, items number] and [0, items number+1] respectively. A randomly generated position/solution (rule) is affected to each bat, and an initial rate and loudness is affected to each bat randomly.
- ***Search the non-dominate solution for the Pareto point*** : For each Pareto point, a new single global objective function is generated based on weights w_k in which their sum is equal to 1 ($\sum_{k=1}^{k} w_k = 1$). The global objective function is generally presented by: $\sum_{k=1}^{k} w_k f_k$, where k is objective functions number used for the mining problem. In our case, we have only two objective functions presented in **Equa.7.1, 7.4**. So the global function is defined as follows:

$$Obj(R) = w_1.Obj_1(R) + w_2.Obj_2(R); \tag{7.5}$$

Algorithm 15 MOB-ARM Algorithm pseudo code

```
objective functions $f_1(x), ..., f_k(x)$.
Initialize the bat population $x_i$ and $v_i$;
Initialize pulse rates $r_i$ and the loudness $A_i$;
for $j = 1$ to $N$ (points on Pareto fronts) do
    Generate $K$ weights $w_k \geq 0$ so that $\sum_{k=1}^{K} w_k = 1$;
    Form a single objective $\sum_{k=1}^{K} w_k f_k$;
    while ($t <$ Max number of iterations) do
        for all Bats in the population do
            Adjust frequency $f_i$ and update velocity $v_i$ using Eq.5.1,5.2;
            Generate a new solution $x_i$ ;
            if ($rand > r_i$) then
                Generate a local solution $x_i$ around the best solution $x_i*$ by changing just one item in the rule
            end if;
            if ($f(x_i) > f(x_i*)$) then
                Accept the new solution $x_i$;
                $x_i* = x_i$;
                Increase $r_i$ and reduce $A_i$ using Eq.4.5,4.6
            end if;
            Determine the global best solution $x*$ amoung $x_i*$;
        end for
    end while
    Record $x*$ as non-dominate solution;
end for
Post-process results and visualize the best detected rules.
```

- ***Search the best solution (Rule) for each bat at the Pareto point***: At each iteration, a new rule is generated based on **Algorithm 10** described in [71] by adjusting frequency, updating velocity and location. If the new fitness is better than the previous one, then the rule will be accepted, the loudness A_i reduced, and the rate r_i increased according to Equations **Equat 4.5**,**4.6** presented in [177].

7.3 Experimentation and results

To prove the effectiveness of our approach, we performed several experiments of the proposed algorithm MOB-ARM. This section describes the used benchmarks. After-that, comparative study with BAT-ARM[71] and MPB-ARM[72], which are two mono-objective versions of bat algorithm updated to association rules mining, is given. Also, we present comparison of our approach to three other multi-objective methods recently published. All algorithms are written in Java and executed on Intel core I5 machine with 4GB of memory running under Linux Ubuntu.

7.3.1 Benchmark and setup description

To test and compare our proposed algorithm MOB-ARM, we use six well known benchmarks. These datasets are frequently used in data mining from: frequent and mining dataset repository [62] and Bilkent University function approximation repository [65]. **Table 7.1** presents different datasets used in our experiments. The databases are classed into groups (*IBM Quest Standard, Chess and Mushroom*) and (*Basketball, Bodyfat and Quake*). The first group is used for the first study where results are compared to those of mono objective optimization methods, and the other group is used for the second study where outcomes are compared to multi-objective algorithms results.

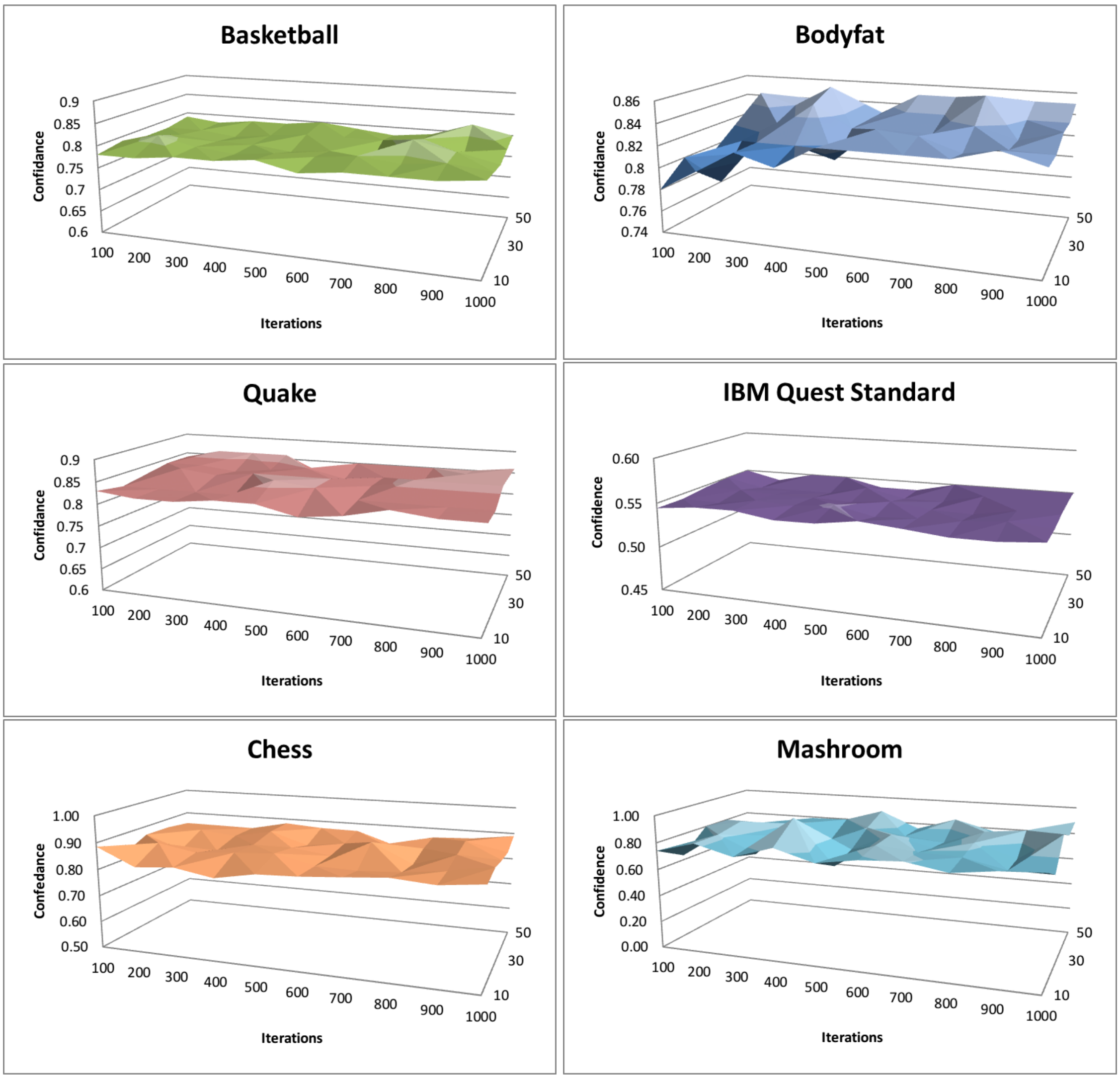
Dataset	Transactions size	Item size
Basketball	96	5
Bodyfat	252	15
Quake	2178	4
IBM Quest Standard	1000	20
Chess	3196	37
Mushroom	8124	23

Table 7.1: Description of experimental Benchmark

In our experiments, we tested our method with different range of parameters such as size of bats population n, iterations number t and Pareto points number p, by varying $n = 10, 25, 50, 100, 150$, $t = 100, 200, .., 1000$ and $p = 10, 30, 50$. The empirical parameters in the objectives functions of **Equa.7.1, 7.4** α ,β, γ and δ are fixed to $0.4, 0.3, 0.2, 0.1$ respectively. We found that the best results are achieved with $n = 50$ and $t = 200$, α and γ in **Equa. 4.5,4.6** defined as $\alpha = \gamma = 0.9$, as mentioned in [178].

