

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

UNIVERSITE SAAD DAHLEB –BLIDA 1
FACULTE DES SCIENCES
DEPARTEMENT DE L'INFORMATIQUE



Mémoire de fin d'étude

Présenté pour l'obtention du diplôme de MASTER

Domaine : Mathématique et Informatique

Filière : Informatique

Spécialité : Système Informatique et Réseaux

Thème :

*Combinaison de l'apprentissage par renforcement et de l'optimisation des
loups gris pour l'extraction d'itemsets*

Réalisé par :

HARAZI Selma

AISSAHINE Kenza Yousra

Encadré par :

Mme ZAHRAFATMAZOHRA

Année universitaire : 2022 / 2023

REMERCIEMENT

En préambule à ce mémoire, nous remercions ALLAH qui nous a aidé et donné la patience et le courage durant cette longue année d'étude.

Nous souhaitons adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

Ces remerciements vont tout d'abord à notre promotrice Mme ZAHRA FATIMA ZAHRA Pour sa disponibilité tout en long de la réalisation de ce Mémoire, Ainsi pour son inspiration, aide et son suivi.

Nos remerciements iront également vers tous ceux qui ont accepté avec bienveillance de participer au jury de ce mémoire.

On n'oublie pas nos parents pour leur contribution, leur soutien leur patience. Enfin, nous adressons nos plus sincères remerciements à tous nos proches et amis, qui nous ont toujours encouragées au cours de la réalisation de ce mémoire

Merci à tous et à toutes

Dédicaces

Je dédie ce modeste travail

Ames chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études.

À mon mari pour ses encouragements, son appui moral et son soutien.

A toute ma famille pour leur soutien tout au long de mon parcours universitaire.

A mon binôme KENZA.

Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infailible.

Merci d'être toujours là pour moi.

Je tiens à remercier aussi et chaleureusement mon encadreuse Mme ZAHRA FATMA ZOHRA de m'avoir permis de mener ce travail, pour leur engagement et leur soutien ainsi que pour la pertinence de leur remarque et de leur feed-back.

Mes vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

Enfin, Je tiens également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail, à mes amis, ma famille, Merci.

HARAZI Selma

Dédicaces

Je dédie ce modeste travail

À mes parents qui depuis mon plus jeune âge ont toujours
fait leur Maximum, en consacrant temps et argent, pour
m'éveiller et M'encourager dans mes passions. C'est grâce
à vous et pour vous que J'ai fait mon mémoire. Aucun mot
sur cette page ne saurait exprimer Ce que je vous dois, ni
combien je vous aime. Qu'Allah vous bénisse, Vous
assistez, vous venez en aide.

A ma tante SOUAD qui m'a encouragée énormément.

A ma promotrice Mme ZAHRA

FATMA ZAHRA

A mon binôme SELMA

A tous mes collègues

A tous ceux qui m'ont soutenu, qu'ils trouvent ici

l'expression de mon Amour et ma profonde

Gratitude.

AISSAHINE Kenza Yousra

Résumé

L'extraction de motifs fréquents est une technique des techniques les plus importantes dans de Data Mining, qui se repose sur un principe relativement simple. Elle consiste à découvrir des motifs ou itemsets (ensemble d'éléments) qui se répètent fréquemment dans un ensemble de données. Il existe plusieurs méthodes exactes pour l'extraction d'itemsets fréquents. Cependant ces solutions souffrent de problème de passage à l'échelle, ce qui a incité les chercheurs de ce domaine à introduire des méthodes approchées. Néanmoins, ces dernières souffrent aussi de problème de qualité de solutions qui est pauvre.

Ce qui rend nécessaire de proposer des méthodes approchées plus efficaces. En effet, notre travail consiste à proposer une méthode approchée efficace pour l'extraction de motifs fréquents à partir données tout en améliorent la qualité des solutions de cette dernière. Et cela en utilisant la combinaison d'une métaheuristique inspirée du comportement des loups gris (Optimisation des Loups Gris, ou Grey Wolf Optimizer, GWO) et l'apprentissage par renforcement, qui ont eu beaucoup de succès dans une variété de domaines.

Deux algorithmes ont été proposés pour l'extraction d'itemsets fréquents, le premier se base sur GWO et le deuxième se base sur une combinaison de l'apprentissage par renforcement avec GWO (appelés respectivement FIM-BGWO et FIM-QL-BGWO). Ces deux algorithmes ont été testés sur plusieurs datasets réels et synthétiques. Les résultats obtenus montrent une qualité de solution meilleure pour l'algorithme FIM-QL-BGWO par rapport son homologue FIM-BGWO tout en gardant un temps d'exécution acceptable.

Mots clés : Extraction de Motifs Fréquents, Itemsets Fréquents, Optimisation des Loups Gris, GWO, Apprentissage par Renforcement, Q-Learning.

Abstract

Frequent pattern extraction is one of the most important techniques in data mining, based on a relatively simple principle. It consists of discovering patterns or itemsets (sets of elements) that are frequently repeated in a dataset. There are several exact methods for extracting frequent itemsets. However, these solutions suffer from scaling problems, which has prompted researchers in this field to introduce approximate methods. However, these methods also suffer from poor solution quality. This makes it necessary to propose more efficient approximate methods.

Indeed, our work consists in proposing an efficient approximate method for the extraction of frequent patterns from data while improving the quality of its solutions. This is done using a combination of metaheuristic inspired by the behaviour of grey wolves (Grey Wolf Optimizer, GWO) and reinforcement learning, which have been very successful in a variety of domains.

Two algorithms have been proposed for frequent itemset extraction, the first based on GWO and the second based on a combination of reinforcement learning and GWO (called FIM-BGWO and FIM-QL-BGWO respectively). These two algorithms have been tested on several real and synthetic datasets. The results obtained show a better solution quality for the FIM-QL-BGWO algorithm compared with its FIM-BGWO counterpart, while maintaining an acceptable execution time.

Keywords: Frequent Pattern Extraction, Frequent Itemsets, Grey Wolf Optimisation, GWO, Reinforcement Learning, Q-Learning.

ملخص

يعد استخراج الأنماط المتكرر أحد أهم التقنيات في استخراج البيانات، بناءً على مبدأ بسيط نسبيًا. يتكون من اكتشاف أنماط أو مجموعات العناصر التي تتكرر بشكل متكرر في مجموعة البيانات. هناك العديد من الطرق الدقيقة لاستخراج مجموعات العناصر المتكررة. ومع ذلك، فإن هذه الحلول تعاني من مشاكل في قابلية التوسع، مما دفع الباحثين في هذا المجال إلى إدخال طرق تقريبية. ومع ذلك، فإن هذه الأساليب تعاني أيضًا من سوء نوعية الحلول. وهذا يجعل من الضروري اقتراح أساليب تقريبية أكثر كفاءة.

في الواقع، يتمثل عملنا في اقتراح طريقة تقريبية فعالة لاستخراج الأنماط المتكررة من البيانات مع تحسين جودة حلولها. يتم ذلك باستخدام مزيج من خوارزمية المستوحاة من سلوك الذئب الرمادية (Grey Wolf Optimizer , GWO) والتعلم المعزز، والتي كانت ناجحة جدًا في مجموعة متنوعة من المجالات.

تم اقتراح خوارزميتين لاستخراج مجموعات العناصر المتكررة، الأولى تستند إلى GWO والثانية تستند إلى مزيج من التعلم المعزز و GWO تسمى (FIM-BGWO و FIM-QL-BGWO على التوالي). تم اختبار هاتين الخوارزميتين على العديد من مجموعات البيانات الحقيقية والاصطناعية. تظهر النتائج التي تم الحصول عليها جودة حل أفضل لخوارزمية FIM-QL-BGWO مقارنة بنظيرتها FIM-BGWO، مع الحفاظ على وقت تنفيذ مقبول.

الكلمات الرئيسية: استخراج الأنماط المتكررة، مجموعات العناصر المتكررة، تحسين الذئب الرمادية GWO، التعلم المعزز، التعلم القائم على الجودة.

Table de matières

Introduction générale.....	1
Motivation et problématique.....	1
Objectif	1
Organisation du mémoire.....	2
Chapitre 1 : Méta-heuristiques et apprentissage par renforcement	3
1. Introduction.....	4
2. Définition de l'optimisation	4
3. Méta-heuristique	4
3.1.Les méthodes de résolutions des problèmes d'optimisation méta- heuristiques	5
3.2.Métaheuristique inspirée du comportement des loups gris	7
3.2.1. Définition	8
3.2.2. Algorithme de base GWO	9
4. Apprentissage automatique	11
5. Apprentissage par renforcement.....	12
5.1. Algorithmes d'apprentissage par renforcement	13
5.2. L'algorithme Q-Learning	14
5.2.1. Définition	14
5.2.2. Principe de Q-Learning	15
5.2.3. Les avantages de Q-Learning	15
6. Conclusion.....	17
Chapitre 2 : Extraction des itemsets fréquent.....	18
1. Introduction	19
2. Data-mining	19
2.1. Les propriétés principales du DataMining	19
3.Extraction des itemsets fréquents	20
3.1.Applications principales d'extraction des itemsets fréquents	21
3.2. Les algorithmes d'extraction des itemsets fréquents	22
3.2.1. Méthodes exactes	22

3.2.2. Comparaison des algorithmes	25
3.2.3. Méthodes approchés	26
4. Conclusion	27
Chapitre 3 : Approche proposé	28
1. Introduction	29
2. Motivation attribuée à l'utilisation du GWO et le RL	29
3. GWO pour l'extraction d'itemsets fréquents	29
3.1. Codification des individus (Wolves)	29
3.2. Fonction de fitness	30
3.3. Espace de recherche	30
4. Extraction d'itemsets fréquents basé sur la combinaison de (GWO) et le RL	32
5. Conclusion	39
Chapitre 4 : Test et validation	40
1.Introduction	41
2. Environnement de développement	41
2.1. Environnement matériel	41
2.2.Environnement logiciel	41
3.Datasets utilisé.....	41
4. Tests et résultats	42
4.1 Tests d'évaluation du temps d'exécution	42
4.2 Tests de la qualité	47
5. Conclusion	50
Conclusion générale	52
Perspectives	52
Bibliographie	54

Liste des figures

Figure 1 : Classification des méta-heuristiques.....	5
Figure 2 : La hiérarchie des loups gris.....	9
Figure 3 : Le moment ou attaquer une proie plutôt que chercher une proie.....	11
Figure 4 : Système d'apprentissage par renforcement.....	12
Figure 5 : Taxonomie des algorithmes d'apprentissage par renforcement.....	14
Figure 6 : Les étapes de la création du Q-Table.....	16
Figure 7 : Algorithme Apriori, Génération d'ensembles d'éléments candidats et fréquents.....	23
Figure 8 : Exemple du tableau Q et du tableau de récompenses.....	34
Figure 9 : Organigramme d'extraction d'itemsets fréquents à partir de données basé sur la combinaison de l'apprentissage par renforcement et GWO.....	36
Figure 10 : Temps d'exécution trouvés pour le dataset RecordLink en comparaison différentes méthodes et en utilisant différents minsup.....	43
Figure 11 : Temps d'exécution trouvés pour le dataset Connect en comparaison avec différentes méthodes et en utilisant différents minsup.....	44
Figure 12 : Temps d'exécution trouvés pour le dataset Mushrooms en comparaison avec différents méthodes et en utilisant différents minsup.....	45
Figure 13 : Temps d'exécution trouvés pour le dataset Chess en comparaison avec différentes et en utilisant différents minsup.....	46
Figure 14 : Pourcentage d'itemsets fréquents trouvés dans le dataset RecordLinken comparaison avec différentes méthodes.....	48
Figure 15 : Pourcentage d'itemsets fréquents trouvés dans le dataset Chess en comparaison avec différentes méthodes.....	49
Figure 16 : Pourcentage d'itemsets fréquents trouvés dans le dataset Mushrooms en comparaison avec différentes méthodes.....	50
Figure 17 : Pourcentage d'itemsets fréquents trouvés dans le dataset Connect en comparaison avec différentes méthodes.....	51

Liste des tableaux

Tableau 1 : Base de transaction.....	21
Tableau 2 : Données transactionnelles au format vertical.....	24
Tableau 3 : Différence entre l'algorithme Apriori et Eclat et entre Apriori et FPGrowth.....	25-26
Tableau 4 : Représentations des datasets utilisés.....	41-42
Tableau 5 : Représentation des paramètres utilisés dans les tests.....	42
Tableau 6 : le temps d'exécution (en seconde) trouvés pour le dataset RecordLink en comparaison avec différentes méthodes et en utilisant différents minsup.....	43
Tableau 7 : le temps d'exécution (en seconde) trouvés pour le dataset Connect en comparaison avec différentes méthodes et en utilisant différents minsup.....	44
Tableau 8 : le temps d'exécution (en seconde) trouvés pour le dataset Mushrooms en comparaison avec différentes méthodes et en utilisant différents minsup.....	45
Tableau 9 : le temps d'exécution (en seconde) trouvés pour le dataset Chess en comparaison avec différentes méthodes et en utilisant différents minsup.....	46
Tableau 10 : Pourcentage d'itemsets fréquents trouvés dans le dataset RecordLink en comparaison avec différentes méthodes.....	47
Tableau 11 : Pourcentage d'itemsets fréquents trouvés dans le dataset Chess en comparaison avec différentes méthodes.....	48
Tableau 12 : Pourcentage d'itemsets fréquents trouvés dans le dataset Mushrooms en comparaison avec différentes méthodes.....	49
Tableau 13 : Pourcentage d'itemsets fréquents trouvés dans le dataset Connect en comparaison avec différentes méthodes.....	50

INTRODUCTION GENERALE

Motivation et problématique

Étant l'étape cœur de plusieurs méthodes de fouille de données, l'extraction de motifs (itemsets) à partir de données a reçu une attention particulière de la part de la communauté des chercheurs de ce domaine. L'extraction de motifs fréquents est une tâche essentielle dans l'analyse des données, car elle est responsable d'extraire des événements, des tendances ou des éléments de données intéressantes ce qui est un avantage important dans les processus décisionnels des différents organismes qui cherche à augmenter leur revenus, réduire les couts et améliorer l'avantage concurrentiel. En outre, le développement rapide des applications dans les différents domaines comme le E-commerce et d'autres a muni à la génération des données imparfaites ou incertaines ce qui a suscité l'intention dans le domaine d'exploration de données.

L'extraction des motifs fréquents que ce soit à partir de données précises ou incertaines est un problème NP-Difficile. Les méthodes exactes d'extraction de motifs fréquents souffrent du problème de passage à l'échelle avec un nombre important des items et un volume de données considérable.

Par conséquent, l'extraction de motifs fréquents a été traitée comme étant un problème d'optimisation combinatoire. Plusieurs approches basées sur des métaheuristiques à population de solutions ont été proposées. Néanmoins, le nombre d'itemsets extraits par ces méthodes est très bas et loin par rapport à la quantité d'itemsets générée par les algorithmes exacts.

Objectif

L'apprentissage par renforcement est une des méthodes d'apprentissage automatique et de l'intelligence artificielle les plus remarquables, qui a connu un grand succès dans de nombreuses applications et dans plusieurs domaines.

Afin de résoudre ce problème de mauvaise qualité de solution duquel les méthodes approchées d'extraction de motifs souffrent, l'utilisation de l'apprentissage par renforcement pour de guider les métaheuristiques à mieux explorer l'espace de recherche de motifs semble une solution prometteuse.

En effet, l'objectif de ce travail est la combinaison de l'apprentissage par renforcement avec une métaheuristique inspirée du comportement des loups gris (Optimisation des Loups Gris, ou Grey Wolf Optimizer, GWO) pour une extraction plus efficace de motifs fréquents à partir de données. Cette métaheuristique a été choisie parce qu'elle a été appliquée avec succès dans plusieurs domaines.

