N°d'ordre: .....

République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la Recherche scientifique

Université Djillali Liabès de Sidi Bel Abbès
Faculté des Sciences Exactes
Département d'informatique
Laboratoire EEDIS

Thèse de Doctorat en Sciences

Filière : Informatique
Spécialité : Informatique

Par

Alaoui Abdiya

Hybridation des métaheuristiques dans le processus d'extraction de connaissances à partir de données

Soutenue le 21/01/2021 devant le jury :

Pr.	Boukli Hacene Sofiane	UDL SBA	Président du jury
Pr.	Sekhri Larbi	Université Oran 1 Oran	Examinateur
Dr.	Tlemsani Radouane	USTO-MB Oran	Examinateur
Pr.	Elberrichi Zakaria	UDL SBA	Directeur de thèse

Année Universitaire : 2020 - 2021

À *mes parents*
À *mes sœurs et frères*
À *mes amis(es)*
À *tous ceux qui m'ont encouragé et aidé de près ou de loin . . .*

Remerciements

Je remercie ALLAH le tout puissant pour toute la volonté, le courage et la patience qui m'a donné pour étudier et pour suivre le chemin de la science.

Je tiens à exprimer toute ma reconnaissance à mes parents qui m'ont soutenu tout au long de ces années et m'ont toujours encouragé à réaliser mes études.

Je souhaite offrir mes plus grands remerciements à mon directeur de recherche, Mr Zakaria Elberrichi. Son aide, son grand soutien et ses précieux conseils ont été très importants tout au long de ma recherche. Grâce à son expérience et sa disponibilité, il a été un guide hors pair dans la complétion de ma recherche. Merci infiniment.

Je présente mes respects et mes remerciements les plus sincères aux membres du jury qui m'ont fait l'honneur d'évaluer ma thèse, je voudrais remercier Mr Boukli Hacene Sofiane, Maître de Conférence A à l'Université Djillali Liabès de Sidi Bel Abbès qui m'a fait l'honneur de présider le jury. J'associe à ces remerciements Mr Sekhri Larbi Professeur à l'Université Oran 1 d'Oran et Mr Tlemsani Radouane Maître de Conférence A à l'Université des Sciences et de la Technologie d'Oran Mohamed-Boudiaf USTO-MB pour l'intérêt qu'ils ont porté à mes recherches en acceptant d'examiner et de juger mon travail.

Je désire remercier aussi, Mme Aicha enseignante à l'université de Tiaret pour leurs soutiens et encouragements. Plus largement, je remercie tous mes enseignants qui ont contribué de près ou de loin à l'aboutissement de mon parcours d'étude. Qu'ils trouvent ici mes sincères vœux.

Finalement, je souhaite remercier mes frères et sœurs, ainsi que mes amis (es), vous m'avez toujours encouragé à terminer ce que j'entreprends et aidé dans les moments les plus difficiles. Merci infiniment.

Table des matières

Liste des Algorithmes

Liste des tableaux

Table des figures

Glossaire

ABC	Artificial Bee Colony algorithm
ACO	Ant Colony Optimization
ADABOOST	Adaptive Boosting
ALO	Ant Lion Optimization algorithm
BB	Branch and Bound algorithm
BBI	Branch and Bound by Interval algorithm
BFS	Best First Search
CCNUMA	Cache Cohérent Non-Uniform Memory Access
CLU	Clustering
CNN	Condensed Nearest Neighbor
CO	Combinatorial Optimization
COMA	Cache-Only Memory Architecture
COGIN	COverage-based Genetic INduction
CORE	CO-Evolutionary Rule Extractor
CS	Class-Separability
DE	Differential Evolution Algorithm
DMEL	Data Mining for Evolutionary Learning
DPGA	Dynamic Parametric Genetic Algorithms
DT-GA	Hybrid Decision Tree - Genetic Algorithm
DTNB	Decision Table/Naive Bayes
EA	Evolutionary Algorithm
ECPM-FS	Ensemble of Cooperative Parallel Metaheuristics – Feature Selection
EDA	Estimation of Distribution Algorithms
EM	Expectation Maximization algorithm
ENN	Edited Nearest Neighbor
FEAST mining	Feature subset sElection Algorithm based on aSsociaTion rule
FLC	Fuzzy Logic Controller
FS-SFS	Filtered and Supported Sequential Forward Search
GA	Genetic Algorithm
GASSIST	Genetic Algorithms based claSSIfier sySTem
GBML	Genetics-Based Machine Learning Algorithms
GCCL	Genetic Cooperative-Competitive Learning
GCNN	Generalized Condensed Nearest Neighbor rule
GIL	Genetic-based Inductive Learning
GIM-FS	Generalized Islands Model - Feature Selection
GP	Genetic Programming
GPIL	Genetic Programming for Inductive Learning
GRASP	Greedy Randomized Search Procedure

GSA	Gravitational Search Algorithm
GCM	Generalized-Modified Chang Algorithm
GSO	Glowworm Swarm Optimization algorithm
HEDT	Hybrid Evolutionary Decision Trees
HIDER	HIerarchical DEcision Rules
HRCNN	hyperrectangular composite neural networks
HMs	Hybrid Metaheuristics
HRH	High-level Relay Hybrid
HS	Harmony Search algorithm
HTH	High-level Teamwork Hybrid
IB2	Instance-Based learning algorithm
IB3	Instance-Based learning algorithm
ILGA	Incremental Learning with Genetic Algorithms
IRL	Iterative rule learning
KDD	Knowledge Discovery in Databases
K-NN	k-nearest neighbors (KNN) algorithm
LEM	Learnable Evolution Model
LEMMO	Learnable Evolution Model for Multi-Objective
LP	Linear Programming
LRH	Low-level Relay Hybrid
LS-HJ	Hooke and Jeeves local search
LTH	Low-level Teamwork Hybrid
LVF	Las Vigas Algorithm
MIMD	Multiple Instructions multiple data
MISD	Multiple instructions single data
MPI	Message Passing Interface
NCA	Neuronal Communication Algorithm
NCGABIL	Neuronal Communication Genetic Algorithm Based Inductive Learning
NNGE	Non-Nested Generalized Exemplars
NoRMA	No Remote Memory Access
NP-HARD	Nondeterministic Polynomial Time
NSB	Nearest Subclass Classifier
NUMA	Non-Uniform Memory Access
Oblique-DT	Oblique Decision Tree
OCEC	Organizational Co-Evolutionary algorithm for Classification
OIGA	Ordered Incremental Genetic Algorithm
ONER	One Rule
OpenMP	Open MultiProcessing
OSC	Object Selection by Clustering
PART	Partial decision tree
PBI	Population-Based Incremental learning
PFHRCNN	PSO-based Fuzzy Hyper-Rectangular Composite Neural Network
Pitts-GIRLA	Pittsburgh Genetic Interval Rule Learning Algorithm
PO	Postoperative
PS	Prototype Selection
PSO	Particle Swarm Optimization
PSR	Prototype Selection by Relevance

QAP	Quadratic Assignment Problem
RCL	Restricted Candidate List
RIDOR	RIpple DOwn Rules
RIPPER	Repeated Incremental Pruning to Produce Error Reduction
SARG	Self-adjusting Associative Rules Generator
SBS	Sequential Backward Selection
SFS	Sequential Forward Selection
SIA	Supervised Inductive Algorithm
SIMD	Single Instruction on Multiple Data
SISD	Single instruction on single data
SNN	Selective Nearest Neighbor rule
SVM	Support Vector Machine
SVR	Support Vector Regression
TARGET	Tree Analysis with Randomly Generated and Evolved Trees
TSP	Travelling Salesman Problem
TSS	Training Set Selection
UCI	University of California, Irvine
UMA	Uniform Memory Access
VNS	Variable Neighborhood Search
WP	Weighting prototypes
ZeroR	Zero Rule

Introduction générale

1 Contexte de l'étude

L'importance des données est reconnue comme fondamentale dans tous les domaines : la bonne gouvernance, la santé efficace, l'économie performante, les entreprises compétitives , etc.
Profiter de cette manne précieuse de données exige néanmoins l'application de processus d'extraction de connaissances à partir de ces données, toujours plus performants. La quantité de plus en plus phénoménale et complexe de ces données rend la tâche encore plus difficile pour les algorithmes existants. Tout algorithme a ses limites devant les problèmes complexes du monde réel.
Un prétraitement des données devient de plus en plus inévitable et constituera une des étapes les plus importantes dans ce processus d'extraction de connaissances à partir de données. Il concerne la mise en forme des données en entrées selon leur type (numérique, symbolique, image, texte, son), le nettoyage de ces données, le traitement des données manquantes, la sélection des attributs pertinents et la sélection des instances pertinentes.
Cette première phase est essentielle car le bon choix des descripteurs et des données de qualité en entrée est directement lié à la qualité des modèles et la connaissance en sortie. L'information nécessaire à la construction d'un meilleur modèle de prévision peut être disponible dans les données mais encore faut-il la rendre plus accessible. Sans prétraitement des données, le meilleur algorithme de classification ne peut démontrer toute sa puissance. Des données de mauvaise qualité signifient des données contenant de nombreuses erreurs, des informations non pertinentes ou choix inapproprié des données ou d'échantillons d'apprentissage. Cela aura sans aucun doute des effets négatifs sur l'opération de l'apprentissage. Cette étape détermine au mieux le succès de l'ensemble du processus. De plus, les informations non pertinentes alourdiront le temps de l'apprentissage et compliqueront l'interprétation du modèle. Un bon prétraitement des données, et spécialement une bonne sélection des données, représente généralement la partie la plus importante du processus de prédiction, et peut être aussi importante que le choix de l'algorithme approprié et son paramétrage.
Dans un contexte plus large, la sélection des données a trois objectifs principaux [Kordos 2017] :
- Diminuer la taille des données,
- Réduire le bruit dans les données,
- Sélectionner des points de données représentatifs (prototype) afin d'en permettre une meilleure compréhension du processus.
Les deux premiers objectifs peuvent être mis en œuvre par la sélection

d'attributs.
Deux options sont possibles pour traiter les données volumineuses : la mise à l'échelle des algorithmes de fouille de données [Provost & Kolluri 1999] qui n'est réalisable qu'avec certains algorithmes et la réduction de données. Cette dernière [Pyle 1999] est la tâche essentielle de prétraitement des données dans un processus de fouille de données. Le but primordial de la réduction des données est de réduire les données initiales en gardant les informations les plus pertinentes. De cette façon, il est souvent possible d'améliorer les résultats obtenus par n'importe quelle tâche de fouille de données, allant des processus prédictifs (classification, régression) aux processus descriptifs (segmentation, extraction de règles d'association).
Donc, la réduction des données consiste à éliminer des données sans importance pour la tâche de la fouille de données en cours. Les techniques de réduction de données emploient différentes approches : sélection des attributs, et la sélection des instances. La discrétisation des attributs numériques est parfois aussi considérée. Les processus de réduction des données peuvent être exécutés par différentes manières, les plus remarquables étant données par [Derrac *et al.* 2010] :
— Sélection des attributs ([Liu & Motoda 1998] .[Liu & Motoda 2008]), réduisant le nombre de colonnes (de la matrice) dans un ensemble de données. Ce processus s'appelle Sélection des attributs.
— Rendre les valeurs des attributs discrètes ([Hussain *et al.* 1999] réduisant le nombre de valeurs possibles des attributs. Ce processus est connu sous le nom Discrétisation des attributs.
— Générer de nouveaux attributs [Guyon *et al.* 2006] décrivant les données de manière plus appropriée à partir des anciens. Ce processus s'appelle Extraction des attributs.
— Sélection d'instances [Blum & Langley 1997]. ([Liu & Motoda 2001]. [Liu & Motoda 2002]), réduisant le nombre de lignes (de la matrice) dans un ensemble de données. Ce processus s'appelle Sélection d'instance (IS).
— Génération de nouvelles instances ([Bezdek & Kuncheva 2001]. [Lozano *et al.* 2006]), qui décrivent le jeu de données initial en générant des exemples artificiels. Ce processus s'appelle la génération d'instance (IG).
La sélection d'attributs est l'une des étapes de prétraitement qui permet de développer la puissance du processus de l'extraction de la connaissance à partir des données, en fournissant une plus grande clarté du problème à étudier. Elle est considérée comme un problème d'optimisation NP-difficile qui peut être résolu par les métaheuristiques. Hybridation de Métaheuristique avec d'autres méthodes (Métaheuristiques, Méthodes exactes, Technique de Fouille de données, ...) permet par définition d'améliorer les méthodes de base en joignant leurs points forts et éliminant leurs points faibles pour une meilleure sélection d'attributs et par la suite de construire un modèle d'apprentissage plus performant.
Après le prétraitement des données vient la partie la plus importante du processus d'extraction de la connaissance à partir des données, qui est le choix et l'application de l'algorithme le plus adéquat pour la tâche en cours de la réalisation (classification supervisée, segmentation, recherche des règles d'association,...).

Trois éléments importants peuvent être définis, cette vue n'est pas nécessairement complète ou englobante, dans n'importe quel algorithme de la fouille de données [Usama *et al.* 1996] pour expliquer les concepts clés des algorithmes d'une façon relativement unifiée et intégrée :
- Représentation du modèle,
- Évaluation du modèle,
- Méthode de recherche.
— La représentation des modèles est l'outil employé pour la description des modèles découvrables. Si la représentation est trop limitée, aucun temps d'apprentissage ou exemples ne peuvent donner un modèle adéquat pour les données.
L'analyste de données doit comprendre totalement les propositions de représentation qui peuvent être inhérentes à un algorithme donné. Il est nécessaire lors de la conception des approches d'indiquer nettement quelles propositions de représentation sont faites par un algorithme particulier. Le problème de surajustement des données peut être survenu et accru par l'augmentation de la puissance de la représentation des modèles, ce qui nous ramène à la diminution de la précision des prédictions sur les données invisibles.
— Un modèle particulier et ses paramètres vérifient les buts du processus KDD par ses critères d'évaluation qui sont des énoncés quantitatifs de la mesure (ou des fonctions d'ajustement). Par exemple, les modèles prédictifs sont généralement évalués par la précision de la prédiction expérimentale sur des ensembles de tests. Les modèles descriptifs peuvent être appréciés suivant les dimensions d'exactitude prédictive d'emploi et de compréhension du modèle adapté.
— La méthode de recherche regroupe deux principes : la recherche des paramètres et la recherche du modèle.
Après la validation de la représentation (ou les représentations) et les critères d'évaluation du modèle, on passe au problème de l'exploration de données, ce dernier sera orienté vers une simple tâche d'optimisation : recherchez les paramètres et les modèles des représentations choisies qui améliorent les critères d'évaluation.
Recherchez les paramètres : l'algorithme doit rechercher les paramètres qui améliorent les critères d'évaluation du modèle en termes de la représentation valide du modèle et des données observées. La recherche du modèle s'exécute itérativement sur la méthode de la recherche des paramètres.
La classification supervisée est l'une des tâches de la fouille de données. Elle requiert la connaissance a priori pour spécifier l'appartenance d'une instance à une classe. À l'aide des données d'entraînement, cette tâche permet d'apprendre un algorithme qui est capable de prédire l'appartenance d'une nouvelle instance à une classe. Elle est considérée comme un problème d'optimisation NP-difficile qui peut être résolu par les Métaheuristiques. D'où l'idée d'utiliser l'hybridation de métaheuristiques avec d'autres méthodes (Métaheuristiques, Méthodes exactes, Technique de Fouille de données, ...) cette hybridation amplifie les méthodes de base en bénéficiant de leurs points forts et excluant leurs points faibles pour une meilleure classification supervisée et par la suite de construire un modèle d'apprentissage plus robuste.

Donc, dans ce contexte particulier, notre souhait est de démontrer l'apport positif de l'utilisation de l'hybridation de métaheuristique dans différentes étapes du processus de l'extraction de la connaissance à partir de données.

2 Problématique et contributions

Hybridation de Métaheuristique dans le domaine d'optimisation combinatoire représente une étude intéressante. L'idée principale de l'hybridation est de combiner les avantages de chaque algorithme pour former une méthode hybride minimisant tout inconvénient présent des algorithmes combinés. Les métaheuristiques hybrides appliquées dans le domaine d'extraction de connaissances donnent des résultats performants.
Comme exemples de bases de données volumineuses, complexes et dignes d'intérêts on trouve les bases de données biomédicales. Ces bases rassemblent des quantités élevées d'informations sur le monde médical. Les relations et les modèles au sein de ces données pourraient fournir de nouvelles connaissances médicales dont l'exploitation pourrait améliorer la prise en charge des patients et même sauver des vies humaines. D'où notre choix et préférence de démontrer l'efficacité de notre approche sur ce genre de bases.
Plusieurs approches ont été développées et appliquées pour découvrir ces connaissances.
L'objectif de cette thèse est de développer des modèles encore plus performants pour réaliser la classification, une tâche de fouille de données, en utilisant l'hybridation des métaheuristiques dans le processus d'extraction de connaissance à partir de données, aussi bien dans l'algorithme créateur du modèle, que dans le prétraitement de données d'apprentissage, pour aboutir au meilleur modèle.

Un résumé de notre problématique peut se formuler au moyen des interrogations suivantes :
- L'hybridation de Métaheuristiques donne-t-elle de bons résultats dans le processus d'extraction de la connaissance à partir de données ?
- La tâche de classification peut-elle être améliorée ? Et comment ?
Afin de répondre à toutes ces questions, on présente de nouvelles approches originales pour une meilleure extraction de la connaissance dans le cadre de la classification de données.
Dans la présente thèse, on contribue, essentiellement, à
- *Neuronal Communication Genetic Algorithm Based Inductive Learning* [Alaoui & Elberrichi 2020b]
L'élaboration de stratégies d'apprentissage puissantes dans les différents domaines constitue un véritable défi. Les algorithmes d'apprentissage automatique sont utilisés pour extraire des connaissances de haut niveau à partir d'ensembles de données. Pour construire un algorithme robuste basé sur des règles, des métaheuristiques hybrides ont été proposées pour la classification des données. L'approche hybride utilise l'apprentissage inductif basé sur la communication neurale et l'algorithme génétique pour créer un modèle robuste de classification.

- Enhanced Ant Colony Algorithm for Best Features Selection for a Decision Tree Classification of Medical data [Alaoui & Elberrichi 2020a]
- Feature Subset Selection Using Ant Colony Optimization For A Decision Trees Classification of Medical Data [Alaoui & Elberrichi 2018]
Les algorithmes de classification sont largement appliqués dans les différents domaines par exemple dans le domaine médical pour classifier les données à des fins diagnostiques. On peut réduire le coût du diagnostic en évitant de nombreux tests, donc en sélectionnant seulement les attributs qui sont importants pour la prédiction de la maladie et pour la construction d'un modèle d'apprentissage robuste et optimal.

Le présent document présentera toutes ces approches, ainsi que la description, l'étude et l'analyse des résultats obtenus.

3 Organisation de la thèse

La thèse est structurée en deux parties, la première comporte un background et un état de l'art nécessaires tant à la définition et qu'à la compréhension des éléments de l'étude, la seconde énonce et explicite nos approches et nos contributions évoquées plus haut. On donne ci-dessous une vue d'ensemble de la structure de la présente thèse suivie d'un bref résumé de chaque chapitre.
Partie A : Background et État de l'art
Premier chapitre : Métaheuristiques
L'objectif de ce chapitre est de donner une présentation générale des différentes métaheuristiques. On énonce, tout d'abord, les notions élémentaires d'optimisation difficile, on présente par la suite les métaheuristiques hybrides et les métaheuristiques parallèles tout en détaillant leurs caractéristiques ainsi que leurs fonctionnalités.
Deuxième chapitre : Processus d'extraction des connaissances à partir de données
Ce chapitre est consacré à la définition des concepts de base du processus d'extraction des connaissances à partir de données. On fournit des définitions concernant la fouille de données et le processus de prétraitement de données et on décrit quelques exemples.
Troisième chapitre : Hybridation des Métaheuristiques pour l'extraction des connaissances
Dans ce chapitre, on positionne sur l'hybridation de métaheuristiques pour l'extraction des connaissances à partir de données, on cite les critères principaux de la classification des métaheuristiques hybrides, on présente également comment concevoir une métaheuristique hybride, afin de définir des directives utiles pour le développement d'une nouvelle métaheuristique hybride pour les différentes tâches de la fouille de données, classification supervisée, les règles d'association et la segmentation tout en appliquant le processus de prétraitement de données.
Partie B : Contributions
La deuxième partie est destinée à la description des approches proposées dont les algorithmes, les fonctionnalités et les résultats sont présentés et détaillés, cette partie est organisée autour de deux chapitres et concerne nos contributions.

Quatrième chapitre : Approches hybrides pour la sélection des attributs.
Ce chapitre présente la contribution qu'on propose pour l'amélioration des modèles d'apprentissage avec l'hybridation de métaheuristiques et un prétraitement des données par la sélection d'attributs, on donne ses algorithmes ainsi que les différentes étapes détaillées et on décrit les expérimentations des approches proposées, les résultats de ces évaluations seront également discutés.
Cinquième chapitre : Approches hybrides pour la classification supervisée.
Il concerne la description des approches qu'on propose pour l'amélioration des modèles d'apprentissage avec l'hybridation de métaheuristiques, on expose les algorithmes utilisés ainsi que les différents détails et résultats.
Conclusion générale et perspectives
La présente thèse s'achève par une conclusion où on récapitule les principales contributions de la présente étude et les perspectives envisagées.

A

Background et État de l'art

Métaheuristiques 1

Sommaire

1.1 Introduction

Les métaheuristiques donnent des solutions de bonne qualité en un temps raisonnable à des problèmes combinatoires connus difficiles pour lesquels il n'existe pas des méthodes classiques plus efficaces. L'hybridation des métaheuristiques donne des solutions de meilleures qualités [Talbi *et al.* 2007], en effet les meilleurs résultats trouvés pour plusieurs problèmes d'optimisation combinatoires ont été obtenus avec des algorithmes hybrides. Il existe des problèmes qui prennent un temps élevé, de ce fait le parallélisme est une solution non seulement de ce problème pour diminuer le temps d'exécution mais il est essentiel pour améliorer la qualité des solutions trouvées.
Dans ce chapitre, les métaheuristiques simples, hybrides et parallèles et leurs caractéristiques sont présentées.

1.2 Problèmes d'optimisation

Les problèmes d'optimisation peuvent se classer en deux groupes : les problèmes où les solutions sont représentées avec des variables réelles et les problèmes où les solutions sont représentées avec des valeurs discrètes de variables. Parmi les derniers, il existe les problèmes d'optimisation combinatoire (CO), on est à la recherche d'un nombre entier, un sous-ensemble, une permutation ou une structure de graphe à partir d'un ensemble dénombrable [Papadimitriou & Steiglitz 1982].
Définition : un problème d'optimisation combinatoire peut être défini par :
— Un ensemble de variables X={x1,. . . ,xn} ;
— Les domaines variables {D1,. . . , Dn } ;
— Les contraintes entre les variables ;
— Une fonction « objectif » f à minimiser (ou à maximiser).
— L'espace de recherche S, chaque élément de S peut être vu comme une solution candidate, pour résoudre un problème d'optimisation combinatoire il faut trouver une solution optimale globale $s* \in S$ tel que f(s*) < f(s), c.à.d. minimiser (maximiser) la valeur de la fonction « objectif ».
Comme des exemples des problèmes combinatoires, on trouve :
— Le Problème du Voyageur de Commerce (TSP) qui repose sur la recherche de plus court chemin et qui permet d'aller vers un certain nombre de villes et revenir à la ville de départ.
— Le Problème d'Affectation Quadratique (QAP) : ce problème a des nombreuses applications telles que la répartition de bâtiments ou de services, l'affectation des portes d'aéroport, la synthèse d'images.

Les algorithmes développés pour la résolution des problèmes d'optimisation peuvent être répartis en algorithmes exacts et algorithmes approchés. Les algorithmes exacts permettent de résoudre certains problèmes en un temps fini ([Papadimitriou & Steiglitz 1982]. [Nemhauser & Wolsey 1988]). Ces méthodes requièrent en général un certain nombre de propriétés de la fonction « objectif », comme la stricte convexité, la continuité ou encore la dérivabilité. On trouve par exemple : la méthode de la programmation linéaire, quadratique ou dynamique, la

méthode du gradient, la méthode de newton, ... [Johann 2006].
Pour les problèmes d'optimisation combinatoire qui sont NP-difficiles [Garey & Johnson 1979], on ne trouve pas un algorithme exact polynomial, supposant que $P \neq NP$. Quand le nombre de combinaisons possibles devient exponentiel par rapport à la taille du problème, le temps de calcul devient vite critique. Si les algorithmes exacts permettent d'obtenir une ou plusieurs solutions dont l'optimalité est assurée, dans certains cas (les algorithmes approchés), on peut trouver de solutions de bonnes qualités sans la vérification d'optimalité mais avec un temps de calcul réduit.

1.3 Méthodes d'optimisations approchées

1.3.1 Heuristiques

L'heuristique est une méthode, conçue pour un problème d'optimisation, qui donne une solution non forcément optimale lorsqu'on lui présente une instance de ce problème.

1.3.2 Métaheuristique

« Un processus itératif qui oriente une heuristique, en combinant intelligemment divers concepts pour l'exploration et l'exploitation de l'espace de recherche. Des techniques d'apprentissage sont implémentées pour structurer l'information afin de découvrir efficacement des solutions optimales, ou presque optimales » [Osman & Laporte 1996].
Les métaheuristiques sont en général des algorithmes stochastiques itératifs, qui avancent vers un optimum global, c'est-à-dire l'extremum global en évaluant une certaine fonction objectif. Elles se comportent comme des algorithmes de recherche, essayant d'apprendre les propriétés d'un problème afin d'en obtenir une approximation de la meilleure solution d'une façon près des algorithmes d'approximations.

1.4 Classification des métaheuristiques

On classe les métaheuristiques selon les critères suivants ([Blum & Roli 2003]. [Clerc & Siarry 2004]) :

1.4.1 Inspiration de la nature

Une façon à organiser les métaheuristiques repose sur la séparation de celles qui sont inspirées d'un phénomène naturel, de celles qui ne le sont pas. Les algorithmes génétiques et les algorithmes de colonies de fourmis sont inspirés de la nature tandis que l'algorithme de Recherche Tabou et l'algorithme de Recherche Locale Itérative sont non inspirés de la nature, d'après les auteurs [Blum & Roli 2003], cette classification n'est pas significative :
- De nombreux algorithmes hybrides récents n'appartiennent pas à ces deux classes
- La difficulté de distinguer une approche dans certaines situations, par

exemple, l'emploi de la mémoire dans la recherche tabou n'est pas inspiré de la nature.

1.4.2 Nombre des solutions

Une autre façon de la classification est de diviser les métaheuristiques en deux grandes classes : les métaheuristiques à solution unique (Recherche Locale, Recherche Tabou, Recuit Simulé,. . . .) et celle à une population de solutions (Algorithmes Evolutionnaires, Recherche par Dispersion, Optimisation par Essaims Particulaires, . . .).
Les algorithmes d'optimisation à population de solutions développent, au fur et à mesure des itérations, une population des solutions. L'avantage de ces algorithmes est d'implémenter la population comme un facteur de diversité.
Les algorithmes d'optimisation à solution unique sont nommés les méthodes trajectoires (décrire une trajectoire en espace de recherche au cours du processus de recherche).

1.4.3 Fonction «objectif»

On peut organiser les métaheuristiques selon l'emploi de la fonction « objectif ». Étant donné un problème d'optimisation reposant sur la minimisation d'une fonction f dans un espace s de solutions, les algorithmes qui travaillent directement sur f sont statiques, les algorithmes sont dynamiques s'ils utilisent une solution g trouvée à partir de f en complétant par quelques éléments qui tentent de modifier la topologie de l'espace de solutions, ces éléments supplémentaires pouvant changer durant la recherche.

1.4.4 Voisinage

Il est peut-être important d'appliquer des métaheuristiques basées sur plusieurs types de voisinage c'est le cas de la Recherche à Voisinage Variable où la diversification est faite.

1.4.5 Emploi d'une mémoire

Une propriété importante est l'emploi de l'historique pour la classification, les algorithmes sans mémoire sont des processus markoviens puisque l'action à effectuer est généralement déterminée par l'état courant. Une mémoire à long terme garde une trace de mouvements récemment réalisés, des solutions visitées ou en général des décisions prises. Une mémoire à court terme est un regroupement de paramètres de synthèse sur la recherche.

1.4.6 Implémentations parallèles

Dans les problèmes de déplacement de composants, les tâches peuvent être distribuées naturellement entre plusieurs processeurs : chacun d'eux est destiné à optimiser une zone géographique donnée, et des informations sont échangées périodiquement entre les processeurs voisins.

1.4.7 Hybridation

Elle consiste à tirer parti des avantages spécifiques de métaheuristiques différentes en les combinant [Talbi 2002].

1.4.8 Optimisation multi-objectif

Il s'agit d'une optimisation simultanée de plusieurs objectifs contradictoires [Collette & Siarry 2002]. La recherche consiste à trouver un ensemble d'optima qui compose la surface de compromis du problème considéré.

1.4.9 Optimisation multimodale

Il consiste à trouver un ensemble de meilleurs globaux et/ou locaux, les algorithmes évolutionnaires sont particulièrement bien adéquats à cette tâche, de par leur nature distribuée. Les variantes de multipopulation exploitent parallèlement plusieurs populations qui s'attachent à repérer des optimums différents.

1.5 Description des principales Métaheuristiques

On se concentre sur la classification qui distingue les approches à solution unique des approches basées sur des populations.

1.5.1 Approches à solution unique

1. Méthode de Descente
Le succès des Méthodes de Descente (Hill Climbing) est lié à leur simplicité de mise en œuvre et à leur rapidité [Harilaos 1976]. Généralement, elle ne fait que compter f(s+i)-f(s), où, i correspond à un déplacement élémentaire, et si cette forme peut se simplifier mathématiquement, alors on pourra estimer plus vite cette différence.
Le principe consiste à déterminer une solution s' dans le voisinage d'une solution s en développant la recherche tel que f(s')< f(s).
On peut soit d'examiner toutes les solutions de voisinage et extraire la meilleure de toutes, soit d'examiner un sous-ensemble de voisinage.
L'algorithme1.1 est représenté comme suit :

Algorithme 1.1 Méthode de Descente

solution initiale s
repeat
  Choisir s' dans N(s) **if** $f(s') < f(s)$ **then**
    $s = s'$
  end
until $f(s') >= f(s), \forall s' \in S$;

L'ensemble S contient l'ensemble des solutions pouvant être sélectionnées pendant la recherche. Le déplacement dans l'espace de recherche est

fait selon la structure de voisinage N. Une variante de la méthode consiste à parcourir N(s) et à prendre la première solution s' trouvée telle que f(s') <f(s).
Le premier inconvénient de la méthode de descente est son arrêt au premier minimum local rencontré. Pour développer les résultats, on peut lancer plusieurs fois l'algorithme en partant d'un ensemble initial de différentes solutions, mais la performance de cette technique décroît rapidement. Pour éviter d'être bloqué au premier minimum local rencontré, on peut décider d'accepter, sous certaines conditions, de se déplacer d'une solution s vers une solution so telle que $f(s_0) > f(s)$.

2. Recuit Simulé

La méthode du Recuit Simulé (Simulated Annealing) est une sorte d'une similitude entre le phénomène physique de refroidissement lent d'un corps en fusion, qui l'amène à une situation solide, de basse énergie. Il faut diminuer lentement la température, en notant des degrés assez longs pour que le corps atteigne l'équilibre thermodynamique à chaque degré de température pour les matériaux, cette énergie s'exprime par l'acquisition d'une structure régulière, comme dans les cristaux et l'acier [Kirkpatrick & Gelatt 1983]. Par similitude avec le processus physique, la fonction à minimiser est l'énergie du système E, on emploie la température du système comme un paramètre.
On joint avec une solution aléatoire s_0 une énergie initiale $E = E_0$ et une température initiale $T = T_0$ élevée. À chaque itération de l'algorithme, la solution est changée, ce qui provoque une variation ΔE de l'énergie du système. Si cette variation réduit l'énergie du système (ΔE est négative) on l'accepte, sinon on l'accepte avec une probabilité $e^{-\Delta E/T}$, La température est restée constante.
Cet algorithme commence par s_0, et continue jusqu'à k_{max} (maximum d'étapes) ou jusqu'à un état ayant une énergie e_{max}, l'appel voisin (s) engendre un état aléatoire voisin d'un état s, l'appel aléatoire renvoie une valeur aléatoire dans l'intervalle [0,1]. L'appel temps(r) renvoie la température à utiliser selon la fraction r du temps total déjà dépensé.

Algorithme 1.2 Recuit Simulé

s=so
e=E(s)
k=0
while $k < k_{max} \wedge e > e_{max}$ **do**
  $s = voisin(s)$ $e_n = E(s_n)$
  if $e_n < e \vee alatoire() < p(e_n - e, temp(k/k_{max}))$ **then**
    $s = s_n$
    $e = e_n$
    $k = k + 1$
  end
  Return s
end

La méthode du recuit simulé a l'avantage d'être souple vis-à-vis des évolutions du problème et facile à implémenter, elle donne généralement des solutions de bonne qualité et des excellents résultats pour un nombre

de problèmes complexes. Les inconvénients majeurs sont : l'utilisation des nombreux tests nécessaires pour trouver les bons paramètres, le temps de calcul est excessif dans certaines applications.

3. Recherche Tabou

La Recherche Tabou (Tabu Search) a été introduite par l'auteur Glover ([Glover 1986]). L'idée de base existe dans [Hansen 2000], elle n'a aucune propriété stochastique et implémente la notion de mémoire pour éviter de tomber dans un optimum local. Dans la recherche Tabou et à partir d'une solution donnée, on évalue toutes ses voisines, on prend la meilleure solution estimée même si elle reste non développée.

On emploie le terme de mémoire (liste Tabou), on conserve les k dernières configurations visitées dans une mémoire à court terme et on empêche tout déplacement qui amène à une de ces configurations. Cette mémoire permet d'éviter tous les cycles de longueur inférieure ou égale à k (une valeur qui dépend du problème à résoudre). La sauvegarde de configurations serait trop élevée en fonction du temps de calcul et d'espace mémoire, ce qui réduit l'efficacité de l'algorithme. La liste Tabou sauvegarde des propriétés de configurations complètes. C'est-à-dire lorsqu'un déplacement vient d'être réalisé, c'est en général la propriété perdue par la configuration actuelle qui devient Tabou.

Lorsque les listes Tabou font intervenir des propriétés de changements, les interdictions qu'elles produisent peuvent s'avérer trop fortes et restreindre l'ensemble des solutions acceptées à chaque itération d'une façon brusque. Un moyen spécial, l'aspiration, est implémenté pour résoudre ce problème, il permet de lever le statut Tabou d'une configuration, sans conduire à un risque de cycles dans le processus de recherche. La fonction d'aspiration est déterminée de différentes façons. La plus simple réside dans la révocation du statut Tabou d'un déplacement si ce dernier permet de trouver une meilleure solution que celle obtenue.

Soient s une solution actuelle, T : liste tabou

$N^T(s)$ est un ensemble de solutions de N(s) vers laquelle la Recherche Tabou subit à une direction. Cet ensemble comporte toutes les solutions qui ne sont pas taboues ainsi que celles qui sont taboues mais dont le statut est retiré en raison de critère d'aspiration.

Comme un critère d'arrêt, on prend un nombre maximum d'itérations sans développer s^* ou un temps limite après lequel la recherche doit s'arrêter.

Algorithme 1.3 Recherche Tabou

$s = s_0$;
$T = \emptyset$;
$s^* = s$;
while *arrêt n'est pas satisfait* **do**
    Déterminer une solution s' qui minimise f(s') dans $N^T(s)$
    if $f(s') < f(s^*)$ **then**
        $s^* = s'$;
        $s = s'$
        mise-à-jour T ;
    end
end

La recherche tabou est une méthode de recherche locale, et la structure de son algorithme de base est proche de celle du recuit simulé, avec l'avantage d'avoir un paramétrage simplifié : le paramétrage consistera d'abord à trouver une valeur indicative t d'itérations pendant lesquelles les mouvements sont interdits. Il faudra également choisir une stratégie de mémorisation. En revanche, la méthode Tabou exige une gestion de la mémoire de plus en plus lourde en mettant des stratégies de mémorisation complexe. L'efficacité de la méthode tabou offre son utilisation dans plusieurs problèmes d'optimisation combinatoire classiques tels que le problème de voyageur de commerce, le problème d'ordonnancement, le problème de tournées de véhicules, etc.

4. GRASP

GRASP (Greedy Randomized search Procedure) est une approche proposée par les auteurs Feo et Resende ([Feo & M.-G.-C.Resende 1989]. [Feo 1994]. C'est une hybridation puisqu'elle associe les avantages de la Méthode de Descente, la recherche aléatoire et la méthode de Voisinage.
C'est une approche itérative (Algorithme 1.4) qui s'exécute en deux étapes :
Une étape de construction d'une solution et une étape développement d'une solution.
Durant la première et d'une façon répétitive une solution est construite : à chaque pas et à partir d'une liste des candidats on additionne une solution, on emploie une liste dynamique RCL (Restricted Candidate List) des meilleurs candidats gardés avec une mesure donnée (fonction gloutonne ou une heuristique permettant de mesurer le bénéfice qu'on peut estimer).
- On continue jusqu'à ce qu'une solution complète soit trouvée.
- Dans la deuxième étape et à partir de la solution obtenue par la première étape, une Méthode de Descente, Recuit Simulé ou Recherche Tabou est utilisée pour développer la solution.

Algorithme 1.4 Méthode GRASP

$f^* = \infty$; (une valeur de meilleure solution rencontrée) ;
while *critère d'arrêt n'est pas vérifié* **do**
    $S = \emptyset$;
    while *solution courante n'est pas complète* **do**
        Évaluer l'ensemble des éléments pouvant être rajoutés à s
        déterminer la liste RCL
        choisir un meilleur élément dans RCL et l'ajouter à S
    end
    Appliquer une Méthode de Descente, Recuit Simulé ou de Recherche Tabou à s : $s' = s$
    if $f(s') < f(s^*)$ **then**
        $s^* = s'$;
        $f(s^*) = f(s')$;
    end
end

La GRASP est une approche simple qui permet de développer les solutions par l'intégration d'une heuristique spécifique. Elle est l'une des métaheuristiques qui possèdent des avantages prometteurs pour les problèmes d'optimisation combinatoires ; parmi lesquels on note : [Mauricio & Ribeiro 2002]
— Nombre de paramètres à régler : il n'y a que deux paramètres à manipuler, Alpha et le nombre des itérations.
— Temps de calcul : par rapport à d'autres métaheuristiques, l'algorithme consomme moins de temps CPU, et il est prédictible.
— Parallélisme : peut être trivialement implémenté en parallèle, donc un gain considérable en temps de calcul.
— Implémentation : relativement plus simple à implémenter.
Néanmoins, comme toute Métaheuristique, GRASP est loin d'être une méthode parfaite [Mauricio & Ribeiro 2002] :
— Incertitude sur la qualité de la solution : la solution finale dépend toujours des paramètres de l'algorithme.
— Construction sans mémoire : informations non sauvegardées d'une itération à l'autre. Elle est moins performante par rapport à d'autres Métaheuristiques (elle peut tomber sur le minimum local et ceci est dû à l'absence de mémoire).

5. Recherche à Voisinages Variables

La méthode de Recherche à Voisinages Variables (VNS) a été proposée par [Mladenovic & Hansen 1997]. Elle consiste à modifier systématiquement la forme de voisinage avec une heuristique de recherche locale au lieu d'employer une forme de voisinage unique. On génère une solution aléatoirement dans le voisinage de la solution actuelle dont laquelle une Méthode de Descente locale est réalisée. Si l'optimum local trouvé n'est pas le meilleur que la solution actuelle, on déplace vers le voisin suivant. On reprend la recherche à partir du premier voisinage pour explorer d'autres meilleures solutions (Algorithme 1.5). Tel que$N^(t)(S)$ est l'ensemble des solutions dans le t^{me} voisinage de s.

Algorithme 1.5 Recherche á Voisinages Variables

Choisir une solution $s \in S$;
$t = 1$;
while *critère d'arrêt n'est pas vérifié* **do**
  - Choisir aléatoirement une solution s' dans $N^(t)(S)$;
  - Appliquer une procédure de recherche locale à s' ;
  - $s'' = s'$;
  if $f(s'') < f(s)$ **then**
    ss'' ; $t = 0$;
  end
  if $t < T$ **then**
    $tt + 1$;
  end
  else
    $t = 1$
  end
  ;
end

La Métaheuristique Recherche à Voisinages Variables exploite l'idée de changer de voisinage au cours des itérations : on prend un voisinage plus grand lorsqu'on est bloqué en un optimum local et un voisinage plus petit lorsque le nombre de voisins réalisables est très grand.

1.5.2 Approches basées sur les populations

1. Algorithmes de Colonies de Fourmis
L'optimisation naturelle par les fourmis est originalement intégrée pour l'optimisation par les auteurs Colorni, dorigo et Maniezzo ([Colorni *et al.* 1992]).
Une colonie de fourmis communique indirectement via des changements dynamiques des pistes de phéromones et construit une solution à un problème donné.
Elle consiste à parcourir un graphe de construction complet. On marque c_{ij} chaque élément instancié $x_i = v_{ij}$ et C l'ensemble des c_{ij} . Le graphe $G_c(V, E)$ est réalisé en liant C à l'ensemble des nœuds V ou l'ensemble des arêtes E. Une trace de phéromone τ_{ij} est liée à chaque c_{ij} .
Les fourmis se déplacent d'un nœud à un autre le long des arêtes du graphe de construction en exploitant les données que produisent les traces de phéromones, chaque fourmi a une quantité déterminée de phéromones pour chaque c_{ij} connu lors de son mouvement. Cette quantité $\Delta\tau$ peut être liée à la qualité de la solution obtenue. Les fourmis suivantes emploient l'information fournie par les traces de phéromones pour se diriger vers les points les plus riches de l'espace de recherche.
La présentation générale de l'algorithme (Ant Colony Optimization) est donnée par [Dorigo 2007] :

Algorithme 1.6 Optimisation par Colonies de Fourmis

Définir les paramètres, initialiser les traces de phéromones ;
while *critère d'arrêt n'est pas vérifié* **do**
- Construction des solutions par les fourmis ;
- Actions Spécifiques facultatives (seules les solutions localement optimisées sont utilisées dans la mise à jour des traces de phéromones) ;
- Mise à jour de phéromones ;

end

Le critère d'arrêt est défini, généralement, par un nombre maximal d'itérations ou un temps CPU maximal.
L'optimisation par colonie de fourmis est efficiente en optimisation combinatoire grâce à leur robustesse ie. La recherche reste efficace même si certains de ces individus sont défaillants, et à sa décentralisation, ie. Les fourmis n'obéissent pas à une autorité centralisée, comme elles ont l'avantage d'être exécutées en parallèle.

2. Algorithmes Génétiques

L'algorithme génétique (Genetic Algorithm) s'appuie sur des méthodes dérivées de la génétique et de l'évolution naturelle : sélection, croisement, mutation. Il a été proposé dans les années de 60 [Holland 1962]. L'auteur Goldberg ([Goldberg 1989]) a donné la naissance de sa théorie des Algorithmes Génétiques :
— Un individu est associé à un environnement par son code d'ADN.
— Une solution est associée à un problème par son indicateur de qualité.
— Une bonne solution à un problème donné peut être vue comme un individu capable de rester vivant dans un environnement donné. Un Algorithme Génétique recherche le ou les extrema d'une fonction définie sur un espace de recherche (Algorithme 1.7), pour l'utiliser, il faut connaître :
— Le principe du codage d'un individu d'une population : après l'étape de modélisation du problème à étudier, on associe à chaque solution de l'espace de recherche une structure de données. Il y a le codage binaire qui est le plus utilisé et le codage réel (employé pour l'optimisation du problème à variables réelles).
— La génération d'une population initiale : le choix d'une population initiale est important pour les prochaines générations et essentiellement pour la convergence rapide vers un optimum global.
— Les opérateurs permettent la diversification de la population au cours des générations et l'exploration d'espace de recherche.
L'opérateur de croisement mélange par la reproduction des particularités des individus choisis. L'opérateur de mutation a pour but de garantir l'exploration de l'espace de recherche, c'est une modification aléatoire des particularités d'un individu.
— Les paramètres de dimensionnement : la taille de la population, le critère d'arrêt, la capacité d'implémenter les opérateurs de croisement et de mutation.

Algorithme 1.7 Algorithme Génétique

— Initialiser aléatoirement une population d'individus.
— Pour aller d'une génération k à une génération k+1, on implémente les trois opérations pour tous les individus de la population k :
- Choisir les couples de parents p_1, p_2 en fonction de leur adéquation.
- Générer les couples d'enfants c_1 et c_2 par l'application l'opérateur de croisement avec une probabilité p_c, on applique l'opérateur de mutation sur des individus choisis avec une probabilité p_m et on génère les individus p'
- Évaluer c_1 et c_2, p' pour l'insérer dans la nouvelle population.
— Le critère d'arrêt : soit le nombre de générations est défini initialement ou un temps déterminé pour obtenir une solution où il n'y a pas d'amélioration.