7.3.2 Stability analysis

In this section, we evaluate the effectiveness of our approach (MOB-ARM) with several experiments on all the described datasets in the previous section. With the aim to determine the stability of MOB-ARM and to study the dealing of our proposal with measurements used in the objective functions with each one of the used benchmarks, we run the proposed algorithm 30 times, by fixing a minimum support and confidence thresholds to 0.2 and 0.5 respectively, for each test and the average results are considered to be analysed.



Basketball

Bodyfat

Quake

IBM Quest Standard

Chess

Mashroom

FIGURE 7.3: Evaluation of confidence level in terms of the number of iterations and Pareto points.

For more illustration, the obtained results are presented in three-dimensional graphs. **Figures.7.3-7.5** show how different measures in objective functions vary in terms of generation number and Pareto points used with the several datasets. We can note the stability of confidence results achieved starting from the first run with 100 iterations and 10 Pareto Points for all the datasets except *Bodyfat* dataset which stabilizes starting from 100 generation with 50 Pareto Points. In the case of comprehensibility, again, the algorithm achieves a state of stability from the first test for all benchmarks. In terms of interestingness, generally, the stabilization can be observed, although it is fairly low in case of basketball, Bodyfat and Quake datasets. These interesting results are achieved mainly due to echolocation system of bat algorithm and the powerful detection method for the Pareto Optimal in multi-objective bat algorithm.

7.3.3 Comparative study to single objective approaches

In this section, we propose a study that compares our new approach to single objective versions of bat algorithm designed for mining association rules (BAT-ARM, MPB-ARM). This experiment was carried out on three datasets with medium transactions size (*IBM Quest Standard, Chess and Mushroom*). The default parameters of the BAT-ARM[71], MPB-ARM[72] and MOB-ARM are defined to make the comparison completely fair. **table.7.2** presents the average results of thirty executions on three algorithms (BAT-ARM, MPB-ARM and MOB-ARM). In our comparison, three axes are taken into account: average support, confidence and number of generated rules. Outcomes show that the proposed algorithms extract less number of rules for all the datasets. This is because new criteria of selection are introduced as objectives (Comprehensibility and interestingness), so MOB-ARM generates only useful and understandable rules for the user. On contrary, mono-objective approaches generate the maximum number of rules that satisfy support and confidence thresholds. In terms of support and confidence we note that MOB-ARM is more robust than BAT-ARM and MPB-ARM because of the small number of extracted rules and dominance conditions applied when mining association rules.

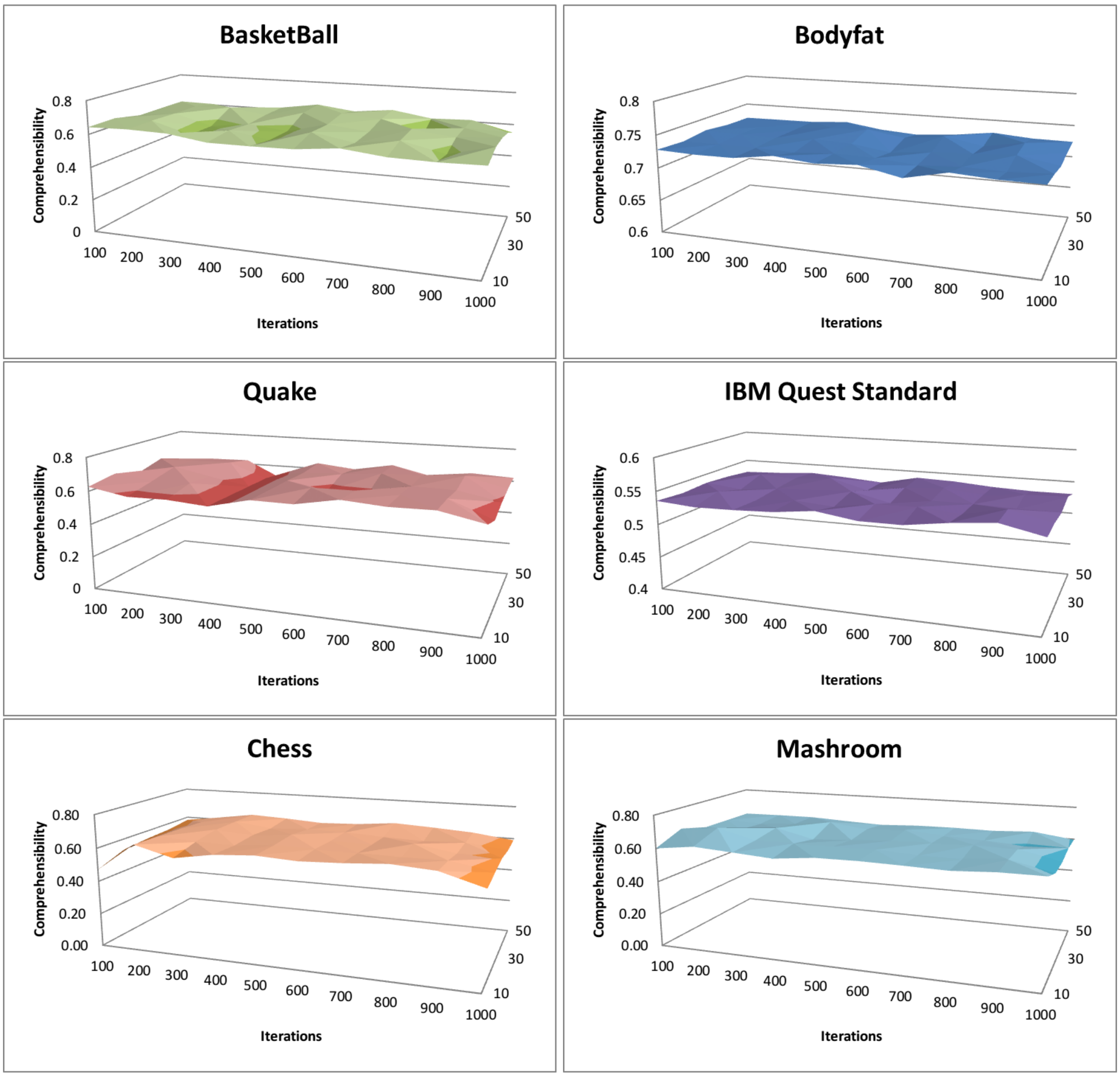


FIGURE 7.4: Evaluation of comprehensibility in terms of the number of iterations and Pareto points.