Organisation du mémoire

Le présent mémoire est composé des parties suivantes :

Chapitre 1 : « Méta-heuristique et Apprentissage par Renforcement »

Dans ce chapitre, nous avons passé en revue les différentes techniques d'apprentissage automatique, en mettant l'accent sur l'apprentissage par renforcement et l'algorithme de la métaheuristique GWO.

Chapitre 2 : « Extraction des motifs fréquents »

Ce chapitre couvre les nombreux types de méthodes les plus couramment utilisées pour extraire des itemsets fréquents à partir de l'ensemble de données.

Chapitre 3 : « Approche proposée »

Ce chapitre présente l'approche proposée à notre problème.

Chapitre 4 : « Tests et Validation »

Le dernier chapitre illustre les tests et la validation de la méthode utilisée.

CHAPITRE 1 : Métaheuristiques et Apprentissage par Renforcement

1. Introduction

Un problème d'optimisation peut se ramener à un problème d'existence de solution de bonne qualité. Il consiste à rechercher une solution de qualité suffisante au regard d'un (des) critère(s) donné(s) et des objectifs à satisfaire. De nombreuses méthodes de résolution de ce genre de problème ont été proposées dans la littérature, parmi eux on trouve les métaheuristiques inspirés de la nature qui ont été utilisées avec succès dans diverses applications.

Ces dernières années, les algorithmes de l'apprentissage automatique (ML) ont également été mis en évidence dans la résolution des problèmes d'optimisation, en particulier l'apprentissage par renforcement.

Dans ce chapitre, nous explicitons différentes métaheuristiques ainsi que le mécanisme de l'apprentissage automatique. Ensuite, nous présentons l'algorithme d'optimisation des loups gris, l'apprentissage par renforcement et le Q-Learning en détails qui font l'objet de notre travail.

2. Définition de l'optimisation

Une fois que l'on dispose d'un outil pour évaluer le comportement et les performances d'un système étudié, on cherche à optimiser son fonctionnement. Il peut s'agir d'une étude monocritère, comme la minimisation du coût ou de la durée, ou d'une étude multi-objectifs dans laquelle on recherche le meilleur compromis entre différents indicateurs de performance comme la minimisation d'un coût. [1]

3. Méta-heuristique

Une méta-heuristique est constituée d'un ensemble de concepts fondamentaux qui permet d'aider à la conception des méthodes heuristiques pour un problème d'optimisation. Ainsi, une méta-heuristique est adaptable et applicable à une large classe de problèmes. Un ensemble de propriétés intéressantes a été proposé qui caractérisent les méta-heuristiques [1] :

- Les méta-heuristiques sont des stratégies qui permettent de guider la recherche d'une solution optimale.
- Le but visé par les méta-heuristiques est d'explorer l'espace de recherche efficacement afin de déterminer des solutions (presque) optimales.
- Les techniques qui constituent des algorithmes de type méta-heuristique vont de la simple procédure de recherche locale à des processus d'apprentissage complexes.

- Les méta-heuristiques sont en général non-déterministes et ne donnent aucune garantie d'optimalité.
- Les méta-heuristiques peuvent contenir des mécanismes qui permettent d'éviter d'être bloqué dans des régions de l'espace de recherche.
- Les méta-heuristiques peuvent faire usage de l'expérience accumulée durant la recherche de l'optimum pour mieux guider la suite du processus de recherche. Ces propriétés définissent le comportement de toutes les méta-heuristiques pendant la recherche d'une solution, allant de la plus simple à la plus complexe.
- Des problèmes d'optimisation avec une fonction objective et/ou des contraintes qui nécessitent beaucoup de temps de calcul. Certains problèmes d'optimisation réels sont caractérisés par un temps de calcul énorme de la fonction objectif [1].

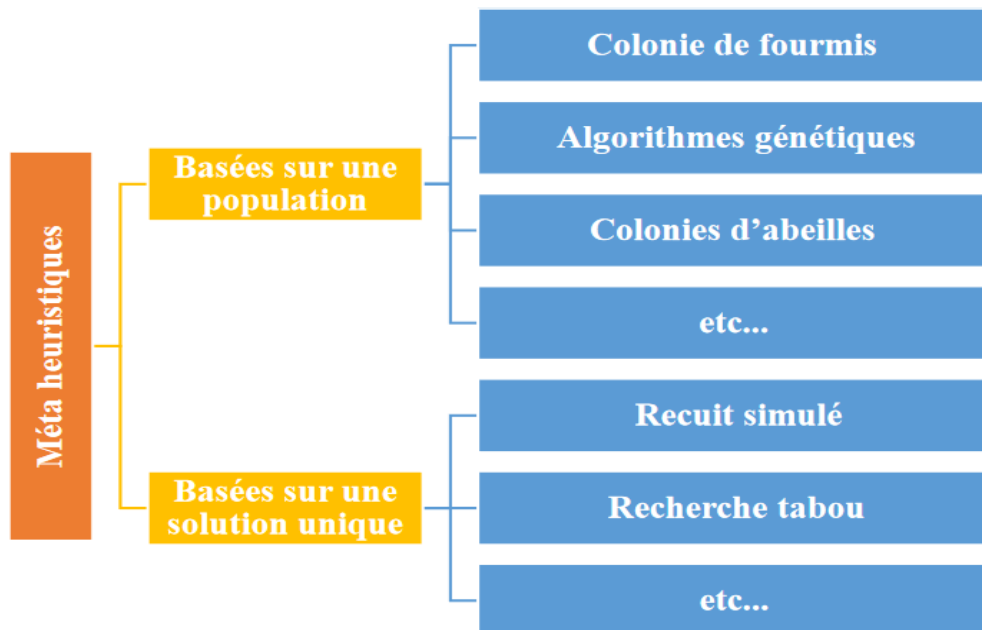


Figure 1: Classification des méta-heuristiques. [1]

3.1 Les méthodes de résolutions des problèmes d'optimisation méta- heuristiques

Dans cette nous présentons quelques métaheuristiques à solution unique et à population de solutions.

a. Méta-heuristique à base de solution unique

Les méthodes à base de solution unique commence initialement avec une seule solution et, à chaque itération, la meilleure solution actuelle est remplacée par une nouvelle solution, les méthodes à base de solution unique sont plus orientées vers l'exploitation. On trouve dans

cette catégorie la méthode de tabou, et la méthode de recuit simulé. [39]

Méthode tabou :

La recherche tabou est une méta-heuristique qui construit un voisinage en excluant un certain nombre de configurations récemment visitées (ou mouvements récents). Elle utilise toujours la meilleure solution voisine indépendamment du fait qu'elle améliore ou non la qualité de la solution précédente. Tant qu'on ne se trouve pas dans un optimum local, cette méthode se comporte comme une méthode de plus forte descente qui améliore à chaque itération la valeur de la fonction objective en choisissant une configuration meilleure. Mais lorsqu'elle a atteint un optimum local, la recherche tabou cherche à s'en : déplacement dans l'espace de recherche en sélectionnant à chaque itération le meilleur voisin dans un voisinage débarrassé des solutions récentes marquées tabou (pour éviter les cycles).[1]

Le recuit simulé :

La méthode du recuit simulé, appliquée aux problèmes d'optimisation, considère une procédure d'exploitation du voisinage qui permet de se diriger vers une solution voisine de moins bonne qualité avec une probabilité non nulle. Ceci permet d'échapper aux optima locaux. Au début de l'algorithme, un paramètre T est déterminé et décroît tout au long de l'algorithme pour tendre vers 0. De la valeur de ce paramètre dépend la probabilité p d'acceptation des solutions détériorantes.

L'intérêt du recuit simulé est qu'il existe une preuve de la convergence asymptotique. Ainsi, lorsque certaines conditions sont vérifiées, on a la garantie d'obtenir la solution optimale. Malheureusement, le paramétrage recommandé par la théorie n'est pas réaliste et il faut beaucoup de temps pour arriver à paramétrer ces méthodes. Cette méthode peut, elle aussi, nécessiter un critère d'arrêt, dans le cas, où le paramétrage "optimal" n'a pas été trouvé. [2]

b. Méta-heuristiques à base de population de solutions

Contrairement aux recherches locales, les méthodes à base de population de solutions améliorent, au fur et à mesure des itérations, une population de solutions. L'intérêt de ces méthodes est d'utiliser la population comme facteur de diversité. Il existe plusieurs stratégies d'évolution de cette population amenant à des méthodes telles que L'optimisation par essaim particulière, les algorithmes génétiques. [2]

L'optimisation par essaim particulière (PSO):

L'optimisation par essaim particulière (OEP), ou Particle Swarm Optimization (PSO) en anglais, est un algorithme évolutionnaire qui utilise une population de solutions candidates

pour développer une solution optimale au problème. Cet algorithme a été proposé par Russel Eberhart (ingénieur en électricité) et James Kennedy (socio-psychologue) en 1995. Cette méthode est inspirée à partir de l'intelligence en groupe qui est basée sur le comportement du déplacement des oiseaux ou des poissons pour la recherche de la nourriture. Lorsque les oiseaux sont à la recherche de nourriture, d'un endroit à l'autre, il y a toujours un oiseau qui peut sentir la nourriture et trouver l'endroit où elle peut être trouvée. Comme s'il y'a un échange d'informations entre eux à tout moment, ils finiront par affluer vers l'endroit où la nourriture peut être trouvée. L'algorithme proposé par Kennedy et Eberhart en 1995 cherche à simuler ce comportement social basé sur l'analyse de l'environnement et du voisinage et constitue alors une méthode de recherche d'optimum par l'observation des tendances des individus voisins. Chaque individu cherche à optimiser ses chances en suivant une tendance qu'il modère par ses propres vécus. Le modèle qu'ils ont proposé à ensuite été étendu en un algorithme simple et efficace d'optimisation. [3]

Algorithmes génétique :

Les algorithmes génétiques (AG) sont une famille très célèbre de méta-heuristiques. Ces algorithmes sont fondés sur des principes qui s'inspirent de la théorie de la sélection naturelle. Pratiquement, la résolution des problèmes d'optimisation par AG repose sur l'évolution d'une population de solutions candidates (individus) codés sous forme de chromosomes en fonction du problème traité ; le but étant de retenir des solutions viables par le biais d'opérateurs génétiques (opérateur de croisement et mutation). [4]

L'optimisation par colonies d'abeilles :

L'algorithme ABC s'inspire du modèle naturel du comportement des abeilles mellifères lors de la recherche de leur nourriture. Le processus de recherche de nourriture chez les abeilles est fondé sur un mécanisme de déplacement très efficace. Il leur permet d'attirer l'attention d'autres abeilles de la colonie aux sources alimentaires trouvées dans le but de collecter des ressources diverses. En fait, les abeilles utilisent un ensemble de danses frétilantes comme moyen de communication entre elles. Ces danses permettent aux abeilles de partager des informations sur la direction, la distance et la quantité du nectar avec ses congénères. La collaboration et la connaissance collective des abeilles de la même colonie sont basées sur l'échange d'information sur la quantité du nectar dans la source de nourriture trouvée par les différents membres. Des études sur le comportement de danses frétilantes des abeilles ont montré. [5]

3.2 Métaheuristique inspirée du comportement des loups gris

Le loup gris appartient à la famille des Canidés. Les loups gris sont considérés comme des prédateurs extrêmes, ce qui signifie qu'ils sont au sommet de la chaîne alimentaire. Les loups gris préfèrent surtout vivre en meute. La taille du groupe est de 5 à 12 en moyenne.

Le premier niveau est le leader le plus élevé des loups gris, appelé α . L'alpha est principalement responsable de prendre des décisions sur la chasse, le lieu de sommeil, le temps de se réveiller, et ainsi de suite. Les décisions de l'alpha sont dictées à la meute.

Le deuxième niveau dans la hiérarchie des loups gris est bêta. Les bêtas sont des loups subordonnés qui aident l'alpha dans la prise de décision ou d'autres activités de meute. Le loup bêta peut être mâle ou femelle, et il/elle est probablement le meilleur candidat pour être l'alpha au cas où un des loups alpha passe ou devient très vieux. Le bêta loup devrait respecter l'alpha, mais commande l'autre niveau inférieur loups aussi. Il joue le rôle d'un conseiller de l'alpha et discipline pour le pack. Le bêta renforce les commandes de l'alpha tout au long du pack et donne un retour à l'alpha.

Le loup gris le plus bas rang est oméga. L'oméga joue le rôle de bouc émissaire. Les loups omégas doivent toujours se soumettre à tous les autres loups dominants. Dans certains cas, l'oméga est également le babysitteur dans groupe.

Si un loup n'est pas un alpha, un bêta ou un oméga, il est appelé subordonné (ou delta dans certaines références). Les loups delta doivent soumettre alphas et bêtas, mais ils dominant les omégas. Scouts, sentinelles, anciens, chasseurs et gardiens appartiennent à cette catégorie. Ils sont responsables de surveiller les limites du territoire et en cas de danger. Les sentinelles protègent et garantissent la sécurité de la meute. Les anciens sont les loups expérimentés qui étaient alpha ou bêta. Les chasseurs aident les alphas et les bêtas lors de la chasse de proies et de fournir de la nourriture pour la meute. Enfin, les gardiens sont responsables de prendre soin des loups faibles, malades et blessés dans la meute.

En plus de la hiérarchie sociale des loups, la chasse de groupe est un autre comportement social intéressant des loups gris. Les principales phases de la chasse au loup gris sont Poursuivre, Encercler, Harceler et Attaquer. [6]

3.2.1 Définition

L'algorithme Grey Wolf Optimizer (GWO) s'inspire de l'intelligence sociale des meutes de loups gris décrite précédemment en matière de leadership et de chasse (Voir **Figure 2**).

Le loup le plus puissant **alpha**, qui dirige toute la meute pour la chasse, la migration et l'alimentation. Quand le loup alpha n'est pas dans la meute, est malade ou mort, le loup le plus fort parmi les des loups **bêta** prend la tête de la meute. Le pouvoir et la domination de **delta** et **oméga** sont inférieurs à ceux d'alpha et beta.

Une autre inspiration est l'approche de chasse des loups gris. Lorsqu'ils chassent une proie, les loups gris suivent une série d'étapes efficaces : poursuivre, encercler, harceler et attaquer. Cela leur permet de chasser de grosses proies. [7]

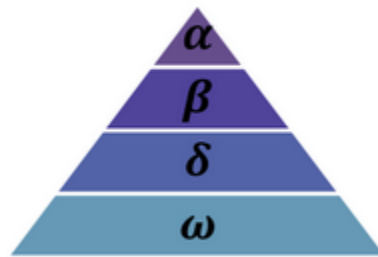


Figure 2: La hiérarchie des loups gris. [8]

3.2.2 Algorithme de base GWO

L'algorithme GWO comporte les différentes étapes qui sont expliquées ci-dessous [7] :

- L'encercllement de la proie ;
- Le comportement de la chasse ;
- L'attaque des proies ;
- Recherche de la proie.

Pour convertir cela en un modèle mathématique, la solution optimale est Alpha. La deuxième et la troisième solution les plus optimales sont respectivement celle des loups Bêta et Delta. Omega suit ces trois loups. [7]

a. L'encercllement de la proie

Les loups gris encerclent les proies pendant la chasse. Afin de modéliser mathématiquement le comportement d'encercllement. Des équations sont proposées :

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)| \quad (1)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad (2)$$

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \quad (3)$$

$$\vec{C} = 2\vec{r}_2 \quad (4)$$

Où t indique l'itération courante, A et C sont des vecteurs de coefficient, Xp est le vecteur de position de la proie, et X indique le vecteur de position d'un loup gris.