L'universalité d'un tel algorithme pose évidemment des problèmes d'efficacité en pratique. En effet, en tant qu'une approche d'optimisation, un algorithme génétique classique se base uniquement sur des opérateurs « aveugles » et est donc rarement en mesure de produire des résultats comparables à ceux d'une méthode de voisinage. Un mécanisme pour pallier ce problème consiste à s'adapter l'algorithme génétique au problème donné. Plus précisément, au lieu d'utiliser aléatoirement les opérateurs, la mutation et le croisement sont adaptés en se concentrant sur des connaissances spéciales du problème. De cette façon, la recherche est bien dirigée et donc plus efficace. Il est nécessaire d'intégrer des connaissances du problème [Grefenstette 1987] un autre chemin intéressant pour développer l'efficacité des algorithmes génétiques consiste à associer le cadre génétique avec d'autres approches d'optimisation.

3. Algorithme d'Estimation de Distribution

Cet algorithme (EDA) a été proposé par l'auteur Baluja en 1994 ([Baluja 1994]) dans ses travaux sur PBI (Apprentissage Incrémental Basé sur Population) et par les auteurs muhlenbein et Paab en 1996 ([Muhlenbein & Paass 1996]).
Il utilise les données collectées tout au long du processus d'optimisation pour former des modèles probabilistes des bonnes régions de l'espace de recherche et pour produire des nouvelles solutions [Larranaga & Lozano 2001]. Elle n'emploie pas les opérateurs de croisement et de mutation implémentés par les algorithmes évolutionnaires.
On construit la nouvelle solution à partir des solutions des générations prédécesseurs (Algorithme 1.8).

Algorithme 1.8 Algorithme d'Estimation de Distribution

Générer aléatoirement une population initiale X^0 ;
Calculer le modèle P^0 ; $X^t = X^0$;
while *le critère d'arrêt n'est pas vérifié* **do**
- choix de la sous-famille X^t_{parent} de X^t ;
- Calcul dep^{t+1} à partir de X^t_{parent} ;
- Echantillonnage de p^{t+1} pour créer $X^t_{offspring}$;
- l'échange de certains éléments de X^t par les éléments de $X^t_{offspring}$ pour créerX^{t+1} ;
$t = t + 1$;

end

- On génère une population de solutions X^0.
- On forme un modèle probabiliste p^0.
- La boucle tant que rassemble quatre phases :
- Sélection d'une sous population de meilleures solutions.
- Établissement du modèle probabiliste p^1 à partir de cette sous population.
- Génération d'une nouvelle population de solutionX^1 offspring par l'échantillonnage du modèle p^1.
- Remplacement de certains éléments de X^0 par les éléments de $X^1_{offspring}$ pour créer X^1.
- On répète le processus jusqu'à ce que le critère d'arrêt soit vérifié.

La distribution probabiliste est calculée par différentes méthodes suivant le modèle utilisé (gaussienne, ...) [Larranaga & Lozano 2001].
EDA peut résoudre plusieurs classes de problèmes importants de manière robuste et évolutive, Nécessitant souvent une croissance polynomiale de faible ordre du nombre d'évaluations de fonction par rapport au nombre de variables de décision ([Larranaga & Lozano 2001]. [Lozano *et al.* 2006]).
La complexité polynomiale de faible ordre est insuffisante pour l'application pratique de l'algorithme EDA spécialement, quand le nombre de variables de décisions est extrêmement large, quand l'évaluation des solutions candidates est coûteuse en termes de calcul ou quand il y a beaucoup d'objectifs contradictoires à optimiser.

4. Optimisation par Essaims Particulaires

Cet algorithme (Particle Swarm Optimization, PSO) s'inspire à l'origine du monde du vivant. Il se base notamment sur un modèle amélioré par le biologiste Craig Reynolds à la fin des années 1980 pour la simulation du mouvement d'un groupe d'oiseaux. Il a été proposé par les auteurs Russel et James en 1995 ([Kennedy & Eberhart 1995]).
L'algorithme PSO (Algorithme 1.9) change la place d'un essaim de particules dans l'espace de recherche. Le déplacement de chaque particule est dominé par sa vitesse, la meilleure position qui a été prise est la meilleure position connue par toutes les particules de l'essaim.
Soit x_i un vecteur de position de la i^{me} particule de l'essaim
v_i un vecteur de vitesse de cette particule. D la dimension de ce problème.
x_i et v_i sont des vecteurs á D éléments dont la j^{me} est notée respectivement x_ij et v_ij Soit p_i un vecteur de dimension D qui correspond à la meilleure position atteinte par la particule i et p_ij sa coordonnée sur la dimension j.

On note g le vecteur de dimension D qui correspond à la meilleure position connue de l'essaim.

Algorithme 1.9 Optimisation par Essaims Particulaires

Générer aléatoirement un essaim initial.
Évaluer la fonction « objectif » pour chaque essaim
$x_i = p_i$; $i = 1, \ldots N$(taille de l'essaim)
Calcul g
while *le critère d'arrêt n'est pas vérifié* **do**
    Mise-à-jourx_i et vi selon les équations (1.1) et (1.2)
    Évaluer la fonction « objectif »
    mise-à-jour p_i
    mise-à-jour g
end

A l'itération t+1 le déplacement des particules est calculé par les équations :

$$v_{ij}(t+1) = w.v_{ij}(t) + c_1.r_1.(p_{ij} - x_{ij}(t)) + c_2.r_2.(g_j - x_{ij}(t)) \quad (1.1)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \quad (1.2)$$

Où,
w : coefficient de l'inertie
c_1, c_2 : coefficients d'accélération.
$r_1, r_2 \in [0, 1]$ (générés aléatoirement).
La méthode d'optimisation par essaims particulaires partage beaucoup de similarité avec l'Algorithme Génétique dans le sens où les caractéristiques d'un individu sont influencées par les caractéristiques des autres.
PSO est efficace sur un grand type de problèmes, sans changement de la structure de base de l'algorithme. Parmi ses inconvénients, on trouve le problème de convergence prématurée, qui peut conduire ces algorithmes de tomber dans un optimum local, et le nombre élevé des paramètres, les performances de la méthode suivant un problème donné étant fortement liées aux valeurs de ses paramètres de réglage, il est souvent difficile et long d'obtenir les valeurs optimales de chacun des paramètres.

5. Recherche par Dispersion

Cette stratégie évolutionnaire (Scatter Search) a été introduite par l'auteur Glover ([Glover 1977]). L'algorithme (Algorithme 1.10) commence par une population initiale de solutions dont laquelle un sous-ensemble de solutions est choisi pour un ensemble de références *ensRef* qui évolue par les mécanismes d'intensification et de diversification. Les solutions de cet ensemble sont reliées pour générer des nouvelles solutions mettant à jour l'ensemble de références (la liaison est guidée, elle n'est pas aléatoire contrairement à l'Algorithme Génétique). L'ensemble de références dans l'algorithme de Recherche par Dispersion est de taille petite que la taille des populations dans les algorithmes évolutionnaires.

Algorithme 1.10 Recherche par Dispersion

```
- Initialiser une population (InitPOP);
- Générer un ensemble de référence (RefSet);
repeat
    repeat
        Choix d'un sous-ensemble (subset);
        Combinaison (subset , s);
        Développement (s, s+);
    until le critère 01 est vérifié;
    Mise à jour ensemble de référence (refSet);
until le critère 02 est vérifié;
```

La méthode fournit non seulement une solution heuristique unique, comme les autres métaheuristiques, mais aussi un ensemble réduit de solutions dispersées de haute qualité.
Il s'agit d'une méthode qui utilise plusieurs solutions de haute qualité (du point de vue du critère à optimiser). Partant de l'une de ces solutions, on crée un chemin, par modifications successives de ses composantes, qui relie cette solution aux autres que l'on a retenues. Un critère de choix doit être défini pour décider, à chaque étape, la composante qui doit être ajoutée.

6. Optimisation par Essaims de Vers Luisants

Le ver luisant est le nom commun pour les différentes collections de larves d'insectes et de femelles larviformes adultes qui brillent par bioluminescence.
Les vers luisants se trouvent en Australie et en Nouvelle-Zélande [Oliver *et al.* 2018]. Contrairement aux coléoptères luminescents comme la luciole, dont la bioluminescence a été bien caractérisée, les spécificités moléculaires de la bioluminescence du ver luisant sont restées insaisissables. Les larves attirent leurs proies par la lumière bleu-vert provenant d'un organe lumineux situé à l'extrémité postérieure du corps.
L'algorithme Optimisation par Essaims de Vers Luisants (GSO) a été proposé par les auteurs ([Krishnanand & Ghose 2005a]. [Krishnanand & Ghose 2005b]. [Krishnanand *et al.* 2006].
[Krishnanand & Ghose 2009]). Aléatoirement, un essaim de vers luisants est dispersé dans l'espace de recherche. Les individus de l'algorithme du ver luisant chargent avec eux une quantité de luminescence appelée luciférine.
Chaque individu est attiré par la lueur plus brillante des individus voisins. Un ver luisant désigne un autre ver luisant en tant que voisin, lorsqu'il se situe dans le même domaine local de décision . L'intensité luciférine des vers luisants est liée à la forme de leurs positions réelles. La meilleure position du ver luisant dans l'espace de recherche est d'autant plus importante que la quantité de luciférine est importante.

Algorithme 1.11 Optimisation par Essaims de Vers Luisants

Initialiser n ; //Nombre des vers luisants
$S = S_0$; // taille de pas
$iter_{max} = iter$; //(Nombre maximal des itérations)
for <*i=1 :n*> **do**
  $l_i(0) = l_0$;
  $r_d^i = r_0$
  Aléatoirement initialiser la position $x_i(t)$de chaque ver luisant dans l'espace de recherche.
  Calculer la valeur de la fitness $f(x_i)$des vers luisants
end
Initialiser la position optimale actuelle x^* et la valeur optimale actuelle fx^* selon la valeur de fitness.
$it = 0$;
while $it < iter_{max}$ **do**
  for <`chaque ver luisant i`> **do**
    Mise à jour de luciferin(1.3) ;
  end
  for <`chaque ver luisant i`> **do**
    Définir l'ensemble de voisinage $N_i(t)$(1.5) ;
    Calculer la probabilité du mouvement $p_{ij(t}$(1.4) ;
    Choisir le voisinage j par probabilité Ver luisant i déplace vers j(1.6) ;
    Mise à jour de la plage de voisinage (1.7) ;
  end
  $it = it + 1$;
end

$$l_j(t+1) = (1-\mathcal{P})l_j(t) + \sigma J_j(t+1) \tag{1.3}$$

Où,
$\mathcal{P}$: Luciférine constante (0<P< 1)
σ : Luciférine améliorée constante
$J_j(t)$: Valeur de la fonction « objectif » à l'emplacement j à l'instant t

$$p_j(t) = \frac{l_j(t) - l_i(t)}{\sum_{k \in N_i(t)} l_k(t) - l_i(t)} \tag{1.4}$$

Où

$$j \in N_i(t), N_i(t) = \{j : d_{i,j}(t) < r_d^i(t); l_(t) < l_j(t)\} \tag{1.5}$$

t : Itération.
$d_{i,j}(t)$: Distance euclidienne entre les vers luisants i et j à l'itération t.
$l_j(t)$: Valeur luciférine de j à l'itération t.
$r_d^i(t)$: Plage de décision locale de i à l'itération t.
r_s : Plage de radians du capteur de luciférine.

$$x_i(t+1) = x_i(t) + s(\frac{x_j(t) - x_i(t)}{||x_j(t) - x_i(t)||}) \tag{1.6}$$

$$r_d^i(t+1) = min\{r_s, max[0, r_d^i(t) + \beta(n_t - |N_i(t)|)]\} \tag{1.7}$$

Où, β : Paramètre constant.

Cette stratégie est basée sur le contenu des vers luisants. Un ver luisant qui donne plus de lumière (haute luciférine) désigne qu'il est plus proche à une position réelle et il a une valeur de fonction « objectif » élevée [Alboaneen *et al.* 2016].

7. Algorithme des Lucioles

L'algorithme des lucioles (Firefly) est une métaheuristique, bio-inspirée, introduite par Yang en 2009 ([Yang 2009]). Il est basé sur le principe d'attraction entre les lucioles et simule le comportement d'un essaim de lucioles dans la nature.

Firefly représente un algorithme itératif basé sur une population composée de nombreuses lucioles pour trouver une solution à un problème d'optimisation donné. La communication entre les lucioles est réalisée à l'aide d'une lumière bioluminescente pour l'exploration de l'espace des fonctions. La solution peut être une luciole qui brille proportionnellement à sa qualité dans un domaine donné, chaque luciole plus brillante attire ses partenaires, et par la suite l'exploration de l'espace de recherche est plus efficace.

Trois règles à suivre pour décrire l'algorithme Firefly [Yang 2009] :

1) Une luciole sera attirée par les autres lucioles, quel que soit leur sexe ;

2) L'attractivité est proportionnelle à la luminosité de la luciole, s'il y a deux lucioles clignotantes, la moins brillante se dirigera vers la plus brillante.

3) La luminosité d'une luciole est déterminée par la fonction « objectif ».

Algorithme 1.12 Algorithme des Lucioles

1) Fonction « objectif » : $f(x), x = (x_1, x_2, \ldots, x_d)$
2) Générer une population initiale de lucioles $x_i (i = 1, 2, \ldots, n)$;
3) Formulez l'intensité lumineuse I de sorte qu'elle soit associée à f(x)
4) Définir le coefficient d'absorption γ
while $t < MaxGeneration$ **do**
  for $<i = 1 : n$ `(toutes les n lucioles)`$>$ **do**
    for $<j = 1 : i(nlucioles)>$ **do**
      if $I_j > I_i$ **then**
        Varier l'attractivité avec la distance r via $exp(-\gamma r)$;
        déplacer la luciole i vers j ;
        Évaluer de nouvelles solutions et mettre à jour l'intensité lumineuse ;
      end
    end
  end
  Classer les lucioles et trouvez le meilleur actuel ;
end
Visualiser les résultats ;

L'algorithme est similaire avec d'autres approches basées sur l'in-

telligence collective du groupe, tel que l'algorithme d'optimisation par essaims particulaires et l'algorithme d'optimisation par colonies d'abeilles. Selon des travaux récents, l'algorithme Firefly est plus performant dans la résolution des problèmes d'optimisation que d'autres algorithmes, tel que les algorithmes génétiques [Yang 2009].

8. Programmation Génétique

L'algorithme de programmation génétique (GP) a été proposé par l'auteur Koza [Koza 1992]. Il s'agit d'un algorithme évolutif utilisé pour créer et développer des programmes informatiques. Cet algorithme est inspiré du processus évolutif biologique, où les solutions sont développées sur la base du principe darwinien de survie et de reproduction des plus aptes et les opérateurs génétiques sont similaires aux opérateurs biologiques.
Une population de programmes constitue est composée d'un ensemble de candidats, qui sont combinés avec des programmes pour produire de nouvelles solutions puissantes. Cette opération est répétée pendant plusieurs générations jusqu'à parvenir à une bonne solution. Cet algorithme a été utilisé pour résoudre de nombreux problèmes difficiles tels que la classification.
L'algorithme 1.13 de programmation génétique de base (GP) :

Algorithme 1.13 Procédure GP

```
t = 0;
Aléatoirement, initialiser population P(t);
Fitness P(t);
while critère d'arrêt n'est pas vérifié do
    t = t+1;
    cas de (i)
    01 : Reproduction P(t);
    02 : Croisement P(t);
    03 : Mutation P(t);
    Fitness P(t);
end
```

Une implémentation du processus complet de GP se présente comme suit :
Étape1 : une population initiale de taille N de programmes générés aléatoirement comprenant des fonctions et des terminaux appropriés au domaine problématique est créée.
Étape2 : chaque solution de la population est mesurée par une valeur de fitness (informations de confiance) en termes de performance dans l'environnement de classification.
Étape3 : la prochaine génération est produite à l'aide des opérations génétiques :
cas1.
Opérateur de reproduction : la reproduction utilise le principe de la sélection naturelle et de la survie du plus apte, un individu est sélectionné avec une sélection basée sur des proportions de fitness dans la population actuelle et il est copié dans la nouvelle population.
cas2.

Opérateur de croisement : cet opérateur a une fonction vitale dans le processus évolutif, deux individus sont choisis. Deux individus avec la meilleure valeur de fitness sont sélectionnés, avec une sélection basée sur les proportions de fitness, pour l'opération de croisement. Ensuite, on prend au hasard un sous-arbre de chacun des individus choisis et échangeons ces deux sous-arbres. Ces deux descendants s'ajoutent à la nouvelle population.
Cas3.
Opération de mutation : la mutation maintient la diversité, un individu est choisi au hasard. Maintenant, un sous-arbre de la solution sélectionnée est choisi au hasard et remplacé par un nouveau sous-arbre généré d'une façon aléatoire. Cet individu muté est autorisé à survivre dans la nouvelle population.
Étape4 : les étapes 2 et 3 sont répétées jusqu'à la solution souhaitée, sinon, un critère de l'arrêt est vérifié comme le nombre maximal de générations prédéfini.
Si le critère de l'arrêt est vérifié, le processus est terminé et l'individu en forme est obtenu comme solution.

9. Communication neuronale

a) **Communication neuronale naturelle**

i) **Système nerveux**

Le système nerveux est similaire à un large réseau de communication à travers tout l'organisme. Les nerfs transportent des messages de tous les éléments du corps au cerveau sous forme de signaux électriques. La production, la supervision et la manipulation des signaux nerveux sont parmi les fonctions de ce système.
Le système nerveux assure un contact entre les deux types d'organes les récepteurs sensoriels et les effecteurs, et il combine d'une manière automatique ces deux éléments par les nerfs après que le message a été en avance intégré par des centres nerveux, où sont traitées les informations. Le système est organisé comme suit :
Récepteurs sensoriels : qu'ils soient externes ou internes, le rôle de récepteur est de prendre le stimulus et de le transformer en message nerveux.
Fibres nerveuses : ils garantissent la transmission du message nerveux. On trouve des fibres spécifiques dans la conduction de messages arrivant (ou partant du)au centre nerveux.
Centres nerveux : ils accueillent des messages nerveux et développent une réponse appropriée.
Effecteurs : ce sont des muscles ou des glandes, qui établaient la réponse après l'accueil du message nerveux arrivant des centres nerveux.

Le système nerveux s'exprime en réseaux neuronaux (Figure1.1), qui transmettent le message nerveux de cellule en cellule par des points de liaison nommés des synapses, et ce jusqu'aux cellules effectrices.

ii) **Neurones**

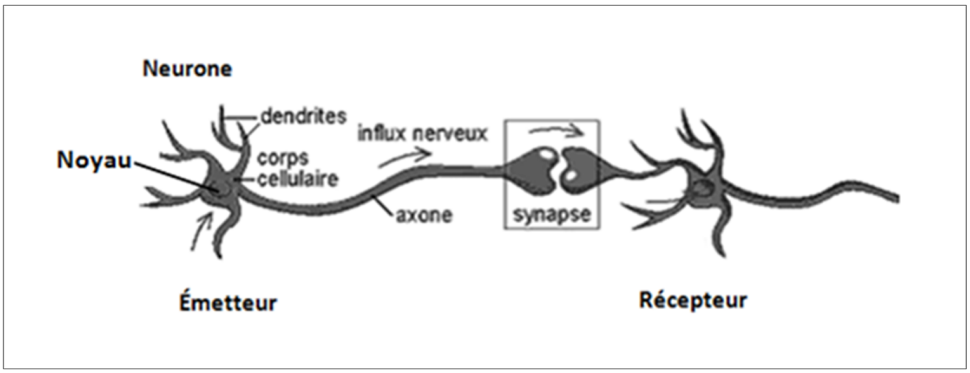

FIGURE 1.1 – *Schéma des connexions neuronales [DOCTISSIMO 2019].*

Un neurone est une cellule nerveuse dont le rôle est de raccorder les messages nerveux. Il reçoit et transmet les messages d'une cellule à l'autre d'une manière spécifique qui est connexe à sa forme. Les neurones disposent les trois éléments de base : l'axone, le corps cellulaire et les dendrites.
Les dendrites sont de petits branchements autour du corps cellulaire. Ils dirigent l'influx nerveux vers ce dernier.
Le corps cellulaire est l'élément central du neurone, borné par la membrane cellulaire et comportant le cytoplasme et le noyau.
L'axone est l'élément le plus long d'un neurone. C'est par cette forme que l'influx est diffusé à l'extérieur du corps cellulaire. Il contient une gaine isolante nommée gaine de myéline. Les terminaisons nerveuses se situent au bout de l'axone.

iii) **Propriétés du message nerveux**
On trouve ces caractéristiques :
Signaux électriques se transmettant rapidement
La stimulation d'un récepteur sensoriel nous autorise à remarquer l'envoi d'un message nerveux sous forme de signaux électriques, dont les propriétés sont liées à l'intensité et à la durée de la stimulation.
Le potentiel complet du nerf correspond à la totalité des potentiels d'actions émises au niveau des neurones et dans les fibres nerveuses employées lors de la transmission du message. Ce message est codé en modulations de la fréquence d'émission des potentiels d'action au niveau des neurones. Le neurone est imposé à la loi du tout ou rien.
Transmission entre neurones
On remarque la présence de synapses au niveau de l'arborisation terminale des neurones, zones de transport unidirectionnel du message nerveux d'un neurone à une autre cellule.
Dans le cas de la transition d'un neurone à un autre, la propagation de l'information se fait d'une manière chimique : les neurotransmetteurs sauvegardés à l'extrémité d'un neurone présynaptique sont libérés à l'arrivée d'un potentiel d'action et transfèrent l'information au neurone postsynaptique.
Traitement de l'information au niveau des centres nerveux
Les connexions entre neurones sont plus complexes dans les

centres nerveux. Elles constituent des réseaux neuroniques, dans lesquels le message peut être diffusé en :
Circulation divergente : si un neurone présynaptique crée des synapses avec de nombreux neurones postsynaptiques.
Circulation convergente : si plusieurs neurones présynaptiques créent des synapses avec un neurone postsynaptique.
Au niveau des centres nerveux, il existe un processus d'intégration, et on remarque que le message est changé d'une manière légère à ce niveau, avant d'être envoyé aux effecteurs. Il y a addition, comparaison et suppression de données avant la transmission des ordres qui sont également codés sous forme messages nerveux.

b) **Algorithme de Communication Neuronale**
Algorithme de Communication Neuronale (NCA) utilise la communication neuronale inspirée par le comportement des neurones biologiques, NCA a été présenté par les auteurs [Ardalan-Asl 2017]. La convergence de l'algorithme de communication neuronale au maximum global (minimum) peut être améliorée par des neurones dotés de connexions efficaces. Chaque neurone a une communication neuronale avec ses voisins, il reçoit et analyse les données provenant de ses voisins. Si la conséquence finale est adéquate, elle sera transmise aux voisins immédiatement connectés. Converger vers le maximum global (minimum) est l'objectif de l'approche NCA. Pour cela, les neurones changent d'emplacement vers un nouvel emplacement plus performant, c'est-à-dire les neurones n'ont pas d'emplacement fixe.
Au premier pas de l'algorithme, les neurones sont distribués aléatoirement dans l'espace de recherche et chacun des neurones est lié à des voisins. Après, la valeur de la fonction fitness de tous les neurones est calculée. En fait, à chaque itération, si l'algorithme NCA peut trouver la nouvelle position des neurones, l'algorithme convergera vers le maximum global (minimum). Dans NCA et pour le mouvement des neurones, chaque neurone peut renseigner les voisins de sa position et de la valeur de sa fonction fitness.
L'interaction entre chaque neurone et la population neuronale se fait selon la procédure suivante :
- Les voisins envoient la position et la valeur de la fonction de fitness au neurone.
- Si la valeur de la fonction fitness du $j^{ième}$ voisin est inférieure (supérieure) à sa contrepartie du $i^{ième}$ neurone, celui-ci aura tendance à s'approcher du $j^{ième}$ voisin.
- Si la valeur de la fonction fitness du $j^{ième}$ voisin est supérieure (inférieure) à sa contrepartie du $i^{ième}$ neurone, celui-ci aura tendance à s'éloigner du $j^{ième}$ voisin.
Le $i^{ème}$ neurone se déplace d'une position avec une valeur de fonction de fitness supérieure (inférieure) à une position avec une valeur inférieure (supérieure).
NCA est capable de diriger le neurone vers la meilleure position en utilisant la fonction pondérée en accord à la valeur de fonction de fitness de ses voisins.
À chaque itération de l'algorithme, les nouveaux voisins d'un neurone identifié doivent être sélectionnés selon une fonction aléatoire pondé-

rée. En fait, la sélection aléatoire réduit la possibilité de se piéger dans les minima locaux et obtient de nouvelles informations, ces dernières aident à changer le chemin et à converger vers le minimum global.
L'algorithme NCA (Algorithme1.14) est représenté en quatre sous-programmes, il permet de diriger les neurones au minimum (maximum) global. À chaque itération et dans le premier pas, les nouveaux voisins du $i^{ème}$ neurone sont sélectionnés aléatoirement. Ensuite, la nouvelle position du $i^{ème}$ neurone est obtenue en utilisant la valeur de la fonction de fitness de ses voisins. Par conséquent, si la nouvelle position est meilleure que la précédente, il sera considéré comme la nouvelle position du $i^{ème}$ neurone.

Algorithme 1.14 Algorithme de Communication Neuronale

Entrer les paramètres et Générer aléatoirement la population pop
while *critère d'arrêt n'est pas vérifié* **do**
  Fonction de groupe de guidage
  for `<i=0 ; i< |pop| ; i++>` **do**
    Sélectionner les voisins
    Génération de Nouveaux agents
    Contrôler résultat, si $f_{nouvelle} < f_{ancienne}$
  end
  Générer aléatoirement des neurones
end

1.6 Métaheuristiques hybrides

Une méthode hybride est une méthode de recherche basée sur au moins de deux approches de recherche. On exploite les avantages de plusieurs méthodes en associant leurs algorithmes en suivant une approche collaborative.
Un algorithme hybride peut être bon ou non selon le choix et les rôles de ses éléments. Pour la construction d'une méthode hybride puissante, il faut connaître les avantages et les inconvénients de chacune de ces méthodes. Les méthodes hybrides associent les différents éléments et notions de métaheuristiques différentes et à cette fin, ils essayent de combiner les concepts forts et évitent les points faibles de ces métaheuristiques. Par conséquent, L'efficacité de l'espace de solution recherchée peut être encore développée et de nouvelles occasions survenues ce qui peut diriger vers des méthodes de recherche plus efficaces et plus flexibles.
Les origines des algorithmes hybrides reviennent probablement aux travaux présentés dans ([Mühlenbein *et al.* 1988]. [Grefenstette 1987]. [Glover & Laguna 1997]). Les auteurs ont facilement prouvé l'efficacité d'intégration d'une méthode de descente à l'intérieur d'une méthode évolutive ([Mühlenbein *et al.* 1988]. [Grefenstette 1987]).
Les auteurs [Glover & Laguna 1997] ont présenté une méthode de descente simple pour développer une recherche évolutive. Mais à cette époque, la majorité des auteurs n'y donnait que peu d'intérêt.
Plusieurs recherches ont été effectuées dans ce domaine, qui devient de plus en plus important ([Cotta 1998]. [Blum *et al.* 2005]. [Cotta *et al.* 2005].

[Puchinger & Raidl 2005]. [Talbi 2002]).
Les meilleurs résultats sont trouvés en combinant un algorithme évolutionnaire avec un algorithme de recherche local, la GRASP, [Khao *et al.* 1999].
L'hybridation a pour buts :
- Développer les performances d'un algorithme évolutionnaire (vitesse de convergence) ;
- Augmenter la qualité de la solution trouvée ;
- Enrichir le domaine d'application des algorithmes évolutionnaires.
Plusieurs hybridations sont présentées dans la littérature combinant un algorithme évolutionnaire avec un autre algorithme évolutionnaire [Grosan & Abraham 2007] , un réseau de neurones, un algorithme d'optimisation par essaims particulaires, un algorithme d'optimisation par colonies de fourmis, un algorithme la recherche locale, la recherche tabou, le recuit simulé, la programmation dynamique, et d'autres.

1.6.1 Classification de Métaheuristiques hybrides

On peut classifier les différentes méthodes d'hybridations selon la taxonomie introduite par l'auteur ([Talbi 2013]. [Talbi 2009]). On compare les algorithmes hybrides d'une manière qualitative. Il y a deux types (Figure 1.2), une classification hiérarchique permet d'abord de définir la forme de l'hybridation. Et une classification générale détermine les détails des algorithmes nécessaires dans l'hybridation.

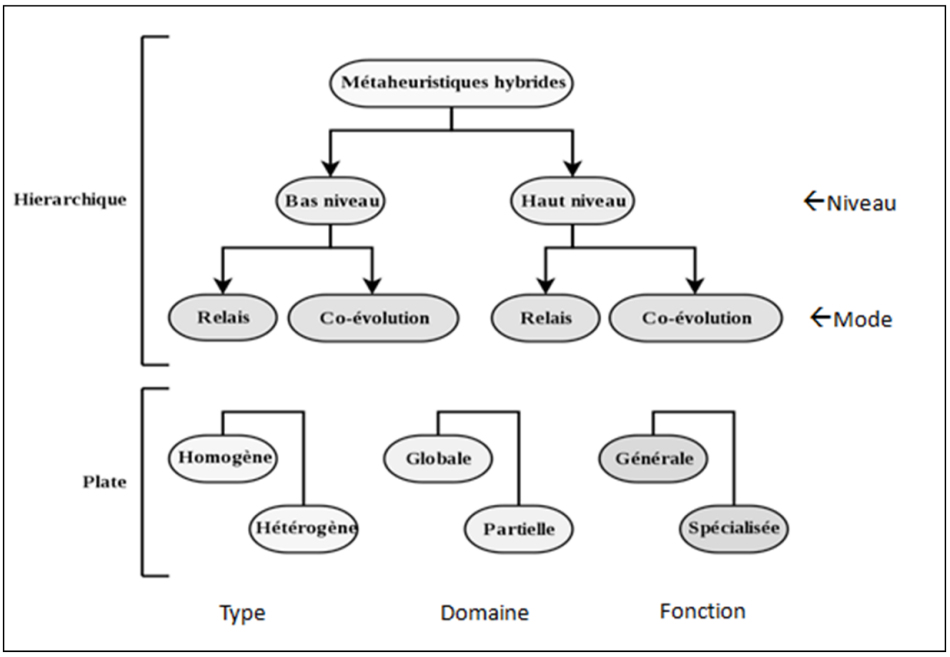

FIGURE 1.2 – *Classification des Métaheuristiques Hybrides [Talbi 2013].*

1) **Classification hiérarchique**
La classification hiérarchique est partitionnée en deux classes : l'hybridation de bas niveau et l'hybridation de haut niveau. On a une hybridation de bas niveau lorsqu'une fonction d'une métaheuristique est

substituée par une autre métaheuristique ou méthodes exactes. On a une hybridation de haut niveau lorsque deux métaheuristiques sont hybridées sans que leur processus interne ne soit en liaison.
Chacune des deux classes d'hybridation précédentes est représentée en deux autres branches : à relais et co-évolutionnaire. L'hybridation co-évolutionnaire se fait lorsque des agents collaborent en parallèle pour l'exploration de l'espace de solutions. Lorsque les métaheuristiques sont effectuées d'une manière séquentielle, l'une employant le résultat de la précédente comme entrée, on a une hybridation à relais.
Par suite la classification hiérarchique fournit les classes suivantes :

a) **Hybridation de bas niveau à relais (Low-level Relay Hybrid (LRH)) :**
représente les algorithmes dans lesquels une métaheuristique est intégrée dans une autre métaheuristique à solution unique.
La classe d'hybridation relais de bas niveau (LRH) est plus efficace en ayant une métaheuristique hybridée avec une méthode exacte. Un exemple de ce type est proposé par Les auteurs [Augerat *et al.* 1998], ils ont proposé d'utiliser un algorithme de Branch and Cut et une recherche Tabou pour résoudre le problème de tournée de véhicule avec contraintes de capacité.

b) **Hybridation de bas niveau co-évolutionnaire (Low-level Teamwork Hybrid (LTH)) :**
consiste à intégrer un algorithme de recherche locale orienté vers l'exploitation (pour améliorer une solution) dans une métaheuristique à population orientée vers l'exploration (pour examiner plus de solutions).
Les auteurs [Elbenani *et al.* 2012] ont utilisé cette forme d'hybridation pour résoudre le problème de partition de graphe en remplaçant l'opérateur de mutation d'un algorithme génétique par une recherche Tabou.

c) **Hybridation de haut niveau à relais (High-level Relay Hybrid (HRH)) :**
Les métaheuristiques complètes travaillent en séquentiel. Dans la forme d'hybridation relais de haut niveau (HRH), les métaheuristiques et les méthodes exactes hybridées sont réalisées séquentiellement en conservant leur caractéristique. Les auteurs [Bent & Van 2004] ont appliqué cette hybridation pour résoudre le problème de tourné de véhicule en diminuant le nombre de véhicules par le recuit simulé puis en améliorant la fonction « objectif » de chaque tournée par une recherche exacte dans un voisinage.
Les auteurs [Applegate *et al.* 2007] ont proposé une coopération séquentielle pour résoudre le problème du voyageur de commerce en appliquant une recherche locale pour produire un ensemble de solutions puis un algorithme exact pour indiquer la solution optimale.

d) **Hybridation de haut niveau co-évolutionnaire (High-level Teamwork Hybrid (HTH)) :**
les métaheuristiques complètes sont exécutées en parallèle et en collaboration pour obtenir la solution optimale d'un problème avec un échange d'informations indispensable entre elles.

Le modèle d'île [Tanese 1987] est le premier modèle hybride qui a été proposé pour les algorithmes génétiques, la population est groupée en sous populations par séparation géographique. Pour cette hybridation, la population est partitionnée en sous-populations réparties sur les nœuds d'un hypercube dont lesquels un algorithme génétique est démarré. Chaque nœud est associé à une partie de recherche de solutions et périodiquement, des solutions migrent entre les nœuds dont le but est d'obtenir des solutions optimales.
L'hybridation co-évolutionnaire de haut niveau (HTH) est difficile à effectuer entre une méthode exacte et une métaheuristique car chaque algorithme résout un problème différent, dans ce contexte, les auteurs [Chabrier *et al.* 2002] ont réalisé une hybridation entre un algorithme de recherche locale et un algorithme de Branch and Price pour résoudre le problème de tournées de véhicules. L'exécution des algorithmes est réalisée en parallèle avec l'échange d'information entre les algorithmes.

2) **Classification générale**
Les méthodes hybrides homogènes versus hétérogènes, globales versus partielles et spécialistes versus générales.

a) **Hybridation homogène et hétérogène**
Une hybridation est dite homogène lorsque les métaheuristiques associées sont similaires. Par contre, une hybridation hétérogène combine des métaheuristiques différentes.
Une approche hétérogène LTH basée sur l'Algorithme Génétique et la Recherche Tabou a été présentée dans [Crainic *et al.* 1997] pour trouver des solutions au problème de conception de réseaux. La population de l'algorithme génétique est mise à jour d'une manière asynchrone par de nombreux algorithmes de recherche Tabou. Les meilleures solutions obtenues par les algorithmes de Recherche Tabou construisent une population élite pour l'Algorithme génétique.

b) **Hybridation globale ou partielle**
Une hybridation est dite globale lorsque toutes les métaheuristiques associées explorent l'ensemble de l'espace de solutions. L'hybridation partielle divise un problème en sous-problèmes ayant leur espace de solutions spécifique, et chaque sous-problème est résolu par un algorithme.
Un modèle général pour un HTH hétérogène Partielle hybride a été proposé dans [Talukdar *et al.* 1998]. Il traite un ensemble de solutions, qui peuvent être globales ou partielles. Un ensemble d'agents collabore à travers un système de tableau noir, une forme de mémoire partagée. Un agent peut être un algorithme de recherche ou un opérateur, l'agent sélectionne une solution partielle du tableau noir, la déplace et retourne le résultat.

c) **Hybridation généraliste ou spécialiste**
Une hybridation est dite générale lorsque tous les algorithmes associés résolvent le même problème d'optimisation. Par contre si chaque algorithme résout un problème d'optimisation différent, les hybridations sont dites spécialistes.

L'approche de l'hybridation spécialiste par la Recherche Tabou parallèle et l'algorithme génétique a été introduite dans[Talbi & Bachelet 2006b] pour résoudre le problème d'affection quadratique (QAP). La Recherche Tabou résout le QAP et l'algorithme génétique effectue la diversification qui est un autre problème d'optimisation pour la génération des solutions à partir d'une mémoire fréquente, cette mémoire sauvegarde les informations liées à toutes les solutions visitées par la Recherche Tabou.

1.6.2 Métaheuristiques parallèles

Les métaheuristiques permettent de trouver des solutions optimales ou sous optimales dans un temps raisonnable, mais il existe des types de problèmes qui consomment un temps élevé, de ce fait le parallélisme est une solution non seulement de ce problème pour réduire le temps d'exécution mais il est important pour améliorer la qualité des solutions obtenues [Cung *et al.* 2002].

1) **Architecture parallèle**
La taxonomie la plus utilisée est la proposition de Flynn [Flynn 1972] faite en 1966 (Figure 1.3).
SISD (single instruction on single data) : l'exécution est séquentielle,

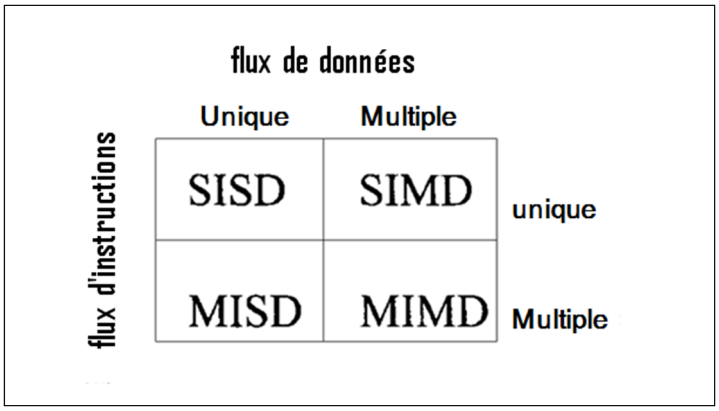

FIGURE 1.3 – *Taxonomie de Flynn [Flynn 1972].*

il n'y a aucun parallélisme au niveau des instructions et au niveau de la mémoire dans l'ordinateur. Cette classe correspond à l'architecture de von Neumann.
SIMD (Single Instruction on Multiple Data) : le parallélisme est réalisé au niveau de la mémoire dans l'ordinateur.
MISD (Multiple instructions single data) : la même donnée est manipulée par plusieurs unités de calcul en parallèle.
MIMD (Multiple Instructions multiple data) : plusieurs unités de calcul manipulent des données différentes, car chacune d'elles possède une mémoire particulière. C'est l'architecture parallèle la plus utilisée.
Dans la classe MIMD, cette taxonomie ne prend pas en considération l'extension de la mémoire sur un espace d'adressage unique ou la répartition de la mémoire entre plusieurs modules.
Une extension de la classification de Flynn est représentée dans la Fi-

gure 1.4 [Alba & Nebro 2005]

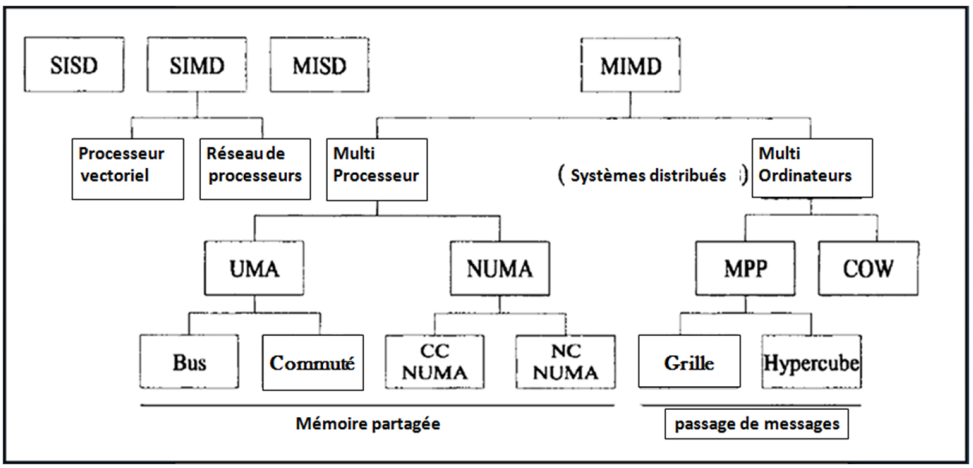

FIGURE 1.4 – *Extension de la taxonomie de Flynn [Alba & Nebro 2005].*

La classe MIMD est divisée en multiprocesseur et multi-ordinateur, on distingue plusieurs architectures pour accéder à la mémoire :
— Systèmes à mémoire partagée
— Systèmes à mémoire distribuée

a) **Mémoire partagée**
Pour augmenter la puissance de calcul avec une mémoire unique, on emploie une architecture parallèle par l'utilisation des processeurs similaires au sein d'un ordinateur, c'est le multiprocesseur symétrique à mémoire partagée (SMP), on trouve les classes suivantes :
— **Accès uniforme à la mémoire** (UMA) : les processeurs ont le même temps d'accès à la mémoire par l'utilisation d'un seul bus.
— **Accès non uniforme à la mémoire** (NUMA) : réduction du temps d'accès à certaines données par l'utilisation d'une mémoire locale avec chaque processeur.
— **Architecture de mémoire cache uniquement** (COMA) : la mémoire centrale est employée comme une mémoire cache.
— **Cache Cohérent NUMA** (ccNUMA)
— **Pas d'accès à la mémoire à distance** (NoRMA) : les processeurs n'ont pas les accès directs à la mémoire, les accès sont effectués par le réseau d'interconnexion vers d'autres processeurs, ce dernier fournit les données demandées dans un message de réponse. La configuration de sauvegarde entière est divisée statiquement entre les processeurs.

b) **Mémoire distribuée**
Un réseau d'interconnexion est créé pour associer les processeurs et leurs mémoires, il peut être représenté par plusieurs structures : carré, grille à deux dimensions, arbre, hypercube, ...
Avec ce type de système, la répartition des données trop volumineuses sur les différentes mémoires des processeurs est possible.

2) **Critères de mesure de performances des métaheuristiques parallèles**
Il existe plusieurs critères pour mesurer l'efficacité des algorithmes parallèles selon les auteurs [Alba & Luque 2005] :

a) **Accélération (Speedup)**
Elle consiste à comparer le rapport entre les temps d'exécution séquentiels et parallèles. La comparaison est faite si l'algorithme parallèle donne au moins la même précision que l'algorithme séquentiel. Le bénéfice apporté par le parallélisme consiste alors à réduire le temps d'exécution de la métaheuristique tout en conservant la même qualité de solution.
Dans un système monoprocesseur, une performance commune est le temps CPU pour résoudre le problème ; c'est le temps pendant lequel le processeur exécute les instructions en excluant le temps d'entrées des données de ce problème, la sortie de résultats et les activités générales du système. Contrairement au cas parallèle qui vise à réduire le temps réel qui est tout le temps des activités réalisées par le système (algorithmes parallèles).
Si l'on désigne par T_m, le temps d'exécution d'un algorithme utilisant m processeurs
Speedup S_m est le rapport entre le temps d'exécution rapide dans un processeur T_1 et le temps d'exécution dans m processeurs T_m
$S_m = T_1/T_m$
Dans le cas des algorithmes non déterministes, on compare le temps moyen d'exécution en série et le temps moyen d'exécution parallèle :
$S_m = M[T_1]/M[T_m]$
L'auteur Alba [Alba & Luque 2005] a proposé une taxonomie de mesure de Speedup (Figure 1.5) qui est proche de [Barr & Hickman 1993].
-L'accélération forte (Strong speedup) compare le temps d'exécution parallèle avec le meilleur temps d'exécution d'algorithme séquentiel. C'est la définition la plus exacte de Speedup, mais en raison de la Difficulté de trouver l'algorithme le plus efficace actuellement, la plupart des concepteurs des algorithmes ne l'utilisent pas.