7.3.4 Comparative study to multi-objective approaches

In this study, effectiveness of the proposed algorithm is compared with three similar algorithms. All of these methods are based on a multi-objective evolutionary approach and designed for association rule mining. The three algorithms are:

1. Multi-Objective Differential Evolution algorithm for mining Numeric Association Rules, i.e., MODENAR [9]
2. Multi-Objective Genetic Algorithm for Association Rule mining, i.e., MOGAR [111]

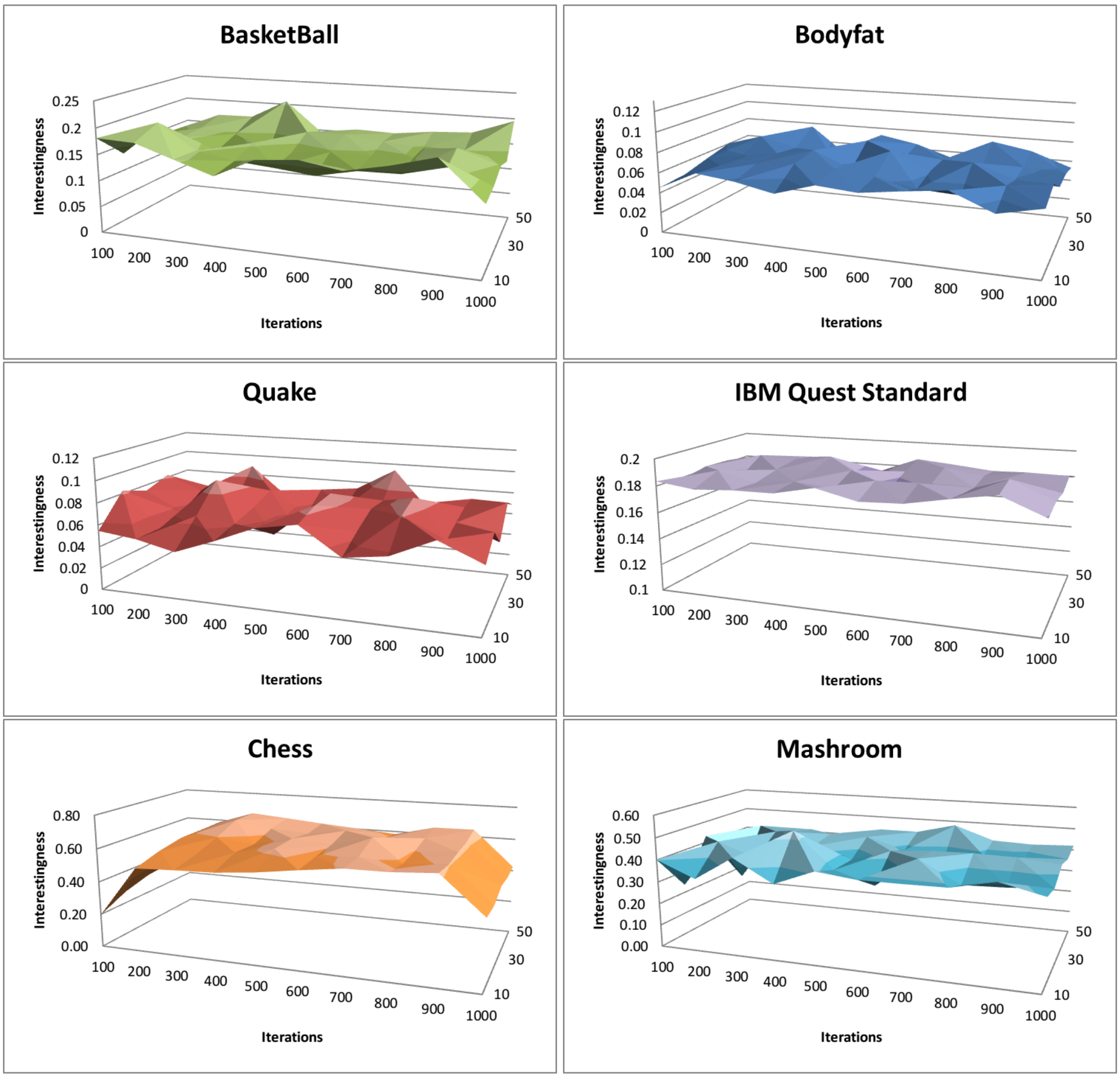


FIGURE 7.5: Evaluation of interestingness in terms of the number of iterations and Pareto points.

3. Multi-objective PSO algorithm for mining numerical association rules, i.e., MOPAR [14]

Table.7.3 compares the outcomes obtained by MOB-ARM to previous similar methods that deal with association rule mining as multi-objective optimization problem in terms of average support. The results show that our proposed method yields a competitive support of the extracted rule. However, in case of Bodyfat datasets the overage support is less than other methods (MODENAR, MOGAR), this is caused by the fact of the strict application of dominance conditions.

	Algorithms	Datasets		
		IBM-standard	Chess	Mashroom
No. of rules	MOB-ARM	215	293	26
	BAT-ARM	485	1870	341
	MPB-ARM	850	739	791
Support(%)	MOB-ARM	26	51	34
	BAT-ARM	25	38	23
	MPB-ARM	23	46	23
Confidance(%)	MOB-ARM	54	83	87
	BAT-ARM	52	72	54
	MPB-ARM	59	79	78.5

Table 7.2: Comparison of results to mono-objective methods in terms of Number of generated rules, Support and confidence

In addition, we compute the average confidence to evaluate the strength of extracted rules. **Table.7.4** shows that our suggested method gives acceptable results. Our results can be improved and give better confidence average because we use a minimum threshold that can be changed by the user according to his exigencies as shown in **Figure 7.6**. To make the study more comprehensive, we

DataBase	Support(%)			
	MOPAR	MODENAR	MOGAR	MOB-ARM
Basketball	30.76	37.20	50.85	37.5
Bodyfat	22.95	65.22	57.22	23
Quake	31.97	39.86	30.12	41

Table 7.3: Comparison of results to similar work in terms of Support

DataBase	Confidence (%)			
	MOPAR	MODENAR	MOGAR	MOB-ARM
Basketball	95 ± 0.12	61 ± 2.1	83	79
Bodyfat	81.8 ± 0.27	62 ± 3.2	85	83
Quake	89.32 ± 0.18	63 ± 2.8	82	88

Table 7.4: Comparison of results to similar work in terms of Confidence

DataBase	Number of rules			
	MOPAR	MODENAR	MOGAR	MOB-ARM
Basketball	69.75	48.0	50	63
Bodyfat	70.8	52.4	84	51
Quake	54.1	55.4	44.87	50

Table 7.5: Comparison of results to similar work in terms of Number of generated rules

calculate the average number of extracted rules for each dataset and the results are presented in **Table.7.5**. From the outcomes, we observe that our method have a stable behavior and it is competitive to the previous methods.

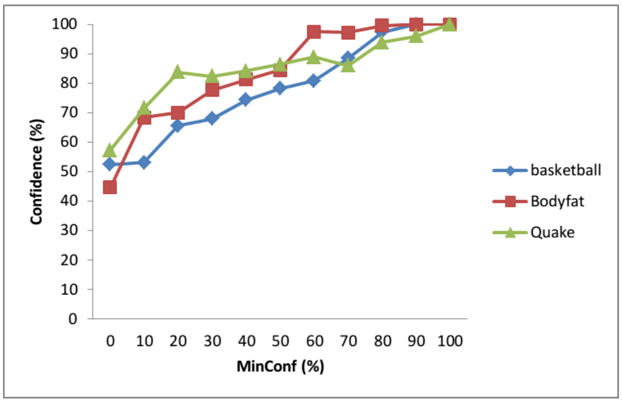


Figure 7.6: Evolution of average confidence in face of minimum confidence

Therefore, as it has been shown above in the results, MOB-ARM generates association rules with competitive measures compared to other methods in terms of support, confidence and number of generated rules. As we include the comprehensibility measure, all the extracted rules are understandable to the user and interestingness measure leads the process to extract the useful rules in which the users have interest. These rules are rare and difficult to detect in large datasets.