A est diminué linéairement de 2 à 0, et r1, r2 sont des vecteurs aléatoires. [6]

b. La chasse

Les loups gris ont la capacité de reconnaître l'emplacement des proies et les encercler. La chasse est généralement guidée par l'alpha. Le bêta et delta peuvent également participer à la chasse occasionnelle. Cependant, dans un espace de recherche abstrait, l'emplacement de l'optimum (proie) n'est pas clair. Afin de simuler mathématiquement le comportement de chasse des loups gris, il est supposé que l'alpha (meilleure solution possible) bêta et delta ont de meilleures connaissances sur l'emplacement potentiel des proies. Par conséquent, nous conservons les trois premières meilleures solutions obtenues jusqu'à présent et obligeons les autres agents de recherche (y compris les omégas) à se concentrer sur l'emplacement potentiel de la proie, et à mettre à jour leurs positions en fonction de la position des meilleurs agents de recherche.

En fonction de la position des meilleurs agents de recherche. Les formules suivantes sont proposées à cet égard [6]:

$$\vec{D}_{\alpha i} = |\vec{C}_1 \cdot \vec{X}_{\alpha i}(t) - \vec{X}_i(t)| \quad (5)$$

$$\vec{D}_{\beta i} = |\vec{C}_2 \cdot \vec{X}_{\beta i}(t) - \vec{X}_i(t)| \quad (6)$$

$$\vec{D}_{\delta i} = |\vec{C}_3 \cdot \vec{X}_{\delta i}(t) - \vec{X}_i(t)| \quad (7)$$

$$\vec{X}_i(nt) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (8)$$

X α , X β et X δ représente les positions des loups.

C1, C2, C3 sont des vecteurs aléatoires et X indique la position de la solution actuelle. [6]

c. Attaquer une proie

Les loups gris terminent la chasse en attaquant la proie lorsqu'elle cesse de se déplacer. Afin de modéliser mathématiquement l'approche de la proie, la valeur de a $\vec{}$ est diminué. Notez que la plage de fluctuation de ~A est également diminuée de a $\vec{}$. En d'autres termes ~A est une valeur aléatoire dans l'intervalle [-2a, 2a] où a est diminué de 2 à 0 au cours des itérations.

Lorsque des valeurs aléatoires de $\sim A$ sont dans $[1,1]$, la position suivante d'un agent de recherche peut être n'importe quelle position entre sa position actuelle et la position de la proie. La **Figure 3** montre que $|A| < 1$ force les loups à attaquer vers la proie.

Grâce à ces opérateurs l'algorithme GWO permet à ses agents de recherche de mettre à jour leur position en fonction de l'emplacement des loups alpha, bêta, et delta; et attaquer vers la proie. Cependant, l'algorithme GWO est sujet à la stagnation dans les solutions locales avec ces opérateurs. Il est vrai que le mécanisme d'encerclement proposé montre l'exploration dans une certaine mesure, mais GWO a besoin de plus d'opérateurs pour mettre l'accent sur l'exploration. [6]

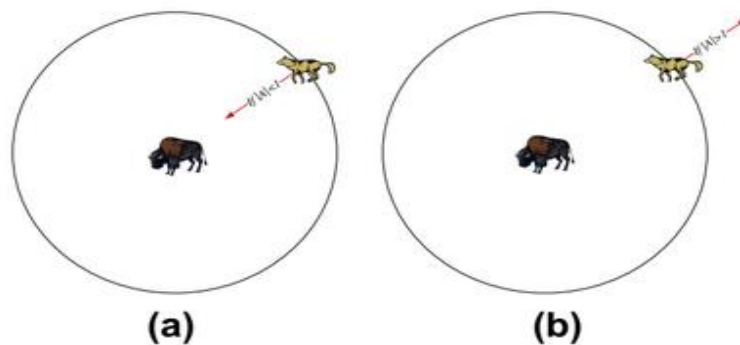


Figure 3: Le moment ou attaquer une proie plutôt que chercher une proie. [6]

d. Recherche de la proie

Puisque l'emplacement de la proie est prédit sur la base de la position de alpha, beta et delta, la procédure de recherche est donc déviée vers trois chemins différents, ce qui montre le comportement d'exploration de la technique GWO et comme la valeur de C se situe entre $[1,0]$, ce qui montre la propriété d'exploitation de l'algorithme.[7]

4. Apprentissage automatique

L'apprentissage automatique est un domaine qui se situe à l'intersection de la statistique, de l'intelligence artificielle et de l'informatique [9], afin de développer des modèles qui peuvent apprendre par l'expérience. Ces modèles peuvent être classés dans la catégorie [10] :

- **Apprentissage supervisé:** les modèles sont formés avec les paires d'entrées et de sorties souhaitées, de sorte que vous pouvez générer des sorties à partir de nouvelles entrées. [11]
- **Apprentissage non supervisé:** seules les données d'entrée sont connues et elles visent à en tirer des connaissances. [12]

Agent : Un agent RL est un programme que nous apprenons à faire de bons choix (par exemple : un robot qui est formé pour se déplacer dans une maison sans s'écraser).

Environnement : L'environnement de l'agent est l'espace dans lequel il ou elle interagit (la maison où le robot se déplace). L'agent n'a aucun contrôle sur l'environnement; tout ce qu'il a est le contrôle de ses propres activités (le robot ne peut pas contrôler où une table est dans la maison, mais il peut marcher autour de lui pour sortir de l'affaire de crash).

État : La situation actuelle de l'agent est définie par l'état (cela peut être la position exacte du Robot dans la maison, ou l'alignement de ses deux jambes, ou sa posture actuelle ; cela dépend de la façon dont nous abordons le problème).

Action : La décision prise par l'agent dans le pas temporel courant (déplacer sa jambe droite ou gauche, lever son bras, soulever un objet, tourner à droite ou à gauche, etc.). Nous savons à l'avance quelles mesures (décisions) l'agent peuvent prendre.

Politique : Une politique est le processus intellectuel qui sert à décider d'une action. Il s'agit d'une distribution de probabilité attribuée à l'ensemble des actions en pratique. Les actions qui sont très enrichissantes auront une forte probabilité, et vice versa. Il est important de noter que ce n'est pas parce qu'une action a une faible probabilité qu'elle ne sera pas choisie. C'est seulement qu'il a moins de chances d'être choisi.

5.1 Algorithmes d'apprentissage par renforcement

Les méthodes de RL sont généralement divisées en approches sans modèle et basées sur modèle, mais en général, les techniques sans modèle sont considérées. Les algorithmes qui composent actuellement les approches sans modèle peuvent être classés en deux sous-groupes : les méthodes fondées sur les politiques et les méthodes fondées sur les valeurs.

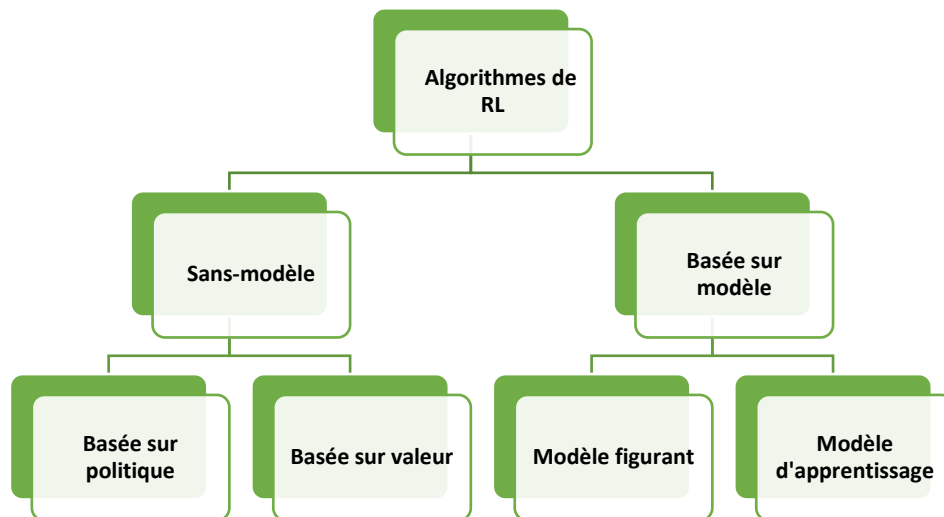


Figure 5: Taxonomie des algorithmes d'apprentissage par renforcement. [19]

Les algorithmes basés sur la valeur sont pratiques pour travailler en harmonie avec les algorithmes métaheuristiques. C'est parce qu'il est sans modèle et ne suit pas une politique spécifique, offrant une plus grande flexibilité. Dans les méthodes RL fondées sur la valeur, un agent de renforcement n'a aucune connaissance préalable de l'environnement et il apprend à décider en fonction de ses actions, et de l'expérience acquise, afin d'atteindre son objectif. [19]

5.2 L'algorithme Q-Learning

5.2.2 Définition

Q-Learning est une technique d'apprentissage par renforcement qui fonctionne en apprenant une fonction de valeur-action qui donne l'utilité attendue de prendre une action donnée dans un état donné et de suivre une politique fixe par la suite. L'une des forces de Q-Learning est sa capacité à comparer l'utilité attendue des actions disponibles sans avoir besoin d'un modèle de l'environnement [20].

Les modèles Q-Learning peuvent améliorer les processus dans divers scénarios. Voici quelques exemples d'utilisation de Q-Learning :

- Systèmes de recommandation: Les modèles Q-Learning peuvent aider à optimiser les systèmes de recommandation, comme les plateformes publicitaires. Par exemple, un système publicitaire qui recommande des produits couramment achetés ensemble peut être optimisé en fonction de ce que les utilisateurs choisissent.

- Robotique: Les modèles Q-Learning peuvent aider les robots à exécuter diverses tâches, comme la manipulation d'objets, l'évitement d'obstacles et le transport.

5.2.3 Principe de Q-Learning

Dans l'algorithme Q-Learning, l'agent essaie de trouver la meilleure action considérant son état actuel selon un mécanisme fondé sur la valeur. L'agent apprend de ses actions et de ses états. Dans cet algorithme, une politique prédéterminée n'est pas nécessaire, vu qu'un agent de RL prend des actions aléatoires, puis obtient une récompense ou pénalité, et progressivement une expérience est construite pour l'agent basé sur ses actions qui conduisent à des récompenses.

Dans l'algorithme Q-Learning, une table appelée Q-Table est définie et l'agent tente de mettre à jour son état pour sélectionner la meilleure action basée sur les valeurs Q-Table en considérant toutes les actions possibles qu'il peut prendre.

Par conséquent, chaque agent en action décide d'explorer ou d'exploiter l'Environnement [19].

Formellement l'algorithme se fait en trois étapes :

1. Dans l'état s_t où l'agent choisit une action et observe la réponse de l'environnement qui donne la récompense r_t et le nouvel état de l'environnement s_{t+1} .
2. Nous calculons une cible composée d'une récompense r_t et d'un ensemble de futures récompenses attendues en $t+1$, approximées en utilisant les connaissances actuelles (Q) et un facteur d'actualisation (γ)

$$\text{valeur } Q \text{ cible} = r_{t+1} + \gamma \text{Max} Q_{t(s_{t+1}, a)} - Q_t(s_t, a_t)$$

3. Nous mettons à jour la fonction pour le couple en mélangeant les connaissances découlant de l'expérience immédiate (cible) et les connaissances antérieures, en utilisant l'équation de Bellman ci-dessous :

$$Q_{(t+1)}(s_t, a_t) = Q_t(s_t, a_t) + \alpha [r_{t+1} + \gamma \text{Max} Q_{t(s_{t+1}, a)} - Q_t(s_t, a_t)]$$

Tel que:

- $Q_t(s_t, a_t)$: Valeur Q actuelle.
- α : le taux d'apprentissage détermine la vitesse ou la lenteur de l'apprentissage du modèle.

- γ : facteur de remise (facteur d'actualisation) qui définit l'importance des récompenses que nous obtenons maintenant par rapport à plus tard dans l'épisode.

- r_{t+1} : Récompense.

- $MaxQ_{t(s_{t+1},a)}$: Future récompense maximale attendue. [18]

Q-Table : c'est la structure de données utilisée pour calculer les futures récompenses maximales attendues pour chaque activité d'apprentissage à chaque état de connaissance de l'agent. Ce tableau guidera l'agent Q-Learning à choisir la « meilleure » décision afin de maximiser son gain d'apprentissage. Chaque valeur de la table Q est d'abord initialisée au hasard, puis apprise via la fonction Q (ou l'équation de Bellman).

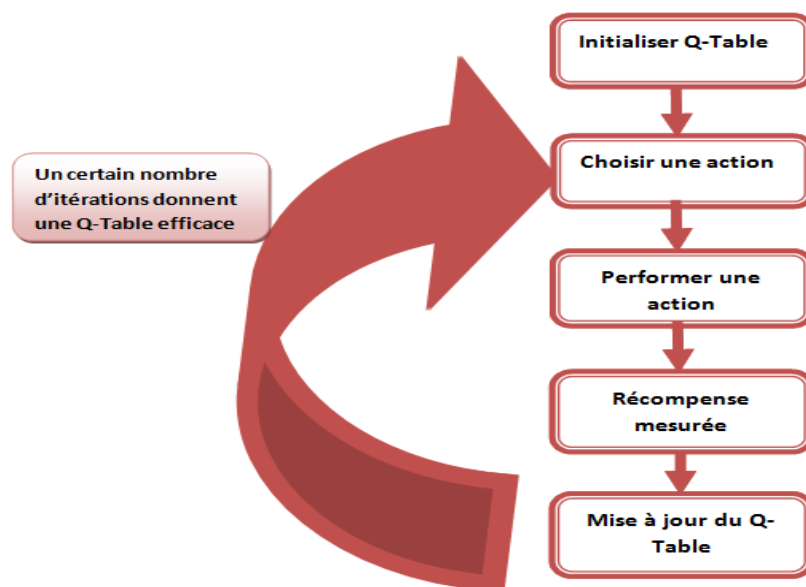


Figure 6 : Les étapes de la création du Q-Table. [18]

5.2.4. Les avantages de Q-Learning

L'approche Q-Learning de l'apprentissage par renforcement peut être potentiellement avantageuse pour plusieurs raisons, notamment:

Sans modèle: L'approche sans modèle est le fondement de Q-Learning et l'un des plus grands avantages potentiels pour certaines utilisations. Plutôt que d'exiger une connaissance préalable d'un environnement, l'agent de Q-Learning peut en apprendre davantage sur

l'environnement pendant sa formation. L'approche sans modèle est particulièrement avantageuse pour les scénarios où la dynamique sous-jacente d'un environnement est difficile à modéliser ou complètement inconnue.

Optimisation hors politique: Le modèle peut optimiser pour obtenir le meilleur résultat possible sans être strictement lié à une politique qui pourrait ne pas permettre le même degré d'optimisation.

Flexibilité: L'approche hors politique sans modèle permet une flexibilité Q-Learning pour travailler sur une variété de problèmes et d'environnements.

Formation hors ligne : Un modèle Q-Learning peut être déployé sur des ensembles de données pré-collectés et hors ligne.

6. Conclusion

Dans ce chapitre, on a présenté les méthodes basées sur l'intelligence par essaim pour la résolution des problèmes d'optimisation, qui ont construit une tendance très active ces dernières décennies. Nous nous focalisons sur l'algorithme d'optimisation des loups gris (GWO).

En outre, Nous avons défini l'apprentissage automatique et ces types en mettant l'accent sur l'apprentissage par renforcement et le Q-Learning ; afin d'expliquer le choix de la structure utilisée dans le chapitre suivant.