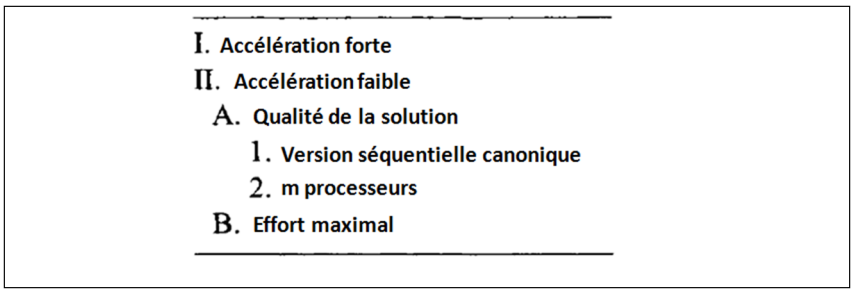

FIGURE 1.5 – *Taxonomie de la mesure de Speedup .[Alba & Luque 2005].*

-L'accélération faible (Weak speedup) compare l'algorithme parallèle développé par un chercheur contre sa propre version série. Dans ce cas, il y a deux critères d'arrêt pour les algorithmes : la qualité de la solution et l'effort maximal. Il y a deux variantes de l'accélération faible : on compare l'algorithme parallèle avec la version séquentielle canonique (Versus panmixia) où on compare le temps d'exécution de l'algorithme parallèle dans un seul processeur

avec le temps d'exécution du même algorithme sur m processeurs (Orthodox).

b) **Efficacité**
Est la normalisation de l'accélération (Speedup), elle permet de comparer les différents algorithmes.
$e_m = S_m/m$ $e_m = 1$ signifie la vitesse linéaire
Il y a plusieurs variantes de la mesure d'efficacité comme l'efficacité incrémentale, les auteurs [Karp & Flatt 1990] ont défini une mesure intéressante de la performance de n'importe quel algorithme parallèle : c'est une fraction série de l'algorithme (eq (1.8)) :

$$f_m = \frac{\frac{1}{s_m} - \frac{1}{m}}{1 - \frac{1}{m}} \tag{1.8}$$

Idéalement, la fraction série doit rester constante pour un algorithme.

3) **Caractéristiques de la stratégie des métaheuristiques parallèles**
Selon l'étude des caractéristiques communes aux différentes applications parallèles retrouvées dans le monde des métaheuristiques, les auteurs ([Verhoeven & Aarts 1995]. [Cung *et al.* 2002]. [Crainic 2005] ont établi une classification à trois dimensions qui se basent sur le contrôle de processus global de résolution de problèmes, les informations échangées entre les processus et la diversification de recherche.
La première dimension exprime si la recherche globale est supervisée par un seul processus ou plusieurs processus qui peuvent être en coopération ou non.
La deuxième dimension concerne l'échange des informations et son application dans le contrôle et le guide de la recherche ; et par conséquent le contrôle de la recherche et la communication. La communication peut être synchrone ou asynchrone. Dans le cas synchrone, à chaque moment donné le processus engage les informations échangées (nombre d'itérations, intervalle de temps, le degré d'algorithme spécifique, . . .). Dans le cas asynchrone, chaque processus réalise ses propres recherches, et assure la communication avec d'autres processus, la recherche finale est faite si toutes les recherches partielles sont satisfaites.
La troisième dimension indique la différentiation ou la diversification de la recherche, si les processus emploient les mêmes solutions initiales ou bien des solutions différentes. Si les processus emploient la même technique de recherche ou bien des techniques différentes.

4) **Différentes Stratégies des métaheuristiques parallèles**

a) **Le parallélisme de bas niveau**
C'est une technique de parallélisme à l'intérieur d'une itération de la métaheuristique. Le parallélisme s'effectue d'une manière relativement directe et a pour but d'accélérer les calculs sans chercher à réaliser une meilleure exploration. L'approche de résolution parallèle est la même que son équivalente séquentielle dans la mesure où le nombre total d'opérations est identique pour les deux versions. À cet effet, il est possible de profiter de la puissance de calcul disponible et de développer ainsi la recherche de solutions sans modifier le principe de base de l'algorithme.

Il s'agit alors d'exécuter un plus grand nombre d'opérations en parallèle tout en gardant un temps d'exécution global égal ou inférieur à la version séquentielle. On trouve l'exemple de la recherche tabou parallèle ([Chakrapani & Skorin-Kapov 1993a] [Chakrapani & Skorin-Kapov 1993b]. [Porto *et al.* 2000]. [Crainic 2005]) pour le problème d'ordonnancement des tâches.

b) **La décomposition de l'espace de recherche**
Elle consiste à partitionner le problème en sous-problèmes et à appliquer la puissance de calcul afin de résoudre ces sous-problèmes en parallèle. Le contenu de recherche est alors modifié par rapport à l'algorithme séquentiel. Un modèle maître-esclave est généralement implémenté pour utiliser cette technique. Le maître détermine séquentiellement les partitions initiales et les échange durant l'exécution à des intervalles qui peuvent être prédéfinis ou déterminés dynamiquement durant l'exécution. Les esclaves explorent d'une manière concurrente et indépendante l'espace de recherche qui a été attribué par le maître et ce dernier construit une solution complète au problème par l'intermédiaire des solutions partielles obtenues par les esclaves. Une meilleure association entre la décomposition de l'espace de recherche et d'autres techniques de parallélisme, la collaboration en particulier est présentée dans [Lahrichi & T.-G.Crainic 2015].

c) **Plusieurs processus de recherche**
Ils sont employés avec différents niveaux de synchronisation et de collaboration, on cherche à réaliser une recherche plus complète en mettant en œuvre plusieurs processus qui opèrent simultanément et qui sont dirigés par différentes manières de coopération et de concurrence. Les processus peuvent communiquer entre eux durant leur recherche ou seulement à la fin pour définir la meilleure solution. Dans cette stratégie il y a deux types : approches indépendantes et approches coopératives.

i) **La multi recherche indépendante**
Plusieurs recherches sont effectuées en parallèle et le meilleur résultat est choisi parmi celles-ci à la fin.

ii) **Une recherche multiple et coopérative** La multi recherche coopérative a été montrée comme une des meilleures métaheuristiques pour la résolution des problèmes d'optimisation difficile [Talbi & Bachelet 2006a]. Les stratégies de recherches coopératives intègrent les techniques de partage de l'information.
Il faut déterminer les paramètres importants :
– La topologie du réseau d'interconnexion qui définit la nature de combinaison des processus.
– La manière de communication entre les processus (diffusion, point à point, mémoire centrale, etc.)
– Le choix des processus participant aux échanges d'informations ;
– La nature de communication (synchrone ou asynchrone) ;
– Les périodes d'échange d'informations ;

– L'information à échanger.
Le mécanisme de coopération fournit les moyens de partager les informations pour la même instance de problème :
La nature des informations à partager
La disponibilité d'une grande quantité d'informations à tous les processus peut produire des coûts de communication et/ou d'accès à la mémoire très chers. Il est possible que certaines informations engendrées par un processus n'apportent rien et même font obstacle aux autres.
L'étude de la qualité des informations à partager est donc très importante. L'appréciation et le savoir du problème sont bien la meilleure manière d'évaluer l'intérêt de partager une telle information.
Le moment où l'information est partagée
Il est important de déterminer le moment où l'information est disponible et accessible par les processus de recherche par des conditions d'accès définies par différentes manières : soit en début de l'exécution ou dynamiquement à chaque évolution de l'algorithme.
Si les processeurs ont un accès limité à l'information partagée, il peut se produire un degré élevé de la performance de recherche de solutions et de diversité et il peut éviter la convergence prématurée de l'algorithme. Par conséquent, il est essentiel d'assurer un équilibre entre l'influence de l'information partagée et celle des paramètres locaux.
Les processus entre lesquels l'information est partagée
Il consiste à définir les règles d'interaction entre les processus et de déterminer le comportement collaboratif de l'algorithme dans la résolution d'un problème d'une manière explicite lors de la conception de l'algorithme. Par établissement des associations logiques qui lient les processus entre eux et qui supervisent la transmission de l'information partagée, on donne la permission aux processeurs pour faire les échanges des informations. L'échange d'informations peut être synchrone ou asynchrone. S'il est synchrone, les processus concernés doivent tous arriver à un certain point de synchronisation avant que l'échange ne soit réalisé et que l'exécution ne soit poursuite par ceux-ci. S'il est asynchrone, l'échange s'exécutera selon la logique interne de chaque processus.
La coévolution
Elle concerne l'évolution simultanée de plusieurs individus (par la compétition ou par la coopération). L'algorithme génétique parallèle en îles est un bon exemple de coévolution où diverses populations indépendantes collaborent en échangeant des individus avec une probabilité ou les meilleurs individus de chaque population qui migrent à un moment donné d'une île à une autre pour remplacer le mauvais d'une autre population. ([Koza 1991]. [Calégari 1999]. [Cantú-Paz 2000]). La compétition peut vérifier la diversification dans la population, éviter la convergence prématurée de l'algorithme et obtenir des

meilleures solutions [Juille & Pollack 1996].

1.7 Conclusion

Les approches hybrides ont permis d'obtenir de bons résultats dans une grande variété des problèmes théoriques d'optimisation combinatoire tels le problème du voyageur de commerce, le problème de la coloration du graphe, le problème d'affectation quadratique, le problème de la tournée des véhicules, le séquençage d'ADN ou encore le calcul des trajectoires des satellites. L'hybridation des métaheuristiques donne des résultats intéressants dans le domaine de l'extraction des connaissances à partir des données tels les problèmes de la sélection d'attributs, la sélection des instances, la classification supervisée, la recherche des règles d'association et la segmentation. On va présenter ces tâches dans le chapitre suivant.

Processus d'extraction des connaissances à partir de données

2

Sommaire

2.1 Introduction

L'objectif de processus d'extraction des connaissances à partir des données (KDD) est de trouver des modèles clairs à partir de données. Le majeur challenge est de traiter un grand volume de données de sorte que le processus doive être entièrement ou partiellement automatique et d'utiliser des techniques permettant d'analyser les données et d'extraire des connaissances utiles.
La réalisation d'un processus automatique est difficile à cause de la complexité des données, ainsi que d'autres problèmes tels que les valeurs manquantes, le bruit, et l'imprécision.
Les données recueillies, données prétraitées, modèles, la connaissance et les différents éléments de KDD sont présentés dans ce chapitre.

2.2 Extraction des connaissances à partir des données (KDD)

La découverte de connaissances est un processus non trivial d'extraction des informations implicites, cachées, et potentiellement utiles au sein d'entrepôts de données volumineux et à grande dimensionnalité ([Han & Cerconet 1992]. [Frawley *et al.* 1991]). La non-trivialité réfère au fait que la découverte de connaissances passe par plusieurs étapes, parmi elles, la fouille de données qui est exploratoire [Wijsen 2001]. Avec le KDD, on ne sait pas a priori ce qu'on pourrait apprendre des données. Cette connaissance a priori se distingue par les résultats mis à nus qui sont plutôt cachés.
Les résultats d'une fouille de données devraient être non seulement bénéfiques mais compréhensibles par les utilisateurs du domaine et servir de support au processus de décision. L'entrepôt comporte un volume intéressant de données et ces données sont décrites par plusieurs attributs.
Le terme KDD a été inventé lors du premier atelier KDD en 1989 pour indiquer que la "connaissance" est le résultat final d'une découverte basée sur des données. Il a été généralisé dans l'intelligence artificielle et l'apprentissage automatique [Fayyad *et al.* 1996].
Il est nécessaire d'indiquer que le KDD est un ensemble des phases et d'actions dont le but est l'extraction des tendances et des corrélations au sein des données, le processus KDD n'est pas défini exclusivement par la fouille de données [Miller & Jiawei 2001] qui en représente une fraction visible du processus. Ce dernier constitue en effet d'un ensemble des phases allant de l'intégration, la collecte de données, la sélection, le nettoyage et la transformation à l'évaluation, l'interprétation et la diffusion des résultats de la fouille en passant par la fouille de données elle-même [Fayyad *et al.* 1996], en ce sens le processus d'extraction des connaissances à partir de données est interactif et itératif.
- Par itératif, il faut comprendre que le KDD n'est pas un processus linéaire où chaque phase est implémentée une seule fois pour amener à la fin avec le modèle de connaissance recherché cela pourrait être le cas dans le meilleur des cas probables mais cela parvient exceptionnel dans la pratique.

- Le KDD est un processus où l'homme est au centre du processus d'où le qualificatif interactif. Les outils de fouille ne sont pas des robots qui seuls doivent parcourir d'un volume important de données afin d'y sortir quelques informations utiles à l'organisation [Fayyad 1998], bien au contraire, il s'agit d'un ensemble d'interactions entre l'utilisateur et les outils de fouille afin que les résultats trouvés au bout du processus soient non seulement compréhensibles mais utiles.

2.2.1 Description du Processus KDD

Le processus d'extraction des connaissances à partir de données montré en (Figure 2.1) est structuré en ces étapes ([Brachman & Anand 1996]. [Fayyad *et al.* 1996]) :

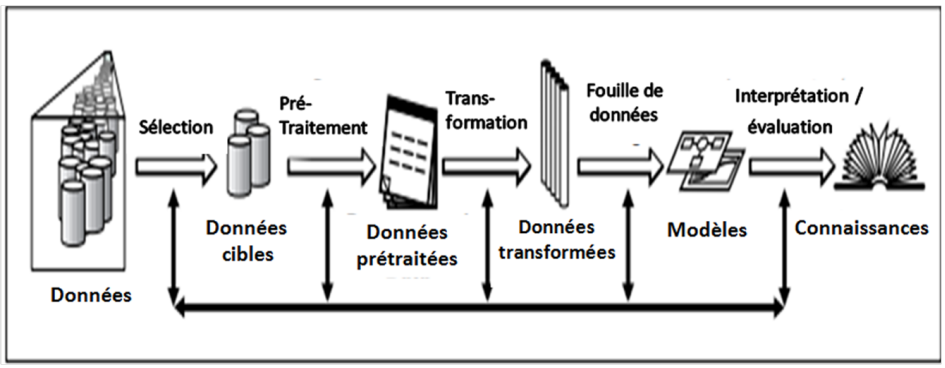

FIGURE 2.1 – *Étapes du processus KDD [Fayyad et al. 1996].*

1. **Identification de l'objectif du processus KDD**
   Elle consiste à développer le domaine d'application ainsi que les connaissances a priori et à définir le but du processus KDD d'un point de vue des utilisateurs.
2. **intégration et collecte de données**
   Elle consiste à créer un ensemble de données cibles relatives au système de l'étude, en unique source : l'entrepôt de données pour simplifier les différentes manipulations. Ces données sont le plus fréquemment hétérogènes. Cette étape se base sur la collecte des données nécessaires pour réaliser la découverte.
3. **sélection, nettoyage**
   Elle consiste à choisir, dans l'entrepôt, les données qui seront prises pour établir le modèle. Le nettoyage de données repose sur la gestion de la qualité des données et plus particulièrement les imprécisions et les incertitudes qu'elles peuvent contenir. Le développeur pourra retenir de corriger ou d'ignorer les données manquantes et erronées. Une donnée est un enregistrement au sens des bases de données, un individu au sens des statistiques ou instance au sens informatique. Elle est distinguée par un ensemble d'attributs.
   En fonction de l'ensemble des valeurs que l'attribut peut prendre, un attribut est qualitatif si on ne peut pas en faire une moyenne,

sa valeur est d'un type déterminé en extension (une couleur de voiture,...) sinon, l'attribut est quantitatif : un entier, un réel, ...; il peut constituer un salaire, une surface, un nombre d'habitants, ...; on peut par suite implémenter les opérateurs arithmétiques habituels sur les attributs quantitatifs, ce qui n'est pas le même pour des attributs qualitatifs. Un attribut peut également être formé par d'autres attributs (une date est composée de jour, mois et année), sur lequel on peut déterminer les opérateurs arithmétiques usuels : donc quantitatif n'indique pas forcément numérique et, réciproquement, numérique n'indique pas forcément quantitatif : un code postal est numérique, mais pas quantitatif.
Si les valeurs sont des symboles (des noms) dont aucune liaison (ordre ou distance) se trouve entre les nominaux alors les attributs sont à valeurs nominales.
Dans les attributs à valeurs ordinales, il n'est pas possible de faire le calcul direct des distances entre les valeurs ordinales, les opérations arithmétiques ne sont pas faisables.
Dans les attributs de type intervalles, les valeurs sont calculées dans des unités particulières et valides. Les opérations arithmétiques de deux intervalles ne sont pas faisables car le point zéro n'existe pas, par exemple : la température est définie en degrés Celsius ou Fahrenheit.
Dans les attributs de type rapport : toutes les opérations arithmétiques sont permises sur les attributs de cette nature. L'étape de sélection, nettoyage comprend ces tâches :
La détection des valeurs erronées.
Les valeurs de données introduites sont incohérentes s'il y a des erreurs, des manques, des duplications. Selon les tâches qu'on va exécuter après la découverte de ces fautes, soit à prendre en compte ou à exclure.
Le traitement des problèmes de mise à échelle.
Le problème de la mise à l'échelle survient lorsqu'une énorme quantité de données doit être manipulée, en surmontant la puissance des algorithmes classiques de fouille de données [Derrac *et al.* 2010].
Le traitement de problème des ensembles de données déséquilibrés
Ce problème survient lorsque la répartition des classes dans les données d'apprentissage n'est pas équilibrée. Par exemple, le nombre d'instances de certaines classes est trop petit, ce qui pose un problème lors de classement des instances appartenant à ces classes [Derrac *et al.* 2010].
Le traitement des valeurs manquantes
Les valeurs manquantes sont inconnues, non importantes, non sauvegardées,..., elles doivent être manipulées particulièrement, une valeur manquante doit être signalée et ajoutée, il est important de les garder car elles expriment qu'il y a un problème dans les données par exemple un examen médical et parfois, on peut ignorer les données avec les valeurs manquantes.
L'échantillonnage
Si l'ensemble des données est volumineux, on peut échantillonner

les données. Cela permet d'accroître l'efficacité du processus de la fouille de données.

La sélection d'attributs

L'exploitation des données de taille importante est difficile en pratique, la sélection d'attributs est une phase de prétraitement qui permet de développer l'efficacité du processus et de faciliter de la compréhension du problème à étudier. Par exemple, en présence d'attributs non pertinents les auteurs [Kohavi & John 1997] ont prouvé une dégradation des performances par l'analyse du comportement de différents modèles de classification sur des bases de données artificielles.

La transformation

Elle facilite les tâches de la fouille de données. Les divers algorithmes de fouille de données sont contraignants sur la forme des données qu'ils acceptent. La transformation repose sur la préparation des données brutes et la conversion en données appropriées. Les opérations de transformation les plus célèbres sont la discrétisation des variables continues, la binarisation des variables nominales, l'agrégation de données.

La discrétisation de variables continues consiste à transformer un attribut continu en partitionnant son domaine en intervalles finis, le domaine de l'attribut transformé devient un ensemble discret.

L'agrégation d'un attribut est un type de transformation en utilisant une règle ou une équation. Par exemple, on veut analyser les salaires annuels des fonctionnaires, on a seulement des salaires mensuels, un nouvel attribut agrégat serait le salaire multiplié par douze.

4. **Méthode de fouille de données particulière**
   Elle consiste à définir la méthode utilisée : classification, régression, segmentation, etc.
5. **Choix de l'algorithme de fouille de données correspondant**
   Il s'agit de définir l'algorithme à utiliser pour la découverte des connaissances à partir de données.
6. **La fouille de données**
   La fouille de données est le cœur du processus, elle permet d'extraire les informations utiles et inconnues de données de taille importante sauvegardées dans des bases ou des entrepôts de données appliquant ces tâches : les règles d'association, la classification supervisée, la segmentation.
7. **L'évaluation et interprétation**
   On évalue les résultats trouvés par l'étape de la fouille de données pour mesurer la qualité des modèles construits. Le choix des critères d'évaluation est lié au contexte étudié : par exemple dans la classification, on utilise généralement la précision et la régression prédictive, le critère employé est l'erreur quadratique moyenne de la valeur prédite à partir de la valeur réelle.

   L'interprétation consiste à analyser les résultats obtenus par un expert du domaine certifié par le développeur. Celui-ci pourra entraîner des changements dans les réglages réalisés lors des phases précédentes (données choisies, seuils de discrétisation, ...).

8. **Interprétation des modèles extraits**
   Elle consiste à revenir à l'une des étapes 1 à 7 pour une itération ultérieure. Cette étape peut également inclure la visualisation des motifs / modèles extraits, ou des données à partir des modèles extraits.
9. **La consolidation de la connaissance découverte et la diffusion**
   Il y a plusieurs tâches à réaliser après la construction des modèles parmi elles : la prise des décisions futures, l'implémentation sur l'ensemble de nouvelles données, l'évaluation du modèle au cours du temps pour assurer l'évolution du modèle.

2.3 Sélection d'attributs

La sélection d'attributs est un processus qui sélectionne un sous ensemble à partir de l'ensemble initial d'attributs que l'on peut mesurer par des critères d'évaluation pour assurer le plus favorable et le meilleur. L'obtention d'un meilleur sous ensemble d'attribut, généralement, n'est pas facile et peut être réputé comme un problème d'optimisation difficile NP-difficile ([Blum & Rivest 1992].[Kohavi & John 1997]. [Cotta & Moscato 2003]).
Les auteurs [Molina *et al.* 2002] ont présenté quelques démarches de ce processus :
Si le nombre optimal d'attributs à choisir est connu à l'avance, on extrait le sous ensemble de m attributs parmi n avec un nombre de combinaisons (m, n) pour assurer une performance au moins égale ou meilleure à celle trouvée avec l'ensemble original mais la progression exponentielle de (m, n) ramène la recherche très chère et par suite l'exploration globale devient non faisable.
Si on valide le critère de performance à satisfaire, on cherche à obtenir une représentation du problème avec la sélection d'un sous ensemble de taille minimale dont la performance soit meilleure ou égale que le critère défini.
Les objectifs de la sélection d'attributs
- L'augmentation des performances des modèles construits.
- La réduction de la consommation de ressources par l'utilisation d'un sous ensemble des attributs.
- L'accroissement de la précision pour la classification employant des attributs pertinents pour comprendre bien les systèmes étudiés.
- la diminution de la complexité pour l'interprétation facile et simple des modèles. Les auteurs ont prouvé que la sélection des attributs pour les arbres de décision fournisse un modèle de classification efficace et des arbres induits de taille réduite ([Seban & Nock 2001].[Perner 2001]).
- La diminution du temps d'apprentissage par l'augmentation de la vitesse l'exécution de l'algorithme d'apprentissage. Les auteurs [Langley & Iba 1993] ont utilisé l'algorithme du plus proche voisin, ils ont prouvé que l'apprentissage est moins cher en fonction des meilleurs attributs.
- Une excellente généralisation des données pour éviter le surapprentissage.
L'objectif de la sélection ([Dash & Liu 1997].[Kohavi & John 1997].

[Liu & Motoda 2008] est d'obtenir un meilleur sous ensemble d'attributs qui a ces caractéristiques : il doit être formé des attributs pertinents et il doit tenter d'éviter les attributs redondants. De plus, cet ensemble doit permettre d'assurer au mieux la précision et la rapidité de l'apprentissage, la compréhension du modèle suggéré.

2.3.1 Pertinence d'un attribut

En apprentissage automatique, l'auteur [Michalski 1983] a classé les attributs pertinents selon trois types :
— Pertinence complète : si tous attributs sont pertinents pour un problème donné.
— Pertinence partielle : pour faciliter la tâche d'apprentissage, on choisit les attributs les plus pertinents.
— Pertinence indirecte : si les attributs non pertinents donnent la naissance à des nouveaux attributs par la génération faite avec les non pertinents.
- Les auteurs [Gennari & Fisher 1989] ont défini les attributs pertinents dont les valeurs des attributs se modifient périodiquement en fonction de l'appartenance de données à telle ou telle classe.
- Les auteurs [Kohavi & John 1997] ont indiqué que les attributs fortement pertinents sont toujours présents dans tout meilleur sous ensemble sélectionné pour s'éloigner d'anomalie de reconnaissance de la fonction « objectif ». La faible pertinence propose que l'attribut ne soit pas toujours important, mais il est essentiel dans certaine situation.
- Les auteurs [Dash & Liu 1998] ont défini un attribut non pertinent comme celui n'a pas d'effet sur la forme de base des données et un attribut redondant comme celui n'apporte rien de nouveau pour représenter la forme de base des données. Le processus de la sélection d'attributs contient diverses étapes et son espace de recherche peut être modélisé en sorte d'un treillis de gallois.

2.3.2 Redondance d'attributs

D'une manière générale, la définition de la redondance des attributs est basée sur la notion de corrélation entre attributs.
- Selon les auteurs [Guyon & Elisseeff 2003], deux attributs sont redondants entre eux si leurs valeurs sont complètement corrélées. Cette condition n'est pas appliquée pour un sous ensemble d'attributs.
Les auteurs ([Koller & Sahami 1996]. [Yu & Liu 2004]) ont implémenté une procédure d'élimination descendante, nommée Filtrage par Couverture de Markov (Markov Blanket Filtering) pour trouver le meilleur sous ensemble d'attributs : les attributs fortement pertinents ne peuvent posséder aucune couverture de Markov. Par contre les attributs redondants et faiblement pertinents doivent être éliminés.
Soit E l'ensemble actuel d'attributs et M l'ensemble actuel d'attributs (E=M initialement).
A chaque pas d'exécution, s'il y a une couverture de Markov pour l'attribut de M_i dans l'ensemble E actuel, M_i est exclu de E. Un attribut exclu dans le pas précédent peut posséder une couverture de Markov dans un pas postérieur.

2.3.3 Schéma général de la sélection d'attributs

Le processus de la sélection d'attributs [Dash & Liu 1997] dans lequel on trouve ces phases (Figure 2.2) :

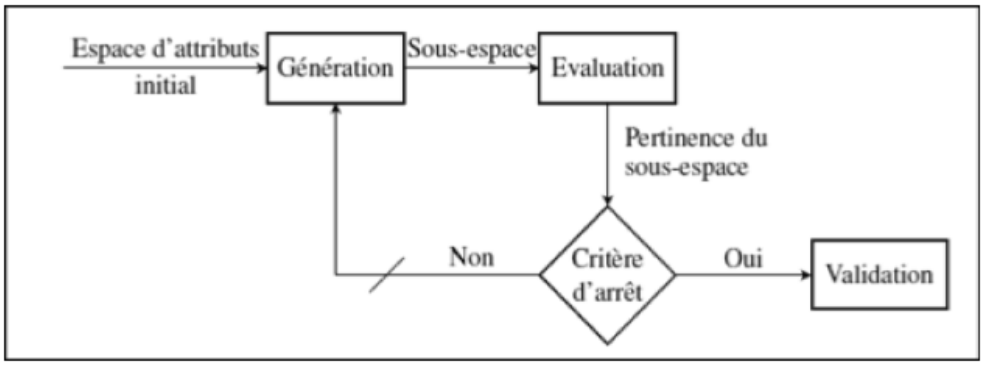

FIGURE 2.2 – *Processus de sélection d'attributs [Dash & Liu 1997].*

1. **Génération du sous ensemble** À partir de l'espace de recherche et selon les deux critères suivants, on génère un sous ensemble candidat pour l'évaluation [Liu & Yu 2005] :
   (a) **Point de départ** : est un point dans l'espace du sous ensemble des attributs pour initialiser la recherche, et ce même point va orienter la direction de recherche. Pour n attributs, l'espace de recherche comporte 2n-1 ensembles possibles.
   - Option vers l'avant (Forward) : on démarre la recherche par un ensemble vide et on ajoute des attributs l'un après l'autre.
   - Option en arrière (Backward) : on démarre avec tous les attributs et on supprime successivement des attributs.
   - Option Pas à pas (Stepwise) : on démarre avec les deux limites, puis on ajoute et on supprime des attributs en commun à la fois.
   Afin d'éviter d'être tombé dans des optima locaux, la recherche peut également démarrer par un sous ensemble sélectionné au hasard.

   (b) **Stratégie de recherche** Elle permet d'explorer les différentes combinaisons des attributs. Pour n attributs, l'espace comporte $2^n - 1$ sous ensemble d'attributs possibles. Cet espace de recherche est exponentiellement illicite pour la recherche totale. Par conséquent, diverses techniques ont été explorées : recherche complète, séquentielle et aléatoire.
      i. **La recherche complète** Elle permet de trouver le meilleur résultat selon le critère d'évaluation employé. La recherche ne doit pas être totale pour qu'elle soit complète. Diverses heuristiques et fonction peuvent être appliquées pour diminuer l'espace de recherche sans avoir de danger de perdre les meilleurs résultats, par exemple : Branch and Bound [Narendra & Fukunaga 1977] et Beam Search [Doak 1992].

ii. **La recherche séquentielle** Elle n'assure pas le meilleur résultat, elle repose sur la répétition par l'ajout ou l'exclusion des attributs. On trouve plusieurs variations de la méthode de descente [Liu & Motoda 1998] comme : sélection en avant (Forward Selection), Élimination rétroactive séquentielle (Sequential Backward Elimination), élection bidirectionnelle (Bidirectional Selection). Diverses techniques de recherches heuristiques peuvent être configurées. Dans la recherche séquentielle ascendante (Forward Selection), l'ensemble de démarrage est l'ensemble vide et à chaque pas de génération du sous ensemble on trouve le meilleur entre les attributs qui ne sont pas encore choisis, et on ajoute à l'ensemble actuel. De la même manière, une recherche descendante commence par l'ensemble complet d'attributs et enlève à chaque pas un attribut. L'avantage principal de ces méthodes est qu'elles sont de complexité quadratique. Sur des données de taille moyenne, elles peuvent être implémentées de manière efficiente.
Les approches de la recherche séquentielle sont faciles à mettre en œuvre et vite à fournir des résultats dont l'ordre de l'espace est souvent $O(N^2)$ ou moins.

iii. **La recherche aléatoire** Au début, choisir un sous ensemble au hasard. L'utilisation de l'aléatoire permet d'éviter des optima locaux dans l'espace de recherche et le sous ensemble optimal retenu dépend des ressources disponibles et le choix des paramètres. Il y a deux chemins à suivre :
Soit à utiliser la recherche séquentielle, exemple : méthode de descente et Recuit Simulé [Doak 1992]. Ou chaque ensemble d'attributs est engendré de façon complètement aléatoire, exemple Algorithme LasVigas [Brassard & Bratley 1996].
La Figure 2.3 montre la classification des approches de sélection d'attributs selon leur technique de recherche [Dash & Liu 1997].
- La stratégie de recherche complète est organisée en deux classes : « exhaustive » et « «non exhaustive ».
—La classe exhaustive : une approche peut évaluer tous les sous ensembles 2n où elle peut faire la recherche La largeur d'abord (Breadth first) pour arrêter la recherche dès qu'un meilleur sous ensemble est obtenu.
—La classe non exhaustive : on implémente des techniques de recherches comme : Branch and Bound , Best First et Beam search.
- La stratégie de générations heuristiques est organisée en sélection « Forward », sélection « Backward », sélection hybride Forward/bacward et « Instance Based ».
- Les stratégies de recherche aléatoire sont organisées en deux classes :
—Classe01 : la probabilité de génération d'un sous ensemble est valide et elle est la même pour tous les sous

ensembles.
—Classe02 : la probabilité de génération d'un sous ensemble se varie lors de l'exécution de procédure.

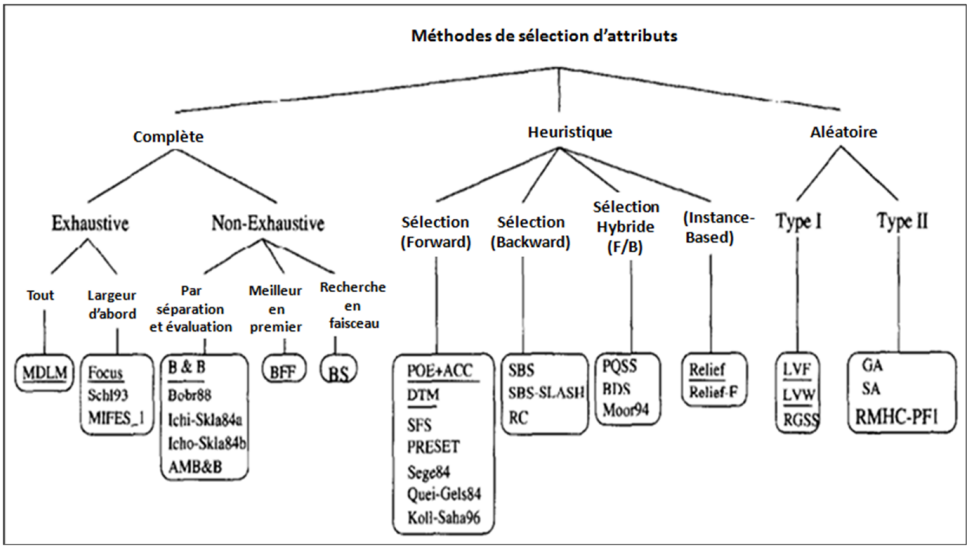

FIGURE 2.3 – *Méthodes de sélection d'attributs [Dash & Liu 1997].*

2. **Evaluation du sous ensemble** On évalue chaque nouveau sous ensemble généré utilisant les critères d'évaluation. Le sous ensemble optimal est défini avec certains critères.
   Les critères d'évaluation sont divisés en deux types d'algorithmes : Filter et Wrapper.
   (a) **Les critères indépendants** L'algorithme de type Filter est basé sur les critères indépendants. L'évaluation par Filter consiste à exploiter des propriétés importantes de données d'apprentissage sans l'utilisation des algorithmes [Liu & Yu 2005], les plus célèbres sont :
   - les mesures de distance
   Si l'attribut alpha engendre une distinction importante dans la loi de probabilités conditionnelles des deux classes de problèmes que l'attribut beta, alors alpha est préféré à l'autre attribut, en tentant d'identifier l'attribut qui peut dissocier les deux classes d'une manière plus éloignée. alpha et beta sont semblables si la différence est nulle. Les mesures appliquées sont : la distance euclidienne, la distance de Mahalanobis, la distance de Bhattacharya et le critère de Fisher.
   - Les mesures d'information
   Elles consistent à calculer le gain d'information de l'attribut alpha. Si le gain d'alpha est important que le gain de beta alors alpha est préféré de l'autre attribut, on utilise comme mesure par exemple : entropie de Shannon.
   - Les mesures de dépendances
   Le facteur est la mesure de dépendance classique et peut être appliqué pour obtenir la similitude entre l'attribut alpha et la classe M, si cette dernière est plus grande que celle de l'attribut

beta avec M, alpha est préféré de l'autre attribut.
- Les mesures de cohérence
Ces mesures tentent de trouver le nombre minimal d'attributs qui séparent les classes comme la consistance de l'ensemble complet d'attributs. L'inconsistance est définie comme deux instances ayant les mêmes valeurs d'attributs mais avec différentes classes.
Tableau2.1 présente une comparaison des fonctions d'évaluation [Liu & Yu 2005], indépendamment de la nature de la procédure de génération implémentée. Pour comparer, les paramètres appliqués sont :
— Généralité : comment le sous ensemble adéquat est choisi pour différents modèles de classification.
— Complexité en temps : le temps retenu pour la classification du sous ensemble d'attributs.
— La précision : est la prédiction du sous ensemble choisi. Le signe * exprime qu'il n'y a pas de conséquence conforme à la fonction d'évaluation.
Pour la classification après la sélection d'attributs, la précision de toutes les fonctions d'évaluation sauf le taux d'erreur de classification est liée de l'ensemble d'attributs et des classifieurs implémentés.
On remarque (Tableau2.1) que la précision est plus grande si la complexité en temps est grande, avec des contraintes de temps, le taux d'erreur du classifieur ne doit pas être retenu comme une fonction d'évaluation.

(b) **Les critères de dépendances** L'algorithme de type Wrapper est basé sur les critères de dépendances, l'approche wrapper utilise un algorithme de recherche prédéfini pour la sélection d'attributs. Cette approche a été proposée par [John *et al.* 1994], le principe consiste à évaluer les sous ensembles de candidats générés par l'algorithme d'induction.
La prédiction de la précision dans la classification est peut-être appliquée comme une mesure de dépendance pour la sélection d'attributs. Un modèle de sélection d'attributs dans la segmentation tente à évaluer le sous ensemble optimal d'attributs par la qualité de clusters trouvés par l'application de l'algorithme de la segmentation sur le sous ensemble sélectionné.
Le plus célèbre des critères de dépendances est :
- Les mesures de taux d'erreurs des classifieurs
Les algorithmes qui appliquent ce genre de critères d'évaluation sont nommés les approches Wrappers dont le classifieur est une fonction d'évaluation.

(c) **Les critères hybrides** Elles intègrent les deux approches Filter et Wrapper, les critères d'indépendances pour trouver le sous ensemble pour une cardinalité donnée et les algorithmes d'induction pour sélectionner le meilleur sous ensemble résultant à travers différentes cardinalités pour l'évaluation des sous ensembles d'attributs.

Fonction d'évaluation	Généralité	Complexité en temps	Précision
Mesure de distance	Oui	Faible	*
Mesure d'information	Oui	Faible	*
Mesure de dépendance	Oui	Faible	*
Mesure de consistance	Oui	Modérée	*
Taux d'erreur du classifieur	Non	élevée	Très élevée

TABLE 2.1 – *Comparaison des fonctions d'évaluation.*

3. **Critères d'arrêt** Le critère d'arrêt peut être un temps de calcul, un nombre d'itérations défini à l'avance, arrêt de l'évolution par rapport aux solutions déjà obtenues ou encore le fait que les sous ensembles concurrents deviennent trop similaires dans le cas méthodes à base de populations.
4. **Procédures de validation** Pour évaluer les attributs choisis, on a besoin d'utiliser un classifieur et d'analyser les performances de la classification. Par exemple, pour un sous ensemble d'attributs sélectionné, si on utilise le taux d'erreur de classification comme un indicateur de l'efficacité pour la manipulation, alors on peut remarquer les valeurs avant l'exécution et après pour comparer le taux d'erreur du classifieur sur l'ensemble total d'attributs et sur le sous ensemble sélectionné.
   La Figure2.4 montre la partition de données en un ensemble d'apprentissage et un ensemble de test. Cette méthode peut être implémentée pour évaluer n'importe quel type de méthode de sélection : Wrapper, Filter ou Hybride.

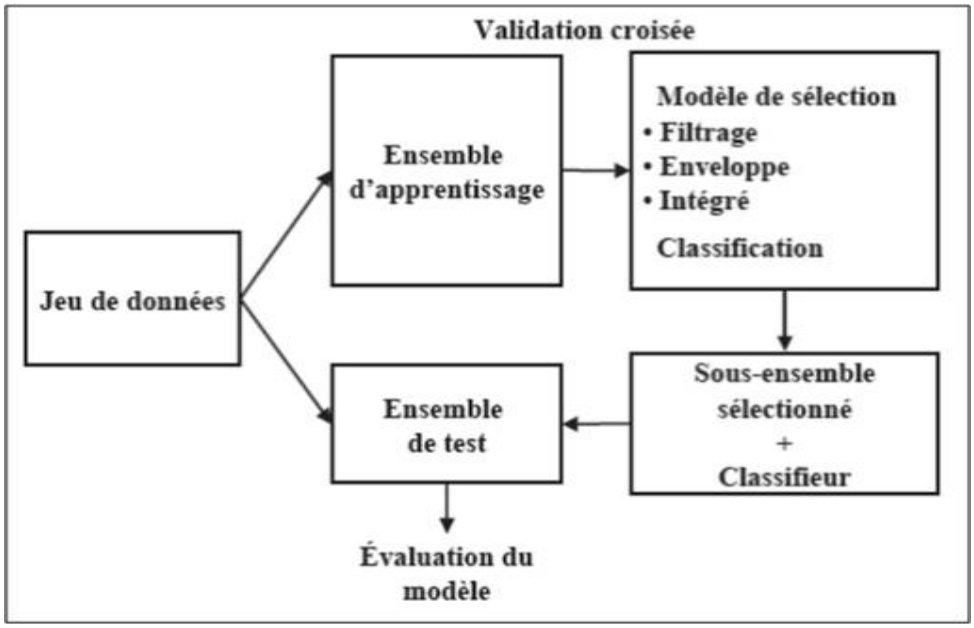

FIGURE 2.4 – *Validation Croisée pour l'évaluation d'un processus de sélection –classification [José-Crispin 2008].*

2.3.4 Organisation des algorithmes de sélection d'attributs

Les mesures d'évaluation et les techniques de recherches vues précédemment sont des éléments importants pour la préparation des algorithmes de sélection d'attributs.

1. **Les algorithmes Filter [Liu & Yu 2005]**
   Elles consistent à utiliser les diverses méthodes et mesures pour déterminer la pertinence d'un attribut avant la construction d'un classifieur par l'apprentissage. Le sous ensemble sélectionné est convenable au classifieur. Mais il y a un danger du sur apprentissage, filter prend un temps élevé de calcul pour l'établissement du classifieur à chaque exécution et évaluation du sous ensemble participant.
   Filter démarre la recherche avec un sous ensemble aléatoire S. Chaque sous ensemble engendré est mesuré par un critère d'indépendance et comparé avec le résultat précédent, s'il est le meilleur alors il est retenu comme le meilleur sous ensemble actuel. La répétition se termine si le critère d'arrêt est vérifié. L'algorithme retourne en résultat le sous ensemble optimal Figure2.5
   Focus [Almuallim & Dietterich 1991] :
   Initialement cet algorithme est proposé pour les types booléens non bruités, il réalise une exploration complète de tous les sous ensembles de l'espace de recherche pour déterminer l'appartenance à une classe de toutes les instances et pour identifier le plus petit sous ensemble. De plus [Dash & Liu 1998] prouvent que cette approche prend du temps si la taille du sous ensemble est grande.
   Branch And Bound [Narendra & Fukunaga 1977] :
   L'approche débute avec l'ensemble complet d'attributs et élimine un attribut à chaque pas. Cet algorithme cherche à obtenir la meilleure solution sans traverser tous les sous ensembles de l'espace de recherches. Les fonctions d'évaluation implémentées sont la fonction discriminante, le critère de Fisher, la distance de Bhattacharya et la divergence. Il est faible pour minimiser l'espace de recherche et la fonction d'évaluation doit être uniforme.
   Las Vigas Algorithm (LVF) [Chen & Liu 1999] :
   L'algorithme est basé sur la probabilité pour les choix des sous ensembles et la recherche de meilleur sous ensemble. LVF est vite dans la réduction du nombre d'attributs mais il est lent pour s'approcher vers l'optimal. Il est aveugle et génère des sous ensembles inutiles à cause au problème de dégradation.
2. **Les algorithmes Wrapper [Liu & Yu 2005]**
   L'algorithme de nature Wrapper emploie un classifieur pour mesurer le sous ensemble d'attributs sélectionné (Figure2.5) par l'erreur de généralisation, il est indépendant du processus de la classification.
   L'approche de type Wrapper est utile pour l'ensemble important d'attributs à cause de sa complexité normale. Elle ne prend que des informations montrées dans les données.
   Mais il reste un défi le choix de paramètres comme le choix d'un seuil pour le critère de pertinence sélectionné ou d'un nombre d'at-

tributs à prendre.
Une démarche exige :
- Un espace de recherches pour former les sous ensembles d'attributs.
- Le sous ensemble d'attributs initial qui est vide si on fait une sélection Forward, ou l'ensemble complet d'attributs si on fait une élimination backward.
- Un critère d'arrêt.
- Un algorithme d'induction pour évaluer chaque sous ensemble généré sur les données et mesurer la qualité des résultats.

Sélection séquentielle en avant (SFS)[Whitney 1971] :
SFS débute avec un sous ensemble d'attributs vide et par la suite ajoute les attributs un après l'autre, en acceptant à chaque pas celui qui optimise un critère d'évaluation. Cet algorithme est moins coûteux lorsque le nombre d'attributs obtenu est plutôt inférieur.

Sélection séquentielle en arrière (SBS)[Marill & Green 1963] :
SBS commence par de l'ensemble original d'attributs, à chaque étape et pour optimiser un critère d'évaluation. Il exclut l'attribut non utile. Cet algorithme est efficace, il achemine la recherche rapidement vers des optima locaux, mais les premières itérations coûtent chères. SBS permet d'évaluer l'impact de chaque attribut sur la classe en présence des autres attributs.

Best First Search (BFS) [Russell & Norvig 1995] :
L'approche consiste à déterminer le nœud le plus promoteur qui n'a pas été étudié.

