

TABLE DES MATIERES

<u>INTRODUCTION GÉNÉRALE</u>	1
<u>CHAPITRE 1 : Méthodes évolutionnaires pour L'optimisation</u>	5
1.1. Introduction	5
1.2. Optimisation et informatique évolutionnaire	6
1.3. Les méthodes évolutionnaires classiques	10
1.3.1. Les algorithmes génétiques	10
1.3.2. La programmation évolutionnaire	12
1.3.3. Les stratégies d'évolution	12
1.3.4. La programmation génétique	12
1.4. L'intelligence en essaims ou Swarm Intelligence	13
1.4.1. Les colonies de fourmis	14
1.4.1.1. Le système de fourmis original	16
1.4.1.2. Les variantes de l'algorithme AS	18
1.4.2. Les essaims particulaires	21
1.4.3. Les algorithmes inspirés des abeilles (Bee inspired algorithms)	25
1.4.3.1. Les colonies d'abeilles artificielles (ABC)	25
1.4.3.2. L'Algorithme d'abeilles (BA)	27
1.4.3.3. L'algorithme d'optimisation par les colonies d'abeilles (BCO)	27
1.4.4. L'algorithme des essaims de poissons artificiels (AFSA)	28
1.4.5. L'algorithme des chauves-souris (Bat algorithm)	29
1.4.6. L'algorithme des essaims de lucioles (Firfly algorithm)	30
1.4.7. L'algorithme d'optimisation de l'exploration bactérienne (BFO)	31
1.5. Autres méthodes évolutionnaires pour l'optimisation	31
1.5.1. L'optimisation basée sur la Biogéographie	31
1.5.2. L'algorithme de recherche gravitationnelle	32
1.5.3. L'optimisation basée sur l'enseignement-apprentissage	32
1.6. Conclusion	33

<u>CHAPITRE 2 : Concepts et méthodes de la sélection d'attributs</u>	35
2.1. Introduction	36
2.2. Définition de la sélection d'attributs	37
2.3. Prosessus de sélection d'attributs	37
2.3.1. La procédure de génération	38
2.3.1.1. Génération complète	39
2.3.1.2. Génération heuristique ou séquentielle	39
2.3.1.3. Génération aléatoire	40
2.3.2. La fonction d'évaluation	40
2.3.2.1. Les mesures indépendantes	40
2.3.2.2. Les mesures dépendantes	41
2.3.3. Le critère d'arrêt	41
2.3.4. La procédure de validation	43
2.4. Catégorisation des méthodes de sélection d'attributs	43
2.4.1. Catégorisation basée sur les critères d'évaluation	43
2.4.2. Catégorisation basée sur la stratégie de recherche	45
2.5. Méthodes classiques de la sélection d'attributs	47
2.5.1. Méthodes complètes	47
2.5.1.1 FOCUS	47
2.5.1.2 Branch and Bound (BB)	48
2.5.1.3 Automatic Branch & Bound (ABB)	50
2.5.2. Méthodes heuristiques	50
2.5.2.1 Sequential Forward Selection (SFS)	51
2.5.2.2 Sequential Backward Selection (SBS)	51
2.5.2.3. Relief	52
2.5.3. Méthodes Aléatoires	53
2.5.3.1 Las Vegas Filter et Las Vegas Wrapper	53
2.5.3.2 Les algorithmes génétiques pour la sélection d'attributs	54
2.6. Méthodes de sélection d'attributs basées sur l'intelligence en essaims	55
2.6.1. Sélection d'attributs par les colonies de fourmis	55
2.6.1.1 Représentation du problème de sélection de caractéristiques avec l'ACO	56
2.6.1.2. Le processus global de la sélection d'attributs avec ACO	58
2.6.2. Sélection d'attributs par les essaims particulaires	59

2.6.2.1. Représentation de la position	60
2.6.2.2. Représentation de la vitesse	60
2.6.2.3. Les stratégies de mise à jour de la position et la vitesse	61
2.7. Applications de la sélection d'attributs	61
2.7.1. La fouille de données	62
2.7.2. La catégorisation de textes	62
2.7.3. La reconnaissance de l'écriture	63
2.7.4. La reconnaissance d'images	63
2.7.5. La bioinformatique	64
2.8. Conclusion	64
<u>CHAPITRE 3 : Approches classiques et bio-inspirées pour la sélection d'attributs</u>	65
3.1. Introduction	65
3.2. Sélection d'attributs basée sur des méthodes bio-inspées non hybrides	66
3.2.1. Sélection d'attributs avec les algorithmes génétiques	66
3.2.2. Sélection de d'attributs basée sur les ACO	68
3.2.3. Sélection de d'attributs avec les PSO	68
3.2.4. Sélection d'attributs avec les algorithmes inspirés des abeilles	69
3.2.5. Synthèse de travaux sur la sélection d'attributs avec des méthodes bio-inspirées	70
3.3. Approches classique proposée pour la sélection d'attributs	73
3.4. Approche proposée pour la sélection d'attributs basée sur ACO	75
3.5. Approche proposée pour la sélection d'attributs basée sur PSO	78
3.6. Évaluation des approches proposées	81
3.6.1. Choix expérimentaux	82
3.6.1.1. Paramètres d'ACO et PSO	82
3.6.1.2. Évaluation des performances de classification	82
3.6.2. Expérimentations dans le domaine du filtrage de spams	83
3.6.2.1. Description de la base de données	83
3.6.2.2. Résultats expérimentaux	84
3.6.3. Expérimentations des approches basées ACO et PSO avec différentes bases de données	87
3.6.3.1. Description des bases de données	87
3.6.3.2. Résultats expérimentaux	87
3.7. Conclusion	89

<u>CHAPITRE 4 : Approches bio-inspirées hybrides pour la sélection d'attributs</u>	91
4.1. Introduction	91
4.2. Synthèse d'approches hybride basées ACO ou PSO	92
4.3. Approches bio-inspirées hybrides pour la sélection d'attributs	97
4.4. Les approches hybrides ACO-PSO proposées pour la sélection d'attributs	99
4.4.1. L'approche proposée ACO-PSO1	99
4.4.2. L'approche proposée ACO-PSO2	103
4.4.3. L'approche proposée ACO-PSO3	105
4.5. Évaluation des approches hybrides proposées	106
4.5.1. Bases de données et outils utilisés	106
4.5.2 Résultats expérimentaux	106
4.6. Conclusion	110
CONCLUSION ET PERSPECTIVES	111
BIBLIOGRAPHIE	115
ANNEXE 1: Description des bases de données	131
ANNEXE 2: Les mesures utilisées pour le feature ranking dans l'approche classique proposée	141
A propos de l'auteur	145

LISTE DES FIGURES

Figure 1.1	Taxonomie des différentes méthodes d'optimisation [DAS 09]	9
Figure 1.2	Fonctionnement général d'un AG de base [JOU 03]	11
Figure 1.3	Un arbre de programme représentant une fonction mathématique	13
Figure 1.4	Expérience de sélection des branches les plus courtes par une colonie de fourmis : (a) au début de l'expérience, (b) à la fin de l'expérience.	15
Figure 1.5	Mouvement d'une particule [ALA 08]	24
Figure 2.1	Principe de la sélection de caractéristiques [GUE 06]	36
Figure 2.2	Principe de l'extraction de caractéristiques [GUE 06]	36
Figure 2.3	Processus de sélection de caractéristiques [DAS 97]	38
Figure 2.4	l'approche Filtre et l'approche Wrapper	44
Figure 2.5	Catégorisation des méthodes de sélection de caractéristiques [BEN 02]	45
Figure 2.6	Depth-first search	48
Figure 2.7	Breadth-first search	48
Figure 2.8	Représentation d'ACO pour le problème de sélection des caractéristiques	57
Figure 2.9	Processus général de la sélection de caractéristiques basée sur les ACO [KAN 07]	59
Figure 2.10	Le principe de mise à jour de la vitesse	61
Figure 3.1	L'approche classique proposée pour la sélection de d'attributs [MEN 10a, MEN 14a]	74
Figure 3.2	Processus général proposé pour la sélection d'attributs basée sur ACO [MEN 14a]	75
Figure 3.3	Processus général proposé pour la sélection de caractéristiques basée sur PSO [MEN 14a]	80

Figure 3.4	L'erreur versus nombre d'itérations pour l'approche classique 'feature ranking'	84
Figure 3.5	L'erreur de la meilleure fourmi versus le numéro d'itération	85
Figure 3.6	L'erreur de la meilleure solution trouvée dans chaque itération de dans l'approche basée sur le PSO	85
Figure 4.1	Le principe de l'approche HGAPSO proposée par Juang [JUA 04]	92
Figure 4.2	L'approche PSO+AG [VAL 08]	93
Figure 4.3	L'approche hybride proposée ACO-PSO1	101
Figure 4.4	L'approche hybride proposée ACO-PSO2	103

LISTE DES TABLEAUX

Tableau 2.1	Une comparaison entre les fonctions d'évaluation [DAS 97]	42
Tableau 2.2	Comparaison wrappers / filters	44
Tableau 2.3	La catégorisation des méthodes de sélection de caractéristiques pour la classification [LIU 05]	46
Tableau 3.1	Synthèse de travaux sur la sélection d'attributs basée sur des méthodes bio-inspirées non hybrides.	70
Tableau 3.2	Les différentes variantes de l'ACO	77
Tableau 3.3	Paramètres d'ACO	82
Tableau 3.4	Paramètres de PSO	82
Tableau 3.5	Description de la base « SpamBase »	83
Tableau 3.6	Comparaison entre les performances de différentes méthodes	86
Tableau 3.7	Description de bases de données	87
Tableau 3.8	Résultats des approches proposées de sélection d'attributs basées sur ACO et PSO avec cinq bases de données publiques	88
Tableau 4.1	Synthèse des travaux utilisant des approches hybrides basées sur ACO ou PSO	95
Tableau 4.2	Synthèse de travaux sur la sélection d'attributs basée sur une approche hybride incluant ACO ou PSO	97
Tableau 4.3	Comparaison des performances des différentes approches de sélection de caractéristiques pour la base Spambase	106
Tableau 4.4	Comparaison des performances des différentes approches de sélection de caractéristiques pour la base Breast Cancer	107
Tableau 4.5	Comparaison des performances des différentes approches de sélection de caractéristiques pour la base Germen	107
Tableau 4.6	Comparaison des performances des différentes approches de sélection de caractéristiques pour la base Hepatitis	107

Tableau 4.7	Comparaison des performances des différentes approches de sélection de caractéristiques pour la base Liver	108
Tableau 4.8	Comparaison des performances des différentes approches de sélection de caractéristiques pour la base Musk	108

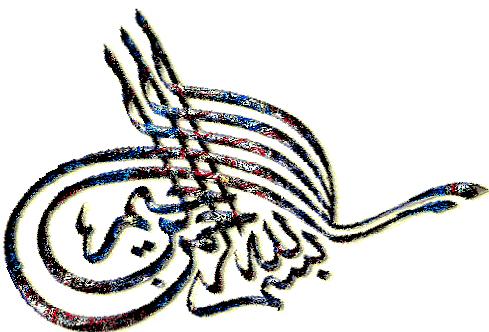
LISTE DES ALGORITHMES

Algorithme 1.1	Procédure général de l’Informatique Évolutionnaire [DAS 09]	9
Algorithme 1.2	Algorithme de colonies de fourmis de base : Ant System	18
Algorithme 1.3	Pseudo-code de l’algorithme de PSO [ALA 08]	24
Algorithme 1.4	L’algorithme ABC [DIW 11]	26
Algorithme 1.5	L’algorithme BA [DIW 11]	27
Algorithme 1.6	L’algorithme de BCO [DIW 11]	28
Algorithme 2.1	L’algorithme FOCUS [CAK 02]	47
Algorithme 2.2	L’algorithme Branch and Bound (BB)[CAK 02]	49
Algorithme 2.3	L’algorithme Automatic Branch and Bound (ABB) [BEL 11]	50
Algorithme 2.4	L’algorithme SFS [LAD 11]	51
Algorithme 2.5	L’algorithme SBS [LAD 11]	52
Algorithme 2.6	Algorithme Relief [BEL 11]	53
Algorithme 2.7	Algorithme LVF [BEL 11]	54

ABREVIATIONS

ABB	Automatic Branch & Bound
ABC	Artificial Bee colony
ACO	Ant Colony Optimization
ACS	Ant colony System
AE	Algorithme Évolutionnaire
AFSA	Artificial Fish Swarm Algorithm
AGs	Algorithmes Génétiques
ANN	Artificial Neural Networks
AS	Ant System
ASrank	Rank based Ant System
BA	Bee Algorithm
BB	Branch and bound
BBO	Biogeography-Based Optimization
BCO	Bee Colony Optimization
BER	Balanced Error Rate
BFO	Bacterial Foraging Optimization
BPNN	Back Propagation Neural Network
BPSO	Binary Particle Swarm Optimization
Chi2	Le test du χ^2 ,
DF	Document Frequency
DPSO	Discrete Particle Swarm Optimization

FFA	Firefly algorithm
FS	Feature Selection
Gbest	La meilleure position rencontrée par l'essaim dans l'algorithme PSO
GSA	Gravitational Search Algorithm
HGA	Hybrid Genetic Algorithm
HGAFS	Hybrid Genetic Algorithm for Feature Selection
HGAPSO	Hybrid Genetic Algorithm and Particle Swarm Optimization
IBPSO	Improved Binary Particle Swarm Optimization
IG	Information Gain
LVF	Las Vegas Filter
LVW	Las Vegas Wrapper
MI	Mutual Information
MLP	Multi-Layer Perceptron
MMAS	Max–Min Ant System
Pbest	La meilleure position rencontrée par la particule dans l'algorithme de PSO
PSO	Particle Swarm Optimization
Relief	une méthode de sélection d'attributs basée sur l'évaluation individuelle de chacun des attributs
SBS	Sequential Backward Selection
SFS	Sequential Forward Selection
Sn2	Signal-to-noise ratio coefficient
SVM	Support Vector Machine
TLBO	Teaching Learning Based Optimization
TSP	Travelling Salesman Problem



INTRODUCTION GÉNÉRALE

L'apprentissage automatique fait référence au développement, à l'analyse et à l'implémentation de méthodes qui permettent à une machine d'évoluer, dans la résolution d'une catégorie de problèmes, grâce à un processus d'apprentissage. Le but de l'apprentissage automatique, et en particulier la classification, est de résoudre automatiquement des problèmes complexes par la prise de décisions sur la base des échantillons de ces problèmes.

Un système de classification permet d'extraire les points communs d'un ensemble d'objets en formant des classes qui partagent des caractéristiques similaires. La complexité de cette tâche s'est fortement développée ces deux dernières décennies lorsque les masses de données disponibles ont vu leur volume exploser. En effet, non seulement le nombre des échantillons dans les bases de données a fortement augmenté, mais également la taille de leur description. La représentation des échantillons permet de convertir les données réelles (mesures physiques, réponse à un stimulus,...) dans un format propre à leur utilisation, afin de ressortir des descripteurs pour représenter ces données. Cependant, les descripteurs fournissent souvent des données de grande dimension et la classification de telles données est un problème difficile.

La réduction de la dimensionnalité via l'extraction et la sélection d'attributs est une étape fondamentale dans le traitement des données qui peut influer considérablement sur la performance du système de classification.

Il existe une différence entre les techniques d'extraction (Feature Extraction) et celles de sélection (Feature Selection) d'attributs. Les techniques d'extraction permettent de créer de nouveaux ensembles de caractéristiques, en utilisant une transformation ou une combinaison d'un espace de départ et en effectuant une réduction du nombre de dimensions. A titre d'exemple, l'analyse en composantes principales (ACP), qui ne prend en compte que les dépendances linéaires entre les caractéristiques, est très souvent utilisée.

Les techniques de sélection, permettent de choisir les caractéristiques, variables ou mesures les plus intéressantes, pertinentes ou informatives dans un système donné, afin de réaliser la tâche pour laquelle il a été conçu.

La sélection d'attributs ou de caractéristiques (Attribute, Feature Selection ou FS) est l'une des étapes les plus importantes dans le traitement de données, c'est une procédure permettant de choisir un sous-ensemble minimum de M caractéristiques à partir de l'ensemble original de N caractéristiques ($M \leq N$), de sorte que l'espace de caractéristiques soit réduit de façon optimale selon certains critères d'évaluation. La sélection de caractéristiques peut être considérée comme un problème d'optimisation avec de nombreux critères concurrents, qui peuvent être, par exemple : réduire au minimum le nombre de caractéristiques sélectionnées, minimiser l'erreur de classification, ... etc.

Le problème de la sélection des caractéristiques est, depuis longtemps, un sujet de recherche actif. Les premières approches de sélection de caractéristiques s'appuient sur des méthodes classiques et des algorithmes de recherche tels que la programmation dynamique et le Branch and Bound. Récemment, la tendance de la recherche dans le domaine de la sélection de caractéristiques est orientée vers les méthodes bio-inspirées, en particulier celles issues de l'intelligence en essaim ou swarm intelligence. Ces méthodes ont suscité un grand intérêt dans les deux dernières décennies, et de nombreux algorithmes d'optimisation basés sur l'inspiration biologique ont gagné une énorme popularité.

Dans le cadre de cette thèse, nous nous intéressons au problème de la sélection d'attributs pour la classification, et plus précisément, à l'utilisation d'approches bio-inspirées pour effectuer la sélection de caractéristiques dans un contexte de classification.

Ainsi, après avoir étudié les concepts et méthodes de sélection d'attributs, nous avons commencé, dans une optique, comparative, par la proposition d'une approche basée sur des méthodes classiques de sélection d'attributs. Cette approche a été testée dans deux domaines applicatifs différents : le filtrage de spams [MEN 10a] et la reconnaissance de chiffres manuscrits [MEN 10b]. D'autre part, nous nous sommes intéressés aux méthodes bio-inspirées évolutionnaires non hybrides utilisées pour la résolution de problèmes d'optimisation. Les méthodes d'optimisation issues de la swarm intelligence

ont particulièrement attiré notre attention et nous nous sommes concentrés sur celles basées sur les colonies de fourmis (ACO) [MEN 11a, MEN 11b] et sur les essaims particulaires (PSO). Nous avons testé et comparé ces approches pour la sélection d'attributs dans le domaine applicatif du filtrage de spams [MEN 14].

L'objectif final de cette thèse est de proposer une approche de sélection d'attributs basée sur une hybridation entre des méthodes bio-inspirées. Dans ce cadre, selon notre étude de la littérature du domaine, nous avons constaté la rareté des approches basées sur une hybridation de l'algorithme des colonies de fourmis (Ant Colony Optimization ou ACO) et de l'algorithme d'optimisation par les essaims de particules (Particle Swarm Optimization ou PSO) pour la sélection des caractéristiques. Dans nos travaux, nous avons proposé [MEN 15] deux approches basées sur l'hybridation des algorithmes ACO et PSO (ACO-PSO1 et ACO-PSO2) pour la sélection hybride de caractéristiques, et nous avons aussi proposé l'approche ACO-PSO3 qui est une adaptation, au problème de sélection de caractéristiques, de l'algorithme hybride proposé initialement par Shuang et al [SHU 11] pour résoudre le problème du voyageur de commerce.

L'idée de base la première approche (ACO-PSO1) est d'utiliser l'ACO et PSO simultanément pour explorer l'espace de recherche. Dans la deuxième approche, (ACO-PSO2), l'exploration de l'espace de recherche commence par les mécanismes d'ACO et lorsque les performances ne sont pas améliorées, après certain nombres d'itérations, les mécanismes de PSO sont utilisés pour continuer l'exploration de l'espace de recherche.

Afin d'évaluer les approches hybrides proposées, nous avons utilisé six bases de données publiques et nous avons comparé les résultats obtenus par les approches hybrides avec ceux des deux approches bio-inspirées non hybrides proposées précédemment, l'une basée sur l'optimisation par les colonies de fourmis, et l'autre sur les essaims de particules [MEN 15].

La thèse est organisée de la manière suivante afin de refléter la synthèse bibliographique effectuée et les propositions que nous avons émises suite à cette étude :

Le premier chapitre introduit le problème d'optimisation, en se focalisant sur les méthodes de l'informatique évolutionnaire largement appliquées dans ce domaine. Nous présentons, dans ce chapitre, des méthodes évolutionnaires classiques telles que les algorithmes génétiques puis des méthodes issues de l'intelligence en essaims ou swarm intelligence telles que les colonies de fourmis et les essaims particulaires, et

finalement d'autres méthodes évolutionnaires récentes, développées dans les années 2000.

Le deuxième chapitre définit la sélection d'attributs et donne un aperçu sur ses concepts de base. Il décrit également les différentes méthodes de sélection de caractéristiques en commençant par les méthodes classiques. Une grande partie de ce chapitre est consacrée aux méthodes basées sur les méthodes de l'intelligence en essaims (swarm intelligence).

Le troisième chapitre commence par présenter une synthèse de travaux relatifs aux approches bio-inspirées de sélection de caractéristiques. Nous y présentons et évaluons, par la suite, l'approche classique et les approches bio-inspirées non hybrides proposées pour la sélection d'attributs dans le filtrage de spams.

Le dernier chapitre est consacré aux méthodes bio-inspirées hybrides. Nous y synthétisons les travaux existants dans la littérature concernant les méthodes bio-inspirées hybrides et leurs applications, notamment dans le domaine de la sélection d'attributs. Une grande partie de ce chapitre est consacrée à la présentation des approches hybrides proposées et à une comparaison de leurs performances à celles des approches bio-inspirées non hybrides proposées antérieurement en utilisant plusieurs bases de données publiques.

CHAPITRE 1

METHODES EVOLUTIONNAIRES POUR L'OPTIMISATION

Ce chapitre présente principalement des méthodes bio-inspirées évolutionnaires pour l'optimisation. Nous commençons par un bref aperçu sur le problème d'optimisation et la notion d'informatique évolutionnaire (evolutionary computing). Le reste du chapitre est consacré à une description de certaines méthodes de l'informatique évolutionnaire, largement utilisées dans les problèmes d'optimisation. Nous y présentons des méthodes évolutionnaires classiques telles que les algorithmes génétiques puis des méthodes issues de l'intelligence en essaims ou swarm intelligence telles que les colonies de fourmis et les essaims particulaires, et finalement d'autres méthodes évolutionnaires récentes développées dans les années 2000.

1.1. Introduction

L'optimisation est un problème très ancien qui a donné lieu à de nombreuses méthodes de calcul. Un grand nombre de problèmes peuvent, en effet, être décrits sous la forme de problèmes d'optimisation. C'est le cas, par exemple, pour les problèmes d'identification, l'apprentissage supervisé de réseaux de neurones ou encore la recherche du plus court chemin.

Les algorithmes de l'informatique évolutionnaire sont des méthodes d'optimisation efficaces, connus pour leur robustesse et leur capacité à trouver de bonnes solutions dans un vaste espace de recherche. Ils sont inspirés du processus d'adaptation biologique des êtres vivants à leur environnement. L'analogie entre un problème d'optimisation et ce phénomène biologique a été formalisée par plusieurs approches, à commencer par les algorithmes génétiques. Dans la suite de cette section, nous allons introduire le problème d'optimisation et donner un aperçu sur quelques méthodes de l'informatique évolutionnaire.

1.2. Optimisation et informatique évolutionnaire

L'optimisation est un aspect fondamental de l'ingénierie et de la résolution de problèmes. L'objectif de l'optimisation est de chercher les valeurs d'un ensemble de paramètres pour maximiser ou minimiser les fonctions objectives soumises à certaines contraintes. Un choix des valeurs, pour l'ensemble de paramètres, qui satisfont toutes les contraintes, est appelé une solution faisable. Les solutions faisables avec des valeurs de la fonction objective qui sont meilleures que les valeurs de toutes les autres solutions possibles, sont appelées les solutions optimales [OMR 04].

Un problème d'optimisation peut être formulé sous la forme d'un problème de minimisation ou d'un problème de maximisation. Parfois, nous essayons de minimiser une fonction et, parfois, nous essayons de maximiser une fonction. Ces deux problèmes sont facilement convertis l'un à l'autre [SIM 13a]:

$$\begin{aligned} \min_x f(x) &\Leftrightarrow \max_x [-f(x)] \\ \max_x f(x) &\Leftrightarrow \min_x [-f(x)] \end{aligned} \tag{1.1}$$

La fonction $f(x)$ est appelée la fonction objective, et le vecteur x est appelé la variable indépendante, ou la variable de décision. Notons, qu'en fonction du contexte, les termes 'variable indépendante' et 'variable de décision' se réfèrent parfois à l'ensemble du vecteur x , et se réfèrent parfois à des éléments spécifiques dans le vecteur x . Les éléments de x sont aussi appelés les attributs de la solution. Le nombre des éléments de x est appelée la dimension du problème. Comme le montre l'équation (1.1), tout algorithme qui est conçu pour minimiser une fonction peut facilement être utilisé pour maximiser une fonction, et tout algorithme qui est conçu pour maximiser une fonction peut être facilement utilisé pour minimiser une fonction.

- Quand nous essayons de minimiser une fonction, nous appelons la valeur de la fonction : la fonction de coût.
- Quand nous essayons de maximiser une fonction, nous appelons la valeur de la fonction : la fitness.

$\min_x f(x) \Rightarrow f(x)$ est appelée "le coût" ou "l'objectif"

$\max_x f(x) \Rightarrow f(x)$ est appelée "la fitness" ou "l'objectif"

Le problème de minimisation peut être défini comme suit :

Soit $f: S \rightarrow \mathbb{R}$ où $f \subseteq \mathbb{R}^{N_d}$ et N_d est la dimension de l'espace de la recherche S

Trouver $x^* \in S$ sachant que $f(x^*) \leq f(x), \forall x \in S$.

La variable x^* est appelée le minimum global (ou simplement le minimiseur) de f et $f(x^*)$ est appelée la valeur minimale globale (ou simplement le minimum) de f . Le processus qui consiste à trouver la solution optimale globale est connu comme étant l'optimisation globale [OMR 04].

Il existe certains types particuliers de problèmes d'optimisation:

- Les problèmes généraux sans contraintes: où une fonction non linéaire est définie sur un ensemble de valeurs réelles sans contraintes.
- Les problèmes généraux avec contraintes: où une fonction non linéaire est définie sur un ensemble limité de valeurs réelles. Généralement, les problèmes d'optimisation sont des problèmes d'optimisation avec contraintes [SIM 13a].
- Les problèmes d'optimisation multi-objectifs : dans lesquels un problème nécessite la résolution de plusieurs problèmes simultanément. Souvent, les solutions aux divers problèmes interfèrent entre elles, la meilleure solution est alors une sorte de compromis [KEN 06]. L'optimisation multi-objectif (Multiobjective Optimization : MO) cherche à optimiser les composantes de valeurs de vecteur d'une fonction de coût. Contrairement à l'optimisation avec objectif unique, la solution à ce problème n'est pas un seul point, mais une famille de points connus comme l'ensemble Pareto-optimal. Chaque point de cette surface est optimal dans le sens qu'aucune amélioration ne peut être obtenue en un composant de vecteur de coût. L'ensemble des solutions est appelé l'ensemble de Pareto [FON 93].
- Les problèmes d'optimisation multimodale: ce sont ceux dans lesquels l'espace de recherche contient plusieurs optimums locaux et il est possible qu'il contienne plus d'un optimum global. Ces problèmes sont intéressants, non seulement en raison du défi qu'ils représentent, en évitant les optimums locaux ou la localisation de plus d'un optimum global en même temps, mais parce qu'il existe beaucoup de problèmes du monde réel présentant ces caractéristiques [BAR 11].

- Les problèmes combinatoires: il existe de nombreux problèmes d'optimisation pour lesquels les variables indépendantes sont limitées à un ensemble de valeurs discrètes. Ces problèmes sont appelés problèmes d'optimisation combinatoire [SIM 13a].

La littérature relative à l'optimisation est très abondante et revêt plusieurs formes différentes. Das et al [DAS 09] divisent les méthodes d'optimisation en plusieurs classes et placent celles-ci dans un arbre (Figure 1.1) selon que les méthodes soient exactes ou approximatives, déterministes ou probabilistes, traitent des solutions complètes ou construisent des solutions lors de la recherche, et enfin si elles entretiennent une solution candidate unique ou une population des solutions. La branche gauche de la Figure 1.1 contient les méthodes traditionnelles. Elles donnent les solutions exactes, sauf pour les techniques de recherche locale, telles que la méthode de Newton et les méthodes de gradient, qui offrent, comme les heuristiques modernes, des solutions approximatives. Les méthodes d'analyse directe peuvent être arrêtées à tout moment et renvoient une solution, car elles traitent des solutions complètes, alors que 'Branch and Bound', la programmation dynamique (dynamic programming), et 'Divide and Conquer' construisent toutes des solutions lors de la recherche, et ne peuvent donc pas être arrêtées pour retourner une solution complète avant que toute la recherche ne soit effectuée.

Dans le cadre de notre travail, nous nous intéressons particulièrement à l'informatique évolutionnaire ou Evolutionary Computing (branche à l'extrême droite de la Figure 1.1) qui est un paradigme récent de l'intelligence informatique, résultant d'une observation attentive de la relation sous-jacente entre l'optimisation et l'évolution biologique.

L'informatique évolutionnaire comprend tous les algorithmes basés sur la notion de population (P) et qui utilisent la sélection et la recombinaison pour générer de nouveaux points dans l'espace de recherche.

La procédure générale de l'informatique évolutionnaire (voir Algorithme 1.1) consiste à sélectionner une nouvelle population P, la modifier et l'évaluer jusqu'à ce qu'une condition d'arrêt soit atteinte.

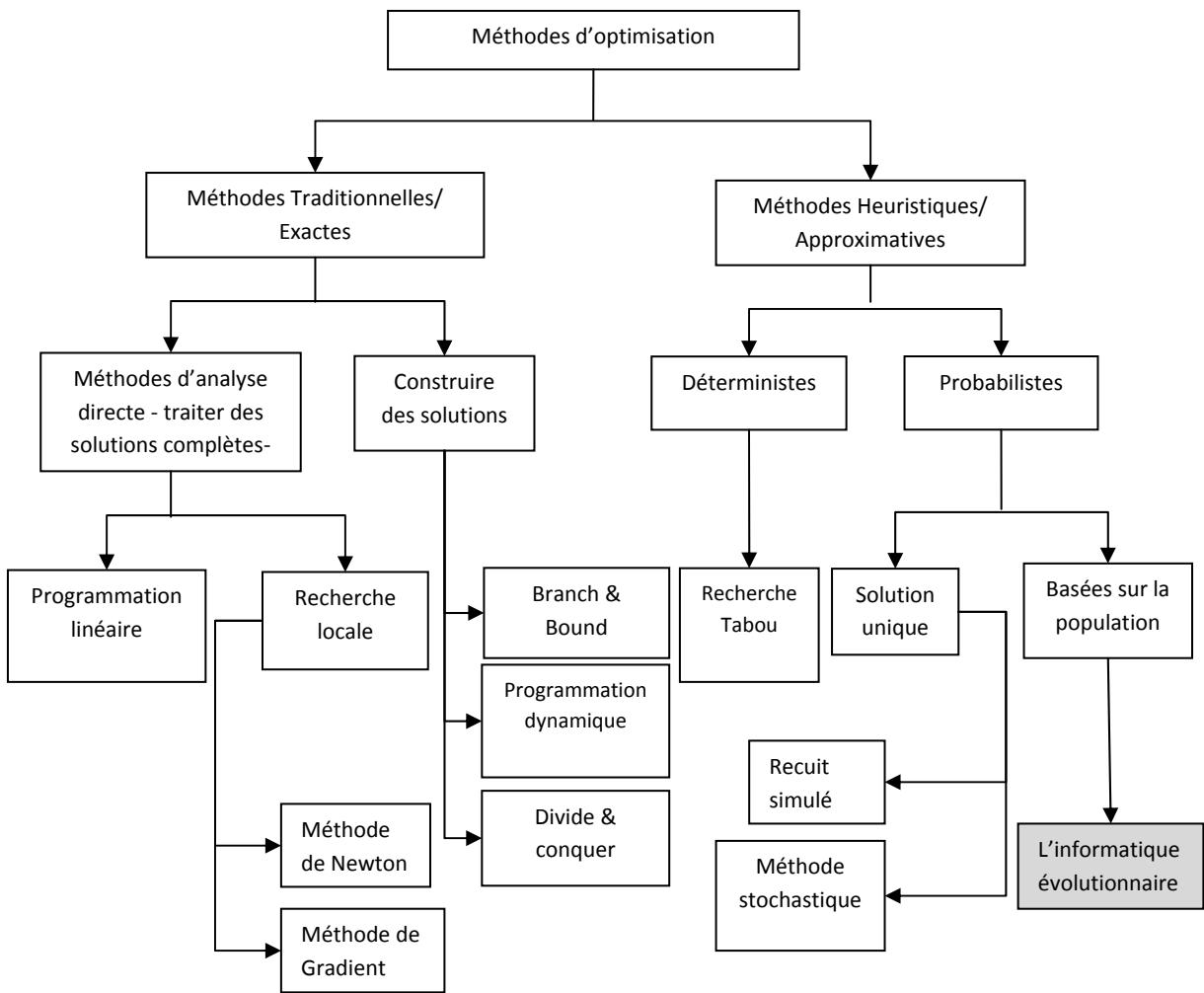


Figure 1.1. Taxonomie des différentes méthodes d'optimisation [DAS 09]

Début

$t \leftarrow 0;$

Initialiser $P(t)$;

Évaluer $P(t)$;

Tant que (la condition d'arrêt n'est pas vérifiée) **Faire**

Début

$t \leftarrow t+1;$

Sélectionner $P(t)$ de $P(t-1)$;

Modifier $P(t)$;

Évaluer $P(t)$;

Fin Tant que;

Fin.

Algorithme 1.1. Procédure générale de l'Informatique Évolutionnaire [DAS 09]

Ainsi l'informatique évolutionnaire inclut les algorithmes évolutionnaires classiques tels que les algorithmes génétiques, la programmation évolutionnaire, les stratégies d'évolution et la programmation génétique, ainsi que les méthodes de l'intelligence en essaims en plus d'autres méthodes d'optimisation basées sur la biogéographie, l'algorithme de recherche gravitationnelle [RAS 09] ou l'optimisation basée sur l'enseignement-apprentissage [RAO 11].

1.3. Les méthodes évolutionnaires classiques

Les méthodes évolutionnaires représentent une classe de métaheuristiques ayant comme principe de faire évoluer une population de base (constituée de solutions au problème à résoudre) par des opérateurs de variations (croisement et mutation). Cette population est soumise à une sélection pour le croisement puis une sélection pour la survie. L'objectif est de faire converger les solutions (appelées « individus ») vers un optimum en respectant certains critères.

Dans les années 70, les premiers travaux sur l'évolution artificielle ont concerné les algorithmes génétiques (AG), les stratégies d'évolution (SE) et la programmation évolutive (PE). Ces trois types d'algorithmes ont utilisé des principes globalement communs car ils se sont tous inspirés des mêmes principes du néo-darwinisme : utilisation d'une population d'individus, évaluation des individus par une fonction, sélection des meilleurs et génération d'une nouvelle population avec des opérateurs de croisement et de mutation. Ensuite, dans les années 90 est apparue la programmation génétique (PG) qui introduit, notamment, des représentations arborescentes [AZZ 04].

1.3.1. Les algorithmes génétiques

Les algorithmes génétiques (Genetic Algorithms) sont des stratégies d'adaptation et des techniques d'optimisation globale [BRO 11]. Ce sont les premiers, les plus connus, et les plus utilisés parmi les méthodes évolutionnaires [SIM 13a]. Les algorithmes génétiques (AGs) ont été développés à l'origine dans les années 60 à l'Université du Michigan par John Holland et son équipe [HOL 75], qui ont mené leur recherche sur des systèmes adaptatifs et robustes. Plus tard, les algorithmes génétiques ont été affinés par De Yong, Goldberg, Michalewicz, et bien d'autres [SER 06]. Ils sont été utilisés au

début avec des représentations binaires, où les opérateurs de croisement et de mutation jouent un rôle majeur.

Les algorithmes génétiques ont trois applications majeures nommées la recherche intelligente, l'optimisation et l'apprentissage automatique.

Un AG a généralement quatre composants : une population d'individus, où chaque individu de la population représente une solution candidate au problème d'optimisation, une fonction de fitness qui est une fonction d'évaluation par laquelle nous pouvons dire si un individu est une bonne solution ou non, une fonction de sélection qui décide comment choisir les bons individus de la population actuelle pour la création de la génération suivante; et des opérateurs génétiques tels que le croisement et la mutation, qui explorent de nouvelles régions de l'espace de recherche tout en gardant une partie de l'information actuelle.