CHAPITRE 2 : Extraction des Itemsets Fréquents

1. Introduction

L'extraction de connaissances dans les bases de données, également appelé Data Mining, désigne le processus non trivial permettant d'extraire des informations et des connaissances utiles qui sont enfouies dans les bases de données, L'extraction de connaissances peut prendre de multiples formes, permettant de délivrer à des experts divers types de connaissances. A cet égard, les règles d'association, leurs variantes étendues plus particulièrement et les motifs fréquents sont des modèles fréquemment fournis aux utilisateurs finaux. En effet, On présente dans ce chapitre les principes de base de l'extraction de motifs fréquents ainsi que leurs importantes méthodes et algorithmes.

2. DataMining

Forage de données, explorations de données ou fouilles de données, ce sont les traductions possibles du Data Mining en Français. En règle générale, le terme Data Mining désigne[20] :« l'analyse de données depuis différentes perspectives et le fait de transformer ces données en informations utiles, en établissant des relations entre les données ou en repérant des patterns ».

Ces informations peuvent ensuite être utilisées par les entreprises pour augmenter un chiffre d'affaires ou pour réduire des coûts. Elles peuvent également servir à mieux comprendre une clientèle afin d'établir de meilleures stratégies marketing.

Les logiciels Data Mining font partie des outils analytiques utilisés pour l'analyse de données. Ils permettent aux utilisateurs d'analyser des données sous différents angles, de les catégoriser, et de résumer les relations identifiées. Techniquement, le Data Mining est [21]:« le procédé permettant de trouver des corrélations ou des patterns entre de nombreuses bases de données relationnelles ».

2.1 Les propriétés principales du Data Mining

On dénombre 3 propriétés principales du Data Mining [21] :

La découverte automatique de patterns

Le Data Mining repose sur le développement de modèles. Un modèle utilise un algorithme pour agir sur un ensemble de données. La notion de découverte automatique se réfère à l'exécution de modèles de Data Mining. Les modèles de Data Mining peuvent être utilisés pour miner les données sur lesquelles ils sont bâtis, mais la plupart des types de modèles peuvent être généralisés à de nouvelles données. Le processus permettant d'appliquer un

modèle à de nouvelles données est appelé scoring.

La prédiction de résultats probables

De nombreuses formes de Data Mining sont prédictives. Par exemple, un modèle peut prédire un résultat basé sur l'éducation et d'autres facteurs démographiques. Les prédictions ont une probabilité associée. Certaines formes de Data Mining prédictif génèrent des règles, qui sont les conditions pour obtenir un résultat. Par exemple, une règle peut spécifier qu'une personne dotée d'un bachelors et vivant dans un quartier précis a une probabilité d'avoir un meilleur salaire que la moyenne régionale.

La création d'informations exploitables

Le Data Mining permet de dégager des informations exploitables de larges volumes de données. Par exemple, un planificateur urbain peut utiliser un modèle pour prédire le revenu en se basant sur les données démographiques pour développer un plan pour les ménages à bas revenu. Une agence de location de voiture peut utiliser un modèle pour identifier des segments de consommateurs afin de créer une promotion ciblant les clients à forte valeur.

3. Extraction des itemsets fréquents

Frequent Itemset Mining (FIM) est l'une des techniques les plus connues pour extraire les connaissances des données. L'exploration des motifs (ensembles d'éléments ou itemsets) fréquents (FIM) est une tâche essentielle de l'analyse des données, car elle permet d'extraire les événements, les modèles ou les éléments qui se produisent fréquemment dans les données. Les résultats d'une telle analyse de motifs offrent des avantages importants dans les processus de prise de décision.

Cependant, les solutions algorithmiques pour l'extraction de ce type de motifs ne sont pas simples car la complexité de calcul augmente de façon exponentielle avec le nombre d'éléments dans les données. Ce problème, ainsi que l'importante consommation de mémoire présente dans le processus d'extraction, rend nécessaire de proposer des solutions extrêmement efficaces. [22]

Le FIM est chargé d'extraire dans les données tout élément fréquent existant (dont la fréquence d'occurrence n'est pas inférieure à un certain seuil). Cette tâche a été proposée au début des années 90 pour découvrir des éléments souvent co-occurents dans l'analyse du panier de consommation, et a été initialement appelé grand itemset mining. [23]

Tableau 1 : Base de transaction. [24]

S1	S2	S3	S4
1	0	1	0
0	1	0	0
0	0	0	1
0	1	1	1
0	1	1	0
0	1	1	0
1	1	1	1
1	0	1	0
1	1	1	0
1	1	1	0

Voici quelque définition des termes utilisées dans l'extraction de motifs [24]:

Item : un item correspond a un produit .dans cette exemple nous avons 4 items (S1, S2, S3, S4).

Support : le support d'un item est égal au nombre de transactions dans lesquelles il apparait.

Itemset : un itemset est un ensemble d'item. (ex. {S1, S2} est un itemset de cardinal CARD ({S1, S2} = 2). Le support d'un itemset comptabilise le nombre de transactions dans lesquelles les items apparaissent simultanément

itemset fréquent : Un itemset est dit fréquent si son support est supérieur à un seuil défini à l'avance, paramètre de l'algorithme de recherche.

Superset : Un superset est un itemset défini par rapport à un autre itemset.

Itemset maximal : si aucun des ses superset n'est fréquent.

3.1 Applications principales d'extraction des itemsets fréquents

L'application principale pour le FIM était l'analyse de panier de marché, où le comportement des clients était rassemblés sous forme de paniers d'articles achetés ensemble .Un exemple assez ancien, mais toujours significatif, d'un modèle fréquent est l'itemset, qui suggère que les deux articles ont tendance à être achetés en même temps.

La FIM a élargi ses horizons [23] et, de nos jours, elle est appliquée à des domaines tels que l'extraction de texte pour découvrir des co-occurrences de mots en termes de contiguïté et trouver des bigrammes, des trigrammes ou des phrases fréquentes dans les textes .L'analyse visant à déterminer si la fréquence des phrases dans un texte augmente ou diminue au fil du temps est essentielle pour fournir des indications précieuses sur les tendances (réseaux sociaux ou nouvelles). Cette analyse spécifique est, par exemple, une question clé dans les domaines médicaux pour identifier et comprendre la propagation des maladies infectieuses.

La FIM a également été pris en considération pour décrire le risque de maladies cardiaques, ce qui a facilité le travail des médecins pour prendre des décisions diagnostiques

et déterminer le niveau de risque des patients à un stade précoce.

En passant au domaine biologique, la FIM a été envisagé pour identifier les propriétés communes et utiles des données sous-jacentes (séquences de molécules biologiques complexes, puces à puces ou réseaux d'interaction protéique) à diverses fins médicales (gènes liés au cancer du côlon, profils d'expression des sous-types de leucémie lymphoblastique aiguë, expressions des gènes pour la tumeur mammaire, etc.)

Les précédents ne sont que quelques exemples d'applications FIM, mais de nombreux problèmes réels ont déjà été abordés par cette tâche.

3.2 Les algorithmes d'extraction des itemsets fréquents

De nombreux algorithmes ont été proposés par différents chercheurs pour améliorer la technique en FIM. Cependant, des améliorations doivent encore être apportées aux performances des algorithmes FIM existants, car la plupart des algorithmes actuels ne conviennent pas à l'extraction d'un énorme ensemble de données avec un nombre croissant de données. Les deux principaux défis auxquels sont confrontés la plupart des algorithmes FPM sont : un temps d'exécution long et une énorme consommation d'espace mémoire dans l'exécution de l'algorithme pour exploiter tous les modèles fréquents cachés. [23]

3.2.1 Méthodes exactes

Dans les méthodes exactes toutes les solutions de l'espace de recherche sont énumérées implicitement. Grâce a ces méthodes on peut trouver des solutions optimales .Mais ces méthodes s'avèrent malgré les progrès réalisé, plutôt inefficaces a mesure que la taille du problème devient important.[25]

Les méthodes exactes ont permis de trouver des solutions optimales pour des problèmes de taille raisonnable et rencontrent généralement des difficultés faces aux applications de taille important [26] .dans cette classe des méthodes exactes on peut trouver les algorithmes classique suivant :

a- Algorithme Apriori

Apriori est le tout premier algorithme pour l'extraction de motifs fréquents. Il a été donné par Agarwal et srikant en 1994 [27]. Apriori est un algorithme classique d'apprentissage des règles d'association. Il est conçu pour fonctionner sur des bases de données contenant des transactions, l'algorithme tente de trouver des sous-ensembles communs à au moins un nombre minimal MinSup (seuil de confiance ou seuil de coupure) des ensembles d'éléments. [28]

- Le premier passage de l'algorithme consiste à compter les occurrences d'éléments pour déterminer les grands ensembles d'items.
- Ce processus est répété jusqu'à ce qu'aucun nouveau motif ne soit identifié. Items (k + 1) les ensembles d'éléments de longueur candidats sont générés à partir de grands ensembles d'éléments de longueur k.
- Les ensembles d'éléments candidats contenant des sous-ensembles de longueur k non importants sont élagués.
- La prise en charge de chaque ensemble d'éléments candidats est comptée en analysant la base de données.
- Éliminer les itemsets candidats qui sont petits (Seuil inférieur au Sup_min), la figure montre tout cette procédure.

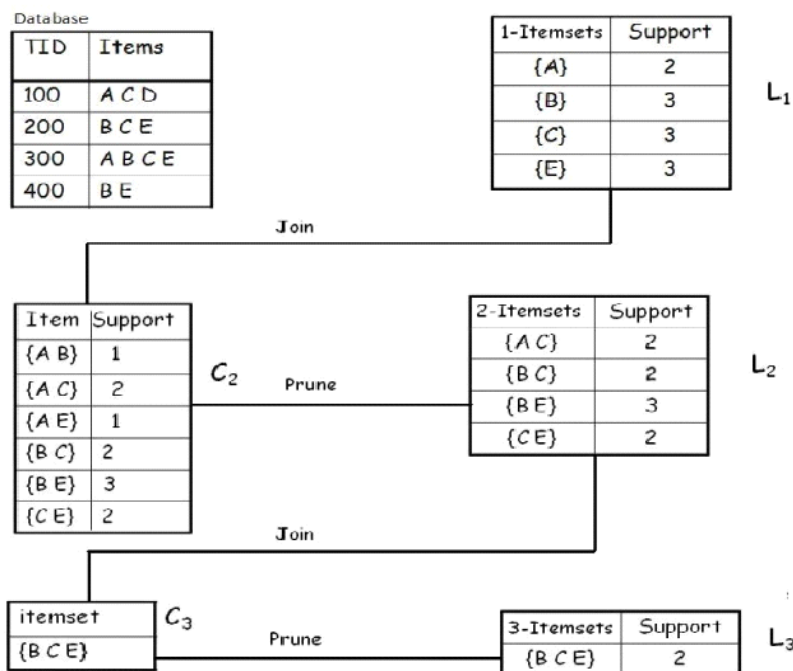


Figure 7 : Algorithme Apriori, Génération d'ensembles d'éléments candidats et Fréquents. [28]

b- Algorithme FP-Growth

L'algorithme FP-Growth a été introduit par Han et al. en 2000, ils ont dit qu'il est actuellement l'une des approches les plus rapides pour l'extraction fréquente d'ensembles d'articles. C'est une méthode différente des approches par niveaux permettant d'extraire des itemsets fréquents sans génération de candidats, chose qui nous permet d'éviter les parcours et les passages répétés de la base de données. [28]

- On effectue un premier parcours de la base T pour déterminer les items fréquents en fonction du support minimum fourni. Ces items seront triés par la suite par ordre décroissant de support dans une liste (L). Les items ainsi triés seront traités dans cet ordre.
- Un second parcours de T est alors effectué. Chaque transaction est alors triée selon l'ordre des items dans L. Le nœud racine de l'arbre {null} est d'abord créé. Durant ce même parcours, une branche sera créé pour chaque transaction, mais des transactions ayant un même préfixe partageront le même début d'une branche de l'arbre, ainsi deux transactions identiques seront représentées par une seule et même branche.

La raison pour laquelle les items sont traités du plus fréquent au moins fréquent est que les items fréquents seront proches de la racine et seront mieux partagés par les transactions. Ceci fait du FP-tree une bonne structure compacte pour représenter les bases transactionnelles.

c- Algorithme Eclat

ECLAT (Equivalence Class Transformation) a été introduit par Zaki, Parthasarathy, Ogihara et Li en 1997[29], Eclat a été conçu pour sur monter les inconvénients d'algorithme Apriori. Il utilise la mémoire agrégée du système en partitionnant les candidats en ensembles disjoints à l'aide du partitionnement par classe d'équivalence. Il dissocie la dépendance entre les transformateurs en droit en commençant de sorte que le coût de redistribution puisse être amorti par les itérations ultérieures. Eclat utilise la structure de base de données verticale qui regroupe toutes les informations pertinentes dans la liste des objets.

Il utilise algorithme de recherche en profondeur d'abord (Depth-First Search) et la base de données n'a pas besoin d'être scannée plusieurs fois pour que identifier les éléments (k + 1).La base de données est analysée une seule fois pour transformer les données du format horizontal dans le format vertical.

Tableau 2: Données transactionnelles au format vertical. [28]

itemset	TID_set
I1	{T100, T400, T500, T700, T800, T900}
I2	{T100, T200, T300, T400, T600, T800, T900}
I3	{T300, T500, T600, T700, T800, T900}
I4	{T200, T400}
I5	{T100, T800}

Eclat est composé de trois phases principales [28]:

- La phase d’initialisation : construction globale des 2-itemsets.
- La phase de transformation : partitionnement de l’ensemble des 2-itemsets fréquents et distribution de ces partitions aux autres processeurs. Transformation verticale de la base.
- La phase asynchrone : construction des k-itemsets fréquents.

3.2.2 Comparaison des algorithmes

Les deux tableaux suivants montrent une comparaison entre Apriori et Eclat, Apriori et FPGrowth.

Tableau 3: Différence entre l’algorithme Apriori et Eclat et entre Apriori et FPGrowth. [28]

Paramètre	Algorithme Apriori	Algorithme Eclat	Algorithme FPGrowth
Technique	Il utilise une approche de recherche d'abord large et utilise la propriété apriori (tous les sous-ensembles non vides d'un ensemble d'éléments fréquents doivent être fréquents) et la méthode joy-prime	Il utilise une approche de recherche en profondeur d'abord et utilise l'intersection de la liste d'identifiants de transaction pour générer des ensembles d'éléments candidats	Il construit une base d'arborescence FP et un modèle conditionnel à partir de la base de données
Utilisation de la mémoire	En raison de la grande quantité de candidats sont produites alors	Nécessite moins de mémoire que apriori si les itemsets sont peu nombreux	Nécessite moins de mémoire que apriori si les itemsets sont peu nombreux

	nécessitent un grand espace mémoire		
Temps	le temps d'exécution est plus que le temps perdu à produire des candidats à chaque fois.	Temps d'exécution est petit que l'algorithme apriori.	Temps d'exécution est petit que l'algorithme apriori

3.2.3 Méthodes approchées

Une méthode approchée est une méthode d'optimisation qui a pour but de trouver une solution réalisable de la fonction objective en un temps raisonnable, mais sans garanties d'optimalité. L'avantage principal de ces méthodes est qu'elles peuvent s'appliquer à n'importe quelle classe de problème, facile ou très difficile. [26]

Les approches d'extraction d'itemsets fréquents en se basant sur les métaheuristiques peuvent être classifiées en deux catégories approches évolutives, et approches d'intelligence d'essaim.

a. Approches évolutives

À notre connaissance, le premier algorithme génétique pour l'extraction de règles d'association et d'items fréquents est GAR [30]. L'une des limites de cet algorithme est qu'il utilise une représentation inefficace des individus (itemsets). Les individus sont représentés par leur taille (nombre d'éléments qu'ils contiennent). La taille des individus peut varier d'une population à l'autre, ce qui dégrade les performances des opérateurs de croisement et de mutation. De nombreux algorithmes génétiques ont ensuite été proposés avec une meilleure représentation des solutions. En particulier, une manière plus efficace de représenter les individus est d'utiliser un vecteur de n éléments, dans lequel le i ème élément est mis à 1 si le i ème élément appartient à l'ensemble d'éléments, et à 0, sinon.