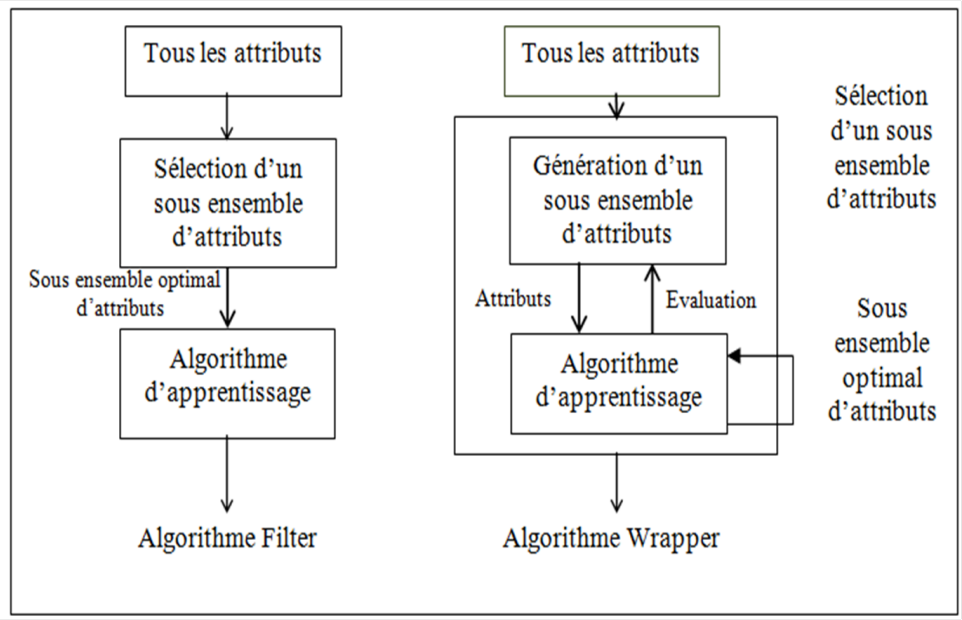

FIGURE 2.5 – *Algorithmes Filter et Wrapper [Jourdan 2003].*

3. **Les algorithmes hybrides**
   La méthode hybride est utile pour une base de données volumineuse [Das 2001]. Elle initialise la recherche avec un sous ensemble souvent vide avec la sélection Forward et suit l'exécution des phases pour obtenir le sous ensemble optimal.

La méthode hybride parcourt et évalue les bons sous ensembles de taille t+1 par l'inclusion d'un attribut du reste des attributs et compare au bon sous ensemble précédent de taille t, s'il est le meilleur, il devient le bon sous ensemble actuel à niveau t+1. L'évaluation est faite par un critère d'indépendance.
Après chaque réalisation d'une itération, l'algorithme de recherche est implémenté sur le bon sous ensemble à niveau t+1, et par la suite le résultat obtenu est comparé avec le meilleur sous ensemble à niveau t, l'algorithme fait la même chose pour le niveau suivant jusqu'à l'obtention du meilleur sous ensemble final.

2.4 Sélection des instances

2.4.1 Méthodes de sélection d'instances

Les méthodes de sélection d'instances peuvent être classées en deux groupes [Derrac *et al.* 2010] :
- Les méthodes de sélection de prototype (PS) [Liu & Motoda 2002] : le but est de trouver des ensembles d'apprentissage permettant d'obtenir la meilleure précision de classification et les meilleurs taux de réduction en appliquant des classifieurs basés sur des instances prises avec une certaine mesure de similarité ou de distance.
- Les méthodes sélection de l'ensemble d'apprentissage (TSS) [José-riquelme *et al.* 2003] : il consiste à implémenter les méthodes de sélection d'instances sur l'ensemble d'apprentissage utilisé pour construire un modèle prédictif (arbres de décision, réseaux de neurones, etc.). Ainsi, le TSS peut être utilisé comme un moyen de développement de la précision et de l'interprétation des modèles prédictifs.

Processus de sélection d'instances
Soit T l'ensemble d'apprentissage, X un ensemble de données.
T comporte des informations non utilisées pour la classification, ces instances inutiles peuvent être bruyantes ou redondantes, un processus de sélection d'instances est essentiel pour enlever les instances dispensables (Figure2.6, [Olvera-Lopez *et al.* 2010].
Grâce à un Algorithme de sélection d'instances, on trouve un sous-ensemble $S \subset T$ en vérifiant ces conditions :
S ne comporte pas d'instances superflues.
Précision (S) et Précision (T) sont égaux ou proches (la précision d'une classification par exemple).
L'ensemble initial de démarrage des Algorithme de sélection d'instances peut être égal à l'ensemble vide c'est le cas d'un algorithme incrémental, les instances sont ajoutées dans S pendant le processus de sélection. Si l'ensemble initial égal à l'ensemble original (total), l'algorithme est effectué par décrémentation, les instances sont enlevées de S pendant le processus de sélection.
Grâce à la sélection d'instances, la taille des données d'apprentissage est diminuée, et par la suite la durée d'exécution du processus d'apprentissage est diminuée.

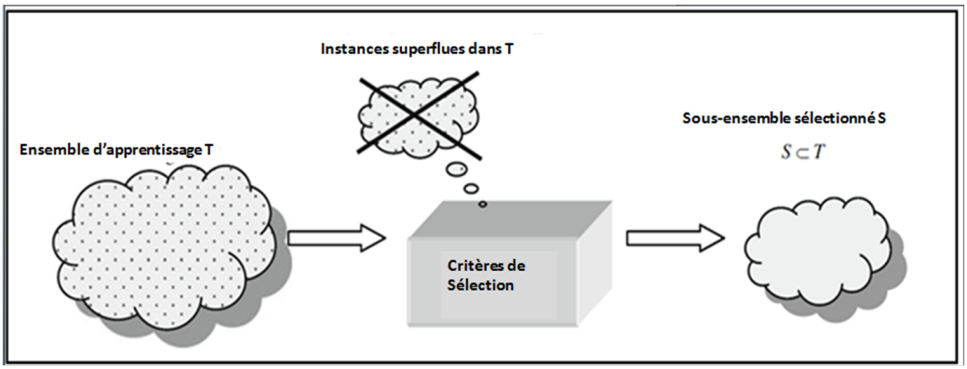

FIGURE 2.6 – *Processus de sélection d'instances [Olvera-Lopez et al. 2010].*

2.4.2 Problèmes résolus par la sélection des instances

Les méthodes de la sélection d'instances ont été appliquées avec réussite pour traiter deux défis en fouilles de données [Derrac *et al.* 2010], le problème de mise à échelle et le problème de déséquilibre des données.
Le problème de mise à l'échelle survient lorsque le nombre d'échantillons d'apprentissage excède la capacité des algorithmes classiques de fouille de données, il affecte sur l'exécution des algorithmes à cause de la taille énorme de données. Ce problème augmente la complexité de temps et affecte sur la précision de la généralisation. On peut le traiter selon deux façons :
- Mise à l'échelle de l'algorithme [Provost & Kolluri 1999] : concevoir des algorithmes plus rapides.
- Diminution de la taille de données [Liu & Motoda 2002] : réduire la taille des données avec des approches spéciales pour l'adapter à l'algorithme d'origine.
Le problème des ensembles de données déséquilibrés survient lorsque les données comportent beaucoup plus d'instances d'une classe que de l'autre et que la classe la moins représentative constitue le concept le plus important pour l'apprentissage [Chawla *et al.* 2004].
Généralement, dans les problèmes de classification déséquilibrés, les instances sont organisées en deux groupes : la classe majoritaire négative et la classe minoritaire positive. La classe positive a plus d'importance et s'accompagne avec des erreurs élevées de classification.

2.4.3 Organisation des algorithmes de sélection d'instances

Selon la stratégie implémentée pour choisir les instances, on peut définir les algorithmes de sélection des instances suivant deux catégories Filter et Wrapper.

1. **Les algorithmes Wrapper** [Olvera-Lopez *et al.* 2010]
   Le plus proche voisin condensé(Condensed Nearest Neighbor, CNN)[Hart 1968] : CNN est un algorithme incrémental. Au début, il ajoute aléatoirement dans le sous ensemble une instance appartenant à chaque classe. Après, chaque instance de l'ensemble original est classée en prenant le sous ensemble comme un ensemble d'apprentissage. Si une instance est mal classée, alors elle est incluse dans

le sous ensemble d'apprentissage pour assurer que les nouvelles instances semblables à cette instance seront classées justement ; avec cette propriété, les instances dispensables sont souvent mal classées par leurs voisines et comme ça, elles sont gardées.
Règle du plus proche voisin sélectif (Selective Nearest Neighbor rule, SNN) [Ritter *et al.* 1975]. SNN est une extension de l'algorithme CNN, il trouve un sous-ensemble tel que chaque instance de l'ensemble original soit de la même classe d'un individu du sous ensemble et plus proche que de toute instance de l'ensemble original, ce qui signifie que l'ensemble original est correctement classé par 1-NN à l'aide du sous ensemble.
Règle du plus proche voisin condensé généralisé (Generalized Condensed Nearest Neighbor rule, GCNN) [Chien-Hsing *et al.* 2006] : GCNN est similaire à l'algorithme CNN, en plus il ajoute dans le sous ensemble les instances assurant le critère d'absorption en fonction d'un seuil donné. Pour chaque instance, l'absorption est calculée en fonction des voisines les plus proches et des voisines les non proches. Les instances de l'ensemble original qui sont absorbées ne sont pas ajoutées dans le sous ensemble.
Le plus proche voisin édité (Edited Nearest Neighbor, ENN) [Wilson 1972] : L'algorithme ENN utilise une méthode de nettoyage pour effectuer un sous-échantillonnage. Il se concentre sur l'enlèvement des instances bruyantes. L'instance est enlevée de l'ensemble d'apprentissage, si elle est mal classée par au moins la majorité de ses k voisines les plus proches. Généralement, cet algorithme utilise k = 3.
All K-NN [Tomek 1976] : la démarche de l'algorithme est structurée comme suit : pour l'itération i jusqu'à k, indiquez les instances mal classées par ses k plus proches voisines. Si l'itération égale à k, toutes les instances indiquées sont enlevées.
Édition multiple [Devijver & Kittler 1980] : il consiste à partitionner l'ensemble d'apprentissage en blocs, et à effectuer l'algorithme ENN à chaque bloc avec la recherche des voisines dans le bloc successeur. Le processus de recherche s'arrête s'il n'y pas d'instances à enlever dans les itérations successives.
IB2, IB3 [Aha *et al.* 1991] : dans ces algorithmes, la recherche est faite par incrémentation. Ib2 sélectionne les instances mal classées par l'algorithme 1-NN. IB3 est une extension de l'algorithme IB2, il détermine les instances dont leur enlèvement n'a aucun effet sur la précision de la classification.

2. **Les algorithmes Filter** [Olvera-Lopez *et al.* 2010]
Arbre Kd [Friedman *et al.* 1997] : c'est la construction d'un arbre binaire, toutes les instances sont incluses dans le nœud racine. Pour construire les nœuds enfants, l'enfant de gauche contient les instances dont les valeurs pour l'attribut correspondant sont inférieures à un critère de différence maximale et le nœud de droite contient les instances restantes, les instances situées dans les feuilles sont sélectionnées.
Algorithme de Chang généralisé-modifié GCM [Mollineda *et al.* 2002] : GCM fusionne les grappes les plus proches de la même classe et

sélectionne les centres à partir des nouvelles grappes fusionnées.
Classifieur de sous-classe la plus proche [Venmann & Reinders 2005] NSB : il permet de choisir un nombre différent des instances (centres) par classe via l'algorithme Grappe de variance maximale [Venmann *et al.* 2002].
Segmentation (CLU) [Lumini & Nanni 2006] : CLU sélectionne les centres des grappes ; cet algorithme a été appliqué dans le domaine de la reconnaissance de signature.
Sélection des objets par segmentation OSC [Olvera-Lopez *et al.* 2007] : le critère de choix prend une instance t comme une bordure si t est la plus proche d'une autre instance de grappe appartenant à une classe différente. Bien que les instances intérieures puissent être enlevées de l'ensemble d'apprentissage, afin de garder des instances significatives de grappes homogènes, OSC sélectionne l'instance centrale de chaque grappe homogène.
Prototypes de pondération (WP) [Paredes & Vidal 2000] : WP emploie la méthode descente de gradient pour calculer les poids pour chaque instance, les instances ayant des poids supérieurs à un seuil donné sont enlevées.
Sélection du prototype par pertinence (PSR) [Olvera-Lopez *et al.* 2008] : l'algorithme PSR est basé sur l'attribution de poids aux instances, il calcule la pertinence en fonction de la similitude moyenne de chaque instance de l'ensemble d'apprentissage. PSR choisit un nombre indiqué par l'utilisateur des instances les plus pertinentes par classe et, à partir d'elles, les instances les plus semblables appartenant aux différentes classes sont choisies.

2.5 Fouille de données (Data Mining)

La fouille de données signifie l'ensemble des algorithmes et des méthodes réservés à l'exploration et à l'analyse d'un ensemble de données volumineux en vue de détecter dans ces données [Berry & Linoff 1997] : des règles, des associations et des tendances inconnues (non fixées a priori) et des structures particulières, restituant de façon concise, l'essentiel de l'information utile pour l'aide à la décision. Elle emploie des méthodes statistiques avancées comme le partitionnement des données (rassemblement des données en paquets homogènes), et utilise régulièrement des techniques de l'intelligence artificielle ou des réseaux de neurones. L'objectif de la fouille de données est de trouver des relations inconnues dans les données, particulièrement quand les données proviennent de diverses bases de données.
Par exemple, on trouve : diagnostic médical, bioinformatique, commerce électronique et les ventes, fouille de textes... On peut préciser ce qui la diffère des domaines d'analyse connexes avec lesquels on pourrait quelques fois la confondre :
Fouille de données versus statistiques
La fouille de données se situe à la croisée des statistiques (Analyse Exploratoire, Modélisation Paramétrique/Non paramétrique,...), de l'intelligence artificielle (Apprentissage Symbolique, Reconnaissance des formes, Réseaux de Neurones,...) et des bases de données (BD relationnelles, En-

trepôts de données,...). Les programmes d'analyse sont lancés sur la base de données, sans buts de la nature « trouver la corrélation entre telle et telle donnée».

Fouille de données versus informatique décisionnelle

L'informatique décisionnelle regroupe les solutions informatiques apportant une aide à la décision avec les rapports et les tableaux de bord de suivi à la loi analytique et prospective.

L'objectif est de rassembler, consolider, modéliser et de restituer les informations disponibles au sein des bases de données de l'entreprise et de permettre aux responsables de la stratégie d'une entreprise d'avoir une vision d'ensemble d'activités manipulées : choix des données, tri, stockage ou répartition de ces données selon certains critères, réalisation des calculs récapitulatifs simples, exposition synthétique des résultats.

La fouille de données ajoute une dimension complémentaire qui est la détection des liens et la validation de leur reproductibilité.

2.5.1 Classification des algorithmes de fouille de données

Selon le type d'apprentissage implémenté dans les algorithmes de la recherche

- Fouille supervisée : elle contient à la fois des données d'entrée et de sortie dont l'objectif est de classer correctement un nouvel exemple.
- Fouille non supervisée : il n'y a pas de notion de classe, c'est un processus dans lequel les exemples d'apprentissage ne contiennent que des données d'entrées dont l'objectif est de classer les exemples en segment (cluster) d'exemples équivalents.

Selon les objectifs :

- Classification : elle consiste à prédire si une instance de donnée est membre d'une classe prédéfinie.
- Segmentation : elle consiste à diviser logiquement la base de données en clusters homogènes à l'intérieur d'une population.
- Règles d'associations : elles consistent à identifier les relations entre les attributs.

Selon les auteurs [Usama *et al.* 1996]

- Les algorithmes prédictifs : elles emploient les données avec des résultats connus pour améliorer des modèles effectuant la prédiction des valeurs des nouvelles données.
- Les algorithmes descriptifs : elles décrivent l'état courant et les caractéristiques des données et concentrent sur la compréhension et l'interprétation de modèles.

2.5.2 Tâches de la fouille de données

1. Régression

La notion de la régression a été introduite par Francis Galton en 18^e siècle [Manic 2011], la régression est un ensemble des approches statistiques employées afin de trouver le meilleur modèle qui décrit la relation entre une variable de sortie et une ou plusieurs variables d'entrée. Elle consiste à trouver une fonction f qui se rapproche le plus possible d'un scénario donné d'entrées et de sorties. La régression peut être linéaire (ajustement

de la courbe à une ligne) ou non linéaire (méthode de Newton), ainsi la régression peut être paramétrique où la fonction régression est définie par des paramètres inconnus (méthode des moindres carrés) ou par non paramétrique (régression polynomiale). La régression floue (régression des moindres carrés linéaires flous) peut être utilisée pour lutter contre le phénomène d'incertitude des données entraînant l'incertitude de la solution.
En apprentissage automatique, on trouve les problèmes de régression et les problèmes de classification. Tel que, le problème de prédiction d'une variable quantitative est un problème de régression et le problème de prédiction d'une variable qualitative est un problème de classification. Tandis que la régression logistique est à la fois une approche de régression au sens où il s'agit de prédire la probabilité d'appartenir à chacune des classes et une approche de classification.

a) **Régression linéaire simple, multiple et logistique**
Un modèle de régression linéaire y=ax+b cherche à établir une relation linéaire entre une variable expliquée y et une variable explicative x (ou plusieurs). Généralement un modèle linéaire simple est un modèle de régression avec une seule variable explicative. L'idée de la régression linéaire simple réside dans l'analyse de la corrélation entre deux variables x et y quantitatives, en représentant cette liaison par une droite. Pour un ensemble des observations (points) dans l'espace R^2, on regroupe le plus convenable des points (x_i, y_i) afin de tracer la droite. Pour résoudre, la méthode des moindres carrés due à Gauss est utilisée.
On définit le modèle de régression linéaire multiple comme tout modèle de régression linéaire avec au moins deux variables explicatives. La régression linéaire multiple est une méthode statistique employée pour l'étude des données multidimensionnelles, elle représente une généralisation de la régression linéaire simple.
Soit le modèle multiple $y = b_0 + b_1x^1 + b_2x^2 + \cdots + b_dx^d$, tel que y est corrélée à un ensemble de variables $x^1, x^2, \ldots, x^d$. La solution est alors une projection de y sur le sous espace w (de degré d+1) engendré par $1, x^1, \ldots, x^d$. [Saporta 2006].
La régression peut être même employée pour l'estimation des variables catégorielles, par exemple les binaires en estimant la probabilité d'appartenance à l'une des deux classes. Ceci peut être réalisé en appliquant une fonction linéaire se basant sur les variables d'entrée et qui donne une probabilité, pour modéliser une variable qualitative binaire y à deux modalités : 0 ou 1, une solution consiste à expliquer la probabilité d'apparition de la variable (P= Pro (y=0) et 1-P=Pro (y=1)) ou la transformation de celle-ci par l'introduction d'une fonction réelle monotone g opérant de [0,1] dans R, la fonction Logit est la plus utilisée [Besse 2005].
Ce type de régression s'appelle la régression logistique, elle est une technique prédictive. Cette méthode permet de construire un modèle pour prédire/expliquer les valeurs prises par une variable cible qualitative généralement binaire (si elle possède plus de 2 modalités, on parle de la régression logistique polytomique) à partir d'un ensemble de variables explicatives quantitatives ou qualitatives. Elle constitue un

cas particulier de modèle linéaire généralisé.

b) **SVM pour la régression (SVR)**
Initialement, les SVMs ont été exploités pour les problèmes de la classification. Cependant, les SVMs peuvent ainsi être utilisés en cas de régression [Besse 2005] pour l'approximation des fonctions quand Y est quantitative. SVM utilise une astuce similaire à celle employée en classification pour la résolution du problème de régression.

2) Classification supervisée
La classification supervisée est une tâche de fouille de données qui nécessite la connaissance a priori pour déterminer l'appartenance d'un exemple à une classe. À l'aide d'un ensemble d'apprentissage, cette classification permet d'apprendre une approche qui permet de prédire l'appartenance d'un nouvel exemple à une classe.
La classification supervisée se réalise en deux phases :
L'apprentissage est la première phase, tout ce qui est appris par l'algorithme est structuré sous la forme des règles de classification que l'on appelle le modèle d'apprentissage.
La deuxième phase est la classification, dans laquelle les données tests vont être prises pour déterminer la précision des règles de classification obtenues pendant la première phase. Si la précision du modèle est observée comme satisfaisante, la règle pourra être implémentée à des nouvelles données.
Un classifieur est un algorithme qui, à partir d'un ensemble d'exemples, donne une prédiction de la classe de toute donnée.
Souvent, un classifieur agit par l'induction : à partir d'exemples spéciaux , on forme une connaissance plus vaste. L'induction de connaissance engendre la généralisation de la connaissance : à partir des connaissances dispersées, les exemples, on induit une connaissance plus vaste et on construit un modèle de données. Il est possible que l'erreur de généralisation s'arrive même si on pense que la classe des labels n'est pas erronée.
Si la taille du modèle construit est supérieure que la taille de l'ensemble des exemples, il n'est pas facile d'effectuer une prédiction efficace pour une donnée qui n'est pas présente dans l'ensemble des exemples, c'est le sur apprentissage.

a) **Validation croisée**
Pour s'éloigner de l'apprentissage par cœur en classification supervisée et pour obtenir une précision efficiente de l'erreur de classification, on prend en test des échantillons qui sont utiles pour l'apprentissage. L'efficacité des algorithmes de la classification est liée souvent au nombre d'échantillons d'apprentissage : plus ce nombre est supérieur, plus fiable seront les règles de classification. En même temps, il est important de garder un nombre significatif d'échantillons de test pour que l'évaluation de ces performances soit efficiente. La méthode de la validation croisée est souvent utilisée pour atteindre ces éléments : elle consiste à partitionner l'ensemble original en un nombre r des sous ensembles de taille égale (une validation croisée d'ordre r et r est supérieur ou égal à 2), chaque sous ensemble étant alors employé comme

une base de test, alors le reste des autres sous ensembles est employé comme une base d'apprentissage Figure 2.7.

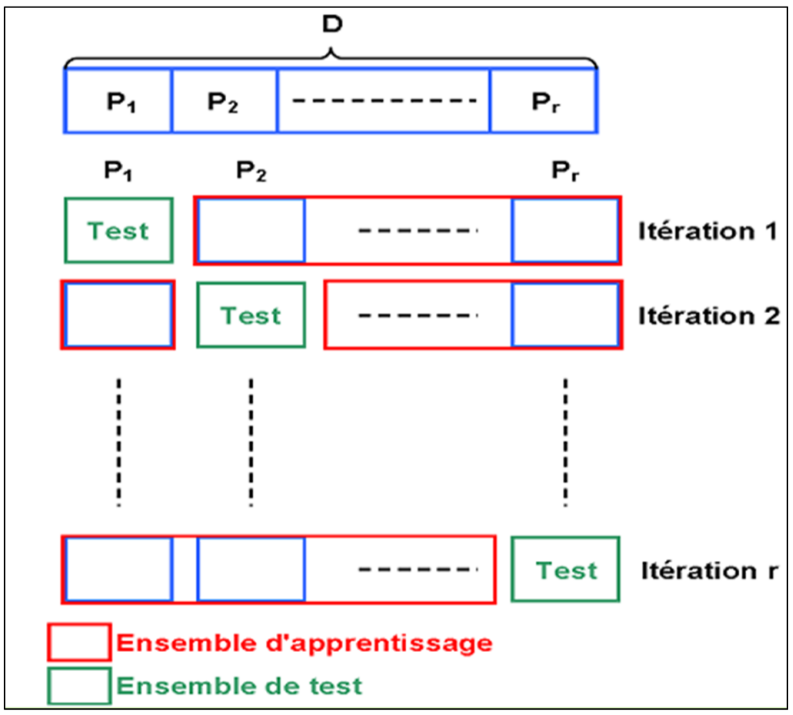

FIGURE 2.7 – *Validation croisée d'ordre r [Sébastien 2010]*

Soit D un ensemble de données de n instances dont la classe est définie, chaque instance contient des attributs. D est alors partitionné en r sous ensembles différents presque de même taille, à chaque pas d'exécution, le nombre des erreurs de classification obtenues est conservé et le taux d'erreur de la classification finale est retenu en divisant ces nombres d'erreurs par n.
La difficulté de la validation croisée est de définir la valeur convenable de d en fonction de la taille de l'ensemble original et de la complexité du problème : plus le problème est complexe, plus le système a besoin plus d'exemples pour apprendre.
Si r est petit, il y a moins de phases d'apprentissage à réaliser et l'obtention des résultats est plus rapide, Cette démarche est importante quand on utilise beaucoup de mesures.
-si r est grand, l'ensemble de test sera plus inférieur à l'ensemble d'apprentissage et il possible d'arriver à l'événement de surapprentissage ou le cas Leave One Out (cas particulier tel que r=n).

b) **Présentation des méthodes de la classification supervisée**

i) **Classifieur bayésien naïf**

Il consiste à représenter, utiliser, et apprendre des connaissances probabilistes. L'objectif est de prédire la précision de la classe pour les données de test et dans lequel les données d'apprentissage intègrent l'information de classes. C'est un type de réseaux bayésien, le terme naïf est employé car le classifieur appuie sur des hypothèses d'indépendance des données. Il suppose que les attri-

buts sont indépendants pour une classe donnée et qu'aucun des attributs cachés ou latents a un effet sur le processus de prédiction [John & Langley 1995], Malgré ses hypothèses, le classifieur performe généralement bien avec peu de données d'apprentissage. Puisque les données sont supposées indépendantes, seule la variance de chaque propriété doit être calculée (non pas la matrice de covariance en entier). Son principe théorique est basé sur le théorème de bayes [Bayes 1764].
Étant donné les hypothèses d'indépendance des variables, le classifieur est identifié comme suit (equation(2.1)) :

$$classe(f_1, \ldots, f_n) = plusgrand_c(p(C = c) \prod_{n=1}^{n} p(F_i = f_i | C = c)) \tag{2.1}$$

La fonction plusgrand ne fait que choisir la plus grande valeur parmi les probabilités que l'objet fait partie à chacune des classes. Cette probabilité est obtenue par la multiplication de la probabilité d'obtenir cette classe parmi les échantillons d'apprentissage avec la probabilité d'obtenir chaque propriété pour cette classe, selon la valeur moyenne et la variance préalablement calculée. La classe qui donne la plus grande probabilité en fonction des propriétés de l'objet présenté est sélectionnée en tant que classe prédite pour l'objet.

ii) **Réseaux de Neurones**
Les premiers principes de réseaux de neurones sont apparus dans les années 1950 par la simulation du fonctionnement du cerveau. Un réseau de neurones est constitué d'un graphe pondéré orienté dont les nœuds représentent les neurones. Ces neurones disposent d'une fonction d'activation (fonction signe, fonction sigmoïde, ...) qui permet d'affecter les autres neurones du réseau. Les liens synaptiques sont les liaisons entre les neurones, ces connexions propagent l'activité de neurones avec une pondération propre de la connexion (poids synaptique). C'est une méthode très utilisée en classification, spécialement dans le Domaine médical [Cang *et al.* 2002].
Il y a plusieurs types de réseaux de neurones, la plus célèbre est le perceptron multicouche. L'interconnexion entre les neurones permet un calcul total complexe qui en classification se traduit par des limites de décision aux formes complexes. L'étape d'entraînement de réseau permet de régler les poids synaptiques grâce à l'ensemble d'apprentissage.

iii) **Séparateurs à Vaste Marge (SVM)**
L'une des approches robustes et précises est le SVM [Cortes & vapnik 1995], les SVMs ont été largement employés dans la reconnaissance des formes, la classification, la régression et l'estimation de densité.
Le principe de base consiste à découvrir l'hyperplan optimal qui sépare deux classes dans l'espace de description. Cet espace est celui qui agrandit l'écart entre les deux classes (la marge), le SVM emploie des fonctions de noyau qui, dans un espace élevé,

permettent une meilleure séparation des points en diverses catégories. Les données d'apprentissage sont utilisées pour trouver l'hyperplan qui séparera d'une manière améliorée les points.
En SVM, on utilise sa fonction de noyau pour reconsidérer le même problème dans un espace de dimension plus grande. Cet espace est déterminé par la possibilité d'y découvrir un séparateur linéaire qui permet de classer les points dans les deux classes appropriées. Le séparateur linéaire peut ensuite être projeté dans l'espace d'origine où il devient habituellement non linéaire. Le critère d'optimisation est la largeur de marge entre les classes (l'espace vide de chaque côté des frontières de décision). La largeur de marge est déterminée par la distance jusqu'aux échantillons d'apprentissage le plus proche. Ces échantillons (vecteurs de supports) identifient la fonction discriminante de la classification, le nombre des échantillons est diminué en maximisant la marge.

iv) **Plus Proche Voisin (KNN)**
Le plus proche voisin est l'un des algorithmes les plus utilisés pour la classification dans les différents domaines de la reconnaissance des formes et de la fouille de données, KNN est basé sur la notion de voisinage entre les exemples et sur le raisonnement à partir de cas similaire pour la prise des décisions.
Un objet est classifié selon un vote majoritaire par ses voisins, l'objet prend la classe qui est la plus commune chez ses K plus proches voisins dans l'espace des recherches. K est un entier positif, souvent petit. On prend un k impair pour éviter l'égalité dans le vote. La distance employée pour le calcul de la proximité des voisins est généralement la distance euclidienne. Les exemples d'apprentissage sont des vecteurs dans un espace multidimensionnel. L'étape d'apprentissage conserve les données dans un format qui permettra le calcul efficace des distances et la recherche des voisins. Plusieurs variantes de KNN ont été proposées dans la littérature afin de diminuer la charge de calcul destinée à la recherche des voisins les plus proches d'un nouvel échantillon [Shakhnarovich *et al.* 2006].

v) **Bagging**
L'approche du Bagging a été proposée par [Breiman 1996]. Elle est composée de deux mots Bootstrap et Aggregating. Cette méthode consiste à entraîner un algorithme d'apprentissage sur différentes bases d'apprentissage résultantes par tirage avec remise (bootstrap) de m exemples d'apprentissage dans l'échantillon d'apprentissage S. Pour chaque tirage b, une hypothèse h_b est trouvée. L'hypothèse finale est basée sur les moyennes des hypothèses résultantes. Son avantage est qu'on optimise la performance des classifieurs instables en calculant la moyenne de leurs résultats. Ainsi, si les hypothèses h_b calculées pour chaque tirage b ont une variance intéressante, alors l'hypothèse finale aura une variance réduite. L'objectif du bagging est de diminuer l'instabilité propre à certains classifieurs, un classifieur est dit instable si un petit chan-

gement dans les données d'apprentissage engendre une modification importante dans le comportement du classifieur.

vi) **Boosting**

Boosting a été introduite par [Schapire 1990]. Cette approche consiste à réunir des classifieurs faibles pour garantir un classifieur fiable. On affecte à chaque exemple d'apprentissage la valeur 1/ le nombre d'échantillons. Ce poids désigne la difficulté de prédire la classe de cet exemple. On construit C classifieurs sur les exemples d'apprentissage pondérés par l'exécution de l'algorithme C fois.
Chaque pas qu'on produit un classifieur, on modifie les poids de nouveaux exemples appliqués pour le classifieur suivant, on incrémente le poids des exemples dont lesquels la classe est mal prédite et on diminue le poids des autres. Ensuite, on calcule l'erreur (r) du modèle sur l'ensemble pondéré. Si r est égale à zéro ou r est supérieure à 0.5 on termine la construction du classifieur. Le principe est de forcer le nouveau classifieur à diminuer l'erreur attendue. Le classifieur final est construit en appliquant un schéma de vote.
Algorithme Adaboost
L'algorithme Adaboost, abréviation de «Adaptive Boosting», est un méta algorithme d'apprentissage automatique formulé par [Freund & Schapire 1996]. C'était le premier algorithme de boosting pratique.
Adaboost génère un ensemble d'hypothèses (classifieurs) et les combine via un vote à la majorité pondérée des classes prédites par les hypothèses individuelles.
Les hypothèses sont générées en formant un classifieur ; des échantillons sont tirés d'une distribution mise à jour d'une manière itérative de l'ensemble d'entrainement. Cette mise à jour de la distribution garantit que les instances mal classées par le classifieur précédent ont plus de chances d'être incluses dans les données d'entraînement du prochain classifieur. Les classifieurs consécutifs sont formés sur des échantillons de plus en plus difficiles à classer. Dans cette contribution, le boosting a été utilisé en combinaison avec l'algorithme d'apprentissage de l'arbre de décision C4.5 en tant qu'un classifieur faible.

vii) **Arbres de Décisions**

Les arbres de décisions permettent de produire des procédures de classification compréhensibles par l'utilisateur. Ils représentent graphiquement un ensemble de règles facilement interprétables. L'induction avec des arbres de décision est l'une des formes d'algorithme d'apprentissage les plus simples et les plus efficientes [Russell & Norvig 2003]. Les arbres de décisions sont une des techniques les plus célèbres de l'apprentissage automatique et de la fouille de données. Les principaux algorithmes de construction d'un arbre de décision sont : ID3 [Quinlan 1986], C4.5 [Quinlan 1993] et CART [Breiman *et al.* 1984].
L'idée de base consiste à partitionner récursivement et d'une manière efficiente les exemples de l'ensemble d'apprentissage par des tests identifiés à l'aide des attributs jusqu'à ce que l'on trouve généralement des sous-ensembles d'exemples comportant

des exemples appartenant tous à une même classe. On trouve les trois opérateurs suivants :
1. Prendre une décision (un nœud doit être labellisé comme une feuille) si un nœud est terminal, Par exemple : tous les exemples sont dans la même classe, il y a moins d'un certain nombre d'erreurs, ...
2. Choisir un test à affecter à un nœud. Par exemple : implémenter aléatoirement des critères statistiques, ...
3. Assigner une classe à une feuille. On prend la classe majoritaire sauf dans le cas où l'on applique des fonctions coût ou risque.
Les approches se distinguent par les sélections réalisées pour ces divers opérateurs, il signifie le choix d'un test (utilisation du gain et de la fonction entropie par exemple) et le critère d'arrêt (la fin du développement de l'arbre ou le nœud est terminal).
Un arbre de décision idéal est un arbre de décision tel que tous les exemples de l'ensemble d'apprentissage soient justement classifiés. Le but est d'obtenir un arbre avec erreur de classification minimale qui est souvent un problème NP-complet.
Lors de l'étape de construction, la taille de l'arbre accroît d'une façon linéaire avec la taille de la base d'apprentissage. De plus, les arbres de décisions complexes peuvent avoir des taux d'erreur très supérieurs à cause de surapprentissage qui résulte lorsque l'ensemble d'apprentissage comporte des données bruitées ou qu'il ne comporte pas les exemples nécessaires. L'élagage est l'une des solutions pour diminuer ces taux d'erreurs en élaguant l'arbre par l'élimination de quelques branches. Il y a plusieurs approches d'élagage [Mingers 1989] pour éviter le surapprentissage.
Algorithme C4.5
C4.5 est un algorithme utilisé pour générer un arbre de décision développé par Ross Quinlan [Quinlan 1993]. Cet algorithme est une extension et une évolution de l'algorithme antérieur ID3 de Quinlan. Les arbres de décision générés par C4.5 peuvent être utilisés pour la classification.
Il utilise le rapport de gain comme un critère de fractionnement. Le fractionnement cesse lorsque le nombre d'instances à fractionner est inférieur à un certain seuil. L'élagage basé sur les erreurs est effectué après la phase de croissance. C4.5 peut gérer les attributs numériques. Il peut générer un jeu d'entraînement intégrant les valeurs manquantes en utilisant des critères de rapport de gain corrigé.
Avec un ensemble de cas S, C4.5 commence par développer un arbre initial à l'aide de l'algorithme de division et de conquête (divide-and-conquer), comme suit :
— Si tous les cas de S appartiennent à la même classe ou si S est petit, l'arbre est une feuille portant la classe la plus fréquente de S ;
— Sinon, choisissez un test basé sur un attribut unique avec deux résultats ou plus. Ce test devient la racine de l'arbre avec une branche pour chaque résultat du test, partitionnez S en sous-ensembles correspondants S1, S2,... en fonction de l'issue de

chaque cas, et appliquez la même procédure de manière récurrente à chaque sous-ensemble. Il existe généralement de nombreux tests qui pourraient être choisis dans cette dernière étape.
C4.5 utilise deux critères heuristiques pour classer les tests possibles : le gain d'information, qui minimise l'entropie totale des sous-ensembles $\{S_i\}$ (mais il est fortement biaisé en faveur de tests avec de nombreux résultats), et le rapport de gain par défaut qui divise le gain d'information par l'information fournie par les résultats du test. L'arbre initial est ensuite élagué pour éviter le sur apprentissage.

3) Segmentation (Clustering)

La segmentation est une tâche de l'apprentissage non supervisé, il n'existe aucune autre indication auparavant que la représentation des exemples. Elle a été appliquée dans les différents domaines tels que la biologie, l'analyse du web, la fouille de texte, l'exploration des données scientifiques, les applications de bases de données spatiales et le marketing [Berkhin 2002]. Pour la segmentation, il n'y a pas de classe à exprimer ou de valeur à prédire identifiée a priori, elle repose sur la création des groupes homogènes (clusters) dans la population. Après la construction des clusters, d'autres méthodes ou une expertise doivent extraire leur signification et leurs buts.

a) **Caractéristiques des algorithmes de segmentation**

Elles dépendent du problème ([Kumar 2000].[Berkhin 2002]) particulier à étudier, les méthodes de segmentation doivent être scalables, en termes de temps et d'espace. Pour une base de données volumineuse, l'algorithme de segmentation implémenté devrait avoir une complexité temporelle linéaire ou proche pour gérer l'ensemble volumineux de données. L'algorithme doit permettre d'enlever le bruit et les points isolés, soit en cours de l'exécution de l'approche ou avant et de déterminer les paramètres par exemple le nombre, la taille ou la densité des segments.
Les algorithmes assurent une fiabilité en présence de différentes propriétés de données et du segment comme la confidentialité, les bruits, la forme, la taille, la densité et la séparation du segment, le type d'espace de données (euclidienne ou non, ...) et les types d'attributs (temporel, catégorie, ...).
L'algorithme est capable de se dérouler d'une manière évolutive pour inclure des nouvelles données ou enlever les anciennes données sans l'exécution à nouveau du modèle sur le nouvel ensemble de données. Pour exprimer des résultats l'algorithme décrit un cluster comme une zone peut être mieux explicable qu'une liste de points.

b) **Système de segmentation**

L'idée réside dans la maximisation de la similarité des observations à l'intérieur d'un cluster et dans la minimisation entre clusters [Candillier 2004], il consiste à suivre :
- représenter les données (inclure éventuellement l'extraction et/ou la sélection d'attributs). C'est pour la production de nouveaux attributs ou l'application d'un sous ensemble efficient pour la segmentation.
- Déterminer la mesure de proximité adéquate au domaine des données pour éviter la domination d'un attribut sur un autre.
- La segmentation selon la méthode choisie.

- L'abstraction des données (si nécessaire), typiquement c'est la description de chaque cluster.
- L'évaluation de la sortie (si nécessaire), par les trois manières :
— Mesure externe : comparer à une solution a priori.
— Mesure interne : définir si la solution est essentiellement adéquate aux données.
— Mesure relative : comparer les différentes solutions possibles.

c) **Principales méthodes de la segmentation**
Parmi les algorithmes de clustering, on trouve K-moyenne [Macqueen 1967], EM [Dempster *et al.* 1977].

i) **K-moyenne (K-means)**
K-means a été proposé par [Macqueen 1967], il consiste à rassembler des données similaires en groupes (clusters), par l'analyse des données distinguées par un ensemble de descripteurs. La similarité des données est déterminée par la distance qui sépare leurs descripteurs, deux données sont dissemblables si leurs descripteurs sont plus loin. Donc il consiste à rechercher les k données prototypes (centroïdes) autour desquelles peuvent être regroupées les autres données.
Les étapes de réalisation de l'algorithme k-moyenne :
1. Sélectionner k centroïdes parmi les instances en prenant des données au hasard dans l'ensemble de jeu de données.
2. Affecter chaque instance au centroïde le plus proche pour construire les clusters.
3. Calculer les nouveaux centroïdes par la moyenne des instances de son cluster.
4. Si les centroïdes n'ont pas été changés pendant la dernière itération, on retourne ces centroïdes, sinon, aller à l'étape 2

Après quelques itérations, l'algorithme trouve un découpage stable du jeu de données : on dit que l'algorithme a convergé.

ii) **Espérance-Maximisation (EM)**
L'algorithme a été proposé par [Dempster *et al.* 1977], c'est une méthode itérative qui permet de découvrir les paramètres du maximum de vraisemblance d'un modèle probabiliste lorsque ce modèle dépend des données cachées. Il est employé spécialement pour l'estimation des Modèles de Markov Cachés et pour l'estimation des Mélanges de Gaussiens.
L'algorithme d'espérance-maximisation passe par ces phases :
- Une phase d'évaluation de l'espérance : on calcule l'espérance de la vraisemblance en prenant en considération des dernières données observées,
- Une phase de maximisation, où l'on détermine le maximum de vraisemblance des paramètres en maximisant la vraisemblance obtenue à la phase précédente.
On emploie ensuite les paramètres obtenus en M comme un point de départ d'une nouvelle étape d'évaluation de l'espérance, et l'on répète les itérations aussi.

4) Règles d'Association

C'est l'une des approches d'apprentissage non supervisées appliquées dans le domaine de la fouille de données et d'extraction des connaissances. La recherche des corrélations fréquentes dans un ensemble d'objets pour l'extraction des règles d'association à partir d'une base de données transactionnelle a été proposée par les auteurs [Agrawal *et al.* 1993].
On peut l'implémenter dans les domaines où il est important de rechercher des masses potentielles des produits ou de services : par exemple, dans le domaine médical pour la recherche de complications causée par des assemblages de médicaments ou à la recherche de falsifications.
La recherche de règles d'association est un processus itératif et interactif organisé par diverses étapes allant des choix et préparation de données jusqu'à l'interprétation des résultats.

a) **Etapes de la recherche des itemsets fréquents**
La majorité des méthodes proposées pour la recherche des itemsets fréquents se basent sur ces étapes :
- Préparation des données
On choisit les attributs de la base de données nécessaire à l'extraction des règles d'association et on transforme ces données en un contexte d'extraction. C'est un triplet T= (ob, At, Re) dans lequel Ob est un ensemble d'objets, At est un ensemble d'attributs, (items), Re est une relation binaire entre Ob et At.
Cette phase est importante pour qu'il soit possible d'utiliser les algorithmes d'extraction des règles d'association sur des données de natures diverses issues de sources diverses, de baser la recherche sur les données importantes pour l'application et de minimiser le temps d'extraction.
- Extraction des ensembles fréquents d'attributs
On extrait du contexte tous les ensembles d'attributs binaires (itemsets) $I \subset At$, les itemsets sont fréquents dans le contexte T, un itemset I est fréquent si son support, qui correspond au nombre d'objets du contexte qui « contiennent » I, est supérieur ou égal au seuil minimal de support min support identifié par l'utilisateur. L'ensemble des itemsets fréquents dans le contexte est noté F. Le problème de l'extraction des itemsets fréquents est de complexité exponentielle dans la taille m de l'ensemble d'items puisque le nombre d'itemsets fréquents potentiels est 2^m. Ces itemsets composent un treillis dont la représentation est sous forme de diagramme de Hasse. Lors de cette phase, des balayages du contexte doivent être appliqués, il est donc essentiel de trouver des méthodes efficaces d'exploration de cet espace de recherche exponentielle.
- Génération des règles d'association
Utiliser les itemsets fréquents trouvés durant l'étape précédente pour générer les règles d'association. Afin de limiter la génération aux règles d'association les plus utiles, seules celles qui ont une confiance supérieure ou égale au seuil minimal identifié par l'utilisateur sont générées.
Soient deux itemsets fréquents $I_1, I_2 \in T$ tel que $I1 \subset I2$, La confiance d'une règle $r : I_1 \rightarrow (I_2/I_1)$ est définie comme la proportion d'objets comportant la conséquence (I_2/I_1) de r parmi ceux qui contiennent l'antécédent I_1 de r. Cette valeur est égale au rapport entre le support

de l'itemset I_2 et le support de l'itemset I_1.

- Interprétation des résultats

À partir des règles d'association extraites du contexte et leur interprétation, l'utilisateur déduit les connaissances importantes pour développer l'activité concernée.

b) **Principales méthodes des règles d'association**

Parmi les algorithmes d'extraction des itemsets fréquents, on trouve l'algorithme Apriori [Agrawal & Srikant 1994], Fp-growth [Han *et al.* 2000].

i) **Apriori**

Apriori [Agrawal & Srikant 1994] est un algorithme de recherche de motifs fréquents, il permet la découverte des associations séquentielles et multidimensionnelles. Apriori est basé sur le principe lié à la notion de support et de confiance.

Pour trouver les itemsets fréquents, Apriori parcourt la grille des itemsets et en tire les règles d'association dont la confiance excède le seuil de confiance minimal.

L'utilisation de treillis des itemsets permet d'appliquer l'algorithme en vérifiant ces deux caractéristiques :

- Tout sous-ensemble d'un Itemset fréquent est fréquent.
- Tout sous-ensemble d'un itemset non fréquent est non fréquent.

Pour la recherche des itemsets fréquents dans une base de données transactionnelle, l'algorithme Apriori fonctionne en deux phases :

La phase de jointure Apriori réalise plusieurs balayages de la base de données. Le premier balayage sert à définir un ensemble candidat d'itemsets C_h, et à compter le nombre de fois d'existence (support) de chaque item. Tous les items dont le support est supérieur à la valeur (min sup) prédéfinie sont maintenus afin d'établir l'ensemble des h-itemsets fréquents Lh. Cet ensemble est utile pour générer l'ensemble de candidats C_{h+1}.