7.4 Conclusion

In this chapter, we proposed a new multi-objective meta-heuristic to deal with association rule mining based on bat algorithm and called MOB-ARM. The proposal uses four quality measures, (Support, Confidence, Comprehensibility and Interestingness) to extract the best rules that help the user in decision making process and can be understood. Our approach is based on vertical dataset representation that reduces the computation time of computing and avoids the repeated scans of the whole datasets in each rule evaluation. The performance of MOB-ARM has been compared to two single objective algorithms based on bat algorithm for mining association rules and three other methods dealing with multi-objective miners. The experimental results prove the effectiveness of our proposed method.

7.5 Publications Associated with this Chapter

- **International Conference:**

  HERAGUEMI, Kamel Eddine, KAMEL, Nadjet, et DRIAS, Habiba. Multi-objective bat algorithm for mining interesting association rules. *Accepted in: the Fourth International Conference on Mining Intelligence and Knowledge Exploration (MIKE 2016).* Springer International Publishing.

Conclusion and future work

In this thesis, we treated the issue of association rule mining which is an important task in data mining. Firstly, we reviewed the association rule mining and its existing algorithms, including exhaustive search algorithms developed in this field. Also, we presented a deep literature review on computational intelligence and detailed its paradigms, that showed the efficiency of this discipline to solve many problems specially optimization issues. Furthermore, we described the actual state of association rule mining domain, including several approaches proposed over the history to mine rule such as: evolutionary algorithms and swarm intelligence algorithms and multi-objective Proposals for ARM.

Secondly, we proposed three advanced proposals, based on bat algorithm which is a new swarm based algorithm, to mine association rules. They aime to discover the best set of interesting association rules from data, Bat algorithm for mining association rule mining, Multi-swarm Cooperative Bat Algorithm for ARM and multi-objective bat-algorithm for ARM, known as: ***BAT-ARM, MSB-ARM and MOB-ARM***, respectively. The obtained results are highly promising and show that our proposals is appropriate technique to undertake the ARM task.

In our first approach BAT-ARM, we defined a new description for bat movement related to association rule mining, where the main concepts in original bat algorithm were redefined: frequency and velocity. As well, we proposed an algorithm as new solution which never generate unacceptable rule due to the defined objective function. In face-of FP-growth algorithm, our proposed algorithm proved its efficiency in term of time, memory usage and quality of generated rules with maximizing the fitness function. However, BAT-ARM sufferes from the less communication between individuals in the population which give a large redundancy in extracted rule set. Therefore, to overcome BAT-ARM drawback and to maintain the best trade-off among exploration and exploitation to extract best rules, we proposed multi-swarm cooperative Bat Algorithm for ARM (MSB-ARM) with three collaboration topologies: Ring, Master-Slave and Hybrid. Results prove the effectiveness and efficiency of the multi-swarm bat (MSB-ARM) algorithm against BAT-ARM and other methods in the literature (BSO-ARM, HBSO-TS, ACO_R, G3PARM and ARMBGSA). Also, it give a competitive outcome in-face-of multi-objective approaches (MODENAR, MOGAR and MOPAR) in terms of overage rules quality. Nevertheless, MSB-ARM stays suffering from large number of generated rules. To overcome this drawback, we proposed multi-objective version of BAT-ARM which uses several interesting measures to find the most interesting and useful rules highly relevant to the user's application. This proposal uses four quality measures, (Support, Confidence, Comprehensibility and Interestingness), as objectives to maximize. The experimental results prove the effectiveness of MOB-ARM against other approaches, including MODENAR, MOGAR and MOPAR in terms of computation time and extracted rules quality.

8.0.1 Future Work and Perspectives

Although we proposed three efficient approaches to mine interesting association rules from a large-scale database, there are other significant topics that we plan to carry out and study further in the research and applications of association rules. We summarize them as follows:

- Improve the proposed approaches by making it capable to treat large databases, especially *WebDocs*, which exceeds a million and half of transactions.
- Apply our algorithms to big data discipline, such as social networks (Facebook, Twitter and LinkedIn, Etc.).
- Implement the algorithms on a Graphics Processing Unit(GPU) to improve both the solution quality and its running time.
- Update the proposed approaches to be used for multi-relational databases that have more complex structure and store more information than raw datasets where existing ARM algorithms cannot be applied directly.

Bibliography

[1] Aboubi, Y., Drias, H., and Kamel, N. Bat-clara: Bat-inspired algorithm for clustering large applications. *IFAC-PapersOnLine 49*, 12 (2016), 243–248.

[2] 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.

[3] Agrawal, R., Imieliński, 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.

[4] Agrawal, R., and Shafer, J. C. Parallel mining of association rules. *IEEE Transactions on knowledge and Data Engineering 8*, 6 (1996), 962–969.

[5] Agrawal, R., Srikant, R., et al. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB* (1994), vol. 1215, pp. 487–499.

[6] Ahmed Majid Taha, S.-D. C., and Mustapha, A. Multi-swarm bat algorithm. In *Research Journal of Applied Sciences, Engineering and Technology* (2015), pp. 1389–1395.

[7] Akhtar, S., Ahmad, A., and Abdel-Rahman, E. A metaheuristic bat-inspired algorithm for full body human pose estimation. In *Computer and Robot Vision (CRV), 2012 Ninth Conference on* (2012), IEEE, pp. 369–375.

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

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

[10] Ali, E. Optimization of power system stabilizers using bat search algorithm. *International Journal of Electrical Power & Energy Systems 61* (2014), 683–690.

[11] Amine, L. M., and Nadjet, K. A multi-objective binary bat algorithm. In *Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication* (2015), ACM, p. 75.

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

[13] 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.

[14] Beiranvand, V., Mobasher-Kashani, M., and Bakar, A. A. Multi-objective {PSO} algorithm for mining numerical association rules without a priori discretization. *Expert Systems with Applications 41*, 9 (2014), 4259 – 4273.

[15] Bell, J. E., and McMullen, P. R. Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics 18*, 1 (2004), 41–48.

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

[17] Bezdek, J. What is computational intelligence? *Computational Intelligence Imitating life* (Dec 1994), 1–12.

[18] Biswal, S., Barisal, A., Behera, A., and Prakash, T. Optimal power dispatch using bat algorithm. In *Energy Efficient Technologies for Sustainability (ICEETS), 2013 International Conference on* (2013), IEEE, pp. 1018–1023.

[19] Biswas, S., Das, S., Debchoudhury, S., and Kundu, S. Co-evolving bee colonies by forager migration: A multi-swarm based artificial bee colony

algorithm for global search space. *Applied Mathematics and Computation 232* (2014), 216–234.

[20] Boettcher, M., Russ, G., Nauck, D., and Kruse, R. From change mining to relevance feedbacka unified view on assessing rule interestingness. *Post-Mining of association rules: Techniques for effective knowledge extraction*, 12–37.

[21] Bonabeau, E., Dorigo, M., and Theraulaz, G. *Swarm intelligence: from natural to artificial systems.* No. 1. Oxford university press, 1999.

[22] Bose, I., and Mahapatra, R. K. Business data mining—a machine learning perspective. *Information & management 39*, 3 (2001), 211–225.