Dans [31], les auteurs ont proposé l'algorithme G3APRM, qui utilise la G3P (programmation génétique guidée par la grammaire) afin que le processus de programmation génétique ne trouve pas d'individus non valides. G3PARM peut également traiter plusieurs types de données en utilisant une grammaire libre de contexte. Plus récemment, un algorithme génétique efficace appelé GAFIM a été proposé [32]. La principale innovation de GAFIM est la stratégie de suppression et de décomposition, qui divise les ensembles d'éléments peu

fréquents en ensembles d'éléments fréquents.

b. Approches fondées sur le renseignement d'essaim

L'optimisation par essaim particulaire a été largement appliquée pour résoudre le problème de la FIM.

Une approche d'optimisation par essaim particulaire a été proposée, dans laquelle l'espace de voisinage est exploré en déplaçant les points avant et arrière de chaque particule. Cet algorithme s'est avéré plus performant qu'un algorithme génétique, mais la recherche basée sur les points avant et arrière produit un grand nombre de voisinages, ce qui favorise l'intensification de la recherche au lieu de la diversité. Dans [36], un algorithme BPSO a été proposé pour extraire les K premiers ensembles fréquents, où une représentation binaire est utilisée pour accélérer le calcul de l'aptitude. L'algorithme est itératif. Vahid et autres chercheurs [37] ont proposé une approche multiobjective d'optimisation des essaims de particules pour l'extraction d'ensembles fréquents dans des données numériques. Cette approche permet de générer des modèles divers et informatifs décrivant des données numériques.

Plus récemment, l'optimisation par essaims d'abeilles (BSO) a été appliquée pour améliorer le processus d'extraction d'items fréquents. La zone de recherche des abeilles est d'abord déterminée, puis chaque région est explorée par une abeille pour trouver des ensembles fréquents. À chaque itération, les abeilles communiquent à l'aide d'une table de danse pour converger vers un bon ensemble d'éléments. [26]

4. Conclusion

Dans ce chapitre, on a explicité les principales propriétés du Data Mining et on a présenté un bref aperçu des principaux algorithmes d'extraction de motifs fréquents tels qu'Apriori, FP-Growth, Eclat. L'objectif de cette étude est de passer en revue les forces et les faiblesses des algorithmes fondamentaux de FIM. La comparaison a porté principalement sur les performances, c'est-à-dire le temps de réponse de ces algorithmes, les techniques et l'espace de mémoire nécessaire aux tâches d'extraction. La dernière section a été consacrée à la présentation des méthodes qui se basent sur les métaheuristiques pour FIM, et à partir de ces méthodes approchées on s'est inspirée pour notre approche.

CHAPITRE 3 : Approche Proposée

1. Introduction

L'approche proposée dans ce chapitre s'intègre dans la classe des méthodes approchées de FIM. En effet, en se basant sur une métaheuristique relativement récente appelé GWO, on propose en premier lieu FIM-BGOW. Ensuite, on propose une amélioration de la qualité des solutions de cet algorithme en utilisant l'apprentissage par renforcement et l'algorithme Q-Learning, l'algorithme résultat est appelé FIM-QL-BGWO.

2. Motivation attribuée à l'utilisation du GWO et le RL

En raison de l'espace de recherche étendu de l'extraction d'itemsets fréquents. Les chercheurs ont créé une variété de structures de données et de contrainte telle que la propriété d'Apriori pour réduire l'espace de recherche. Ces approches d'extraction d'itemsets sont inflexibles, néanmoins, l'espace de recherche croît exponentiellement à mesure que le nombre d'éléments augmente, ainsi que le temps d'exécution de ces algorithmes se détériore à mesure que l'ensemble de données s'agrandit. [36]

Compte tenu de leur succès récent, Le GWO et l'apprentissage par renforcement ont donné de bons résultats dans divers domaines d'optimisation, y compris la conception technique, l'extraction de donnée, et bien d'autres. Leur polyvalence et leur capacité à s'adapter à différents types de problèmes en font un choix attrayant pour les chercheurs. L'algorithme GWO peut rapidement localiser des régions prometteuses dans l'espace de recherche et converger vers des solutions optimales dans un nombre raisonnable d'itérations. L'apprentissage par renforcement qui est basé sur la psychologie comportementale, il est utilisé dans Itemsets Mining pour former un agent artificiel pour explorer les différentes actions, d'évaluer leurs conséquences et d'améliorer progressivement ses capacités décisionnelles.

Les qualités du l'algorithme GWO et du RL nous poussent pour les combiner dans notre projet afin d'améliorer les résultats en gardant un temps d'exécution acceptable.

3. GWO pour l'extraction d'itemsets fréquents

La version basique de GWO ne s'adapte pas à notre problème, une version binaire est plus adapté à résoudre le problème d'extraction de motifs fréquents.

3.1. Codification des individus (Wolves)

Chaque wolf représente une solution probable par conséquent il représente un itemset. Les itemsets sont codifiés en binaire dans un vecteur de taille N (nombre des items).

Par exemple :

itemset de taille 5 = {0,1,1,0,0}

Les datasets sont aussi converties en binaire afin de faciliter les calculs.

Transaction :

{1,4,7,11,13,16,19,22,25,28} ≡ {0,1,0,0,1,0,0,1,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1}

3.2. Fonction de fitness

La fonction fitness définit la fréquence d'apparition d'un itemset X dans toutes les transactions T de la base de données.

3.3. Espace de recherche

Dans le GWO l'espace de recherche est continue les itemsets peut se trouver dans n'importe quel position, Dans La version binaire de GWO (FIM-BGWO) l'espace de recherche est discret c'est-à-dire les positions des itemsets sont des (1, 0), par conséquent il représente notre dataset binaire. Pour aller de GWO au BGWO on passe par les étapes de binarisations suivantes :

Etape1 : Le FIM-BGWO adapte la même équation du GWO pour le calcul des distances entre les vecteurs de positions des itemsets Eq (1) dans le chapitre 1.

Etape2 : pour obtenir les vecteurs de positions des itemsets α, β, δ on utilise la fonction de transfert Sigmoid suivante [37]:

$$s_1^d = 1/(1 + e^{-10(A^d.D_\alpha^d-0.5)}) \quad (9)$$

$$s_2^d = 1/(1 + e^{-10(A^d.D_\beta^d-0.5)}) \quad (10)$$

$$s_3^d = 1/(1 + e^{-10(A^d.D_\delta^d-0.5)}) \quad (11)$$

Telque : d est la d'iem dimension d'un itemset.

$$bstep_1^d = \begin{cases} 1 & \text{si } (s_1^d \geq randn) \\ 0 & \text{sinon} \end{cases} \quad (12)$$

$$bstep_2^d = \begin{cases} 1 & \text{si } (s_2^d \geq randn) \\ 0 & \text{sinon} \end{cases} \quad (13)$$

$$bstep_3^d = \begin{cases} 1 & \text{si } (s_3^d \geq randn) \\ 0 & \text{sinon} \end{cases} \quad (14)$$

Telque : randn est une valeur aléatoire entre [0,1]

$$X_1^d = \begin{cases} 1 & \text{si } (X_\alpha^d + bstep_1^d \geq 1) \\ 0 & \text{sinon} \end{cases} \quad (15)$$

$$X_2^d = \begin{cases} 1 & \text{si } (X_\beta^d + bstep_2^d \geq 1) \\ 0 & \text{sinon} \end{cases} \quad (16)$$

$$X_3^d = \begin{cases} 1 & \text{si } (X_8^d + bstep_3^d \geq 1) \\ 0 & \text{sinon} \end{cases} \quad (17)$$

Etape3 : Pour mettre à jour les vecteurs de position des itemsets restants [37] :

$$X_i^d(nt) = \begin{cases} X_1^d & \text{si } (rand < 1/3) \\ X_2^d & \text{sinon si } (\frac{1}{3} \leq rand < 2/3) \\ X_3^d & \text{sinon} \end{cases} \quad (18)$$

Pseudo code de l'algorithme FIM-BGWO

Entrée

Dataset binaire (D),
 Taille de Population (M),
 Nombre maximale d'itérations (Max_IT),
 Seuil(Minsup) ;

Sortie

S : ensemble des itemset fréquents

Début

Initialisation des paramètres du BGWO ;
 Génération aléatoires des positions des vecteurs d'itemsets;
 Calcule de la valeur de support pour chaque itemset ;
 Trier par ordre décroissant les valeurs de support ;
 Trouver Alpha_itemset, Beta_itemset, Delta_itemset;

Pour i= 1 : Max_IT **Faire**

//Mettre à jour a

$a = 2 - 2 * i / Max_IT$

Pour j= 1 : M **Faire**

Mettre à jour les vecteurs A, C Eqs (3), (4)

Trouver les vecteurs de positions des itemsets X_1, X_2, X_3, X_i Eqs (1),(9-18)

Calculer la valeur de support de l'itemset (X_i)

Fin Pour ;

Recalculer les valeurs de support pour chaque itemset

Trier par ordre décroissant les valeurs de support

// Filtrer les itemsets par rapport a Minsup

Si (la valeur du support d'itemset > Minsup)

Retourner Alpha_itemset, Beta_itemset, Delta_itemset;

Fin Pour ;

FIN.

Afin d'améliorer la qualité des solutions (le nombre des itemsets trouvés), nous proposons une autre version de cette algorithme en le combinant avec l'algorithme Q-Learning.

4. Extraction d'itemsets fréquents basé sur la combinaison de GWO et le RL

On va présentée notre algorithme FIM-QL-BGWO qui est une combinaison de l'optimisation des loups gris et l'apprentissage par renforcement.

L'algorithme commence par initialiser une population des itemsets dans le dataset binaire. Où chaque itemset représente une solution candidate à notre problème d'optimisation. Au début de l'algorithme, les vecteurs positions des itemsets alpha, beta et delta sont définies comme les meilleures solutions initiales trouvées. À chaque itération, les positions des itemsets sont mises à jour en fonction de formules spécifiques déjà présenté dans le chapitre précédent.

Les itemset alpha, beta et delta jouent un rôle important dans cette mise à jour. Les positions des autres itemsets sont ajustées pour se rapprocher des positions des itemsets alpha, beta et delta. Après chaque mise à jour de position, les solutions candidates sont évaluées à l'aide d'une fonction d'évaluation appelé la fonction de support qui mesure leur qualité. Une fois que l'algorithme a terminé son exécution, les meilleures solutions trouvées, associée aux itemset alpha, beta, delta, sont considérées comme les solutions optimales à notre problème d'extraction de motif fréquent.

La méthode Q-Learning appliquée à l'algorithme FIM-BGWO, afin de contrôler l'exploration et l'exploitation à l'aide du Q-Table. Où un agent généré par ordinateur interagit avec un

certain environnement pour apprendre de manière adaptative la politique d'action optimale. L'agent est alors formé pour extraire un itemset d'un certain ensemble de données comme suit :

L'agent de renforcement utilise les valeurs Q-Table qui sont mis à jour à l'aide d'un tableau de récompenses comme guide pour les agents de recherche de l'algorithme métaheuristique pour décider entre les phases d'exploration ou d'exploitation étant les deux actions possibles. L'une des quatre options suivantes peut être choisie en fonction de la décision la plus appropriée [19] :

- (1) Pendant que l'agent est en état d'exploration, il décide de rester en exploration (action).
- (2) Pendant que l'agent est dans l'état d'exploitation, il décide de rester dans l'exploitation (action).
- (3) Pendant que l'agent est dans l'état d'exploration, il décide de transiter pour exploiter (action).
- (4) Pendant que l'agent est dans l'état d'exploitation, il décide de transiter pour explorer (action).

Dans cet algorithme hybride proposé (FIM-QL-BGWO), les deux tableaux (le tableau de récompense et le Q-Table) sont définis comme des matrices 2-dimensions.

La matrice du tableau de récompense (Reward table) contient la récompense positive (+1) ou négative (-1) pour chaque couple état /action.

Les valeurs dans la matrice Q-Table décrivent le résultat des choix qui sont pris par l'agent RL dans les épisodes précédents. La matrice Q-Table peut être considérée comme l'expérience d'un agent. Pendant la phase de découverte, un agent n'a aucune connaissance ou expérience de l'environnement auquel il appartient. Chaque agent acquiert de l'expérience en explorant l'environnement dans des épisodes/itérations spécifiques et met à jour la matrice Q-Table correspondante avec l'équation de Bellman. À l'aide de cette équation, l'agent évalue la probabilité de se rapprocher de la solution en évaluant chaque état suivant possible et chaque mesure (action) possible qui peut être prise à partir de cet état [19].

$$Q_{(t+1)}(s_t, a_t) = Q_t(s_t, a_t) + \alpha[r_{t+1} + \gamma \text{Max}Q_{t(s_{t+1}, a)} - Q_t(s_t, a_t)] \quad (19)$$

Où:

- s_t et s_{t+1} représentent les états courant et suivant relativement, a_t est l'action actuelle.

- α est la valeur du taux d'apprentissage et γ est le facteur d'actualisation. En outre, α et γ sont des nombres compris entre 0 et 1. Le α définit la vitesse à laquelle notre algorithme devrait apprendre tandis que γ définit combien notre algorithme apprend de ses erreurs.

En effet, le taux d'apprentissage détermine dans quelle mesure l'information nouvellement acquise supplante l'information ancienne et est utilisée pour contrôler la convergence du processus d'apprentissage. D'autre part, le facteur de remise détermine l'importance des récompenses futures. Le facteur de remise est une constante qui représente la façon dont l'agent prêtera attention aux récompenses dans un avenir lointain. Il contrôle l'importance des récompenses à long terme [19].

$-r_{t+1}$ est la récompense immédiate accordée à un agent ou le montant qu'il est pénalisé pour l'action en cours.

$-Q_{(t+1)}(s_t, a_t)$ est la valeur Q pré estimée pour l'état suivantes s_{t+1} .

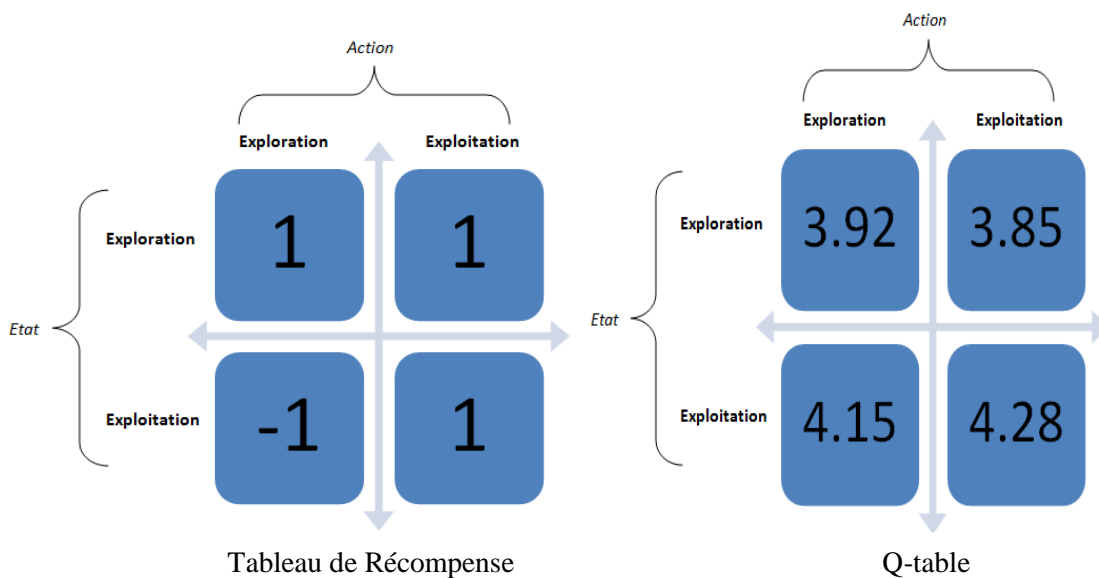


Figure 8:Exemple du tableau Q et du tableau de récompenses [19].