Pour chaque pas suivant, C_h des h-itemsets candidats est formé en liant les h-1 itemsets fréquents dans L_{h-1}. Deux itemsets I_1 et I_2 de l_{h-1} sont liés, si I_1 et I_2ont en commun (h-2) items. Cette liaison va produire un h-itemset candidat potentiellement fréquent qui formera C_h.

L'étape d'élagage : après la génération de l'ensemble de candidats C_h, le support de tous les h-itemsets est calculé, les h-itemsets dont le support est inférieur au support minimum défini par l'utilisateur (min sup) sont enlevés de la liste des candidats. La liste des candidats C_h est déterminée à partir de la liste précédente des candidats C_{h-1}, tout candidat enlevé à l'itération h-1 n'est plus considéré dans l'itération h. Si L_h est égal à l'ensemble nul alors aucun itemset candidat supplémentaire ne sera produit.

ii) **Fp-growth**

L'algorithme Fp-growth [Han *et al.* 2000] permet d'identifier les itemsets fréquents sans la détermination des itemsets candidats. Fp-growth emploie la politique « diviser et dominer », il représente les itemsets fréquents dans la base de données à l'aide d'un FP-Tree dont les branches comportent les associations potentielles des items. Chaque association peut être partitionnée en fragments

qui représentent les itemsets fréquents.
Pour former l'arbre FP-Tree, le processus suit ces étapes :
- Déterminer le support minimal (sup min).
- Déterminer chacune des occurrences d'un item représentant la base de transactions.
- Définir un critère de priorité pour ces items.
- Trier les items selon leur priorité.
- Établir le nœud racine.
- À partir de chaque nœud père ajouter les enfants en partant du nœud racine.
- Valider la forme de l'arbre FP-Growth.

2.6 Conclusion

Dans ce chapitre, on a décrit le processus KDD, on a commencé par la définition des différentes étapes de ce processus : identification de l'objectif du processus KDD, intégration et collecte de données, sélection, nettoyage, la détection des valeurs erronées, le traitement des problèmes de mise à échelle, le traitement de problème des ensembles de données déséquilibrés, le traitement des valeurs manquantes, l'échantillonnage, la sélection d'attributs, la transformation, La méthode de fouille de données particulière, le choix de l'algorithme de fouille de données correspondant, la fouille de données et l'évaluation et interprétation.
On a concentré sur les tâches des prétraitements de données et les tâches de fouilles de données. Amélioration du processus d'extraction des connaissances à partir de données par les métaheuristiques est un domaine très vaste, on va le présenter dans le prochain chapitre.

3 Hybridation des Métaheuristiques pour l'extraction des connaissances

Sommaire

3.1 Introduction

Avant de concevoir des métaheuristiques hybrides, il est essentiel de définir un choix bien adapté au problème d'optimisation à résoudre. Ceci peut être réalisé en donnant des réponses à ces questions :
Quel est l'objectif de l'optimisation ?
Existe-t-il des Métaheuristiques qui peuvent obtenir la solution requise pour le problème d'optimisation donné ?
Et quel type de métaheuristique hybride fonctionnera bien pour ce problème d'optimisation ?
Dans ce chapitre, on va présenter les critères classification des métaheuristiques hybrides, avec des exemples dans la sélection d'attributs, la sélection des instances, la classification supervisée et la segmentation.

3.2 Critères de la Classification des Métaheuristiques hybrides

La classification des Métaheuristiques hybrides est faite selon les auteurs ([Raidl 2015], [Raidl *et al.* 2010], [Blum *et al.* 2010], [G.Raidl 2006]) avec quatre propriétés (Figure 3.1) :

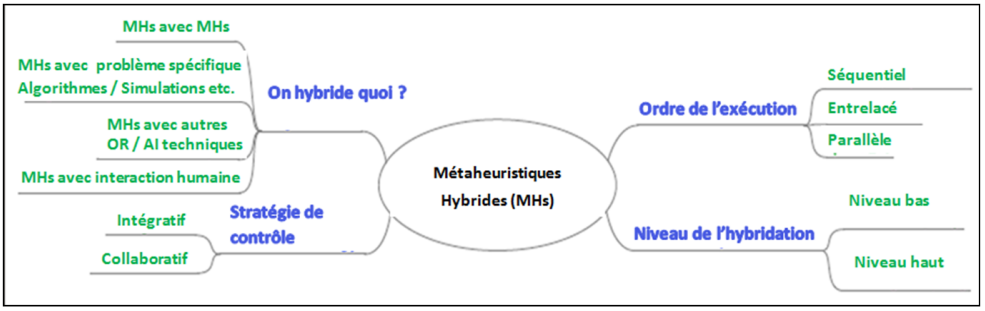

Figure 3.1 – *Critères de classification des Métaheuristiques hybrides [Raidl 2015].*

3.2.1 On hybride quoi ?

À cette étape, on définit la nature des algorithmes hybrides c'est-à-dire la combinaison est faite avec quels algorithmes (les méthodes) ?
- Métaheuristiques avec métaheuristiques.
- Métaheuristiques avec méthodes exactes : Branch-and Bound, la programmation dynamique, la programmation linéaire (LP), la programmation non linéaire et la programmation avec contraintes.
- Métaheuristiques avec les techniques de data mining et de l'intelligence artificielle.

Dans [Lee & Takagi 1993], les auteurs ont proposé un algorithme génétique à paramétrage dynamique (DPGA) qui implémente un contrôleur flou (Fuzzy logic controller FLC) pour superviser les paramètres de l'algorithme tels que le taux de mutation (P_m), le taux de croisement (P_c) et

la taille de la population (N), pour assurer la bonne exploitation et exploration de l'espace de recherche. Les entrées du FLC sont déterminées par une liaison de mesures de performance, et les sorties peuvent être n'importe quel paramètre de contrôle de l'algorithme génétique. Par exemple, le taux de mutation (P_m) peut être adéquat pour éviter la convergence rapide et pour accélérer l'optimisation.
L'évaluation de la fonction « objectif » dans les algorithmes évolutionnaires est généralement coûteuse, pour résoudre ce problème et pour développer la qualité des solutions trouvées, une approche hybride a été proposée [Javadi 2005]. Elle est basée sur l'hybridation des réseaux de neurones avec les algorithmes génétiques, la convergence de l'algorithme génétique a été accélérée par l'utilisation du réseau de neurones.
Les auteurs [Lin & Liang 2006] ont proposé une méthode hybride qui utilise l'algorithme de colonies de fourmis ACO, l'algorithme génétique GA et la recherche locale LS. Cette méthode a été implémentée pour résoudre le problème d'affectation quadratique (QAP). Au lieu de construire initialement une population aléatoire, ACO est utilisé pour la générer, pour garantir une bonne initialisation. La phéromone agit comme une technique de rétroaction de l'étape GA à l'étape ACO. Lorsque l'étape GA satisfait le critère d'arrêt, le contrôle est déplacé vers l'étape ACO. Ensuite ACO emploie la phéromone mise à jour par l'étape GA pour l'exploration de l'espace solution et la production d'une population prometteuse pour la prochaine exécution de l'étape GA. La méthode de recherche locale est implémentée pour développer les solutions trouvées par l'ACO et GA.

3.2.2 Niveau de l'hybridation

a) **Niveau bas**
On a une hybridation de bas niveau lorsqu'une fonction d'un algorithme est remplacée par un autre algorithme.

b) **Niveau haut**
On obtient une hybridation de haut niveau lorsque deux algorithmes sont hybridés sans que leurs fonctionnements internes ne soient en relation.

3.2.3 Ordre de l'exécution des méthodes hybrides

a) **Séquentiel**
Le résultat du premier algorithme est utilisé comme entré dans le deuxième algorithme.

b) **Entrelacé**
L'échange des informations entre les algorithmes se fait d'une manière bidirectionnelle.

c) **Parallèle**
Ce type est un domaine de recherche important qui se caractérise par :
- Architecture : SIMD/MIMD.
- Granularité du parallélisme : Fin (grande communication entre les algorithmes)/ Gros (faible communication entre les algorithmes).
- Environnement : Homogène/Hétérogène.
- Mémoire : Partagée/Distribuée.

- Allocation des données et des tâches : Statique/Dynamique.
- Type de synchronisation : Synchrone/Asynchrone.
La méthode la plus connue de cette classe est celle déterminée par [Talukdar 1999]pour trouver la solution à des problèmes de type asynchrone. Cette approche comporte un ensemble d'agents et de mémoires liées à un réseau guidé fortement cyclique. Chacun des agents est une méthode d'optimisation approchée ou exacte qui peut être utilisée pour résoudre le problème. Une population d'individus est présente dans la mémoire et les différents agents peuvent ajouter, enlever ou modifier des individus.
Une autre technique hybride associant GA et PSO appelée GSO (Genetic Swarm Optimization) a été introduite par les auteurs [Grimaldi *et al.* 2004] pour résoudre un problème d'optimisation en électromagnétique. Ce mécanisme est basé sur une forte collaboration entre GA et PSO tout au long de son exécution. À chaque pas, la population est découpée en deux tranches, chacune est évaluée respectivement par un algorithme. Les solutions seront ensuite reliées lors de la mise à jour de la population qui sera découpée en deux tranches de manière aléatoire dans le pas suivant.
L'hybridation des algorithmes évolutionnaires (EA) et l'algorithme d'optimisation par essaim de particules (PSO) a été introduite par [Shi *et al.* 2005], elle consiste à exécuter les deux algorithmes simultanément. Après n itérations, choisir P individus de chaque algorithme pour les échanger. L'individu ayant une meilleure fonction « objectif » a une forte possibilité d'être choisi. Les principales phases de la méthode proposée sont données ci-dessous :
1. Initialiser les deux algorithmes EA et PSO.
2. Exécuter les deux algorithmes EA et PSO simultanément.
3. S'arrêter si le critère d'arrêt est vérifié dans l'un des deux algorithmes et sauvegarder la meilleure solution finale.
4. Réaliser le mécanisme hybride après n itérations en choisissant les P individus en fonction de leur fonction « objectif » des deux algorithmes puis les échanger. Revenir à l'étape 3.
Les auteurs [Alliot *et al.* 2012] utilisent des algorithmes (algorithme génétique (GA) et Branch and Bound par intervalles (BBI)) effectués dans des processus indépendants, ces algorithmes communiquent via une mémoire partagée. Le GA est appliqué pour assurer une bonne exploration de l'espace de recherche. Le BBI réduit l'espace de recherche en enlevant plus sérieusement des sous-espaces ne pouvant pas comporter de solution optimale. La meilleure solution trouvée est mémorisée en mémoire partagée par le BBI, puis ajoutée à la population de l'Algorithme GA pour éviter la convergence rapide de la population vers des minima locaux.

3.2.4 Stratégie de contrôle

- intégratif : l'un des algorithmes est intégré dans l'autre, par exemple une métaheuristique est intégrée dans une méthode exacte ou inversement.
- Collaboratif (coopérative) : dans l'hybridation collaborative, les deux approches communiquent entre elles en échangeant des informations sé-

quentiellement ou parallèlement.
Les auteurs [Cotta *et al.* 1995] associent en parallèle un algorithme Branch and Bound (BB) et un algorithme génétique (GA) pour résoudre le problème du voyageur de commerce. Le GA donne au BB un majorant du minimum global, afin d'écarter les sous-problèmes sous-optimaux. Le BB ajoute dans la population de l'algorithme GA les composants utiles. Mais l'échange des informations avec des vitesses différentes dans les processus est une mission difficile. En plus de ça , ajouter des bonnes solutions en début de la convergence de l'algorithme GA est susceptible de diminuer la diversité de la population. Les auteurs ont utilisé deux méthodes : une hybridation intégrative (remplacer le croisement de GA par une recherche arborescente) ou un schéma maître esclaves (composé d'un BB et de m GA en parallèle).
Les auteurs [Cotta & Troya 2003] ont appliqué l'algorithme Branch and Bound pour réaliser le meilleur croisement probable entre les individus dans un algorithme évolutionnaire, l'opérateur de croisement est remplacé par un algorithme de BB qui choisit la meilleure combinaison possible.
Les auteurs [Feltl & RaidI 2004] implémentent la solution de la relaxation d'un programme linéaire à une fonction d'arrondissement aléatoire pour produire les solutions initiales d'un Algorithme génétique (GA), afin de résoudre le problème donné. Si une solution est impossible, une étape de correction est nécessaire en utilisant une fonction de réparation aléatoire.
L'hybridation collaborative parallèle est réalisée par la communication parallèle entre les métaheuristiques et les méthodes exactes. Par exemple, appliquer l'algorithme Branch and Bound pour accélérer la recherche et l'algorithme génétique pour diminuer l'espace de recherche.
Les auteurs [Gallardo *et al.* 2007] ont présenté des mécanismes parallèles : la méthode exacte définit des emplacements prometteurs de l'espace de recherche, cette dernière sera explorée ensuite par la métaheuristique. L'algorithme mimétique donne un minimum global à l'algorithme arborescent afin de réduire l'espace de recherche, et de diriger en retour vers des emplacements prometteurs de l'espace de recherche.

3.3 Conception des Métaheuristiques hybrides

Selon les auteurs [Blum *et al.* 2011], avant de concevoir une métaheuristique hybride, il est important de déterminer un choix approprié au problème d'optimisation donné. Ceci peut être réalisé en répondant aux questions suivantes :
Quel est l'objectif de l'optimisation ? A-t-on besoin d'une bonne solution raisonnable ? Et si cette solution est nécessaire très rapidement ou non ? Ou le temps de calcul n'est pas une contrainte pour obtenir une très bonne solution ?
Ces questions nous dirigent en général vers l'utilisation des Métaheuristiques ou des méthodes exactes. Dans ce cas, lorsque la solution requise est très bonne et que les algorithmes exacts existants ne peuvent pas l'obtenir dans un temps raisonnable, on doit connaître la réponse à la question suivante afin de décider de développer une métaheuristique hybride.
Existe-t-il des Métaheuristiques qui peuvent obtenir la solution requise pour le problème d'optimisation donné ? Si non, peut-on améliorer les Métaheuristiques disponibles pour mieux répondre à ce problème d'optimisation ?

Si non, la décision est de développer une métaheuristique hybride et on devra connaître la réponse à la question suivante :
Quel type de métaheuristique hybride fonctionnera bien pour ce problème d'optimisation ? Cependant, jusqu'à présent, il n'y a pas de réponse exacte à cette question car il est difficile d'établir des directives pour développer une métaheuristique hybride performante, mais ce qui suit peut nous aider :
En recherchant attentivement dans la littérature les algorithmes d'optimisation les plus efficaces pour le problème d'optimisation considéré ou pour des problèmes d'optimisation similaires, et en étudiant les différentes manières de combiner les caractéristiques les plus prometteuses des algorithmes d'optimisation sélectionnés à hybrider.
En outre, afin de définir des directives utiles pour le développement d'une nouvelle métaheuristique hybride, il est crucial d'améliorer la méthodologie de recherche utilisée par la combinaison des différents composants algorithmiques sans identifier les contributions de ces composants à la performance de la métaheuristique hybride résultante. La méthodologie utilisée consiste en des modèles théoriques pour les caractéristiques des métaheuristiques hybrides. Il peut être expérimental, comme ceux utilisés en sciences naturelles. De plus, les tests et l'évaluation statistique des résultats obtenus doivent également être inclus.

3.3.1 Hybridation Métaheuristiques/Fouille de données

Une autre hybridation a été proposée selon la taxonomie des auteurs [Jourdan *et al.* 2006], elle sert à développer les métaheuristiques par l'intégration de techniques la fouille de données.
La Figure 3.2 illustre les différents chemins pour intégrer la connaissance dans la Métaheuristique.

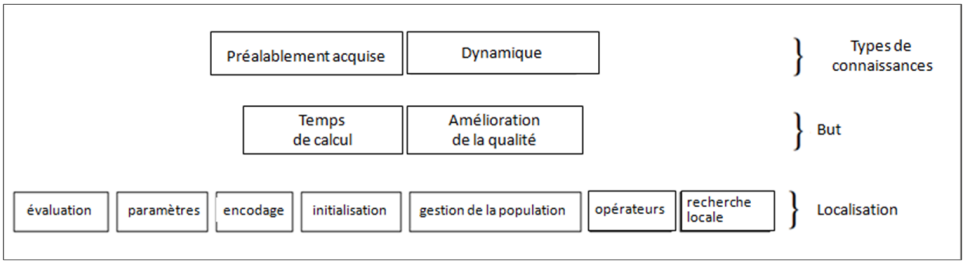

FIGURE 3.2 – *Taxonomie proposée par [Jourdan et al. 2006].*

1er **chemin :**
Deux types de connaissances peuvent être distingués : une connaissance Préalablement acquise (A Priori Knowledge) et une connaissance dynamique, qui est extraite ou découverte au cours de la recherche (Dynamic knowledge).
2ème **chemin :** Il définit le but de la coopération soit de réduire le temps de calcul (speeding up) par l'approximation de la fonction fitness ou par la réduction significative de l'espace de recherche soit par l'amélioration de la qualité de la recherche par l'introduction de connaissance dans les opérateurs ou dans les différentes parties des métaheuristiques.
3ème **chemin :**

Il définit la partie concernée par l'incorporation des techniques de la fouille de données dans les métaheuristiques (paramétrage, encodage, évaluation, initialisation, opérateurs, etc.).
L'hybridation de Métaheuristique avec la fouille de données est moins étudiée pour le problème d'optimisation multi-objectif.
L'inconvénient de cette hybridation est le réglage des paramètres.
Les auteurs ([Dalboni *et al.* 2003]. [Santos *et al.* 2006]) ont hybridé l'algorithme génétique avec l'algorithme Apriori (pour générer les règles d'association) pour le problème de routage de véhicule.
LEM ([Michalski 2000]. [Michalski *et al.* 2000]) intègre une composante d'apprentissage symbolique au calcul évolutionnaire ; il cherche des règles expliquant les différences entre le meilleur et le moins performant dans la population et génère de nouveaux individus basés sur des modèles spécifiés dans ces règles. LEMMO est l'extension LEM, elle est proposée par les auteurs [Jourdann *et al.* 2005]. [Falkenauer 1994]), pour l'optimisation multi-objectif afin de chercher des règles qui expliquent la domination des individus entre eux d'un point de vue multi-objectif. LEMMO génère de nouveaux individus grâce aux règles en créant des solutions qui correspondent à des règles positives et ne correspondent pas aux règles négatives. L'approche a donné de bons résultats dans l'application de système d'eau (accélérer l'algorithme multi-objectif et améliorer la qualité de solutions).
Un algorithme hybride de classification ACO-AC a été proposé par les auteurs [Shahzad & Baig 2011]. L'hybridation est faite en combinant l'exploitation des règles d'association et la classification supervisée en utilisant l'algorithme de colonie de fourmis pour la découverte des règles de haute qualité pour améliorer la performance des classifieurs. L'algorithme de colonie de fourmis est utilisé pour extraire un sous ensemble de règles d'association. Ces règles sont utilisées pour classifier les données invisibles.

3.3.2 Hybridation métaheuristiques/métaheuristiques

L'hybridation de différentes métaheuristiques est aujourd'hui très répandue, en particulier en ce qui concerne l'implémentation des approches à solution unique dans des approches basées sur des populations [Blum *et al.* 2010].
L'optimisation par colonies de fourmis implémente généralement des mécanismes de recherches locales pour améliorer les solutions générées au cours du processus de recherche. Ces algorithmes inspirés par la nature sont utiles pour l'exploration de l'espace de recherche et l'identification des espaces garantissant des solutions de meilleure qualité. La puissance de la recherche locale réside dans la capacité d'obtenir rapidement de meilleures solutions au voisinage de solutions de départ.
Les approches basées sur la population permettent de définir les zones prometteuses de l'espace de recherche dans lesquelles les algorithmes de recherche locale peuvent ensuite trouver rapidement les meilleures solutions [Blum *et al.* 2010].
Dans [Tan *et al.* 2003], une technique de classification évolutionnaire hybride à deux phases a été proposée afin d'extraire les règles de clas-

sification à utiliser dans le domaine de la prévention médicale. Dans la première phase, un algorithme évolutionnaire est utilisé pour limiter l'espace de recherche en évoluant une population de règles candidates utiles.
La programmation génétique est appliquée pour évoluer les attributs nominaux pour la structuration des règles et l'algorithme génétique est utilisé afin d'optimiser les attributs numériques, ce qui permet une classification concise des règles. Les règles candidates sont ensuite utilisées dans la deuxième phase afin d'optimiser l'ordre et le nombre de règles dans le but de former un ensemble de règles cohérentes, précises et compréhensibles.
Une métaheuristique hybride (HColonies) à deux phases a été proposée par les auteurs [AlMuhaideb & Menai 2014]pour la classification des données médicales. Dans la première phase, ACO (AntMiner+) construit la liste de décision à partir de sous ensembles d'apprentissage de données, cette liste est utilisée dans la deuxième phase ABC (optimisation par colonie des abeilles artificielles) comme une solution initiale. La nouvelle variante des opérateurs de l'algorithme ABC a été proposée suivant la tâche de la classification. Cette approche est fiable pour améliorer les résultats obtenus par un classifieur de base en termes d'exactitude, elle est robuste à la modification de ces paramètres, mais elle s'affaiblit de problème multi classe.
DE–HS–HJ est une hybridation de trois Métaheuristiques : algorithme d'évolution différentielle (DE), algorithme de la recherche harmonie (HS) et algorithme de la recherche locale « Hooke and Jeeves » (LS-HJ). Cette approche a été proposée par les auteurs [Duan *et al.* 2013] avec une implémentation de 18 chemin différents de L'algorithme de recherche locale et une évaluation sur 19 problèmes en termes de l'exactitude et d'efficacité.
Les solutions initiales sont obtenues par la combinaison de l'algorithme DE et l'algorithme HS. Les meilleurs résultats sont obtenus avec l'implémentation de trois stratégies de recherche locale
- Application de la recherche locale sur chaque solution générée avec une probabilité et sur chaque solution mise à jour,
- Application de la recherche locale sur chaque solution générée avec une probabilité,
- Application de la recherche locale sur chaque solution générée avec une probabilité et sur la meilleure solution globale mise à jour.
La classification des données médicales est un domaine de recherche qui permettra d'améliorer le coût, l'accessibilité et la qualité des soins de santé. La complexité associée à la classification des données médicales interdit l'utilisation des méthodes exactes.
L'article de [AlMuhaideb & Menai 2013]présente un état d'art de différentes approches de la classification de données médicales.
Une approche hybride parallèle a été proposée par les auteurs [Boucheham *et al.* 2015] pour résoudre le problème de la sélection des biomarqueurs significatifs à partir d'un ensemble volumineux de données génétiques en génomique. Elle consiste à utiliser un modèle parallèle GIM-FS composé de deux algorithmes ACO+PSO, avec une approche wrapper pour la classification cancer.
L'approche proposée peut être facilement étendue à tout problème de

sélection d'attributs et elle convient aux grands ensembles de données.
Dans [Jimenez & Rodriguez 2014], Résoudre le problème d'inférence phylogénétique par l'application des schémas MPI/OpenMP de deux métaheuristiques (algorithme génétique et algorithme des lucioles).
Cette approche permet une meilleure exploitation des ressources matérielles, comme perspectives, étudier des nouvelles approches parallèles, développement des algorithmes asynchrones dans un environnement à mémoire partagée avec l'application d'un grand nombre de noyaux de traitement.
Appliquer les modèles basés sur l'île, et les algorithmes parallèles bio inspirés pour améliorer les résultats biologiques. Évaluer les algorithmes multi-objectifs basés sur la décomposition.

3.4 Métaheuristiques hybrides pour la sélection des attributs

La sélection des attributs permet la découverte d'un sous-ensemble des attributs avec une meilleure prédiction de la classification et permet d'obtenir un modèle précis plus simple et plus compréhensible de la classification, avec un temps d'exécution réduit.
La sélection des attributs est un problème combinatoire qui nécessite une connaissance de haut niveau pour le résoudre.
L'hybridation des métaheuristiques peut résoudre ce type de problème et peut trouver des solutions optimales.
Les algorithmes génétiques ont été utilisés pour le problème de sélection d'attributs, par exemple, hybrider l'algorithme génétique avec une recherche locale afin de sélectionner le sous-ensemble d'attributs le plus approprié avant de construire le modèle de classifieur [Oh *et al.* 2004].
Un autre exemple est le travail des auteurs [Vieira *et al.* 2010], ils ont utilisé deux colonies de fourmis artificielles coopératives : une pour déterminer le nombre d'attributs à sélectionner et la seconde pour sélectionner les attributs en fonction de la cardinalité donnée par la première colonie.
Les auteurs [Boucheham *et al.* 2015]ont proposé une méthode basée sur un ensemble des métaheuristiques parallèles coopératives (ECPM-FS) pour la découverte d'un nombre prédéfini de marqueurs utiles en génomique. La coopération de plusieurs métaheuristiques inspirées par la nature (optimisation par essaims particulaires, optimisation par colonie de fourmis et algorithme génétique) s'est révélée être une solution efficace pour éviter une convergence prématurée tout en accélérant le processus de recherche.
Cette approche pourrait être étendue à tout problème de la sélection des attributs par la mise en œuvre d'un autre cadre pour la métaheuristique parallèle et coopérative avec des différentes topologies et d'une autre stratégie d'initialisation de la population.

3.5 Métaheuristiques hybrides pour la sélection des instances

La sélection des instances est principalement la version orthogonale du problème de sélection des attributs, car il implique la découverte d'un sous ensemble des instances pertinentes utile pour la construction d'un classifieur performant avec une meilleure précision prédictive, ou de la même manière qu'un classifieur construit à partir de l'ensemble complet des instances [de Souza *et al.* 2010]. Il est clair que la sélection d'instances nettoie le jeu de données en cours d'utilisation : elle supprime les instances non pertinentes, bruyantes et redondantes. La sélection des instances permet d'améliorer l'efficacité des calculs, l'algorithme d'apprentissage ne prend pas en compte qu'un sous-ensemble des données d'origine, et permet l'induction des meilleurs classifieurs [Blum & Langley 1997].
La sélection des instances est un problème combinatoire qui nécessite une connaissance de haut niveau pour le résoudre.
L'hybridation des métaheuristiques peut résoudre ce type de problème et peut trouver des solutions optimales.
Les données FlowShop typique enregistrées pendant le processus de production peuvent ne pas être un bon exemple pour acquérir des connaissances importantes. Par conséquent, la sélection des instances pertinentes peut être considérée comme un problème d'optimisation et résolue par les métaheuristiques. Les auteurs [Balasundaram & Sathiya 2019] ont proposé une approche hybride pour la sélection des instances utiles basée sur l'algorithme génétique et la recherche par dispersion pour minimiser le critère de l'ordonnancement par permutation du flowshop. Les résultats expérimentaux sont performants et concurrents.

3.6 Métaheuristiques hybrides pour la classification supervisée

La première classe d'utilisation des métaheuristiques hybrides pour des tâches de classification supervisée concerne l'utilisation des métaheuristiques pour le problème de sélection d'attributs en plus du classifieur employé. En effet, la sélection d'un sous ensemble le plus pertinent en entrée pour construire le modèle joue un rôle essentiel dans la classification. L'utilisation des métaheuristiques hybrides est importante pour trouver des solutions de meilleure qualité, en particulier pour des applications réelles (telles que la planification du personnel et des machines, l'alignement des protéines). Un exemple de problèmes de classification du monde réel est le travail de [Tantar *et al.* 2008] qui ont proposé une approche hybride des métaheuristiques (algorithme génétique et recuit simulé) pour prédire la structure de la protéine.
Les auteurs [Soltani *et al.* 2015]ont proposé une approche hybride pour améliorer la classification par l'intégration de la Programmation génétique pour obtenir les populations des meilleures fonctions discriminantes (une population pour chaque classe) et dans une deuxième phase, ils ont employé l'algorithme de colonie de fourmis, appelée GP-Ant, et l'algorithme génétique, appelée GP-GA, pour trouver la meilleure combinaison de

fonctions discriminantes obtenue dans la phase précédente.

3.6.1 Métaheuristiques hybrides pour la découverte des règles de classification

Les données analysées à l'aide de techniques de la fouille de données sont incomplètes (valeurs d'attributs manquantes ou présentant un intérêt particulier, ou ne comportant que des données agrégées), bruyantes (comportant des erreurs ou des valeurs aberrantes s'écartant de celles attendues) et incohérentes (comportant par exemple des anomalies).
On analyse les données de la base de données pour faire une description exacte ou construire un modèle ou une règle de classification pour chaque catégorie, et utiliser la règle de classification pour classer des nouvelles instances.
La classification consiste à trouver un ensemble de règles puissant à partir de l'ensemble de données. Une règle est de la forme : IF (conditions) THEN (classe), ce qui signifie qu'une instance vérifie les conditions de la règle, il est prédit que la classe est mentionnée dans la règle.
Une règle de la classification est organisée comme suit : IF <terme1 ET terme2 ET ... > ALORS <classe>. Chaque terme est un triple <attribut, opérateur, valeur>, où valeur est l'une des valeurs appartenant au domaine de l'attribut.
Les métaheuristiques hybrides sont employées afin de trouver efficacement des modèles de classification quasi optimaux ou optimaux.
Dans le travail des auteurs [Rajiv *et al.* 2012], il y a une proposition d'une approche hybride entre l'algorithme d'optimisation par essaims particulaires et l'algorithme d'optimisation par colonie d'abeilles. La qualité de la règle peut être estimée en termes de précision. Avec cette méthode hybride la règle trouvée améliore la précision et l'efficacité du résultat dans le domaine médical de traitement des patients. Cette méthode met à jour les informations de la particule dans chaque interaction en appliquant l'équation de mise à jour de l'algorithme d'optimisation par colonie d'abeilles. L'hybridation réalisée pénalise fortement les faux positifs, ce qui est une propriété voulue pour l'exploration de données dans le domaine médical.
La méthode hybride de l'algorithme d'optimisation par essaims particulaires et l'algorithme optimisation par colonie de fourmis [Holden & Freitas 2008] génère souvent des ensembles de règles beaucoup plus simples et plus petits. Les auteurs ont réalisé un modèle de l'exploration de données permettant de découvrir des connaissances non seulement précises, mais également compréhensibles par l'utilisateur. Cette approche est au moins compétitive avec un algorithme de classification standard PART en termes de précision.
Dans l'article des auteurs [Chen & Ludwig 2018] une implémentation discrète basée sur l'algorithme d'optimisation par essaims particulaires avec une stratégie de recherche locale (DPSO-LS) a été proposée. L'approche de recherche locale permet d'excéder les optima locaux afin d'améliorer la qualité de la solution.

3.7 Métaheuristiques hybrides pour la segmentation

La segmentation (clustering) est une technique de fouille de données, elle est employée pour regrouper les données dans différents sous-groupes afin de récupérer des informations importantes à partir d'un ensemble volumineux de données. Une revue de la littérature existante montre qu'il existe de nombreux algorithmes de clustering pour différents domaines de problèmes [Oduntan & Thulasiraman 2018]. Le problème de regroupement des données est un problème combinatoire qui implique une connaissance de haut niveau pour le résoudre. Les métaheuristiques hybrides peuvent résoudre ce type de problème et peuvent trouver des solutions optimales.
Les approches basées sur les métaheuristiques remplacent l'algorithme de classification classique avec l'objectif de développer les performances [Mageshkumar *et al.* 2018], les auteurs ont proposé une nouvelle approche hybride ACO – ALO (optimisation par colonie de fourmis et optimisation par Antlion) pour résoudre le problème du clustering de données. L'opérateur de mutation de Cauchy est employé avec cet algorithme pour éviter de tomber dans les minima locaux. L'objectif principal est de diminuer la distance intra-cluster dans le problème de clustering.
L'analyse de données haute dimension et ses grandes chances de sur ajustement posent de grands défis pour la construction de modèles efficaces dans des applications pratiques. Pour surmonter ces problèmes, des algorithmes d'intelligence en essaim peuvent être utilisés. Cependant, l'équilibre entre la recherche globale et la recherche locale tout au long d'une analyse est essentiel au succès d'un algorithme d'optimisation par l'intelligence [Sun & Peng 2014], ces auteurs ont présenté une hybridation de l'optimisation par essaim de particules (PSO) avec un algorithme de recherche gravitationnelle (GSA). Dans le cadre de garantir une meilleure recherche, ils ont en profité de la capacité d'exploitation dans le PSO et la capacité d'exploration dans le GSA dans la mise à jour des équations de vitesse. Ils ont appliqué également l'algorithme proposé à l'analyse par grappes de données de puces à ADN.
Une approche hybride a été proposée par [Oduntan & Thulasiraman 2018] pour la mise en clusters combinant une technique de regroupement inspirée par la nature et une métaheuristique qui utilise l'historique de recherche et les stratégies de voisinage dynamiques pour trouver la solution globale optimale (recherche tabou). Il s'agit d'une nouvelle métaheuristique hybride du clustering mettant l'accent sur la flexibilité et moins de spécificité.

3.8 Conclusion

Dans ce chapitre, on a décrit les métaheuristiques hybrides, comment concevoir, ainsi que la définition des critères de la classification des métaheuristiques. On a pu remarquer, au fur et à mesure d'un court état de l'art pour chaque utilisation, que leur implémentation en extraction de connaissances est un domaine intéressant pour résoudre des problèmes des différentes tâches telles que : la sélection d'attributs, la sélection d'instances, la classification supervisée et la segmentation.

Afin de contribuer à des exemples pour la résolution de ces problèmes, on présente dans la deuxième partie de cette thèse les approches proposées.

B

Contributions

Approches Hybrides Pour La Sélection Des Attributs

4

Sommaire

4.1 Introduction

Comme on l'a vu dans la partie background et état de l'art, il existe plusieurs façons de résoudre les problèmes d'extraction des connaissances à partir des données. Parmi celles-ci, il y a les méthodes hybrides, qui consistent à combiner au moins deux méthodes de résolution. Dans ce chapitre, on présente la sélection du sous ensemble des attributs pertinents basée sur l'optimisation par colonie de fourmis(ACO) et l'Adaboost (arbres de décision C4.5) pour l'amélioration de la classification des données [Alaoui & Elberrichi 2018].
Par exemple, pour la prédiction de la maladie, le coût de diagnostic médical peut être réduit en évitant de nombreux de tests par la sélection des attributs importants pour la classification supervisée, un modèle d'apprentissage robuste sera ainsi construit.
Afin d'améliorer la précision de la classification et de construire un modèle robuste, on propose une stratégie d'hybridation entre l'ACO et l'Adaboost(C4.5). L'approche proposée et les résultats expérimentaux obtenus avec discussion sont présentés. Ensuite, on donne une version améliorée de l'approche proposée [Alaoui & Elberrichi 2020a].

4.2 Motivations

Pour certains domaines d'application, il est essentiel de produire des procédures de classification explicites pour l'utilisateur. Par exemple, un arbre de décision pour un diagnostic médical. Les données médicales contiennent de nombreux attributs, le diagnostic des maladies est coûteux car plusieurs tests sont nécessaires pour prédire la maladie.
La complexité de calcul d'un algorithme de classification peut souffrir du problème de la dimensionnalité causée par de nombreux attributs. Souvent, un jeu de données a trop d'attributs non pertinents. Ne sélectionner que les plus importants ne peut être que bénéfique. La réduction de la dimensionnalité joue un rôle important dans les différents domaines car les jeux de données contiennent des attributs non pertinents et redondants.
De nombreux problèmes liés à la sélection des attributs ont été montrés pour être NP-Hard [Cotta & Moscato 2003] et peuvent être résolus par les métaheuristiques. Différentes techniques d'optimisation telles que les algorithmes génétiques [Feng *et al.* 2007], Recherche Tabou [Hongbin & Guangyu 2002], Recuit Simulé [Ronen & Jacob 2004] et Optimisation par Essaims particulaires([Bing *et al.* 2014]. [Liam *et al.* 2012]) ont été utilisés pour résoudre le problème de sélection des attributs. Parmi ces approches, l'optimisation par colonie de fourmis (ACO) a été explorée d'une manière significative pour résoudre ces problèmes.
Cette méthode est prometteuse pour une bonne sélection des attributs ([AlAni 2005].[Kumar & Mewada 2011]. [Shahzad 2010]).
Plusieurs approches ont été proposées dans la littérature pour le problème de sélection des attributs dans plusieurs domaines. Par exemple le domaine médical, certains algorithmes sont présentés dans le Tableau4.1
Les métaheuristiques sont utilisées pour déterminer le sous-ensemble des attributs optimal avec une précision de classification améliorée dans le diagnostic de la maladie. Les méthodes de sélection des attributs (Filter et

wrapper) sont basées sur des algorithmes évolutifs et en essaim.

Références	Algorithmes	Type	Jeux de données
[Subanya & Rajalaxmi 2014]	Colonie d'abeilles artificielles binaires / K –le plus proche voisin (BABC k – NN)	Wrapper	Maladie cardiaque
[Harb & Desuky 2014]	Optimisation par Essaims particulaires (PSO)	Filter et Wrapper	Cancer du sein, bilan cardiaque, dermatologie
[Kumar *et al.* 2014]	Algorithmes Génétiques /SVM	Wrapper	Wisconsin cancer du sein, cancer du sein, dermatologie, chirurgie thoracique, Masses mammographiques,
[Brezocnik *et al.* 2018]	Algorithmes d'intelligence en essaim, revues	Filter, Wrapper et intégré	Données médicales et autres
[Darwish *et al.* 2018]	Algorithmes bio-inspirés	Wrapper	cancer du sein (WBCD et WPBC)
[Nematzadeh *et al.* 2019]	Algorithme de baleine avec congestion mutuelle	Filter	Côlon

TABLE 4.1 – *Méthodes de sélection des attributs Filter et Wrapper.*

4.3 Travaux Connexes

Certains attributs de l'ensemble de données peuvent être inutiles et peuvent être donc éliminés avant l'apprentissage. Cette section explore les travaux de recherche liés au travail proposé.
L'ACO est un algorithme puissant et utile, il a été employé dans plusieurs recherches [Kabir *et al.* 2013]pour trouver les attributs pertinents. Pendant l'exécution du processus de l'ACO, un certain nombre de fourmis artificielles parcourent l'espace de recherches pour construire d'une manière itérative des sous-ensembles des attributs. Lors de la construction, des approches existantes identifient la taille des sous-ensembles construits par un nombre fixe à chaque itération, alors que d'autres suivent la stratégie de la recherche séquentielle en l'avant (SFS). Afin de mesurer les valeurs heuristiques des attributs pendant la sélection, certaines approches utilisent l'approche Filter. La plupart des algorithmes conçoivent les règles de mise à jour des phéromones sur la base des résultats des évaluations des sous-ensembles sélectionnées des attributs. À cet égard, des algorithmes

utilisent l'approche wrapper. Les algorithmes ACO pour la sélection des attributs avec leurs références sont indiqués dans le Tableau 4.2

Algorithmes	Types	Références
ACO/ Réseaux de Neurones artificiels	Wrapper	[Sivagaminathan & Ramakrishnan 2007]
ACO /Plus Proche Voisin	Wrapper	[Aghdam *et al.* 2009]
ACO/Analyse Discriminante Linéaire	Wrapper	[Khushaba *et al.* 2008]
ACO/Plus Proche Voisin	Wrapper	[Kanan *et al.* 2007]
ACO /Réseaux de Neurones artificiels	Wrapper	[Ani 2008]
ACO /Modèle à Variables Latentes	Wrapper	[Robbins *et al.* 2008]
ACO/ Théorie des ensembles approximatifs	Filter	[Ke *et al.* 2008]
ACO /Réseaux de Neurones artificiels	Hybride	[Kabir *et al.* 2013]

TABLE 4.2 – *Exemples de la sélection des attributs par ACO*

Les auteurs [Rizwan *et al.* 2017] ont proposé une approche filter basée sur l'algorithme de colonies de fourmis, la pertinence maximale et la redondance minimale des attributs pour une évaluation efficace des sous-ensembles des attributs.
L'avantage de l'algorithme proposé est que la sélection de sous-ensembles des attributs est basée sur l'approche filtre qui est, en termes de calcul, moins coûteuse que la plupart des méthodes wrapper. La méthode de type filtre est généralement inférieure à la méthode de type wrapper en termes de la précision de la classification. Les auteurs ont proposé d'utiliser un meilleur mécanisme de recherche, capable de parcourir tous les sous-ensembles des attributs clés et d'évaluer ces sous-ensembles avec une mesure d'évaluation des attributs plus complète, et comme ça la méthode de type filter peut être comparable aux méthodes de type wrapper.
Cette contribution garantit le choix des attributs les plus pertinents pour le concept cible, faiblement redondants les uns par rapport aux autres et avec des meilleures prédictions de la classification. Les auteurs utilisent les informations mutuelles pour mesurer la redondance des attributs et la pertinence avec le concept cible.
L'évaluation des sous ensembles des attributs choisis est déterminée à l'aide de la stratégie de la pertinence maximale et la redondance minimale des attributs.
La performance de la méthode proposée a été comparée à des classifieurs populaires d'apprentissage automatique (C4.5, KNN, RIPPER) et à d'autres algorithmes de sélection des attributs (ACO, PSO, Tabu, GA) répartis sur onze jeux de données. Il a été conclu que cette approche, comparée avec des méthodes existantes, permet une meilleure précision de la classification et utilise un nombre réduit d'attributs.
En perspective, les auteurs ont proposé de trouver des mécanismes de recherche hybrides pour améliorer l'approche. De plus, le réglage des paramètres peut produire une recherche optimisée. Et la sélection des attributs peut être étendue pour contenir la fouille de flux de données.
Les auteurs [El-Houby *et al.* 2017] ont proposé une approche d'optimisation par colonie de fourmis pour la sélection des attributs pertinents. La sélection séquentielle en avant des attributs est utilisée. Ils choisissent

donc les attributs les plus pertinents d'une manière séquentielle jusqu'à ce que la précision soit améliorée. La précision a été mesurée par le classifieur K-plus proche voisin.
L'algorithme de colonies de fourmis a été appliqué à la sélection des attributs car les fourmis découvrent de bons chemins des attributs dans l'espace de recherche.
L'algorithme proposé emploie plusieurs groupes de fourmis ; en utilisant des critères différents (les plus proches et les plus éloignés), chaque groupe choisit les attributs candidats pour atteindre la meilleure solution possible. Cette approche utilise une fonction de fitness composée de la valeur heuristique et de la valeur de phéromone. Les informations heuristiques sont représentées avec la valeur de séparabilité de classe (CS) de l'attribut candidat. Le calcul de la valeur de phéromone est basé sur la précision de la classification obtenue en ajoutant un attribut candidat. La meilleure solution globale est obtenue à partir des différents critères appliqués.
La méthode proposée a été appliquée sur divers jeux de données médicales (Maladie cardiaque, Cancer du sein, Thyroïde) donnant de bons résultats.
En perspective, d'autres critères peuvent être appliqués pour obtenir les sous-ensembles d'attributs, le réglage des paramètres peut être étudié.
Les auteurs [Jameel & ur rehman 2018] ont proposé un mécanisme de la sélection des attributs basé sur l'optimisation par colonies de fourmis de type wrapper.
L'avantage de l'ACO est qu'il suit la nature d'une manière parallèle, fournit une structure multi-agents, des retours positifs et une bonne capacité de recherche locale et globale. Les auteurs ont utilisé l'incertitude symétrique pour calculer la fonction heuristique. La symétrie est une caractéristique requise pour mesurer les corrélations entre les attributs. Cela réduit le nombre de comparaisons. La fonction heuristique n'est pas influencée par les attributs à valeurs multiples comme dans l'information de gain, et ses valeurs sont normalisées dans la plage [0, 1]. La fonction fitness est la précision de la classification obtenue par le classifieur bayésien naïf. La méthode proposée a amélioré la précision de 5% en moyenne en expérimentant tous les jeux de données utilisés lors de la sélection du sous-ensemble des attributs.
Les travaux futurs seront orientés vers l'application des autres algorithmes de la classification de la fouille de données.

Les meilleurs résultats obtenus sur les jeux de données liés au travail dans des travaux connexes sont indiqués ci-dessous :
— L'approche colonie d'abeilles artificielles binaire (BABC) k – NN [Subanya & Rajalaxmi 2014] a été proposée pour déterminer les meilleurs attributs du diagnostic des maladies cardiaques. Elle donne une précision de 92, 4% pour ce jeu de données ;
— Une méthode de sélection des attributs basée sur l'optimisation par essaims particulaires et la corrélation en tant qu'un mécanisme d'évaluation du sous-ensemble a été utilisée [Harb & Desuky 2014] avec un jeu de donnée médical cancer du sein. Cette méthode a été appliquée avec 05 classifieurs weka comme l'arbre de décision c4.5 avec une précision de

74.13%.
— Les auteurs [Kumar *et al.* 2014] ont proposé un algorithme d'optimisation basé sur l'algorithme génétique, qui permet d'optimiser les valeurs de paramètre pour SVM. La méthode GA-SVM a été appliquée pour supprimer les attributs non significatifs et pour trouver efficacement les meilleures valeurs de paramètres dans les jeux des données médicales mammographies (précision 0,781% et F-mesure 0,78%) et cancer du sein au Wisconsin (avec une précision de 0,97% et F – mesure 0,97%) ;
— Un nouvel algorithme de sélection des attributs FEAST (Feature subset sElection Algorithm based on aSsociaTion rule mining) a été proposé dans [Wang & Song 2012] sur la base de l'exploration de règles d'association. Les résultats obtenus sur les jeux de données du monde réel montrent que FEAST donne un résultat compétent en termes de précision de la classification, par exemple l'arbre de décision C4.5 avec l'algorithme FEAST pour le jeux des données tumeur primitive avec une précision de 43,65% ;
— Une méthode de sélection d'attributs appelée : recherche avancée séquentielle filtrée et prise en charge (FS-SFS) avec SVM est proposée dans [Liu & Zheng 2005], par exemple le résultat de la précision du jeu de données patients postopératoires (patients PO) est égal à 73,1%.