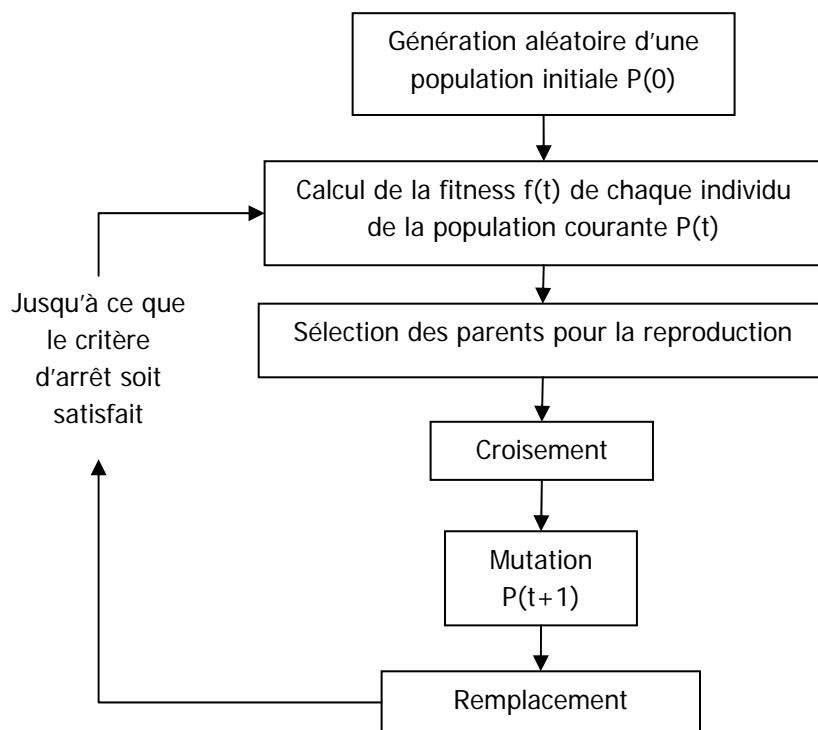


Figure 1.2. Fonctionnement général d'un AG de base [JOU 03]

La Figure 1.2 montre le fonctionnement itératif simplifié d'un algorithme génétique qui fonctionne grâce à un cycle simple d'étapes [SWA 09]:

- i) Création d'une «population» d'individus,
- ii) L'évaluation de chaque individu,

- iii) La sélection des meilleures individus et
- iv) La manipulation génétique pour créer une nouvelle population d'individus.

1.3.2. La programmation évolutionnaire

La programmation évolutionnaire (Evolutionary Programming) est une autre technique évolutive développée par Fogel et son équipe dans les années 60 [FOG 66]. L'objectif de l'algorithme de programmation évolutionnaire est de maximiser l'adéquation d'une collection de solutions candidates dans le cadre d'une fonction objective à partir du domaine. Cet objectif est poursuivi par l'utilisation d'un modèle adaptatif avec substituts pour le processus d'évolution, en particulier héréditaire (reproduction avec des variations) sous compétition. La représentation utilisée pour les solutions candidates est évaluable directement par un coût ou une fonction objective à partir du domaine [BRO 11].

1.3.3. Les stratégies d'évolution

Les stratégies d'évolution (Evolution Strategies) ont été développées indépendamment par Rechenberg et Schwefel [REC 73] en Allemagne comme une méthode pour résoudre des problèmes d'optimisation pratiques [SER 06]. L'objectif des algorithmes des stratégies d'évolution est de maximiser l'aptitude d'une collection de solutions candidates dans le cadre d'une fonction objective à partir d'un domaine. L'objectif est classiquement atteint grâce à l'adoption de la variation dynamique ; un substitut pour la descendance avec modification ; lorsque le montant de la variation a été adapté de manière dynamique avec des heuristiques basées sur la performance. Les approches contemporaines co-adaptent des paramètres qui contrôlent la quantité et la partialité de variation avec les solutions candidates [BRO 11].

1.3.4. La programmation génétique

La programmation génétique (Genetic Programming) est une technique d'optimisation évolutionnaire proposé par Koza en 1992 [SER 06]. Techniquement, l'algorithme de programmation génétique est une extension de l'algorithme génétique. L'objectif de l'algorithme de programmation génétique consiste à utiliser l'induction pour élaborer un programme d'ordinateur. Ce résultat est obtenu en utilisant des opérateurs évolutifs sur

les programmes candidats avec une structure d'arborescence afin d'améliorer l'ajustement adaptatif entre la population de programmes candidats et une fonction objective. Une évaluation d'une solution candidate implique son exécution [BRO 11].

La programmation génétique fait évoluer des programmes informatiques, traditionnellement représentés dans la mémoire avec des structures arborescentes (similaires à celle représenté dans la Figure 1.3). Les arbres peuvent être facilement évalués de manière récursive. Chaque nœud de l'arbre possède une fonction d'opérateur et chaque nœud terminal possède un opérande, ce qui rend les expressions mathématiques faciles à évoluer et à évaluer [SWA 09].

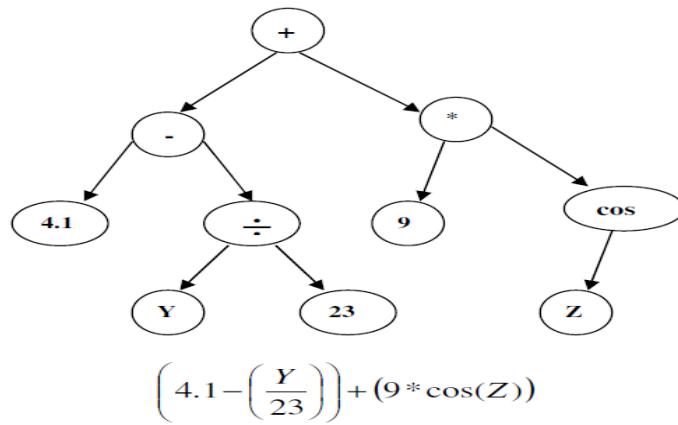


Figure 1.3. Un arbre de programme représentant une fonction mathématique.

1.4. L'intelligence en essaims ou Swarm Intelligence

Certains chercheurs séparent l'intelligence en essaims de l'informatique évolutionnaire. Un algorithme d'intelligence en essaims est fondé sur le comportement des essaims qui existent dans la nature (par exemple, des essaims d'insectes ou d'oiseaux). L'optimisation par colonies de fourmis et par les essaims de particules sont les premiers algorithmes de l'intelligence en essaims, et certains chercheurs insistent sur le fait qu'ils ne doivent pas être considérés comme des AEs. Cependant, d'autres auteurs considèrent l'intelligence en essaims comme étant un sous-ensemble de l'informatique évolutionnaire. Par exemple, l'un des pionniers de l'optimisation par les essaims de particules s'y réfère comme un algorithme évolutionnaire [SHI 99].

L'expression «intelligence en essaims» a été présentée par Beni et Wang à la fin des années 80 dans le contexte des systèmes robotiques cellulaires, où de nombreux robots

simples sont auto-organisés par des interactions avec leurs voisins les plus proches [KOR 10]. L'intelligence en essaims étudie le comportement collectif des systèmes composés de nombreux individus qui interagissent localement entre eux et avec leur environnement. Les essaims intrinsèques utilisent les formes de contrôle décentralisé et l'auto-organisation pour atteindre leurs objectifs. Les chercheurs en informatique ont développé des systèmes basés sur les essaims, suite à la réussite et l'efficacité observée de différents essaims pour résoudre des problèmes difficiles dans la nature. Dans ces essaims biologiques, les individus (fourmis, abeilles, termites, oiseaux ou poissons) ne sont pas des ingénieurs complets, mais ce sont des créatures simples avec des capacités cognitives limitées et des moyens limités de communication. Pourtant, l'essaim complet présente un comportement intelligent, en fournissant des solutions efficaces pour les problèmes complexes tels que l'évasion des prédateurs et la recherche de plus court chemin [VAN 14]. L'intelligence en essaims est basée sur l'étude du comportement collectif dans des systèmes auto-organisés décentralisées.

Selon Millonas [MIL 94], il ya cinq principes de base de l'intelligence en essaims.

- (i) la proximité: la capacité d'effectuer des calculs simples (temps et espace) de manière à répondre aux stimuli de l'environnement;
- (ii) la qualité: la capacité de réagir à des facteurs de qualité, par exemple la nourriture et la sécurité;
- (iii) la réaction: la capacité de répartir les ressources et de se protéger contre les modifications de l'environnement;
- (iv) la stabilité: la capacité de maintenir le comportement de groupe contre toute variation de l'environnement;
- (v) l'adaptabilité: la possibilité de modifier le comportement de groupe qui conduit à une meilleure adaptation à l'environnement. A ce titre, il est nécessaire de trouver un équilibre entre les principes de stabilité et l'adaptabilité [LIM 09].

1.4. 1. Les colonies de fourmis

Les algorithmes de colonies de fourmis ont vu le jour suite à un constat : les insectes sociaux en général, et les fourmis en particulier, résolvent naturellement des problèmes relativement complexes. Les biologistes ont étudié comment les fourmis arrivent à

résoudre collectivement des problèmes trop complexes pour un seul individu, notamment les problèmes de choix lors de l'exploitation de sources de nourriture.

Les fourmis ont la particularité d'employer, pour communiquer, des substances volatiles appelées phéromones. Elles sont attirées par ces substances, qu'elles perçoivent grâce à des récepteurs situés dans leurs antennes. Ces substances sont nombreuses et varient selon les espèces. Les fourmis peuvent déposer des phéromones au sol, grâce à une glande située dans leur abdomen, et former ainsi des pistes odorantes, qui pourront être suivies par leurs congénères.

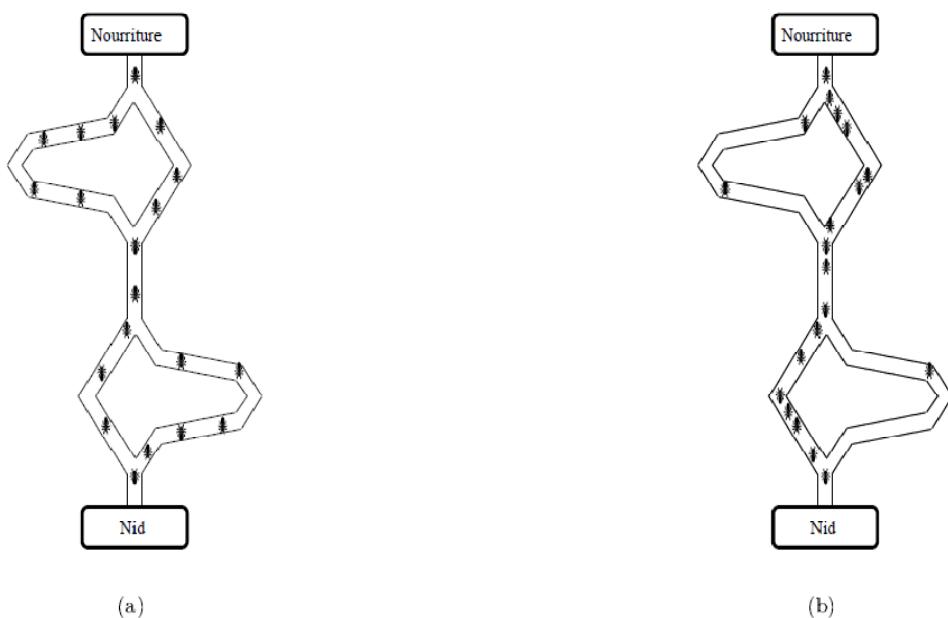


Figure 1.4. Expérience de sélection des branches les plus courtes par une colonie de fourmis : (a) au début de l'expérience, (b) à la fin de l'expérience.

Les fourmis utilisent les pistes de phéromone pour marquer leur trajet, par exemple entre le nid et une source de nourriture. Une colonie est ainsi capable de choisir (sous certaines conditions) le plus court chemin vers une source à exploiter, sans que les individus aient une vision globale du trajet.

En effet, comme l'illustre la Figure 1.4, les fourmis les plus rapidement arrivées au nid, après avoir visité la source de nourriture, sont celles qui empruntent les branches les plus courtes. Ainsi, la quantité de phéromone présente sur le plus court trajet est légèrement plus importante que celle présente sur le chemin le plus long. Or, une piste présentant une plus grande concentration en phéromone est plus attrayante pour les

fourmis, elle a une probabilité plus grande d'être empruntée. La piste courte va alors être plus renforcée que la longue, et sera choisie par la grande majorité des fourmis.

1.4.1.1. *Le système de fourmis original ou Ant System (AS)*

Le système de fourmis « Ant system » (AS) est le premier algorithme d'optimisation par les colonies de fourmis (ACO) proposé dans la littérature, il a été développé dans les années 90 par Dorigo et al. [DOR 97]. Il a été conçu pour le problème du voyageur de commerce (Travelling Salesman Problem : TSP).

Pour un ensemble de villes M , le problème TSP consiste à trouver le plus court chemin avec un circuit fermé en visitant chaque ville seulement une fois. En d'autres termes, trouver le plus court chemin pour visiter chaque ville une fois, en revenant, à la fin, à la ville de départ. Dans le cas du TSP euclidien, on considère que la longueur du chemin entre deux villes données i et j est la distance euclidienne entre i et j :

$$d_{ij} = \sqrt{[(x_i - x_j)^2 + (y_i - y_j)^2]} \quad (1.2)$$

Chaque fourmi dans le système dispose les caractéristiques suivantes [MUL 09]:

- Une fourmi décide à quelle ville elle doit aller à l'aide d'une règle de transition qui est une fonction de la distance entre la ville actuelle et la ville suivante, et de la quantité actuelle de phéromone existant le long du chemin reliant les villes.
- Les transitions aux villes déjà visitées sont ajoutées à une liste taboue et ne sont pas autorisées.
- Une fois le tour terminé, la fourmi aura déposé une piste de phéromone le long de chaque chemin visité dans le tour.

L'algorithme AS

Dans l'algorithme AS, à chaque itération t ($1 \leq t \leq t_{\max}$), chaque fourmi k ($k=1, \dots, m$) parcourt le graphe et construit un trajet complet de $n = |N|$ étapes (on note $|N|$ le cardinal de l'ensemble N). Pour chaque fourmi, le trajet entre une ville i et une ville j dépend de:

1. La liste des villes déjà visitées, qui définit les mouvements possibles à chaque pas, quand la fourmi k est sur la ville i : J_i^k ;

2. L'inverse de la distance entre les villes : $\eta_{ij} = \frac{1}{d_{ij}}$, appelé visibilité. Cette information statistique est utilisée pour diriger le choix des fourmis vers des villes proches, et éviter les villes trop lointaines ;
3. La quantité de phéromone déposée sur l'arête reliant les deux villes, appelée l'intensité de la piste. Ce paramètre définit l'attractivité d'une partie du trajet global et change à chaque passage d'une fourmi. C'est, en quelque sorte, une mémoire globale du système, qui évolue par apprentissage.

La règle de déplacement, appelée « règle aléatoire de transition proportionnelle », est la suivante :

$$P_{ij}^k(t) = \begin{cases} \frac{(\tau_{ij}(t))^\alpha \cdot (\eta_{ij})^\beta}{\sum_{l \in J_i^k} (\tau_{il}(t))^\alpha \cdot (\eta_{ij})^\beta} & \text{si } j \in J_i^k \\ 0 & \text{si } j \notin J_i^k \end{cases} \quad (1.3)$$

Où α et β sont deux paramètres contrôlant l'importance relative de l'intensité de la piste $\tau_{ij}(t)$, et de la visibilité η_{ij} . Avec $\alpha = 0$, seule la visibilité de la ville est prise en compte: la ville la plus proche est donc choisie à chaque pas. Au contraire, avec $\beta = 0$, ce sont les pistes de phéromone qui sont prises en compte. Pour éviter une sélection trop rapide d'un trajet, un compromis entre ces deux paramètres est nécessaire, en jouant sur les comportements de diversification et d'intensification.

Après un tour complet, chaque fourmi laisse une certaine quantité de phéromones $\Delta_{ij}^k(t)$ sur l'ensemble de son parcours, quantité qui dépend de la qualité de la solution trouvée :

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L^k(t)} & \text{si } (i, j) \in T^k(t) \\ 0 & \text{si } (i, j) \notin T^k(t) \end{cases} \quad (1.4)$$

Où $T^k(t)$ est le trajet effectué par la fourmi k à l'itération t , $L^k(t)$ la longueur du tour et Q un paramètre fixé.

L'algorithme ne serait pas complet sans le processus d'évaporation des pistes de phéromone. En effet, pour éviter d'être piégé dans des solutions sous-optimales, il est nécessaire de permettre au système « d'oublier » les mauvaises solutions. On contrebalance ainsi l'additivité des pistes par une décroissance constante des valeurs des arêtes à chaque itération. La règle de mise à jour des pistes est donc :

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (1.5)$$

Où $\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t)$ et m est le nombre de fourmis. La quantité initiale de phéromone sur les arêtes est une distribution uniforme d'une petite quantité $\tau_0 \geq 0$.

Pour $t = 1, \dots, t_{\max}$

Pour chaque fourmi $k = 1, \dots, m$

Choisir une ville au hasard

Pour chaque ville non visitée i

Choisir une ville j , dans la liste J_i^k des villes restantes,
selon la formule 1.3

Fin pour

Déposer une piste $\Delta\tau_{ij}^k(t)$ sur le trajet $T^k(t)$ conformément à
l'équation 1.4

Fin pour

Évaporer les pistes selon la formule 1.5

Fin pour

Algorithme 1.2. Algorithme de colonies de fourmis de base : Ant System

1.4.1.2. Les variantes de l'algorithme AS

À la suite de l'AS original, diverses améliorations ont été effectuées et ont donné lieu à plusieurs variantes de Ant System qui, collectivement, forment les principaux algorithmes d'optimisation par les colonies de fourmis (ACO).

(a) MAX-MIN Ant System (MMAS)

MAX-MIN Ant System est une amélioration par rapport à l'idée originale. MMAS a été proposé par Stützle et Hoos [STU 00], qui ont présenté un certain nombre de modifications dont les plus importantes sont les suivantes:

- seule la meilleure fourmi peut mettre à jour les traînées de phéromone.
- les valeurs minimum et maximum de phéromone sont limitées.

L'équation (1.4) prend la nouvelle forme suivante :

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \Delta\tau_{ij}^{best} \quad (1.6)$$

où Δ est la valeur de mise à jour phéromone définie par :

$$\Delta\tau_{ij}^{best} = \begin{cases} \frac{1}{L_{best}} & \text{si la meilleure fourmi utilisée l'arête (i, j) dans son tour} \\ 0 & \text{sinon} \end{cases} \quad (1.7)$$

L_{best} est la longueur de la tournée de la meilleure fourmi. Ceci peut-être (sous réserve de la décision de l'algorithme désigné) soit le meilleur tour trouvé dans l'itération courante (la meilleure itération L_{ib}) , ou la meilleure solution trouvée depuis le début de l'algorithme best-so-far L_{bs} , ou une combinaison des deux.

En ce qui concerne les limites des valeurs minimales et maximales autorisées de phéromone, τ_{min} et τ_{max} respectivement, Stützle et Hoos [STU 00] suggèrent qu'elles devraient être choisies expérimentalement en se basant sur le problème considéré. La valeur maximale τ_{max} est calculée analytiquement à condition que la longueur optimale de la tournée de la fourmi soit connue. Dans le cas du TSP, τ_{max} est donnée par

$$\tau_{max} = \frac{1}{\rho} \cdot \frac{1}{L^*} \quad (1.8)$$

où L^* est la longueur de la tournée optimale. Si L^* n'est pas connue, elle peut être approximée par L_{bs} . La valeur minimale de phéromone τ_{min} doit être choisie avec prudence, car elle a une influence assez forte sur la performance de l'algorithme.

Stützle et Hoos [STU 00] présentent une approche analytique pour trouver cette valeur en fonction de la probabilité p_{best} qu'une fourmi fasse le meilleur tour trouvé jusqu'ici. Cela se fait de la manière suivante : d'abord, il est supposé, qu'à chaque étape de construction, une fourmi a un nombre k constant des choix disponibles. Par conséquent, la probabilité qu'une fourmi prenne la bonne décision (par exemple, la décision qui appartient à la séquence des décisions menant à la construction de la meilleure tournée trouvée jusqu'ici) à chacune des n étapes est donnée par $p_{dec} = \sqrt[n-1]{p_{best}}$. La formule analytique qu'ils proposent pour trouver le τ_{min} est

$$\tau_{min} = \frac{\tau_{max} \cdot (1 - p_{dec})}{k \cdot p_{dec}} \quad (1.9)$$

Le processus de mise à jour des phéromones dans MMAS est effectué en vérifiant que toutes les valeurs de phéromones sont dans les limites imposées:

$$\tau_{ij} = \begin{cases} \tau_{min} & \text{si } \tau_{ij} < \tau_{min} \\ \tau_{ij} & \text{si } \tau_{min} \leq \tau_{ij} \leq \tau_{max} \\ \tau_{max} & \text{si } \tau_{ij} > \tau_{max} \end{cases} \quad (1.10)$$

MAX-MIN Ant System a fourni une amélioration significative de la performance du système AS de base. Les premières implémentations de MAX-MIN Ant System se concentraient sur le TSP, il a été appliqué, plus tard, à beaucoup d'autres problèmes.

(b) *Ant Colony System (ACS)*

Une autre amélioration du Ant System original est Ant Colony System (ACS), cet algorithme a été présenté par Gambardella et Dorigo [GAM 96]. La contribution la plus intéressante de l'ACS est l'introduction d'une mise à jour locale de phéromone, en plus de la mise à jour de phéromone effectuée à la fin du processus de construction (appelé ici mise à jour hors ligne de phéromone).

La mise à jour locale de phéromone est assurée par toutes les fourmis après chaque étape de la construction. Chaque fourmi ne l'applique qu'à la dernière arête traversée:

$$\tau_{ij} = (1 - \varphi) \cdot \tau_{ij} + \varphi \cdot \tau_0 \quad (1.11)$$

où $\varphi \in (0, 1)$ est le coefficient de désintégration des phéromones, et τ_0 est la valeur initiale de la phéromone.

De la même façon que l'algorithme MMAS, la mise à jour hors ligne de phéromone est appliquée à la fin de chaque itération par une seule fourmi (celle qui a trouvé la meilleure solution dans l'itération). La fonction de mise à jour hors ligne de phéromone est comme suit:

$$\tau_{ij} \leftarrow \begin{cases} (1 - \rho) \cdot \tau_{ij} + \rho \cdot \Delta\tau_{ij} & \text{si l'arête}(i, j) \text{ appartient au meilleur} \\ & \text{tour trouvé dès le début} \\ \tau_{ij} & \text{sinon} \end{cases} \quad (1.12)$$

Comme dans MMAS, $\Delta\tau_{ij} = \frac{1}{L_{best}}$, où L_{best} peut être soit la meilleure dans l'itération ou le meilleur dès le début (best-so-far). En plus des différences avec la procédure de mise à jour de phéromone, l'ACS utilise également une règle de transition différente, appelée la règle proportionnelle pseudo-aléatoire. Soit k une fourmi qui se trouve à l'état i , soit un paramètre $q_0 \in [0, 1]$, et q une valeur aléatoire dans $[0, 1]$, alors le prochain noeud, j , est choisi selon la distribution de probabilité suivante :

Si $q \leq q_0$:

$$p_{ij}^k = \begin{cases} 1 & \text{si } j = \text{argmax}_{j \in N_k(l)} \tau_{ij} \cdot \eta_{ij}^\beta \\ 0 & \text{sinon} \end{cases} \quad (1.13)$$

Sinon ($q > q_0$) et Eq. (1.3) est utilisée.

Ant Colony System a été initialement développé pour le problème TSP, mais il a été utilisé ensuite pour résoudre des problèmes tels que le cheminement de véhicules, la commande séquentielle et les calendriers.

(c) Rank-based Ant System (ASRank)

The Rank-based Ant System (ASRank) [BUL 99] intègre la notion de classement dans la procédure de mise à jour de phéromones. Les N fourmis sont classées en fonction de la qualité décroissante de leurs solutions, à savoir la longueur de la tournée ($L_1 \leq L_2 \leq \dots \leq L_N$). Les pistes de phéromones sont mises à jour hors ligne de telle sorte que seuls les chemins traversés par les $\sigma - 1$ meilleures fourmis reçoivent des phéromones, et la quantité déposée dépend directement du classement de la fourmi, μ , et de la qualité de sa solution. En outre, les chemins parcourus par la meilleure solution globale L^{gb} , reçoivent une quantité additionnelle de phéromone qui dépend de la qualité de cette solution, pondérée par le paramètre σ .

La fonction de mise à jour de phéromone se présente comme suit:

$$\tau_{ij} = \rho \tau_{ij} + \Delta\tau_{ij}^{gb} + \Delta\tau_{ij}^{rank} \quad (1.14)$$

Où

$$\Delta\tau_{ij}^{rank} = \begin{cases} \sum_{\mu=1}^{\sigma-1} (\sigma - \mu) \frac{Q}{L_\mu} & \text{si la } \mu \text{ ème meilleure fourmi se déplace} \\ 0 & \text{sinon} \end{cases} \quad (1.15)$$

Et

$$\Delta\tau_{ij}^{gb} = \begin{cases} \sigma \frac{Q}{L^{gb}} & \text{si l'arête } (i, j) \text{ fait partie de la meilleure solution trouvée} \\ 0 & \text{sinon} \end{cases} \quad (1.16)$$

1.4.2. Les essaims particulaires

L'optimisation par les essaims de particules (Particle Swarm Optimization ou PSO) a aussi une inspiration biologique. Elle a été introduite en 1995 par Kennedy et Eberhart,

psychologue social et ingénieur en électricité [KEN 95]. C'est un algorithme d'optimisation simple, efficace et global qui peut résoudre des nouveaux problèmes discontinus, multimodaux et non-convexes.

PSO est une méthode heuristique basée sur la population découverte grâce à la simulation des modèles sociaux de volées d'oiseaux, bancs de poissons ou essaims d'insectes. Supposons le scénario suivant: un groupe d'oiseaux recherche aléatoirement de la nourriture dans une région. Il n'y a qu'un seul morceau de nourriture dans la zone fouillée. Les oiseaux ne savent pas où est la nourriture, mais ils savent à quelle distance elle se trouve et les positions de leurs pairs. Alors, *quelle est la meilleure stratégie pour trouver la nourriture?* Une stratégie efficace est de suivre l'oiseau qui est le plus proche de la nourriture [XIA 04]. PSO se base sur ce scénario et l'utilise pour résoudre les problèmes d'optimisation. Dans PSO, chaque solution unique est comme un "oiseau" dans l'espace de recherche, qui est appelé "particule".

Dans PSO, un essaim se compose de N particules se déplaçant dans un espace de recherche à D -dimensions. La position de la particule i à l'itération t est utilisée pour évaluer la qualité de la particule et représente une solution candidate pour la recherche ou l'optimisation des problèmes. Elle est représentée par $x_i(t) = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$, où $x_{i,n}(t) \in [l_n, u_n]$, et $1 \leq n \leq N$, avec l_n et u_n qui sont les limites inférieure et supérieure pour la dimension n , respectivement.

Pendant le processus de recherche, la particule ajuste itérativement sa position vers l'optimum global en fonction de deux facteurs: la meilleure position rencontrée par lui-même (p_{best}), notée $p_{i,j} = (p_{i,1}, p_{i,2}, \dots, p_{i,D})$ et la meilleure position rencontrée par l'essaim (g_{best}) notée $p_g = (p_{g,1}, p_{g,2}, \dots, p_{g,D})$. Sa vitesse à l'itération t est représentée par $v_i(t) = (v_{i,1}, v_{i,2}, \dots, v_{i,D})$. La vitesse est bornée par une vitesse maximale $v_{imax} = (v_{imax1}, v_{imax2}, \dots, v_{imaxD})$. La vitesse et la position de la particule à la prochaine itération sont calculées selon les équations suivantes:

$$\begin{aligned} v_{i,j} &= w v_{i,j}(t) + c_1 r_{1,j}(t) (p_{best_{i,j}}(t) - x_{i,j}(t)) + \dots \\ &\dots + c_2 r_{2,j}(t) (g_{best_i}(t) - x_{i,j}(t)) \end{aligned} \quad (1.17)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (1.18)$$

Ici, r_1 et r_2 sont deux fonctions aléatoires dans l'intervalle $w \in [0.8, 1.2]$ est appelé poids d'inertie et, dans le PSO original, il a été fixé à 1 [ALA 08]. Un grand poids d'inertie atteint l'exploration globale et un faible poids d'inertie tend à faciliter l'exploration locale pour affiner la zone de recherche en cours. Par conséquent, le poids d'inertie w est critique pour le comportement de convergence du PSO. Une valeur appropriée pour le poids d'inertie fournit habituellement l'équilibre entre les capacités d'exploration globales et locales et fournit, par conséquent, une meilleure solution optimale.

Les paramètres c_1 et $c_2 \in [0, 2]$ sont des coefficients d'accélération notamment appelés paramètres cognitif et social, respectivement. Ils contrôlent à quelle distance une particule se déplace dans une seule itération. Les valeurs par défaut $c_1 = c_2 = 2$ ont été proposées [KEN 95] et des travaux récents ont suggéré qu'il pourrait être préférable de choisir un plus grand paramètre cognitif, c_1 , mais avec $c_1 + c_2 = 4$.

$r_1 \sim U(0, 1)$ et $r_2 \sim U(0, 1)$ sont utilisés pour la nature stochastique de l'algorithme. Habituellement, la vitesse maximale V_{max} est fixée à la moitié de la longueur de l'espace de recherche. Si V_{max} est trop petite, les particules ne peuvent pas explorer suffisamment au-delà des régions localement bonnes [ALA 08].

Si V_{max} est trop élevée les particules peuvent survoler de bonnes solutions. La première partie de l'équation (1.17) rend « les particules volantes » capables de mémoriser et d'explorer de nouveaux espaces de recherche. La deuxième partie est la partie « connaissance », qui représente la pensée privée de la particule elle-même. La troisième partie est la partie « sociale », qui représente la collaboration entre les particules. L'équation (1.17) est utilisée pour mettre à jour la vitesse de la particule. Ensuite, la particule vole vers une nouvelle région selon l'équation (1.17). La performance de chaque particule est mesurée selon une fonction de fitness prédéfinie [XIA 05].

Bien que l'algorithme PSO recherche des valeurs optimales, il est possible, pour une particule, d'échapper à son espace de recherche dans toutes les dimensions. Quand une restriction d'une limite inférieure ou une limite supérieure n'est pas satisfaisante, une règle de réparation devrait être appliquée. Une règle appropriée peut être appliquée conformément aux équations (1.19) et (1.20), respectivement [ALA 08]:

$$x_i = x_i + \alpha \times r \times (u_n(x_i) - l_n(x_i)) \quad (1.19)$$

$$x_i = x_i - \alpha \times r \times (u_n(x_i) - l_n(x_i)) \quad (1.20)$$

$\alpha \in [0, 1]$ est un paramètre utilisateur spécifié et $r \sim U(0, 1)$. Le mouvement d'une particule en deux dimensions selon les équations (1.17) et (1.18) a été schématisé par la Figure 1.5. Le pseudo-code de la procédure de recherche de PSO est donné dans l'algorithme 1.3.

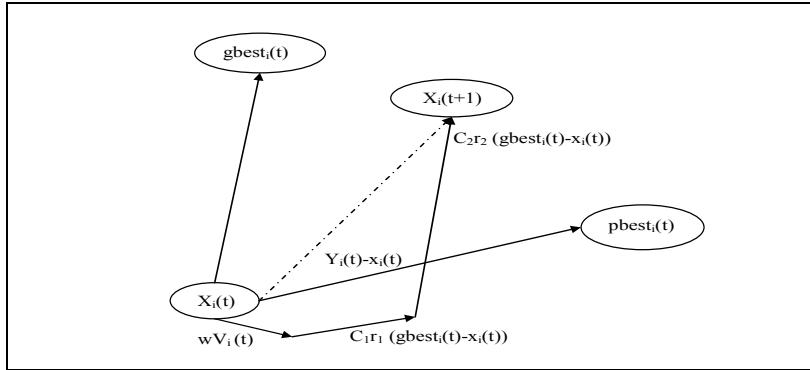


Figure 1.5. Mouvement d'une particule [ALA 08]

Pour chaque particule

 Initialiser les particules

Fin

Faire

Pour chaque particule

 Calculer la valeur de fitness

 Si la valeur de fitness est mieux que la meilleure valeur de fitness, pbest, dans l'histoire de particule

 Régler la valeur actuelle comme la nouvelle pbest

Fin

 Choisir la particule avec la meilleure valeur de fitness de toutes les particules comme le gbest

Pour chaque particule

 Calculer la vitesse des particules selon l'équation 1.16

 Mettre à jour la position de la particule selon l'équation 1.17

Fin

Tant que les critères d'arrêt ne sont pas atteints

Algorithme 1.3. Pseudo-code de l'algorithme de PSO [ALA 08]

1.4.3. Les algorithmes inspirés des abeilles (Bee inspired algorithms)

Les algorithmes inspirés des abeilles ont émergé dans le domaine de l'intelligence en essaims au cours des dernières années. Ils sont inspirés par le comportement des abeilles dans la nature. L'idée de base de ces algorithmes est de créer une colonie d'abeilles artificielles capable de résoudre, avec succès, des problèmes difficiles d'optimisation.

Diwold et al. [DIW 11] divisent les algorithmes inspirés des abeilles en deux classes :

- Les algorithmes qui utilisent des mécanismes génétiques et comportementaux sous-jacents au comportement d'accouplement de l'abeille: *Honeybee Mating Optimisation Algorithm*, *Honey Bee Inspired Evolutionary Computation*.
- Les algorithmes qui s'inspirent du comportement de l'abeille : *The Artificial Bee Colony Algorithm (ABC)*, *Bees Algorithm (BA)*, *Bee Colony Optimisation Algorithm (BCO)* *The Bee Colony-Inspired Algorithm (BCiA)*, *The Virtual Bee Algorithm (VBA)*.

Nous présentons ici quelques exemples des algorithmes cités précédemment.

1.4.3.1. Les colonies d'abeilles artificielles (ABC)

L'algorithme Artificial Bee Colony (ABC) a été proposé par Karaboga en 2005 pour résoudre des problèmes d'optimisation multidimensionnelle et multimodale [KAR 05]. Parmi les algorithmes inspirés des abeilles, c'est le plus étudié et le plus utilisé pour résoudre des problèmes du monde réel. Jour après jour, le nombre de chercheurs s'intéressant à ABC augmente rapidement. Dans [KAR 14], les auteurs présentent une étude exhaustive de l'algorithme ABC, de ses variantes et de ses applications.

L'algorithme ABC commence par n solutions (sources de nourriture) de dimension d qui sont modifiées par les abeilles artificielles. Chaque solution $\vec{x}_i = [x_{i1}, x_{i2}, \dots, x_{id}]$ est évaluée par une fonction de fitness $f(\vec{x}_i)$, $i = 1, \dots, n$. Les abeilles visent à découvrir des endroits contenant des sources de nourriture (régions de l'espace de recherche) avec une grande quantité de nectar (une bonne fitness). Il existe trois types d'abeilles: les abeilles scouts qui volent au hasard dans l'espace de recherche sans orientation; les abeilles employées qui exploitent le voisinage de leur emplacement en sélectionnant une

solution aléatoire; et les abeilles spectatrices qui utilisent la fitness de la population pour choisir, de manière probabiliste, une solution de guidage pour exploiter leur voisinage.

Si la quantité de nectar d'une nouvelle source est supérieure à celle de la précédente dans la mémoire, ils mettent à jour la nouvelle position et oublient la précédente (greedy selection). Si une solution n'est pas améliorée par un nombre prédéterminé d'essais, contrôlé par le paramètre de la limite, alors la source de nourriture est abandonnée par l'abeille employée correspondante et celle-ci devient une abeille scout.

Placer chaque abeille employée sur une position aléatoire dans l'espace de recherche.

Tant que le critère d'arrêt n'est pas atteint **faire**

Pour toutes les abeilles employées **faire**

Si le nbr d'étapes sur la même position = la limite **alors**

Choisir une position aléatoire dans l'espace de recherche ;

Sinon

tenter d'améliorer la position ;

Si la meilleure position est trouvée, **alors**

changer la position ;

Réinitialisation le nbr d'étapes sur la même position ;

Fin Si ;

Fin Si ;

Fin Pour ;

Pour toutes les abeilles spectateurs **faire**

Choisir la position d'abeille employée ;

tenter d'améliorer la position ;

Fin pour ;

Fin tant que

Algorithme 1.4. L'algorithme ABC [DIW 11]

L'algorithme ABC tente d'équilibrer l'exploration et l'exploitation en utilisant les abeilles employées et spectatrices pour effectuer la recherche locale, et les abeilles scouts pour effectuer la recherche globale, respectivement [PAR 11].

1.4.3.2. L'Algorithme d'abeilles (BA)

L'algorithme d'abeilles (Bee Algorithm ou BA) a été introduit par Pham et al. en 2005 [PHA 06] comme une méthode d'optimisation des fonctions continues et combinatoires. Cet algorithme simule aussi le comportement alimentaire des abeilles en effectuant une recherche de voisinage combinée avec la recherche aléatoire.

Comme dans ABC, la population d'abeilles est divisée en deux groupes: les scouts et les recrues. Alors que les scouts sont responsables de l'exploration de l'espace de recherche, les recrues tentent d'exploiter l'espace pour trouver des solutions via une recherche locale.

Placer chaque abeille sur une position aléatoire dans l'espace de recherche ;

Évaluer la fitness de la population ;

Tant que le critère d'arrêt n'est pas satisfait **faire**

 Choisir des solutions pour une recherche locale (exploitation) ;

 Affecter les abeilles pour commettre la recherche locale sur les solutions choisies et la fitness évaluée ;

 Pour chaque solution, sélectionner la meilleure amélioration ;

 Remplacer les solutions restantes avec des solutions aléatoires (scouts) ;

Fin Tant que

Algorithme 1.5. L'algorithme BA [DIW 11]

1.4.3.3. L'algorithme d'optimisation par les colonies d'abeilles (BCO)

L'algorithme Bee Colony Optimization ou BCO est un algorithme basé sur la population [TEO 05]. Une population d'abeilles artificielles cherche la solution optimale. Les abeilles artificielles représentent des agents en collaboration, qui résolvent des problèmes complexes d'optimisation combinatoire. Chaque abeille artificielle génère une solution au problème. L'algorithme se compose de deux phases alternées: passage forward et passage backward. Dans chaque passage forward, chaque abeille

artificielle explore l'espace de recherche. Elle s'applique à un nombre prédéfini de mouvements, qui construisent et/ou améliorent la solution, créant ainsi une nouvelle solution. Après avoir obtenu de nouvelles solutions partielles, les abeilles vont de nouveau au nid et commencent la deuxième phase : le passage dit backward. Dans le passage backward, toutes les abeilles artificielles partagent des informations sur leurs solutions [TEO 09].