Le paramètre $|A|$ contrôle à la fois les phases d'exploration et d'exploitation dans FIM-QL-BGWO algorithme. De plus, le Q-Table est également inclus avec $|A|$, dans le compte et la décision d'exploration ou d'exploitation la plus appropriée est prise selon les valeurs de ce tableau. ces dernières sont comparées pour mettre à jour la position de vecteur de chaque itemset dans l'algorithme métaheuristique utilisé(GWO).

En outre, le paramètre $|A|$ vérifie les phases et les contrôles de valeur Q et permet

ensuite de prendre la décision la plus appropriée pour l'exploration et l'exploitation. Lorsque la décision la plus appropriée est prise, elle s'applique à la fonction support. Tout ce processus se poursuit jusqu'à ce qu'il trouve les meilleures solutions possibles ou qu'il atteigne la fin des itérations définies. Dans cette étape, la nouvelle valeur de la fonction support suit deux mécanismes qui sont calculés à l'aide des équations respectives [19]:

$$\vec{X}_l(nt) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (20)$$

$$\vec{X}_l(nt) = \frac{\vec{X}_1 * \sigma_1 + \vec{X}_2 * \sigma_2 + \vec{X}_3 * \sigma_3}{3} \quad (21)$$

Le schéma suivant représente la vue générale de l'approche dont nous avons proposé :

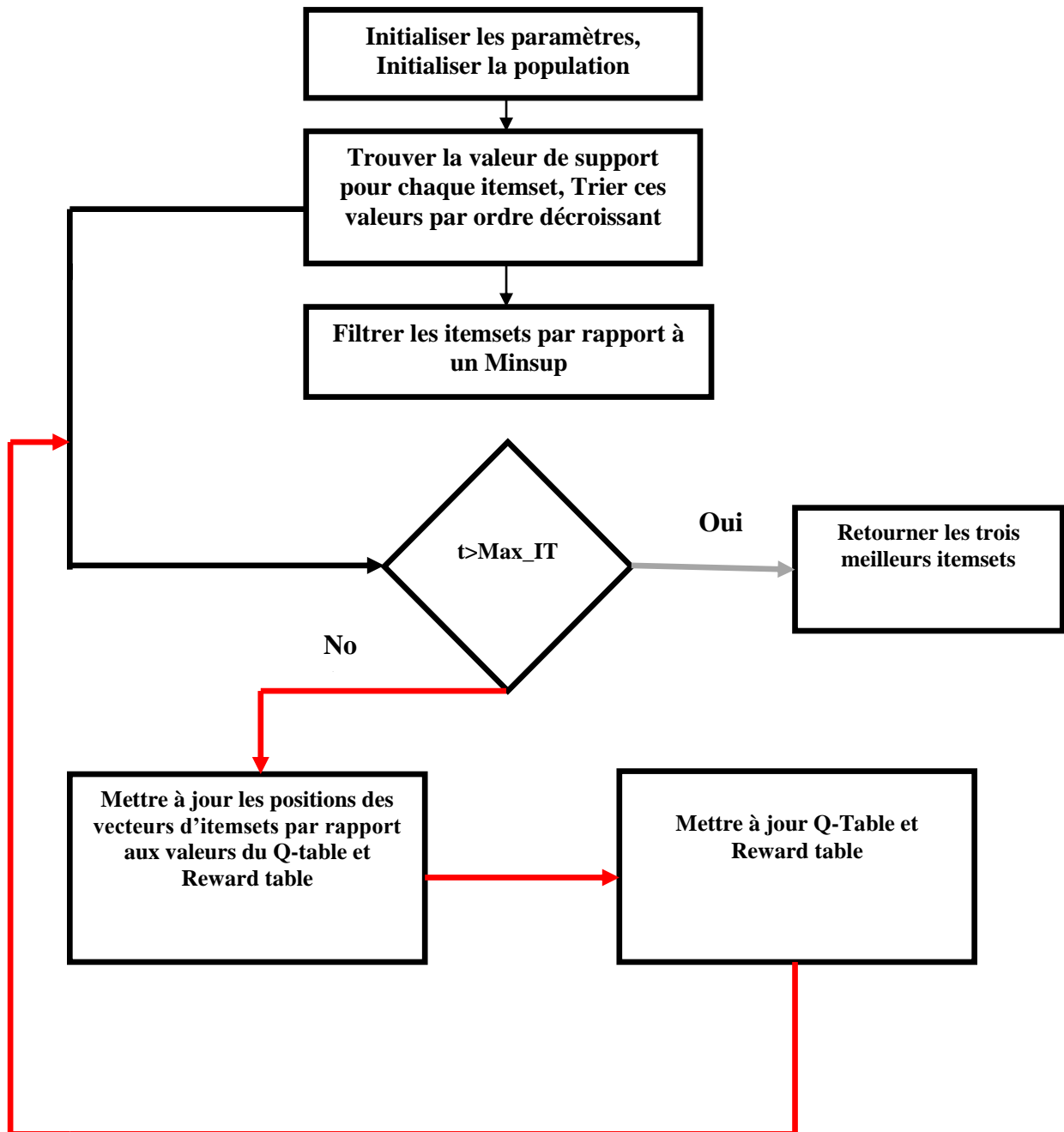


Figure 9 : Organigramme d'extraction d'itemsets fréquents à partir de données basé sur la combinaison de l'apprentissage par renforcement et GWO.

Pseudo algorithme

L'algorithme proposé nommé **FIM-QL-BGWO** est présenté comme suit :

Algorithme: Extraction de motifs fréquents basé sur la combinaison de l'optimisation de loup gris (GWO) et l'apprentissage par renforcement (**FIM-QL-BGWO**)

Entrés

Dataset binaire (D),
Taille de Population (M),
Nombre maximale d'itérations (Max_IT),
Taux d'apprentissage (alpha),
Taux de remise (gamma) ;
Seuil(Minsup) ;

Sortie

S : ensemble des itemset fréquents

Début

Initialisation des paramètres du BGWO ;
Initialisation du Q-Table et Reward Table pour chaque itemset ;
Génération aléatoires des positions des vecteurs d'itemsets;
Calcul de la valeur de support pour chaque itemset ;
Trier par ordre décroissant les valeurs de support ;
Trouver Alpha_itemset, Beta_itemset, Delta_itemset;

Pour i= 1 : Max_IT **Faire**

//Mettre à jour a

$$a = 2 - 2 * i / Max_IT$$

Pour j= 1 : M **Faire**

Mettre à jour Q-Table avec l'équation (19)

// Mettre à jour A, C

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}$$

$$\vec{C} = 2\vec{r}_2$$

//Trouver les vecteurs de positions des itemsets X_1, X_2, X_3

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A} \cdot \vec{D}_\alpha$$

$$\vec{X}_2 = \vec{X}_\beta - \vec{A} \cdot \vec{D}_\beta$$

$$\vec{X}_3 = \vec{X}_\delta - \vec{A} \cdot \vec{D}_\delta$$

Si $|A| \geq 1$

Si (valeur de l'exploration > valeur de l'exploitation)

Calculer la nouvelle valeur de support avec l'équation (20)

Si (la nouvelle valeur de support > la valeur actuelle de support)

Récompense = 1

Sinon

Récompense = -1

Fin Si ;

Sinon

Calculer la nouvelle valeur de support avec l'équation (21)

Si (la nouvelle valeur de support > la valeur actuelle de support)

Récompense = 1

Sinon

Récompense = -1

Fin Si ;

Fin Si ;

Sinon

Si (valeur de l'exploration > valeur de l'exploitation)

Calculer la nouvelle valeur de support avec l'équation (20)

Si (la nouvelle valeur de support > la valeur actuelle de support)

Récompense = 1

Sinon

Récompense = -1

Fin Si ;

Sinon

Calculer la nouvelle valeur de support avec l'équation (21)

Si (la nouvelle valeur de support > la valeur actuelle de support)

Récompense = 1

Sinon

Récompense = -1

Fin Si ;

Fin Si ;

Fin Pour ;

```
Recalculer les valeurs de support pour chaque itemset  
Trier par ordre décroissant les valeurs de support  
// Filtrer les itemsets par rapport a Minsup  
Si (la valeur du support d'itemset > Minsup)  
    Retourner Alpha_itemset, Beta_itemset, Delta_itemset;
```

Fin Pour ;

FIN.

5. Conclusion :

Dans ce chapitre, nous avons présenté notre approche pour extraire des itemsets fréquents basés sur la combinaison de GWO et l'apprentissage par renforcement, nous avons également présenté les mesures à prendre pour renforcer le GWO avec le Q-Learning. Le chapitre suivant est consacré à la démonstration de l'efficacité de notre méthode par une étude expérimentale détaillée sur divers ensembles de données.

CHAPITRE 4 : Tests et Validation

1. Introduction

Nous parlerons dans ce chapitre de la section expérimentale de notre approche proposée. L'environnement de travail et les outils que nous avons utilisés seront présentés dans la première section. Ensuite, nous présenterons notre ensemble de tests, puis nous montrerons les résultats que nous avons générés.

2. Environnement de développement

2.1. Environnement matériel

Ordinateur portable avec les caractéristiques suivantes :

- CPU : Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.71 GHz
- Mémoire RAM : 8 GO
- Système d'exploitation : Windows 10 Professionnel 64 bits

2.2. Environnement logiciel

Python :

Python est un langage de programmation open source **créé par le programmeur Guido van Rossum en 1991**. Il s'agit d'un **langage de programmation interprété**, qui ne nécessite donc pas d'être compilé pour fonctionner. Un programme interpréteur permet d'exécuter le code Python sur n'importe quel ordinateur. Ceci permet de voir rapidement les résultats d'un changement dans le code. En revanche, ceci rend ce langage plus lent qu'un langage compilé comme le C. En tant que **langage de programmation de haut niveau**, Python permet aux programmeurs de se focaliser sur ce qu'ils font plutôt que sur la façon dont ils le font. Ainsi, écrire des programmes prend moins de temps que dans un autre langage. Python est simple à comprendre et à utiliser. Il s'agit d'un langage de programmation très développé dont la syntaxe est proche de l'anglais. Il s'agit d'un langage interprété, ce qui signifie que le code est implémenté ligne par ligne.

3. Datasets utilisé

Nous avons utilisé dans nos tests des données réelles et semi-synthétiques pour évaluer l'approche proposée. Le descriptif de ses datasets est représenté dans le tableau suivant :

Tableau 4 : Représentations des datasets utilisés

Dataset	Longueur de la transaction	Nombre d'items	Nombre des transactions	Densité
RecordLink	10	29	574913	34.48%

Connect	43	129	67557	33.33%
Chess	37	75	3196	49.33%
Mushrooms	23	119	8416	19.33%

Ces datasets ont été obtenues de la bibliothèque SPMF Data Mining (Fournier-Viger 2016)[38]. Afin d'utiliser la version binaires du Grey Wolf Optimization pour l'extraction des itemsets fréquents (FIM-BGWO), nous avons transformé ces datasets en binaires tel que on trouve '1' si l'item apparaisse dans la transaction sinon on trouve '0' . Voici quelques exemples extraient du dataset RecordLink :

Transaction1 :

{1,5,7,11,13,16,19,22,24,28} \equiv {0,1,0,0,0,1,0,1,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,1,0,0,0,1}

Transaction 2 :

{1,4,7,11,13,16,19,22,25,28} \equiv {0,1,0,0,1,0,0,1,0,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1}

4. Tests et résultats

Les tests de notre approche proposée on était fait avec les paramètres démontrer dans le tableau suivant :

Tableau 5 : Représentation des paramètres utilisés dans les tests

Paramètres	Valeur
Nombre de la population	30
Nombre d'exécution	5
Nombre d'itérations	200

4.1 Tests d'évaluation du temps d'exécution

Dans ce cas , nous avons comparer le temps d'exécution consommé pour l'extraction des

itemsets fréquents à partir des données certaines pour l’algorithme exacte Apriori ,la version basique binaire de la méthode approchée (FIM-BGWO) ,et aussi avec la méthode de l’approche proposée (FIM-QL -BGWO) pour chacune des datasets cités au paravent .

Tableau 6 : le temps d'exécution (en seconde) trouvés pour le dataset RecordLink en comparaison avec différentes méthodes et en utilisant différents minsup

Minsupport	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Apriori	23.542	13.013	8.623	3.906	2.140	1.515	0.937	0.547	0.516
FIM-BGWO	107	107	107	107	107	107	107	107	107
FIM-QL-BGWO	151	151	151	151	151	151	151	151	151

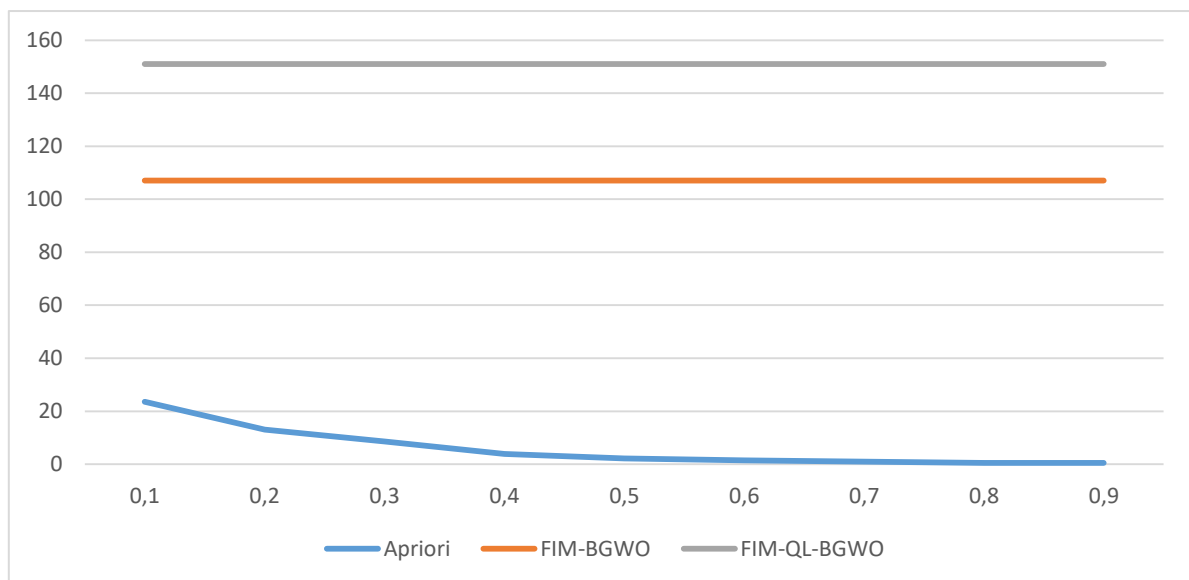


Figure 10 : Temps d'exécution trouvés pour le dataset RecordLink en comparaison différentes méthodes et en utilisant différents minsup.

Comme démontré dans la **Figure 10**, le temps d’exécution de FIM-QL-BGWO et de FIM-BGWO est stable par rapport au celle de Apriori, on marque un temps d’exécution élevé pour FIM-QL-BGWO, FIM-BGWO que la méthode exacte Apriori, qui est un inconvénient

pour notre approche proposée FIM-QL-BGWO.