[23] Brin, S., Motwani, R., and Silverstein, C. Beyond market baskets: Generalizing association rules to correlations. In *ACM SIGMOD Record* (1997), vol. 26, ACM, pp. 265–276.

[24] Brin, S., Motwani, R., Ullman, J. D., and Tsur, S. Dynamic itemset counting and implication rules for market basket data. In *ACM SIGMOD Record* (1997), vol. 26, ACM, pp. 255–264.

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

[26] Bullnheimer, B., Hartl, R. F., and Strauss, C. A new rank based version of the ant system. a computational study.

[27] Christian, B., and Daniel, M. Swarm intelligence. introduction and applications. *Natural Computing Series, Springer* (2008).

[28] Coello, C. A. C., Van Veldhuizen, D. A., and Lamont, G. B. *Evolutionary algorithms for solving multi-objective problems*, vol. 242. Springer, 2002.

[29] Colorni, A., Dorigo, M., Maniezzo, V., et al. Distributed optimization by ant colonies. In *Proceedings of the first European conference on artificial life* (1991), vol. 142, Paris, France, pp. 134–142.

[30] Dehuri, S., Jagadev, A., Ghosh, A., and Mall, R. Multi-objective genetic algorithm for association rule mining using a homogeneous dedicated cluster of workstations. *American Journal of Applied Sciences 3*, 11 (2006).

[31] Del Valle, Y., Venayagamoorthy, G. K., Mohagheghi, S., Hernandez, J.-C., and Harley, R. G. Particle swarm optimization: basic concepts, variants and applications in power systems. *IEEE Transactions on evolutionary computation 12*, 2 (2008), 171–195.

[32] Dhivya, M., Sundarambal, M., and Vincent, J. O. Energy efficient cluster formation in wireless sensor networks using cuckoo search. In *International Conference on Swarm, Evolutionary, and Memetic Computing* (2011), Springer, pp. 140–147.

[33] Diwold, K., Beekman, M., and Middendorf, M. Honeybee optimisation–an overview and a new bee inspired optimisation scheme. In *Handbook of swarm intelligence*. Springer, 2011, pp. 295–327.

[34] Djenouri, Y., Bendjoudi, A., Habbas, Z., Mehdi, M., and Djenouri, D. Reducing thread divergence in gpu-based bees swarm optimization applied to association rule mining. *Concurrency and Computation: Practice and Experience* (2016), n/a–n/a. cpe.3836.

[35] Djenouri, Y., Drias, H., and Chemchem, A. A hybrid bees swarm optimization and tabu search algorithm for association rule mining. In *Nature and Biologically Inspired Computing (NaBIC), 2013 World Congress on* (2013), IEEE, pp. 120–125.

[36] Djenouri, Y., Drias, H., and Habbas, Z. Bees swarm optimisation using multiple strategies for association rule mining. *International Journal of Bio-Inspired Computation 6*, 4 (2014), 239–249.

[37] Djenouri, Y., Drias, H., and Habbas, Z. Hybrid intelligent method for association rules mining using multiple strategies. *International Journal of Applied Metaheuristic Computing (IJAMC) 5*, 1 (2014), 46–64.

[38] 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.

[39] Dorigo, M., and Gambardella, L. M. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on evolutionary computation 1*, 1 (1997), 53–66.

[40] Dorigo, M., Maniezzo, V., and Colorni, A. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 26*, 1 (1996), 29–41.

[41] Drias, H. Genetic algorithm versus memetic algorithm for association rules mining. In *Nature and Biologically Inspired Computing (NaBIC), 2014 Sixth World Congress on* (2014), IEEE, pp. 208–213.

[42] Drias, H., Sadeg, S., and Yahi, S. Cooperative bees swarm for solving the maximum weighted satisfiability problem. In *International Work-Conference on Artificial Neural Networks* (2005), Springer, pp. 318–325.

[43] 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.

[44] Eberhart, R. C., and Shi, Y. *Computational intelligence: concepts to implementations*. Elsevier, 2011.

[45] Farahani, S. M., Abshouri, A. A., Nasiri, B., and Meybodi, M. Some hybrid models to improve firefly algorithm performance. *International Journal of Artificial Intelligence 8*, 12 (2012), 97–117.

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

[47] Fister, I., Rauter, S., Yang, X.-S., and Ljubič, K. Planning the sports training sessions with the bat algorithm. *Neurocomputing 149* (2015), 993–1002.

[48] Fister Jr, I., Fister, D., and Fister, I. A comprehensive review of cuckoo search: variants and hybrids. *International Journal of Mathematical Modelling and Numerical Optimisation 4*, 4 (2013), 387–409.

[49] Fister Jr, I., Fister, D., and Yang, X.-S. A hybrid bat algorithm. *arXiv preprint arXiv:1303.6310* (2013).

[50] Fister Jr, I., Fister, I., and Yang, X.-S. Towards the development of a parameter-free bat algorithm. In *StuCoSReC: Proceedings of the 2015 2nd Student Computer Science Research Conference* (2015), pp. 31–34.

[51] Fogel, D. B. *Artificial intelligence through simulated evolution*. Wiley-IEEE Press, 2009.

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

[53] Frawley, W. J., Piatetsky-Shapiro, G., and Matheus, C. J. Knowledge discovery in databases: An overview. *AI magazine 13*, 3 (1992), 57.

[54] Freitas, A. A. A survey of evolutionary algorithms for data mining and knowledge discovery. In *Advances in evolutionary computing*. Springer, 2003, pp. 819–845.

[55] Gada, F. V., Yadav, K. R., and Shah, R. D. Data mining in medical application for better detection of disease. *IJCER 2*, 2 (2013), 173–176.

[56] Gandomi, A. H., Yang, X.-S., Alavi, A. H., and Talatahari, S. Bat algorithm for constrained optimization tasks. *Neural Computing and Applications 22*, 6 (2013), 1239–1255.

[57] Ganghishetti, P., and Vadlamani, R. Association rule mining via evolutionary multi-objective optimization. In *Multi-disciplinary Trends in Artificial Intelligence*. Springer, 2014, pp. 35–46.

[58] Geetharamani, R., Revathy, P., and Jacob, S. G. Prediction of users webpage access behaviour using association rule mining. *Sadhana 40*, 8 (2015), 2353–2365.

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

[60] Gheraibia, Y., Moussaoui, A., Djenouri, Y., Kabir, S., and Yin, P. Y. Penguins search optimisation algorithm for association rules mining. *CIT. Journal of Computing and Information Technology 24*, 2 (2016), 165–179.

[61] Goel, S., Goel, N., and Gupta, D. Unconstrained optimisation through bat algorithm. *International Journal of Intelligent Engineering Informatics 2*, 4 (2014), 259–270.

[62] Goethls, B., and Zaki, M. J. Frequent itemset mining dataset repository. http://fimi.ua.ac.be/data/.

[63] Gupta, M., and Sikka, G. Association rules extraction using multi-objective feature of genetic algorithm. In *Proceedings of the World Congress on Engineering and Computer Science* (2013), vol. 2.

[64] Gupta, N., Mangal, N., Tiwari, K., and Mitra, P. Mining quantitative association rules in protein sequences. In *Data Mining* (2006), Springer, pp. 273–281.

[65] Guvenir, H. A., and Uysal, I. Bilkent university function approximation repository. http://funapp.cs.bilkent.edu.tr/DataSets/.