Initialisation

Pour toutes les I itérations **Faire**

Pour tous les m stages **Faire**

Pour toutes les B abeilles **Faire**

 Passage forward: choisir une solution partielle

Fin Pour

Pour toutes les B abeilles **Faire**

 passage backward: l'échange d'informations sur les solutions partielles avec les abeilles dans le nid

Fin Pour

Fin Pour

Si la meilleure solution obtenue dans l'itération est la meilleure globale, mettre à jour la solution la plus connue

Fin Pour

Algorithme 1.6. L'algorithme de BCO [DIW 11]

1.4.4. *L'algorithme des essaims de poissons artificiels (AFSA)*

L'algorithme des essaims de poissons artificiels (Artificial Fish Swarm Algorithm ou AFSA) a été proposé par Li et al [LI 03], il est parfois appelé Artificial Fish School Algorithm, il est vaguement basé sur le comportement de l'essaimage de poissons [SIM 13a].

Dans les zones d'eau, en suivant d'autres poissons, un poisson peut toujours trouver de la nourriture à un endroit où il y en a beaucoup. Généralement, la zone qui contient plus de nourriture, contient plus de poissons. Selon ce phénomène, AFSA construit certains

poissons artificiels (Artificial Fish : AF), qui recherchent une solution optimale dans l'espace des solutions (l'environnement dans lequel vivent les poissons artificiels) en imitant le comportement des essaims de poissons. Trois notions de base des poissons artificiels sont définies comme suit [CAI 10]:

Proies: Le poisson perçoit la concentration de nourriture dans l'eau pour déterminer le mouvement par la vision ou le sens, puis choisit la tendance.

L'essaim: Les poissons se réunissent en groupes naturellement dans le processus de déplacement, qui est une sorte de mode de vie afin de garantir l'existence de la colonie et éviter les dangers.

Suivre: Dans le processus de déplacement de l'essaim de poissons, quand un seul poisson ou plusieurs poissons trouvent la nourriture, les partenaires du voisinage seront entraînés et atteindront la nourriture rapidement. Pour plus de détails sur l'algorithme voir [NES 12].

1.4.5. L'algorithme des chauves-souris (Bat algorithm)

Les chauves-souris sont des animaux fascinants et leur capacité avancée de l'écholocation a attiré l'attention des chercheurs de différents domaines. L'écholocation fonctionne comme un type de sonar: les chauves-souris, émettent une impulsion forte et courte du son, attendent qu'elle se heurte à un objet et qu'après un certain temps, l'écho retourne à leurs oreilles. Ainsi, les chauves-souris peuvent déterminer à quel point elles sont loin de l'objet. En outre, ce mécanisme d'orientation étonnant rend les chauves-souris capables de distinguer la différence entre un obstacle et une proie, ce qui leur permet de chasser même dans l'obscurité complète [ROD 14].

En se basant sur le comportement des chauves-souris, Yang [YAN 10] a développé une nouvelle technique intéressante de méta-heuristique d'optimisation appelée l'algorithme des chauves-souris (Bat algorithm). Cette technique a été développée pour imiter le comportement d'une bande de chauves-souris qui suivent des proies/aliments en utilisant leur capacité d'écholocation. Dans [YAN 13] Yang fournit une revue des algorithmes de chauve-souris et de ses nouvelles variantes.

Dans l'algorithme de base des chauves-souris développé par Yang [YAN 10,11] les règles approximatives ou idéalisées suivantes ont été utilisées :

1. Toutes les chauves-souris utilisent l'écholocation pour détecter la distance, et connaissent aussi la différence entre la nourriture/les proies et les obstacles.
2. Les chauves-souris volent au hasard avec une vitesse v_i à la position x_i avec une fréquence f_{min} , longueur d'onde λ et l'intensité A_0 pour la recherche d'une proie. Elles peuvent ajuster automatiquement la longueur d'onde (ou fréquence) dès l'émission de leurs impulsions et ajuster le taux d'émission d'impulsions en fonction de la proximité de leur cible.
3. Bien que l'intensité puisse varier de plusieurs façons, nous supposons qu'elle varie d'une grande valeur positive A_0 à une valeur constante minimale A_{min} .

1.4.6. L'algorithme des essaims de lucioles (Firefly algorithm)

L'algorithme d'optimisation basé sur les essaims de lucioles (insectes lumineux) ou Firefly algorithm (FA ou FFA) développé par Xin-She Yang à l'Université de Cambridge en 2007 [VAN 14], utilise les trois règles idéalisées suivantes:

1. Tous les lucioles sont unisexes, cela signifie qu'une luciole est attirée par une autre, indépendamment de son sexe.
2. L'attractivité et la luminosité sont proportionnelles entre elles, de sorte que, pour deux lucioles clignotantes, la moins lumineuse se déplacera vers la plus lumineuse. L'attractivité et la luminosité diminuent à l'augmentation de leur distance. S'il n'y a pas une luciole plus brillante que l'autre, elle vont se placer de manière aléatoire.
3. La luminosité d'une luciole est déterminée par le point de vue de la fonction objective.

Pour un problème de maximisation, la luminosité est simplement proportionnelle à la valeur de la fonction objective.

D'autres formes de la luminosité peuvent être définies d'une manière identique à la fonction de fitness dans les algorithmes génétiques.

1.4.7. L'algorithme d'optimisation de l'exploration bactérienne (BFO)

L'algorithme d'optimisation basé sur l'exploration bactérienne (Bacterial Foraging Optimization ou BFO) proposé en 2002 par Passino [PAS 02] est basé sur le comportement alimentaire des bactéries d'Escherichia coli (E. coli) présentes dans l'intestin humain [PAT 11].

La théorie de l'exploration pour la recherche de nourriture est basée sur l'hypothèse que les bactéries obtiennent les constituants énergétiques maximaux dans un temps supposé être petit. Le BFO de base se compose de trois principaux mécanismes : la chimiotaxie, la reproduction et l'élimination-dispersion [PAS 02, PAT 11].

1.5. Autres méthodes évolutionnaires pour l'optimisation

Nous présentons brièvement, dans cette section quelques techniques très récentes de l'informatique évolutionnaire qui n'appartiennent pas à l'intelligence en essaims.

1.5.1. L'optimisation basée sur la Biogéographie

La Biogéographie est l'étude de la spéciation, l'extinction, et la distribution géographique des espèces biologiques [SIM 13a]. L'optimisation basée sur la biogéographie (Biogeography Based Optimization ou BBO) est un algorithme évolutionnaire qui a été introduit en 2008 [SIM 08], et s'inspire de la science de la biogéographie. La biogéographie décrit la migration des espèces entre habitats en fonction des facteurs environnementaux. Ces facteurs peuvent être représentés quantitativement et sont appelés variables de l'indice d'aptitude (suitability index variables ou SIV). Des exemples de ces facteurs incluent la quantité de précipitations, la quantité d'eau douce disponible, la diversité de la végétation, et la gamme de température. Une zone qui est très appropriée pour l'habitation des espèces biologiques est considérée comme ayant un indice de qualité élevé de l'habitat. Les biologistes ont développé des modèles mathématiques de la migration, la spéciation, et l'extinction [SIM 13b].

Dans BBO, chaque solution du problème est considérée comme un « habitat» avec l'indice de survie de l'habitat, qui est similaire à la fitness dans le calcul évolutionnaire,

pour évaluer un individu. Les habitats avec un indice de survie élevé partagent leurs caractéristiques avec les habitats avec un indice de survie faible. Le processus de partage des bonnes caractéristiques parmi les solutions est noté comme la migration. BBO adopte la stratégie de migration pour partager l'information entre les solutions. Les bonnes informations d'individus peuvent être conservées pendant le processus évolutionnaire afin d'assurer la convergence de la population. Un opérateur de mutation est utilisé pour générer des diverses solutions visant à promouvoir la diversité de la population [MO 14].

1.5.2. L'algorithme de recherche gravitationnelle

L'algorithme de recherche gravitationnelle (Gravitational Search Algorithm ou GSA) a été introduit en 2009 par Rashedi et al [RAS 09], c'est un algorithme de recherche basée sur la population, et sur la loi de la gravité et l'interaction de masse.

GSA est similaire à l'optimisation par les essaims de particules (PSO) et fonctionne avec le principe que chaque individu de la population a une position et une vitesse dans l'espace de recherche, mais il comprend également une accélération. Les particules s'attirent mutuellement en fonction de leur valeur de masse, qui est proportionnelle à leur valeur de fitness (c'est-à-dire inversement proportionnelle à leur valeur de coût) [SIM 13a].

Dans cet algorithme, les agents sont considérés comme des objets et leur performance est mesurée par leurs masses. Tous ces objets s'attirent mutuellement par la force de gravité, et cette force provoque un mouvement global de tous les objets vers les objets avec des masses plus lourdes. Par conséquent, les masses coopèrent en utilisant une forme de communication directe, par la force gravitationnelle. Les lourdes masses - qui correspondent à de bonnes solutions - se déplacent plus lentement que les plus légères, ce qui garantit l'étape de l'exploitation de l'algorithme [RAS 09].

1.5.3. L'optimisation basée sur l'enseignement-apprentissage

L'optimisation basée sur l'enseignement-apprentissage (Teaching Learning Based Optimization ou TLBO) est un algorithme inspiré de processus d'enseignement-apprentissage proposé par Rao et al. [RAO 11]. Cet algorithme est basé sur l'effet de l'enseignant sur les apprenants dans la classe. De manière similaire à d'autres méthodes

de l'informatique évolutionnaire, TLBO est un algorithme d'optimisation basé sur la population, et utilise une population de solutions pour trouver la solution globale. La population est considérée comme un groupe d'apprenants ou une catégorie d'apprenants. Dans cet algorithme d'optimisation, différents sujets proposés aux apprenants sont considérés comme des variables de conception du problème d'optimisation. Le résultat d'un apprenant est analogue à la valeur du 'fitness' du problème d'optimisation. La meilleure solution dans l'ensemble de la population est considérée comme l'enseignant. Les variables de conception sont les paramètres impliqués dans la fonction objective du problème d'optimisation donné, et la meilleure solution est la meilleure valeur de la fonction objective [RAO 14]. Le processus de TLBO est divisé en deux parties: la première partie représente la «Phase de l'enseignant» qui signifie l'apprentissage de l'enseignant et la deuxième partie représente la « Phase de l'apprenant» qui signifie l'apprentissage par l'interaction entre les apprenants.

Par rapport aux autres algorithmes évolutionnaires, TLBO a moins de paramètres qui doivent être déterminés au cours du processus de renouvellement, et il est très efficace pour certains problèmes d'optimisation. Cependant, comme un algorithme basé sur la population, certains problèmes complexes provoquent le phénomène de convergence locale de TLBO [CHE 15].

1.6. Conclusion

Les approches évolutionnaires se sont révélées être des méthodes puissantes pour résoudre des problèmes d'optimisation. Dans ce chapitre, nous avons défini le problème d'optimisation puis nous avons présenté quelques méthodes évolutionnaires qui sont largement appliquées dans divers problèmes d'optimisation. Les méthodes évolutionnaires, basées sur la notion de population, ont été présentées en les divisant en trois classes. Dans notre classification, nous adoptons la vision de certains chercheurs qui considèrent que l'intelligence en essaims (swarm intelligence) fait partie de l'informatique évolutionnaire alors que d'autres sont opposés à cette inclusion. Ainsi, selon notre vision, la première classe d'approches évolutionnaires contient les méthodes classiques telles que les algorithmes génétiques, les stratégies d'évolution, la programmation évolutive ainsi que la programmation génétique. La deuxième classe inclut les méthodes évolutionnaires récentes issues de l'intelligence en essaims (swarm

intelligence). La troisième classe contient d'autres méthodes récentes telles que l'optimisation basée sur la Biogéographie, l'algorithme de recherche gravitationnelle et l'optimisation basée sur l'enseignement-apprentissage.

La sélection d'attributs ou de caractéristiques peut être définie comme un problème d'optimisation qui a intéressé beaucoup de chercheurs. Le chapitre suivant présente les concepts de base et les méthodes les plus utilisées dans le domaine de la sélection de caractéristiques.

CHAPITRE 2

CONCEPTS ET METHODES DE LA SELECTION D'ATTRIBUTS

Ce chapitre permet de présenter les concepts de base ainsi que les méthodes de la sélection d'attributs ou de caractéristiques. La sélection d'attributs est définie dans la section 2. Le processus de sélection d'attributs est présenté dans la section 3. La section 4 est consacrée à la catégorisation des méthodes de sélection d'attributs. La section 5 décrit les différentes méthodes classiques de sélection d'attributs, alors que la section 6 est dédiée à une brève description de la manière avec laquelle les algorithmes de l'intelligence en essaims peuvent être utilisés pour la sélection d'attributs, particulièrement ceux basés sur les colonies de fourmis et les essaims particulaires. La section 7 présente certains domaines d'application de la sélection d'attributs.

2.1. Introduction

Plusieurs paramètres peuvent influer sur les performances d'un système de classification. Les caractéristiques extraites à partir des entités afin de représenter ces dernières peuvent être considérées parmi les paramètres les plus importants. La réduction de la dimensionnalité, via l'extraction et la sélection d'attributs, est l'une des étapes les plus importantes dans le traitement de données. Son but est de sélectionner ou d'extraire un sous-ensemble optimal de caractéristiques pertinentes pour un critère fixé auparavant. Les principaux objectifs de la réduction de dimension sont :

- Faciliter la visualisation et la compréhension des données.
- Réduire l'espace de stockage nécessaire.
- Réduire le temps d'apprentissage et d'utilisation.
- Identifier les facteurs pertinents.
- Améliorer la précision du module de classification.

- Améliorer la fiabilité de l'estimation de la performance.

Guerif [GUE 06] classe toutes les techniques de réduction de dimension en deux grandes catégories :

- la sélection de caractéristiques: permet de sélectionner un sous-ensemble de caractéristiques parmi un ensemble de départ, en utilisant divers critères et différentes méthodes (Figure 2.1).
- l'extraction de caractéristiques: permet de créer de nouveaux ensembles de caractéristiques, en utilisant une transformation ou une combinaison d'un espace de départ et en effectuant une réduction du nombre de dimensions (Figure 2.2).

Dans le cadre de notre travail, nous nous intéressons aux méthodes de sélection de caractéristiques, ce sont ces dernières qui seront étudiées dans la suite de ce chapitre.

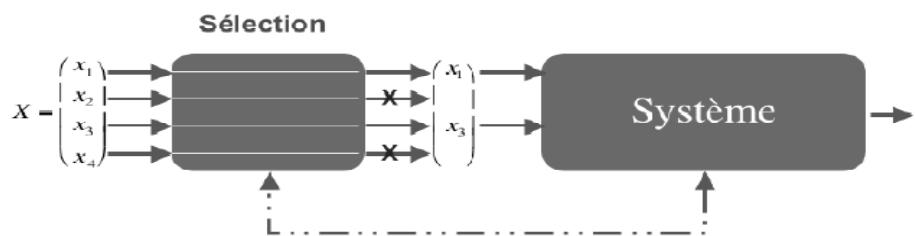


Figure 2.1 – Principe de la sélection de caractéristiques [GUE 06].

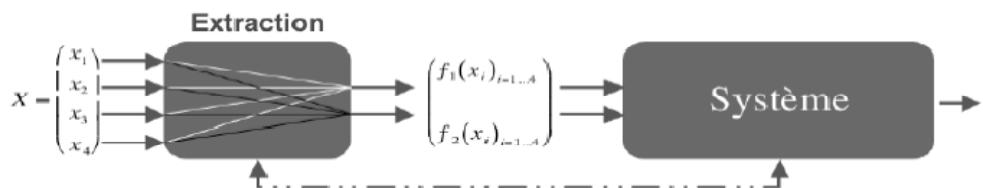


Figure 2.2 – Principe de l'extraction de caractéristiques [GUE 06].

2.2. Définition de la sélection d'attributs

La sélection d'attributs, de caractéristiques ou *feature selection* en anglais, est un problème difficile qui a été étudié depuis les années 70.

La sélection de caractéristiques est un processus de recherche ou une technique utilisée pour choisir les caractéristiques, les variables ou les mesures les plus intéressantes,

pertinentes ou informatives d'un système donné, dans le but de réaliser la tâche pour laquelle il a été conçu.

Dans le domaine d'apprentissage automatique ou, plus précisément, celui de la classification, certaines caractéristiques non pertinentes et/ou redondantes, existant généralement dans les données d'apprentissage, non seulement rendent l'apprentissage plus difficile, mais dégradent aussi les performances de généralisation des modèles d'apprentissage.

Dans la littérature, les auteurs donnent des définitions différentes à la sélection de caractéristiques, Unler et Murat [UNL 10] définissent la sélection de caractéristiques pour un problème de classification de la manière suivante:

Soit G une base de données avec R échantillons et K dimensions (attributs ou caractéristiques), représentée sous forme d'une matrice $G = R * K$. L'objectif de la tâche de sélection de sous-ensemble de caractéristiques est d'obtenir k dimensions, de l'ensemble total de l'espace de caractéristiques où $k < K$, qui permettent d'optimiser une fonction de critère $J(\cdot)$, par exemple, la performance de la classification.

Étant donné l'ensemble des caractéristiques F , le problème de sélection de sous-ensemble de caractéristiques est défini par : trouver un sous-ensemble $L \subseteq F$ et $|L| = k$ qui maximise la fonction de critère,

$$J(L) = \max_{Y \subseteq F, |Y|=k} J(Y). \quad (2.1)$$

2.3. Processus de sélection d'attributs

Idéalement, les méthodes de la sélection d'attributs recherchent à travers tous les sous-ensembles de caractéristiques, et essaient de trouver le meilleur parmi 2^N sous-ensembles candidats concurrents selon une fonction d'évaluation. Toutefois, cette procédure est exhaustive, elle peut être trop coûteuse et pratiquement prohibitive, même pour une taille moyenne d'ensemble de caractéristiques (N). D'autres approches basées sur des méthodes de recherche heuristiques ou aléatoires tentent de réduire la complexité de calcul en compromettant les performances. Ces méthodes ont besoin d'un critère d'arrêt pour éviter la recherche exhaustive de sous-ensembles. D'après Dash et Liu [DAS 97], il y a quatre étapes de base pour une méthode typique de sélection de caractéristique (voir Figure 2.3):

1. **Une procédure de génération** : pour générer le prochain sous-ensemble de caractéristiques.
2. **Une fonction d'évaluation** : pour évaluer le sous-ensemble généré par la procédure de génération.
3. **Un critère d'arrêt** : pour décider quand arrêter.
4. **Une procédure de validation** : pour vérifier si le sous-ensemble est valide ou pas.

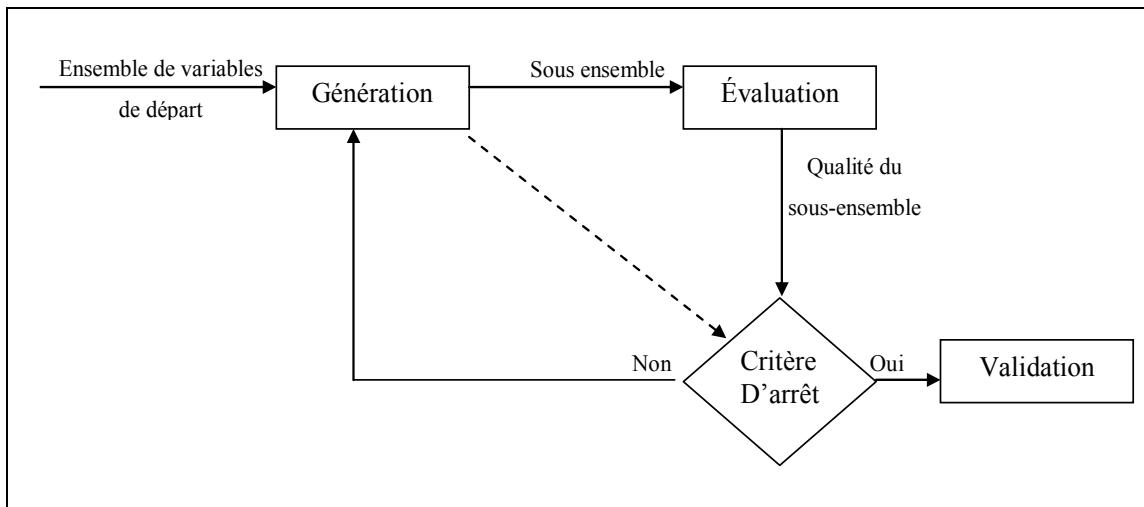


Figure 2.3. Processus de sélection de caractéristiques [DAS 97].

2.3.1. La procédure de génération

La procédure de génération est une procédure de recherche permettant d'explorer l'espace de recherche pour construire les différentes combinaisons de caractéristiques. La génération de sous-ensemble est essentiellement un processus de recherche heuristique qui, à chaque étape, détermine un sous-ensemble candidat dans l'espace de recherche pour l'évaluation. La nature de ce processus est déterminée par deux questions fondamentales [LIU 05].

La première est de déterminer le point (ou les points) de départ de la recherche. En effet, la sélection d'un point dans l'espace sous-ensemble de caractéristiques, pour commencer la recherche, peut influer sur l'orientation de la recherche. Cette dernière peut commencer avec un ensemble vide de caractéristiques et successivement ajouter des caractéristiques. Dans ce cas, la recherche est dite *forward*. Inversement, la recherche peut commencer avec toutes les caractéristiques et successivement en supprimer. Dans ce cas, la recherche est dite *backward*. Une autre alternative, est de commencer quelque

part au milieu et se déplacer vers l'extérieur de ce point, et, dans ce cas, la recherche est dite *bidirectionnelle*.

La deuxième question fondamentale est de choisir une stratégie de recherche. Pour un ensemble de données avec N caractéristiques, il existe 2^N sous-ensembles candidats. Par conséquent, différentes stratégies ont été explorées: recherche complète, séquentielle et aléatoire.

2.3.1.1. *Génération complète*

Les approches regroupées dans cette catégorie effectuent une recherche complète du sous-ensemble optimal par rapport à la fonction d'évaluation choisie. Différentes fonctions d'évaluation sont utilisées pour réduire l'espace de recherche sans perdre les chances de trouver le sous-ensemble optimal. Dans ce cas, bien que l'espace de recherche soit de l'ordre de $O(2^N)$, peu de sous-ensembles sont évalués. L'optimalité du sous-ensemble de caractéristiques, par rapport à la fonction d'évaluation, est garantie car ces approches utilisent une procédure de Backtracking (retours arrière) qui peut être réalisée en utilisant des techniques telles que : *Branch and Bound (BB)*, *Best First Search (BFF)*, *Beam Search*, *Approximate Branch and Bound (AB&B)*, *Automatic Branch and Bound (ABB)*, *Minimum Description Length Method (MDLM)*, *Focus*, *Méthode de Shlimmer et Min-Features (MIFFES_1)* [DAS 97, BEN 02].

2.3.1.2. *Génération séquentielle ou heuristique*

Cette catégorie regroupe les algorithmes itératifs pour lesquels chaque itération permet de sélectionner ou rejeter une ou plusieurs caractéristiques. Les algorithmes avec une génération séquentielle sont simples à implémenter et rapides dans la production des résultats, l'espace de recherche utilisé est de l'ordre $O(N^2)$. Dans ces algorithmes, on abandonne l'exhaustivité et on risque ainsi de perdre les sous-ensembles optimaux.

Parmi les techniques utilisées, on trouve: *Sequential Forward Selection (SFS)*, *Sequential Backward Selection (SBS)*, *sequential backward selection-SLASH (SBS-SLASH)*, *bi-directional search (BDS)*, *Relief*, *Relief_F relevance in context (RC)*, *(p,q) sequential search (PQSS)*, *Probability Of Error & Average Correlation Coefficient (POE+ACC)*, *decision tree method (DTM)*, *relevance in context (RC)*.

2.3.1.3. Génération aléatoire

Par rapport aux deux premières catégories, l'utilisation de la génération aléatoire dans la procédure de sélection de caractéristiques est la plus récente. La procédure commence avec un sous ensemble sélectionné aléatoirement et procède de deux façons différentes. L'une consiste à continuer la génération des sous-ensembles avec la recherche séquentielle (type I) alors que l'autre consiste à générer le sous ensemble suivant d'une manière complètement aléatoire (type II). Ces méthodes recherchent des sous-ensembles en effectuant un maximum d'itérations. Le choix de valeurs appropriées aux paramètres pour la procédure de recherche est une étape très importante afin d'obtenir des résultats satisfaisants.

Parmi les techniques utilisées, on peut citer les algorithmes génétiques (AG), le recuit simulé (*Simulated Annealing ou SA*), *Las Vegas Filter et Wrapper (LVF et LVW)* *Random Generation plus Sequential selection (RGS)*, *Random Mutation Hill Climbing-prototype (RMHC)*, *Randomized Variable Elimination (RVE)*.

2.3.2. La fonction d'évaluation

Typiquement, une fonction d'évaluation tente de mesurer la capacité de discrimination d'une caractéristique ou d'un sous-ensemble pour distinguer les différentes classes. Plusieurs critères d'évaluation ont été proposés, basés sur des hypothèses statistiques ou sur des heuristiques. Dash et Liu [DAS 97] divisent les mesures d'évaluation en cinq catégories alors que Liu [LIU 05] regroupe les mesures en deux catégories : des mesures dépendantes et indépendantes.

2.3.2.1. Les mesures indépendantes

Elles sont utilisées dans les algorithmes de l'approche Filtre (voir section 2.4.1) et exploitent les propriétés intrinsèques des caractéristiques utilisées, sans référence à aucune application.

Mesure de distance : également appelée mesure de séparabilité, de divergence ou de discrimination. Elle permet de mesurer les capacités de discrimination. Pour un problème à deux classes, une variable X est préférée à Y, si X introduit une plus grande différence, entre les probabilités conditionnelles des deux classes, que Y, car on essaye

de trouver la caractéristiques qui permet de séparer les classes au maximum. X et Y sont indifférenciables si la différence est égale à zéro.

Mesure d'information : permet d'estimer le gain d'information d'une caractéristique. Le gain d'information de la caractéristique X est défini comme la différence entre l'incertitude a priori et celle a posteriori, c'est-à-dire avant et après la sélection de la caractéristique X. La caractéristique X est préférée à Y, si le gain d'information de X est plus grand de celui de Y.

Mesure de dépendance : également appelée mesure de corrélation ou mesure de similarité. Elle permet de mesurer la capacité d'une caractéristique à prédire la valeur d'une autre. Dans la sélection de caractéristiques pour un problème de classification, on cherche à quel point une caractéristique est associée à une classe. La caractéristique X est préférée à Y, si la corrélation d'une variable X avec une classe C est plus importante que la corrélation de la variable Y avec C.

Mesure de cohérence : elles sont typiquement différentes des mesures citées ci-dessus en raison de leur forte dépendance de l'information de classe et l'utilisation de Min-Features bias pour la sélection d'un sous-ensemble de caractéristiques. Ces mesures tentent de trouver un nombre minimum de caractéristiques, qui séparent les classes, aussi cohérentes que l'ensemble des caractéristiques possibles. Une incohérence est définie par deux instances ayant les mêmes valeurs de caractéristiques, mais des étiquettes de classes différentes.

2.3.2.2. Les mesures dépendantes

Les mesures dépendantes du taux d'erreur de classification sont utilisées dans les algorithmes de l'approche Wrapper (voir section 2.4.1). Les approches Wrapper définissent la pertinence des caractéristiques par l'intermédiaire d'une prédiction de la performance du système final.

Le Tableau 2.1 présente une comparaison entre les fonctions d'évaluation vues précédemment sans tenir compte du type de procédure de génération utilisée. Les différents paramètres utilisés pour la sélection sont :

1. La généralité : le sous-ensemble sélectionné est convenable pour différents classifiEURS ou non (pas seulement pour un classifieUR).

2. La complexité en termes de temps : le temps nécessaire pour sélectionner un sous-ensemble de caractéristiques.
3. La précision: la qualité de la prédiction en utilisant la sélection de sous-ensembles.

Le ‘–’ dans la dernière colonne, du Tableau 2.1, signifie que rien ne peut être conclu concernant la décision de la fonction d'évaluation correspondante. La précision de toutes les fonctions d'évaluation dépend de l'ensemble des données et du classifieur utilisés (pour la classification après sélection de caractéristiques) sauf «le taux d'erreur de classification». Les résultats dans le Tableau 2.1 présentent une tendance prévisible : plus le temps passe, plus on a de précision. Le tableau nous indique également que la mesure doit être utilisée dans des circonstances différentes, par exemple, les contraintes de temps, compte tenu de plusieurs classifieurs à choisir, la mesure taux d'erreur ne devrait pas être choisie comme une fonction d'évaluation.

Fonction d'évaluation	Généralité	Temps	Précision
Mesures de distance	Oui	Court	–
Mesures d'information	Oui	Court	–
Mesures de dépendance	Oui	Court	–
Mesures de cohérence	Oui	Modéré	–
Mesures de taux d'erreur de classification	Non	Élevé	Très élevé

Tableau 2.1. Une comparaison entre les fonctions d'évaluation [DAS 97]

2.3.3. *Le critère d'arrêt*

Les itérations du processus de sélection de caractéristiques continuent à s'exécuter jusqu'à ce qu'un critère d'arrêt soit rempli. La procédure de génération et la fonction d'évaluation peuvent influencer sur le choix d'un critère d'arrêt.

Les critères d'arrêt basés sur la procédure de génération incluent :

- le nombre de caractéristiques sélectionnées est égal à un nombre prédéfini,
- un nombre prédéfini d'itérations est atteint.

Le critère d'arrêt basé sur la fonction d'évaluation peut être:

- l'ajout (ou suppression) d'une caractéristique ne produit pas un meilleur sous-ensemble,
- un sous-ensemble optimal de caractéristiques est obtenu à partir de certaines fonctions d'évaluation.

2.3.4. La procédure de validation

La procédure de validation n'est pas une étape du processus de sélection de caractéristiques lui-même, mais une méthode de sélection d'attributs (en pratique) doit être validée. Dash et Liu [DAS 97] proposent d'ajouter une quatrième composante à un algorithme de sélection de caractéristiques : une procédure de validation. Elle essaie de tester la validité des sous-ensembles de caractéristiques sélectionnées avec la réalisation de différents tests, et la comparaison des résultats obtenus avec les résultats précédents, ou avec des résultats d'autres méthodes de sélection de caractéristiques en utilisant des données artificielles, réelles, ou les deux.

2.4. Catégorisation des méthodes de sélection d'attributs

Dans la littérature, différentes méthodes ont été développées et utilisées pour la sélection d'attributs. Les auteurs proposent plusieurs catégorisations de ces méthodes selon les stratégies de recherche et les fonctions d'évaluations utilisées.

2.4.1. Catégorisation basée sur les critères d'évaluation

Plusieurs auteurs préfèrent séparer les méthodes de sélection de caractéristiques en trois approches (Filtre, Wrapper et hybride) selon leur dépendance ou indépendance par rapport à l'algorithme d'induction [CAK 02, LIU 05].

L'approche Filtre : cette approche est réalisée comme un prétraitement, elle exploite les propriétés intrinsèques des caractéristiques utilisées, sans référence à une quelconque application. Elle se base sur la performance de la fonction d'évaluation calculée directement sur l'ensemble d'apprentissage comme : la distance, l'information, la dépendance, et la cohérence [SOM 04].

Pour donner un score à un sous-ensemble, une première solution est de donner un score à chaque variable indépendamment des autres et de faire la somme de ces scores. Cette

approche est nommée *Feature Ranking*. L'autre solution est d'évaluer un sous-ensemble dans sa globalité, cette approche est nommée *subset ranking* [DAN 04].

L'approche Wrapper : elle a été introduite plus récemment par John, Kohavi et Pfleger en 1994 [DAN 04]. L'approche Wrapper utilise l'algorithme d'apprentissage comme une fonction d'évaluation, elle définit donc la pertinence des attributs par l'intermédiaire d'une prédiction de la performance du système final.

L'approche filtre est plus rapide que l'approche Wrapper en terme de génération de résultats. Cependant, cette dernière à l'avantage de fournir généralement des résultats plus pertinents pour la classification. La Figure 2.4 présente deux modèles généraux des approches Filtre et Wrapper pour la sélection de caractéristiques. Le Tableau 2.2 présente une comparaison entre les approches Filtre et Wrapper.

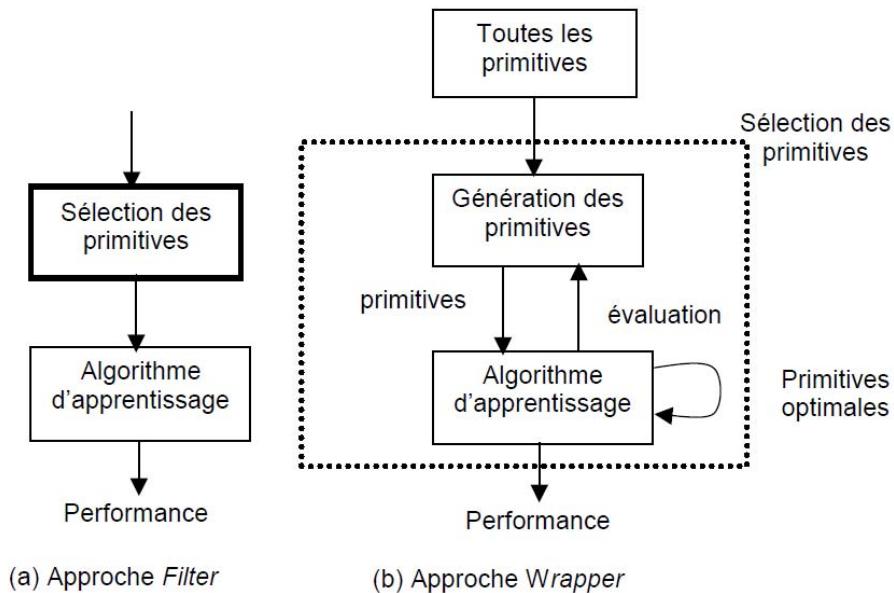


Figure 2.4 L'approche Filtre et l'approche Wrapper [BEN 02]

	Wrappers	Filters
Performances	+ Biais pris en compte	-
Vitesse	-	+
Théorie	+/- Simplicité	+/- Explications
Spécificité	-	+

Tableau 2.2. Comparaison des approches wrapper et filtre

L'approche hybride :

Les algorithmes hybrides ont été plus récemment proposés par rapport aux précédents (filtre et wrapper). Dans ces algorithmes, d'abord, une mesure de « goodness » de sous-ensembles de caractéristiques, basée sur les caractéristiques des données, est utilisée pour choisir les meilleurs sous-ensembles pour une cardinalité donnée. Par la suite, la validation croisée est exploitée pour décider du meilleur sous-ensemble final entre les différentes cardinalités. Ces algorithmes se concentrent principalement sur la combinaison des algorithmes Filtre et Wrapper pour obtenir les meilleures performances possibles, avec un algorithme d'apprentissage particulier et une complexité temporelle comparable à celle des algorithmes de type Filtre.

2.4.2. Catégorisation basée sur la stratégie de recherche

Une autre catégorisation des méthodes de sélection de caractéristiques est celle basée sur la stratégie de recherche. Liu et Yu [LIU 05] proposent de séparer les méthodes en fonction de la stratégie utilisées (complète, heuristique ou aléatoire).

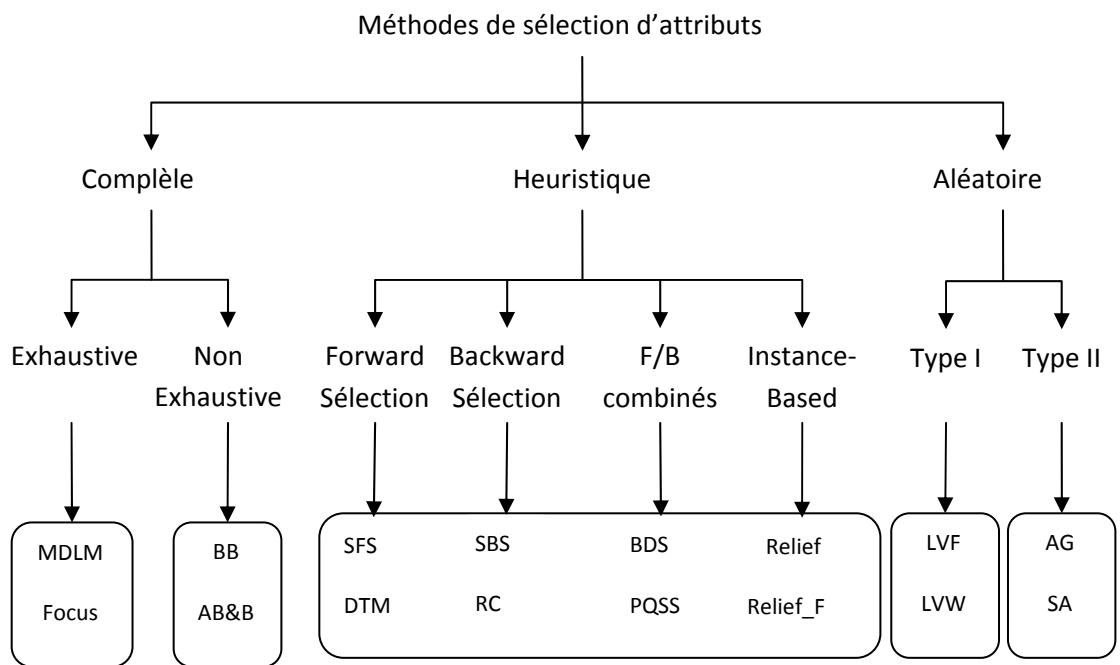


Figure 2.5 Catégorisation des méthodes de sélection de caractéristiques [BEN 02].