Tableau 7 : le temps d'exécution (en seconde) trouvés pour le dataset Connect en comparaison avec différentes méthodes et en utilisant différents minsup

Min support	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Apriori	Plus 4 heures	Plus 4 heures	Plus 4 heures	Plus 4 heures	Plus 4 heures	Plus 4 heures	Plus 4 heures	1619.811	94.204
FIM-BGWO	457	457	457	457	457	457	457	457	457
FIM-QL-BGWO	95	95	95	95	95	95	95	95	95

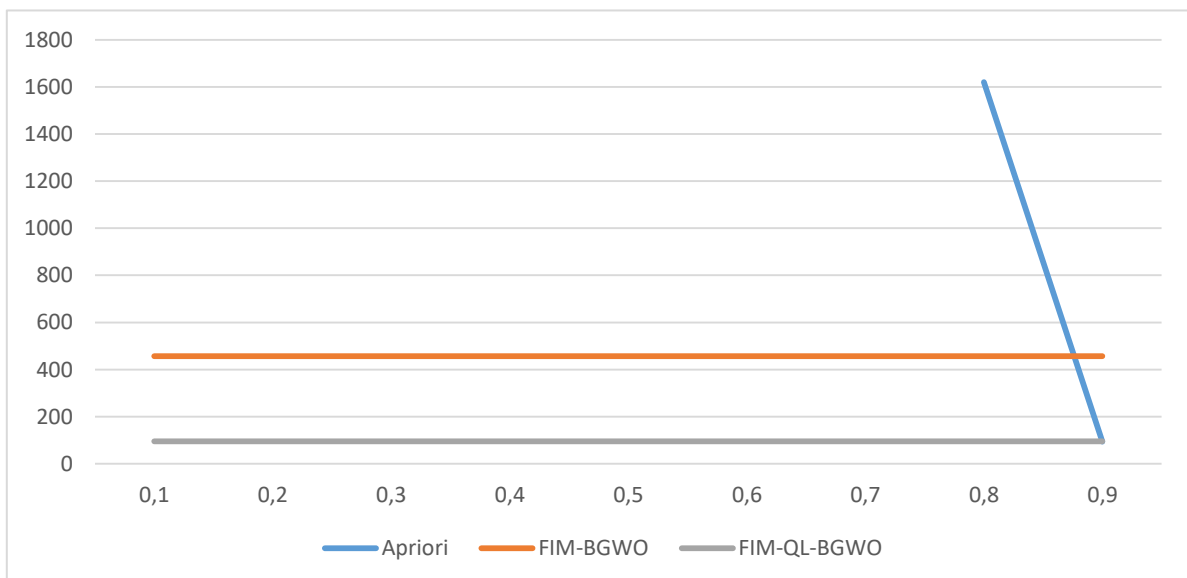


Figure 11 : Temps d'exécution trouvés pour le dataset Connect en comparaison avec différentes méthodes et en utilisant différents minsup

Comme démontré dans la **Figure 11**, le temps d'exécution de FIM-QL-BGWO et de FIM-BGWO est stable par rapport au celle de Apriori ,de plus notre approche proposée FIM-QL-BGWO marque un temps d'exécution beaucoup mieux que la méthode basique FIM-BGWO et la méthode exacte Apriori qui prend plus que 4 heures pour les seuils de 0.1 jusqu'à 0.7.

Tableau 8: le temps d'execution(en seconde) trouvés pour le dataset Mushrooms en comparaison avec différentes méthodes et en utilisant différents minsup

Min support	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Apriori	139.868	7.405	0.75	0.094	0.047	0.031	0.047	0.047	0.062
FIM-BGWO	27	27	27	27	27	27	27	27	27
FIM-QL-BGWO	8	8	8	8	8	8	8	8	8

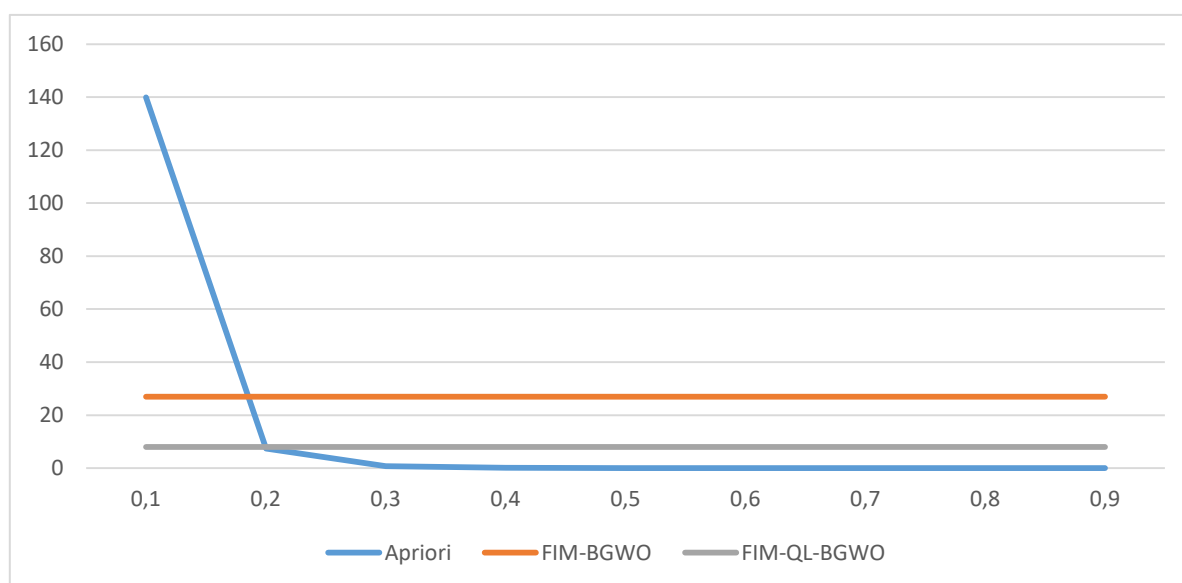


Figure 12 : Temps d'exécution trouvés pour le dataset Mushrooms en comparaison avec différentes méthodes et en utilisant différents minsup

Comme démontré dans la **Figure 12**, le temps d'exécution de FIM-QL-BGWO et de FIM-BGWO est stable par rapport au celle de Apriori, on marque un temps d'exécution élevé pour la méthode basique FIM-BGWO et un temps d'exécution acceptable pour notre approche proposée FIM-QL-BGWO en comparaison avec la méthode exacte Apriori.

Tableau 9 : le temps d'exécution (en seconde) trouvés pour le dataset Chess en comparaison avec les différentes méthodes et en utilisant différents minsup

Min support	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Apriori	Plus 4 heures	Plus 4 heures	Plus 4 heures	Plus 4 heures	924	56	8	2	0.075
FIM-BGWO	5	5	5	5	5	5	5	5	5
FIM-QL-BGWO	5	5	5	5	5	5	5	5	5

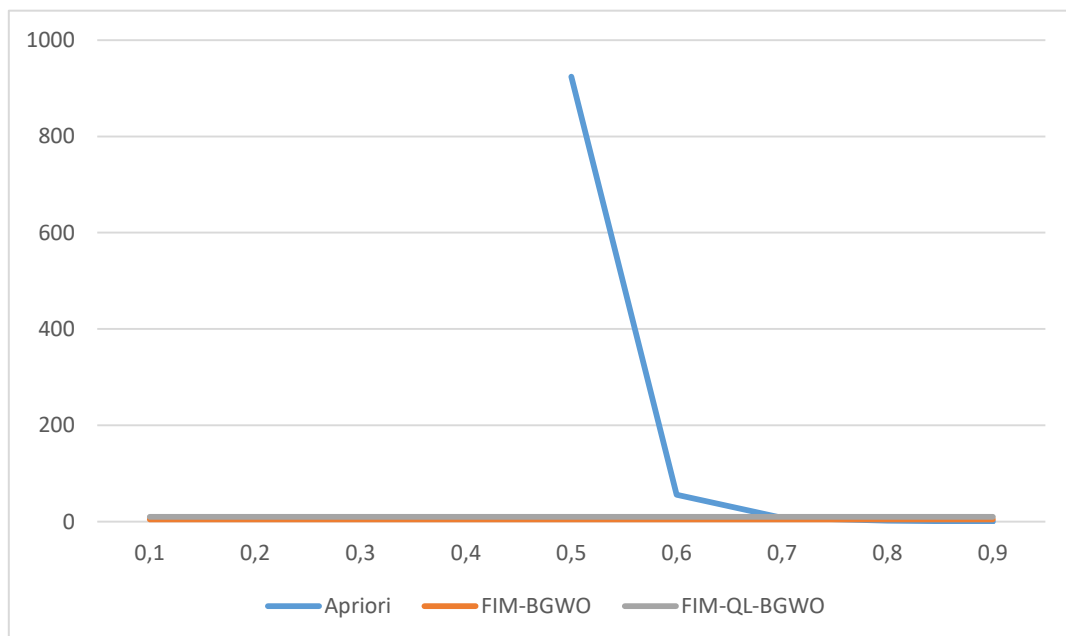


Figure 13 : Temps d'exécution trouvés pour le dataset Chess en comparaison avec différentes méthodes et en utilisant différents minsup

Comme démontré dans la **Figure 13**, le temps d'exécution de FIM-QL-BGWO et de FIM-BGWO est stable par rapport à celle de Apriori, de plus notre approche proposée FIM-QL-BGWO marque un temps d'exécution beaucoup mieux que la méthode exacte Apriori qui prend plus que 4 heures pour les seuils de 0.1 jusqu'à 0.4.

Conclusion : On remarque que notre approche proposée FIM-QL -BGWO performe bien de manière générale par rapport à la méthode basique FIM-BGWO et d'une manière acceptable par rapport à la méthode exacte Apriori qui prend dans certains datasets plus que 4 heures en terme de temps d'exécution.

4.2 Tests de la qualité

Dans ce cas, nous avons comparé les itemsets fréquents trouvés pour l'algorithme exacte Apriori, la version basique binaire de la méthode approchée (FIM-BGWO), et aussi avec la méthode de l'approche proposée (FIM-QL -BGWO) pour chacune des datasets cités au paravent.

Tableau 10 : Pourcentage d'itemsets fréquents trouvés dans le dataset RecordLink en comparaison avec différentes méthodes.

Dataset	Apriori	FIM-BGWO	FIM- QL -BGWO
RecordLink	100%	40%	81%

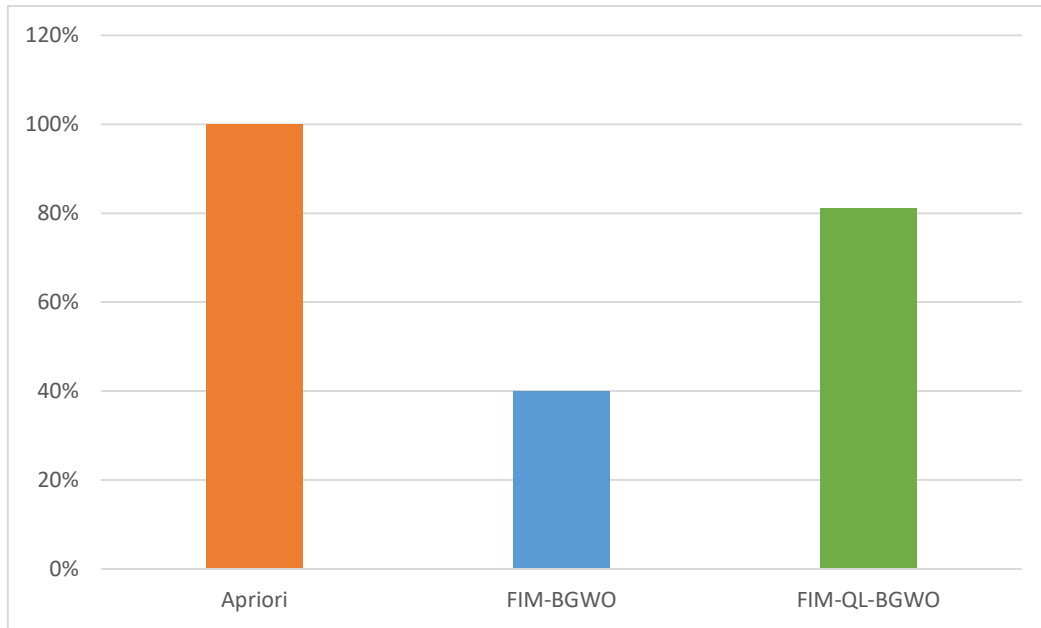


Figure 14 : Pourcentage d'itemsets fréquents trouvés dans le dataset RecordLink en comparaison avec différentes méthodes.

D'après la **Figure 14**, on remarque que le Pourcentage d'itemsets fréquents trouvés est intéressantes pour notre approche proposée FIM- QL –BGWO en comparaison avec la méthode basique FIM-BGWO, et proche au résultat de la méthode exacte Apriori.

Tableau 11 : Pourcentage d'itemsets fréquents trouvés dans le dataset Chess en comparaison avec différentes méthodes

Dataset	Apriori	FIM-BGWO	FIM- QL -BGWO
Chess	100%	21%	48%

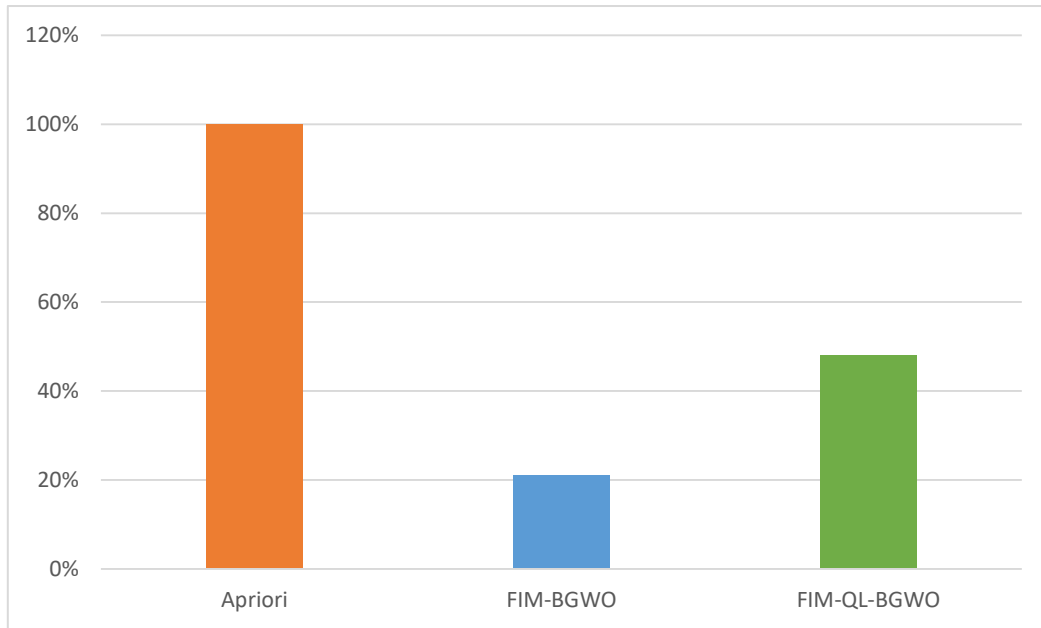


Figure 15 : Pourcentage d'itemsets fréquents trouvés dans le dataset Chess en comparaison avec différentes méthodes

D'après la **Figure 15**, on remarque que le Pourcentage d'itemsets fréquents trouvés est intéressantes pour notre approche proposée FIM- QL –BGWO en comparaison avec la méthode basique FIM-BGWO, et aussi bon par rapport au résultat de la méthode exacte Apriori.