[66] Haldulakar, R., and Agrawal, J. Optimization of association rule mining through genetic algorithm. *International Journal on Computer Science & Engineering 3*, 3 (2011).

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

[68] Han, J., Pei, J., and Yin, Y. Mining frequent patterns without candidate generation. In *ACM SIGMOD Record* (2000), vol. 29, ACM, pp. 1–12.

[69] Hansen, N., Arnold, D. V., and Auger, A. Evolution strategies. In *Springer Handbook of Computational Intelligence*. Springer, 2015, pp. 871–898.

[70] Hassanzadeh, T., and Kanan, H. R. Fuzzy fa: a modified firefly algorithm. *Applied Artificial Intelligence 28*, 1 (2014), 47–65.

[71] Heraguemi, K. E., Kamel, N., and Drias, H. Association rule mining based on bat algorithm. *Journal of Computational and Theoretical Nanoscience 12*, 7 (2015), 1195–1200.

[72] Heraguemi, K. E., Kamel, N., and Drias, H. Multi-population cooperative bat algorithm for association rule mining. In *Computational Collective Intelligence*. Springer, 2015, pp. 265–274.

[73] Heraguemi, K. E., Kamel, N., and Drias, H. Multi-swarm bat algorithm for association rule mining using multiple cooperative strategies. *Applied Intelligence* (2016), 1–13.

[74] Holland, J. H. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* U Michigan Press, 1975.

[75] 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.

[76] Indira, K., and Kanmani, S. Performance analysis of genetic algorithm for mining association rules.

[77] Indira, K., and Kanmani, S. Mining association rules using hybrid genetic algorithm and particle swarm optimisation algorithm. *International Journal of Data Analysis Techniques and Strategies 7*, 1 (2015), 59–76.

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

[79] Jha, R. Association rules mining for business intelligence. *International Journal of Scientific and Research Publications 4*, 5 (2014).

[80] Kabir, M. M. J., Xu, S., Kang, B. H., and Zhao, Z. Discovery of interesting association rules using genetic algorithm with adaptive mutation. In *International Conference on Neural Information Processing* (2015), Springer, pp. 96–105.

[81] Kabir, M. M. J., Xu, S., Kang, B. H., and Zhao, Z. A new evolutionary algorithm for extracting a reduced set of interesting association rules. In *Neural Information Processing* (2015), Springer, pp. 133–142.

[82] Kacprzyk, J., and Zadrożny, S. Derivation of linguistic summaries is inherently difficult: can association rule mining help? In *Towards advanced data analysis by combining soft computing and statistics*. Springer, 2013, pp. 291–303.

[83] Kahvazadeh, I., and Abadeh, M. S. Mocanar: Amulti-objective cuckoo search algorithm for numeric association rule discovery.

[84] Karaboga, D., and Akay, B. A survey: algorithms simulating bee swarm intelligence. *Artificial Intelligence Review 31*, 1-4 (2009), 61–85.

[85] Khademolghorani, F., Baraani, A., and Zamanifar, K. Efficient mining of association rules based on gravitational search algorithm. *IJCSI* (2011).

[86] Khan, K., Nikov, A., and Sahai, A. A fuzzy bat clustering method for ergonomic screening of office workplaces. In *Third International Conference on Software, Services and Semantic Technologies S3T 2011* (2011), Springer, pp. 59–66.

[87] Khan, K., and Sahai, A. A comparison of ba, ga, pso, bp and lm for training feed forward neural networks in e-learning context. *International Journal of Intelligent Systems and Applications 4*, 7 (2012), 23.

[88] KHAN, K., SAHAI, A., AND CAMPUS, A. A fuzzy c-means bi-sonar-based metaheuristic optomization algorithm. *IJIMAI 1*, 7 (2012), 26–32.

[89] KOMARASAMY, G., AND WAHI, A. An optimized k-means clustering technique using bat algorithm. *European Journal of Scientific Research 84*, 2 (2012), 26–273.

[90] KOZA, J. R. *Genetic programming: on the programming of computers by means of natural selection*, vol. 1. MIT press, 1992.

[91] 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.

[92] LAAMARI, M. A., AND KAMEL, N. A hybrid bat based feature selection approach for intrusion detection. In *Bio-Inspired Computing-Theories and Applications*. Springer, 2014, pp. 230–238.

[93] LAVRAČ, N., FLACH, P., AND ZUPAN, B. Rule evaluation measures: A unifying view. In *International Conference on Inductive Logic Programming* (1999), Springer, pp. 174–185.

[94] LAYEB, A. A novel quantum inspired cuckoo search for knapsack problems. *International Journal of Bio-Inspired Computation 3*, 5 (2011), 297–305.

[95] LEUNG, K.-S., WONG, K.-C., CHAN, T.-M., WONG, M.-H., LEE, K.-H., LAU, C.-K., AND TSUI, S. K. Discovering protein–dna binding sequence patterns using association rule mining. *Nucleic acids research 38*, 19 (2010), 6324–6337.

[96] LI, C., NGUYEN, T. T., YANG, M., YANG, S., AND ZENG, S. Multi-population methods in unconstrained continuous dynamic environments: The challenges. *Information Sciences 296* (2015), 95–118.

[97] LI, C., YANG, S., AND YANG, M. An adaptive multi-swarm optimizer for dynamic optimization problems. *Evolutionary computation 22*, 4 (2014), 559–594.

[98] LI, J., SHEN, H., AND TOPOR, R. Mining the optimal class association rule set. *Knowledge-Based Systems 15*, 7 (2002), 399–405.

[99] LIN, J.-H., CHOU, C.-W., YANG, C.-H., TSAI, H.-L., ET AL. A chaotic levy flight bat algorithm for parameter estimation in nonlinear dynamic biological systems. *source: Journal of Computer and Information Technology 2*, 2 (2012), 56–63.

[100] Lin, W., Alvarez, S. A., and Ruiz, C. Efficient adaptive-support association rule mining for recommender systems. *Data mining and knowledge discovery 6*, 1 (2002), 83–105.

[101] Liu, D. Improved genetic algorithm based on simulated annealing and quantum computing strategy for mining association rules. *Journal of Software 5*, 11 (2010), 1243–1249.

[102] Luo, X., and Wang, W. Improved algorithms research for association rule based on matrix. *Intelligent Computing and Cognitive Informatics, International Conference on 0* (2010), 415–419.

[103] Marichelvam, M., and Prabaharam, T. A bat algorithm for realistic hybrid flowshop scheduling problems to minimize makespan and mean flow time. *ICTACT Journal on Soft Computing 3*, 1 (2012), 428–433.

[104] Martens, D., De Backer, M., Haesen, R., Vanthienen, J., Snoeck, M., and Baesens, B. Classification with ant colony optimization. *IEEE Transactions on Evolutionary Computation 11*, 5 (2007), 651–665.

[105] Martín, D., Alcalá-Fdez, J., Rosete, A., and Herrera, F. Nicgar: A niching genetic algorithm to mine a diverse set of interesting quantitative association rules. *Information Sciences 355* (2016), 208–228.

[106] Martínez-Ballesteros, M., Bacardit, J., Troncoso, A., and Riquelme, J. C. Enhancing the scalability of a genetic algorithm to discover quantitative association rules in large-scale datasets. *Integrated Computer-Aided Engineering 22*, 1 (2015), 21–39.

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

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

[109] 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.

[110] Millonas, M. M. Swarms, phase transitions, and collective intelligence. In *SANTA FE INSTITUTE STUDIES IN THE SCIENCES OF COMPLEXITY-PROCEEDINGS VOLUME-* (1994), vol. 17, ADDISON-WESLEY PUBLISHING CO, pp. 417–417.