La Figure 2.5 présente une classification des méthodes de sélection de caractéristiques basée sur les trois types de procédures de génération (voir section 2.3.1). Les procédures

complètes sont subdivisées en «exhaustive» et «non exhaustive», les procédures heuristiques sont subdivisées en «sélection forward», «sélection backward», «forward/backward combinés», et les catégories «d'instance-based». De même, les procédures de génération aléatoires sont regroupés en «type I» et «type II» (voir la section 2.3.1.3).

			La stratégie de recherche		
			Complète	Séquentielle	Aléatoire
Critère d'évaluation	Filtre	Distance	B&B BFF	Relief ReliefF SFS Segen's	
		Information	MDLM	DTM Koller's	
		Dépendance	Bobrowski's	POE+ACC	
		Cohérence	Focus ABB MIFFES_1Schlimmer's	Set Cover	LVF
	Wrapper	L'exactitude des prévisions	BS AB&B	SBS-SLASH PQSS RC Queiros'	SA RGS LVW RMHC AG RVE
	Hybride	Filtre + Wrapper		Xing's	

Tableau 2.3. Catégorisation des méthodes de sélection de caractéristiques pour la classification [LIU 05].

Liu et Yu [LIU 05] ont proposé deux dimensions afin de catégoriser les méthodes de sélection de caractéristiques pour la classification : la stratégie de recherche (complète, séquentielle et aléatoire) et le critère d'évaluation (Filtre, Wrapper et Hybride). Plusieurs méthodes de sélection de caractéristiques sont collectées dans le Tableau 2.3 selon cette catégorisation (voir la section 2.3.1 pour les abréviations utilisées).

2.5. Méthodes classiques de la sélection d'attributs

Un grand nombre de méthodes classiques de sélection d'attributs sont décrites dans la littérature, nous avons choisi d'en présenter certaines, parmi les plus connues, en les catégorisant selon la technique utilisée pour la procédure de génération de sous ensembles (voir section 2.3.1).

2.5.1. Méthodes complètes

Ces méthodes examinent toutes les combinaisons d'attributs possibles. Elles ont une complexité de calcul très élevée (espace de recherche de l'ordre de $O(2^N)$ pour N attributs) mais elles garantissent de trouver le sous ensemble optimal d'attributs. A titre d'exemples de ces méthodes, nous allons présenter : FOCUS [ALM 91], Branch and Bound [NAR 77], et Automatic Branch & Bound [LIU 98].

2.5.1.1. FOCUS

FOCUS est une méthode simple [ALM 91] qui utilise la mesure de l'incohérence pour évaluer les sous-ensembles d'attributs. Elle considère toutes les combinaisons d'attributs à partir d'un sous-ensemble vide, et l'évaluation des sous-ensembles avec un attribut, des sous-ensembles avec deux attributs, et ainsi de suite.

La réalisation de cet algorithme est donnée dans l'algorithme 2.1.

```
Entrées : F - l'ensemble complet d'attributs, INCONSIST () – mesure de l'incohérence
S←□

Pour i← 1 à |F| Faire

    Pour chaque sous-ensemble S de taille I Faire

        Si INCONSIST(S) = 0 retourner S

        Fin pour

    Fin pour

Sortie: S
```

Algorithme 2.1. L'algorithme FOCUS [CAK 02]

Cet algorithme doit générer $\sum_{i=1}^m \binom{N}{i} = \sum_{i=1}^m \frac{N!}{i!(N-i)!}$ sous-ensembles afin de trouver le sous-ensemble de m attributs (dans un ensemble de N attributs) qui satisfait la mesure de l'incohérence. Il est clair que le nombre de sous-ensembles d'attributs évalués est trop grand, même en cas de valeurs modérées de m .

2.5.1.2. Branch and Bound (BB)

Branch and Bound (BB) est l'un des algorithmes les plus anciens en sélection d'attributs [NAR 77]. C'est une variation de l'algorithme de recherche en profondeur d'abord (depth-first search) d'un treillis (Figure 2.6). La recherche en profondeur d'abord descend dans une branche jusqu'à ce que cela ne soit plus possible, puis fait un retour en arrière à une autre branche jusqu'à ce que toutes les branches soient vérifiées [CAK 02].

La méthode BB permet de restreindre la recherche et donner le sous ensemble optimal d'attributs, sous l'hypothèse de monotonicité du critère de sélection $J(X)$. Le critère $J(X)$ est dit monotone si :

$$x_1 \subset x_2 \subset \dots \subset x_m : J(x_1) < J(x_2) < \dots < J(x_m)$$

où x_k est l'ensemble contenant k attributs sélectionnés.

Cette contrainte limite, bien sûr, le choix de la mesure de distance. Généralement, celles qui sont utilisées sont : la distance de Mahalanobis, la distance de Bhattacharya, le critère de Fisher, la fonction discriminante et la divergence [GRA 03].

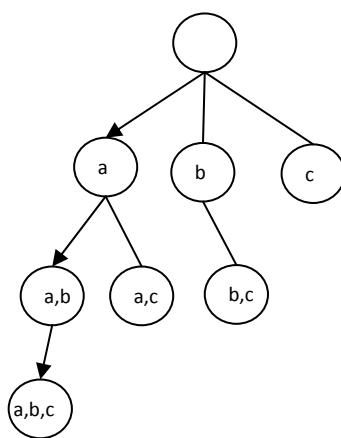


Figure 2.6 Depth-first search

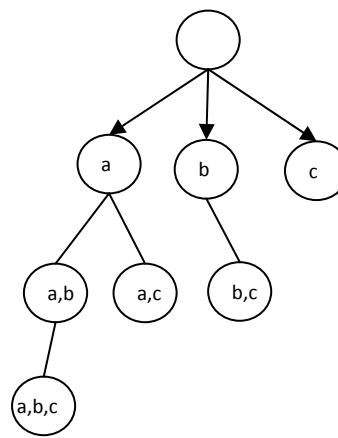


Figure 2.7 Breadth-first search

Supposant un seuil β (défini par l'utilisateur), la recherche s'arrête à chaque nœud dont la fonction d'évaluation est inférieure à β , de sorte que les branches efférentes soient taillées, et beaucoup de combinaisons d'attributs peuvent être omises de l'évaluation.

Entrées: F- l'ensemble complet de caractéristiques, $J()$ - une mesure

Nœud := F

Best := β

BB (nœud)

Pour tous les enfants C de Nœud **faire**

si $J(C) \geq \beta$ **alors**

si $J(C) > \text{Best}$ **alors**

$S := C$

 best := $J(S)$

Fin si

Fin si

Fin pour

Sortie: S

Algorithme 2.2. L'algorithme Branch and Bound (BB)[CAK 02]

L'utilisation de l'hypothèse de monotonicité dans la recherche du sous-ensemble optimal signifie que l'ajout de nouveaux attributs à un sous-ensemble d'attributs ne peut jamais diminuer la fonction d'évaluation. Mais nous savons que cela ne peut pas être vrai dans le cas des petites tailles d'échantillons, d'après le phénomène de dimension. En outre, l'utilisation de la méthode BB est encore impossible pour des problèmes avec des très grands ensembles d'attributs, car la pire complexité de cet algorithme est exponentielle.

Xu et al [XU 88] ont proposé un algorithme semblable (BFF), où la procédure de recherche est modifiée pour résoudre le problème de la recherche d'un chemin optimal dans un arbre pondérée. Cette recherche est effectuée avec la stratégie de recherche du premier meilleur informé d'abord (informed best first search strategy) en intelligence artificielle. Cet algorithme garantit le meilleur sous-ensemble global sans énumération exhaustive, pour n'importe quel critère satisfaisant le principe de monotonicité [DAS 97].

Foroutan et Sklansky [FOR 87] proposent une autre version de l'algorithme BB non-optimale mais permettant d'utiliser des critères d'évaluation non monotones. Pour cela ils définissent le concept de monotonicité approximative et proposent une version relaxée de cet algorithme (Relaxed Branch and Bound). Kudo et Sklansky proposent une autre variante en définissant le concept de k-monotonicité, permettant de violer ce critère sous certaines conditions [GRA 03].

2.5.1.3. Automatic Branch & Bound (ABB)

ABB (Automatic Branch & Bound) [LIU 98] est une procédure de recherche BB optimale automatisée qui utilise la stratégie backward. C'est une variante ayant sa limite comme le taux d'incohérence des données lorsque l'ensemble complet d'attributs est utilisé (Algorithme 2.3). Elle commence par l'ensemble complet d'attributs et supprime un attribut, à chaque fois, dans un algorithme de recherche en largeur d'abord (breadth-first search) (Figure 2.7) jusqu'à ce qu'aucun attribut ne puisse être enlevé alors que le critère de l'incohérence est respecté.

```
Entrées : S(X) – un échantillon S décrit par X, |X| = n, J – mesure d'évaluation (monotonique)
procédure ABB ( S(X) : échantillon ; var L' : liste des ensembles )
    Pour chaque x dans X Faire
        enqueue (Q, X \ {x}) // supprimer un attribut à chaque fois.
    Fin pour
    Tant que non vide (Q) Faire
        X' := dequeue (Q) // X' est légitime si elle n'est pas un sous-ensemble d'un état coupé
        Si légitime (X') et J(S(X')) ≥ J0 Faire
            L' := append (L', X')
            ABB(S(X'), L')
        Fin si
    Fin tant que
Fin
Début
    Q := □ // file d'attente d'états en instance
    L' := [X] // Liste des solutions
    J0 := J(S(X)) // la valeur minimale autorisée de J
    ABB (S(X), L') // appelle initial à ABB
    k := la plus petite taille d'un sous-ensemble dans L'
    L := ensemble d'éléments de L' de taille k
Fin
Sortie : L – toutes les solutions équivalentes trouvées
```

Algorithme 2.3. L'algorithme Automatic Branch and Bound (ABB) [BEL 11]

2.5.2. Méthodes heuristiques

La recherche s'effectue de sorte qu'il ne soit pas nécessaire d'évaluer l'ensemble de tous les sous-ensembles d'attributs possibles. Ainsi, une plus grande vitesse est atteinte, étant donné que l'espace de recherche est plus petit que dans le cas des méthodes complètes. Les méthodes heuristiques ne garantissent pas de trouver le sous-ensemble optimal, nous allons, à titre d'exemple présenter les méthodes Sequential Forward Selection (SFS), Sequential Backward Selection (SBS) et Relief [KIR 92].

2.5.2.1. Sequential Forward Selection (SFS):

Sequential Forward Selection est l'algorithme de recherche gloutonne (greedy) le plus simple, il a été proposé en 1963 par Marill [GRA 03]. A partir de l'ensemble vide, ajouter successivement le meilleur attribut X^+ qui est trouvé par l'augmentation de la fonction objective $J(Y_k + X^+)$ lorsqu'il est combiné avec les attributs Y_k qui ont déjà été sélectionnés.

1. Commencer avec un ensemble vide $Y_0 = \{\square\}$
2. Sélectionner le meilleur attribut $X^+ = \operatorname{argmax} [J(Y_k + X)]$; $x \notin Y_k$
3. Mettre à jour $Y_{k+1} = Y_k + X^+$; $k = k + 1$
4. Aller à 2

Algorithme 2.4. L'algorithme SFS [LAD 11]

SFS donne les meilleures performances lorsque le sous-ensemble optimal a un petit nombre d'attributs. Lorsqu'on est proche de l'ensemble vide, un grand nombre d'états peuvent être potentiellement évalués. Quand on est proche de l'ensemble complet, la région examinée par SFS est étroite car la plupart des attributs ont déjà été sélectionnés.

L'espace de recherche est dessiné comme une ellipse afin d'insister sur le fait qu'il y a moins d'états vers l'ensemble vide ou plein. Le principal inconvénient de SFS est qu'il est incapable de supprimer des attributs qui deviennent impertinents après l'ajout d'autres attributs.

2.5.2.2. Sequential Backward Selection (SBS):

Sequential Backward Selection (SBS) fonctionne dans le sens inverse de SFS. On parle aussi de Sequential Backward Elimination. A partir de l'ensemble complet, retirer séquentiellement l'attribut X^- qui est trouvé par la diminution de la valeur de la fonction objective $J(Y_k - X^-)$. Notons que la suppression d'un attribut peut en fait conduire à une augmentation de la fonction objective $J(Y_k - X^-) > J(Y_k)$. Ces fonctions sont dites non monotones.

SBS fonctionne mieux lorsque le sous-ensemble optimal a un grand nombre d'attributs, car SBS passe la plupart de son temps à visiter les grands sous-ensembles. La principale

limitation de SBS est son incapacité à réévaluer l'utilité d'un attribut après sa suppression.

1. Commencer avec l'ensemble complet $Y_0=X$
2. Enlever le mauvais attribut $X^-=\text{argmax}[J(Y_k-X)]$; $X^- \notin Y_k$
3. Mettre à jour $Y_{k+1}=Y_k - X^-$; $k=k+1$
4. Aller à 2

Algorithme 2.5. L'algorithme SBS [LAD 11]

2.5.2.3. Relief

Relief [KIR 92] est un algorithme basé sur les poids d'attributs. Cet algorithme est inspiré de l'apprentissage basé sur les exemples (instance-based learning). En tenant compte des données d'apprentissage S , de la taille de l'échantillon m , et d'un seuil de pertinence τ ($0 \leq \tau \leq 1$), Relief détecte les attributs qui sont statistiquement pertinents au concept cible. Nous supposons que le type de chaque attribut est soit nominal (y compris booléen) ou numérique (entier ou réel). La différence entre les valeurs d'attributs de deux instances X et Y est définie par la fonction suivante ;

$$diff(x_k, y_k) = \begin{cases} 0 & \text{si } x_k \text{ et } y_k \text{ sont identiques} \\ 1 & \text{si } x_k \text{ et } y_k \text{ sont différents} \end{cases}$$

dans le cas où x_k , et y_k sont numériques.

Dans le cas où x_k , et y_k sont nominales

$$diff(x_k, y_k) = (x_k, y_k)/nu_k$$

où nu_k est une unité de normalisation pour normaliser les valeurs de diff dans l'intervalle $[0, 1]$

L'algorithme choisit aléatoirement une instance $I \in S$ et trouve ses deux voisins les plus proches. Le premier est l'instance la plus proche de la même classe que I (near hit) et le deuxième est l'instance la plus proche d'une classe différente (near miss). L'idée sous-jacente est que l'attribut est le plus pertinent dans I s'il est le plus capable de séparer I de son near miss, et le moins capable de la séparer de son near hit. Le résultat est une version pondérée de l'ensemble d'attributs d'origine. L'algorithme pour un problème à deux classes est décrit dans l'algorithme 2.6.

Entrées : p – pourcentage d'échantillonnage; d – mesure de distance

$S(X)$ – un échantillon S est décrit par X , $|X| = n$

$m := p|S|$

Initialiser le tableau $w []$ à zéro

Faire m fois

$I :=$ Instance aléatoire (S)

$Inh :=$ Near–Hit (I, S)

$Inm :=$ Near–Miss (I, S)

Pour chaque $i \in [1..n]$ **Faire**

$w[i] := w[i] + di(I, Inm)/m - di(I, Inh)/m$

Fin pour

Fin

Sortie : w – tableau des poids d'attributs

Algorithme 2.6. Algorithme Relief [BEL 11]

2.5.3. Méthodes Aléatoires

Ces méthodes n'ont pas une manière spécifique pour générer le sous-ensemble d'attributs à analyser, mais utilisent des méthodes aléatoires. Ainsi, une recherche probabiliste est effectuée dans l'espace des caractéristiques. Les résultats obtenus en utilisant ces types de méthodes dépendent du nombre des itérations, sans garantir que le sous-ensemble optimal soit atteint. Les méthodes LVF et les algorithmes génétiques appartiennent à cette catégorie.

2.5.3.1. Las Vegas Filter et Las Vegas Wrapper

LVF (Las Vegas Filter) est un algorithme qui génère plusieurs fois des sous-ensembles aléatoires et calcule la consistance de l'échantillon: une incohérence dans X' et S est définie par deux instances dans S , qui sont égales lorsque l'on considère seulement les attributs de X' et qui appartiennent à des classes différentes. Le but est de trouver le sous-ensemble minimal d'attributs conduisant à des incohérences nulles [LAD 11].

LVF est décrit dans l'algorithme 2.7, il est particulièrement efficace pour des ensembles de données ayant des attributs redondants [BEL 11]. Cette méthode peut trouver le sous-ensemble optimal même pour les ensembles de données bruitées si le niveau de bruit est spécifié au début [DAS 97]. On peut dire que le principal avantage de LVF est

qu'il réduit rapidement le nombre d'attributs dans les premières étapes avec une certaine confiance [BEL 11]. Son inconvénient est qu'il peut prendre, dans certains problèmes, plus de temps pour trouver le sous-ensemble optimal, par rapport aux algorithmes heuristiques, car il ne peut pas profiter des connaissances antérieures [DAS 97].

Entrées :

max – le nombre maximum d'itérations,
 J – mesure d'évaluation
 $S(X)$ – un échantillon S décrite par X , $|X| = n$
 $L := []$ // L enregistre les meilleurs ensembles
 $Best := X$ // Initialiser la meilleure solution
 $J0 := J(S(X))$ // valeur minimum autorisée de J

Répéter max fois

$X' :=$ sous ensemble aléatoire ($Best$)
Si $J(S(X')) \geq J0$ **Faire**
 Si $|X'| < |Best|$ **Faire**
 $Best := X'$
 $L := [X']$ // L est réinitialisée

Sinon

Si $|X'| = |Best|$ **Faire**
 $L :=$ ajouter (L, X')

Fin si

Fin si

Fin si

Fin

Sortie :

L – toutes les solutions équivalentes trouvées

Algorithme 2.7. Algorithme LVF [BEL 11]

2.5.3.2. Les algorithmes génétiques pour la sélection d'attributs

Les algorithmes génétiques (voir section 1.3.1) sont des techniques de recherche adaptative basées sur les principes de la sélection naturelle en biologie. Ils sont considérés parmi les méthodes de sélection classiques de caractéristiques et sont largement appliqués dans ce domaine [ZHA 03, CHO 08].

Pour la sélection de caractéristiques, une solution est généralement codée sous forme d'une chaîne binaire de longueur fixe qui représente un sous-ensemble de caractéristiques. La valeur de chaque position dans la chaîne représente la présence ou l'absence d'une caractéristique particulière. L'algorithme de sélection est un processus itératif où chaque génération est produite par l'application d'opérateurs génétiques, tels que la mutation et le croisement sur les membres de la génération actuelle. La mutation modifie certaines valeurs (ajout ou suppression d'attributs) dans un échantillon aléatoire. Le croisement combine les différentes caractéristiques d'une paire de sous-ensembles dans un nouveau sous-ensemble.

L'application des opérateurs génétiques sur une population d'individus est déterminée par leur fitness (quel sous-ensemble de caractéristiques est le meilleur par rapport aux autres en utilisant une stratégie d'évaluation). La meilleure caractéristique d'un sous ensemble a plus de chances d'être sélectionnée pour former un nouveau sous ensemble par le biais du croisement ou de la mutation [HAL 99].

2.6. Méthodes de sélection d'attributs basées sur l'intelligence en essaims

Parmi les nombreuses méthodes qui sont proposées pour la sélection d'attributs, les algorithmes d'optimisation basés sur la notion de population tels que la méthode d'optimisation par les essaims particulaires (PSO) ou les colonies de fourmis (ACO) ont attiré beaucoup d'attention.

Cette section présente les principales approches de l'intelligence en essaims qui sont appliquées dans le domaine de la sélection des caractéristiques.

2.6. 1. Sélection d'attributs par les colonies de fourmis

Comme le problème de sélection d'attributs peut être considéré comme un problème d'optimisation, les colonies de fourmis (voir section 1.4) ont été largement appliquées pour la sélection de caractéristiques, et divers algorithmes basés sur l'ACO (Ant Colony Optimization) ont été présentés dans la littérature.

La recherche de l'espace de caractéristiques dans le problème de sélection d'attributs est différente de celle des problèmes traditionnels d'optimisation que les chercheurs ont

essayé de résoudre en utilisant l'ACO [ANI 05]. Voici les principales différences entre le problème de sélection de caractéristiques et celui du voyageur de commerce (TSP) :

- Dans TSP, une fourmi doit passer par toutes les villes, et son ordre de déplacement est une solution. Cependant, dans le problème de sélection d'attributs, une fois que l'ensemble des sommets C' parcourus par la fourmi satisfait l'objectif de la sélection, la fourmi arrête le déplacement, puis l'ensemble est une solution [DEN 09].
- Dans TSP, supposons que le nœud courant de la i ème fourmi est i , la i ème fourmi choisit le nœud suivant seulement à partir du nœud courant. Dans le problème de sélection d'attributs, il n'est pas nécessaire de le faire. C'est la raison pour laquelle la sélection d'attributs a comme résultat un sous-ensemble de tous les attributs, et seuls les attributs sélectionnés sont significatifs pour la sélection d'attributs, mais leur séquence de sélection n'est pas significative [DEN 09].
- Les distances entre les villes sont fixées dans le TSP. En ajoutant une ville en plus, le changement de la fonction objective est affecté seulement par la distance entre les deux dernières villes. Contrairement au TSP, ajouter une caractéristique à un sous-ensemble existant d'attributs peut avoir un impact sur la performance globale. Une caractéristique appropriée produira un meilleur sous-ensemble, et par conséquent améliorera la performance, alors qu'une caractéristique non pertinente peut dégrader la performance du sous-ensemble original. Lorsqu'on ajoute une caractéristique au sous-ensemble courant de caractéristiques, la mesure de performance locale devrait prendre en considération le rapport avec toutes les caractéristiques précédemment choisies et pas seulement les dernières. Ceci fait que le problème de la sélection de caractéristiques soit plus compliqué [KHU 08].

2.6.1.1 Représentation du problème de sélection de caractéristiques avec l'ACO

Généralement, un algorithme d'optimisation par les colonies de fourmis peut être appliqué à n'importe quel problème combinatoire dans la mesure où il est possible de définir les éléments suivants [JEN 03]:

- *Une représentation appropriée du problème* : Le problème doit être décrit comme un graphe avec un ensemble de nœuds et des arêtes entre les nœuds.
- *Une opportunité heuristique des arêtes* : Une mesure heuristique appropriée, de « goodness » ; des chemins qui lient un nœud à chaque autre nœud dans le graphe.
- *La construction des solutions faisables* : Un mécanisme doit être mis en place par lequel des solutions possibles sont efficacement créées.
- *La règle de mise à jour de phéromone* : Une méthode appropriée de mise à jour des niveaux de phéromones sur les arêtes est nécessaire avec une règle d'évaporation correspondante. Les méthodes typiques impliquent la sélection des meilleures n fourmis et la mise à jour des chemins qu'elles ont choisis.
- *La règle de transition probabiliste* : La règle qui détermine la probabilité qu'une fourmi traverse d'un nœud à l'autre dans le graphe.

La tâche de sélection de caractéristiques peut être reformulée en un problème ACO-convenable. ACO exige qu'un problème soit représenté comme un graphe. Ici, les nœuds représentent les caractéristiques, les arêtes entre eux indiquent le choix de la prochaine caractéristique. La recherche du sous-ensemble optimal de caractéristiques est alors un parcours des fourmis dans le graphe avec un nombre minimal de nœuds visités satisfaisant le critère d'arrêt du parcours.

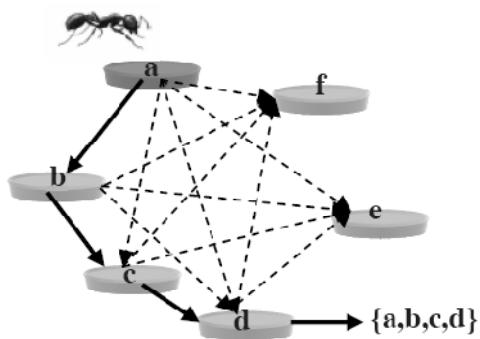


Figure 2.8. Représentation d'ACO pour le problème de sélection des caractéristiques

La Figure 2.8 illustre cette configuration. La fourmi est actuellement au nœud a et dispose d'un choix de la prochaine caractéristique qui peut être ajoutée à son chemin (lignes en pointillés). Elle choisit, par la suite, la caractéristique b (d'après la règle de

transition), la caractéristique c , puis la caractéristique d . A l'arrivée au nœud d , le sous-ensemble en cours $\{a; b; c; d\}$ satisfait le critère d'arrêt du parcours (par exemple, une précision de classification suffisamment élevée a été atteinte avec ce sous-ensemble). La fourmi termine son parcours et produit ce sous-ensemble de caractéristiques en tant que candidat pour la réduction de données.

Une opportunité heuristique appropriée de parcours entre les caractéristiques pourra être n'importe quelle fonction d'évaluation de sous-ensemble, par exemple, une mesure basée sur l'entropie, la mesure approximative de dépendance d'ensemble ou le taux de discrimination de Fisher (FDR). L'opportunité heuristique des niveaux de parcours et la phéromone d'une arête, sont combinées pour former ce qu'on appelle la règle probabiliste de transition, dénotant la probabilité d'une fourmi sur la caractéristique i choisissant à voyager à la caractéristique j à l'instant t par :

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in J_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta} \quad (2.2)$$

Où k est le nombre de fourmis, J_i^K l'ensemble des caractéristiques non visitées par la fourmi k , η_{ij} est l'opportunité heuristique de choisir la caractéristique j lorsque la fourmi est sur la caractéristique i et $\tau_{ij}(t)$ est la quantité de phéromone virtuelle sur l'arête (i, j) . Les valeurs de α et β sont choisies expérimentalement.

Selon la façon dont l'optimalité est définie par rapport à l'application considérée, la phéromone peut être mise à jour en conséquence. Par exemple, dans le problème de sélection d'attributs, la minimisation de taille et la 'goodness' du sous-ensemble sont deux facteurs clés de sorte que la mise à jour de phéromone doive être proportionnelle à la goodness et inversement proportionnelle à la taille.

2.6.1.2. Le processus global de sélection d'attributs avec ACO

Le processus global de sélection d'attributs avec ACO peut être représenté par la Figure 2.9. Le processus commence par générer un certain nombre de fourmis k , qui sont ensuite placées au hasard sur le graphe (chaque fourmi commence avec une configuration aléatoire).

Alternativement, le nombre de fourmis à placer sur le graphe peut être défini comme étant le nombre de caractéristiques dans les données; chaque fourmi commence la

construction du chemin à une caractéristique différente. A partir de ces positions initiales, elles parcourront les arêtes probabilistes jusqu'à ce qu'un critère d'arrêt de parcours soit satisfait. Les sous-ensembles résultants sont recueillis et évalués. Si un sous-ensemble optimal a été trouvé ou que l'algorithme a été exécuté un certain nombre de fois, alors le processus s'arrête et donne le meilleur sous-ensemble d'attributs produit. Si aucune de ces conditions n'est vérifiée, alors la phéromone est mise à jour, un nouvel ensemble de fourmis est créé et le processus réitère, une fois de plus.

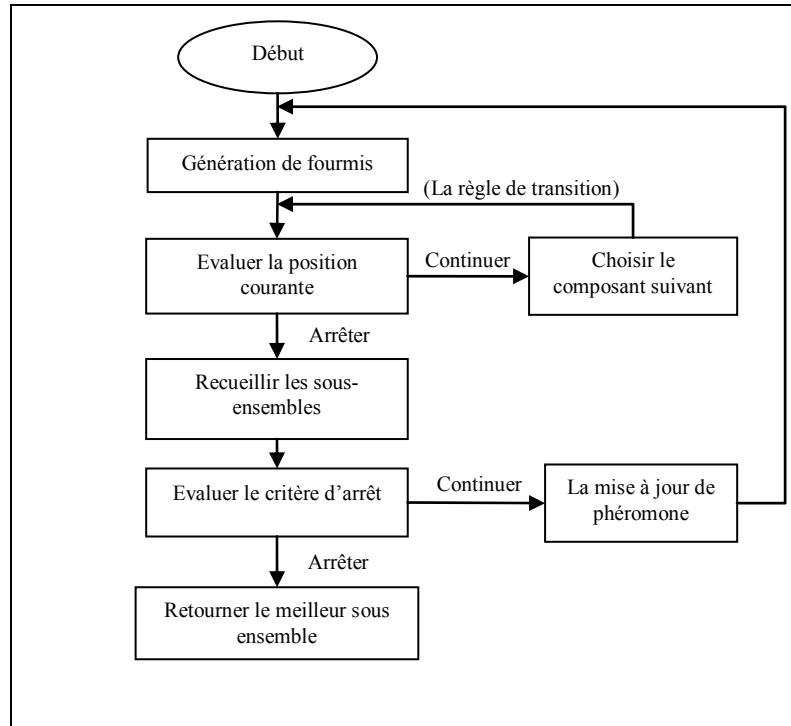


Figure 2.9. Processus général de sélection d'attributs basée sur ACO [KAN 07]

2.6.2. Sélection d'attributs par les essaims particulaires

L'algorithme PSO original d'optimisation par les essaims de particules (voir section 1.4.2) a été essentiellement développé pour des problèmes d'optimisation continue. Cependant, beaucoup de problèmes pratiques sont formulés comme des problèmes d'optimisation combinatoire (voir section 1.2). Le terme «PSO binaire» ou «PSO discret» est apparu quand PSO a été utilisé pour résoudre des problèmes discrets [ZAI 07].

Kennedy et Eberhart ont développé, pour ces problèmes, une version binaire discrète de PSO (DPSO) [KEN 97], dans laquelle les particules prennent comme valeurs des vecteurs binaires de longueur N et la vitesse est interprétée comme une probabilité de changer un bit de 0 à 1, ou de 1 à 0 lors de la mise à jour de position des particules [CHI 09].

Pour appliquer le principe de PSO à la sélection de caractéristiques, certaines questions doivent d'abord être prises en considération, telles que la représentation et la mise à jour de la position ainsi que la vitesse d'une particule.

2.6.2.1. Représentation de la position

Soit N le nombre total de caractéristiques, l'espace de recherche est binaire (une caractéristique est 'on' ou 'off') et de N dimensions. La position de la particule représente les caractéristiques qui sont sélectionnées. Dans l'approche proposée dans [XIA 05] et [WAN 07], la position est codée comme une chaîne binaire de longueur N , où N est le nombre total de caractéristiques. Chaque bit représente une caractéristique, la valeur «1» signifie que la caractéristique correspondante est sélectionnée tandis que «0» signifie qu'elle n'est pas sélectionnée. Chaque position est un sous-ensemble de caractéristiques.

2.6.2.2. Représentation de la vitesse

La vitesse de chaque particule est représentée comme un nombre entier positif, variant entre 1 et V_{max} . Elle implique que le nombre de bits de la particule (caractéristiques) devrait être modifié, à un moment précis dans le temps, pour être le même que celui de la meilleure position globale, c'est à dire la vitesse de la particule qui se déplace vers la meilleure position. Le nombre de bits différents entre deux particules se rapporte à la différence entre leurs positions. Voir la Figure 2.10 pour le principe de la mise à jour de vitesse.

Par exemple, $P_{gbest} = [1 0 1 1 1 0 1 0 0 1]$, $P_i = [0 1 0 0 1 1 0 1 0 1]$. La différence entre $gbest$ et la position actuelle de la particule est $P_{gbest} - P_i = [1 -1 1 1 0 -1 1 -1 0 0]$. Une valeur de 1 indique que, par rapport à la meilleure position, ce bit (caractéristique) devrait être choisi, mais il ne l'est pas, ce qui diminue la qualité de la classification et conduit à une valeur inférieure à la valeur de fitness. D'autre part, une valeur de -1 indique que, en comparant à la meilleure position, ce bit ne devrait pas être sélectionné,

mais il est sélectionné. Les caractéristiques redondantes superflues augmentent la longueur du sous-ensemble et conduisent à une valeur de fitness inférieure. Supposons que le nombre de 1 est a et le nombre de -1 est b. Nous utilisons la valeur de (a-b) pour exprimer la distance entre deux positions; (a-b) peut être positif ou négatif. Cette variation rend les particules présentant une capacité d'exploration dans l'espace des solutions. Dans cet exemple, (a- b) = 4-3 = 1, alors $P_g - P_i = 1$.

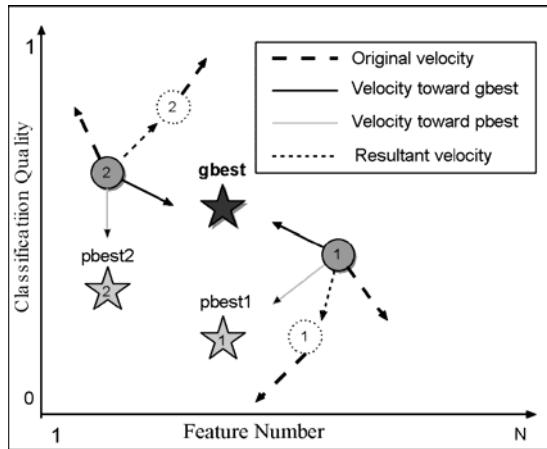


Figure 2.10. Le principe de mise à jour de la vitesse.

Dans la Figure 2.10, les particules individuelles (1 et 2) sont accélérées vers l'emplacement de la meilleure solution « gbest », et l'emplacement de leur propre meilleure solution personnelle « pbest », dans l'espace des solutions à deux dimensions.

2.6.2.3. Les stratégies de mise à jour de la position et la vitesse

Lors de la mise à jour de la position des particules. La formule de mise à jour de vitesse (Equation 1.17, Section 1.4.2) reste inchangée, mais elle est contrainte à l'intervalle [0.0, 1.0]. Toutefois, la formule de mise à jour de position pour DPSO est remplacée par l'équation probabiliste définie comme suit:

$$x_{i,j}(t+1) = \begin{cases} 1 & \text{si } \rho_j(t) < \text{sig}(v_{i,j}(t+1)) \\ 0 & \text{sinon} \end{cases} \quad (2.3)$$

Où $\rho_j(t)$ est un nombre quasi aléatoire choisi parmi une distribution uniforme dans [0,0, 1,0], alors que $\text{sig}(\bullet)$ est une fonction de transformation de limitation sigmoïde définie par [CHI 09] :

$$\text{sig}(x) = \frac{1}{1 + \exp(-x)} \quad (2.4)$$

2.7. Applications de la sélection d'attributs

Comme beaucoup de systèmes, dans une vaste variété de domaines, traitent de grandes quantités de données, la sélection d'attributs a trouvé une très large applicabilité.

Parmi les principaux domaines d'application nous pouvons citer la fouille de données de manière générale et la catégorisation de textes en particulier, la reconnaissance de formes telle que l'écriture et les images ainsi que le domaine de la bioinformatique [JEN 05].

2.7.1. La fouille de données

La fouille de données (data mining) dans de très grandes bases devient un enjeu crucial pour des applications telles que le génie génétique, la finance, les études de marché, les processus industriels complexes, etc.. Il s'agit, en fait, de résumer et d'extraire intelligemment de la "connaissance" à partir de données brutes. La fouille de données est un domaine basé sur les statistiques, l'apprentissage automatique et la théorie des bases de données.

La sélection d'attributs joue un rôle important dans le data mining, en particulier dans la préparation des données avant leur traitement. En effet, les intérêts de la sélection d'attributs sont les suivants :

- Lorsque le nombre de variables est vraiment trop grand (il peut aller jusqu'à plusieurs dizaines de milliers dans certaines applications), l'algorithme d'apprentissage ne peut pas se terminer en un temps convenable.
- La sélection réduit la dimension de l'espace d'attributs, augmente la compréhensibilité des données et améliore les performances de la classification, sa vitesse et son pouvoir de généralisation.

2.7.2. La catégorisation de textes

La catégorisation des textes consiste à affecter automatiquement des catégories prédéfinies à des documents de textes libres. La catégorisation des textes considère les documents comme une collection de mots. Les documents sont examinés, avec leurs mots-clés constitutifs extraits et classés en fonction de critères tels que leur fréquence d'apparition. Comme le nombre de mots clés extraits est généralement de l'ordre de plusieurs dizaines de milliers, la réduction de la dimensionnalité doit être effectuée en

utilisant une méthode de filtrage simple tel que la suppression des mots vides. Toutefois, ceux-ci ne fournissent pas assez de réduction pour l'utilisation d'une catégorisation automatique, raison pour qu'un nouveau processus de sélection d'attributs soit utilisé. Plusieurs travaux ont été présentés dans la littérature, utilisant des méthodes de sélection d'attributs classiques telles que la fréquence d'apparitions des termes dans la collection, l'information mutuelle, et le gain d'information ainsi que des méthodes plus récentes de sélection d'attributs pour la catégorisation de textes [AGH 09, MEE 12].