Tableau 12 : Pourcentage d'itemsets fréquents trouvés dans le dataset Mushrooms en comparaison avec différentes méthodes

Dataset	Apriori	FIM-BGWO	FIM- QL -BGWO
Mushrooms	100%	26%	76%

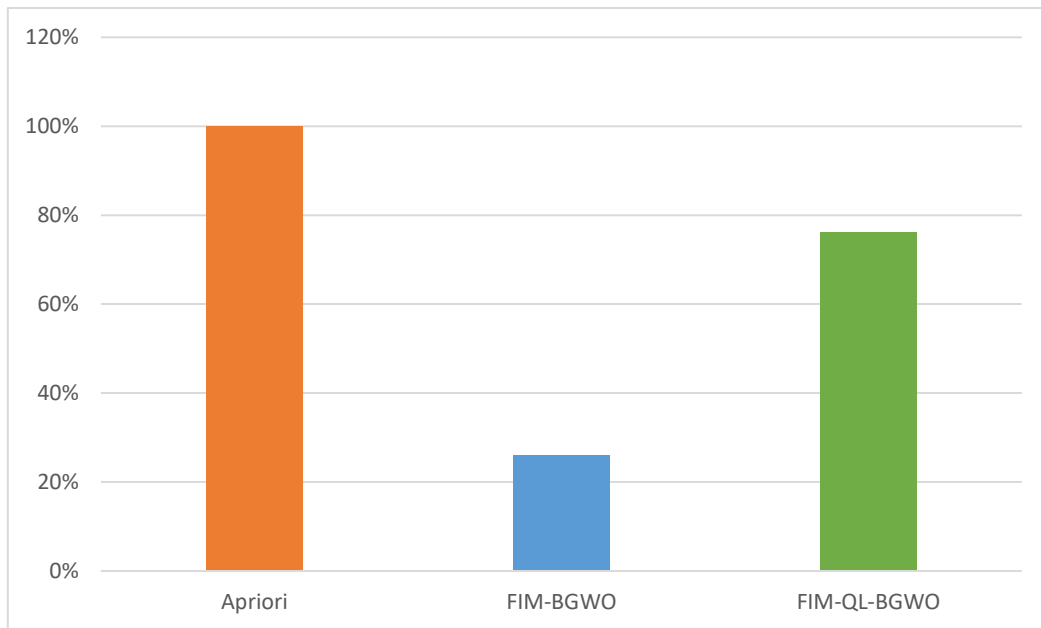


Figure 16 : Pourcentage d'itemsets fréquents trouvés dans le dataset Mushrooms en comparaison avec différentes méthodes.

D'après la **Figure 16**, on remarque que le Pourcentage d'itemsets fréquents trouvés est très élevé pour notre approche proposée FIM- QL –BGWO en comparaison avec la méthode basique FIM-BGWO, de plus il aussi proche au résultat de la méthode exacte Apriori.

Tableau 13 : Pourcentage d'itemsets fréquents trouvés dans le dataset Connect en comparaison avec différentes méthodes

Dataset	Apriori	FIM-BGWO	FIM- QL -BGWO
Connect	100%	11%	26%

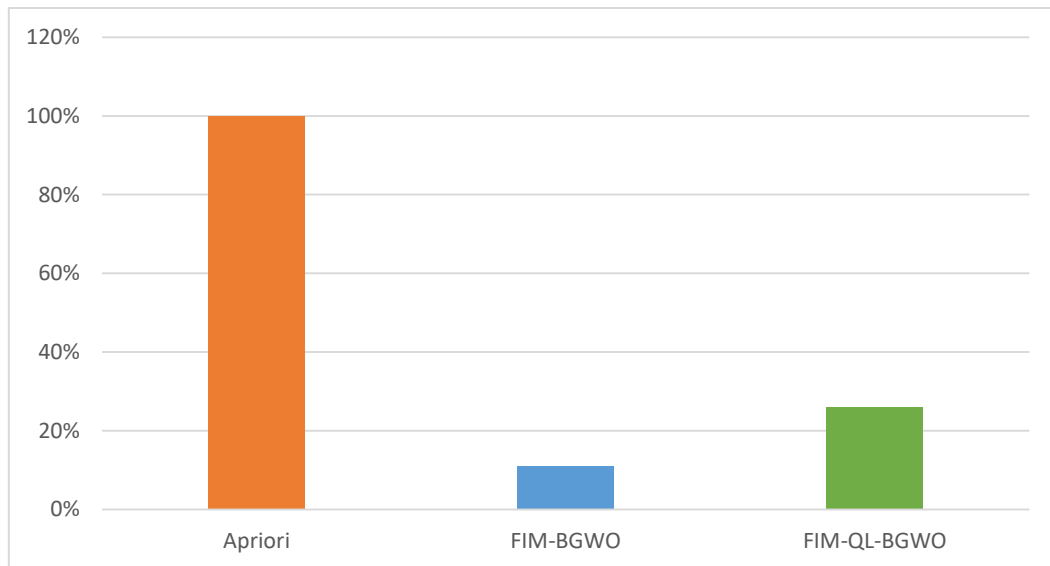


Figure 17 : Pourcentage d'itemsets fréquents trouvés dans le dataset Connect en comparaison avec différentes méthodes.

D'après la **Figure 17**, on remarque que le Pourcentage d'itemsets fréquents trouvés est bon pour notre approche proposée FIM- QL –BGWO en comparaison avec la méthode basique FIM-BGWO, mais très loin par rapport au résultat de la méthode exacte Apriori

Conclusion : On constate que notre approche proposée FIM-QL -BGWO travail mieux que la méthode basique FIM-BGWO avec une amélioration de 20% en terme de qualité, en outre cette amélioration est proche aux résultats de la méthode exacte Apriori.

5. Conclusion

Dans ce chapitre nous avons présenté l'environnement matériel et l'environnement logiciel, les différents datasets utilisés pour implémenté notre approche. On a aussi comparé les résultats dans le cas d'évaluation de temps d'exécution et dans le cas de qualité avec les différentes méthodes à savoir Apriori, FIM-BGWO, FIM-QL-BGWO.

CONCLUSION GENERALE

Dans ce travail nous avons traité le problème de l'extraction de motifs (itemsets) fréquents à partir de données précises. Cette tâche est présente dans de nombreux domaines d'application, et ça reste toujours l'un des sujets d'actualité pour les chercheurs car les domaines et applications qui lui font appel ne cessent de s'accroître.

Etant un problème NP-Difficile, l'extraction de motifs fréquents a été traitée comme un problème d'optimisation. En effet, notre travail s'intègre dans ce contexte. Afin de comprendre les principes de base de l'optimisation combinatoire, le premier chapitre était consacré en partie à introduire les méthodes de résolution des problèmes d'optimisation, en particulier les métaheuristiques tout en se focalisant sur l'optimisation des lous gris. Cette dernière métaheuristique fait partie de l'approche proposée et par conséquent, elle était présentée en détails.

La deuxième partie de ce chapitre présente les principes de base de l'apprentissage automatique, en particulier l'apprentissage par renforcement qui fait à son tour partie de l'approche proposée.

Une étude générale sur les algorithmes d'extraction de motifs fréquents était faite dans le deuxième chapitre. Les algorithmes présentés ont été classifiés en deux catégories générales. Celle des algorithmes exactes et celle des méthodes approchées qui se base en général sur les différentes métaheuristiques bioinspirées.

Les méthodes approchées d'extraction de motifs fréquents souffrent de problème de qualité des solutions (nombre des itemsets fréquents trouvés). L'approche proposée présentée dans le troisième chapitre tente à résoudre ce problème en utilisant une combinaison de l'optimisation des lous gris avec l'apprentissage par renforcement pour mieux calibrer cette métaheuristique et basculer entre l'exploration et l'exploitation d'une façon efficace afin d'améliorer la qualité des solutions et éviter le problème de l'optimum local.

Perspectives

Les perspectives de ce travail sont les suivantes :

- Optimiser les algorithmes utilisés afin de réduire le temps d'exécution, plus précisément, en utilisant des datasets large qui contiennent un grand nombre d'items et

de transactions.

- Améliorer la qualité des résultats et faire en sorte qu'elle soit plus précise durant l'extraction des motifs fréquents.
- Mettre en place une interface graphique afin que notre méthode soit utilisable auprès d'une plus large communauté de chercheurs.

Bibliographie

- [1] C.KARAALI , Z.SELLAM , Etude comparative entre les algorithmes Méta-heuristique , Université Saad DAHLAB BLIDA-1.master 2022.
- [2] Marie-Eleonore Marmion, Recherche locale et optimisation combinatoire de l'analyse structurelle d'un problème à la conception d'algorithmes efficaces : Mathématique discrète 2011.
- [3] Cooren, Y. Perfectionnement d'un algorithme adaptatif d'optimisation par Essaim Particulaires Applications en génie médical et en électronique. Thèse de doctorat, Université de Paris 12 Val de Marne, Paris, 2008.
- [4] N.HAMRI , Approche basée méta-heuristique pour la sélection des services d'adaptation de documents multimédia , Université L'arbi Ben M'hidi , master 2019.
- [5] N. BOUAFIA , Optimisation par colonies d'abeilles pour l'extraction des itemsets fréquents à partir de données évidentielles , Université Saad Dahlab Blida 1, master 2019.
- [6] S .Mirjalili , S-M. Mirjalili , A. Lewis, “ Grey Wolf Optimizer”, in : Advanced in Engineering Software 2014.
- [7] S.FARES , A.SERDOUK , Optimisation des paramètres de la commande prédictive généralisée par les méthodes méta-heuristique , Université Mohamed Larbi Ben M'hidi, master 2022.
- [8]. Z Teng , J-L Lv , and Li-wen Guo , “An improved hybrid grey wolf optimization algorithm”, 2018.
- [9] “Summary for Policymakers,” in Cambridge University Press eBooks, 2014, pp. 1–30. doi: 10.1017/cbo9781107415324.004.
- [10] Bishop, C.M.: Pattern Recognition and Machine Learning (2006).

- [11] Kotsiantis, S.B.: Supervised machine learning: a review of classification techniques. In: Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in EHealth, HCI, Information Retrieval and Pervasive Technologies, pp. 3–24. IOS Press, NLD (2007).
- [12] Celebi, M.E., Aydin, K. (eds.): Unsupervised Learning Algorithms. Springer, Cham (2016).
- [13] Sutton, R.S., Barto, A.G.: Reinforcement learning: an introduction 2018. Technical report (2017).
- [14] J.G. Greener, S.M. Kandathil, L. Moffat, D.T. Jones: A guide to machine learning for biologists. Nat Rev Mol Cell Biol, pp.40-55,(2022)
- [15] J. M. Carew, « reinforcement learning » , Enterprise AI, févr. 2023, [En ligne]. Disponible sur : <https://www.techtarget.com/searchenterpriseai/definition/reinforcement-learning>.
- [16] Khadija El Bouchefry PhD, Rafael S. de Souza PhD: in Knowledge Discovery in Big Data from Astronomy and Earth Observation, 2020.
- [17] OULD BABAALI Rania , ELY BOUCELHAB Khadijetou, Apprentissage par renforcement pour l’optimisation de la consommation d’énergie dans un contexte IoT (Thème de Master). Université Saad Dahleb de Blida,2019.
- [18] MOULOUD Chaima, ELAMRANI Fatima Zahra , Reinforcement Learning Based Uncertain Pattern Mining (Thème de Master). Université Saad Dahleb de Blida,2022.
- [19] Amir Seyyedabbasi , Royal Aliyev , Farzad Kiani , Murat Ugur Gulle , Hasan Basyildiz , Mohammed Ahmed Shah, in Knowledge-Based Systems :Hybrid algorithms based on combining reinforcement learning and metaheuristic methods to solve global optimization problems, 2021.

- [20] (2012). Q-Learning. In: Seel, N.M. (eds) Encyclopedia of the Sciences of Learning. Springer, Boston, MA.
- [21] B. L, “Data Mining : qu’est ce que l’exploration de données ?,” LEBIGDATA.FR, May 2022, [Online]. Available: <https://www.lebigdata.fr/data-mining-definition-exemples>.
- [22] P. S. Pramod and O. P. Vyas, “Survey on Frequent Itemset Mining Algorithms,” *International Journal of Computer Applications*, vol. 1, no. 15, pp. 94–100, Feb. 2010, doi: <https://doi.org/10.5120/316-484>.
- [23]. J. M. Luna, P. Fournier- Viger, and S. Ventura, “Frequent itemset mining: A 25 years review,” *WIREs Data Mining and Knowledge Discovery*, vol. 9, no. 6, Jul. 2019, doi: <https://doi.org/10.1002/widm.1329>.
- [24] N.Pasquier ,Extraction de bases pour les règles d'association à partir des itemsets fermés fréquents,2000.
- [25] A.BECHIR, Resolution des probleme d’optimisation par les systemes multi-agents et les approches évolutionnaire .These de doctorat.Université Mohamed Khider-Biskra.2016 .
- [26] A.M.Mammeri ,S.Rekia, Extraction ces motifs a partir des données incertaines en utilisant l’optimisation de Grey Wolf, université saad dahleb 1 , master 2022.
- [27] AgrawalR, SrikantR. Fast algorithms for mining association rules. In: Paper presented at the proceedings of the 20th international conference on very large data bases, Santiago, 1994.
- [28] Z.Daaou , A.KERIOUI , Extractions des motifs fréquents orientés besoins du décideur , Université Saad Dahlab Blida , master 2019.
- [29] Javeed,MZ. Scalable algorithms for association mining. *IEEE Trans Knowl Data Eng* 12(3):372–390, 2000.

- [30] Mata J., Alvarez J., Riquelme J.: An evolutionary algorithm to discover numeric association rules. In: Proceedings of the ACM symposium on Applied computing SAC, pp. 590–594 (2002)
- [31] Romero, C., Zafra, A., Luna, J.M., Ventura, S.: Association rule mining using genetic programming to provide feedback to instructors from multiple-choice quiz data. *Expert. Syst.* 30(2),162–172 (2013).
- [32]. Djenouri, Y., Bendjoudi, A., Nouali-Taboudjemat, N.: Association rules mining using evolutionary algorithms. In: The 9th International Conference on Bio-inspired Computing: Theories and Applications (BIC-TA 2014). LNCS (2014).
- [33] . Martinez-Ballesteros, M., Bacardit, J., Troncoso, A., Riquelme, J.C.: Enhancing the scalability of a genetic algorithm to discover quantitative association rules in large-scale datasets. *Integr.Comput.-Aided Eng.* 22(1), 21–39 (2015).
- [34] . Sarath, K.N.V.D., Ravi, V.: Association rule mining using binary particle swarm optimization. *Eng. Appl. Artif. Intell.* 26(8), 1832–1840 (2013).
- [35] . Beiranvand, V., Mobasher-Kashani, M., Bakar, A.A.: Multi-objective PSO algorithm for mining numerical association rules without a priori discretization. *Expert. Syst. Appl.* 41(9), 4259–4273 (2014).
- [36]. .Sarath, K.N.V.D., Ravi, V.: Association rule mining using binary particle swarm optimization. *Eng. Appl. Artif. Intell.* 26(8), 1832–1840 (2013)
- [37]. P. Hu, J. Pan and S. Chu, Improved Binary Grey Wolf Optimizer and Its application for feature selection, *Knowledge-Based Systems* (2020).
- [38]. Youcef Djenouri , Marco Comuzzi “Combining Apriori heuristic and bio-inspired algorithms for solving the frequent itemsets mining problem “’,2017.
- [39]. M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah, “Metaheuristic Algorithms: A Comprehensive Review,” *Computational Intelligence for Multimedia Big Data on the Cloud*

with Engineering Applications, pp. 185–231, 2018, doi: <https://doi.org/10.1016/b978-0-12-813314-9.00010-4>.