[111] Minaei-Bidgoli, B., Barmaki, R., and Nasiri, M. Mining numerical association rules via multi-objective genetic algorithms. *Information Sciences 233* (2013), 15–24.

[112] Mirjalili, S., Mirjalili, S. M., and Yang, X.-S. Binary bat algorithm. *Neural Computing and Applications 25*, 3-4 (2014), 663–681.

[113] Mishra, S., Shaw, K., and Mishra, D. A new meta-heuristic bat inspired classification approach for microarray data. *Procedia Technology 4* (2012), 802–806.

[114] Mobasher, B., Dai, H., Luo, T., and Nakagawa, M. Effective personalization based on association rule discovery from web usage data. In *Proceedings of the 3rd international workshop on Web information and data management* (2001), ACM, pp. 9–15.

[115] Mohamad, B. Al-maqaleh," discovering interesting association rules: A multi-objective genetic algorithm approach. *International Journal of Applied Information Systems 5*, 3 (2013), 47–52.

[116] Mohamad, M. M., Dunnigan, M. W., and Taylor, N. K. Ant colony robot motion planning. In *EUROCON 2005-The International Conference on" Computer as a Tool"* (2005), vol. 1, IEEE, pp. 213–216.

[117] Musikapun, P., and Pongcharoen, P. Solving multi-stage multi-machine multi-product scheduling problem using bat algorithm. In *2nd international conference on management and artificial intelligence* (2012), vol. 35, IACSIT Press Singapore, pp. 98–102.

[118] Nahar, J., Imam, T., Tickle, K. S., and Chen, Y.-P. P. Association rule mining to detect factors which contribute to heart disease in males and females. *Expert Systems with Applications 40*, 4 (2013), 1086–1093.

[119] Nakamura, R. Y., Pereira, L. A., Costa, K., Rodrigues, D., Papa, J. P., and Yang, X.-S. Bba: a binary bat algorithm for feature selection. In *2012 25th SIBGRAPI Conference on Graphics, Patterns and Images* (2012), IEEE, pp. 291–297.

[120] Natarajan, A., Subramanian, S., and Premalatha, K. A comparative study of cuckoo search and bat algorithm for bloom filter optimisation in spam filtering. *International Journal of Bio-Inspired Computation 4*, 2 (2012), 89–99.

[121] Olmo, J., Luna, J., Romero, J., and Ventura, S. Association rule mining using a multi-objective grammar-based ant programming algorithm. In *Intelligent Systems Design and Applications (ISDA), 2011 11th International Conference on* (Nov 2011), pp. 971–977.

[122] Ordonez, C. Comparing association rules and decision trees for disease prediction. In *Proceedings of the international workshop on Healthcare information and knowledge management* (2006), ACM, pp. 17–24.

[123] Ouaarab, A., Ahiod, B., and Yang, X.-S. Discrete cuckoo search algorithm for the travelling salesman problem. *Neural Computing and Applications 24*, 7-8 (2014), 1659–1669.

[124] Parisa, M., Behrouz, M., Mahdi, N., and Afshin, S. Multi-objective numeric association rules mining via ant colony optimization for continuous domains without specifying minimum support and minimum confidence. *International Journal of Computer Science Issues 8*, 1 (2011).

[125] Park, J. S., Chen, M.-S., and Yu, P. S. *An effective hash-based algorithm for mining association rules*, vol. 24. ACM, 1995.

[126] Park, J. S., Chen, M.-S., and Yu, P. S. Efficient parallel data mining for association rules. In *Proceedings of the fourth international conference on Information and knowledge management* (1995), ACM, pp. 31–36.

[127] Pedrycz, W. Fuzzy sets engineering. *CRC Press* (1996), 73 – 106.

[128] Poonam Sehrawat, M., and Rohil, H. Association rule mining using firefly algorithm. *International Journal of Latest Trends in Engineeri ng and Technolog 3*, 2 (2013), 46–64.

[129] Pourvaziri, H., and Naderi, B. A hybrid multi-population genetic algorithm for the dynamic facility layout problem. *Applied Soft Computing 24* (2014), 457–469.

[130] Qodmanan, H. R., Nasiri, M., and Minaei-Bidgoli, B. Multi objective association rule mining with genetic algorithm without specifying minimum support and minimum confidence. *Expert Systems with applications 38*, 1 (2011), 288–298.

[131] Quinlan, J. R. Induction of decision trees. *Machine learning 1*, 1 (1986), 81–106.

[132] Radovich-cliintel.com, A. 3.17 billion internet users. `http://cliintel.com/3-17-billion-internet-users/`, 2015.

[133] Raghavan, S., Sarwesh, P., Marimuthu, C., and Chandrasekaran, K. Bat algorithm for scheduling workflow applications in cloud. In *Electronic Design, Computer Networks & Automated Verification (EDCAV), 2015 International Conference on* (2015), IEEE, pp. 139–144.

[134] Rahman, A., Ezeife, C. I., and Aggarwal, A. K. Wifi miner: an online apriori-infrequent based wireless intrusion system. In *Knowledge Discovery from Sensor Data*. Springer, 2010, pp. 76–93.

[135] Ramezankhani, A., Pournik, O., Shahrabi, J., Azizi, F., and Hadaegh, F. An application of association rule mining to extract risk pattern for type 2 diabetes using tehran lipid and glucose study database. *International journal of endocrinology and metabolism 13*, 2 (2015).

[136] Rechenberg, I. Evolution strategy: Optimization of technical systems by means of biological evolution. *Fromman-Holzboog, Stuttgart 104* (1973).

[137] Reddy, V. U., and Manoj, A. Optimal capacitor placement for loss reduction in distribution systems using bat algorithm. *IOSR journal of Engineering 2*, 10 (2012), 23–27.

[138] Redpath, R. *A Comparative Study of Visualization Techniques for Data Mining*. Monash University, 2000.

[139] Rekaby, A. Directed artificial bat algorithm (daba)-a new bio-inspired algorithm. In *Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on* (2013), IEEE, pp. 1241–1246.

[140] Romero, C., Romero, J. R., Luna, J. M., and Ventura, S. Mining rare association rules from e-learning data. In *Educational Data Mining 2010* (2010).

[141] Romero, C., and Ventura, S. Educational data mining: A survey from 1995 to 2005. *Expert systems with applications 33*, 1 (2007), 135–146.

[142] Romero, C., and Ventura, S. Educational data mining: a review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 40*, 6 (2010), 601–618.

[143] Russell, S. J., Norvig, P., Canny, J. F., Malik, J. M., and Edwards, D. D. *Artificial intelligence: a modern approach*, vol. 2. Prentice hall Upper Saddle River, 2003.

[144] Saida, I. B., Nadjet, K., and Omar, B. A new algorithm for data clustering based on cuckoo search optimization. In *Genetic and Evolutionary Computing*. Springer, 2014, pp. 55–64.

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

[146] Sánchez, D., Vila, M., Cerda, L., and Serrano, J.-M. Association rules applied to credit card fraud detection. *Expert Systems with Applications 36*, 2 (2009), 3630–3640.

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

[148] Savasere, A., Omiecinski, E. R., and Navathe, S. B. An efficient algorithm for mining association rules in large databases.

[149] Schaffer, D. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the 1st international Conference on Genetic Algorithms* (1985), L. Erlbaum Associates Inc., pp. 93–100.