Les applications récentes de la sélection d'attributs dans ce domaine comprennent, entre autres, la catégorisation des pages web (l'indexation des pages), le text mining et le filtrage des spams [GAV 06, MEN 08, GUZ 09].

2.7.3. La reconnaissance de l'écriture.

Dans le domaine de la reconnaissance de l'écriture, les attributs peuvent être décrits comme un moyen permettant de distinguer une entité (mot, lettre, chiffre) appartenant à une classe d'une autre entité (mot, lettre, chiffre) appartenant à une autre classe. Il est nécessaire de définir des attributs significatifs lors du développement d'un système de reconnaissance. Plusieurs types d'attributs peuvent être extraits et sont généralement représentés de manière vectorielle. La taille du vecteur peut être très grande en fonction du nombre d'attributs extraits. En plus, ces derniers ne sont pas tous informatifs: certains attributs peuvent correspondre à un bruit, être peu significatifs, corrélés ou non pertinents pour la tâche à réaliser. Pour cela, il est fondamental de mettre en place des outils de traitement de données permettant une meilleure utilisation des connaissances disponibles dans ces données. Ainsi, la sélection d'attributs pertinents est devenue une étape importante dans tous les systèmes de reconnaissance de l'écriture [GRA 03; COR 08] et précisément la reconnaissance des chiffres [OLI 01; OLI 03a; BEN 02; SIL 08].

2.7.4. La reconnaissance d'images

La reconnaissance d'image est le processus d'identification des groupes de caractéristiques d'images similaires. Ces caractéristiques d'image peuvent être des pixels, des régions, des éléments de ligne et ainsi de suite, cela en fonction du problème rencontré [OMR 04]. Un des plus grands problèmes de la procédure de reconnaissance d'images avec des grandes tailles est toujours le «problème de la dimension». La dimension de l'espace d'attributs est normalement de l'ordre de 10^2 . La réduction de

dimension est une approche prometteuse pour résoudre ce problème, et les algorithmes de sélection d'attributs sont souvent appliqués pour optimiser les performances de classification des systèmes de reconnaissance d'images [SIN 02, KAN 07, KAN 08].

2.7.5. La bioinformatique

Au cours de la dernière décennie, la motivation pour l'application des techniques de la sélection d'attributs dans le domaine de la bioinformatique a dépassé le fait d'être un exemple illustratif pour devenir une véritable condition préalable à la construction des modèles. Deux questions principales se posent comme des problèmes communs dans le domaine de la bioinformatique: la grande dimension des entrées, et la petite taille des échantillons. Dans de nombreux problèmes bioinformatiques, tels que la classification de cancer et l'analyse de corrélation de séquences/expressions, le nombre d'attributs atteint plusieurs milliers (par exemple dans des tâches de classification du cancer le nombre d'attributs -à savoir le nombre de gènes dont l'expression est mesurée- peut aller de 6000 jusqu'à 60000). Pour faire face à ces problèmes, plusieurs techniques de sélection d'attributs ont été conçues par des chercheurs en bioinformatique, en apprentissage automatique et en extraction de connaissances. Les principales contributions de la recherche en sélection d'attributs dans un ensemble d'applications bioinformatiques bien connues sont présentées dans [SAE 07, AWA 12].

2.8. Conclusion

La sélection d'attributs est un domaine de recherche très actif, incluant un grand nombre d'algorithmes appliqués dans plusieurs domaines. Ce chapitre décrit les concepts et méthodes de la sélection de caractéristiques en se concentrant, vers sa fin, sur les méthodes de l'intelligence en essaims. Nous avons terminé le chapitre par quelques domaines d'application de la sélection d'attributs.

Nous avons défini la sélection d'attributs, présenté les différentes composantes nécessaires pour un algorithme de sélection de caractéristiques, et les alternatives possibles pour leurs mises en œuvre. Un certain nombre d'algorithmes ont été décrits en fonction de ces composantes (la fonction d'évaluation et la stratégie de recherche). Ensuite nous avons présenté une brève description de la manière avec laquelle les algorithmes de l'intelligence en essaims peuvent être utilisés pour la sélection d'attributs.

CHAPITRE 3

APPROCHES CLASSIQUES ET BIO-INSPIREES POUR LA SELECTION D'ATTRIBUTS

Ce chapitre présente une partie de nos contributions dans le cadre de la sélection d'attributs. L'objectif final de notre travail, dans ce cadre, est de proposer une approche de sélection basée sur une hybridation de deux méthodes bio-inspirées. Avant d'envisager l'hybridation, nous avons commencé par la proposition d'approches classiques et bio-inspirées non hybrides que nous avons appliquées, en premier lieu, au filtrage de spams. Ce chapitre présente une synthèse des méthodes et travaux existants sur la sélection bio-inspirée d'attributs avec notre proposition de deux approches bio-inspirées non hybrides de type wrapper (voir section 2.4.1), en plus d'une approche classique de sélection basée sur l'ordonnancement des attributs (feature ranking) qui intègre l'approche filtre avec wrapper (voir section 2.4.1). Dans les trois approches proposées, l'évaluation des sous ensembles sélectionnés est effectuée avec un classifieur de type naïve bayes.

Nous commençons, dans la section 2, par présenter un état de l'art des approches proposées dans la littérature, pour la sélection d'attributs, basées sur des méthodes bio-inspirées non hybrides telles que les algorithmes génétiques (AGs), les colonies de fourmis (ACO) et les essaims particulaires (PSO). Dans les sections 3, 4 et 5, nous décrivons les approches qui nous avons proposées en commençant par l'approche classique, puis l'approche basée sur ACO et finalement l'approche basée sur PSO. Dans la section 6, nous présentons l'évaluation de ces approches et nous comparons les résultats obtenus.

3.1. Introduction

Récemment, la tendance des procédures de recherche dans le domaine de l'optimisation, est basée sur l'intelligence des essaims ou Swarm Intelligence. Depuis 1990, plusieurs

algorithmes inspirés de comportements naturels collectifs ont été proposés. Les domaines d'application de ces algorithmes se réfèrent aux problèmes d'optimisation identifiés comme NP-difficiles : le routage du réseau, le clustering, l'extraction de données, la planification des tâches, etc. [GRO 06]. Les algorithmes de l'intelligence en essaims, en particulier l'optimisation par les colonies de fourmis et l'optimisation par les essaims de particules, sont actuellement les algorithmes les plus populaires utilisés pour résoudre des problèmes d'optimisation. Comme nous l'avons mentionné précédemment (voir section 2.6), la sélection de caractéristiques peut être considérée comme un problème d'optimisation avec de nombreux critères concurrents, qui peuvent être la réduction du nombre de caractéristiques sélectionnées ou la maximisation du pourcentage d'échantillons d'apprentissage classées correctement (amélioration de l'erreur d'apprentissage).

3.2. Sélection d'attributs basée sur des méthodes bio-inspirées non hybrides

Dans cette section, nous présentons une étude de quelques travaux récents qui proposent des méthodes de sélection d'attributs (Feature Selection : FS) basées sur les algorithmes génétiques et des algorithmes issus de l'intelligence en essaims. Ces deux catégories de méthodes peuvent être considérées comme étant les principales tendances de la recherche dans le domaine de la sélection bio-inspirée d'attributs.

3.2.1. Sélection d'attributs avec les algorithmes génétiques

Les algorithmes génétiques sont largement appliqués, depuis des années, dans le domaine de la sélection d'attributs [ZHA 03, CHO 08]. Dans [CHO 08], les auteurs proposent une méthode rapide de sélection de caractéristiques basée sur un algorithme génétique. L'évaluation des individus dans l'algorithme génétique se fait par une fonction de “fitness” basée sur la combinaison de classificateurs entraînés par Adaboost pour chacune des caractéristiques. Cette méthode est implémentée et testée sur la base des images de chiffres manuscrits MNIST et les résultats montrent la robustesse de cette approche ainsi que ses performances. En moyenne, le nombre d'attributs est divisé par deux sans diminuer les taux de reconnaissance des chiffres.

En outre, dans la résolution de la tâche de sélection d'attributs, certains auteurs ont présenté des algorithmes génétiques hybrides (HGAs) où différentes stratégies ont été introduites dans leurs opérations de recherche locale.

Kabir et al [KAB 09] proposent une nouvelle approche de recherche locale en HGA pour la sélection d'attributs, qui est basée sur l'observation de l'espace de caractéristiques. Le calcul de l'information de corrélation des caractéristiques a été utilisé pour reconnaître lesquelles sont distinctes et informatives. Elles ont été utilisées plus tard dans l'opération de recherche locale de HGAFS. Ainsi, l'opération de recherche locale proposée contribue à réduire la redondance d'information dans le sous-ensemble généré. En revanche, un plan aléatoire restreint a été constitué pour décider du nombre de 1-bits dans le chromosome individuel d'un ensemble de populations.

Les algorithmes génétiques sont largement appliqués pour la sélection de caractéristiques dans le domaine de la reconnaissance de l'écriture manuscrite. Oliveira et al. [OLI 01] comparent les résultats de deux approches basées, l'une sur un algorithme génétique simple, et l'autre sur un algorithme génétique itératif. Ils ont trouvé que le premier était plus convenable que le second dans la sélection d'attributs pour la reconnaissance des chiffres.

Benahmed [BEN 02] propose une méthode d'optimisation pour la sélection et la pondération des caractéristiques d'un système de reconnaissance de chiffres manuscrits isolés, basé sur les algorithmes génétiques dans le but d'optimiser un système de reconnaissance de l'écriture manuscrite.

Oliveira et Sabourin [OLI 03a] proposent un algorithme génétique multi-objectifs où l'analyse de sensibilité et les réseaux de neurones sont utilisés pour permettre l'utilisation d'une base de données représentatives afin d'évaluer la fonction d'adaptation, et l'utilisation d'une base de données de validation pour identifier les sous-ensembles de caractéristiques choisies qui permettent une bonne généralisation.

Oliveira et al [OLI 03b] proposent une approche de sélection d'attributs basée sur un algorithme génétique multi-objectif hiérarchique. Le premier niveau de la hiérarchie effectue une sélection de caractéristiques afin de générer un ensemble de bons classificateurs, tandis que le second combine les bons classificateurs pour fournir un ensemble des ensembles puissants.

Silva et al. [SIL 08] présentent un algorithme génétique qui intègre un classifieur de type SVM (Support Vector Machine) avec un noyau à base de fonction radiale. L'approche proposée effectue une sélection d'attributs appliquée sur des chiffres manuscrits extraits à partir d'images de documents historiques.

3.2.2. Sélection d'attributs basée sur ACO

Les colonies de fourmis ont été largement appliquées pour la sélection de caractéristiques, et divers algorithmes basés sur ACO ont été présentés dans la littérature. Certains d'entre eux ont utilisé des techniques hybrides (filtre et wrapper) pour estimer l'information heuristique et la performance globale. A titre d'exemple, Al-Ani [ANI 05] a proposé un algorithme de sélection de caractéristiques, basé sur ACO, qui estime les valeurs de phéromone par le biais de la mesure d'information mutuelle, et la performance globale à l'aide d'un classifieur neuronal. La méthode a été testée sur deux problèmes de classification (classification de segments de parole et classification de texture) et a réalisé des résultats supérieurs à une approche de sélection de caractéristiques basée sur les algorithmes génétiques.

Dans [BAS 08], une approche ACO est utilisée pour la prévision d'activité post-synaptique dans les protéines. La performance de classification a été utilisée comme une information heuristique pour l'ACO dans les expérimentations de Kabir et al [KAB 09] et celles de Kanan et al [KAN 07].

Une autre application est présentée dans [KAN 08], les auteurs ont adopté la performance du classifieur et la longueur du vecteur de caractéristiques sélectionnées en tant qu'information heuristique pour l'ACO, et ils les ont utilisées pour l'évaluation des performances. Ils ont particulièrement mis en œuvre l'algorithme proposé en utilisant les variantes ACS et ASrank. Pour montrer l'intérêt de leur algorithme, deux ensembles d'expériences ont été effectuées dans un système d'identification de visages.

3.2.3. Sélection d'attributs basée sur PSO

Certains chercheurs ont opté pour une sélection d'attributs basée sur PSO pour des applications diverses. Par exemple, Liu et al. [LIU 04] utilisent PSO afin de sélectionner un sous-ensemble de caractéristiques pour la tâche de classification et pour construire des réseaux de neurones RBF simultanément. Ye et al. [YE 07] ont utilisé un algorithme

de PSO binaire amélioré combiné avec un mécanisme de l'immunité pour résoudre un problème d'optimisation transformé. Dans [AKI 08], les auteurs implémentent PSO pour sélectionner le sous-ensemble puissant de caractéristiques rectangulaires non limitrophes (Non-Neighboring Rectangular Features NNRFs)

En outre, Chuang et al. [CHU 08] ont utilisé un algorithme binaire amélioré d'optimisation par les essaims de particules (IBPSO) pour la sélection d'attributs, et la méthode de K plus proches voisin (K-NN) comme un évaluateur de l'algorithme IBPSO dans un problème de classification d'expression des gènes.

Niiniskorpi et al. [NII 09] prouvent que le PSO est efficace dans l'identification des sous-ensembles de voxels (pixels en 3D) de haute performance pour la formation d'images de résonance magnétique fonctionnelle (functional magnetic resonance imaging IRMf) de classification de volume.

3.2.4. Sélection d'attributs avec les algorithmes inspirés des abeilles

Les algorithmes inspirés des abeilles représentent une tendance qui a émergé dans le domaine de l'intelligence en essaims au cours des dernières années. Leur utilisation dans la sélection d'attributs est très récente par rapport aux autres algorithmes bio-inspirés.

Dans [SHR 11], les auteurs utilisent une approche wrapper et un algorithme inspiré des abeilles pour trouver le sous ensemble optimal des caractéristiques. La méthode proposée emploie les réseaux de neurones multicouches (MLP) comme classificateurs pour guider la sélection des caractéristiques. Les auteurs ont optimisé la conception du classifieur par la sélection des meilleures caractéristiques à partir de la combinaison d'un ensemble de caractéristiques statistiques et spectrales.

Dans [PRA 12], les auteurs introduisent la méthode des colonies d'abeilles artificielles (ABC) comme une méthode de sélection d'attributs pour améliorer la performance des processus de réduction de dimension. Les SVM sont utilisés pour évaluer la valeur de la fitness d'ABC.

3.2.5. Synthèse de travaux sur la sélection d'attributs avec des méthodes bio-inspirées

Le Tableau 3.1 synthétise certains des travaux sur la sélection d'attributs avec des méthodes bio-inspirées non hybrides. Les deux premières colonnes contiennent des informations sur les auteurs et la référence du travail en question, alors que les 3ème et 4ème colonnes indiquent les méthodes bio-inspirées utilisés pour la sélection de caractéristiques et l'approche de sélection appliquée (voir section 2.4.1) respectivement. La 5ème colonne présente le domaine d'application ou les bases de données utilisées pour les expérimentations. La présence de l'astérisque (*) indique que la base de données est disponible dans l'UCI machine learning repository (<http://archive.ics.uci.edu/ml/datasets.html>). La dernière colonne donne les résultats de ces travaux en termes de nombre de caractéristiques sélectionnées: l'ensemble d'origine représente l'ensemble initial des caractéristiques et FS représente la taille du sous-ensemble optimal sélectionné par la méthode considérée.

Auteurs	Référence	Méthode bio-inspirée	Approche de FS	Domaine d'application ou noms des bases de données	Résultats		
					Ensemble d'origine	FS	
Oliveira et al.	[OLI 01]	AG simple	Wrapper	Reconnaissance des chiffres manuscrits	132	95	
		AG itératif			132	104	
Benahmed	[BEN 02]	AGs	Filtre	Reconnaissance des chiffres manuscrits	132	100	
Oliveira et al.	[OLI 03a]	AGs multi-objectifs	Wrapper	Reconnaissance des chiffres manuscrits	132	100	
Oliveira et al.	[OLI 03b]	AG multi-objectifs hiérarchique	Wrapper	Reconnaissance des chiffres manuscrits	132	24-125	
					96	30-84	
					125	35-113	
Zhang et al.	[ZHA 03]	AG	Wrapper	Le cancer du sein	14	3	
Silva et al.	[SIL 08]	AG intégré SVM avec fonction à base radiale	Wrapper	Reconnaissance des chiffres manuscrits	1024	511	
Chouaib et al.	[CHO 08]	AG	Wrapper	Base des images de chiffres manuscrits MNIST	RS	180	79
					ZER	47	36
					Pixels	1024	514
					RS+ZER	227	101
					RS+Pixels	1204	585
					ZER+Pixels	1071	524
Kabir et al.	[KAB 09]	AG hybride	Filtre	Breast cancer* Glass* Vehicle* Sonar*	9	≈ 3.25	
					9	≈ 3.45	
					18	≈ 4.40	
					60	≈ 5.35	

Tableau 3.1. Synthèse de travaux sur la sélection d'attributs basée sur des méthodes bio-inspirées non hybrides.

Auteurs	Référence	Méthode bio-inspirée	Approche de FS	Domaine d'application ou noms des bases de données	Résultats	
					Ensemble d'origine	FS
Jensen et Shen	[JEN 03]	ACO	Wrapper	Vote *	16	8
				Mushroom *	22	4
				Letters *	25	8
				Lung *	56	4
Al-Ani	[ANI 05]	ACO	Wrapper	Classification de segments de parole	152	64
				Classification de texture	50	20
Kanan et al.	[KAN 07, KAN 08]	ACS	Wrapper	La reconnaissance de visages (avec les attributs PZMI)	-	49
		ASrank		La reconnaissance de visages (avec les attributs DWT)	168	42
		ACS				42
		ASrank				30
Basiri et al.	[BAS 08]	ACO	Wrapper	Prédiction d'activité post-synaptique dans les protéines	443	5
Deng et al.	[DEN 09]	ACO	Wrapper	Dermatologie	34	16
				SPECT Heart	22	2
				Le cancer du poumon	56	49
				séquences de gènes	58	51
				Page-blocks *	10	2
Kabir et al.	[KAB 09]	ACO	Wrapper	breast cancer *	9	3
				Glass *	9	3
				Thyroid *	21	2
				Sonar *	60	6
Chen et al.	[CHE 10]	ACO	Filtre	Audiology *	70	4
Kabir et al.	[KAB 12]	ACO	Wrapper + filtre	Cancer *	9	3
				Vehicle*	18	3
				Thyroid*	21	3
				Ionosphere*	34	4
				Credit card*	51	6
				Sonar *	60	6
				Gene*	120	7
				Colon cancer*	100	5
Meena et al.	[MEE 12]	ACO	Wrapper	Catégorisation de textes	Entre 10% et 20% des caractéristiques sont sélectionnées	

Tableau 3.1. (suite) Synthèse de travaux sur la sélection d'attributs basée sur des méthodes bio-inspirées non hybrides.

Auteurs	Référence	Méthode bio-inspirée	Approche de FS	Domaine d'application ou noms des bases de données	Résultats	
					Ensemble d'origine	FS
Liu et al.	[LIU 04]	PSO	Wrapper	Wine *	13	4
				Ionosphere *	34	8
				Image segmentation *	30	5
Wang et al.	[WAN 05]	PSO	Wrapper	Breast cancer *	9	4
				Vote*	16	8
				Zoo *	16	5
				Lung *	56	4
Ye et al	[YE 07]	PSO	Wrapper	Zoo*	16	5
				House*	14	4
				Lymphography	18	8
				Soybean*	35	9
				Lung cancer*		4
Akinori et Takio	[AKI 08]	PSO	Wrapper	Classification des images	5 ⁴	13 ⁸
Chuang	[CHU 08]	PSO	Wrapper	9-Tumors	5726	22%
				11-Tumors	12533	24%
				14-Tumors	15009	19%
				Brain-Tumor1	10365	13%
				Brain-Tumor2	5327	12%
				LeuKemia1	5327	19%
				LeuKemia2	11225	12%
				Lung-cancer	12600	15%
				Prostate-Tumor	10509	12%
Niiniskorpi et al.	[NII 09]	PSO	Wrapper	Classification de l'état du cerveau à partir de données 2D de l'IRMf	5650	64.8
				Classification de l'état du cerveau à partir de données 3D de l'IRMf	61078	89
Lai et al.	[LAI 09]	PSO	Filtre	Filtrage de spams	12800	46
Bae et al.	[BAE 10]	PSO	Wrapper	Zoo *	16	5
				Mushroom *	22	4
				Breast cancer *	9	4
				Lung *	56	4
Liu et al.	[LIU 11]	PSO	Wrapper	Australian *	14	9
				Breast cancer *	9	5
				Hill-valley *	100	41

Tableau 3.1. (suite) Synthèse de travaux sur la sélection d'attributs basée sur des méthodes bio-inspirées non hybrides.

Auteurs	Référence	Méthode bio-inspirée	Approche de FS	Domaine d'application ou noms des bases de données	Résultats	
					Ensemble d'origine	FS
Bing et al.	[BIN 13]	PSO	Wrapper	Zoo*	17	7
				Lung *	56	6
				Vehicle*	18	10
				Wine *	13	7
				Mushroom *	23	4
				Vote *	17	9
Shrme	[SHR 11]	BA	Wrapper	Reconnaissance de signaux de communication automatique	31	7
Prasartvit et al.	[PAR 12]	ABC	Wrapper	Ionosphere *	34	13
				Sonar *	60	31
				Vowel *	10	8
				Wine *	13	6
				WDBC *	30	13

Tableau 3.1. (suite) Synthèse de travaux sur la sélection d'attributs basée sur des méthodes bio-inspirées non hybrides.

3.3. Approche classique proposée pour la sélection d'attributs

Dans nos premiers travaux dans le cadre du filtrage de spams [MEN 10a, MEN10b, MEN 14], nous avons proposé une approche de sélection d'attributs basée sur le principe du Feature Ranking (ordonnancement de caractéristiques) utilisé dans les méthodes classiques de sélection de caractéristiques (voir section 2.4.1).

Nous avons proposé une démarche de sélection de variables suivant un processus à deux étapes (voir Figure 3.1): la première étape effectue un filtrage des caractéristiques avec une méthode de type feature ranking (évaluation individuelle des caractéristiques), et la deuxième étape évalue le sous ensemble sélectionné selon l'approche wrapper ou symbiose.

La procédure de sélection est une procédure itérative. A chaque itération, on calcule le poids de chaque caractéristique à l'aide d'une mesure d'évaluation. Ces poids sont utilisés pour ordonner puis sélectionner les f caractéristiques les plus pertinentes parmi toutes les caractéristiques (f étant le nombre des caractéristiques utilisées dans

l’itération). Ensuite, cet ensemble sera évalué par un classifieur. Si cet ensemble ne donne pas la solution recherchée, nous choisissons un nouveau sous ensemble et nous réitérons l’algorithme d’apprentissage.

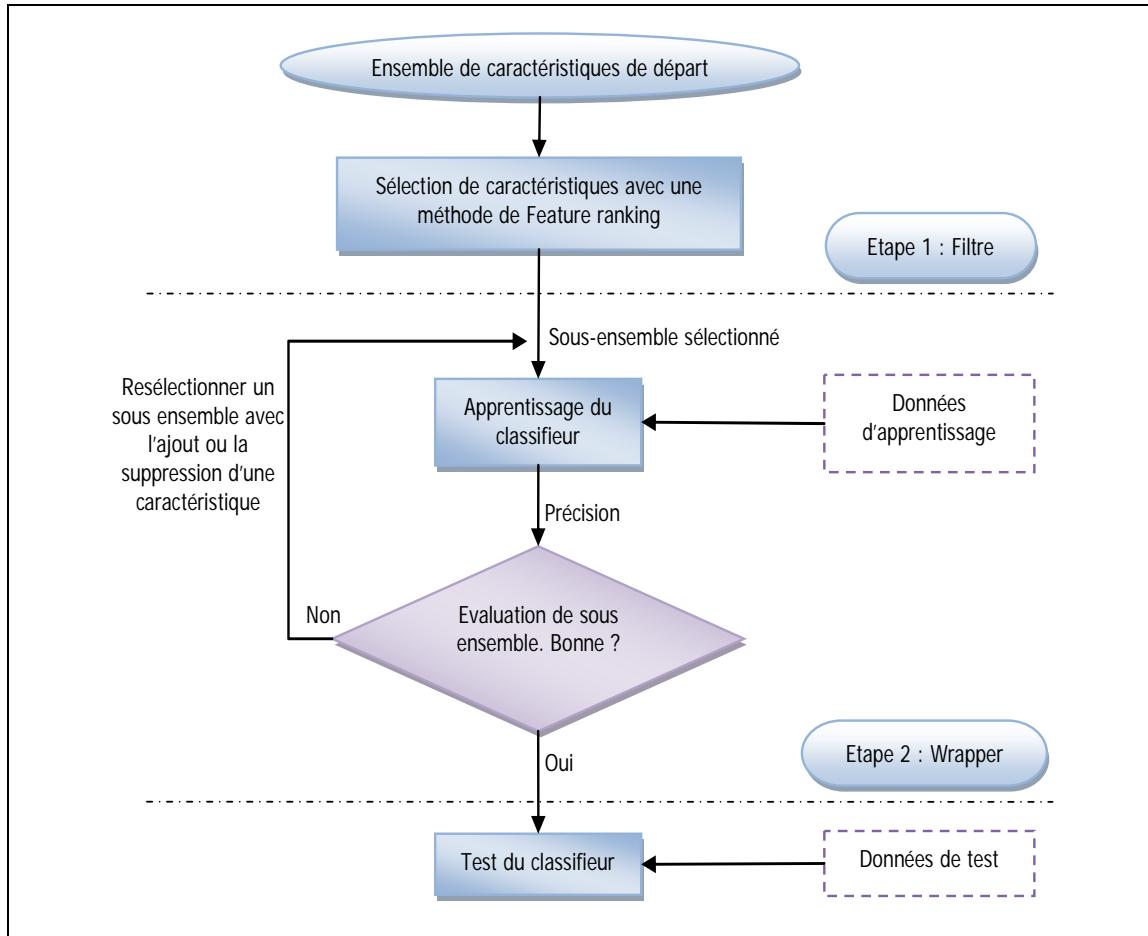


Figure 3.1. L’approche classique proposée pour la sélection d’attributs

[MEN 10a, MEN 14]

Pour générer ce nouveau sous ensemble, nous avons utilisé deux stratégies : forward (ascendante) ou backward (descendante). Dans la première, à chaque itération, nous ajoutons la meilleure caractéristique, et dans la deuxième, nous supprimons la moins bonne. La procédure de sélection s’arrête, dans le cas de la stratégie backward, si la suppression d’une caractéristique augmente l’erreur de classification. Par contre, dans la stratégie forward, la procédure s’arrête si l’erreur d’apprentissage de l’ensemble courant est inférieure à un seuil prédéfini (le seuil utilisé dans notre cas est l’erreur d’apprentissage avant sélection).

Pour le feature ranking, nous avons utilisé les mesures suivantes: la fréquence d'apparition d'un terme dans un document DF, l'information mutuelle IM, le gain d'information GI, Relief, signal to noise (s2n) et Chi² (voir Annexe 2).

3.4. Approche proposée pour la sélection d'attributs basée sur ACO

Le processus global de l'approche proposée pour la sélection d'attributs basée sur ACO peut être représenté par la Figure 3.2. Le processus commence par générer un certain nombre de fourmis p , qui sont ensuite placées au hasard sur le graphe (chaque fourmi commence avec une caractéristique aléatoire).

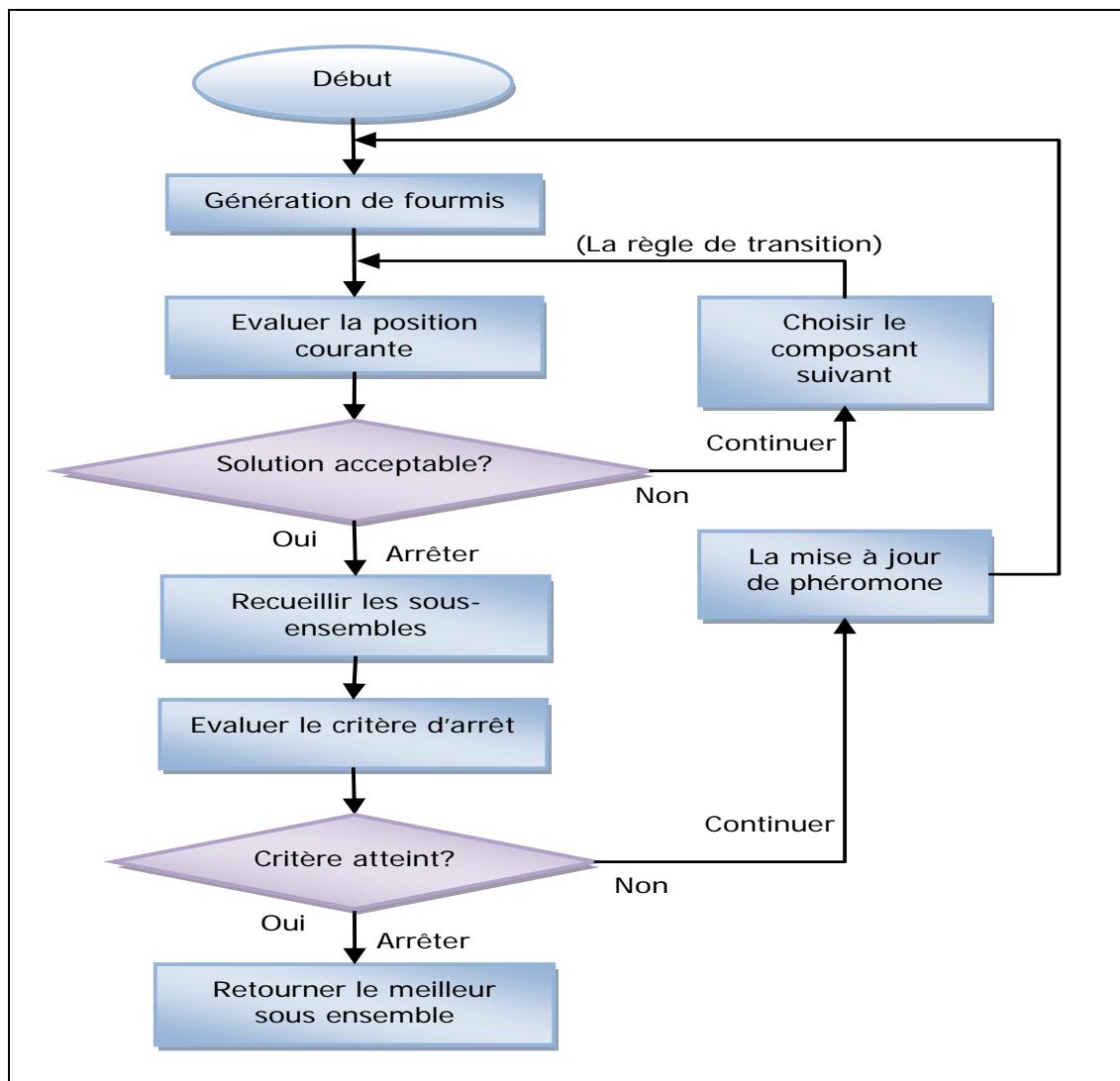


Figure 3.2. Processus général proposé pour la sélection d'attributs basée sur ACO

[MEN 14].

Le nombre de fourmis à placer sur le graphe peut être défini comme le nombre de caractéristiques dans les données; chaque fourmi commence la construction du chemin à une caractéristique différente. A partir de ces positions initiales, elles parcourent les arêtes probabilistes jusqu'à ce qu'un critère d'arrêt de parcours soit satisfait (l'erreur de classification courante est inférieure à l'erreur de classification trouvée avec l'ensemble complet des caractéristiques). Les sous-ensembles résultants sont recueillis et évalués. Si l'algorithme a été exécuté un certain nombre de fois, alors le processus s'arrête et donne le meilleur sous-ensemble de caractéristiques produit durant toutes les itérations. Si la condition n'est pas vérifiée, alors la phéromone est mise à jour, un nouvel ensemble de fourmis est créé et le processus réitère une fois de plus.

Les principales étapes de l'algorithme proposé sont les suivantes:

Étape 1: Initialisation.

- Déterminer la population de fourmis (p). p est égal au nombre complet des caractéristiques
- Définir l'intensité de phéromones associée à chaque caractéristique.
- Fixer le nombre maximum d'itérations (k).

Étape 2: Génération de fourmis et évaluation de chaque fourmi.

- Toute fourmi (A_i , $i = 1, 2, \dots, P$) est attribuée au hasard à une caractéristique et doit visiter toutes les caractéristiques et créer des solutions.
- Dans cette étape, le critère d'évaluation de sous-ensembles est l'erreur du classifieur.
- A chaque itération, les fourmis suivent ces étapes pour construire un sous-ensemble :
 - Au début, le sous-ensemble contient une seule caractéristique (la stratégie de recherche utilisée pour la sélection de caractéristiques est la stratégie forward) .
 - **Calculer $p_{ij}^k(t)$ (voir la Formule 1.3) : la probabilité de la k ième fourmi effectuant une transition de la caractéristique i à la caractéristique j à l'instant t , avec η_{ij} qui est l'opportunité heuristique appropriée de parcours entre ces caractéristiques.

$$\eta_{ij} = \frac{1}{\text{corr}_{ij}} \quad (3.1)$$

Où corr_{ij} est la corrélation entre les caractéristiques i et j .

- Ajouter la caractéristique qui a la valeur maximale de $p_{ij}^k(t)$ au sous-ensemble.
- Tester l'erreur de classification du sous-ensemble courant. Si l'erreur est diminuée par rapport à l'erreur originale (l'erreur trouvée avec l'ensemble complet des caractéristiques) la fourmi arrête le parcours et donne le sous-ensemble trouvé comme une solution. Sinon aller à **.

L'algorithme	La mise à jour	Formule de mise à jour
Ant System (AS)	Dépôt sur tous les arcs visités par toutes les fourmis	$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t)$ <p>Où $\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t)$</p> <p>Et $\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L^k(t)} & \text{si } (i, j) \in T^k(t) \\ 0 & \text{si } (i, j) \notin T^k(t) \end{cases}$</p>
Rank based Ant System (ASrank)	Fourmis triées selon la longueur du tour de façon croissante; dépôt pondéré en fonction du rang de chaque fourmi	$\tau_{ij} = \rho \tau_{ij} + \Delta\tau_{ij}^{\text{best}} + \Delta\tau_{ij}^{\text{rank}}$ <p>où</p> $\Delta\tau_{ij}^{\text{rank}} = \begin{cases} \sum_{\mu=1}^{\sigma-1} (\sigma - \mu) \frac{Q}{L_\mu} & \text{si la } \mu^{\text{ème}} \text{ meilleure} \\ & \text{fourmi se déplace sur l'arête } (i, j) \\ 0 & \text{sinon} \end{cases}$ <p>et $\Delta\tau_{ij}^{\text{gb}} = \begin{cases} \sigma \frac{Q}{L_{\text{best}}} & \text{si l'arête } (i, j) \text{ fait} \\ & \text{partie de la meilleure} \\ 0 & \text{solution trouvée} \end{cases}$</p>
Max-Min Ant System (MMAS)	Dépôt seulement, soit par la meilleure fourmi dans l'itération, ou la meilleure dans toutes les itérations précédentes; intervalle $\{\tau_{\min}, \tau_{\max}\}$	$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \Delta\tau_{ij}^{\text{best}}$ $\Delta\tau_{ij}^{\text{best}} = \begin{cases} \frac{1}{L_{\text{best}}} & \text{si la meilleure} \\ & \text{fourmi a utilisé} \\ & \text{l'arête } (i, j) \text{ dans} \\ & \text{son tour} \\ 0 & \text{sinon} \end{cases}$
<p>Remarque :</p> <p>ρ le taux d'évaporation, Q une constante, L_μ la longueur de tour de la $\mu^{\text{ème}}$ meilleure fourmi, L^k la longueur de tour de la $k^{\text{ème}}$ fourmi, L_{best} la longueur de tour de la meilleure fourmi, μ l'indice d'ordonnancement et σ le nombre de fourmis élitistes.</p> <p>Sachant que, dans les trois variantes, la règle de transition est une règle proportionnelle aléatoire et pour le phénomène d'évaporation, tous les arcs sont diminués avec le facteur constant ρ</p>		

Tableau 3.2. Les différentes variantes de l'ACO.

Étape 3: Évaluation du sous-ensemble sélectionné par chaque fourmi.

- Dans cette étape, l'importance du sous-ensemble sélectionné par chaque fourmi est évaluée par la performance de classifieur.

Étape 4: Vérifier le critère d'arrêt.

- Si le nombre d'itérations est plus grand que le nombre maximal d'itérations, arrêter, sinon continuer.

Étape 5: Mise à jour de phéromone. Elle est effectuée selon la variante utilisée pour ACO (voir Tableau 3.2).

- Pour les caractéristiques qui sont sélectionnées à l'étape 2, l'intensité de phéromone est mise à jour selon une des formules présentées dans le Tableau 3.2 (selon la variante utilisée).
- L'erreur du classifieur est utilisée pour mettre à jour la phéromone. Les valeurs de L_{μ} , L_k, L_{best} (dans le Tableau 3.2) sont remplacées par les valeurs des erreurs des fourmis correspondantes.

Étape 6: Passer à l'étape 2 et continuer.

3.5. Approche proposée pour la sélection d'attributs basée sur PSO

L'approche proposée pour la sélection d'attributs basée sur PSO a été implémentée en utilisant la version binaire de PSO (BPSO) (voir Figure 3.3) et le classifieur de type Naïve Bayes est utilisé pour évaluer les sous ensembles obtenus par BPSO (approche Wrapper). Le processus commence par l'initialisation aléatoire de la position et la vitesse de chaque particule. La position est représentée sous forme d'une chaîne binaire: les valeurs de bits 1 et 0 représentent une caractéristique sélectionnée et non sélectionnée, respectivement.