4.4 Approche proposée

L'élément important de cette approche est de trouver un sous-ensemble d'attributs à partir d'un graphe connecté N^2, où N est le nombre d'attributs présent dans les ensembles de données à l'exclusion de l'attribut cible. Le graphe constitue l'espace de recherche de déplacement des fourmis, les liens représentent la connexion entre les attributs d'un jeu de données et les nœuds sont les attributs. Chaque fourmi sélectionne d'une manière probabiliste des attributs à la base de phéromones et de valeurs heuristiques associées à chaque lien. Lorsqu'une fourmi termine sa tournée (elle construit une solution candidate à partir de cet espace de recherche en parcourant un chemin des nœuds et des liens), elle utilise ensuite Adaboost avec les arbres de décision pour évaluer la fonction fitness du sous-ensemble des attributs sélectionné par elle. Pour évaluer la précision du classifieur, on utilise une 10-fold cross-validation. L'approche proposée est de type wrapper (cette approche effectue une recherche dans l'espace des sous-ensembles des attributs en utilisant la précision estimée à partir d'un algorithme d'induction en tant que mesure de la pertinence des sous-ensembles).
Ce processus continue jusqu'à ce que le critère d'arrêt soit vérifié. On obtient un sous-ensemble des attributs offrant la meilleure précision. Toutes les étapes d'approche proposées sont indiquées ci-dessous.
Espace de recherche
L'espace de recherche contient N^2 nœuds (N attributs du jeu de données à l'exclusion de l'attribut cible) et le nœud «END» [Shahzad 2010], le lien entre les nœuds indique le choix du nœud suivant (attribut suivant Ai). Le nœud «END» est utilisé pour terminer la recherche d'un sous-ensemble d'attributs (nœuds). Il est connecté à chaque nœud du graphe. Lorsqu'une fourmi sélectionne le nœud 'END', son chemin est complet et il cesse d'ajouter des nœuds (Figure4.1).

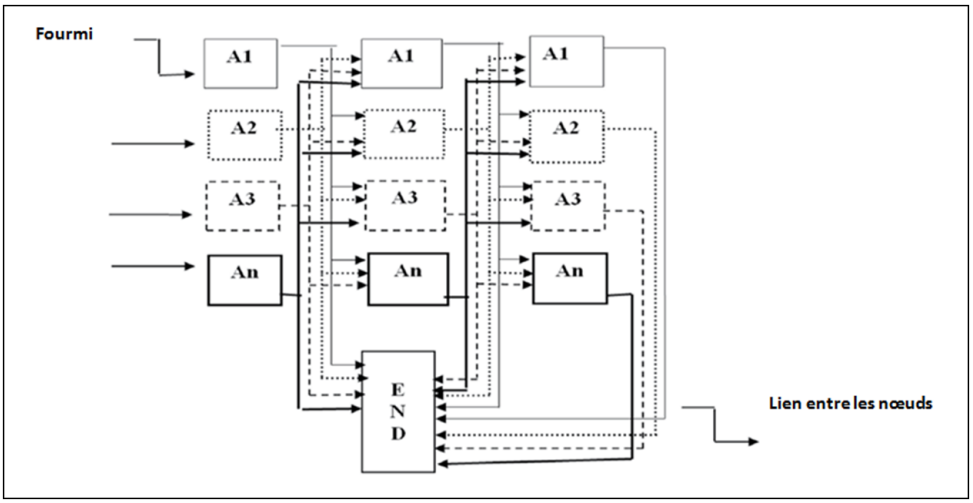

FIGURE 4.1 – *Espace de recherche N^2 pour le chemin d'une fourmi [Shahzad 2010].*

Valeurs de phéromone
La valeur de phéromone est associée à chaque lien entre les nœuds. Au début de l'algorithme, les valeurs de phéromone sont initialisées avec la même valeur aléatoire ([Shahzad 2010]. [Dorigo 2007])(aucun attribut n'est préféré à d'autres attributs), la phéromone initiale est calculée avec (equation(4.1)) :

$$\tau_{ij}(t = 1) = 1/N \tag{4.1}$$

Où :
t = 1 est la première itération de l'algorithme.
N est le nombre d'attributs présents dans le jeu de données à l'exclusion de l'attribut cible.
i est le nœud actuel (attribut), j : nœud suivant (attribut).
Sélection d'un attribut
Une fourmi se déplace du nœud actuel au nœud suivant avec une probabilité calculée en fonction de deux composantes (equation(4.2)) : une valeur de phéromone (elle est présente sur le lien entre le nœud actuel et le prochain nœud) et des valeurs heuristiques (elle représente la valeur du nœud suivant) :

$$\rho_{ij} = \frac{\left[\tau_{ij}\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}{\sum_{k\in S}\left[\tau_{ik}\right]^{\alpha}\left[\eta_{ik}\right]^{\beta}} \tag{4.2}$$

Où :
τ_{ij} est la valeur de phéromone entre le nœud actuel i et le prochain nœud j.
η_{ij} est la valeur heuristique du nœud j.
S : L'ensemble des nœuds qui n'ont pas été visités par le nœud actuel i, les paramètres α et β influent respectivement sur la valeur de la phéromone et la valeur heuristique.
Valeurs heuristiques
La valeur heuristique indique le mérite de l'attribut, une fourmi utilise

cette valeur pour décider le déplacement d'un nœud à un autre. Le rapport de gain, utilisé comme une valeur heuristique, est calculé de chaque attribut de l'ensemble de données (equation(4.3)). Le rapport de gain «normalise» le gain d'informations de la manière suivante [Quinlan 1993] :

$$GainRation(a_i, S) = \frac{InformationGain(a_i, S)}{Entropy(a_i, S)} \tag{4.3}$$

Où a_i est un attribut et S est un échantillon d'un jeu de données.

Le gain d'information est calculé pour tous les attributs. En conséquence, en prenant en compte uniquement les attributs qui ont réalisé au moins autant que le gain d'information moyen, l'attribut qui a obtenu le meilleur rapport de gain est sélectionné.

Il a été démontré que le rapport de gain tend à surpasser les critères de gain d'information simples, tant du point de vue de la précision que du point de vue de la complexité du classifieur [Quinlan 1988].

Lorsqu'une fourmi veut sélectionner le nœud suivant, le rapport de gain de correspondant est utilisé comme une valeur heuristique et est utilisé dans (equation(4.2)).

Fonction de Fitness

Le sous-ensemble des attributs sélectionnés par une fourmi est mesuré par la fonction de fitness (Evaluer(S)). On utilise Adaboost (C4.5) avec 10-fold cross-validation pour la construction du modèle à partir du sous-ensemble des attributs sélectionné puis on évalue le modèle appris (classifieur).

Pour évaluer la précision du modèle, on utilise 10-fold cross-validation.

Les étapes suivies :

- Au hasard, diviser le jeu de données en 10 sous-ensembles (dix de tailles égales ou proches).
- Entraîner sur 9 sous-ensembles et tester sur 1.
- Répéter 10 fois et prendre la précision moyenne (la moyenne). La fonction fitness est calculée pour chaque échantillon, puis on prend la moyenne de la collection.

La fonction fitness d'un sous-ensemble des attributs particulier est mesurée par (equation(4.4)) :

$$fitness = \sigma / N \tag{4.4}$$

Où σ est le nombre d'instances incorrectement classées par le classifieur et N le nombre total d'instances de test.

Evaluer(S) : Évaluer le sous ensemble d'attributs S (avec (Adaboost(C4.5)) et retourner fitness (equation(4.4)).

Mise à Jour de phéromone

Les valeurs de phéromone sont mises à jour une fois que toutes les fourmis ont terminé leurs visites (à chaque itération). Lors de la prochaine itération, les fourmis peuvent utiliser cette valeur dans leurs recherches. La quantité de phéromone sur chaque lien existant dans le sous-ensemble des attributs en cours (à l'itération t + 1) est mise à jour d'une manière conforme à (equation(4.5)) :

$$\tau_{ij} = (1 - \rho).\tau_{ij}(t) + (1 - \frac{1}{1 + fitness})\tau_{ij}(t) \tag{4.5}$$

Où :
$\tau_{ij}(t)$ est la valeur de phéromone entre le nœud i et le nœud j dans l'itération (t) en cours.
ρ : le taux d'évaporation de la phéromone.
fitness est la qualité du meilleur chemin (sous-ensemble des attributs) construit par une fourmi à itération t. La mise à jour des phéromones a pour objectif d'augmenter les valeurs de phéromones associées aux solutions prometteuses et de réduire celles associées aux mauvaises.

4.4.1 Algorithme proposé ACO/Adaboost (C4.5)

Algorithme 4.1 ACO/Adaboost (C4.5)

Chargez l'ensemble de données.
for <chaque attribut **do**
  Déterminer la valeur heuristique(eq(4.3)).
end
Produire une population de k fourmis.
Initialiser les paramètres : phéromone(eq(4.1)), α, β, ρ, critère d'arrêt.
T = 1 (première itération)
while *le critère d'arrêt n'est pas vérifié* **do**
  for <chaque fourmi i = 1 à k> **do**
    Générer un sous-ensemble S (chaque attribut du sous-ensemble est sélectionné avec probabilité (eq(4.2))).
    Evaluer(S) (eq(4.4)).
  end
  Retourner le meilleur sous-ensemble (avec la meilleure fitness)
  if *la fonction fitness (itération courante) est meilleure que la meilleure globale de l'itération précédente* **then**
    Définir la précision actuelle du sous-ensemble S comme la meilleure précision globale.
  end
  Mettre à jour les valeurs de phéromone (eq(4.5))).
end
Retourner le meilleur sous-ensemble final.

Dans cet algorithme, on charge le jeu de données et on initialise les différents paramètres, l'espace de recherche qui est un graphe N^2 où N est le nombre des attributs du jeu de données à l'exclusion de l'attribut classe.
Les valeurs de phéromone sont initialisées sur toutes les liaisons entre les nœuds (equation(4.1)), une population de fourmis est définie et les paramètres (α, β, ρ) sont initialisés. La valeur heuristique (rapport de gain) est calculée pour chaque attribut dans le jeu de données (equation(4.3)).
Dans une itération de l'algorithme, chaque fourmi sélectionne un attribut initial au hasard et elle complète son chemin pour la suite (sélectionne d'autres attributs). Quand une fourmi commence sa recherche, elle sélectionne le prochain nœud (attribut) selon (equation(4.2)). Chaque fourmi termine son chemin avec le nœud "END", elle a un sous-ensemble des attributs différents, chaque sous-ensemble des attributs retenus est utilisé pour construire un classifieur Adaboost (C4.5), on retient la fonction fit-

ness (equation(4.4)) de chaque sous-ensemble généré par les fourmis, les fourmis recherchent le meilleur sous-ensemble avec la meilleure précision (meilleure fitness) parmi tous les sous-ensembles générés dans cette itération. Si la précision du sous-ensemble des attributs actuel est la meilleure que la précédente, ce sous-ensemble actuel deviendra le meilleur sous-ensemble global et sa précision sera la meilleure précision globale. Pour chaque itération, ce processus se poursuit jusqu'à ce qu'un critère d'arrêt soit vérifié (un nombre défini d'itérations). Le meilleur sous-ensemble global est retenu en tant que sous-ensemble final des attributs.

4.5 Expérimentations, résultats et discussions

Pour évaluer notre approche, on a mené une série d'expérimentations en utilisant Java (langage de programmation) et Weka (code source ouvert développé en Java, Université de Waikato, Hamilton, Nouvelle-Zélande) [Group 2019]. Il n'existe pas de valeurs standard des paramètres utilisés dans l'algorithme ACO. Dans la littérature, cela dépend des expérimentations et de ses résultats.
Les valeurs des paramètres utilisés sont :
— Le nombre de fourmis utilisées : 50
— α (indique l'importance des valeurs de phéromone) : 1
— β (indique l'importance des valeurs heuristiques) : 1
— Seuil de convergence (critère d'arrêt) : 50
— Taux d'évaporation (p) : 0,15

Jeux de données	**Nombre des attributs**	**Nombre des instances**	**Nombre des classes**
Fièvre (Anneal)	38	898	6
Ecoli (Ecoli)	7	336	8
Sein-w (Breast-w)	9	699	2
Maladie cardiaque (Heart-disease)	13	303	5
Tumeur primaire (Primary-tumor)	17	339	22
Masses mammographiques (Mammographic_masses)	5	961	2
Patient postopératoire (Postoperative_patient_data)	8	90	3

Table 4.3 – *Jeux de données de l'UCI.*

La méthode proposée est expérimentée sur 07 jeux de données du référentiel d'apprentissage automatique pour la gestion de la classification des jeux de données médicales [Blake & Merz 1998].
Les détails des jeux de données médicales sont présentés dans le tableau 4.3.
Par exemple, le jeu de données tumeur primaire contient 17 attributs, un attribut de classe et 339 instances. Il contient au total 22 classes de tumeur primitive. Les autres attributs indiquent les zones de départ des tumeurs primaires.
Les jeux de données sont sélectionnés avec des caractéristiques différentes : certains d'entre eux ont un nombre important d'attributs et d'autres moins, certains jeux de données ont deux classes et d'autres ont

plus de deux classes, certains ont un nombre important d'instances et d'autres ont moins, certains des attributs sont nominaux et d'autres sont numériques.
Une autre approche hybride (algorithme génétique / Adaboost (arbres de décision C4.5)) a été mise en œuvre pour la comparer à notre approche (optimisation par colonies de fourmis / Adaboost (arbre de décision C4.5)). Cette approche, disponible sous le logiciel Weka « Genetic Search », réalise la recherche en utilisant l'algorithme génétique simple décrit dans Goldberg [Goldberg 1989].
L'algorithme génétique (GA) et l'optimisation par colonie de fourmis (ACO) sont des métaheuristiques basées sur la population.
Pour chaque jeu de données, le nombre original d'attributs est utilisé. Lorsqu'on met en œuvre le processus de la sélection d'attributs (notre stratégie et l'algorithme génétique avec l'Adaboost (C4.5)), on obtient un sous-ensemble d'attributs pouvant diminuer ou augmenter le taux d'erreur de la classification supervisée. (Adaboost (C4.5)).
Les valeurs des paramètres utilisés de « Genetic Search » sont :
— La probabilité de croisement : 0.6
— La probabilité de la mutation : 0,033
— Le nombre de générations : 20
— La taille de la population : 20
Les valeurs des paramètres utilisés de l'algorithme Adaboost sont les suivantes :
— Le nombre d'itérations à effectuer : 10
— Le seuil de poids pour l'élagage : 50
La comparaison est basée sur le nombre des attributs réduits (par rapport au jeu de données original) et sur la meilleure précision du modèle d'apprentissage, c'est-à-dire un taux d'erreur moindre et la meilleure F-Mesure (par rapport à l'algorithme C4.5).
La métrique d'évaluation classique de F-mesure est utilisée pour évaluer l'efficacité de l'approche proposée, F-mesure est calculée par la précision et le rappel (la précision est la proportion des prédictions positives qui sont correctes et le rappel est la proportion des échantillons positifs correctement prédits positifs).
Les résultats des expérimentations sont donnés ci-dessous, Tableau 4.4 montre la comparaison de l'approche proposée avec la recherche génétique/Adaboost (C4.5) et l'algorithme C4.5.
Le jeu de données médicales tumeur primaire contient 17 attributs et 339 instances. Le diagnostic de la maladie est coûteux car de nombreux tests sont nécessaires pour prédire la maladie. La sélection attributs les plus importants peut être bénéfique (attributs pertinents : 07) via notre approche avec un taux d'erreur moindre (0,545) et une F-Mesure (0,38).
Le nombre d'attributs sélectionnés par l'algorithme proposé (la quatrième colonne) est inférieur au nombre total original d'attributs des jeux de données (la deuxième colonne, Tableau 4.3) dans la majorité des jeux de données.
Sur la base du taux d'erreur de classification obtenu par l'algorithme C4.5 (il est implémenté sur l'ensemble original des attributs), notre stratégie donne un taux d'erreur de classification inférieur. Par exemple, avec notre approche, le jeu de données tumeur primaire a un taux d'erreur moindre

(0,545) par rapport à la recherche génétique/Adaboost(C4.5) (0.546) et par rapport à l'algorithme C4.5 (0,602).

Jeux des données	ACO/ AdaBoost (C4.5)			Genetic Search/ adaboost(C4.5)			C4.5	
	Taux d'erreur	F-Mesure	Nombre des attributs	Taux d'erreur	F-Mesure	Nombre des attributs	Taux d'erreur	F-Mesure
Fièvre	0.001	0.99	16	0.001	0.99	16	0.016	0.984
Ecoli	0.173	0.836	6	0.167	0.836	6	0.158	0.836
Sein-w	0.038	0.944	6	0.038	0.946	9	0.054	0.946
Maladie cardiaque	0.188	0.784	8	0.171	0.814	3	0.224	0.774
Tumeur primaire	0.545	0.38	7	0.546	0.383	10	0.602	0.358
Masses mammographiques	0.166	0.765	3	0.166	0.765	3	0.178	0.822
Patient postopératoire	0.288	0.591	4	0.289	0.591	1	0.3	0.586

TABLE 4.4 – *Comparaison du taux d'erreur de la classification, du nombre d'attributs et de la F-mesure.*

Sur la base de la F-Mesure, l'approche proposée donne un résultat significatif de la classification des jeux de données Patient postopératoire, tumeur primaire, maladie cardiaque, et Fièvre par rapport à l'algorithme C4.5.

Ces premiers résultats expérimentaux indiquent que l'approche de la sélection des sous-ensembles d'attributs proposée (Optimisation par colonie de fourmis avec Adaboost(C4.5)) sélectionne les attributs pertinents dans les jeux de données médicales, ce qui entraîne une diminution du taux d'erreur, une diminution significative du nombre d'attributs et une augmentation de la F-Mesure.

4.6 Approche améliorée

Dans l'algorithme de colonie de fourmis d'origine, les valeurs de phéromone sont mises à jour lorsque toutes les fourmis ont terminé leurs chemins. Au début, la recherche du chemin optimal n'en est encore qu'à la phase d'exploration. Les fourmis peuvent suivre certains détours. Si la mise à jour des phéromones est faite à chaque phase pour les chemins non liés au chemin optimal, que les prochaines fourmis seront induites en erreur, pour éviter la convergence lente [Li *et al.* 2017], on utilise une règle améliorée de mise à jour des phéromones globales. Avec les premiers résultats expérimentaux, l'algorithme d'amélioration donnait de bons résultats en termes de la précision et de F-Mesure par rapport à la version précédente de l'algorithme [Alaoui & Elberrichi 2018].

4.6.1 Amélioration de la règle de mise à jour globale des phéromones

Les valeurs de phéromone sont mises à jour une fois que toutes les fourmis ont terminé leurs chemins (à chaque itération). Lors de la prochaine

itération, les fourmis peuvent utiliser ces valeurs dans leur recherche. À l'itération t + 1, la quantité de phéromone sur chaque lien existant dans le sous-ensemble des attributs actuel [Li *et al.* 2017] est mise à jour conformément à (equations (4.6), (4.7)) :

$$\tau_{ij} = (1 - \rho).\tau_{ij}(t) + \triangle\tau_{ij}(t) \tag{4.6}$$

$$\triangle\tau_{ij} = \begin{cases} Q/fitness, & t < A \\ Q/fitness_{opt}, & x >= A \end{cases} \tag{4.7}$$

Où :

τ_{ij} : La valeur de phéromone entre nœud i et nœud j dans l'itération (t) en cours.
ρ : Le taux d'évaporation de la phéromone.
Q : Le facteur d'intensité de phéromone.
fitness : La qualité du meilleur chemin (sous-ensemble des attributs) construit par une fourmi à itération t.
$fitness_{opt}$: La qualité du meilleur chemin (sous-ensemble des attributs) construit par toutes les fourmis à itération t.
La constante A est liée au nombre d'itérations prédéterminé.
Lorsque t <A, la valeur de phéromone est mise à jour pour tous les chemins traversés afin de localiser le meilleur chemin.
Lorsque t >= A, l'algorithme de colonie de fourmis met à jour la valeur de phéromone uniquement sur le chemin optimal à chaque itération.
La mise à jour des phéromones a pour objectif d'augmenter les valeurs de phéromones associées aux solutions prometteuses et de réduire celles associées aux mauvaises.

4.6.2 Expérimentations, Résultats et discussions

L'approche proposée est implémentée en Java comme langage de programmation en exploitant Weka (code source ouvert développé en Java, Université de Waikato, Hamilton, Nouvelle-Zélande). Les détails des jeux de données médicales sont présentés dans le tableau 4.3
Les valeurs des paramètres utilisés pour l'optimisation par colonie de fourmis, la recherche génétique et l'algorithme Adaboost sont présentées ci-dessous (voir tableau4.5).

Optimisation par Colonie de Fourmis	**Recherche Génétique**	**Algorithme Adaboost**
Nombre de fourmis : 50	Probabilité de croisement : 0.6	
α (Importance des valeurs de phéromone) : 01		Nombre d'itérations à effectuer : 10
β (Importance des valeurs heuristiques) : 01	Nombre de générations à évaluer : 20	
critère d'arrêt : 50		
Taux d'évaporation (p) : 0.1	Probabilité de mutation : 0.033	
Constante A : 25		Seuil de poids pour la taille : 50
Facteur d'intensité de phéromone (Q) : 01	Taille de la population : 20	

TABLE 4.5 – *Valeurs initiales des paramètres.*

Les résultats des expérimentations sont donnés ci-dessous (Tableau4.6 montre la comparaison de l'approche proposée avec la recherche génétique/Adaboost (C4.5) et l'algorithme C4.5). Le jeu de données médical tumeur primaire contient 17 attributs et 339 instances. Le diagnostic de la maladie est coûteux car de nombreux tests sont nécessaires pour prédire la maladie. La sélection des attributs les plus importants peut être bénéfique (attributs pertinents : 08) par notre approche avec un taux d'erreur moindre (0.540) et une F-Mesure (0.387). Le nombre d'attributs sélectionnés par l'algorithme proposé (la quatrième colonne) est nettement inférieur au nombre total d'attributs des jeux de données d'origine (la deuxième colonne, Tableau4.3) dans la majorité des jeux de données.

Jeux des données	ACO/Adaboost (C4.5) amélioré			Genetic Search / Adaboost(C4.5)			C4.5	
	Taux d'erreur	F-Mesure	Nombre des attributs	Taux d'erreur	F-Mesure	Nombre des attributs	Taux d'erreur	F-Mesure
Fièvre	0.001	0.999	16	0.001	0.99	16	0.016	0.984
Ecoli	0.155	0.842	6	0.167	0.836	6	0.158	0.836
Sein-w	0.049	0.951	5	0.038	0.946	9	0.054	0.946
Maladie cardiaque	0.178	0.819	3	0.171	0.814	3	0.224	0.774
Tumeur primaire	0.540	0.387	8	0.546	0.383	10	0.602	0.358
Masses mammographiques	0.166	0.834	3	0.166	0.765	3	0.178	0.822
Patient postopératoire	0.289	0.591	1	0.289	0.591	1	0.3	0.586

TABLE 4.6 – *Comparaison du taux d'erreur de classification, du nombre d'attributs et de la F-mesure.*

Sur la base du taux d'erreur de classification obtenu par l'algorithme C4.5 (il est implémenté sur l'ensemble de données d'origine), l'approche proposée donnait un taux d'erreur de classification inférieur (0,540) par rapport à la recherche génétique/Adaboost(C4.5) (0.546) et par rapport à l'algorithme C4.5 (0,602).
Sur la base de la mesure-F, l'approche proposée donne un résultat significatif de la classification de tous les jeux de données par rapport à l'algorithme C4.5.
Ces premiers résultats expérimentaux indiquent que l'approche de sélection des sous-ensembles d'attributs améliorée (Optimisation par colonie de fourmis avec Adaboost (C4.5)) sélectionne les attributs pertinents dans les jeux de données médicales, ce qui entraîne une diminution du taux d'erreur, une diminution significative du nombre d'attributs et une augmentation de la F-Mesure.
Les figures4.2 et 4.3 représentent les taux d'amélioration de la précision et les taux de la sélection des attributs par les deux approches (ACO/Adaboost (C4.5) amélioré, ACO/Adaboost (C4.5)).
L'étude comparative entre ACO/Adaboost (C4.5) amélioré et ACO/Adaboost (C4.5) montre que les deux hybridations dépassent les performances de C4.5 dans presque la majorité des cas. Une amélioration remarquable a été constatée dans le cas des jeux de données Fièvre, Sein-w, Maladie cardiaque, Tumeur primaire, Masses mammographiques et Patient postopératoire. Respectivement, les taux d'amélioration de la précision pour ces jeux de données étaient de 0.02%, 0.01%, 0.05%, 0.06%, 0.01% et 0.01% dans le cas de ACO/Adaboost (C4.5) amélioré par rapport à C4.5.
Les taux d'amélioration de la précision étaient de 0.02%, 0.02%, 0.04%, 0.06%, 0.01% et 0.01% dans le cas de ACO/Adaboost (C4.5) par rapport à C4.5.
Pour le jeu de données Ecoli, l'approche développée par ACO/Adaboost (C4.5) amélioré a donné une bonne précision par rapport à la méthode ACO/Adaboost (C4.5).
Sur la base de ces résultats, on observe que la classification a plus de chances d'être améliorée par les deux approches d'hybridation que par C4.5.
La figure4.3 montre les taux des attributs sélectionnés obtenus par les méthodes ACO/Adaboost (C4.5) amélioré et ACO/Adaboost (C4.5). On observe que le nombre des attributs sélectionnés diffère de moins de 85.71% du nombre total dans tous les cas et que le pourcentage des attributs sélectionnés est presque le même pour les jeux de données Fièvre, Ecoli et Masses mammographiques, il est meilleur dans le cas ACO/Adaboost (C4.5) amélioré pour les jeux de données Sein-w, Maladie cardiaque et Patient postopératoire.

4.7 Conclusion

ACO est une approche importante pour le problème de sélection des attributs et Adaboost (C4.5) est utilisée pour créer des modèles d'apprentissage robustes. Les performances de l'approche proposée ont été comparées à celles de l'algorithme hybride Genetic Search / Adaboost

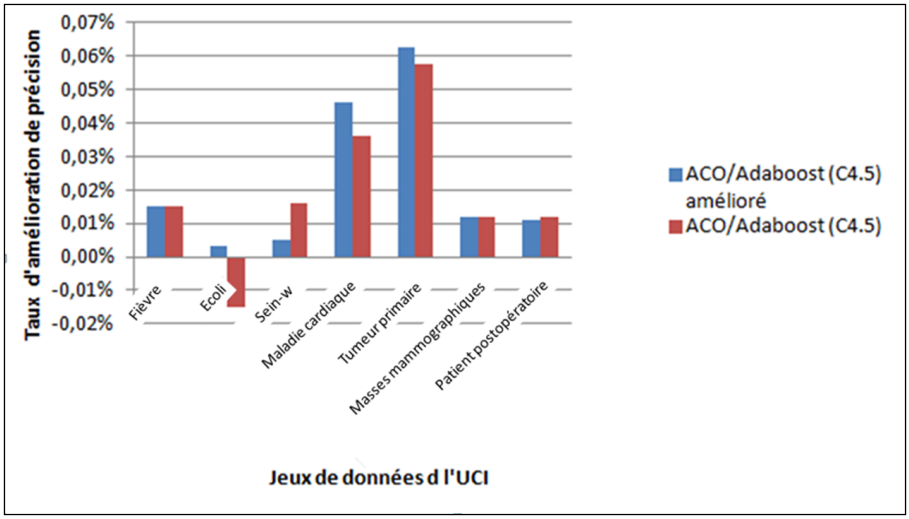

FIGURE 4.2 – *Taux d'amélioration de la précision par les deux approches.*

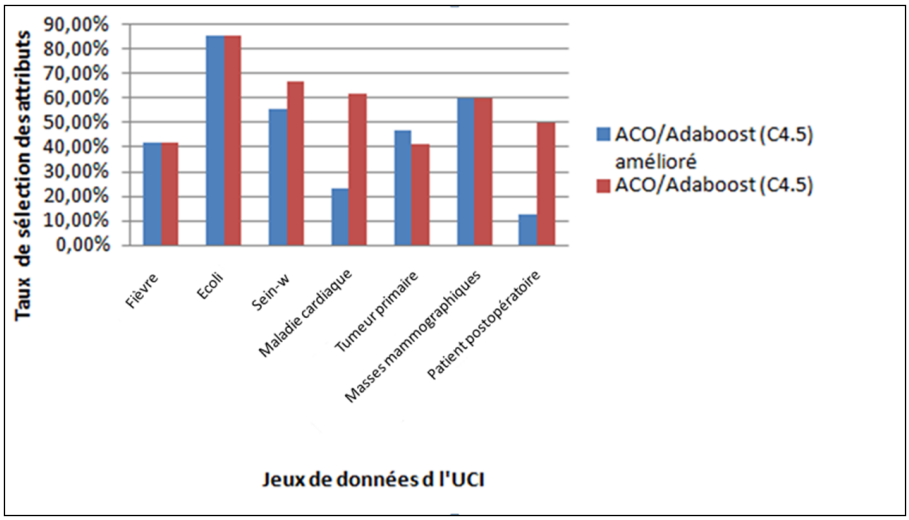

FIGURE 4.3 – *Taux de la sélection des attributs par les deux approches.*

(C4.5) et C4.5. On a conclu que l'optimisation par colonies de fourmis et l'algorithme Adaboost (C4.5) sont vraiment utiles pour la réduction de la dimensionnalité des attributs et pour la création d'un modèle rentable de prévision des maladies. Les résultats expérimentaux montrent que cette approche est compétitive. Les résultats sont prometteurs en termes de qualité de la solution et du nombre des attributs sélectionnés dans les jeux de données médicales.
En perspective, on peut appliquer cette approche sur d'autres données. Pour améliorer les solutions et éviter la convergence vers l'optimum local, la mise à jour améliorée a donné des solutions prometteuses.
On présente dans le chapitre suivant un exemple de la résolution des problèmes de la tâche de la classification supervisée.

Approches hybrides pour la classification supervisée

5

Sommaire

5.1 Introduction

Dans ce chapitre, un nouvel algorithme hybride d'apprentissage des modèles de classification est présenté. Le résultat du modèle de classification est un ensemble de règles explicite. L'approche hybride est composée de deux algorithmes. De nombreux modèles sont construits par l'algorithme de communication neuronale [Ardalan-Asl 2017] en fonction de différentes instances des jeux de données (NCA). L'apprentissage inductif basé sur l'algorithme génétique [De-Jong *et al.* 1993] est utilisé pour générer de nouvelles solutions. Cet algorithme (GABIL) remplace les solutions qui ont la plus mauvaise évaluation de la fonction de fitness par des nouvelles solutions, l'utilisation de nouvelles solutions évite de se piéger dans le maximum local et améliore les modèles résultants en termes de la précision de la classification.
Ce chapitre décrit les algorithmes NCA et GABIL utilisés et présente l'approche hybride proposée [Alaoui & Elberrichi 2020b] et les résultats empiriques avec discussions.

5.2 Motivations

Par exemple dans le domaine médical, le diagnostic identifie la maladie ou l'état qui explique les symptômes et les signes d'une maladie. Les informations nécessaires au diagnostic sont généralement obtenues à partir des antécédents et de l'examen physique du patient concerné par les soins médicaux. Les données médicales présentent un certain nombre de caractéristiques qui rendent leur classification complexe, les algorithmes d'apprentissage automatique supervisé ont été utilisés pour traiter les données étiquetées des soins de santé. L'apprentissage des données peut être obtenu par la conception et le développement des algorithmes utilisant l'apprentissage automatique avec l'intelligence compréhensible, afin d'améliorer leur comportement d'apprentissage au fil du temps et d'obtenir des connaissances à partir de l'expérience. La connaissance pertinente est cachée dans des espaces arbitraires de hautes dimensions, qui ne sont pas accessibles à l'être humain, le défi est donc de trouver cette connaissance. L'apprentissage automatique est mis en œuvre dans divers domaines, comme la santé intelligente [Holzinger 2017].
La tâche de classification des jeux de données sous la forme d'un ensemble de règles est un problème NP-Hard [Cotta & Moscato 2003]. Les métaheuristiques sont des solutions pour résoudre ce type de problèmes tels que l'algorithme de communication neuronale (NCA) et les algorithmes génétiques (GA).
L'apprentissage inductif basé sur NCA peut trouver la précision maximale globale pour la classification des données à l'aide de l'apprentissage inductif basé sur un algorithme génétique.

5.3 Apprentissage inductif basé sur un algorithme génétique GABIL

L'algorithme génétique (GA) est un algorithme d'optimisation qui repose sur des mécanismes dérivés de l'évolution naturelle et génétique : sélection, croisement, mutation. Il est apparu dans les années 60 [Holland 1962], GA fournit une méthode alternative aux problèmes de recherche adaptative qui s'est révélée être une stratégie puissante pour plusieurs problèmes. Goldberg [Goldberg 1989]a étudié l'algorithme génétique et a enrichi sa théorie en :
- Un individu est lié à un environnement par son code ADN.
- Une solution est liée à un problème par son indice de qualité.
- Une bonne solution à un problème donné peut être considérée comme un individu susceptible de survivre dans un environnement donné.
Le GA peut être utilisé dans de nombreux problèmes d'optimisation en tant qu'un élément clé de la conception de stratégies d'apprentissage robustes telles que la sélection des attributs [Jourdan *et al.* 2001]et la sélection d'instances ([Ishibuchi *et al.* 2001].[Min 2016]).
Le principal avantage des algorithmes génétiques réside dans leur capacité à exploiter l'espace de recherches à un niveau beaucoup plus détaillé que les méthodes traditionnelles basées sur des heuristiques de gradient ou de recherche et à trouver ainsi de meilleures solutions. L'algorithme génétique peut être appliqué dans une approche d'apprentissage inductif basée sur un algorithme de communication neuronale pour éviter de se piéger dans le maximum local.
L'objectif des algorithmes d'induction de règles est de découvrir un ensemble de règles explicite formulé en termes de tests de certaines valeurs des attributs, ces règles sont utilisées comme une représentation de connaissances, elles peuvent connaître correctement les instances du concept cible et les distinguer des objets qui ne lui appartiennent pas. Les règles peuvent être lues facilement par des experts humains ([Alberto *et al.* 2010]. [Furnkranz 1999]).
GABIL est un algorithme d'apprentissage inductif basé sur l'algorithme génétique. Cet algorithme a été inventé par l'auteur [De-Jong *et al.* 1993], il s'agit d'un algorithme génétique permettant de rechercher dans l'espace des jeux de règles celles qui fonctionnent bien sur un ensemble donné d'exemples positifs et négatifs. Il est basé sur les modes batch (toutes les instances sont présentées au système en même temps) et incremental (une ou quelques-unes des instances sont présentées au système à la fois).
À la première étape, on initialise aléatoirement les jeux de règles sous forme de population, puis, à l'aide de la fonction fitness (equation(5.1)), on évalue chaque jeu de règles. À l'étape suivante, les ensembles de règles sont choisis d'une manière probabiliste pour survivre en accord avec leur fonction fitness. Avec une probabilité, des opérateurs de croisement et de mutation sont appliqués aux jeux de règles choisis afin de générer une nouvelle population. On répète ces itérations jusqu'à ce que le critère d'arrêt soit vérifié (temps / espace de contraintes données).

$$Fitness(individu_i) = (pourcentage \quad correct)^2 \tag{5.1}$$

Où

La fonction fitness de chaque individu, jeu de règles, est calculée en testant le jeu de règles sur les instances de l'ensemble d'apprentissage.
L'algorithme génétique de base (Algorithme5.1) est donné par les auteurs [De-Jong *et al.* 1993]

Algorithme 5.1 Procédure GA.

```
t = 0;
Initialiser la population P (t);
Calculer fitness P(t);
while critère d'arrêt n'est pas vérifié do
    t = t + 1;
    sélectionner P(t) parmi P (t-1);
    croisement P(t);
    muter P(t);
    fitness P(t);
end
```

5.4 Travaux connexes

Cette section explore les travaux de recherche liés à l'approche hybride proposée.
L'auteur [Lavangnananda 2006] a proposé de créer un générateur autoréglable des règles associatives (SARG) pour produire des règles associatives de classification. SARG (Self-adjusting Associative Rules Generator) est une amélioration de l'approche de la programmation génétique pour l'apprentissage inductif (GPIL) [Lavangnananda 2004]. Les deux approches utilisent les algorithmes évolutionnaires dans l'apprentissage inductif. Dans GPIL, la méthode de croisement génère des nouveaux chromosomes d'une manière aléatoire avec des valeurs de la fonction fitness qui peuvent être plus élevées par hasard. La méthode de la sélection n'est pas appropriée lorsque les meilleures règles ne seront pas générées après plusieurs générations, l'inconvénient de l'approche GPIL réside dans les opérateurs de croisement et de la sélection. Ces deux opérateurs étaient incapables de s'ajuster pour choisir les méthodes adéquates à la tâche à réaliser. Mais dans SARG, les auteurs ont utilisé deux méthodes : croisement Max à Min et la sélection autoréglable. Le croisement Max à Min peut générer des meilleurs chromosomes tandis que la sélection autoréglable peut donner un nouveau chemin pour générer des nouveaux chromosomes lorsqu'il n'y a pas de développement après un certain nombre de générations consécutives.
En termes de la précision de la classification et d'efficacité de calcul, les performances de SARG sont supérieures à celles de GPIL. SARG a permis d'obtenir une plus grande précision de la classification et de trouver des meilleures solutions plus rapidement pour les données iris, heart, student avec (100%, 87.83% et 55.26%) en comparaison avec l'approche GPIL (96%, 83.11% et 53.94%) respectivement.
En perspective, les travaux futurs seront orientés vers l'utilisation de Fonction de fitness générique, d'autres méthodes pour la construction

des règles finales et la détermination de la valeur optimale de certains paramètres par exemple la taille maximale de l'arbre et le nombre des attributs dans un échantillon.
Les auteurs [Alberto *et al.* 2010] ont donné une étude exhaustive des algorithmes d'apprentissage automatique basés sur l'algorithme génétique (GBML) pour l'induction de règles, une analyse expérimentale pour la classification et une étude comparative des algorithmes GBML avec d'autres algorithmes non évolutifs sont présentées dans cette contribution. Les algorithmes évolutifs (EA) donnent des résultats compétitifs par rapport aux algorithmes classiques d'apprentissage automatique (analyse CART [Breiman *et al.* 1984], AQ [Michalksi *et al.* 1986], CN2 [Clark & Niblett 1989], C4.5 [Quinlan 1993], C4.5-Rules [Quinlan 1995], Ripper [Cohen 1995]) en termes de la précision de la classification, ces algorithmes avec leurs références sont indiqués dans le Tableau5.1[Alberto *et al.* 2010].
Dans l'approche du Michigan, l'ensemble des règles est actualisé par une observation consécutive des exemples d'entraînement avec leur classification ([Booker *et al.* 1989].[Holland 1986]). Lorsque l'algorithme évolutionnaire apprend les règles d'une manière itérative et élimine les exemples entourés par chaque nouvelle règle générée de l'ensemble d'entraînement, l'approche est un apprentissage itératif de règles (IRL) [Venturini 1993].
Si les règles/chromosomes sont améliorés dans les algorithmes évolutifs par la concurrence entre eux pour être la meilleure règle et survivre dans la base de règles, la méthode est nommée l'apprentissage coopératif génétique (GCCL) [Greene & Smith 1993]. Les règles définies dans les approches de Pittsburgh sont codées dans les chromosomes et sont évoluées jusqu'à la vérification de la convergence. Dans les arbres de décision évolutifs hybrides (HEDT), les auteurs découvrent l'utilisation d'arbres de décision avec l'algorithme génétique.
En perspective, dans le domaine des algorithmes GBML, les travaux futurs seront orientés vers l'apprentissage à partir de grands ensembles de données, l'adaptation à des ensembles de données présentant un taux de déséquilibre élevé, l'application de la complexité des données pour caractériser les performances de différents algorithmes GBML pour des ensembles de données spécifiques et l'importance de l'interprétation des modèles obtenus.

Algorithme	**Famille**	**Référence**
XCS	Michigan	[Wilson 1995]
UCS	Michigan	[Bernado & Garrell 2003]
Algorithme inductif supervisé SIA	IRL	[Venturini 1993]
Règles de décision hiérarchiques HIDER	IRL	[Aguilar-Ruiz *et al.* 2007]
Extracteur de règles co-évolutif CORE	GCCL	[Tan *et al.* 2006]
Algorithme de co-évolution organisationnel pour la classification OCEC	GCCL	[Jiao *et al.* 2006]
Induction génétique basée sur la couverture COGIN	GCCL	[Greene & Smith 1993]
Apprentissage inductif basé sur la génétique GIL	Pittsburgh	[Janikow 1993]
Algorithme d'apprentissage de règle d'intervalle génétique de Pittsburgh Pitts-GIRLA	Pittsburgh	[Corcoran & Sen 1994]
Fouille de données pour l'apprentissage évolutif DMEL	Pittsburgh	[Au *et al.* 2003]
Système du classifieur basé sur des algorithmes génétiques GASSIST	Pittsburgh	[Bacardit & andM. V. Butz 2007]
Algorithme génétique incrémental ordonné OIGA	Pittsburgh	[Zhu & Guan 2004]
Apprentissage incrémental avec des algorithmes génétiques ILGA	Pittsburgh	[Guan & Zhu 2005]
Arbre de décision - algorithme génétique hybride DT-GA	HEDT	[Carvalho & Freitas 2004]
Arbre de décision oblique Oblique-DT	HEDT	[Cantu-Paz 2003]
Analyse d'arbres avec des arbres générés et évolués de manière aléatoire TARGET	HEDT	[Gray & Fan 2008]

TABLE 5.1 – *Algorithmes d'apprentissage automatique.*

Les réseaux de neurones ont été largement appliqués dans les divers domaines, ils peuvent apprendre par induction des connaissances à partir de

données expérimentales. Dans les réseaux de neurones composites hyper-rectangulaires (HRCNN), les poids synaptiques peuvent être interprétés comme un ensemble de règles If-Then ; mais, un HRCNN formé peut produire des règles inefficaces ne pouvant justifier qu'un petit nombre d'exemples positifs, les auteurs [Hsieh *et al.* 2014] ont proposé un extracteur de règles floues basé sur l'optimisation par essaims particulaires, appelé PFHRCNN, qui a enrichi sa précédente classe de travaux de HRCNN, il est utilisé pour élaguer les règles générées par les HRCNN formés et pour améliorer les performances, les règles sont facilement interprétables par l'être humain.
L'idée principale de l'approche proposée est comme suit : les auteurs forment un HRCNN afin d'extraire un ensemble de règles précises à partir d'un ensemble de données. Après, ils implémentent l'optimisation par essaims particulaires (PSO) pour supprimer les règles inefficaces et ajuster les règles restantes sans altérer les performances de reconnaissance, ni même les améliorer. PSO se caractérise par sa capacité d'optimisation prometteuse et sa simplicité.
Cette approche a été appliquée dans le domaine médical, y compris les jeux de données des troubles hépatiques, de cancer du sein et la maladie de Parkinson.
Les résultats de la simulation et de la comparaison avec méthodes existantes ont démontré l'efficacité de l'approche proposée.