[150] Senthilnath, J., Kulkarni, S., Benediktsson, J., and Yang, X.-S. A novel approach for multispectral satellite image classification based on the bat algorithm. *IEEE Geoscience and Remote Sensing Letters 13*, 4 (2016), 599–603.

[151] Serban, G., Czibula, I.-G., and Campan, A. A programming interface for medical diagnosis prediction. *Studia Universitatis" Babes-Bolyai", Informatica, LI (1), pag* (2006), 21–30.

[152] Shi, Y., and Eberhart, R. C. Empirical study of particle swarm optimization. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on* (1999), vol. 3, IEEE.

[153] Srinivasan, S., and Ramakrishnan, S. Evolutionary multi objective optimization for rule mining: a review. *Artificial Intelligence Review 36*, 3 (2011), 205–248.

[154] Statista. Number of internet users worldwide from 2005 to 2016 (in millions). `https://www.statista.com/statistics/273018/number-of-internet-users-worldwide/`, 2016.

[155] Stützle, T., and Hoos, H. H. Max–min ant system. *Future generation computer systems 16*, 8 (2000), 889–914.

[156] Sumathi, S., and Paneerselvam, S. *Computational intelligence paradigms: theory & applications using MATLAB.* CRC Press, 2010.

[157] Tan, P.-N., and Kumar, V. Interestingness measures for association patterns: A perspective. In *Proc. of Workshop on Postprocessing in Machine Learning and Data Mining* (2000).

[158] Tan, P.-N., Kumar, V., and Srivastava, J. Selecting the right objective measure for association analysis. *Information Systems 29*, 4 (2004), 293–313.

[159] 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.

[160] Toivonen, H. Sampling large databases for association rules. In *Proceedings of the 22th International Conference on Very Large Data Bases* (San Francisco, CA, USA, 1996), VLDB '96, Morgan Kaufmann Publishers Inc., pp. 134–145.

[161] Toivonen, H., et al. Sampling large databases for association rules. In *VLDB* (1996), vol. 96, pp. 134–145.

[162] Vieira, J., Dias, F. M., and Mota, A. Neuro-fuzzy systems: a survey. In *5th WSEAS NNA International Conference* (2004).

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

[164] Wakabi-Waiswa, P. P., and Baryamureeba, V. Mining high quality association rules using genetic algorithms. In *MAICS* (2011), Citeseer, pp. 73–78.

[165] Wang, G., and Guo, L. A novel hybrid bat algorithm with harmony search for global numerical optimization. *Journal of Applied Mathematics 2013* (2013).

[166] Wang, G., Guo, L., Duan, H., Liu, L., and Wang, H. A bat algorithm with mutation for ucav path planning. *The Scientific World Journal 2012* (2012).

[167] Wang, G.-G., Chang, B., and Zhang, Z. A multi-swarm bat algorithm for global optimization. In *Evolutionary Computation (CEC), 2015 IEEE Congress on* (May 2015), pp. 480–485.

[168] Wang, G.-G., Chu, H. E., and Mirjalili, S. Three-dimensional path planning for ucav using an improved bat algorithm. *Aerospace Science and Technology 49* (2016), 231–238.

[169] Wang, M., Zou, Q., and Lin, C. Multi dimensions association rules mining on adaptive genetic algorithm. In *IEEE International Conference on Uncertainly Reasoning on Knowledge Engineering* (2011).

[170] Wong, M.-H., Sze-To, H.-Y., Lo, L.-Y., Chan, T.-M., and Leung, K.-S. Discovering binding cores in protein-dna binding using association rule mining with statistical measures. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) 12*, 1 (2015), 142–154.

[171] Wu, H., Lu, Z., Pan, L., Xu, R., and Jiang, W. An improved apriori-based algorithm for association rules mining. In *Proceedings of the 6th International Conference on Fuzzy Systems and Knowledge Discovery - Volume 2* (Piscataway, NJ, USA, 2009), FSKD'09, IEEE Press, pp. 51–55.

[172] Wu, J.-Z., Liu, H.-W., and Wu, F.-L. A recommender system based on car pairwise comparisons on a mobile application using association rules. In *2016 IEEE International Conference on Industrial Technology (ICIT)* (2016), IEEE, pp. 1344–1346.

[173] Xie, J., Zhou, Y., and Chen, H. A novel bat algorithm based on differential operator and lévy flights trajectory. *Computational intelligence and neuroscience 2013* (2013).

[174] 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.

[175] Yang, X.-S. Firefly algorithms for multimodal optimization. In *International Symposium on Stochastic Algorithms* (2009), Springer, pp. 169–178.

[176] Yang, X.-S. Firefly algorithm. *Engineering Optimization* (2010), 221–230.

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

[178] Yang, X.-S. Bat algorithm for multi-objective optimisation. *International Journal of Bio-Inspired Computation 3*, 5 (2011), 267–274.

[179] Yang, X.-S. Multiobjective firefly algorithm for continuous optimization. *Engineering with Computers 29*, 2 (2013), 175–184.

[180] Yang, X.-S. *Nature-inspired optimization algorithms*. Elsevier, 2014.

[181] Yang, X.-S., and Deb, S. Cuckoo search via lévy flights. In *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on* (2009), IEEE, pp. 210–214.

[182] Yang, X.-S., and Deb, S. Engineering optimisation by cuckoo search. *International Journal of Mathematical Modelling and Numerical Optimisation 1*, 4 (2010), 330–343.

[183] Yang, X.-S., and Deb, S. Cuckoo search: recent advances and applications. *Neural Computing and Applications 24*, 1 (2014), 169–174.

[184] Yang, X.-S., Deb, S., Karamanoglu, M., and He, X. Cuckoo search for business optimization applications. In *Computing and Communication Systems (NCCCS), 2012 National Conference on* (2012), IEEE, pp. 1–5.

[185] Yang, X.-S., and He, X. Firefly algorithm: recent advances and applications. *International Journal of Swarm Intelligence 1*, 1 (2013), 36–50.

[186] Zadeh, L. A. Fuzzy sets. *Information and control 8*, 3 (1965), 338–353.

[187] Zadeh, L. A. Fuzzy logic and approximate reasoning. *Synthese 30*, 3-4 (1975), 407–428.

[188] Zadeh, L. A. Foreword. *Intelligent Automation & Soft Computing 1*, 1 (1995), 1–1.

[189] Zaiane, M.-L. A. O. R. Mining positive and negative association rules: An approach for confined rules, 2007.

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

[191] Zaki, M. J., and Gouda, K. Fast vertical mining using diffsets. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (2003), ACM, pp. 326–335.

[192] Zaki, M. J., Parthasarathy, S., Ogihara, M., Li, W., et al. New algorithms for fast discovery of association rules. In *KDD* (1997), vol. 97, pp. 283–286.

[193] Zhang, J. W., and Wang, G. G. Image matching using a bat algorithm with mutation. In *Applied Mechanics and Materials* (2012), vol. 203, Trans Tech Publ, pp. 88–93.

[194] Zhao, X., Liu, Z., and Yang, X. A multi-swarm cooperative multistage perturbation guiding particle swarm optimizer. *Applied Soft Computing 22* (2014), 77–93.

[195] Zhou, Y., Luo, Q., Xie, J., and Zheng, H. A hybrid bat algorithm with path relinking for the capacitated vehicle routing problem. In *Metaheuristics and Optimization in Civil Engineering*. Springer, 2016, pp. 255–276.