La performance de classification est utilisée pour mesurer la fitness de l'individu. La meilleure valeur de fitness pour chaque particule est notée P_{best} . Elle définit la meilleure valeur de fitness trouvée par la particule pendant toutes les itérations précédentes. La meilleure valeur de fitness dans le groupe de P_{best} est la valeur de fitness globale notée G_{best} . Après l'évaluation de la fitness, la position et la vitesse de

chaque particule sont mises à jour. Si l'algorithme s'est exécuté un certain nombre de fois ou la meilleure valeur de fitness n'a pas été modifiée pendant dix itérations successives, alors le processus s'arrête, et donne comme résultat le meilleur sous ensemble de caractéristiques trouvé pendant toutes les itérations. Si les conditions ne sont pas vérifiées, le processus réitère une fois de plus.

Les étapes de base de l'algorithme proposé sont les suivantes :

Étape 1 : Initialisation des paramètres, la population initiale, la particule initiale et la vitesse initiale sont générées aléatoirement. Les paramètres d'apprentissage $c1$ et $c2$, le poids d'inertie w (dans l'équation 1.17), le nombre maximum des itérations et la vitesse maximale v_{max} pour l'algorithme PSO ont été définies expérimentalement.

Étape 2 : Evaluation de la fitness. La tâche la plus importante est d'évaluer la particule par une fonction de fitness ; le choix de cette fonction de fitness est très important dans l'utilisation de l'algorithme PSO, il détermine ce qu'un PSO devrait optimiser. En général, l'objectif de la sélection d'attributs est de maintenir ou d'améliorer la performance de classification. La fonction de fitness choisie pour le PSO peut refléter directement les performances de classification comme suit :

$$\text{Fitness} = 1 / \text{erreur.} \quad (3.2)$$

Où *erreur* représente l'erreur globale de classification pour chaque individu d'une population obtenue en utilisant le classifieur Naïve Bayes.

Étape 3: Mise à jour des meilleures valeurs.

1. Mise à jour de la meilleure position rencontrée par la particule.

Si $\text{newFitness} > P_{best} \text{ fitness}$

$$P_{best} \text{ Fitness} = \text{newFitness}$$

et

$$P_{best} \text{ Position} = \text{newPosition.}$$

2. Mise à jour de la meilleure position rencontrée par l'essaim.

Si $\text{newFitness} > G_{best} \text{ Fitness}$

$$G_{best} \text{ Fitness} = \text{new Fitness}$$

et

$$G_{best} \text{ Position} = \text{newPosition.}$$

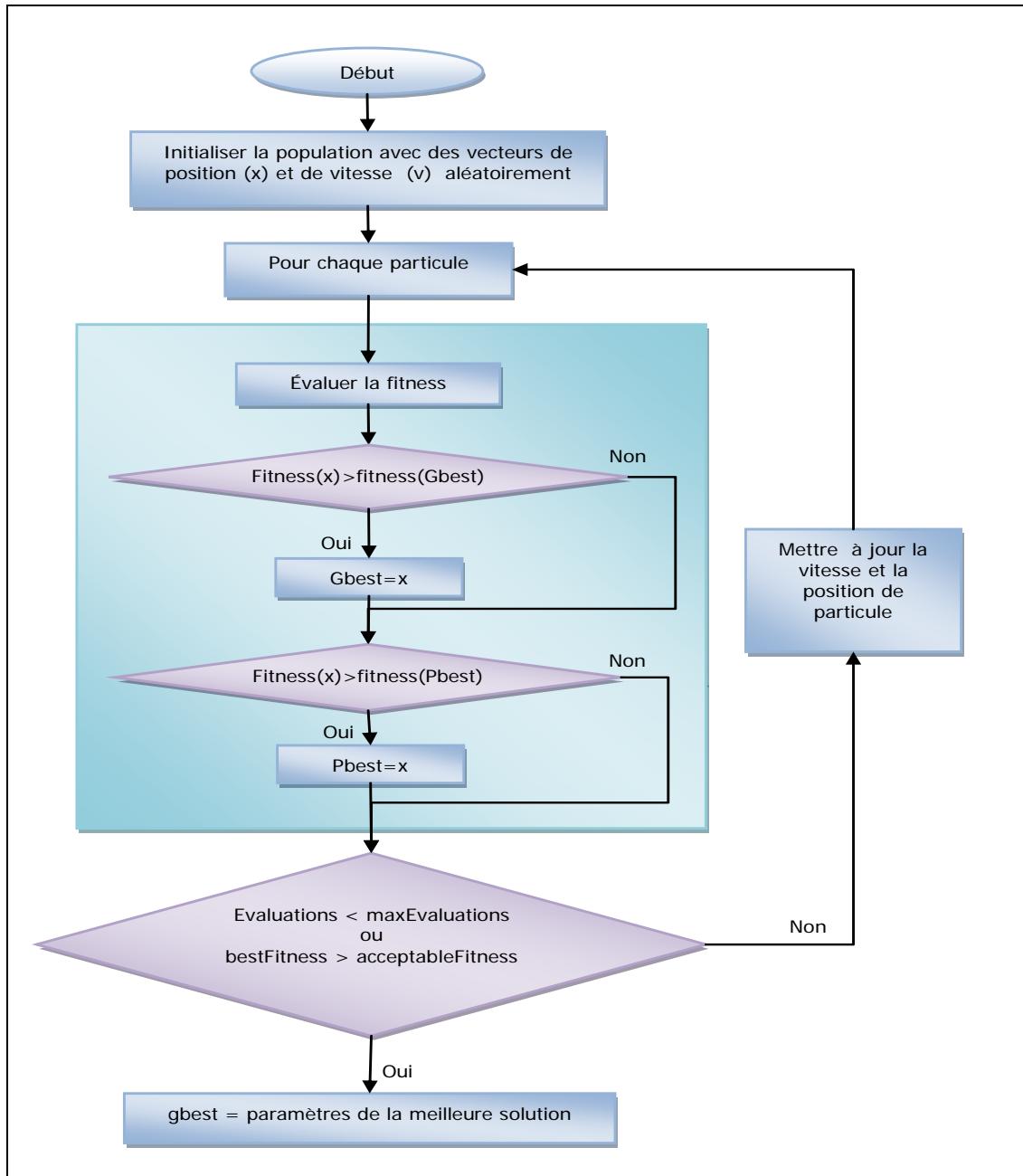


Figure 3.3. Processus général proposé pour la sélection de caractéristiques basée sur PSO [MEN 14].

Étape 4: Mise à jour de la vitesse et la position d'une particule.

Pour l'algorithme BPSO, le code binaire est mis à jour, où x_i et $Pbest_i$, pour chaque dimension sont à 0 ou 1, mais cette limitation n'est pas utilisée pour la vitesse des particules. La fonction sigmoïde de la vitesse est comme suit:

$$s(v_{ij}) = \frac{1}{1 + \exp(-v_{ij})} \quad (3.3)$$

La formule de la mise à jour de vitesse n'est pas changée (voir l'équation 1.17). Mais la position est mise à jour comme suit:

$$x_{ij}^k = \begin{cases} 1 & \text{si } (rand < s(v_{ij}^k)) \\ 0 & \text{sinon} \end{cases} \quad (3.4)$$

où *rand* définit la valeur de v_{ij} dans l'intervalle [0.0,1.0]. La vitesse maximale v_{max} peut être utilisée pour limiter la probabilité à 0 ou 1 [YAN 07].

Étape 5: Si un critère d'arrêt est rencontré (le nombre d'évaluations égal au nombre maximal d'évaluations ou la meilleure valeur de fitness n'est pas modifiée pendant plus de 10 itérations successives), le processus s'arrête et le meilleur sous ensemble de caractéristiques est donné. Sinon, évaluations = évaluations+1, et retour à l'étape 2.

3.6. Évaluation des approches proposées

Dans cette section, nous présentons une évaluation des approches proposées par leur application dans le domaine de filtrage de spams, suivie par l'évaluation de ces approches avec autres cinq bases de données publiques. Nous avons utilisé des données provenant de l'UCI Machine Learning Repository.

L'UCI Machine Learning Repository contient une collection de bases et de générateurs de données qui sont utilisés par la communauté de l'apprentissage automatique (Machine Learning) pour l'analyse empirique des algorithmes d'apprentissage automatique. L'archive a été créée en 1987 à l'Université de Californie à Irvine (UCI) (<http://archive.ics.uci.edu/ml/datasets>).

Afin de pouvoir évaluer les résultats des approches proposées, nous avons implémenté un algorithme génétique en utilisant l'outil Weka (<http://www.cs.waikato.ac.nz/ml/weka>).

Le classifieur de type Naïve Bayes utilisé dans nos expérimentations a été fourni par Spider (object-oriented machine learning package : <http://people.kyb.tuebingen.mpg.de/spider/>).

3.6.1. Choix expérimentaux

3.6.1.1. Paramètres d'ACO et PSO

Les paramètres de l'ACO et PSO ont été déterminés expérimentalement. Ils sont présentés dans les Tableaux 3.3 et 3.4, respectivement, et sont utilisés dans les approches proposées.

Paramètres	Valeur
Nombre de fourmis	Nombre de caractéristiques dans la base de données utilisée
Nombre de nœuds	
α	1
β	5

Tableau 3.3. Paramètres d'ACO

Paramètres	Valeur
La taille de l'essaim	Nombre de caractéristiques dans la base de données
W	0.6571
c1	1.6319
c2	0.6239

Tableau 3.4. Paramètres de PSO

3.6.1.2. Évaluation des performances de classification

Afin de comparer les résultats issus de ces différents algorithmes, nous allons utiliser l'erreur d'apprentissage BER pour évaluer les performances du classifieur.

BER: Balanced Error Rate est la moyenne entre le taux d'erreur de la classe positive et celui de la classe négative.

$$\text{BER} = (\text{L'erreur des exemples positifs (faux positifs)} + \text{L'erreur des exemples négatifs (faux négatifs)})/2.$$

L'erreur des exemples positifs (faux positifs) = nombre des exemples positifs mal classés/nombre total des exemples positifs.

L'erreur des exemples négatifs (faux négatifs) = nombre des exemples négatifs mal classés/nombre total des exemples négatifs.

3.6.2. *Expérimentations dans le domaine du filtrage de spams*

3.6.2.1. *Description de la base de données*

Pour nos expérimentations dans le domaine du filtrage de spams, nous avons utilisé la collection SpamBase disponible dans UCI Machine Learning Repository (<http://archive.ics.uci.edu/ml/datasets/spambase>).

Caractéristiques	Type	Explication
1-48	word_freq_WORD	Pourcentage des mots dans l'email qui sont égaux à WORD, i.e. $100 * (\text{nombre d'occurrences du mot WORD dans l'email}) / \text{nombre total des mots dans l'email}$. Un mot dans ce cas est une chaîne de caractères alphanumériques délimitée par des caractères non alphanumériques ou terminée par une fin de chaîne.
49-54	char_freq_CHAR	Pourcentage de caractères dans l'email qui sont égaux à CHAR, i.e. $100 * (\text{nombre des occurrences de caractère}) / \text{nombre total des caractères dans l'email}$.
55	capital_run_length_average	Moyenne des longueurs des séquences ininterrompues de lettres majuscules.
56	capital_run_length_longest,	Longueur de la séquence ininterrompue de lettres majuscules la plus longue.
57	capital_run_length_total	Somme des longueurs des séquences ininterrompues de lettres majuscules = nombre total de lettres majuscules dans l'email

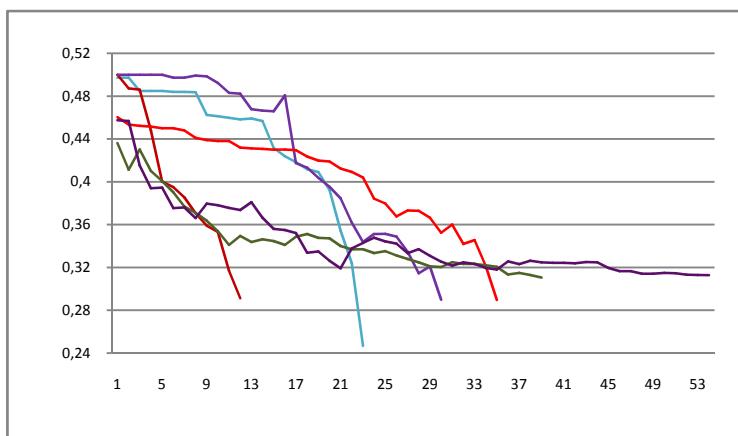
Tableau 3.5. Description de la base SpamBase

SpamBase contient les informations relatives à 4601 messages, avec 1813 (39,4%) de spams. Chaque message est décrit en fonction de 57 attributs, le 58ème correspond à la classe du message : spam ou message légitime (voir Tableau 3.5 et Annexe 1).

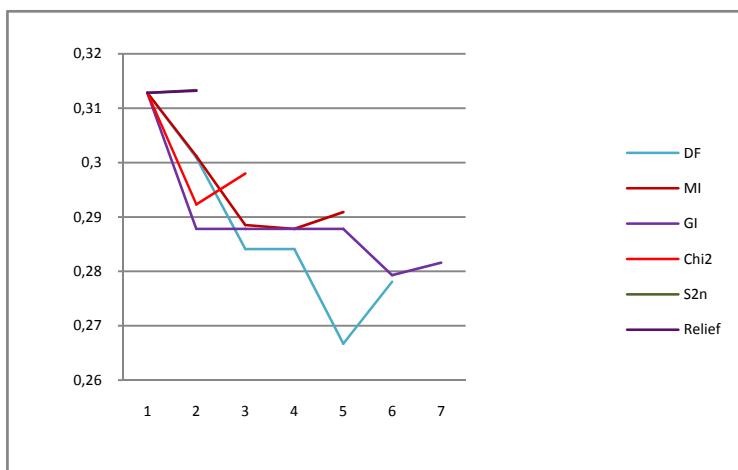
Nous avons divisé la base SpamBase en 2 ensembles : un ensemble d'apprentissage contenant 3250 exemples (avec 1250 spams), et un ensemble de test contenant 1351 exemples (avec 583 spams).

3.6.2.2. Résultats expérimentaux

Les résultats de nos expérimentations, en utilisant les approches proposées et décrites dans ce chapitre: classique, ACO et PSO sont illustrées dans les Figures 3.4, 3.5, 3.6 respectivement, et sont résumés dans le Tableau 3.6. Ce tableau synthétise, pour chaque approche, la meilleure erreur trouvée pendant toutes les itérations, les attributs sélectionnés ainsi que leur nombre. Ces résultats sont obtenus avec la base d'apprentissage.

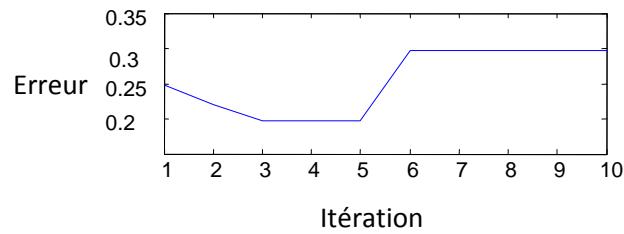


a. La stratégie forward

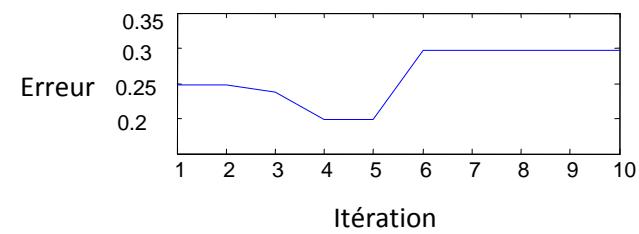


b. La stratégie backward

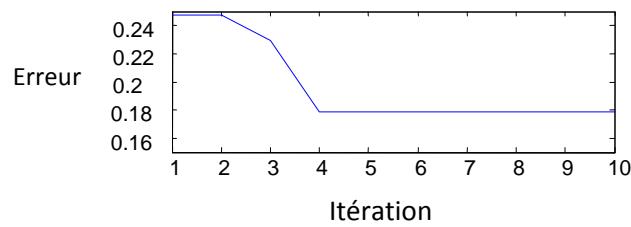
Figure 3.4. L'erreur versus nombre d'itérations pour l'approche classique 'feature ranking'.



a. AS



b. MMAS



c. ASRank

Figure 3.5. L'erreur de la meilleure fourmi versus le numéro d'itération.

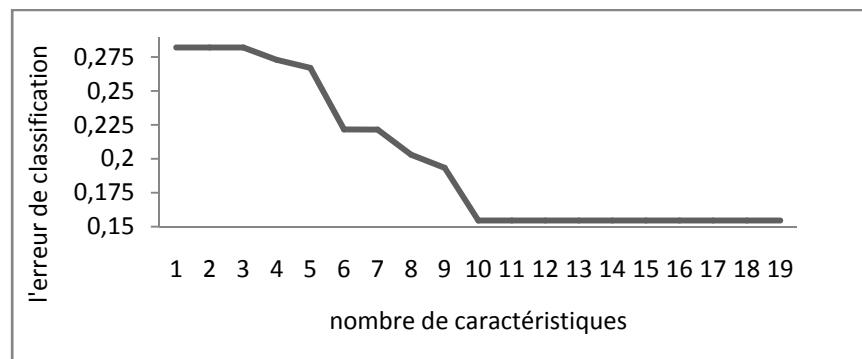


Figure 3.6. L'erreur de la meilleure solution trouvée dans chaque itération pour l'approche basée sur PSO.

La Figure 3.5 présente les résultats obtenus par la meilleure fourmi dans chaque itération avec les trois variantes d'ACO. La Figure 3.6 représente la meilleure erreur globale de classification trouvée par PSO dans chaque itération.

Comme nous pouvons le voir dans la Figure 3.4.a, les meilleurs résultats de la sélection d'attributs classique par les stratégies forward sont obtenus en utilisant la mesure DF pour le classement des attributs. Elle produit la valeur d'erreur la plus faible (0,24) avec un petit sous-ensemble de 24 attributs.

Dans la Figure 3.4.b, qui représente les résultats de la sélection par la stratégie backward, nous voyons que le nombre d'attributs n'est pas significativement réduit (seulement entre 1 et 5 attributs sont supprimés). En termes d'erreur d'apprentissage, l'approche produit des taux d'erreur similaires (entre 0,26 et 0,29) pour les méthodes MI, IG, Chi2 et DF, et on note très peu d'amélioration avec les mesures S2n et Relief.

Méthode	Erreur	Nombre d'attributs sélectionnés	Numéros des attributs sélectionnés (voir Annexe 1)
Avant sélection	0.3128	57	/
PSO	0.1545	16	2, 15, 24, 25, 26, 27, 30, 31, 32, 33, 35, 38, 42, 43, 46, 48
AS	0.1972	14	45, 38, 30, 48, 33, 39, 27, 43, 2, 55, 14, 36, 47, 25
MMAS	0.1992	8	36, 47, 25, 41, 48, 33, 39, 27
ASRank	0.1992	8	36, 47, 25, 41, 48, 33, 39, 27
Algorithmes génétiques	0.4368	32	1, 3, 4, 6, 7, 8, 9, 10, 11, 15, 17, 19, 20, 21, 22, 23, 25, 26, 27, 28, 32, 33, 39, 45, 46, 48, 49, 52, 53, 54, 55, 57
Classique	0.2467	24	4, 47, 38, 22, 41, 48, 32, 46, 34, 15, 44, 31, 42, 14, 39, 20, 43, 29, 33, 40, 28, 30, 35, 51

Tableau 3.6. Comparaison entre les performances de différentes méthodes pour la base Spambase.

Le Tableau 3.6 compare les résultats de différentes approches. Dans ce tableau, nous pouvons voir que, généralement, les attributs 55-57 (qui représentent la longueur des

séquences de lettres majuscules consécutives) ne sont pas sélectionnés dans les meilleurs sous-ensembles d'attributs. Certains attributs comme les attributs numéro 33 et 47 (qui représentent word_freq_data et word_freq_table respectivement) sont choisis avec toutes les approches.

En analysant ce tableau, il est clair que PSO est le meilleur en termes de précision de la classification, suivie par l'ACO (plus précisément les variantes MMAS et ASRank). Les algorithmes génétiques augmentent l'erreur et ne donnent pas de résultats intéressants par rapport aux approches bio-inspirées proposées. En termes de réduction de la taille du sous-ensemble, les deux approches proposées pour la sélection d'attributs, basée ACO et PSO réduisent considérablement le nombre de caractéristiques originales. Mais les approches MMAS et ASRank sélectionnent moins d'attributs.

3.6.3. Expérimentations des approches basées ACO et PSO avec différentes bases de données

3.6.3.1. Description des bases de données

Dans le but de mieux évaluer les approches bio-inspirées de sélection d'attributs proposées dans ce chapitre, nous avons utilisé cinq autres bases de données publiques disponibles dans l'UCI Machine Learning Repository. Le Tableau 3.7 présente une brève description de ces bases, une description plus détaillée est donnée en Annexe 1.

La base de données	Nombre initial d'attributs	Type des attributs	Nombre d'exemple
Breast Cancer	9	Entier	699
German	24	Catégorique, entier	1000
Hepatitis	19	Catégorique, entier, réel	155
Liver	6	Catégorique, entier, réel	345
Musk	166	Entier	476

Tableau 3.7. Description des bases de données

3.6.3.2. Résultats expérimentaux

Dans cette section, nous présentons les résultats de nos approches de sélection d'attributs basées sur ACO et PSO avec les cinq bases de données décrites

précédemment. Le Tableau 3.8 présente, pour chaque base de données, la meilleure erreur de classification, le nombre total de caractéristiques originales, l'ensemble des caractéristiques sélectionnées (leurs numéros) ainsi que leur nombre.

Base de données	Méthode	Erreur	Nombre d'attributs sélectionnés	Numéros des attributs sélectionnés
Breast Cancer	Avant sélection	0.4433	9	/
	ACO	0.3033	2	9, 1
	PSO	0.3483	2	1, 5
German	Avant sélection	0.4990	24	/
	ACO	0.3680	2	1, 9
	PSO	0.2775	8	1, 2, 8, 13, 14, 16, 21, 22
Hepatitis	Avant sélection	0.5000	19	/
	ACO	0.3836	2	5, 7
	PSO	0.2760	7	2, 5, 9, 12, 15, 17, 19
Liver	Before selection	0.5	6	/
	ACO	0.5	6	1, 2, 3, 4, 5, 6
	PSO	0.4182	1	5
Musk	Avant sélection	0.3725	166	/
	ACO	0.2510	16	25, 147, 90, 75, 62, 111, 9, 132, 18, 165, 103, 36, 73, 137, 104, 163
	PSO	0.2277	26	1, 15, 16, 18, 20, 23, 26, 36, 37, 48, 50, 51, 52, 53, 60, 67, 78, 80, 82, 92, 94, 97, 103, 104, 106, 124, 129, 132, 136, 145, 147, 151, 155, 160

Tableau 3.8. Résultats des approches proposées de sélection d'attributs basées sur ACO et PSO avec cinq bases de données publiques

D'après le Tableau 3.8, nous pouvons voir que les performances du classifieur sont améliorées et les sous-ensembles des attributs sont réduits pour toutes les bases de données utilisées dans nos expérimentations.

3.7. Conclusion

Dans ce chapitre nous avons commencé par une synthèse des travaux de sélection de caractéristiques basées sur des méthodes bio-inspirées non hybrides : algorithmes génétiques, colonies de fourmis, essaims particulaires et méthodes inspirées des abeilles. Une grande partie de ce chapitre est consacrée à la présentation des trois approches que nous avons proposées pour la sélection d'attributs. La première est classique, basée sur le principe du Feature Ranking et les deux autres sont bio-inspirées, non hybrides, basées, respectivement, sur ACO et PSO.

Nos résultats expérimentaux montrent que les approches de sélection de caractéristiques proposées peuvent atteindre un taux d'erreur de prédiction considérablement réduit et produisent un petit sous-ensemble de caractéristiques. En outre, la sélection d'attributs basée sur PSO donne, généralement, de meilleurs résultats par rapport à ACO en termes de précision de classification. D'un autre côté, ACO permet généralement d'obtenir de meilleurs résultats en termes de réduction de sous-ensemble de caractéristiques.

CHAPITRE 4

APPROCHES BIO-INSPIREES HYBRIDES POUR LA SELECTION D'ATTRIBUTS

Ce chapitre est dédié aux méthodes bio-inspirées hybrides. Notre but est de proposer une nouvelle approche de sélection d'attributs basée sur une hybridation entre des méthodes bio-inspirées et pouvant être utilisée dans un ou plusieurs domaines applicatifs. Nous commençons ce chapitre par une étude de méthodes bio-inspirées hybrides existantes indépendamment de la sélection d'attributs, suivie par une synthèse des hybridations de méthodes bio-inspirées appliquées à la sélection de caractéristiques. Dans la section 4, nous présentons les trois approches hybrides proposées basées sur des hybridations entre les colonies de fourmis (ACO) et les essaims particulaires(PSO). Les expérimentations effectuées sur six bases de données publiques sont présentées dans la section 5.

4.1. *Introduction*

L'utilisation des méthodes inspirées de la biologie a attiré l'attention de la communauté de recherche depuis long temps. Malgré le succès de l'utilisation des méthodes bio-inspirées pour résoudre les problèmes d'optimisation, plusieurs limitations ont été rencontrés, à savoir :

- La distinction entre une solution optimale et une autre globale est par fois difficile à percevoir.
- Le problème de la convergence prématurée
- Le problème d'ajustement des paramètres.

De nos jours, les chercheurs essaient de proposer des améliorations importantes qui permettent l'apparition d'une nouvelle génération de méthodes puissantes et générales. Ces dernières années, il est devenu évident que le fait de se concentrer sur une seule

méthode pour résoudre un problème complexe est plutôt restrictif, une intégration de plusieurs approches peut améliorer l'efficacité et la flexibilité, surtout lorsqu'il s'agit de problèmes à grande dimension du monde réel. La tendance actuelle est l'émergence de méthodes hybrides, qui essayent de tirer parti des avantages spécifiques de différentes approches en les combinant.

L'hybridation de méthodes bio-inspirées d'optimisation permet d'avoir un équilibre entre l'exploration et l'exploitation de l'espace de recherche des solutions. Pour avoir une bonne solution, un algorithme peut être utilisé pour localiser les meilleures solutions locales de l'espace de recherche alors qu'un autre permet de converger vers l'optimum global. L'hybridation peut être utilisée aussi pour optimiser les paramètres généraux. Le premier algorithme pourrait servir, par exemple, à déterminer les meilleurs paramètres du deuxième.

Dans notre travail, nous avons choisi d'aborder l'hybridation de deux méthodes inspirées de l'intelligence en essaims qui sont ACO et PSO. La méthode ACO offre l'avantage principal de la recherche locale. D'autre part, PSO considère une perspective globale en agissant sur la population complète dès le début. En plus, le PSO n'a pas besoin d'opérateurs complexes, il ne nécessite que des opérateurs mathématiques primitifs et simples, et le calcul est peu coûteux en termes de mémoire et de temps.

La section suivante présente une synthèse de méthodes bio-inspirées hybrides incluant ACO ou PSO, suivie par celle des hybridations appliquées dans le domaine de la sélection d'attributs.

4.2. Synthèse d'approches hybrides basées ACO ou PSO

Récemment, de nombreux auteurs proposent des approches hybrides pour résoudre plusieurs problèmes d'optimisation dans différents domaines d'application. L'hybridation des méthodes évolutionnaires a été étudiée dans de nombreux travaux mais comme nous nous concentrons sur ACO et PSO, nous avons choisi d'étudier un ensemble d'approches hybrides basées soit sur ACO, soit sur PSO.

En se basant sur la propriété compensatoire des AG et PSO, Juang a proposé un nouvel algorithme appelé HGAPSO qui combine les idées évolutionnaires des deux [JUA 04]. Dans HGAPSO (Figure 4.1), les individus d'une nouvelle génération sont créés, non

seulement par les opérations de croisement et de mutation de l'AG, mais aussi par PSO. Le concept de la stratégie d'élite est adopté en HGAPSO, où la moitié des individus les plus performants dans une population sont considérés comme des élites. Cependant, au lieu d'être reproduites directement à la génération suivante, ces élites sont d'abord améliorées. Le groupe constitué par les élites est considéré comme un essaim, et chaque élite correspond à une particule dans l'essaim. A cet égard, les élites sont renforcées par PSO, une opération qui imite le phénomène de maturation dans la nature. Ces élites améliorées constituent la moitié de la population dans la nouvelle génération, tandis que l'autre moitié est générée en effectuant les opérations de croisement et de mutation sur ces élites améliorées.

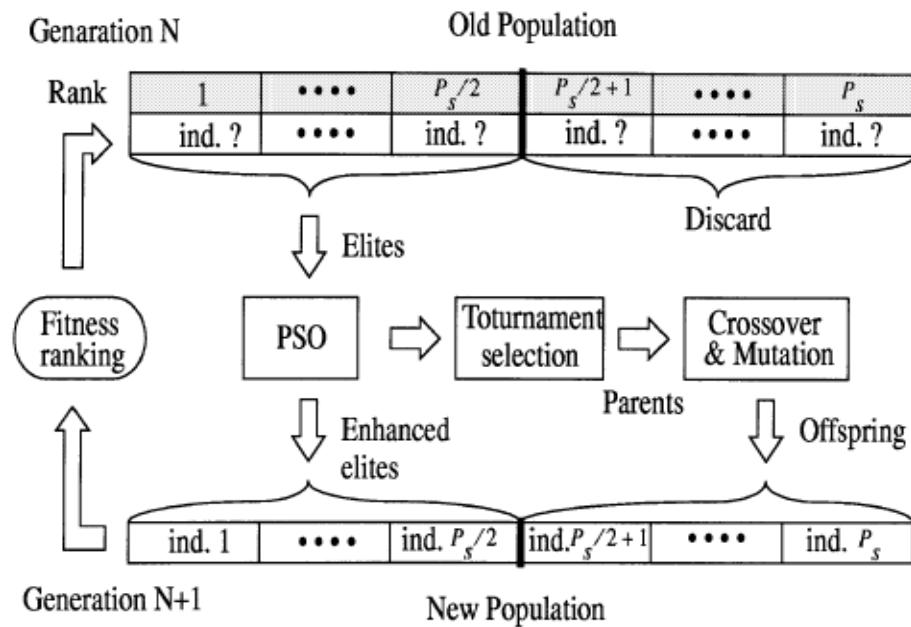


Figure 4.1. Le principe de l'approche HGAPSO proposée par Juang [JUA 04]

Guangdong et al [GUA 07] présentent une météuristique hybride ACO-GA pour le problème de la planification de la compétition sportive. ACO-GA combine l'optimisation par les colonies de fourmis (ACO) et les algorithmes génétiques. L'algorithme génétique génère les informations distribuées et ACO cherche la solution exacte. Au début, l'AG génère des solutions et la phéromone est laissée sur les excellentes solutions, puis l'ACO est exécutée. Quand ACO se termine, les opérations de croisement et la mutation de l'AG génèrent une nouvelle population. L'ACO et l'AG effectuent leur recherche en alternance et en coopération dans l'espace de solution.

Shi et al. [SHI 08] présentent un algorithme hybride ACO-PSO afin de résoudre le problème de planification de chemin des robots mobiles dans un environnement à trois dimensions. Il y a deux étapes de base dans les méthodes de planification de chemin: la modélisation de l'environnement et l'algorithme de planification. Les auteurs de ce travail planifient les chemins de robots mobiles en utilisant un algorithme d'intelligence en essaim, cet algorithme est basé sur l'hybridation de l'algorithme des colonies de fourmis et celui des essaims de particules. L'optimisation par les colonies de fourmis a été utilisée pour planifier des chemins sur le territoire de robots alors que l'optimisation par les essaims particulaires a été appliquée pour optimiser les paramètres de l'ACO.

Dans [VAL 08] une nouvelle méthode évolutionnaire a été présentée. Cette méthode combine les avantages de PSO et des algorithmes génétiques pour donner une méthode hybride améliorée PSO +AG. L'idée de base de la méthode PSO + AG est de combiner les avantages des méthodes individuelles en utilisant un système flou pour prendre la décision et deux autres systèmes flous pour améliorer les paramètres de l'AG et PSO quand cela est nécessaire. Le système flou principal est responsable de recevoir les valeurs résultantes de PSO et AG, et de décider quelle méthode utiliser (AG ou PSO). Par la suite, un autre système flou reçoit les erreurs comme des entrées pour les évaluer (si nécessaire), et modifier les paramètres d'AG ou PSO. L'approche générale de la méthode PSO + AG proposée dans [VAL 08] est présentée dans la Figure 4.2.

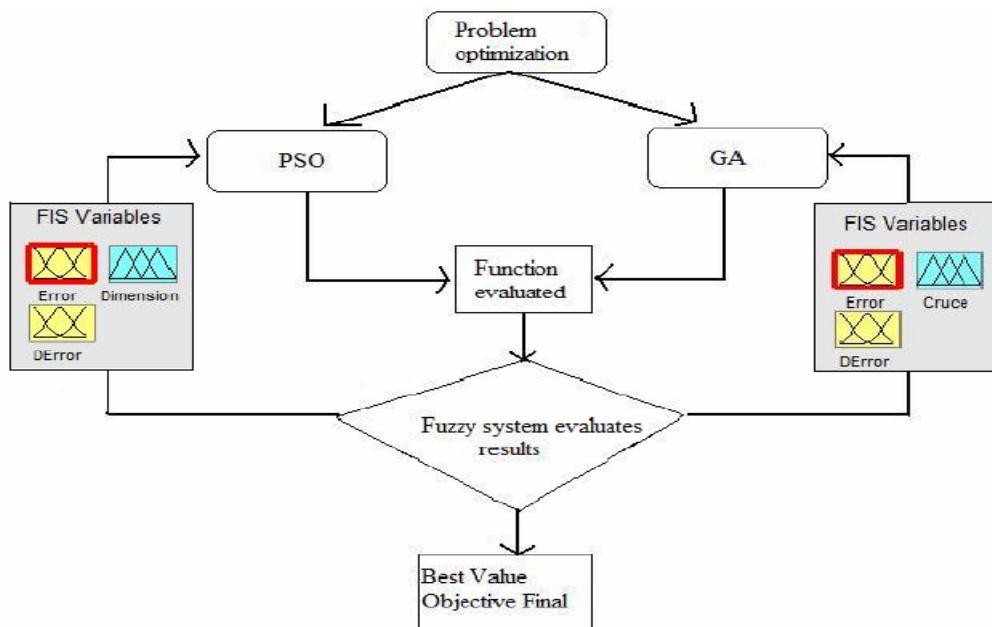


Figure 4.2. L'approche PSO+AG [VAL 08]

Une autre hybridation entre les algorithmes génétiques et les essaims de particules est présentée dans [PRE 09]. Dans cet algorithme hybride, la mise à jour de la position des meilleures particules est effectuée à travers un mécanisme hybride des AGs (en utilisant les opérations génétiques), afin de prévenir le problème de la convergence prématurée de PSO. Trois approches hybrides différentes sont proposées :

1. PSO-GA (type 1): La meilleure position globale de particules *Gbest* n'est pas modifiée dans un intervalle de temps désigné, l'opération de croisement est effectuée sur *Gbest* de la particule avec le chromosome de l'AG. Dans cette approche, PSO et AG sont exécutés en parallèle.
2. PSO-GA (type 2): Les particules *Pbest* stagnées changent leurs positions par l'opérateur de mutation de l'AG.
3. PSO-GA (type 3): La population initiale de PSO est définie par la solution de l'AG. Le nombre total d'itérations est partagé par AG et PSO. La première moitié des itérations est gérée par AG et les solutions sont utilisées comme population initiale de PSO. Les itérations restantes sont gérées par PSO.

Dans [PAT 14], les auteurs présentent un algorithme intelligent basé sur l'essaimage des agents en utilisant une technique hybride (ACO/PSO) pour optimiser l'arbre multicast (multidiffusion). L'algorithme commence par la génération d'une grande quantité d'agents mobiles dans l'espace de recherche. L'algorithme ACO guide le mouvement des agents par les phéromones partagées dans l'environnement local, et le maximum global des valeurs d'attributs est obtenu grâce à l'interaction aléatoire entre les agents en utilisant l'algorithme PSO.

Le Tableau 4.1 présente une synthèse de certaines approches hybrides récentes basées sur ACO ou PSO.