5.5 Approche proposée de l'algorithme NCGABIL

L'approche de l'apprentissage inductif basé sur l'algorithme de communication neuronale et l'algorithme génétique (NCGABIL) utilise l'algorithme NCA [Ardalan-Asl 2017] et l'algorithme GABIL [De-Jong *et al.* 1993] pour apprendre les concepts booléens effectués par un ensemble disjonctif de règles propositionnelles appliquées aux tâches de la classification, la conjonction de contraintes correspond à un ensemble stable de représentations d'attributs en chaînes de bits de règles individuelles, la longueur de la chaîne de bits dépend du nombre de règles en termes de la longueur.
Exemple [Riddle 1998] : l'hypothèse constituée des deux règles est la suivante :
▷$Si \quad a_1 = T \quad \wedge \quad a_2 = F \quad ALORS \quad c = T$;
$Si \quad a_2 = T \quad ALORS \quad c = F$
▷1001111100
L'objectif est de rechercher la précision maximale globale pour la classification des données à l'aide d'un apprentissage inductif basé sur l'algorithme génétique et la communication neuronale avec 10-fold cross-validation ;
On génère aléatoirement des neurones dans l'espace de recherches qui sont utilisés après approximation en tant que nombre de règles d'hypothèses, la valeur de la fonction de fitness (equation (5.2)) de tous les neurones est calculée par la fonction de fitness GABIL [De-Jong *et al.* 1993].

$$Fitness(h) = (correct(h))^2 \tag{5.2}$$

Où, correct (h) est le pourcentage de tous les exemples d'entraînement cor-

rectement classés par l'hypothèse h, un ensemble disjonctif de règles propositionnelles représente l'hypothèse h.
À chaque itération et pour chaque neurone, on sélectionne leurs voisins au hasard. Ensuite, le nouvel emplacement de chaque neurone est obtenu en utilisant la valeur de la fonction de fitness de ses voisins.
Pour évaluer la précision du modèle, on utilise 10-fold cross-validation. On suit ces étapes :
- Au hasard, diviser le jeu de données en 10 sous-ensembles (dix de tailles égales ou proches).
- Entraîner sur 9 sous-ensembles et tester sur 1.
- Répéter 10 fois et prendre la précision moyenne.
La fonction fitness est calculée pour chaque échantillon, puis on prend la moyenne.
L'approche proposée est expliquée ci-dessous.

Algorithme 5.2 Algorithme NCGABIL

Entrer les paramètres ;
Génération aléatoire de la population ;
while *critère d'arrêt n'est pas vérifié* **do**
    Fonction de groupe de guidage ;
    for `<i = 1 ; i<taille (population) ; i++>` **do**
        Sélection des voisins ;
        Génération de Nouveaux agents ;
        Contrôler résultat, si $f_{nouvelle} > f_{ancienne}$;
    end
    Générateur de neurones avec GABIL.
end

Pour une nouvelle itération, chaque neurone déplace d'un emplacement avec une valeur de fonction de fitness inférieure à un emplacement avec une fonction de fitness plus élevée, en fonction des informations de ses voisins de l'itération en cours.
L'algorithme NCGABIL est présenté comme suit [Ardalan-Asl 2017]. [De-Jong *et al.* 1993].

Les entrées et la génération aléatoire de la population

Paramètres d'entrée

On a défini les valeurs de différents paramètres de l'algorithme de communication neuronale et de l'apprentissage inductif basé sur les algorithmes génétiques :
Le nombre maximal (N_i) d'itérations et le nombre initial (N_{ns}) de population de l'algorithme NCA.
La taille de la population, le nombre de générations, le taux de croisement et le taux de mutation pour GABIL.
Le nombre maximum de règles de chaque individu.

Génération aléatoire de la population

Sachant que X est un vecteur de solution, chaque solution est utilisée après approximation en tant que nombre de règles, Au premier pas, les neurones étaient répartis au hasard (equation5.3).

$$\overrightarrow{X^1} = \overrightarrow{a} + (\overrightarrow{b} - \overrightarrow{a}) \times rand \tag{5.3}$$

Où, rand est un générateur de nombres aléatoires uniformes, qui donne des nombres réels compris dans l'intervalle de [0, 1].
$\overrightarrow{X^1}$ est le vecteur de localisation de neurone au début de l'algorithme.
$\overrightarrow{a}$, $\overrightarrow{b}$ sont des vecteurs incluant les limites (a, b) de l'espace de recherche $(a \leq X \leq b)$.
La fonction de fitness de chaque location est calculée et stockée en F.
Fonction de groupe de guidage
Dans cette étape, on prend un échantillon (i_m) de m neurones avec la meilleure fonction de fitness, cet échantillon s'appelle groupe de guidage et il est utilisé pour éviter le cas de non-convergence de certains neurones (i_{ma}) au maximum global.
La moyenne (Mean) et l'écart type (SD) de l'emplacement de i_m sont utilisés pour déplacer les neurones avec la fonction de fitness la plus mauvaise (i_{ma}) vers des meilleurs emplacements (equation 5.4) :

$$\begin{cases} \mu = Mean\left[\overrightarrow{X^h}(I_m)\right] \\ \sigma = SD\left[\overrightarrow{X^h}(I_m)\right] \end{cases} \rightarrow \overrightarrow{X^h}(I_{ma}) = a_1 \times (2 \times rand - 1)\sigma + \mu \quad (5.4)$$

Où, $\overrightarrow{X^h}(I_m)$ représente l'emplacement du groupe de guidage (i_m). Le coefficient a1 dans l'intervalle [1, 5] contrôle l'intervalle de distribution des neurones générés.
Sélection des voisins
On choisit les voisins du $i_{ème}$ neurone dans la population neuronale (R), à l'exception du neurone présentant le meilleur résultat (i_m), les neurones produits par le groupe de guidage (i_{ma}) et le $i_{ème}$ neurone.
La sélection aléatoire des voisins (equation5.5) tente d'éviter de se piéger dans le maximum local en modifiant les voisins et en obtenant de nouvelles informations.

$$w_j = \frac{F_{max} - F_j}{F_{max} - F_{min}}, \quad \alpha_j = \frac{w_j}{\sum_{t=1}^{N_n} w_t} \quad for\ j \in R - i, i_m, i_{ma} \quad (5.5)$$

Où le poids du $j^{ème}$ neurone est dans [0, 1],
$F_{max} et F_{min}$ sont respectivement la fonction de fitness supérieure et inférieure des neurones.
La somme du poids du $j^{ième}$ neurone et de tous ses neurones précédents donne un élément du tableau (G), le tableau peut être calculé à l'aide de (equation5.6) :

$$G_1 = \alpha_1 \quad G_j = G_{j-1} + \alpha_j \quad For \quad j = 2, 3, \cdots \in R\breve{}\, i, i_m, i_{ma} \quad (5.6)$$

Pour le $i^{ème}$ neurone, un voisin est choisi par la valeur minimale de j,
Où : $j = 1, 2, 3, \cdots \in R - i, i_m, i_{ma}$ et rand $(\leq Gj)$ est un générateur de nombres aléatoires uniformes dans [0, 1]. On répète l'exécution de la procédure Nb fois pour obtenir les voisins du i^{th} neurone (I_b),
Génération de nouveaux agents

Lorsque la valeur de la fonction de fitness de la localisation proposée est supérieure à la précédente, cette localisation sera la nouvelle localisation du $i^{ème}$ neurone.
Le nouvel emplacement proposé du $i^{ème}$ neurone est défini comme suit (equation5.7) :

$$\overrightarrow{X_i^{n+1}} = \overrightarrow{X_i^n} + rand \times \frac{\overrightarrow{T_1}}{||\vec{T_1}||} \times T_2 \tag{5.7}$$

Où,$\overrightarrow{X_i^{n+1}}$ and $\overrightarrow{X_i^n}$ sont l'emplacement proposé et l'emplacement actuel du neurone, respectivement.
Le vecteur de direction T1 et la longueur de pas T2 du mouvement du $i^{ème}$ neurone sont obtenus en utilisant les voisins du $i^{ème}$ neurone dans $n^{ième}$ et $n-1^{ième}$ itérations. Avec cette méthode, la recherche globale est progressivement convertie en recherche locale.
Ainsi, l'emplacement suggéré $\overrightarrow{X_i^{n+1}}$ ne sera le nouvel emplacement du $i^{ème}$ neurone que si la valeur de la fonction de fitness de $\overrightarrow{X_i^{n+1}}$ est supérieure à la valeur de l'emplacement actuel $\overrightarrow{X_i^n}$. La découverte du nouvel emplacement de chaque neurone doit être répétée à chaque itération.
Générateur aléatoires de neurones
Dans cette étape, pour éviter de se piéger dans le maximum local, l'algorithme GABIL [De-Jong *et al.* 1993] est appliqué pour générer de nouveaux neurones. Cet algorithme remplace les neurones avec la pire fonction de fitness (I_x) par les nouveaux neurones.
L'apprentissage inductif basé sur l'algorithme génétique utilise un algorithme génétique pour apprendre les concepts booléens illustrés par un ensemble disjonctif de règles propositionnelles. On utilise GABIL avec un opérateur de mutation standard comme l'algorithme génétique classique et une extension standard d'un croisement en deux points. Les deux points de croisement sont choisis au hasard dans la première chaîne de parent, on calcule d1(d2), la distance entre le point le plus à gauche (le plus à droite) et la limite de la règle située immédiatement à gauche, dans le deuxième parent, les points de croisement sont choisis au hasard avec les mêmes valeurs d1 et d2.
L'algorithme continue jusqu'à ce que les critères d'arrêt suivants soient vérifiés : il dépasse le nombre maximal d'itérations autorisé.

5.6 Expérimentations, résultats et discussions

Le but de cette étude est d'évaluer la puissance de l'approche proposée et de comparer ses performances à d'autres algorithmes de règles de classification.
Dans notre expérimentation, l'algorithme proposé est écrit en python comme un langage de programmation sous Linux. En outre, on met en œuvre l'algorithme GABIL et algorithmes de règles de classification (Tableau 5.2) (RIDOR, ONER, NNGE, DTNB, TABLE DE DÉCISION, RÈGLE CONJONCTIVE, PART, ZeroR et JRip) disponibles dans Weka [Group 2019]. Weka (Waikato Environment for Knowledge Analysis) est un logiciel open source. C'est une collection d'algorithmes d'apprentis-

sage automatique pour les tâches d'exploration de données, les outils de classification sont disponibles, Weka fournit de nombreux algorithmes de classification et on ne s'intéresse qu'aux méthodes de classification des règles.

Classifieur	**Fonction**	**Référence**
RIDOR	Classe pour une application de règle apprenant RIpple-DOwn	[Brian & Compton 1995]
ONER	Classe de construction et d'utilisation d'un classifieur 1R	[Holte 1993]
NNGE	Classe pour une application de l'algorithme semblable au plus proche voisin utilisant des exemples généralisés non imbriqués	[Brent 1995]
DTNB	Classe de construction et d'utilisation d'un classifieur hybride naïf de bayes et table de décision.	[Hall & Frank 2008]
DECISION TABLE	Classe de construction et d'utilisation d'un classifieur majoritaire à table de décision simple.	[Kohavi 1995]
CONJUNCTIVE RULE	Cette classe utilise un seul apprenant de règles conjonctives capable de prédire les étiquettes de classe numériques et nominales.	[Group 2019]
PART	Classe permettant de générer une liste de décisions PART. un arbre de décision partiel C4.5 dans chaque itération est en train de se construire et la "meilleure" feuille dans une règle est produite.	[Eibe & Ian 1998]
ZeroR	Classe pour la construction et la mise en oeuvre d'un classifieur 0-R.	[Group 2019]
JRip	Classe pour une application d'un apprenant à une règle propositionnelle, élagage incrémentiel répété pour produire une réduction des erreurs (RIPPER).	[Cohen 1995]

TABLE 5.2 – *Classifieurs disponibles dans Weka.*

L'approche proposée et les autres algorithmes sont expérimentés sur des ensembles de données UCI du référentiel d'apprentissage automatique pour la gestion de la classification des ensembles de données médicales [Blake & Merz 1998]. Ces jeux de données sont présentés dans le Tableau

5.3. La taille des instances varie entre 90 et 1151, le nombre d'attributs varie entre 5 et 22 et le nombre de classes varie entre 2 et 22.

Jeux de données	Nombre des attributs	Nombre des instances	Nombre des classes
Cancer du sein	9	286	2
Wisconsin- Cancer du sein	9	699	2
Colique du cheval	22	368	2
Maladie cardiaque Cleveland	13	303	2
Hépatite	19	155	2
Masses mammographiques	5	961	2
Maladie cardiaque hongrois	13	294	5
État du cœur	13	270	2
Chirurgie thoracique	16	470	2
Coeur Spect	22	267	2
Patient postopératoire	8	90	3
Tissu mammaire	9	106	2
Rétinopathie diabétique de Debrecen	19	1151	2
Maladie cardiaque va	13	200	5
Tumeur primaire	17	339	22
Lymphographie	18	148	4

TABLE 5.3 – *Jeux de données de l'UCI.*

Les valeurs initiales des paramètres utilisés de l'approche proposée et de l'algorithme d'apprentissage inductif basé sur l'algorithme génétique sont présentées dans le Tableau 5.4.

NCGABIL	**GABIL**
NCA : Taille de la Population : 100 Nombre maximum d'itérations :50 Nombre maximum de règles pour chaque individu : 50 GABIL : Taille de la Population : 100 Nombre de Générations : 100 Taux de Croisement : 0.06 Taux de Mutation : 0.01	Taille de la Population : 200 Nombre de Générations : 200 Taux de Croisement : 0.06 Taux de Mutation : 0.01

TABLE 5.4 – *Valeurs initiales des paramètres.*

Le résultat de l'exécution est décrit dans les Tableaux5.5 et Tableaux5.6 respectivement.

Jeux de Données	NCGABIL		GABIL	
	Précision	**règle**	**Précision**	**règle**
Cancer du sein	82.9713	02	74.4458	12
Wisconsin- Cancer du sein	92.7987	10	84.5528	06
Colique du cheval	87.6923	02	81.2462	04
Maladie cardiaque Cleveland	88.4697	04	88.1398	14
Hépatite	91.3099	06	90.2917	10
Masses mammographiques	99.4778	42	99.8958	11
Maladie cardiaque hongrois	97.2554	05	97.6207	12
État du cœur	86.7094	04	76.6667	08
Chirurgie thoracique	92.4835	03	82.1277	03
Coeur Spect	57.6857	09	58.8034	04
Patient postopératoire	97.1825	03	94.4444	06
Tissu mammaire	98.1187	06	98.1818	06
Rétinopathie diabétique de Debrecen	81.2598	06	75.4955	04
Maladie cardiaque va	97.9796	05	97,0000	17
Tumeur primaire	72.6626	08	63.3957	04
Lymphographie	68.2084	04	60.2857	04

TABLE 5.5 – *Précision et nombre de règles obtenus pour NCGABIL et GABIL.*

La fonction de fitness se concentre sur toutes les instances d'entraînement correctement classées par un ensemble disjonctif de règles propositionnelles en tant que facteur. Le tableau5.5 compare la précision prédictive mesurée sur les 16 jeux de données, pour les modèles générés par l'approche proposée par rapport à ceux générés par les algorithmes de base, l'algorithme de l'apprentissage inductif basé sur l'algorithme génétique. Les paramètres utilisés sont présentés dans le tableau5.4

Le tableau5.5 représente les résultats obtenus, en termes de la précision et du nombre des règles. L'étude comparative entre NCGABIL et GABIL montre que les deux approches sont plus performantes dans la majorité des cas. Le Tableau5.5 montre que la précision prédictive globale de la méthode proposée est supérieure à celle de l'algorithme de base dans la majorité des cas. Une amélioration remarquable concerne les jeux de données cancer du sein, wisconsin-cancer du sein, colique du cheval, état du cœur, chirurgie thoracique, rétinopathie diabétique de Debrecen, tumeur primaire et lymphographie. À la base de ces résultats, on observe que l'apprentissage par induction a plus de chances d'être amélioré par l'approche par hybridation. Le tableau5.5 présente l'ensemble des résultats obtenus par les deux méthodes. On observe que le nombre de l'ensemble de règles obtenu par NCGABIL est presque inférieur à celui de GABIL.

Les tableaux5.5 et 5.6 montrent que l'algorithme proposé est en concurrence avec les algorithmes de classification de base en termes de la précision de la classification, car il a obtenu la meilleure précision globale. Le tableau5.6 représente les résultats obtenus, en termes de la précision. L'étude comparative entre NCGABIL, GABIL, RIDOR, ONER, NNGE, DTNB, TABLE DE DÉCISION, RÈGLE CONJONCTIVE et PART a montré que l'approche proposée surperforme l'apprentissage inductif dans presque tous les cas. Une amélioration remarquable dans les cas des ensembles de données : cancer du sein, colique du cheval, maladie cardiaque cleveland , hépatite, État du cœur, chirurgie thoracique, Patient

postopératoire, rétinopathie diabétique de debrecen, maladie cardiaque va, et tumeur primaire.
Cette étude montre le potentiel de l'approche proposée. Plus précisément, il peut améliorer les modèles de classification existants en termes de la précision prédictive.

5.7 Conclusion

Dans ce chapitre, on a présenté l'apprentissage inductif basé sur l'algorithme génétique et l'algorithme de communication neuronale. L'algorithme NCGABIL peut trouver la précision maximale globale pour la classification des données à l'aide d'un apprentissage inductif basé sur la communication neuronale et l'algorithme génétique avec 10-fold cross-validation pour éviter de se piéger dans un maximum local. Cet algorithme hybride exécute un modèle de classification qui peut être utile pour le diagnostic et le pronostic. Il est utilisé pour obtenir le meilleur ensemble de règles permettant d'optimiser le classifieur. La performance de l'approche proposée est comparée à plusieurs approches disponibles dans Weka. Ces approches ont été évaluées d'une manière expérimentale sur 16 jeux de données bien connus, découvrant des résultats fiables en identifiant des règles pertinentes.
Il a été conclu que l'approche proposée comparée avec des approches existantes est généralement plus performante et puissante pour différents jeux de données. Les résultats obtenus à l'aide d'algorithmes hybrides montrent que la méthode proposée est capable de trouver un ensemble de règles approprié et d'obtenir de meilleurs résultats.
En perspective, on peut appliquer cette approche sur d'autres données. Pour éviter la convergence vers l'optimum local, les paramètres utilisés peuvent être optimisés.

Jeux de Données	NCGA-BIL	GABIL	ZeroR	RIDOR	part	ONER	NNGE	JRip	DTNB	DECI-SION TABLE	CONJU-CTIVE RULE
Cancer du sein	**82.9713**	74.4458	70.2797	70.979	71.3287	65.7343	65.035	70.979	71.6783	73.4266	66.4336
Wisconsin-Cancer du sein	92.7987	84.5528	65.5222	95.8512	93.8484	92.7039	95.9943	95.422	**96.8526**	95.279	91.7024
Colique du cheval	**87.6923**	81.2462	63.0435	83.6957	84.7826	81.5217	80.4348	84.2391	83.4239	82.8804	81.5217
Maladie cardiaque Cleveland	**88.4697**	88.1398	54.4554	79.538	79.868	71.6172	80.8581	81.5182	79.868	76.2376	71.2871
Hépatite	91.3099	90.2917	79.3548	78.7097	84.5161	83.2258	84.5161	78.0645	81.2903	76.129	78.7097
Masses mammographiques	99.4778	**99.8958**	53.6941	81.7898	82.102	81.9979	77.2112	82.7263	82.3101	81.9979	81.2695
Maladie cardiaque hongrois	97.2554	**97.6207**	63.9456	80.9524	80.9524	78.5714	79.5918	78.9116	82.6531	80.6122	77.8912
État du cœur	**86.7094**	76.6667	55.5556	78.1481	73.3333	71.1111	78.1481	78.8889	81.4815	84.8148	74.0741
Chirurgie thoracique	**92.4835**	82.1277	85.1064	84.8936	79.1489	83.617	81.0638	84.6809	81.7021	84.4681	85.1064
Coeur Spect	57.6857	58.8034	58.8015	70.0375	64.4195	**72.6592**	68.9139	72.2846	71.5356	71.9101	72.2846
Patient postopératoire	**97.1825**	94.4444	71.1111	71.1111	61.1111	67.7778	57.7778	71.1111	56.6667	68.8889	71.1111
Tissu mammaire	98.1187	**98.1818**	80.1887	95.283	94.3396	93.3962	91.5094	94.3396	84.9057	96.2264	90.566
Rétinopathie diabétique de Debrecen	**81.2598**	75.4955	53.0843	63.4231	64.987	53.258	62.033	62.9018	62.3805	64.8132	57.7758
Maladie cardiaque va	**97.9796**	97,0000	28	32	33	26,5	35	30	26	28,5	29
Tumeur primaire	**72.6626**	63.3957	24.7788	37.1681	40.708	27.4336	40.708	39.233	41.5929	39.823	28.9086
Lymphographie	68.2084	60.2857	54.7297	**85.1351**	76.3514	75	78.3784	77.7027	73.6486	77.027	75

TABLE 5.6 – *Précision de l'approche proposée et des algorithmes de règles de classification Disponible dans Weka.*

Mes contributions scientifiques

— Alaoui, A., and Elberrichi, Z. (2018), Feature Subset Selection Using Ant Colony Optimization for a Decision Trees Classification of Medical Data. International Journal of Information Retrieval Research, 8(4).
— Alaoui, A., and Elberrichi, Z. (2020), Enhanced Ant Colony Algorithm for Best Features Selection for a Decision Tree Classification of Medical Data. In Sarfraz, M., ed. : Critical Approaches to Information Retrieval Research, IGI Global, 278-293
— Alaoui, A., and Elberrichi, Z. (2020), Neuronal Communication Genetic Algorithm Based Inductive Learning. Journal of Information Technology Research (JITR), 13(2).

Conclusion générale et perspectives

L'hybridation des Métaheuristiques dans le domaine d'extraction de connaissances à partir de données permet le développement et la découverte des modèles intelligibles, compétents et efficaces. L'idée principale consiste à profiter des caractéristiques bénéfiques des algorithmes participants à l'hybridation pour former une nouvelle approche efficiente, et à essayer d'éviter les inconvénients présents des algorithmes combinés.
L'objectif de cette thèse est d'obtenir des modèles compétents pour améliorer la tâche de la classification supervisée, tout en hybridant les métaheuristiques et en étudiant le prétraitement de données dans le processus d'extraction des connaissances à partir de données. L'hybridation des métaheuristiques est une méthode efficiente pour trouver des solutions à des problèmes d'optimisation réputés difficiles et principalement dans le processus d'extraction de connaissance à partir de données.
Durant ce travail doctoral, des algorithmes ont été utilisés et développés pour découvrir des connaissances. Les algorithmes coopératifs hybrides peuvent résoudre les problèmes d'optimisation dans le domaine de la classification supervisée et du prétraitement des données en améliorant significativement les résultats obtenus.
La première partie de ce document a présenté généralement le contexte et l'état de l'art de notre travail sur les métaheuristiques et leurs implémentations par hybridation dans le processus d'extraction des connaissances à partir de données. Le premier chapitre est consacré aux différents éléments de base qu'il nous faut savoir sur les architectures des métaheuristiques et les techniques d'hybridation des métaheuristiques. Tandis que le deuxième chapitre présente les notions de base des processus d'extraction des connaissances à partir de données. On a donné des définitions et des descriptions qu'il nous faut savoir sur le prétraitement de données, la fouille de données, la régression, la classification supervisée, la recherche des règles d'association et la segmentation. On a cité dans le troisième chapitre les critères principaux de la classification des métaheuristiques hybrides et une méthodologie de conception des méthodes hybrides par les métaheuristiques pour l'extraction de connaissances à partir de données, on a donné dans ce chemin des exemples sur le développement des tâches de la fouille de données tout en appliquant le processus de prétraitement de données.
La deuxième partie de cette thèse a présenté les contributions réalisées. Une première contribution a été proposée. Il s'agit d'un algorithme hybride d'optimisation par colonies de fourmis et l'algorithme Adaboost (C4.5), il est appliqué aux données médicales pour le diagnostic comme un exemple du domaine utilisé. Dans cet algorithme on sélectionne les

attributs pertinents non redondants qui sont importants pour la prédiction et pour la construction d'un modèle d'apprentissage efficace. Et on a donné une version améliorée de la première approche proposée qui est une version compétente. Dans le dernier chapitre, une troisième contribution a été proposée. Il s'agit d'un algorithme hybride d'apprentissage puissant basé sur des règles pour extraire des connaissances de haut niveau à partir d'ensembles de données pour la classification des données. Cette approche hybride emploie l'apprentissage inductif basé sur la communication neurale et l'algorithme génétique.
Nos contributions nous donnent des résultats compétents en comparant à d'autres approches disponibles dans la littérature, nous motivent et nous permettent d'ouvrir plusieurs perspectives de continuer vers l'implémentation des métaheuristiques hybrides, distribuées et parallèles pour la classification supervisée et le prétraitement des données. Et par la suite, les autres problèmes d'optimisation dans les différents domaines peuvent être étudiés, tels que les différents éléments du prétraitement de données et de la réduction des données comme la sélection des instances. Cela peut se réaliser en employant les approches développées dans cette thèse, ou bien en proposant de nouvelles métaheuristiques. D'un côté, ces résultats nous permettent de proposer d'autres métaheuristiques hybrides. D'un autre côté, la combinaison de métaheuristiques fournit généralement des algorithmes de recherche de haute performance pour de nombreuses classes de problèmes, le résultat de l'hybridation nous permet de proposer des approches de résolution à des problèmes de la segmentation, de la recherche des règles d'association et d'investiguer vers les solutions du problème de la régression.

Bibliographie

[Aghdam *et al.* 2009] M.-H. Aghdam, N.-G. Aghaee et M.-E. Basiri. *Test Feature Selection using Ant Colony Optimization.* Expert Systems with Applications, vol. 36, pages 6843–6853, 2009.

[Agrawal & Srikant 1994] R. Agrawal et R. Srikant. *Fast algorithms for mining association rules in large databases*. Proceedings of the 20th International Conference on Very Large Data Bases, pages 478–499, 1994.

[Agrawal *et al.* 1993] R. Agrawal, T. Imielinski et A. Swami. *Mining association rules between sets of items in large databases*. In Proceedings of the ACM SIGMOD International Conference on Management of Data, pages 207–216. ACM Press, 1993.

[Aguilar-Ruiz *et al.* 2007] J.-S. Aguilar-Ruiz, R. Giraldez et J.-C. Riquelme. *Natural encoding for evolutionary supervised learning*. IEEE Transactions on Evolutionary Computation, vol. 11, no. 4, page 466–479, 2007.

[Aha *et al.* 1991] D.-W. Aha, D. Kibler et M.-K. Albert. *Instance-based learning algorithms*. Machine Learning, vol. 6, no. 1, page 37–66, 1991.

[AlAni 2005] A. AlAni. *Ant Colony Optimization for feature subset selection*. In Proceedings of the World Academy Of Science, Engineering And Technology, volume 4, pages 35–38. PWASET, 2005.

[Alaoui & Elberrichi 2018] A. Alaoui et Z. Elberrichi. *Feature Subset Selection Using Ant Colony Optimization for a Decision Trees Classification of Medical Data*. International Journal of Information Retrieval Research, vol. 8, no. 4, pages 39–50, 2018.

[Alaoui & Elberrichi 2020a] A. Alaoui et Z. Elberrichi. *Enhanced Ant Colony Algorithm for Best Features Selection for a Decision Tree Classification of Medical Data.* In Sarfraz, M., ed. : Critical Approaches to Information Retrieval Research, IGI Global, pages 278–293, 2020.

[Alaoui & Elberrichi 2020b] A. Alaoui et Z. Elberrichi. *Neuronal Communication Genetic Algorithm Based Inductive Learning.* Journal of Information Technology Research (JITR), vol. 13, no. 2, pages 141–154, 2020.

[Alba & Luque 2005] E. Alba et G.-T. Luque. *Measuring the Performance of Parallel Metaheuristics.* In Parallel Metaheuristics : A New Class of Algorithms, pages 43–62. Wiley, 2005.

[Alba & Nebro 2005] E. Alba et A.-J. Nebro. *New Technologies in Parallelism*. Parallel Metaheuristics, pages 63–78, 2005.

[Alberto *et al.* 2010] F. Alberto, S. Garcia, J. Luengo, E. Bernado mansilla et F. Herrera. *Genetics based machine learning for rule induction : state of the art, taxonomy, and comparative study*. IEEE Transactions on Evolutionary Computation, vol. 6, no. 14, page 913–941, 2010.

[Alboaneen *et al.* 2016] A.-D. Alboaneen, T. Huaglory et Z. Yan. *Glowworm swarm optimisation algorithm for virtual machine placement in cloud computing*. Intl IEEE, pages 808–814, 2016.

[Alliot *et al.* 2012] J.-M. Alliot, N. Durand, D. Gianazza et J.-B. Gotteland. *Finding and Proving the Optimum : Cooperative Stochastic and Deterministic Search*. 20th European Conference on Artificial Intelligence, pages 55–60, 2012.

[Almuallim & Dietterich 1991] H. Almuallim et T.-G. Dietterich. *Learning with many irrelevant features*. In Proceedings of the Ninth National Conference on Artificial Intelligence, page 547–552. AAAI Press, 1991.

[AlMuhaideb & Menai 2013] S. AlMuhaideb et M.-B. Menai. *Hybrid Metaheuristics for Medical Data Classification*. International Journal of Metaheuristics, vol. 3, no. 1, pages 59–80, 2013.

[AlMuhaideb & Menai 2014] S. AlMuhaideb et M.-B. Menai. *HColonies : a new hybrid metaheuristic for medical data Classification*. Springer Sciences Business Media New York, page 282–298, 2014.

[Ani 2008] A. Ani. *Feature Subset Selection using Ant Colony Optimization.* International Journal of Computational Intelligence, vol. 2, pages 53–58, 2008.

[Applegate *et al.* 2007] D. Applegate, W. Cook, S. Dash et D. Espinoza. *New Technologies in Parallelism*. Operations Research Letters, vol. 6, no. 35, page 693–699, 2007.

[Ardalan-Asl 2017] S. Asil-Gharebaghiand M. Ardalan-Asl. *International journal of optimization in civil engineering*. New metaheuristic optimization algorithm using neuronal communication, vol. 7, no. 3, pages 413–431, 2017.

[Au *et al.* 2003] W.-H. Au, K.-C. Chan et X. Yao. *A novel evolutionary datamining algorithm with applications to churn prediction*. IEEE Transactions on Evolutionary Computation, vol. 7, no. 6, page 532–545, 2003.

[Augerat *et al.* 1998] P. Augerat, J.-M, Belenguer, E. Benavent, A.Corber et D. Naddef. *Separating capacity constraints in the CVRP using tabu search*. European Journal of Operational Research, vol. 106, no. 2-3, page 546 –57, 1998.

[Bacardit & andM. V. Butz 2007] J. Bacardit et D.-E. Goldberg andM. V. Butz. *Improving the performance of a Pittsburgh learning classifier system using a default rule*. International Workshop on Learning Classifier Systems, page 291–307, 2007.

[Balasundaram & Sathiya 2019] R. Balasundaram et S.-D. Sathiya. *Genetic algorithm-based hybrid approach for optimal instance selection of minimising makespan in permutation flowshop scheduling*. International Journal of Business Intelligence and Systems Engineering, vol. 1, no. 3, 2019.

[Baluja 1994] S. Baluja. *Population-based incremental learning : A method for integrating genetic search based function optimization and competitive learning*. Technical Report CMU-CS-94-163 Carnegie Mellon University Pittsburgh PA, 1994.

[Barr & Hickman 1993] R.S. Barr et B.-L. Hickman. *Reporting Computational Experiments with Parallel Algorithms : Issues, Measures and Experts Opinions*. ORSA Journal on Computing, vol. 5, no. 1, pages 2–18, 1993.

[Bayes 1764] T. Bayes. *An Essay towards solving a Problem in the Doctrine of Chances*. Philosophical Transactions of the Royal Society of London, pages 370–418, 1764.

[Bent & Van 2004] R. Bent et P.-H. Van. *A two stage hybrid local search for the vehicle routing problem with time windows*. TRANSPORTATION SCIENCE, vol. 38, no. 4, pages 515–530, 2004.

[Berkhin 2002] P. Berkhin. *Survey of Clustering Data Mining Techniques*. Technical report, Accrue software, San Jose,California, pages 1–56, 2002.

[Bernado & Garrell 2003] M.-E. Bernado et J.-M. Garrell. *Accuracy-based learning classifier systems : Models, analysis and applications to classification tasks*. Evolutionary Computation, vol. 11, no. 3, page 209–238, 2003.

[Berry & Linoff 1997] J.-A.-M. Berry et G. Linoff. *Data Mining :Techniques appliquées au marketing, à la vente et aux services clients*. 1997.

[Besse 2005] P. Besse. *Data mining II Modelisation Statistique et Apprentissage.* Laboratoire de Statistique et Probabilites — UMR CNRS C5583 Universite Paul Sabatier — 31062 – Toulouse cedex 4, pages 1–115, 2005.

[Bezdek & Kuncheva 2001] J.-C Bezdek et L.-I. Kuncheva. *Nearest prototype classifier designs : An experimental study*. International Journal of Intelligent Systems, vol. 16, no. 12, page 1445–1473, 2001.

[Bing *et al.* 2014] X. Bing, Z. Mengjie et B. Will. *Particle swarm optimization for feature selection in classification : Novel initialization and updating mechanisms*. Applied Soft Computing, vol. 18, page 261 – 276, 2014.

[Blake & Merz 1998] C. Blake et C. Merz. *UCI Repository of machine learning DataBases*. University of California, Irvine, 1998.

[Blum & Langley 1997] A. Blum et P. Langley. *Selection of relevant features and examples in machine learning.* Artificial Intelligence, vol. 97, page 245–271, 1997.

[Blum & Rivest 1992] A.-L. Blum et R.-L Rivest. *Training a 3-node neural networks is NP-complete*. Neural Networks, vol. 5, no. 1, page 117–127, 1992.

[Blum & Roli 2003] C. Blum et A. Roli. *Metaheuristics in Combinatorial Optimization : Overview and Conceptual Comparison*. ACM Computing Survey, vol. 35, no. 3, pages 268–308, 2003.

[Blum *et al.* 2005] C. Blum, A. Roli et E. Alba. *An introduction to metaheuristic techniques. In : Parallel Metaheuristics, a New Class of Algorithms*. John Wiley, page 3–42, 2005.

[Blum *et al.* 2010] C. Blum, J. Puchinger, G. Raidl et A. Roli. *a brief survey on hybrid metaheuristics*. In Bioinspired optimization methods and their applications :proceedings of the Fourth International Conference on Bioinspired Optimization Methods and their Applications - BIOMA, pages 3–18. Slovenian Research Agency, 2010.

[Blum *et al.* 2011] C. Blum, J. Puchinger, G. Raidl et A. Roli. *Hybrid metaheuristics in combinatorial optimization : A survey*. Applied Soft Computing, vol. 11, no. 6, pages 4135–4151, 2011.

[Booker *et al.* 1989] L.-B. Booker, D.-E. Goldberg et J.-H. Holland. *Classifier systems and genetic algorithms, Artificial. Intelligence*. Artificial intelligence, vol. 40, no. 1-3, page 235–282, 1989.

[Boucheham *et al.* 2015] A. Boucheham, M. Batouche et S. Meshoul. *An Ensemble of Cooperative Parallel Metaheuristics for Gene Selection in Cancer Classification*. International Conference on Bioinformatics and Biomedical Engineering, pages 301–312, 2015.

[Brachman & Anand 1996] R. Brachman et T. Anand. *The Process of Knowledge Discovery in Databases : A Human Centered Approach*. In Advances in Knowledge Discovery and Data Mining, pages 37–58. AAAI/MIT Press, 1996.

[Brassard & Bratley 1996] G. Brassard et P. Bratley. *Fundamentals of algorithmics*. Prentice Hall, New Jersey, 1996.

[Breiman *et al.* 1984] L. Breiman, L. Friedman, J. Olshen et C.Stone. *Classification and regression trees.* CA : Wadsworth International Group, 1984.

[Breiman 1996] L. Breiman. *Bagging predictors*. Machine Learning, vol. 24, page 123–140, 1996.

[Brent 1995] M. Brent. *Instance-Based learning : Nearest Neighbor with Generalization*. Hamilton, New Zealand, 1995.

[Brezocnik *et al.* 2018] L. Brezocnik, I. Fister et V. Podgorelec. *Swarm Intelligence Algorithms for Feature Selection : A Review*. Applied Sciences Journal, vol. 8, no. 9, 2018.

[Brian & Compton 1995] R.-G. Brian et P. Compton. *Induction of Ripple-Down Rules Applied to Modeling Large Databases*. Journal Intelligence Information System, vol. 5, no. 3, page 211–228, 1995.

[Calégari 1999] PR Calégari. *Parallelization of population-based evolutionary algorithms for combinatorial optimization problems*. PhD thesis, Lausanne, 1999.

[Candillier 2004] L. Candillier. *la classification non supervisée*. équipe GRAPPA, 2004.

[Cang *et al.* 2002] Y. Cang, J. Yang et D.-Y. Geng. *Fuzzy rules to predict degree of malignancy in brain glioma*. Med Biol Eng Comput, vol. 40, no. 2, pages 145–152, 2002.

[Cantú-Paz 2000] E. Cantú-Paz. *Efficient and Accurate Parallel Genetic Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 2000.

[Cantu-Paz 2003] E. Cantu-Paz. *Inducing oblique decision trees with evolutionary algorithms*. IEEE Transactions on Computer, vol. 7, no. 1, page 54–68, 2003.

[Carvalho & Freitas 2004] D.-R. Carvalho et A.-A. Freitas. *A hybrid decision tree/genetic algorithm method for data mining*. Information Sciences, page 13–35, 2004.

[Chabrier *et al.* 2002] A. Chabrier, E. Danna et C. L. Pape. *Coopération entre génération de colonnes sans cycle et recherche locale appliquée au routage de véhicules*. In JNPC, 2002.

[Chakrapani & Skorin-Kapov 1993a] J. Chakrapani et J. Skorin-Kapov. *Connection Machine Implementation of a Tabu Search Algorithm for the Traveling Salesman Problem*. Journal of Computing and Information Technology, vol. 1, no. 1, pages 29–36, 1993.

[Chakrapani & Skorin-Kapov 1993b] J. Chakrapani et J. Skorin-Kapov. *Massively Parallel Tabu Search for the Quadratic Assignment Problem*. Annals of Operations Research, vol. 41, no. 1, pages 327–341, 1993.

[Chawla *et al.* 2004] N.-V. Chawla, N. Japkowicz et A. Kotcz. *Editorial : Special issue on learning from imbalanced data sets*. ACM Special Interest Group on Knowledge Discovery and Data Mining . SIGKDD Explorations, vol. 6, no. 1, page 1–6, 2004.

[Chen & Liu 1999] K. Chen et H. Liu. *Towards an evolutionary algorithm : A comparison of two feature selection algorithms*. In 1999 Congress on Evolutionary Computation, page 1309–1313, 1999.

[Chen & Ludwig 2018] M. Chen et S.-A. Ludwig. *Discrete Particle Swarm Optimization with local search strategy for Rule Classification*. 2012 Fourth World Congress on Nature and Biologically Inspired Computing (NaBIC), page 162–167, 2018.

[Chien-Hsing *et al.* 2006] C. Chien-Hsing, K. Bo-Han et C. Fu. *The generalized condensed nearest neighbor rule as a data reduction method*. 18th international conference on pattern recognition. IEEE Computer Society Hong-Kong, page 556–559, 2006.

[Clark & Niblett 1989] P. Clark et T. Niblett. *The CN2 induction algorithm*. Machine Learning, vol. 3, no. 4, page 261–283, 1989.

[Clerc & Siarry 2004] M. Clerc et P. Siarry. *Une nouvelle métaheuristique pour l'optimisation difficile : la méthode des essaims particulaires Fast Effective Rule Induction*. Twelfth International Conference on Machine Learning, vol. 3, no. 7, 2004.

[Cohen 1995] W. Cohen. *Fast effective rule induction*. 12th International Conference Machine Learning, page 115–123, 1995.

[Collette & Siarry 2002] Y. Collette et P. Siarry. *Optimisation multiobjectif*. Eyrolles edition, 2002.

[Colorni *et al.* 1992] A. Colorni, M. Dorigo et V. Maniezzo. *Distributed optimization by ants colonies.* European Conf. ECAL'91, pages 134–142, 1992.

[Corcoran & Sen 1994] A.-L. Corcoran et S. Sen. *Using Real-Valued Genetic Algorithms to Evolve Sets for Classification*. EEE conference Evolutionary Computation, pages 120–124, 1994.

[Cortes & vapnik 1995] C. Cortes et V. vapnik. *Support-Vector Networks*. Kluwer Academic Publishers, Boston. Manufactured in The Netherlands, vol. 20, page 273–297, 1995.

[Cotta & Moscato 2003] C. Cotta et P. Moscato. *The k-feature set problem is w[2]-complete*. Journal of computer and system science, pages 686–690, 2003.

[Cotta & Troya 2003] C. Cotta et J.-M. Troya. *Embedding branch and Bound Evolutionary Algorithms*. Applied Intelligence, pages 137–153, 2003.

[Cotta *et al.* 1995] C. Cotta, J.-F. Aldana, A.-J. Nebro et J.-M. Troya. *Hybridizing genetic algorithms with branch and bound techniques for the resolution of the tsp*. Artificial Neural Nets and Genetic Algorithms Pearson D.W., Steele N.C., Albrecht R.F. (eds.) Springer Verlag Wien New York, pages 277–280, 1995.

[Cotta *et al.* 2005] C. Cotta, E.-G. Talbi et E. Alba. *Parallel hybrid metaheuristics, a New Class of Algorithms*. John Wiley, page 347–370, 2005.

[Cotta 1998] C. Cotta. *A study of hybridisation techniques and their application to the design of evolutionary algorithms*. AI Communications, vol. 11, no. 3-4, page 223–224, 1998.

[Crainic *et al.* 1997] T.-G. Crainic, A.-T Nguyen et M. Gendreau. *Cooperative multi-thread parallel tabu search with evolutionary adaptive memory*. In : 2nd Int. Conf. on Metaheuristics, Sophia Antipolis, France, 1997.

[Crainic 2005] T.-G. Crainic. *Parallel meta-heuristic search*. Tech Rep CIRRELT, 2005.

[Cung *et al.* 2002] V.-D. Cung, S.-L. Martins, C.-C. Ribeiro et C. Roucairol. *Strategies for the parallel implementation of metaheuristics*. Dans Essays and Surveys in Metaheuristics, Kluwer Academic Publishers, page 263–308, 2002.

[Dalboni *et al.* 2003] F.-L. Dalboni, L.-S. Ochi et L.-M.-A. Drummond. *On improving evolutionary algorithms by using data mining for the oil collector vehicle routing problem*. International Network Optimization Conference, 2003.

[Darwish *et al.* 2018] A. Darwish, G.-I. Sayed et E. Aboul-Hassanien. *MetaHeuristic Optimization Algorithms Based Feature Selection For Clinical Breast Cancer Diagnosis*. Journal of the Egyptian Mathematical Society, vol. 26, no. 3, pages 321–336, 2018.

[Das 2001] S. Das. *Filters, wrappers and a boosting-based hybrid for feature selection*. In Proceedings of the Eighteenth International Conference on Machine Learning, page 74–81. Morgan Kaufmann, San Francisco, CA, 2001.

[Dash & Liu 1997] M. Dash et H. Liu. *Feature selection for classification*. Intelligent Data Analysis, vol. 1, no. 1-4, page 131–156, 1997.

[Dash & Liu 1998] M. Dash et H. Liu. *Hybrid search of feature subsets*. In Proceedings of the Fifth Pacific Rim International Conference on AI : PRICAI'98 Singapore, volume 1531, pages 238–249. Springer, 1998.

[De-Jong *et al.* 1993] K.-A. De-Jong, W.-M. Spears et F.-D. Gordon. *Using genetic algorithms for concept learning*. Machine Learning, vol. 13, pages 161–188, 1993.

[de Souza *et al.* 2010] J. Teixeira de Souza, R. Augusto Ferreira do Carmo et G. Augusto Campos de Lima. *On The Combination of Feature and Instance Selection*. Machine Learning InTech, pages 157–170, 2010.

[Dempster *et al.* 1977] P. Dempster, N.-M. Laird et D.-B. Rubin. *Maximum Likelihood from Incomplete Data via the EM Algorithm Journal of the Royal Statistical Society*. Series B (Methodological), vol. 39, no. 1, pages 1–38, 1977.

[Derrac *et al.* 2010] J. Derrac, S. García et F. Herrera. *A survey on evolutionary instance selection and Generation*. International Journal of Applied Metaheuristic Computing, vol. 1, no. 1, pages 60–92, 2010.

[Devijver & Kittler 1980] P. A. Devijver et J. Kittler. *On the edited nearest neighbor rule*. In Proceedings of the 5th international conference on pattern recognition. Los Alamitos, CA, pages 72–80. IEEE Computer Society Press, 1980.

[Doak 1992] J. Doak. *An evaluation of feature selection methods and their application to computer security*. Technical report, Davis CA : University of California, Department of Computer Science, 1992.

[DOCTISSIMO 2019] DOCTISSIMO. *https ://www.doctissimo.fr/html/dossiers/alzheimer/ 11723-transmission-neuronale.htm*. 2019.

[Dorigo 2007] M. Dorigo. *Ant colony optimization*. Retrieved from http ://www.scholarpedia.org/article/Ant_colony_optimization, 2007.

[Duan *et al.* 2013] Q. Duan, T.W. Liao et H.-Z. Yi. *A comparative study of different local search application strategies in hybrid metaheuristics*. Applied Soft Computing, vol. 13, no. 3, pages 1464–1477, 2013.