Auteurs	Référence	Hybridation proposée	Principe de l'hybridation proposée	Domaine d'application
Juang	[JUA 04]	PSO-AG	Les individus de la nouvelle génération sont créés non seulement par les opérations de croisement et de mutation comme dans les AGs, mais aussi par PSO.	Automatisation de la conception de réseaux de neurones flous récurrents
Shi et al.	[SHI 05]	PSO-AG	Exécuter les deux algorithmes simultanément (PSO, AG) après sélectionner P individus de chaque système d'échange après le N itérations désignées.	Optimisation des fonctions non linéaires
Valdez et al.	[VAL 08]	PSO-AG	Utiliser la logique floue pour intégrer les résultats des deux méthodes AG et PSO.	Optimisation d'une fonction mathématique
Li et al.	[LI 08]	PSO -AG	Après le classement des individus dans la génération actuelle, les meilleurs individus sont sélectionnés comme les élites en les reproduisant directement à la prochaine génération, les individus suivants sont évolués avec PSO, leurs meilleures positions sont mises à jour et les individus inférieurs sont évolués avec une amélioration de l'algorithme génétique	Antenna array pattern synthesis
Niknam et Amiri	[NIK 10]	Fuzzy adaptive PSO-ACO et Kmeans	La Sélection de <i>Gbest</i> pour chaque particule est en fonction de la méthode de sélection du meilleur chemin d'ACO. L'algorithme ACO-PSO est utilisé comme état initial de l'algorithme de K-means.	Problème de clustering partitionnel non linéaire (Nonlinear partitional clustering problem)
Shuang et al.	[SHU 11]	ACO-PSO	Les règles de mise à jour de phéromone de l'ACO sont combinées avec les mécanismes de recherche locale et globale de PSO	Problème de voyageur de commerce
Kiran et al.	[KIR 12]	ACO-PSO	La meilleure solution est attribuée à la meilleure solution globale de l'hybridation selon la comparaison des meilleures solutions trouvées par ACO et PSO	Estimation de la demande d'énergie de la Turquie

Tableau 4.1. Synthèse de travaux utilisant des approches hybrides basées sur ACO ou PSO.

4.3. Approches bio-inspirées hybrides pour la sélection d'attributs

Certaines méthodes bio-inspirés hybrides ont été proposées pour la sélection d'attributs telles que l'hybridation des colonies de fourmis avec les algorithmes génétiques [SHE 12; SHE 13; BAS 09; NEM 09] et celles intégrant les essaims particulaires.

Une approche hybride basée sur l'ACO et les réseaux de neurones artificiels (RNA) pour trouver le sous-ensemble optimal de caractéristiques est présentée dans [SIV 07]. Le modèle hybride proposée est évalué en utilisant des bases de données pour le diagnostic médical.

Dans [YAN 08], un algorithme binaire amélioré d'optimisation par les essaims particulaires a été intégré dans un algorithme génétique afin de servir comme optimiseur local pour chaque génération dans le problème de sélection des caractéristiques.

Basiri et Nemati [BAS 09] ont proposé un nouvel algorithme hybride ACO-AG pour la sélection d'attributs dans la catégorisation des textes. La performance de classification et la taille du sous-ensemble de caractéristiques sélectionnées sont adoptées comme des informations heuristiques.

Sheikhan et Mohammadi [SHE 12] ont développé un modèle hybride pour la prévision de charge à court terme. AG et ACO sont combinés dans ce modèle pour explorer l'espace de tous les sous-ensembles d'un ensemble de caractéristiques données, et le perceptron multicouche est utilisé pour la prédiction de la charge horaire. Sheikhan et Mohammadi [SHE 13] utilisent une méthode de sélection de caractéristiques AG-ACO hybride pour obtenir le sous-ensemble de caractéristiques le plus petit et le plus efficace avec la base de données IEEE load dataset.

Jin et al. [JIN 12] proposent sort-based BPNN-PSO (BPNN: Back Propagation Neural Network) pour sélectionner des attributs essentiels afin d'améliorer les performances de généralisation et réduire le coût de calcul de BPNN. Dans la méthode de sélection d'attributs proposée, la corrélation des entrées et des sorties est appliquée pour calculer l'importance des caractéristiques.

Le Tableau 4.2 présente une synthèse de certains travaux sur la sélection d'attributs avec des hybridations d'approches bio-inspirées incluant ACO ou PSO (pour des explications sur le contenu du tableau, voir section 3.2.5).

Auteurs	Référence	Hybridation bio-inspirée	Approche de FS	Domaine d'application ou les noms des bases de données utilisées	Résultats	
					Ensemble d'origine	FS
Sivagaminathan et Ramakrishnan	[SIV 07]	ACO-ANN	Wrapper	Base : Thyroid disease (Australia)*	28	12
				Base : Thyroid disease (discordant)*	28	4
				Base : Thyroid disease (Hypothyroid)*	21	14
				Base : Dermatology*	34	7
				Base : Breast cancer (Wisconsin diagnostic)*	32	12
				Base : Breast cancer (Wisconsin prognostic)*	30	14
Yang et al	[YAN 08]	IBPSO-AG	Wrapper	Base : 9 Tumors	5726	2140
				Base : Brain Tumor1	5920	2030
				Base : Brain Tumor2	10367	3773
				Base : Leukemia1	5327	1803
				Base : SRBCT	2305	1175
Basiri et Nimati	[BAS 09]	ACO-AG	Wrapper	Catégorisation de texte	-	> 20%
Nemati et al	[NEM 09]	ACO-AG	Wrapper	prédiction de la fonction des protéines	Base : GPCR-PROSITE	
					Base : ENZYME-PROSITE	
Sheikhan et Mohammadi	[SHE 12]	ACO-AG	Wrapper	la prévision de charge à court terme		- 20
Sheikhan et Mohammadi	[SHE 13]	ACO-AG	Wrapper	Base : IEEE load dataset		28 8
Jin et al	[JIN 12]	BPNN-PSO	Hybride	Parkinsons Telemonitoring *		21 11
				Concrete Slump*		7 5
				JM1*		21 17

Tableau 4.2. Synthèse de travaux sur la sélection d'attributs basée sur une approche hybride incluant ACO ou PSO.

4.4. Les approches hybrides ACO-PSO proposées pour la sélection d'attributs

Nous avons étudié et présenté différentes méthodes bio-inspirées hybrides basées ACO et PSO existant dans la littérature, et appliquées dans différents domaines, en plus de celui de la sélection de caractéristiques. Après notre étude, nous avons constaté l'absence des hybridations entre ACO et PSO pour la sélection d'attributs. Dans cette section, nous décrivons les trois approches de sélection d'attributs que nous proposons et qui sont basées sur une hybridation entre ACO et PSO [MEN 15].

4.4.1. L'approche proposée ACO-PSO1

L'approche ACO-PSO1 que nous proposons (Figure 4.3) est inspirée de celle proposée par Nemati et al. [NEM 09] où notre idée est de remplacer les algorithmes génétiques (AG) par les PSO.

Dans [NEM 09], les auteurs présentent un algorithme hybride de sélection de caractéristiques basé sur une hybridation des colonies de fourmis (ACO) et des algorithmes génétiques (AG). ACO offre l'avantage principal de la recherche locale, absente dans AG. D'autre part, AG considère une perspective globale en agissant sur la population complète dès le début. Par conséquent, ACO et GA peuvent annuler leurs inconvénients mutuels lors de l'hybridation.

L'utilisation de l'algorithme génétique pour effectuer une sélection de caractéristiques nécessite de nombreuses itérations. Si l'algorithme d'apprentissage est un processus itératif, le coût de calcul de l'ensemble du processus sera très haut, puisque l'algorithme d'apprentissage est utilisé à chaque itération pour évaluer la fitness de chaque individu.

Comme les AGs, le PSO est aussi un type d'algorithmes évolutionnaires. Mais par rapport à AG, PSO n'a pas besoin d'opérateurs complexes comme le croisement et la mutation, il ne nécessite que des opérateurs mathématiques primitifs et simples, et le calcul est peu coûteux en termes de mémoire et de temps [WAN 05; WAN 07]

Les principales étapes de l'ACO-PSO1 proposée sont les suivantes:

Étape 1 : Initialisation.

- Initialiser les paramètres de PSO: Générer la position et la vitesse de chaque particule aléatoirement.
- Initialiser les paramètres de l'ACO:
 - Déterminer la population de fourmis. p est égal au nombre initial des caractéristiques.
 - Définir l'intensité de la trace de phéromone associée à chaque caractéristique.
- Déterminer le maximum d'itérations autorisées.

Étape 2 : Exécuter ACO et PSO simultanément.

- ACO: génération des fourmis et évaluation de chacune.
 - Chaque fourmi (A_i , $i = 1, 2, \dots, p$) est assignée aléatoirement à une caractéristique et elle doit visiter toutes les caractéristiques et construire des solutions.
 - Dans cette étape, l'erreur de classification est utilisée en tant que mesure d'évaluation.
 - A chaque itération, les fourmis suivent ces étapes pour construire un sous-ensemble :
 - Au début, le sous-ensemble ne contient qu'une seule caractéristique (la stratégie de recherche utilisée pour la sélection de caractéristiques est une stratégie forward).
 - ** Calculer la probabilité de la transition de la k -ième fourmi de caractéristique i à caractéristique j par l'équation 2.2

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l=1}^k [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta} \quad (4.1)$$

Avec $\tau_{ij}(t)$ est la quantité de phéromone, η_{ij} est l'opportunité heuristique de choisir la caractéristiques j quand la fourmi est sur la caractéristique i .

$$\eta_{ij} = \frac{1}{\text{corr}_{ij}} \quad (4.2)$$

où corr_{ij} définir la corrélation entre les caractéristiques i et j.

- Ajouter la caractéristique qui a la valeur max de $p_{ij}^k(t)$ au sous-ensemble.
- Tester l'erreur de classification du sous-ensemble actuel. Si l'erreur est réduite par rapport à l'erreur d'origine (l'erreur trouvée avec l'ensemble complet de caractéristiques), la fourmi termine son parcours et fournit le sous ensemble de caractéristiques comme une solution. Sinon retourner à **.
- PSO: évaluation de la fitness des particules
 - L'exactitude prédictive est utilisée pour mesurer la fitness d'un individu.
 - La meilleure valeur de fitness pour chaque particule est $Pbest$ qui définit la meilleure valeur de fitness trouvée par la particule dans toutes les itérations précédentes.
 - La meilleure valeur de fitness dans un groupe de $Pbest$ est la valeur globale de fitness $Gbest$.
 - Mettre à jour la position et la vitesse de chaque particule.

Étape 3: Évaluer le sous-ensemble sélectionné.

- Évaluer l'importance du sous-ensemble sélectionné de chaque fourmi ($Fbest$) et particule ($Pbest$) grâce à la performance du classifieur et choisir le meilleur.
 - Si $Pbest > Fbest \rightarrow Fbest = Pbest$
 - Sinon $\rightarrow Pbest = Fbest$

Étape 4: Vérifier le critère d'arrêt.

- Si le nombre d'itérations est supérieur au nombre maximal d'itérations, le processus est terminé, sinon il continue.

Étape 5: Mettre à jour la phéromone, et aller à l'étape 2.

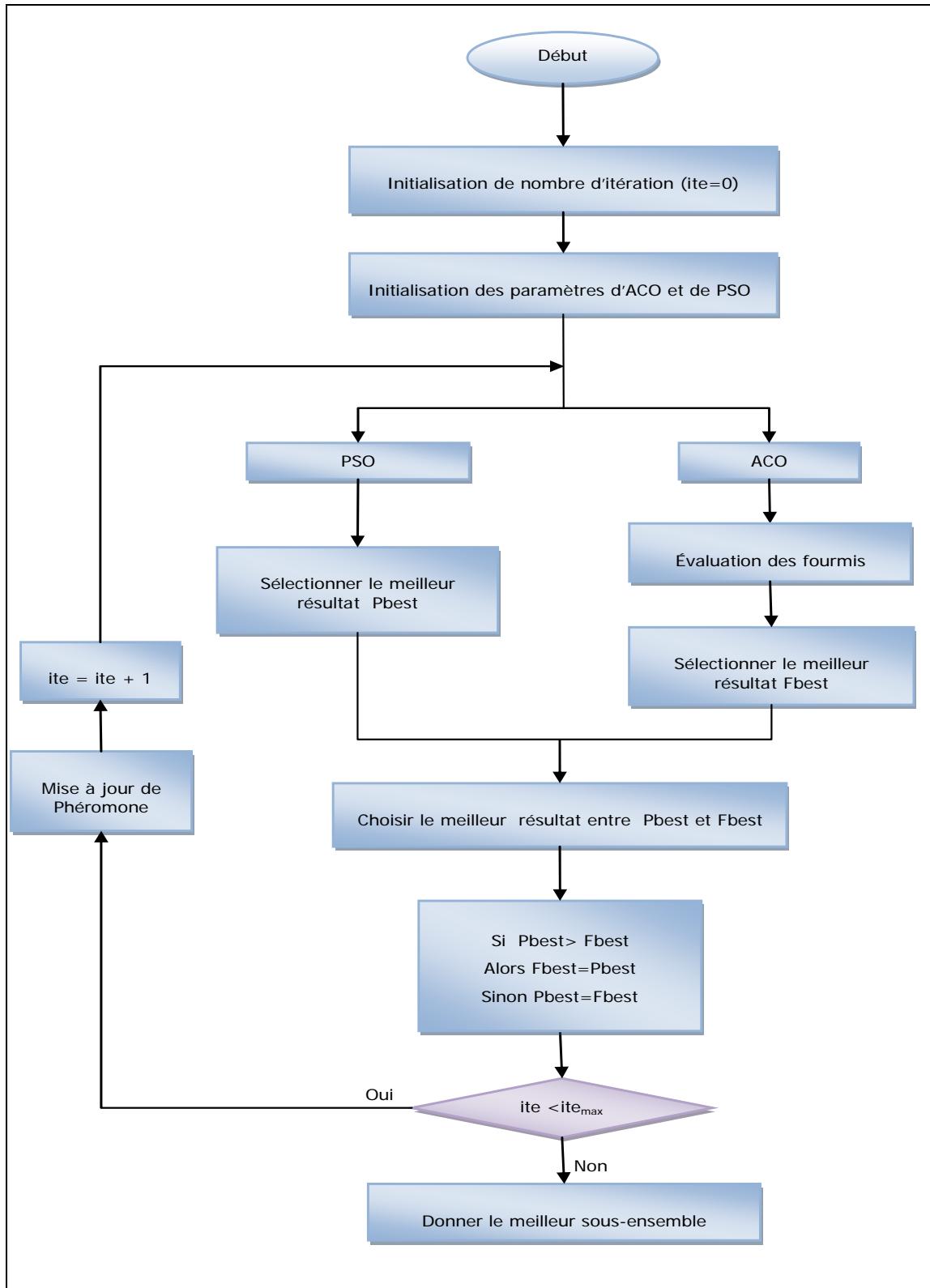


Figure 4.3. L'approche hybride proposée ACO-PSO1

4.4.2. L'approche proposée ACO-PSO2

L'idée fondamentale de cette approche hybride est d'utiliser ACO au début du processus de recherche. Lorsque la valeur de la fonction de fitness n'a pas été modifiée pendant plusieurs générations, ou qu'elle a été modifiée avec une valeur très faible (par exemple: inférieure à 0,0001), le processus de recherche est passé à l'algorithme PSO. Les meilleurs sous-ensembles trouvés par la population de fourmis de l'ACO forment donc la population initiale de PSO.

L'approche ACO-PSO2 est représentée par la Figure 4.4, le processus commence par la génération d'un nombre de fourmis qui est égal au nombre de caractéristiques dans la base de données, ces fourmis sont aléatoirement placées dans le graphe (c'est à dire que chaque fourmi commence par un sous ensemble qui contient une seule caractéristique aléatoire). Chaque fourmi commence la construction du chemin d'accès à une caractéristique différente. A partir de ces positions initiales, elles parcourent les arcs de façon probabiliste (en utilisant l'équation 2.2) jusqu'à ce qu'un critère d'arrêt soit satisfait (l'erreur courante du classifieur est inférieure à l'erreur du classifieur obtenue avec l'ensemble complet des caractéristiques). Les sous-ensembles qui en résultent sont collectés puis évalués. Si un certain nombre d'itérations est atteint, ou la valeur de la fitness n'est pas modifiée (ou qu'elle l'est très légèrement) pendant plusieurs générations, alors le processus de recherche passe à l'algorithme PSO. Dans le cas contraire, la phéromone est mise à jour, un nouvel ensemble de fourmis est créé et le processus se répète, une fois de plus.

Dans cette étape, la position et la vitesse de chaque particule sont générées en fonction des meilleurs résultats trouvés par les fourmis. La position est représentée sous forme d'une chaîne binaire où les valeurs binaires 1 et 0 représentent une caractéristique sélectionnée et non sélectionnée, respectivement.

L'exactitude prédictive est utilisée pour mesurer l'aptitude d'un individu. La meilleure valeur de fitness pour chaque particule est P_{best} qui définit la meilleure valeur de fitness trouvée par la particule dans toutes les itérations précédentes. La meilleure valeur de fitness dans un groupe de P_{best} est le G_{best} qui est la valeur de fitness globale. Après l'évaluation de la fitness, la position et la vitesse de chaque particule sont mises à jour. Si l'algorithme atteint un certain nombre d'itérations ou la meilleure valeur de fitness n'est pas modifiée après 10 itérations successives, le processus s'arrête

et fournit en sortie le meilleur sous-ensemble de caractéristiques rencontrées au cours de toutes les itérations. Si les conditions ne sont pas vérifiées, le processus réitère une fois de plus.

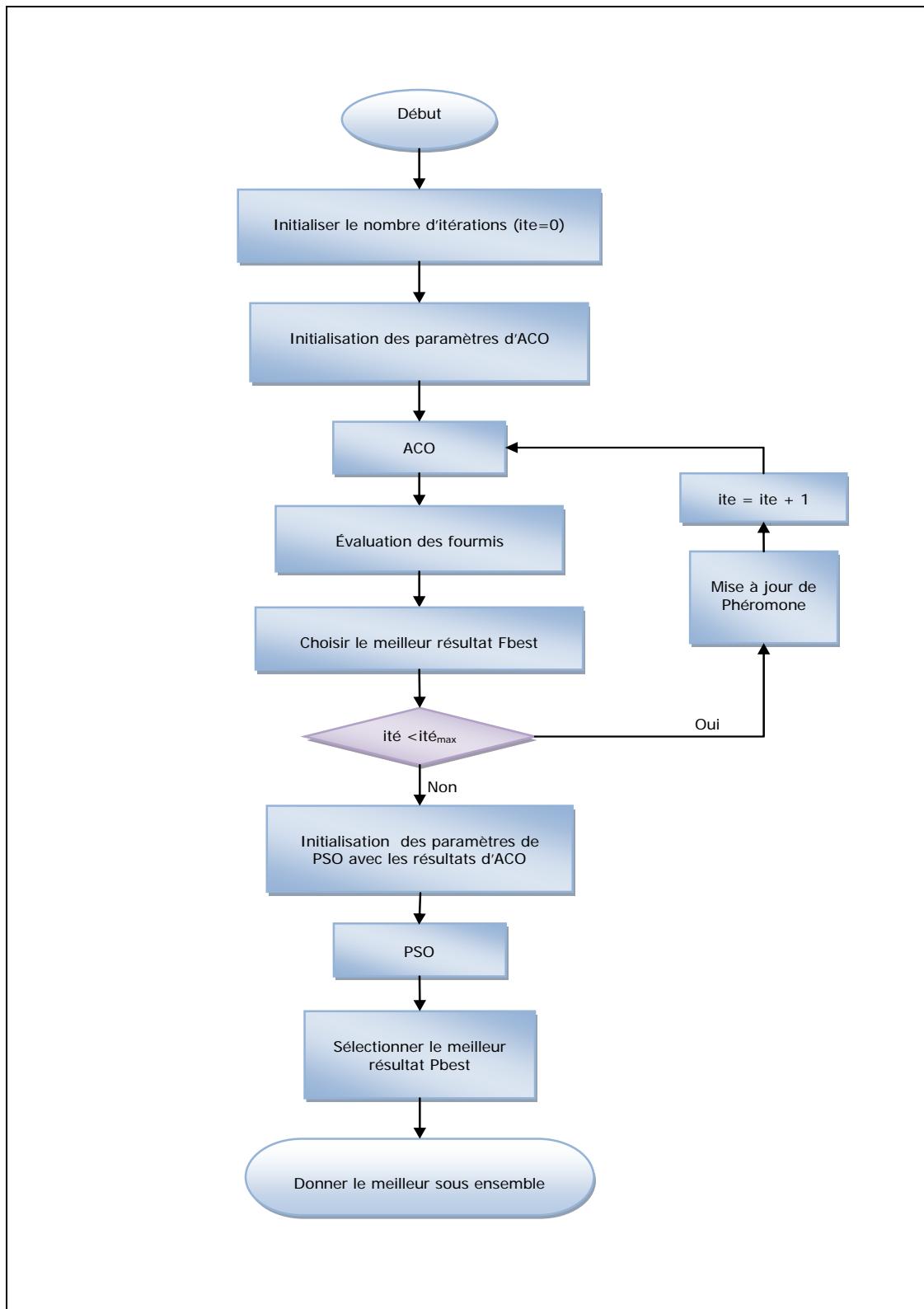


Figure 4.4. L'approche hybride proposée ACO-PSO2

4.4.3. L'approche proposée ACO-PSO3

L'approche proposée ACO-PSO3 pour la sélection d'attributs est une adaptation de l'algorithme PS-ACO, proposé par Shuang et al. [SHU 11], à notre problème de sélection des caractéristiques. Shuang et al [SHU 11] ont présenté un nouvel algorithme hybride intégrant les idées des algorithmes PSO et ACO et ont testé cet algorithme avec le problème bien connu du voyageur de commerce. Dans ce nouvel algorithme, les règles de mise à jour de la phéromone de l'ACO sont combinées avec les mécanismes de recherche locale et globale de PSO. D'une part, l'espace de recherche est développé par l'exploration locale; d'autre part, le processus de recherche est dirigé par l'exploration globale.

Dans PS-ACO, les pistes de phéromones sont incrémentées. La mise à jour de phéromone $\Delta\tau_{ij}$ est améliorée avec l'idée de l'algorithme de PSO et elle est définie par:

$$\Delta\tau_{ij} = \sum_{k=1}^n \Delta\tau_{ij}^k + c1 * r1 * \sum_{k=1}^n \Delta\tau_{ij}^{\text{local } k} + c2 * r2 * \Delta\tau_{ij}^{\text{global}} \quad (4.3)$$

où $\Delta\tau_{ij}^k$ a la même formule présentée dans le Tableau 3.2 (selon la variante de ACO utilisée), $\Delta\tau_{ij}^{\text{local } k}$ est la valeur de mise à jour locale de la fourmi k sur l'arête (vi, vj) , et elle est définie par:

$$\Delta\tau_{ij}^{\text{local } k} = \frac{Q}{f_{\text{best_so_far}}^k} \quad (4.4)$$

où $f_{\text{best_so_far}}^k$ est la valeur de la fitness de la meilleure solution rencontrée par la fourmi k dès le début.

$\Delta\tau_{ij}^{\text{global}}$ est la valeur de la mise à jour globale de l'arête (vi, vj) qui est donnée par:

$$\Delta\tau_{ij}^{\text{global}} = \frac{Q}{f_{\text{best_so_far}}} \quad (4.5)$$

où $f_{\text{best_so_far}}$ est la valeur de fitness de la meilleure solution rencontrée par toutes les fourmis dès le début.

L'équation (4.3) montre le renforcement des meilleures solutions rencontrées par les fourmis et l'expérience de toute la colonie de fourmis. La règle de mise à jour locale souligne les meilleures solutions locales de fourmis. Les meilleures solutions locales sont concentrées autour d'une petite région de l'espace total de recherche, de sorte que l'espace de recherche soit distribué ou élargi avec des pistes de phéromones

supplémentaires déposées sur différents meilleures solutions locales de fourmis. La règle globale de mise à jour met l'accent sur les meilleures solutions globales de la procédure de recherche, qui exploite et dirige la recherche de toute la colonie avec le renforcement des meilleures solutions trouvées globalement. En conséquence, le nouvel algorithme hybride PS-ACO fait bouger la recherche vers les meilleurs espaces de solutions, en maintenant la possibilité de rechercher des zones plus larges autour des meilleures solutions de manière synchrone, c'est-à dire le PS-ACO peut converger plus rapidement et plus précisément.

4.5. Évaluation des approches hybrides proposées

4.5.1. Bases de données et outils utilisés

Nous avons utilisé les mêmes outils présentés dans la section 3.6 :

- Les paramètres d'ACO et de PSO sont les mêmes utilisées dans les approches non hybrides basées sur l'ACO et le PSO (voir les Tableaux 3.3 et 3.4).
- Nous avons utilisé le taux d'erreur équilibré (BER) pour évaluer la performance des approches hybrides (voir section 3.6.1.2).
- Le classifieur Naïve Bayes est utilisé dans nos expérimentations.

Pour pouvoir comparer les résultats des approches hybrides avec ceux des approches non hybrides, nous avons utilisé les mêmes bases de données présentées dans le chapitre précédent (voir les Tableaux 3.5 et 3.7) : Spambase, Breast Cancer, German, Hepatitis, Liver et Musk. Pour ACO, nous avons utilisées la variante MMAS, puisque nous avons constaté qu'elle a donné les meilleurs résultats précédemment (voir section 3.6.2.2).

4.5.2. Résultats expérimentaux

L'évaluation des performances des méthodes hybrides de sélection d'attributs proposées, nécessite une comparaison avec les performances des approches de sélection de caractéristiques basées sur ACO et PSO développées dans le chapitre précédent. Les

Tableaux 4.3 jusqu'à 4.8 présentent les résultats en termes de nombre de caractéristiques sélectionnées et erreurs de classification pour les six bases de données.

A partir des Tableaux 4.3 jusqu'à 4.8, nous pouvons observer que toutes les approches hybrides ont produit des résultats significatifs en termes de réduction du nombre des caractéristiques sélectionnées et d'amélioration des performances de classification avec toutes les bases de données utilisées. L'approche hybride ACO-PSO1 a atteint les meilleurs résultats en termes de performances de classification avec la majorité des bases de données, suivie par l'approche basée sur PSO et ACO-PSO2. L'approche basée sur PSO a produit les meilleurs résultats avec 2 bases de données (German and Musk) alors que la meilleure réduction d'erreur est atteinte pour la base Spambase. Si on exclut les bases de données Spambase et German, l'approche hybride ACO-PSO2 n'a pas produit un meilleur résultat que l'approche basée ACO.

Méthode	Erreur	Nbr de caractéristiques sélectionnées	Les caractéristiques sélectionnées
Avant sélection	0.3128	57	/
ACO	0.1992	8	36, 47, 25, 41, 48, 33, 39, 27
PSO	0.1545	16	2, 15, 24, 25, 26, 27, 30, 31, 32, 33, 35, 38, 42, 43, 46, 48
ACO-PSO1	0.1362	34	7, 8, 20, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 37, 38, 39, 40, 41, 45, 47, 50, 51, 53
ACO-PSO2	0.1937	21	2, 4, 8, 12, 14, 25, 27, 30, 32, 33, 34, 36, 38, 40, 43, 44, 45, 46, 47, 48, 55
ACO-PSO3	0.1951	15	45, 38, 30, 47, 36, 14, 55, 2, 43, 27, 39, 33, 48, 41, 25

Tableau 4.3. Comparaison des performances des différentes approches de sélection de caractéristiques pour la base Spambase

Méthode	Erreur	Nbr de caractéristiques sélectionnées	Les caractéristiques sélectionnées
Avant sélection	0.4433	9	/
ACO	0.3033	2	9, 1
PSO	0.3483	2	1, 5
ACO-PSO1	0.2550	2	1, 7
ACO-PSO2	0.3033	2	1, 9
ACO-PSO3	0.3033	2	9, 1

Tableau 4.4. Comparaison des performances des différentes approches de sélection de caractéristiques pour la base Breast Cancer

Méthode	Erreur	Nbr de caractéristiques sélectionnées	Les caractéristiques sélectionnées
Avant sélection	0.4990	24	/
ACO	0.3680	2	1, 9
PSO	0.2775	8	1, 2, 8, 13, 14, 16, 21, 22
ACO-PSO1	0.2825	8	1, 4, 9, 12, 14, 16, 23, 24
ACO-PSO2	0.2855	4	1, 4, 16, 21
ACO-PSO3	0.3355	2	4, 1

Tableau 4.5. Comparaison des performances des différentes approches de sélection de caractéristiques pour la base German

Méthode	Erreur	Nbr de caractéristiques sélectionnées	Les caractéristiques sélectionnées
Avant sélection	0.5000	19	/
ACO	0.3836	2	5, 7
PSO	0.2760	7	2, 5, 9, 12, 15, 17, 19
ACO-PSO1	0.2701	3	1, 2, 5
ACO-PSO2	0.2930	8	4, 5, 7, 10, 11, 15, 16, 17
ACO-PSO3	0.3836	3	9, 7, 5

Tableau 4.6. Comparaison des performances des différentes approches de sélection de caractéristiques pour la base Hepatitis

Méthode	Erreur	Nbr de caractéristiques sélectionnées	Les caractéristiques sélectionnées
Avant sélection	0.5	6	/
ACO	0.5	6	1, 2, 3, 4, 5, 6
PSO	0.4182	1	5
ACO-PSO1	0.4010	4	2, 3, 4, 5
ACO-PSO2	0.4071	5	2, 3, 4, 5, 6
ACO-PSO3	0.5	6	1, 2, 3, 4, 5, 6

Tableau 4.7. Comparaison des performances des différentes approches de sélection de caractéristiques pour la base Liver

Méthode	Erreur	Nbr de caractéristiques sélectionnées	Les caractéristiques sélectionnées
Avant sélection	0.3725	166	/
ACO	0.2510	16	25, 147, 90, 75, 62, 111, 9, 132, 18, 165, 103, 36, 73, 137, 104, 163
PSO	0.2277	26	1, 15, 16, 18, 20, 23, 26, 36, 37, 48, 50, 51, 52, 53, 60, 67, 78, 80, 82, 92, 94, 97, 103, 104, 106, 124, 129, 132, 136, 145, 147, 151, 155, 160
ACO-PSO1	0.2474	52	1, 6, 8, 12, 13, 17, 24, 25, 28, 29, 32, 34, 46, 49, 50, 52, 53, 54, 56, 73, 74, 76, 83, 84, 91, 96, 97, 98, 102, 104, 106, 108, 115, 116, 122, 123, 124, 129, 131, 133, 134, 135, 140, 147, 148, 150, 152, 155, 161, 162, 165, 166
ACO-PSO2	0.2309	28	8, 11, 14, 15, 18, 36, 38, 40, 59, 62, 67, 73, 75, 90, 94, 104, 126, 130, 131, 132, 135, 137, 138, 144, 147, 161, 163, 164
ACO-PSO3	0.2527	41	15, 32, 38, 161, 40, 164, 152, 147, 90, 75, 62, 107, 70, 5, 149, 106, 127, 59, 79, 125, 20, 92, 19, 157, 29, 103, 165, 83, 67, 4, 135, 130, 18, 132, 143, 104, 137, 73, 36, 14, 163

Tableau 4.8. Comparaison des performances des différentes approches de sélection de caractéristiques pour la base Musk

4.6. Conclusion

Dans ce chapitre, nous avons présenté des approches hybrides bio-inspirés proposées pour la sélection d'attributs. Ces approches sont basées sur l'intégration de deux méthodes qui sont l'optimisation par les colonies de fourmis (ACO) et l'optimisation par essaim de particules (PSO). Les performances de ces approches sont comparées avec les approches bio-inspirées non hybrides proposées pour la sélection d'attributs basée sur ACO et PSO et présentées dans le chapitre précédent. Toutes les approches bio-inspirées proposées sont testées sur six bases de données publiques. Nos résultats expérimentaux montrent l'intérêt de l'utilisation des approches bio-inspirées ainsi que celui de leur hybridation dans le cadre de la sélection d'attributs car ces approches peuvent efficacement réduire le nombre de caractéristiques et améliorer la précision de la classification.

CONCLUSION ET PERSPECTIVES

Plusieurs paramètres peuvent influer les performances d'un système de classification. Les caractéristiques extraites à partir des entités considérées afin de représenter ces dernières, peuvent être considérées parmi les paramètres les plus importants. La sélection des caractéristiques ou attributs est l'une des étapes les plus importantes dans le traitement des données dans différents domaines applicatifs. Elle consiste à trouver un sous-ensemble optimal de caractéristiques pertinentes. La sélection d'attributs est un problème d'optimisation combinatoire multi-objectif. Son but principal étant de réduire le nombre de caractéristiques utilisées, tout en essayant de maintenir ou d'améliorer les performances de la classification.

La sélection d'attributs peut ainsi être définie comme un problème d'optimisation. Nous avons donc commencé notre travail en étudiant les méthodes d'optimisation avec un intérêt particulier pour les algorithmes bio-inspirés évolutionnaires qui sont des méthodes très efficaces et largement utilisées dans ce domaine. Nous nous sommes ensuite focalisés sur les concepts de base de la sélection de caractéristiques ainsi que les différentes catégories de méthodes utilisées pour sélectionner des attributs.

Afin de mieux orienter nos travaux, nous avons effectué une étude bibliographique approfondie de différents travaux existants dans la littérature concernant les approches bio-inspirées de sélection d'attributs. Dans la plupart des travaux que nous avons étudiés, les auteurs présentent des méthodes de sélection d'attributs basées sur des approches wrapper, faisant intervenir les performances des classifieurs utilisés. La majorité de ces méthodes utilisent les mécanismes de recherche issus de l'intelligence en essaims (particulièrement les colonies de fourmis et les essaims particulaires) pour explorer l'espace de recherche afin de produire des sous ensembles optimaux. Après l'analyse des travaux existants, nous avons constaté que les colonies de fourmis (ACO) offrent l'avantage principal de la recherche locale tandis que les essaims particulaires (PSO) considèrent une perspective globale. D'autre part, la tendance récente des recherches, dans l'informatique bio-inspirée, s'oriente vers l'hybridation des méthodes

bio-inspirées pour profiter des avantages de deux ou plusieurs méthodes et en atténuer les inconvénients.

Ainsi, l'objectif final que nous nous sommes fixés pour notre travail est d'étudier l'apport de l'hybridation entre des méthodes bio-inspirées dans le domaine de la sélection d'attributs.

Nous avons commencé par la proposition de trois approches pour la sélection d'attributs, la première est classique, basée sur le principe du Feature Ranking et les deux autres sont bio-inspirées, non hybrides, basées, respectivement, sur ACO et PSO.

Dans l'approche de sélection basée sur ACO nous avons proposé d'utiliser une mesure hybride d'évaluation qui peut estimer la performance globale du sous-ensemble aussi bien que l'importance locale des caractéristiques. L'algorithme de classification est utilisé pour estimer la performance des sous-ensembles (autrement dit, une fonction d'évaluation de type wrapper). D'autre part, l'importance locale d'une caractéristique donnée est mesurée en utilisant la corrélation entre les caractéristiques, qui représente une fonction d'évaluation de type filtre. L'approche de sélection basée sur les essaims particulaires est une implémentation de la variante binaire de l'algorithme PSO. Dans cette approche l'importance des sous-ensembles de caractéristiques sélectionnées est évaluée en utilisant la performance du classifieur.

Les résultats de nos expérimentations montrent que les approches proposées ont permis d'atteindre des taux d'erreur réduits avec de petits sous-ensembles de caractéristiques. Parmi les variantes de l'algorithme ACO, la variante MMAS est la plus performante. De plus, la méthode de sélection basée sur les PSO donne généralement des résultats meilleurs que ceux de la méthode basée sur les ACO en termes de réduction de l'erreur de classification. D'autre part, ACO a permis d'obtenir des résultats meilleurs que PSO, en termes de nombre de caractéristiques sélectionnées.

La contribution finale de notre travail consiste à proposer trois approches de sélection d'attributs basées sur des hybridations entre les méthodes d'optimisation par les essaims particulaires et par les colonies de fourmis.

L'idée de base de la première approche hybride proposée (ACO-PSO1) est d'utiliser les mécanismes de recherche d'ACO et de PSO simultanément pour explorer l'espace de recherche. Les performances des sous-ensembles sélectionnés sont évaluées avec la

précision de la classification après chaque itération, et le meilleur sous-ensemble est utilisé comme meilleure solution trouvée pour ACO et PSO en même temps.

Dans la deuxième approche hybride proposée (ACO-PSO2), ACO est utilisé pour explorer l'espace de recherche. Lorsque la performance du meilleur sous-ensemble sélectionné n'est plus améliorée de manière significative pendant plusieurs générations, le processus de recherche continue avec PSO.

La troisième approche hybride proposée (ACO-PSO3) est une adaptation de l'algorithme PS-ACO proposé par Shuang et al [SHU 11] à notre problème de sélection d'attributs.

Les résultats de ces trois approches hybrides sont comparés avec ceux des approches bio-inspirées non hybrides proposées précédemment en utilisant six bases de données publiques. Nos résultats expérimentaux ont montré, qu'avec toutes les approches testées, nous avons réduit le nombre de caractéristiques et amélioré les taux de classification correcte.

Comme dans tous les travaux de recherche, il est certain que les solutions proposées ne sont pas les meilleures d'une manière absolues, plusieurs améliorations sont envisageables et plusieurs points peuvent être étudiés dans le cadre de travaux futurs, tels que :

- L'utilisation d'autres jeux de données pour la validation des approches proposées. Les données utilisées sont généralement de taille moyenne, nous avons choisi des bases de données publiques qui sont largement utilisées dans les travaux existants. On peut appliquer les approches proposées dans des domaines applicatifs précis tels que l'identification biométrique ou la reconnaissance de l'écriture manuscrite en utilisant des bases de données spécifiques avec de très grandes tailles.
- L'utilisation de mécanismes pour trouver les meilleures valeurs possibles des paramètres des approches de sélection basées ACO et PSO. En effet, dans notre travail, ces paramètres sont réglés de manière expérimentale alors qu'ils peuvent l'être en utilisant des méthodes ou des fonctions d'adaptation. Dans nos travaux futurs, on peut envisager d'utiliser ces mécanismes pour l'optimisation de ces paramètres.