[Eibe & Ian 1998] F. Eibe et H. Ian. *Generating Accurate Rule Sets Without Global Optimization*. Fifteenth International Conference on Machine Learning, page 144–151, 1998.

[El-Houby *et al.* 2017] E.-M.-F. El-Houby, N. Yassin et S. Omran. *Hybrid Approach From Ant Colony Optimization And K-Nearest Neighbor For Classifying Datasets Using Selected Features*. Informatica, vol. 41, no. 4, pages 495–506, 2017.

[Elbenani *et al.* 2012] B. Elbenani, A. Jacques et Ferland J. Bellemare. *Genetic algorithm and large neighbourhood search to solve the cell formation problem*. Expert Systems with Applications, vol. 39, no. 3, page 2408–2414, 2012.

[Falkenauer 1994] E. Falkenauer. *new representation and operators for genetic algorithms applied to grouping problems*. Evolutionary Computation, vol. 2, no. 2, page 123–144, 1994.

[Fayyad *et al.* 1996] U. Fayyad, S. Piatetsky, Gregory et S. Padhraic. *The KDD process for extracting useful knowledge from volumes of data*. ACM New York, NY, USA, vol. 39, no. 11, pages 37–14, 1996.

[Fayyad 1998] U. Fayyad. *Data Mining and Knowledge Discovery*. ed. Kluwer Academic Publishers, vol. 2, no. 1, pages 5–7, 1998.

[Feltl & RaidI 2004] H. Feltl et G.-R. RaidI. *An Improved Hybrid Genetic Algorithm for the Generalized Assignment Problem*. In Proceedings of the ACM Symposium on Applied Computing, Nicosia, Cyprus, pages 990–995. ACM Press, 2004.

[Feng *et al.* 2007] T. Feng, F. Xuezheng, Z. Yanqing et G. Bourgeois. *A genetic algorithm-based method for feature subset selection*. Soft Computing, vol. 12, pages 111–120, 2007.

[Feo & M.-G.-C.Resende 1989] T.-A. Feo et M.-G.-C.Resende. *A probabilistic heuristic for a computationally difficult set covering problem*. Operations Research Letters, vol. 8, pages 67–71, 1989.

[Feo 1994] T.-A. Feo. *Greedy randomized adaptive search procedures*. Journal of Global Optimization, vol. 6, pages 109–133, 1994.

[Flynn 1972] M.-J. Flynn. *Some computer organizations and their effectiveness*. IEEE Transactions on Computers archive, vol. 21, no. 9, pages 948–960, 1972.

[Frawley *et al.* 1991] W.-J. Frawley, G. Piatetsky-Shapiro et C.-J. Matheus. *Knowledge Discovery in Databases : An Overview*. in G. Piatetsky-Shapiro and W. J. Frawley (eds.), Knowledge Discovery in Databases, AAAIIMIT Press, pages 1–27, 1991.

[Freund & Schapire 1996] Y. Freund et R. Schapire. *Experiments with a New Boosting Algorithm*. In Proceedings of the Thirteenth International Conference Machine Learning, pages 148–156. Morgan Kaufmann Publishers Inc, 1996.

[Friedman *et al.* 1997] J.-H. Friedman, J.-L. Bentley et R.-A. Finkel. *An algorithm for finding best matches in logarithmic expected time*. ACM Trans Math Softw, vol. 3, no. 3, page 209–226, 1997.

[Furnkranz 1999] J. Furnkranz. *eparate-and-conquer rule learning*. Artificial Intelligence, vol. 13, no. 1, pages 3–54, 1999.

[Gallardo *et al.* 2007] J.-E. Gallardo, C. Cotta et A.-J. Fernandez. *On the hybridization of memetic algorithms with branch-and-bound techniques*. Systems, Man, and Cybernetics, Part B : Cybernetics, IEEE Transactions on, vol. 37, no. 1, pages 77–83, 2007.

[Garey & Johnson 1979] M.-R. Garey et D.-S. Johnson. *Computers and intractability ; a guide to the theory of NPcompleteness*. W.H. Freeman Company, pages 17–44, 1979.

[Gennari & Fisher 1989] J.-H. Gennari et D. Fisher. *Models of incremental concept formation*. Artificial Intelligence, pages 11–61, 1989.

[Glover & Laguna 1997] F. Glover et M. Laguna. *Tabu Search*. In HANDBOOK OF COMBINATORIAL OPTIMIZATION, volume 3, pages 621–757. Kluwer Academic Publishers, 1997.

[Glover 1977] F. Glover. *Heuristics for integer programming using surrogate constraints*. Decision Sciences, vol. 8, no. 1, pages 156–166, 1977.

[Glover 1986] F. Glover. *Future paths for integer programming and links to artificial intelligence*. Computers and Operations Research, vol. 13, page 533−549, 1986.

[Goldberg 1989] D.-E. Goldberg. *Genetic Algorithms - in Search, Optimization and Machine Learning*. Edison-Wesley Publishing Company, 1989.

[G.Raidl 2006] G.Raidl. *Unified View on Hybrid Metaheuristics*. In Proceedings of HM 2006 – Third International Workshop on Hybrid Metaheuristics, 2006.

[Gray & Fan 2008] J.-B. Gray et G. Fan. *Classification tree analysis using TARGET*. Computer Statistic Data Analytic, vol. 52, no. 3, page 1362–1372, 2008.

[Greene & Smith 1993] D.-P. Greene et S-F. Smith. *Competition-based induction of decision models from examples*. Machine Learning volume, vol. 13, no. 2-3, pages 229–257, 1993.

[Grefenstette 1987] J.-J. Grefenstette. *Incorporating problem specific knowledge into genetic algorithms*. L.Davis (Ed.) Genetic Algorithms and Simulated Annealing, Morgan Kaufmann Publishers, pages 42–60, 1987.

[Grimaldi *et al.* 2004] E.-A. Grimaldi, F. Grimacia, M. Mussetta, P. Pirinoli et R. Zich. *A new hybrid genetical-swarm algorithm for electromagnetic optimization*. In Proceedings of the International Conference on Computational Electromagnetics and its Applications, Beijing, China, pages 157–160. Chinese Institute of Electronics, 2004.

[Grosan & Abraham 2007] C. Grosan et A. Abraham. *Hybrid Evolutionary Algorithms : Methodologies, Architectures, and Reviews*. Studies in Computational Intelligence (SCI), vol. 75, pages 1–17, 2007.

[Group 2019] M.-L. Group. Machine Learning Group at the University of Waikato.
Retrieved from https ://www.cs.waikato.ac.nz/ml/weka/, 2019.

[Guan & Zhu 2005] S.-U. Guan et F. Zhu. *An incremental approach to genetic algorithms-based classification*. IEEE Transactions on System, Man and Cybernetics, vol. 35, no. 2, page 227–239, 2005.

[Guyon & Elisseeff 2003] I. Guyon et A. Elisseeff. *An introduction to variable and feature selection*. Journal of Machine Learning Research, page 1157–1182, 2003.

[Guyon *et al.* 2006] I. Guyon, S. Gunn, M. Nikravesh et L. Zadeh. *Feature Extraction Foundations and Applications*. (Eds.) Feature extraction. Heidelberg, Germany : Springer, vol. 207, 2006.

[Hall & Frank 2008] M. Hall et E. Frank. *Combining Naive Bayes and Decision Tables*. In Proceedings of the 21st Florida Artificial Intelligence Society Conference (FLAIRS), page 318–319. AAAI Press, 2008.

[Han & Cerconet 1992] Y.-C. Han et N. Cerconet. *Knowledge Discovery in Databases : An Attribute-oriented approach*. In Proceedings of the 18th VLDB Conference, pages 547–559, 1992.

[Han *et al.* 2000] J. Han, J. Pei et Y. Yin. *Mining frequent patterns without candidate generation*. ACMSIGMOD Int. Conf. Management of Data (SIGMOD'00), pages 1–12, 2000.

[Hansen 2000] P. Hansen. *Tabu Search for Multiobjective Combinatorial Optimization*. Control and Cybernetics, vol. 29, no. 3, pages 799–818, 2000.

[Harb & Desuky 2014] M. Harb et A. Desuky. *Feature Selection on Classification of Medical Datasets based on Particle Swarm Optimization*. International Journal of Computer Applications, vol. 104, no. 5, pages 14–17, 2014.

[Harilaos 1976] Papadimitriou Harilaos. *The complexity of combinatorial optimization problems*. PhD thesis, 1976.

[Hart 1968] P.-E. Hart. *The condensed nearest neighbor rule*. IEEE Trans Inf Theory, vol. 14, page 515–516, 1968.

[Holden & Freitas 2008] N. Holden et A. Freitas. *A Hybrid PSO/ACO Algorithm for Discovering Classification Rules in Data Mining*. Journal of Artificial Evolution and Applications, pages 1–11, 2008.

[Holland 1962] J. Holland. *Outline for a logical theory of adaptive systems*. Journal of the Association of Computing Machinery, vol. 9, no. 3, pages 297–314, 1962.

[Holland 1986] J.-H. Holland. *Escaping brittleness : The possibilities of general purpose learning algorithms applied to parallel rule-based systems*. In Machine Learning : An Artificial Intelligence Approach, vol. 2, page 593–623, 1986.

[Holte 1993] R.-C. Holte. *Very simple classification rules perform well on most commonly used datasets*. Machine Learning, vol. 11, pages 63–91, 1993.

[Holzinger 2017] A. Holzinger. *Introduction to Machine Learning et Knowledge Extraction (MAKE)*. Holzinger Group, HCI-KDD, Institute for Medical Informatics et Statistics, Medical University Graz, vol. 1, no. 1, pages 1–20, 2017.

[Hongbin & Guangyu 2002] Z. Hongbin et S. Guangyu. *Feature selection using Tabu Search method*. Pattern Recognition, pages 701–711, 2002.

[Hsieh *et al.* 2014] Z. Hsieh, M. Chunsu et P. Chunwang. *A PSO-based rule extractor for medical diagnosis*. Journal of Biomedical Informatics, no. 49, pages 53–60, 2014.

[Hussain *et al.* 1999] F. Hussain, H. Liu et C. Tanand M. Dash. *Discretization : an enabling technique (Technical Report TRC6/99)*. School of Computing, National University of Singapore, 1999.

[Ishibuchi *et al.* 2001] H. Ishibuchi, T. Nakashima et M. Nii. *Learning of neural networks with GA-based instance selection*. In Proceedings of the 20th North American Fuzzy In- formation Processing Society International Conference, Vancouver, Canada, volume 4, page 2102–2107, 2001.

[Jameel & ur rehman 2018] S. Jameel et Saif ur rehman. *An Optimal Feature Selection Method Using A Modified Wrapper Based Ant Colony Optimization*. Journal of the National Science Foundation of Srilanka, vol. 46, no. 2, pages 143–151, 2018.

[Janikow 1993] C.-Z. Janikow. *A knowledge-intensive genetic algorithm for supervised learning*. Machine Learning, vol. 13, no. 2-3, page 189–228, 1993.

[Javadi 2005] A.-A. Javadi. *A hybrid intelligent genetic algorithm*. Advanced Engineering Informatics, pages 255–262, 2005.

[Jiao *et al.* 2006] L. Jiao, J. Liu et W. Zhong. *An organizational co-evolutionary algorithm for classification*. IEEE Transactions on Evolutionary Computation, vol. 10, no. 1, page 67–80, 2006.

[Jimenez & Rodriguez 2014] S.-S Jimenez et M.-V. Rodriguez. *Parallel Multiobjective Metaheuristics for Inferring Phylogenies on Multicore Clusters*. IEEE Transactions on Parallel and Distributed Systems, vol. 26, no. 6, pages 1678 – 1692, 2014.

[Johann 2006] Dréo Johann. *Adaptation de la méthode des colonies de fourmis pour l'optimisation en variables continues : application en génie biomédical*. PhD thesis, Université Paris 12 Val de Marne, 2006.

[John & Langley 1995] G.-H. John et P. Langley. *Estimating Continuous Distributions in Bayesian Classifiers*. In Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, pages 338–345. Morgan Kaufmann, 1995.

[John *et al.* 1994] G.-H. John, R. Kohavi et K. Pfleger. *Irrelevant features and the subset selection problem*. In Proceedings of the Eleventh International Conference on Machine Learning, pages 121–129. Morgan Kaufmann, 1994.

[José-Crispin 2008] Hernandez José-Crispin. *Algorithmes Métaheuristiques Hybrides Pour La Sélection De Gènes Et La Classification De Données De Biopuces*. PhD thesis, Université Angers, 2008.

[José-riquelme *et al.* 2003] C. José-riquelme, J. Aguilar-ruiz et M. Toro. *Finding representative patterns with ordered projections*. Pattern Recognit, vol. 36, page 1009–1018, 2003.

[Jourdan *et al.* 2001] L. Jourdan, C. Dhaenens et G. Talbi. *A Genetic Algorithm for Feature Selection in Data-Mining for Genetics*. 4th Metaheuristics International Conference, page 841–855, 2001.

[Jourdan *et al.* 2006] L. Jourdan, C. Dhaenens et G. Talbi. *Hybridation Métaheuristiques/DataMining Using Datamining Techniques to Help Metaheuristics : A Short Survey*. Springer-Verlag Berlin, Heidelberg, page 841–855, 2006.

[Jourdan 2003] Laetitia Jourdan. *Métaheuristiques Pour L'extraction De Connaissances : Application À La Génomique*. PhD thesis, Université Des Sciences Et Technologies De Lille, 2003.

[Jourdann *et al.* 2005] L. Jourdann, D. Corne, D.-A. Savic et G.-A. Walters. *Preliminary investigation of the learnable evolution model for faster/better multiobjective water systems design*. In LNCS 3410, editor, Third International Conference on Evolutionary Multi-Criterion Optimization (EMO'05), page 841–855, 2005.

[Juille & Pollack 1996] H. Juille et J.-B. Pollack. *Dynamics of co-evolutionary learning*. In Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior : From animals to animats 4, Cape Code, USA, volume 24 of *59*, page 526–534. MIT Press, 1996.

[Kabir *et al.* 2013] M. Kabir, Shahjahan et K. Murase. *Ant Colony Optimization Toward Feature Selection*. Ant Colony Optimization - Techniques and Applications, vol. 17, no. 1, pages 3–44, 2013.

[Kanan *et al.* 2007] H.-R. Kanan, K. Faez et S.-M. Taheri. *Feature Selection using Ant Colony Optimization (ACO) : A New Method and Comparative Study in the Application of Face Recognition System.* Proceedings of International Conference on Data Mining, pages 63–76, 2007.

[Karp & Flatt 1990] A.-H. Karp et H.-P. Flatt. *Measuring parallel processor performance*. Communications of the ACM, vol. 33, no. 5, pages 539–543, 1990.

[Ke *et al.* 2008] L. Ke, Z. Feng et Z. Ren. *An Efficient Ant Colony Optimization Approach to Attribute Reduction in Rough Set Theory.* Pattern Recognition Letters, vol. 29, pages 1351–1357, 2008.

[Kennedy & Eberhart 1995] J. Kennedy et R.-C. Eberhart. *Particle Swarm Optimisation*. IEEE Int. Conf. On Neural Networks, pages 1942–1948, 1995.

[Khao *et al.* 1999] J.-K. Khao, P. Galinier et M. Habib. *Metaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes*. Revue d'Intelligence Artificielle, vol. 13, no. 2, pages 283–324, 1999.

[Khushaba *et al.* 2008] R.-N. Khushaba, A. Alsukker, A. Ani et A.-A. Jumaily. *Enhanced Feature Selection Algorithm using Ant Colony Optimization and Fuzzy Memberships :*. proceedings of the sixth international conference on biomedical engineering. IASTED, pages 34–39, 2008.

[Kirkpatrick & Gelatt 1983] S. Kirkpatrick et C.-D. Gelatt. *Optimization by Simulated Annealing*. Vecchi M.P., SCIENCE, pages 671–680, 1983.

[Kohavi & John 1997] R. Kohavi et G.-H. John. *Wrappers for feature subset selection*. Artif. Intell, page 273–324, 1997.

[Kohavi 1995] R. Kohavi. *The Power of Decision Tables*. 8th European Conference on Machine Learning, page 174–189, 1995.

[Koller & Sahami 1996] D. Koller et M. Sahami. *Toward optimal feature selection*. 13th International conference on machines learning, pages 1–15, 1996.

[Kordos 2017] M. Kordos. *Optimization of Evolutionary Instance Selection*. University of Bielsko-Biala, Department of Computer Science and Automatics. in : Lecture Notes in Artificial Intelligence, vol. 10245, pages 1–11, 2017.

[Koza 1991] J.-R. Koza. *Genetic evolution and co-evolution of computer programs*. Dans Langton, C. T. C., Farmer, J. D. et Rasmussen, S., éditeurs : Artificial Life II, volume X de SFI Studies in the Sciences of Complexity, 603–629. Addison-Wesley, Santa Fe Institute, New Mexico, USA, pages 603–629, 1991.

[Koza 1992] J.-R. Koza. *Genetic Programming : On the Programming of Computers by Means of Natural Selection.* MIT Press, 1992.

[Krishnanand & Ghose 2005a] K.-N. Krishnanand et D. Ghose. *Detection of Multiple Source Locations using a Glowworm Metaphor with Applications to Collective Robotics*. Swarm Intelligence Symposium, page 84–91, 2005.

[Krishnanand & Ghose 2005b] K.-N. Krishnanand et D. Ghose. *Multimodal Function Optimization using a Glowworm Metaphor with Applications to Collective Robotics*. In Proceedings of the the 2nd Indian International Conference on Artificial, page 328–346, 2005.

[Krishnanand & Ghose 2009] K.-N. Krishnanand et D. Ghose. *Glowworm Swarm Optimization for Simultaneous Capture of Multiple Local Optima of Multimodal Functions*. Swarm Intelligence, vol. 3, no. 2, page 84–91, 2009.

[Krishnanand *et al.* 2006] K.-N. Krishnanand, P. Amruth, M.-H. Guruprasad, S.-V. Bidargaddi et D. Ghose. *Glowworminspired Robot Swarm for Simultaneous Taxis towards Multiple Radiation Sources*. IEEE International Conference on Robotics and Automation, page 958–963, 2006.

[Kumar & Mewada 2011] S. Kumar et P. Mewada. *ACO based Feature subset Selection for multiple K-Nearest Neighbor Classifiers*. International Journal on Computer Science and Engineering, vol. 3, no. 5, pages 1831–1838, 2011.

[Kumar *et al.* 2014] G. Kumar, G.-A. Ramachandra et K. Nagamani. *An Efficient Feature Selection System to Integrating SVM with Genetic Algorithm for Large Medical Datasets*. International Journal of Advanced Research in Computer Science and Software Engineering, vol. 4, no. 2, pages 272–277, 2014.

[Kumar 2000] V. Kumar. *An Introduction to Cluster Analysis for Data Mining*. Technical report, C.S. Dept. Univ. Minnesota., 2000.

[Lahrichi & T.-G.Crainic 2015] N. Lahrichi et W. Rei T.-G.Crainic M-Gendreau. *An Integrative Cooperative Search Framework for Multi-Decision-Attribute Combinatorial Optimization : Application to the MDPVRP*. European Journal of Operational Research, vol. 246, no. 2, pages 400–412, 2015.

[Langley & Iba 1993] P. Langley et W. Iba. *Average-case analysis of a nearest neighbor algorithm*. Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI), vol. 2, page 889–894, 1993.

[Larranaga & Lozano 2001] P. Larranaga et J.-A. Lozano. *Estimation of Distribution Algorithms : A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, 2001.

[Lavangnananda 2004] K. Lavangnananda. *A Genetic Programming Approach to Inductive Learning*. In Proceedings of the International Conference Computational Intelligence for Modelling, Control and Automation, Gold Coast, Australia. CIMCA, 2004.

[Lavangnananda 2006] K. Lavangnananda. *Self-adjusting Associative Rules Generator for Classification : An Evolutionary Computation Approach*. IEEE, pages 237–242, 2006.

[Lee & Takagi 1993] M. Lee et H. Takagi. *Dynamic control of genetic algorithms using fuzzy logic techniques*. In Proceedings of the 5th International Conference on Genetic Algorithms, page 76–83. Morgan Kaufmmann, San Mateo, 1993.

[Li *et al.* 2017] P. Li, H. Wang et X. Li. *Improved ant colony algorithm for global path planning*. Advances in Materials, Machinery, Electronics I, 2017.

[Liam *et al.* 2012] C. Liam, X. Bing, Z. Mengjie et S. Lin. *Binary Particle Swarm Optimization for Feature Selection : A Filter Based Approach*. In Proceedings of the IEEE World Congress on Computational Intelligence. IEEE, 2012.

[Lin & Liang 2006] T. Lin et S.-C. Liang. *A Hybrid metaheuristic for the quadratic assignment problem*. Computational Optimization and Applications, vol. 34, no. 1, pages 85–113, 2006.

[Liu & Motoda 1998] H. Liu et H. Motoda. *Feature selection for knowledge discovery and data mining*. Norwell Kluwer, 1998.

[Liu & Motoda 2001] H. Liu et H. Motoda. *Instance selection and construction for data mining*. New York Springer, 2001.

[Liu & Motoda 2002] H. Liu et H. Motoda. *On issues of instance selection*. Data Mining and Knowledge Discovery, vol. 6, no. 2, page 115–130, 2002.

[Liu & Motoda 2008] H. Liu et H. Motoda. *Computational methods of feature selection*. Taylor Francis group, 2008.

[Liu & Yu 2005] H. Liu et L. Yu. *Toward Integrating Feature Selection Algorithms for Classification and Clustering*. IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 4, pages 491 – 502, 2005.

[Liu & Zheng 2005] Y. Liu et F. Zheng. *FS SFS : A novel feature selection method for support vector machines*. The journal of Pattern Recognition Society, vol. 39, no. 7, pages 1333–1345, 2005.

[Lozano *et al.* 2006] M. Lozano, J.-M. Sotoca, J.-S. Sánchez, F. Pla, E. Pekalska et R.-P.-W. Duin. *Experimental study on prototype optimisation algorithms for prototype-based classification in vector spaces*. Pattern Recognition, vol. 39, no. 10, pages 827–1838, 2006.

[Lumini & Nanni 2006] A. Lumini et L. Nanni. *A clustering methods for automatic biometric template selection*. pattern recognition, vol. 39, pages 495–497, 2006.

[Macqueen 1967] J.-B. Macqueen. *Some Methods for classification and Analysis of Multivariate Observations*. In Proceedings of the 5-th Berkeley Symposium on Mathematical Statistics and Probability, volume 1, pages 281–297. Berkeley, University of California Press, 1967.

[Mageshkumar *et al.* 2018] C. Mageshkumar, S. Karthik et V.-P. Arunachalam. *Hybrid metaheuristic algorithm for improving the efficiency of data clustering*. Cluster Comput, vol. 22, page 435–442, 2018.

[Manic 2011] M. Manic. *Data Mining*. Edited by M.B.Wilamowski and J.D.Irwin The industrial electronics handbook second edition. Intelligent system., 2011.

[Marill & Green 1963] T. Marill et D-M. Green. *On the effectiveness of receptors in recognition systems*. IEEE Transactionson Computers, vol. 9, pages 11–17, 1963.

[Mauricio & Ribeiro 2002] G.-C.-R. Mauricio et C. Ribeiro. *Greedy randomized adaptive search procedures*. To appear in State of the Art Handbook in Metaheuristics, F. Glover and G. Kochenberger, eds., Kluwer, vol. 2, pages 219–249, 2002.

[Michalksi *et al.* 1986] R.-S. Michalksi, I. Mozetic et N. Lavrac. *The multipurpose incremental learning system AQ15 and its testing application to three medical domains*. In Proceedings of the Artificial Intelligence, Fifteenth International Conference, page 1041–1045. AAAI, 1986.

[Michalski *et al.* 2000] R.-S. Michalski, G. Cervon et K.-A. Kaufman. *Speeding up evolution through learning : Lem*. In Intelligent Information Systems, page 243–256, 2000.

[Michalski 1983] R.-S. Michalski. *A theory and methodology of inductive learning*. Artificial Intelligence, page 111–116, 1983.

[Michalski 2000] R.-S. Michalski. *Learnable evolution model : Evolutionary processes guided by machine learning*. Machine Learning, vol. 38, no. 1-2, pages 9–40, 2000.

[Miller & Jiawei 2001] H.-J. Miller et H. Jiawei. *Geographic data mining and knowledge discovery : an overview*. New York : Taylor et Francis, 2001.

[Min 2016] S.-H. Min. *Integrating Instance Selection and Bagging Ensemble using a Genetic Algorithm*. International Journal of Applied Engineering Research, vol. 11, no. 7, pages 5060–5066, 2016.

[Mingers 1989] J. Mingers. *An empirical comparison of pruning methods for decision tree induction*. Machine Learning, vol. 4, no. 2, page 227–243, 1989.

[Mladenovic & Hansen 1997] N. Mladenovic et P. Hansen. *Variable Neighborhood Search*. Computers et Operations Research, vol. 24, page 1097–1100, 1997.

[Molina *et al.* 2002] L.-C. Molina, L. Belanche et A. Nebot. *Evaluating feature selection algorithms*. CCIA-LNCS, vol. 2504, page 216–227, 2002.

[Mollineda *et al.* 2002] R.-A. Mollineda, F.-J. Ferri et E. Vidal. *An efficient prototype merging strategy for the condensed 1-NN rule through class-conditional hierarchical clustering*. Pattern Recognit, vol. 35, no. 12, page 2771–2782, 2002.

[Muhlenbein & Paass 1996] H. Muhlenbein et G. Paass. *From recombination of genes to the estimation of distributions i. binary parameters*. In PPSN, vol. 1141, pages 178–187, 1996.

[Mühlenbein *et al.* 1988] H. Mühlenbein, M. Gorges-Schleuter et O. Krämer. *Evolution algorithms in combinatorial optimization*. Parallel Computing, vol. 7, pages 65–85, 1988.

[Narendra & Fukunaga 1977] P.-M. Narendra et K. Fukunaga. *A branch and bound algorithm for feature subset selection*. IEEE Trans. on Computer, page 917–922, 1977.

[Nematzadeh *et al.* 2019] H. Nematzadeh, R. Enayatifar, M. Mahmud et E. Akbari. *Frequency based feature selection method using whale algorithm*. Genomics, vol. 111, no. 6, pages 1946–1955, 2019.

[Nemhauser & Wolsey 1988] G.-L. Nemhauser et A.-L. Wolsey. *Integer and Combinatorial Optimization*. John Wiley et Sons New York, 1988.

[Oduntan & Thulasiraman 2018] O.-I. Oduntan et P. Thulasiraman. *Hybrid Metaheuristic Algorithm for Clustering*. IEEE Symposium Series on Computational Intelligence (SSCI), pages 1–9, 2018.

[Oh *et al.* 2004] I. Oh, J. Lee et B. Moon. *Hybrid GAs for feature selection*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 11, page 1424–1437, 2004.

[Oliver *et al.* 2018] C. Oliver, L.-S. Miriam, B.-P. Nigel et L.-K. Kurt. *New Zealand glowworm (Arachnocampa luminosa) bioluminescence is produced by a firefly-like luciferase but an entirely new luciferin*. Scientific Reports, vol. 8, no. 3278, pages 1–15, 2018.

[Olvera-Lopez *et al.* 2007] J.-A. Olvera-Lopez, J.-A. Carrasco-Ochoa et J.-A. Martinez-trinidad. *Object selection based on clustering and border objects*. In : Kurzynski M et al (eds) Computer recognition systems 2, ASC 45. Wroclaw, Poland, page 27–34, 2007.

[Olvera-Lopez *et al.* 2008] J.-A. Olvera-Lopez, J.-A. Carrasco-Ochoa et J.-F. Martinez-trinidad. *Prototype selection via prototype relevance*. In :Ruiz-Shulcloper J,KropatschWG(eds)CIARP 2008,LNCS5197. Habana,Cuba, page 153–160, 2008.

[Olvera-Lopez *et al.* 2010] J.-A. Olvera-Lopez, A.-J. Carrasco-Ochoa, J.-F. Martinez-trinidad et J. Kittler. *A review of instance selection methods*. Artif Intell Rev, vol. 34, page 133–143, 2010.

[Osman & Laporte 1996] I.-H. Osman et G. Laporte. *Metaheuristics : A bibliography*. Annals of Operations Research, vol. 63, pages 513–623, 1996.

[Papadimitriou & Steiglitz 1982] C.-H. Papadimitriou et K. Steiglitz. *Combinatorial Optimization - Algorithms and Complexity*. Dover Publications, Inc., New York, 1982.

[Paredes & Vidal 2000] R. Paredes et E. Vidal. *Weighting prototypes. A new editing approach*. In Proceedings of the international conference on pattern recognition, volume 2, pages 25–28. IEEE Computer Society, 2000.

[Perner 2001] P. Perner. *Improving the accuracy of decision tree induction by feature pre-selection*. Applied Artificial Intelligence, pages 747–760, 2001.

[Porto *et al.* 2000] S.-C.-S. Porto, J.-P.-F.-W. Kitajima et C.-C. Ribeiro. *Performance Evaluation of a Parallel Tabu Search Task Scheduling Algorithm*. Parallel Computing, vol. 26, no. 1, pages 73–90, 2000.

[Provost & Kolluri 1999] F.-J. Provost et V. Kolluri. *A survey of methods for scaling up inductive learning algorithms*. Data Mining and Knowledge Discovery, vol. 2, page 131–169, 1999.

[Puchinger & Raidl 2005] J. Puchinger et G.-R. Raidl. *Combining metaheuristics and exact algorithms in combinatorial optimization : A survey and classification*. In Proceedings of the First International Work-Conference on the Interplay Between Natural and Artificial Computation, volume 3562, page 41–53. LNCS, 2005.

[Pyle 1999] D. Pyle. *Data preparation for data mining*. San Francisco : Morgan Kaufmann, 1999.

[Quinlan 1986] R.-J. Quinlan. *Induction of decision trees*. Machine Learning, pages 81–106, 1986.

[Quinlan 1988] R.-J. Quinlan. *Decision Trees and Multivalued Attributes*. Machine Intelligence, vol. 11, pages 305–318, 1988.

[Quinlan 1993] R.-J. Quinlan. *Programs for Machine Learning*. San Mateo, CA : Morgan Kaufmann, 1993.

[Quinlan 1995] R.-J. Quinlan. *MDL and categorical theories (continued).* In Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, page 464–470. Morgan Kaufmann, 1995.

[Raidl *et al.* 2010] G. Raidl, J. Puchinger et C. Blum. *Metaheuristic Hybrids*. In Handbook of Metaheuristics, volume 146, page 469–496. Springer., 2010.

[Raidl 2015] G. Raidl. *Decomposition Based Hybrid Metaheuristics*. European Journal of Operational Research, vol. 244, no. 1, pages 66–76, 2015.

[Rajiv *et al.* 2012] G.-K. Rajiv, S.-M. Uma et M. Karnan. *A Hybrid Meta Heuristic Algorithm for Discovering Classification Rule in Data Mining*. IJCSNS International Journal of Computer Science and Network Security, vol. 12, no. 4, pages 116–122, 2012.

[Riddle 1998] P. Riddle. *GABIL Extensions*. Retrieved from https ://www.cs.auckland.ac.nz/ pat/706_98/ln/node193.html, 1998.

[Ritter *et al.* 1975] G. Ritter, H. Woodruff, S. Lowry et T. Isenhour. *An algorithm for a selective nearest neighbor decision rule*. IEEE Trans Inf Theory, vol. 21, no. 6, page 665–669, 1975.

[Rizwan *et al.* 2017] M. Rizwan, S. Waseem et A. Ejaz. *Maximum Relevancy Minimum Redundancy Based Feature Subset Selection using Ant Colony Optimization*. Journal of Applied Environmental and Biological Sciences, vol. 7, no. 4, pages 118–130, 2017.

[Robbins *et al.* 2008] K.-R. Robbins, W. Zhang, J.-K. Bertrand et R. Rekaya. *The Ant Colony Algorithm for Feature Selection in High-Dimension Gene Expression Data for Disease Classificafition.* Journal of Mathematical Medicine and Biology, pages 1–14, 2008.

[Ronen & Jacob 2004] M. Ronen et Z. Jacob. *Using simulated annealing to optimize feature selection problem in marketing applications*. European Journal of Operational Research, vol. 171, no. 3, pages 842–858, 2004.

[Russell & Norvig 1995] S.-J. Russell et P. Norvig. *Artificial Intelligence : A Modern Approach*. Englewood Cliffs, NJ, 1995.

[Russell & Norvig 2003] S.-J. Russell et P. Norvig. *Artificial Intelligence : A Modern Approach*. Pearson Education, 2003.

[Santos *et al.* 2006] H.-G. Santos, L.-S. Ochi, E.-H. Marinho et L.-M.-A. Drummond. *Combining an evolutionary algorithm with data mining to solve a vehicle routing problem*. Neurocomputing, 2006.

[Saporta 2006] G. Saporta. *Probabilité, analyse des données et statistique*. Editions Technip, paris, page 622, 2006.

[Schapire 1990] R. Schapire. *The strength of weak learning*. Machine Learning, vol. 5, no. 2, pages 197–227, 1990.

[Seban & Nock 2001] M. Seban et R. Nock. *Impact of learning set quality and size on decision tree performances*. International Journal of Computers, Systems and Signals, pages 85–105, 2001.

[Shahzad & Baig 2011] W. Shahzad et R.-A. Baig. *Hybrid associative classification algorithm using ant colony optimization*. International journal of innovative computing, information & control : IJICIC, vol. 7, no. 12, pages 6815–6826, 2011.

[Shahzad 2010] Waseem Shahzad. *Classification and associative classification rule discovery using ant colony optimization*. PhD thesis, Islamabad-Pakistan, 2010.

[Shakhnarovich *et al.* 2006] G. Shakhnarovich, T. Darrell et P. Indyk. *Nearest-Neighbor Methods in Learning and Vision*. The MIT Press, 2006.

[Shi *et al.* 2005] X. H. Shi, Y.C.Liang, H.P.Lee et C.Lu L.M.Wang. *An improved GA and a novel PSO-GA-based hybrid algorithm*. Information Processing Letters, vol. 93, no. 5, pages 255–261, 2005.

[Sivagaminathan & Ramakrishnan 2007] R.-K. Sivagaminathan et S. Ramakrishnan. *A Hybrid Approach for Feature Subset Selection using Neural Networks and Ant Colony Optimization.* Expert Systems with Applications, pages 33–49, 2007.

[Soltani *et al.* 2015] A. Soltani, S.-M. Ahadi, N. Faraji et S. Sharifian. *Designing efficient discriminant functions for multi-category classification using evolutionary methods*. Neurocomputing, vol. 173, no. 3, pages 1885–1897, 2015.

[Subanya & Rajalaxmi 2014] B. Subanya et R. Rajalaxmi. *A Novel Feature Selection Algorithm for Heart Disease Classification*. International Journal of Computational Intelligence and Informatic, vol. 4, no. 2, pages 117–124, 2014.

[Sun & Peng 2014] S. Sun et Q. Peng. *A hybrid PSO-GSA strategy for high-dimensional optimization and microarray data clustering*. IEEE International Conference on Information and Automation (ICIA), page 41–46, 2014.

[Sébastien 2010] Parfait Sébastien. *Classification de spectres et recherche de biomarqueurs en spectroscopie par résonance magnétique nucléaire du proton dans les tumeurs prostatiques*. PhD thesis, UFR tciences et techniques université de bourgogne, 2010.

[Talbi & Bachelet 2006a] E.-G. Talbi et V. Bachelet. *COSEARCH : A parallel cooperative metaheuristic*. Journal of Mathematical Modelling and Algorithms (JMMA), vol. 5, no. 2, pages 5–22, 2006.

[Talbi & Bachelet 2006b] E.-G. Talbi et V. Bachelet. *A parallel hybrid genetic algorithm for protein structure prediction on the computational grid*. Journal of Mathematical Modelling and Algorithms, vol. 5, no. 1, pages 5–22, 2006.

[Talbi *et al.* 2007] E.-G. Talbi, S. Cahon et N. Melab. *Designing cellular networks using a parallel hybrid metaheuristic*. Journal of Computer Communications, vol. 30, no. 4, pages 698–713, 2007.

[Talbi 2002] E.-G. Talbi. *A Taxonomy of Hybrid Metaheuristics*. Journal of Heuristics, vol. 8, no. 5, pages 541–564, 2002.

[Talbi 2009] E.-G. Talbi. *Metaheuristics : From Design to Implementation*. Wiley, 2009.

[Talbi 2013] E.-G. Talbi. *A Unified Taxonomy of Hybrid Metaheuristics with Mathematical Programming, Constraint Programming and Machine Learning Studies*. in Computational Intelligence, pages 3–76, 2013.

[Talukdar *et al.* 1998] S.-N. Talukdar, L. Baerentzen, A. Gove et P. Desouza. *Asynchronous teams : cooperation schemes for autonomous agents*. Journal of Heuristics, vol. 4, no. 4, page 295–321, 1998.

[Talukdar 1999] S.-N. Talukdar. *Collaboration rules for autonomous software agents*. Decision Support Systems, vol. 24, no. 3-4, pages 269–278, 1999.

[Tan *et al.* 2003] K. Tan, Q. Yu, C.-M. Heng et T.-H. Lee. *Evolutionary computing for knowledge discovery in medical diagnosis*. Artificial Intelligence in Medicine, vol. 27, no. 2, pages 129–154, 2003.

[Tan *et al.* 2006] K. Tan, Q. Yu et J.-H Ang. *A co-evolutionary algorithm for rules discovery in data mining*. International Journal of Systems Science, vol. 37, no. 12, page 835–864, 2006.

[Tanese 1987] R. Tanese. *Parallel genetic Algorithms for a Hypercube*. International Conference on Genetic Algorithms MIT Cambridge MA USA, pages 177–183, 1987.

[Tantar *et al.* 2008] A. Tantar, N. Melab et E. Talbi. *A grid-based genetic algorithm combined with an adaptive simulated annealing for protein structure prediction*. Soft Computing, page 1185–1198, 2008.

[Tomek 1976] I. Tomek. *An experiment with the edited nearest-neighbor rule*. IEEE Trans Syst Man Cybern, vol. 6, no. 6, page 448–452, 1976.

[Usama *et al.* 1996] M. Usama, Fayyad, P.-S. Gregory et S. Padhraic. *From data mining to knowledge discovery : An overview*. In Advances in Knowledge Discovery and Data Mining, vol. 1, no. 34, 1996.

[Venmann & Reinders 2005] C.-J. Venmann et M.-J.T. Reinders. *The nearest sub-class classifier : a compromise between the nearest mean and nearest neighbor classifier*. IEEE Trans Pattern Anal Mach Intell, vol. 27, no. 9, page 1417–1429, 2005.

[Venmann *et al.* 2002] C.-J. Venmann, M.-J.T. Reinders et E. Backer. *A maximum variance clustering algorithm*. IEEE Trans Pattern Anal Mach Intell, vol. 24, no. 9, page 1273–1280, 2002.

[Venturini 1993] G. Venturini. *SIA : A supervised inductive algorithm with genetic search for learning attributes based concepts*. In Proceedings of the Machine Learning, 12th International Conference Berlin, Germany, page 280–296, 1993.

[Verhoeven & Aarts 1995] M.-G. Verhoeven et E.-H.-L. Aarts. *Parallel local search*. Journal of Heuristics, page 43–65, 1995.

[Vieira *et al.* 2010] S.-M. Vieira, J.-M.-C. Sousa et T.-A. Runkler. *Two cooperative ant colonies for feature selection using fuzzy models*. Expert Systems with Applications, vol. 37, page 2714–2723, 2010.

[Wang & Song 2012] G. Wang et Q. Song. *Selecting Feature Subset via Constraint Association Rules*. In Advances in Knowledge Discovery and Data Mining, pages 304–321, 2012.

[Whitney 1971] W. Whitney. *A direct method of nonparametric measurement selection*. IEEE Transactions on Computers, vol. 20, page 1100–1103, 1971.

[Wijsen 2001] J. Wijsen. *Data Mining et Data Warehousing*. 2001.

[Wilson 1972] D.-L. Wilson. *Asymptotic properties of nearest neighborrules using editing data.* IEEE Trans. Syst., Man, Cybern, vol. 2, no. 3, pages 408–421, 1972.

[Wilson 1995] S. Wilson. *Classifier fitness based on accuracy*. Evolutionary Computation, vol. 3, no. 2, page 149–175, 1995.

[Yang 2009] Y. Yang. *Firefly algorithms formultimodal optimization*. In Proceedings of the Stochastic Algorithms : Foundations and Applications in Computing Sciences, volume 5792, pages 178–178. Springer, Sapporo, Japan, 2009.

[Yu & Liu 2004] L. Yu et H. Liu. *Efficient feature selection via analysis of relevance and redundancy*. Journal of Machine Learning Research, page 1205–1224, 2004.

[Zhu & Guan 2004] F. Zhu et S.-U. Guan. *Ordered incremental training with genetic algorithms*. International Journal Intelligence System, vol. 19, no. 12, page 1239–1256, 2004.

الملخص

اكتشاف نماذج واضحة وفعالة من البيانات هو هدفنا مع التحدي الذي هو العمل مع كمية كبيرة من البيانات واستخدام خوارزميات قوية لاستخراج المعرفة الأساسية. يعد التصنيف الخاضع للإشراف من بين المهام التي تمت دراستها على نطاق واسع في استخراج البيانات. تولد هذه العملية نموذجًا عن طريق تحليل الحالات المصنفة ثم يتم استغلال هذا النموذج للتنبؤ بفئات الحالات غير المسماة. يعتمد أداء النموذج الذي تم إنشاؤه على المصنف وجودة البيانات. أهداف اختيار البيانات هي تقليل حجم البيانات والقضاء على الضوضاء من أجل تحسين جودة البيانات. يوفر اختيار البيانات دقة عالية، ووقتًا أقل للتعلم، ونماذج تنبؤية أقل تعقيدًا، وتفسير البيانات المفهومة. يُعتبر التصنيف الخاضع للإشراف واختيار البيانات من مشكلات التحسين التي يصعب حلها والتي توجد لها حلول عن طريق علم الاكتشاف المتعدد. تهجين علم الاكتشاف المتعدد مع الخوارزميات الأخرى (علم الاكتشاف المتعدد، تقنية استخراج البيانات والتعلم الآلي، ...) يسمح لتطوير مرحلة ما قبل المعالجة وبعد ذلك لبناء نموذج تعليمي فعال وقوي الذي هو مساهمة هذه الأطروحة.

الكلمات المفتاحية : استخراج المعرفة من البيانات، المعالجة المسبقة للبيانات ، استخراج البيانات، التصنيف، علم الاكتشاف المتعدد، التهجين، مشاكل التحسين الصعبة.

Résumé

Notre objectif est de découvrir des modèles intelligibles et efficaces à partir de données. Le challenge est de travailler avec une grande masse de données et d'utiliser des algorithmes puissants permettant d'extraire des connaissances indispensables. La classification supervisée fait partie des tâches largement étudiées dans la fouille de données. Ce processus génère un modèle en analysant des instances étiquetées. Ensuite, ce modèle sera exploité pour prédire les classes des instances, non étiquetées. Les performances du modèle généré dépendent des classifieurs et de la qualité des données. Pour cela un bon prétraitement des données, et spécialement la sélection d'attributs, permet d'obtenir une précision élevée, une durée d'apprentissage inférieure, une complexité moindre des modèles prédictifs et une interprétation compréhensible des données. La classification supervisée et le prétraitement des données sont considérés comme des problèmes d'optimisation NP-difficile qui peuvent être résolus par les métaheuristiques. Hybridation de métaheuristiques avec d'autres algorithmes (métaheuristiques, technique de fouille de données et l'apprentissage automatique,...) permet de développer la phase de prétraitement et la tâche de la classification supervisée par la suite de construire un modèle d'apprentissage efficient et puissant qui est la contribution de cette thèse.

Mots Clés : ECD (extraction de connaissances à partir de données), prétraitement de données, sélection d'attributs, Fouille de données, classification supervisée, Métaheuristiques, Hybridation, problèmes d'optimisation difficile.

Abstract

Our goal is to discover intelligible and efficient models from data. The challenge is to work with a large amount of data and to use powerful algorithms to extract essential knowledge. Supervised classification is among the widely studied tasks in data mining. This process generates a model by analyzing labeled instances. Then, this model will be exploited to predict the classes of unlabeled instances. The performance of the generated model depends on the classifier and the quality of data. The main idea of the data selection is the reduction of the data size and the elimination of the noise, which improves the quality of the data. Feature selection provides high accuracy, lower learning time, less complexity of predictive models, and understandable data interpretation. The supervise classification and the pre-processing of the data are considered to be NP-hard optimization problems that can be solved by meta-heuristics. Hybridization of Metaheuristics with other Algorithms (Metaheuristics, data mining technique and machine learning, ...) allows to develop the preprocessing phase and afterwards to build an efficient and powerful learning model that is the contribution of this thesis.
Keywords: KDD (knowledge discovery in databases), data preprocessing, Feature selection, data mining, classification, metaheuristics, hybridization, difficult optimization problems.