- L'utilisation et, éventuellement, l'hybridation d'autres méthodes de la swarm intelligence différentes d'ACO et de PSO. En effet, suite à notre étude et au constat de la rareté des travaux incluant les colonies des abeilles pour la sélection d'attributs, nous envisageons, éventuellement, de l'intégrer dans nos futurs travaux. Il est aussi possible d'envisager d'intégrer d'autres méthodes de la Swarm intelligence telles que Bat algorithm, Bacterial Foraging Optimization, Firefly algorithm,...etc.

BIBLIOGRAPHIE

- [AGH 09] Aghdam M. H., Ghasem-Aghaee N., Basiri M. E., “Text feature selection using ant colony optimization”, Expert Systems with Applications Vol. 36, Issue 3, Part 2 , pp 6843–6853, 2009.
- [AKI 08] Akinori H., Takio K., “Non-Neighboring Rectangular Feature Selection using Particle Swarm Optimization”, In IEEE Proc. of Inter. Conference on Pattern Recognition, pp 1-4, Tampa, FL, États-Unis, 8-11 Dec. 2008.
- [ALA 08] Alatas B., Erhan A., “Rough particle swarm optimization and its applications in data mining”, Soft Comput, Vol. 12, pp 205–1218., Springer-Verlag, 2008.
- [ALM 91] Almuallim H., Dietterich T.G., “Learning with many irrelevant features”. In: Proc. of The Ninth National Conference on Artificial Intelligence (AAAI-91), Anaheim, California pp 547–552, July 14-19, 1991.
- [ANI 05] Al-Ani A. “Feature subset selection using ant colony optimization”, International Journal of Computational Intelligence, Vol. 2, pp 53–58. 2005
- [AWA 12] Awada W., Khoshgoftaar T.M., Dittman D., Wald R., Napolitano A., “A review of the stability of feature selection techniques for bioinformatics data”, In proc.of IEEE 13th International Conference on Information Reuse and Integration (IRI), Tuscany Las Vegas, NV, USA , pp 356–363, 08-10 août 2012.
- [AZZ 04] Azzag H., Picarougne F., “Guinot C. and Venturini G. “Un survol des algorithmes biomimétiques pour la classification”, in Classification et Fouille de données, Vol. RNTI-C-1, pp 13-24, 2004
- [BAE 10] Bae, C., Yeh, W.C., Chung, Y.Y. and Liu, S.L. ‘Feature selection with Intelligent Dynamic Swarm and Rough Set’, Expert Systems with Applications, Vol. 37, pp. 7026–7032. 2010

- [BAR 11] Barrera J., Coello C. A. C. “Test Function Generators for Assessing the Performance of PSO Algorithms in Multimodal Optimization”, In: Panigrahi B.K., Shi Y., Lim M.-H. (Eds.): “Handbook of Swarm Intelligence”, Vol. 8, pp 89–117, 2011.
- [BAS 08] Basiri,M.E., Ghasem,A.N., Aghdam M.H. “Using Ant Colony Optimization-Based Selected Features for Predicting Post-synaptic Activity in Proteins”, In Marchiori E., Moore J.H. (Eds.): “EvoBIO” proc. of 6th European conference, , LNCS Vol. 4973, pp 12–23, Naples, Italie, 26-28 mars 2008.
- [BAS 09] Basiri M.E., Nemati S. “A novel hybrid ACO-GA algorithm for text feature selection”. in Proc. of IEEE Congress on Evolutionary Computation, pp 2561-256, Trondheim, Norvège, 18-21 Mai 2009
- [BEL 11] Belanche L.A., Gonzalez F.F., “Review and Evaluation of Feature Selection Algorithms in Synthetic Problems”, Arxiv preprint arXiv:1101.2320, 2011.
- [BEN 02] Benahmed N., “Optimisation de réseaux de neurones pour la reconnaissance de chiffres manuscrits isolés : Sélection et pondération des primitives par algorithmes génétiques”, Mémoire de Maîtrise, Université du Québec, Canada, 2002.
- [BIN 13] Bing, X., Mengjie, Z. and Will, N.B. ‘Novel Initialisation and Updating Mechanisms in PSO for Feature Selection in Classification’, EvoApplications, LNCS, Vol. 7835, pp. 428–438. 2013
- [BRO 11] Brownlee J. “Clever Algorithms Nature-Inspired Programming Recipes”, Lulu Enterprises, Australia, pp 257-264, 2011.
- [BUL 99] Bullnheimer B., Hartl R.F., Strauss C. “A new rank-based version of the ant system: a computational study”, Central European Journal for Operations Research and Economics, Vol. 7, pp 25–38. 1999
- [CAI 10] Cai Y., “Artificial fish school algorithm applied in a combinatorial optimization problem”. International Journal of Intelligent Systems and Applications, Vol.1, pp 37–43. 2010.
- [CAK 02] Cakmakov D., Bennani, Y. “Feature selection for pattern recognition”, Skopje, Informa. 2002.

- [CHE 10] Chen Y., Miao D., Wang, R. “A rough set approach to feature selection based on ant colony optimization”, *Pattern Recognition Letters*, Vol. 31. pp 226–233. 2010.
- [CHE 15] Chen D., Zou F., Li Z., Wang J., Li S., “An improved teaching-learning-based optimization algorithm for solving global optimization problem, *Information Sciences*”, *Information Sciences*, Vol. 297, pp 171–190, 2015.
- [CHI 09] Chiong R., Neri F., McKay R.I. “Nature that Breeds Solutions”, In Chiong R. (Eds,) *Nature-Inspired Informatics for Intelligent Applications and Knowledge Discovery: “Implications in Business, Science, and Engineering”*, *Information science reference*, Hershey, New York, pp 1-24, 2009.
- [CHO 08] Chouaib H., Terrades O. R. Tabbone S., Cloppet F., Vincent N., “Feature selection combining genetic algorithm and Adaboost classifiers”. In proc. of 19th International Conference on Pattern Recognition, Tampa, FL, 8-11 Déc 2008.
- [CHU 08] Chuang, L.Y., Chang, H.W., Tu C.J., Yang C.H. “Improved binary PSO for feature selection using gene expression data”, *Computational Biology and Chemistry*, Vol. 32, No. 1, pp 29–38. 2008.
- [COR 08] Cordella L., De Stefano C., Fontanella F., Marrocco C., “A Feature Selection Algorithm for Handwritten Character Recognition”. In: proc. Of the 19th Internat. Conf. On Pattern Recognition (ICPR), pp 1–4. 2008.
- [DAN 04] Dangauthier P, Spalanzani A et Bessière P, “Méthodes statistiques et algorithmes génétiques pour la sélection de connaissances préalable”, *Actes du congrès francophone de Reconnaissance des Formes et Intelligence Artificielle*, Toulouse (FR), France, Jan 2004.
- [DAS 97] Dash M. and Liu H. “Feature Selection Methods for Classification”, *Intelligent Data Analysis*, Vol. 1, pp 131–156. 1997.
- [DAS 09] Das S., Abraham A., Konar A. “Metaheuristic Clustering”, *Studies in Computational Intelligence* (Book 178), Springer-Verlag Berlin Heidelberg. pp 1–62. 2009.

- [DEN 09] Deng T., Yang C., Zhang Y., Wang X. “An Improved Ant Colony Optimization Applied to Attributes Reduction”, In Cao B., Zhang C. et Li T. (Eds.): “Fuzzy Info. and Engineering”, Advances in Soft Computing Vol. 54, Springer, pp. 1–6. 2009.
- [DIW 11] Diwold K., Beekman M., Middendorf M. “Honeybee Optimisation – An Overview and a New Bee Inspired Optimisation Scheme”, In Panigrahi B.K., Shi Y., et Lim M.-H. (Eds.): Handbook of Swarm Intelligence, Adaptation, Learning, and Optimization Vol. 8, pp 295–327. 2011
- [DOR 97] Dorigo M., Gambardella L.M., “Ant Colony System: a cooperative learning approach to the traveling salesman problem”. Evolutionary Computation, IEEE Transactions on, Vol. 1, issue 1, pp 53-66, 1997.
- [FOG 66] Fogel L. J., Owens A. J., Walsh M. J. “Artificial Intelligence Through Simulated Evolution”. John Wiley, Chichester, UK, (1966): cité dans [SIM 13a].
- [FON 93] Fonsecay C. M., Fleming P. J. “Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization”, In S. Forrest, (ed.) : “Genetic Algorithms”: Proc. of the Fifth International Conference on Genetic Algorithms, pp 416-423, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA , 1993.
- [FOR 87] Foroutan I., Sklansky J., “Feature selection for automatic classification of non- Gaussian data”, IEEE Trans. on Systems, Man and Cybernetics, Vol. 17, issue 2, pp 187-198, 1987.
- [GAM 96] Gambardella L.M., Dorigo M. “Solving symmetric and asymmetric TSPs by ant colonies”, Proc. IEEE Int. Conf. on Evolutionary Computation, ICEC'96, pp 622-627, Nagoya, Japon, 20-22 Mai 1996.
- [GAV 06] Gavrilis D., Tsoulos I.G., Dermatas E., “Neural Recognition and Genetic Features Selection for Robust Detection of E-Mail Spam”, in G. Antoniou et al. (Eds.): proc. Of 4th Hellenic Conference on AI, SETN 2006, LNAI Vol. 3955, pp 498 – 501 Heraklion, Crete, Greece, 18-20 Mai 2006.

- [GRA 03] Grandidier F., “Un nouvel algorithme de sélection de caractéristiques-Application à la lecture automatique de l’écriture manuscrite”, Thèse de doctorat, Université du Québec, Canada 2003.
- [GRO 06] Grosan A., Abraham A., Chis M. “Swarm intelligence in data mining”, Studies in Computational Intelligence, Vol. 34, pp 1–20. 2006
- [GUA 07] Guangdong H., Ping L., Qun W., “A Hybrid Metaheuristic ACO-GA with an Application in Sports Competition Scheduling”. In proc. of Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, Vol. 3, pp. 611– 616, Qingdao, Chine 30 Juil- 1 août 2007.
- [GUE 06] GUÉRIF S., “Réduction de dimension en Apprentissage Numérique Non Supervisé”, Thèse de doctorat, Université de Paris13, France, 11 déc 2006.
- [GUZ 09] Guzella T.S., Caminhas W.M, “A review of machine learning approaches to Spam filtering”, Expert Systems with Applications Vol. 36, issue 7, pp10206–10222, 2009.
- [HAL 99] Hall M.A. “Correlation-based Feature Selection for Machine Learning”. PhD Thesis University of Waikato, Australia. 1999.
- [HOL 75] Holland, J. H. “Adaptation in Natural and Artificial Systems”. University of Michigan Press, Ann Arbor, 1975. cité dans [SIM 13a].
- [JEN 03] Jensen R., Shen Q., “Finding Rough Set Reducts with Ant Colony Optimization”, In Proc. of the 2003 UK Workshop on Computational Intelligence, pp. 15-22. 2003.
- [JEN 05] Jensen R., “Combining rough and fuzzy sets for feature selection”, thèse de Doctorat, Université de Edinburgh, Royaume-Uni, 2005
- [JIN 12] Jin C., Jin S-W., Qin, L-N. “Attribute selection method based on a hybrid BPNN and PSO algorithms”. Applied Soft Computing, Vol.12, issue 8. pp 2147–2155. 2012.

- [JOU 03] Jourdan L., “Métaheuristiques Pour L'extraction De Connaissances : Application À La Génomique”, thèse de doctorat, Université Des Sciences Et Technologies De Lille I, Frence 2003.
- [JUA 04] Juang C-F, “A hybrid of genetic algorithm and particle swarm optimization for recurrent network design”.Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions, Vol. 34, No 2, pp 997–1006. 2004
- [KAB 09] Kabir M.M., Shahjahan M., Murase K. “Involving New Local Search in Hybrid Genetic Algorithm for Feature Selection”, In: Leung, C.S., Lee, M. and Chan, J.H. (Eds.): “ICONIP”, Part II, , Vol. 5864, pp 150–158. 2009.
- [KAB 12] Kabir M.M., Shahjahan M., Murase K, “A new hybrid ant colony optimization algorithm for feature selection”, Expert Systems with Applications, Vol. 39, pp. 3747–3763. 2012
- [KAN 07] Kanan H., Faez K., Taheri S., “Feature selection using ant colony optimization (ACO): A new method and comparative study in the application of face recognition system”, In: Perner, P. (ed.): Advances in Data Mining. Theoretical Aspects and Applications proc. of the 7th Industrial Conference, ICDM 2007, Vol. 4597, pp. 63–76, Leipzig, Germany, 14-18 juillet, 2007.
- [KAN 08] Kanan H.R., Faez K. “An improved feature selection method based on ant colony optimization (ACO) evaluated on face recognition system”, Applied Mathematics and Computation, Vol 205, issue 2, pp 716–725, 2008.
- [KAR 05] Karaboga D., “An idea based on honey bee swarm for numerical optimization”. Technical report. Computer Engineering Department, Engineering Faculty, Erciyes University, Turkey, 2005.
- [KAR 14] Karaboga D. Gorkemli B. Ozturk C. Karaboga N., “ A comprehensive survey: artificial bee colony (ABC) algorithm and applications”, Vol 42, Issue 1, pp 21-57, 2014
- [KEN 95] Kennedy J., Eberhart RC., “Particle swarm optimization”. In: Proc. of IEEE conference on neural networks, Perth, Australia, pp 1942–1948. 1995.

- [KEN 97] Kennedy J., Eberhart R.C., “A discrete binary version of the particle swarm algorithm”, in Proc. of the International Conf. on Systems, Man and Cybernetics, Computational Cybernetics and Simulation, Vol 5, pp. 4104–4109, Orlando, FL, États-Unis, 12-15 Oct 1997.
- [KEN 06] Kennedy J. “Swarm Intelligence”, in Zomaya A.Y. (Ed.): “Handbook Of Nature-Inspired And Innovative Computing: Integrating Classical Models with Emerging Technologies”, Springer-Verlag US , , pp. 187-219, 2006.
- [KHU 08] Khushaba R.N., Al-Ani A., AlSukker A., Al-Jumaily A., “A Combined Ant Colony and Differential Evolution Feature Selection Algorithm”, In: Dorigo M. et al. (Eds.): proc. of 6th International Conference ANTS 2008, LNCS Vol. 5217, pp. 1–12, Brussels, Belgium, 22-24 Sep 2008.
- [KIR 92] Kira K., Rendell L, “A practical approach to feature selection”, in: Proc. of the ninth international workshop on machine learning Morgan Kaufmann Publishers Inc. San Francisco, CA, USA , pp.249-256 .1992.
- [KIR 12] Kiran M.S., Ozceylan E., Gunduz, M., Paksoy T., “A novel hybrid approach based on Particle Swarm Optimization and Ant Colony Algorithm to forecast energy demand of Turkey”. Energy Conversion and Management Vol. 53, issue 1, pp 75–83. 2012
- [KOR 10] Kordon A., “Applying Computational Intelligence: How to Create Value”. Chapter 6: Swarm Intelligence: The Benefits of Swarms. Springer-Verlag Berlin Heldelberg, 2010.
- [LAD 11] Ladha L., Deepa T., “Feature Selection Methods And Algorithms”, International Journal on Computer Science and Engineering (IJCSE), Vol. 3 No. 5, pp 1787- 1797, May 2011.
- [LAI 09] Lai C.C., Wu C-H., Tsai M-C. “Feature selection using particle swarm optimization with application in spam filtering”, International Journal of Innovative Computing, Information and Control, Vol. 5, No. 2, pp. 423–432. 2009.

- [LI 03] Li X. L., Shao Z. J., Qian J. X., “An optimizing method based on autonomous animats: Fish-swarm Algorithm,” System Engineering Theory and Practice, Vol. 22, pp 32-38, November 2003.
- [LI 08] Li W. T., Shi X., Xu W.L., Hei Y.Q., “Improved GA and PSO celled hybrid algorithm for antenna array pattern synthesis”. Progress In Electromagnetics Research, PIER Vol. 80, pp 461-476, 2008
- [LIM 09] Lim C.P. and Jain L.C., “Advances in Swarm Intelligence”. In: Lim C.P. et al. (Eds.): “Innovations in Swarm Intelligence”, ISBN Springer-Verlag Berlin Heidelberg, Vol 248, pp. 1–7. 2009.
- [LIU 04] Liu Y., Qin Z., Xu Z., He X. “Feature Selection with Particle Swarms”. In Zhang J., He J.-H. et Y. Fu, (Eds): “Computational and Information Science”, LNCS, Vol. 3314, pp 425–430. 2004
- [LIU 98] Liu H., Motoda H., Dash M. “A Monotonic Measure for Optimal Feature Selection”. In Proc. of the European Conference on Machine Learning, Springer Verlag. pp 101–106.1998.
- [LIU 05] Liu H., Yu L. “Toward Integrating Feature Selection Algorithms for Classification and Clustering”, IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No 4, pp 491–502. 2005
- [LIU 11] Liu Y., Wang G., Chen H., Dong H., Zhu X., Wang S. “An Improved Particle Swarm Optimization for Feature Selection”, Journal of Bionic Engineering, Vol. 8, No 2, pp. 191–200. 2011
- [MEE 12] Meena M.J., Chandran K.R., Karthik A., Samuel A.V., “An enhanced ACO algorithm to select features for text categorization and its parallelization”, Expert Systems with Applications, Vol. 39, pp 5861–5871. 2012.
- [MEN 08] Méndez J.R., Cid I., Glez-Peña D., Rocha M., Fdez-Riverola F., “A Comparative Impact Study of Attribute Selection Techniques on Naïve Bayes Spam Filters”, Perner P. (Ed.): ICDM, pp. 213–227, 2008.

- [MEN 10a] Menghour K., Souici-Meslati L., “Sélection de Caractéristiques pour le Filtrage de Spams”, CORIA’2010, Conférence sur la Recherche d’Information et ses Applications, Sousse, Tunisie, pp. 349-360, 18-20 Mars 2010.
- [MEN 10b] Menghour K., Souici-Meslati L., “Une approche de sélection d’attributs pour la reconnaissance de chiffres manuscrits”, EGC-M’2010, Conférence Internationale Maghrébine sur l’Extraction et la Gestion de Connaissances, Alger, Algérie, 13-14 Décembre 2010.
- [MEN 11a] Menghour K., Souici-Meslati L., “Sélection de caractéristiques basée sur les colonies de fourmis pour le filtrage de spams”, EGC-M’2011, Conférence Internationale Maghrébine sur l’Extraction et la Gestion de Connaissances, pp. 168-179, Tanger, Maroc, 23-25 Novembre, 2011.
- [MEN 11b] Menghour K., Souici-Meslati L., “Ant colony optimization for feature subset selection: An application to spam filtering”, CISC’2011, Conférence Internationale sur les Systèmes Complexes, Jijel, Algérie, 6-8 Décembre, 2011.
- [MEN 14] Menghour K., Souici-Meslati L., “Classical and swarm based approaches for feature selection in spam filtering”, International Journal of Advanced Intelligence Paradigms, IJAIP, InderScience Publishers, Vol. 6, No. 3, pp. 214-234, 2014.
- [MEN 15] Menghour K., Souici-Meslati L., “Hybrid ACO-PSO approaches for feature selection”, Soumis pour publication dans un journal international, en phase de reviewing après révision.
- [MIL 94] Millonas M., “Swarms, Phase Transitions, and Collective Intelligence”. In: Langton, C.G.(ed.): “Artificial Life III”, Vol. 17, pp. 417–445. Addison-Wesley Publishing Company, Reading. 1994
- [MO 14] Mo H., Xu Z., Xu L., Wu Z. et Ma H. “Constrained Multiobjective Biogeography Optimization Algorithm”. In the proc. of 4th International Conference, ICSI 2013, pp 323-329, Harbin, China, 12-15 Juin 2014

- [MUL 09] Mullen R.J., Monekosso D., Barman S., Remagnino P., “A review of ant algorithms”, Expert Systems with Applications Vol. 36, Issue 6, pp 9608-9617, 2009.
- [NAR 77] Narendra P. M., Fukunaga K., “A Branch and Bound Algorithm for Feature Subset Selection”. IEEE Transactions On Computers, Vol. C-26, No. 9, pp. 917-922, September 1977.
- [NEM 09] Nemati S., Basiri M-E., Ghasem-Aghaee N., Aghdam M. H. “A novel ACO-GA hybrid algorithm for feature selection in protein function prediction”. Expert Systems with Applications, Vol. 36, issue 10, pp 12086–12094, 2009.
- [NES 12] Neshat M., Sepidnam G., Sargolzaei M., Toosi A. N. “Artificial fish swarm algorithm: a survey of the stateof-the-art, hybridization, combinatorial and indicative applications”. Artificial Intelligence Review. Vol. 42, Issue 4, pp 965-997, December 2014.
- [NII 09] Niiniskorpi T., Aberg M.B., Wessberg J. “Particle swarm feature selection for fMRI pattern classification”, In Proc. of the 2nd ICBSSP (BIOSIGNALS), pp. 279-284, Porto, Portugal. 2009
- [NIK 10] Niknam T., Amiri B. “An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis”. Applied Soft Computing 10, issue 1, pp 183–197. 2010.
- [OH 04] Oh I-S., Lee J-S., Moon B-R., “Hybrid Genetic Algorithms For Feature Selection”, IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 26, No 11, November 2004.
- [OLI 01] Oliveira L. S., Benahmed N., Sabourin R., Bortolozzi F., Suen C. Y., “Feature Subset Selection Using Genetic Algorithms for Handwritten Digit Recognition” in the Proc. of XIV Brazilian Symposium on Computer Graphics and Image Processing, pp 362 - 369, Florianopolis, Brazil 15-18 oct 2001.

- [OLI 03a] Oliveira L. S., Sabourin R., “A methodology for feature selection using multiobjective genetic algorithms for handwritten digit string recognition”, *Inter. Journal of Pattern Recognition and Artificial Intelligence*, Vol. 17, No. 6, pp. 903-929., World Scientific Publishing Company., 2003.
- [OLI 03b] Oliveira L. S., Sabourin R., Bortolozzi F., Suen C. Y., “Feature Selection for Ensembles: A Hierarchical Multi-Objective Genetic Algorithm Approach”, *Inproc. of 7th International Conference on Document Analysis and Recognition IEEE*, Scotland, pp. 676–680. 2003.
- [OMR 04] Omran M. G. H. “Particle Swarm Optimization Methods for Pattern Recognition and Image Processing”, PHD Thesis, University of Pretoria, Pretoria, South Africa, 2004.
- [PAR 11] Parpinelli R.S., Lopes H.S., “New inspirations in swarm intelligence: a survey”, *Int. J. Bio-Inspired Computation*, Vol. 3, No. 1, pp.1–16, 2011.
- [PAR 12] Prasartvit T., Kaewkamnerpong B., et Achalakul T. “Dimensional Reduction Based on Artificial Bee Colony for Classification Problems”. D.-S. Huang et al. (Eds.): *ICIC 2011*, LNBI 6840, pp. 168–175, 2012.
- [PAS 02] Passino K.M., “Biomimicry of bacterial foraging for distributed optimization and control”. *IEEE Control System Magazine*, Vol. 22, Issue 3, pp 52–67, 2002.
- [PAT 11] Pattnaik S.S., Bakwad K.M., Devi S., Panigrahi B.K., Das S. “Parallel Bacterial Foraging Optimization” In: Panigrahi B.K., Shi Y., et Lim M.-H. (Eds.): “*Handbook of Swarm Intelligence*”, *Adaptation, Learning, and Optimization*, Vol. 8, Springer-Verlag Berlin Heidelberg, pp. 487–502. 2011.
- [PAT 14] Patel M. K., Kabat M. R., Tripathy C. R., “A hybrid ACO/PSO based algorithm for QoS multicast routing problem”. *Ain Shams Engineering Journal*, Vol. 5, pp. 113–120. 2014.
- [PHA 06] Pham D.T., Ghanbarzadeh A., Koc E., Otri S., Rahim S., Zaidi M. “The bees algorithm - a novel tool for complex optimisation problems”. In: *Proc. of IPROMS 2006 Conference*, pp. 454–461. 2006

- [PIN 02] Pintér J.D., “Global optimization: software, test problems, and applications”, in Pardalos Panos M. et Romeijn H. Edwin (Eds.) “Handbook Of Global Optimization”, Vol 2, pp 515-569, 2002.
- [PRA 12] Prasartvit T., Kaewkamnerpong B., Achalakul T. “Dimensional Reduction Based on Artificial Bee Colony for Classification Problems”, in D.-S. Huang et al. (Eds.): Bio-Inspired Computing and Applications: Proc. of 7th International Conference on Intelligent Computing, Zhengzhou, China, August 11-14. 2011, Vol. 6840, pp. 168–175, 2012.
- [PRE 09] Premalatha K., Natarajan A.M. “Hybrid PSO and GA for global maximization”. International Journal of Open Problems in Computer Science and Mathematics, Vol. 2, No 4, pp. 597-608. 2009.
- [RAO 11] Rao R.V., Savsani V.J., Vakharia D.P., “Teaching–learningbased optimization: a novel method for constrained mechanical design optimization problems”. Computer-Aided Design, Vol. 43, issue 3, pp 303–315. 2011.
- [RAO 14] Rao R. V., Waghmare G.G. “A comparative study of a teaching–learning-based optimization algorithm on multi-objective unconstrained and constrained functions”. Journal of King Saud University – Computer and Information Sciences. Vol. 26, issue 3, pp 332–346. 2014.
- [RAS 09] Rashedi E., Nezamabadi-pour H., Saryazdi S. G., “A gravitational search algorithm”. Information Sciences, Vol. 179, issue 13, pp 2232-2248, 2009.
- [REC 73] Rechenberg I. “Evolution strategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution”. Friedrich Fromann Verlag, Stuttgart, 1973. cité dans [SIM 13a]
- [ROD 14] Rodrigues D., Pereira L.A.M., Nakamura R. Y.M., Costa K. A.P., Yang X-S, Souza A. N., Papa J. P. “A wrapper approach for feature selection based on Bat Algorithm and Optimum-Path Forest”. Expert Systems with Applications, Vol. 41, Issue 5, pp. 2250-2258. 2014.
- [SAE 07] Saeys Y., Inza I., Larrañaga P., “A review of feature selection techniques in bioinformatics”, Bioinformatics, Vol. 23, issue 19, pp 2507–2517, 2007.

- [SER 06] Seredyński F., “Evolutionary paradigms”, in Zomaya A.Y. (Eds.): Handbook of nature-inspired and innovative computing, Springer US, pp 111-145, 2006.
- [SHE 12] Sheikhan M., Mohammadi N., “Neural-based electricity load forecasting using hybrid of GA and ACO for feature selection”. Neural Comput Appl., Vol. 21, Issue 8, pp 1961-1970, 2012
- [SHE 13] Sheikhan M., Mohammadi N. “Time series prediction using PSO-optimized neural network and hybrid feature selection algorithm for IEEE load data”. Neural Comput & Applic Vol. 23, Issue 3-4, pp 1185-1194, 2013.
- [SHI 99] Shi Y., Eberhart R., “Empirical study of particle swarm optimization”, In Proc. of the 1999 Congress on Evolutionary Computation, Washington, District of Columbia, pp 1945- 1950. 6-9 Juillet 1999.
- [SHI 05] Shi X.H, Lianga Y.C, Leeb H.P, LubC., Wang, L.M., “An improved GA and a novel PSO-GA-based hybrid algorithm”. Information Processing Letters Vol. 93, issue 5, pp 255–261. 2005
- [SHI 08] Shi C., Bu Y., Liu J., “Mobile robot path planning in three-dimensional environment based on ACO-PSO hybrid algorithm”, In Proc. of IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 252-256, Xian, Chine, 2-5 Juillet 2008
- [SHR 11] Shrme A. E., “Hybrid intelligent technique for automatic communication signals recognition using Bees Algorithm and MLP neural networks based on the efficient features”. Expert Systems with Applications, Vol. 38, Issue 5, pp 6000-6006, 2011.
- [SHU 11] Shuang B., Chen J., Li Z. “Study on hybrid PS-ACO algorithm”. Applied Intelligence, Vol. 34, No 1., pp. 64–73. 2011.
- [SIL 08] Silva Jr E.R. et al., “Feature Selection and Model Design through GA Applied to Handwritten Digit Recognition from Historical Document Images”, In proc. of International Conference on Frontiers in Handwritten Recognition, Canada, 2008

- [SIM 08] Simon D., “Biogeography-Based Optimization”. IEEE Transactions On Evolutionary Computation, Vol. 12, No 6, December 2008.
- [SIM 13a] Simon D., “Evolutionary optimization algorithms : Biologically-Inspired and Population-Based Approaches to Computer Intelligence”, John Wiley & Sons, Inc., Hoboken, New Jersey, 2013
- [SIM 13b] Simon D., Shah A., Scheidegger C., “Distributed Learning with Biogeography-Based Optimization: Markov Modeling and Robot Control”, Swarm and Evolutionary Computation, Vol. 10, pp. 12–24, 2013.
- [SIN 02] Singh S., Singh M., Markou M. “Feature Selection for Face Recognition based on Data Partitioning”. In Proc. of the 16th International Conference on Pattern Recognition (ICPR 02), Vol. 1, pp. 680–683, 2002.
- [SIV 07] Sivagaminathan R.K., Ramakrishnan S. “A hybrid approach for feature subset selection using neural networks and ant colony optimization”. Expert Systems with Applications, Vol. 33, issue 1, pp. 49–60. 2007
- [SOM 04] Somol P., Novovicova J., Pudil P., “Advances In Feature Selection Methodology: An Overview of Recent Utia Results”, Kybernetika, Vol. 4, No 1, 2004.
- [STU 00] Stützle T., Hoos H.H., “MAX–MIN ant system”, Future Generation Computer Systems, Vol. 16, No 8, pp. 889–914. 2000.
- [SWA 09] Swagatam Das, Ajith Abraham and Amit Konar, “Metaheuristic Pattern Clustering – An Overview”, In: Metaheuristic Clustering”, Studies in Computational Intelligence, Vol. 178, pp.1-53, 2009
- [TEO 05] Teodorovic D., Dell’Orco M. “Bee colony optimization - a cooperative learning approach to complex transportation problems”. In: Advanced OR and AI Methods in Transportation. Proc. the 10th EWGT Meeting, pp. 51–60. Poznan, Pologne, 13-16 sep 2005.
- [TEO 09] Teodorovic D. “Bee Colony Optimization (BCO)”, In: Chee P. L., Lakhmi C. J., et Satchidananda D. (Eds.) Innovations in Swarm Intelligence, Studies in Computational Intelligence, Vol. 248, pp. 39-60, 2009.

- [UNL 10] Unler A., Murat A., "A discrete particle swarm optimization method for feature selection in binary classification problems", European Journal of Operational Research, Vol. 206, issue 3, pp 528–539, 2010.
- [VAL 08] Valdez F., Melin P., Castillo O. "A New Evolutionary Method Combining Particle Swarm Optimization and Genetic Algorithms Using Fuzzy Logic". Castillo, O. et al. (Eds.): Soft Computing for Hybrid Intel. Systems, Vol. 154. pp. 347–361, 2008.
- [VAN 14] Vanitha S., Dr. T.Padma, "A Survey on Swarm Intelligence Algorithms". In International Journal of Computer Science and Mobile Computing, Vol. 3 Issue.5, pp. 994-998, May- 2014.
- [WAN 05] Wang, X., Yang, J., Peng, N. and Teng, X. "Finding Minimal Rough Set Reducts with Particle Swarm Optimization". Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, LNCS Vol. 3641, pp 451-460, 2005.
- [WAN 07] Wang X., Yang J., Teng X., Xia, W., et Jensen R. "Feature selection based on rough sets and particle swarm optimization". Pattern Recognition Letters, Vol. 28, issue 4, pp 459–471, 2007.
- [XIA 04] H. Xiaohui, Y. Shi, R. Eberhart "Recent advances in particle swarm", In Proc. of the 2004 Congress on Evolutionary Computation, Vol. 1, 19-23 june 2004.
- [XIA 05] Xiangyang Wang, Jie Yang, Ningsong Peng, and Xiaolong Teng, "Finding Minimal Rough Set Reducts with Particle Swarm Optimization", Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing: proc. of the 10th International Conference, RSFDGrC 2005, Vol. 3641, pp. 451–460, Regina, Canada, 31 Aug- 3 Sep, 2005
- [XU 88] Xu L., Yan P., Chang T., "Best first strategy for feature selection". In: Proc. of 9th International Conference on Pattern Recognition, Vol. 2, pp. 706–708, Rome, Italie, 14-17 nov 1988.
- [YAN 08] Yang, C.S., Chuang, L-Y., Ho, C-H. and Yang, C-H. "Microarray Data Feature Selection Using Hybrid GA-IBPSO". In Oscar Castillo et al. (eds.), Trends in Intelligent Systems and Computer Engineering, Springer, pp. 243-253. 2008

- [YAN 10] Yang, X. S., “A New Metaheuristic Bat-Inspired Algorithm”, in: Cruz, C.; González, J. R., Pelta, D. A., Terrazas, G. (Eds.): Nature Inspired Cooperative Strategies for Optimization (NISCO 2010), Studies in Computational Intelligence Vol. 284, Springer Berlin, pp. 65–74. 2010.
- [YAN 11] Yang, X. S., Bat algorithm for multi-objective optimisation, Int. J. Bio-Inspired Computation, Vol. 3, No. 5, pp. 267–274. 2011
- [YAN 13] Yang Xin-She, “Bat algorithm: literature review and applications”, Int. J. Bio-Inspired Computation, Vol. 5, No 3, pp. 141–149. 2013.
- [YE 07] Ye D., Chen Z., Liao J., “A New Algorithm for Minimum Attribute Reduction Based on Binary Particle Swarm Optimization with Vaccination”. In: Advances in Knowledge Discovery and Data Mining, proc. of the 11th Pacific-Asia Conference, PAKDD 2007, Vol. 4426, pp. 1029-1036, Nanjing, China, Mai 22-25, 2007
- [ZAI 07] Zainal, A., Maarof, M. and Shamsuddin, S., “Feature Selection using Rough-DPSO in Anomaly Intrusion Detection”, In proc. of International Conference on Computational Science and Its Applications, ICCSA 2007, LNCS, Vol. 4705, pp. 512–524, Kuala Lumpur, Malaysia, 26-29 août 2007
- [ZHA 03] Zhang P, Verma B, Kumar K., “Neural vs. Statistical classifier in conjunction with genetic algorithm feature selection in digital mammography”. In Proc. Congress on Evolutionary Computation (CEC-2003), Vol 2, pp 1206 – 1213, 8-12 déc 2003.

ANNEXE 1

DESCRIPTION DES BASEES DE DONNEES

Les bases de données utilisées dans les expérimentations effectuées ont été extraites de l'UCI Machine Learning Repository (<http://archive.ics.uci.edu/ml/datasets>)

1. La base de données *SpamBase*

La collection SpamBase contient des informations relatives à 4601 messages, avec 1813 (39,4%) courriers indésirables ou spams. Cette collection a été prétraitée, et les textes des messages ne sont pas disponibles (pour éviter les problèmes de violation de la vie privée).

Nom

Spambase Data Set

Source

Créateur :

Mark Hopkins, Erik Reeber, George Forman, Jaap Suermondt. Hewlett-Packard Labs,
1501 Page Mill Rd., Palo Alto, CA 94304

Donneur :

George Forman (gforman at nospam hpl.hp.com) 650-857-7835

Informations sur les caractéristiques

Numéro de la caractéristique	Nom de la caractéristique	Description de la caractéristique
1	word_freq_make.	Caractéristiques de type word_freq_WORD = pourcentage des mots dans l'email qui sont égaux à WORD, i.e. $100 * (\text{nombre d'occurrence du mot WORD dans l'email}) / \text{nombre total des mots dans l'email}$. Un mot dans ce cas est une chaîne de caractères alphanumériques délimitée par des caractères non alphanumériques ou terminée par une fin de chaîne.
2	word_freq_address.	
3	word_freq_all.	
4	word_freq_3d.	
5	word_freq_our.	
6	word_freq_over.	
7	word_freq_remove.	
8	word_freq_internet.	
9	word_freq_order.	
10	word_freq_mail.	
11	word_freq_receive.	
12	word_freq_will.	
13	word_freq_people.	
14	word_freq_report.	
15	word_freq_addresses.	
16	word_freq_free.	
17	word_freq_business.	
18	word_freq_email.	
19	word_freq_you.	
20	word_freq_credit.	
21	word_freq_your.	
22	word_freq_font.	
23	word_freq_000.	
24	word_freq_money.	
25	word_freq_hp.	
26	word_freq_hpl.	
27	word_freq_george.	
28	word_freq_650.	
29	word_freq_lab.	
30	word_freq_labs.	
31	word_freq_telnet.	
32	word_freq_857.	
33	word_freq_data.	
34	word_freq_415.	
35	word_freq_85.	
36	word_freq_technology.	
37	word_freq_1999.	

Numéro de la caractéristique	Nom de la caractéristique	Description de la caractéristique
38	word_freq_parts.	Caractéristiques de type word_freq_WORD = pourcentage des mots dans l'email qui sont égaux à WORD.
39	word_freq_pm.	
40	word_freq_direct.	
41	word_freq_cs.	
42	word_freq_meeting.	
43	word_freq_original.	
44	word_freq_project.	
45	word_freq_re.	
46	word_freq_edu.	
47	word_freq_table:	
48	word_freq_conference:	
49	char_freq_;	
50	char_freq_(:	
51	char_freq_[:	
52	char_freq_!:	
53	char_freq_\$.	
54	char_freq_#.	
55	capital_run_length_average.	Moyenne des longueurs des séquences ininterrompues de lettres majuscules.
56	capital_run_length_longest.	Longueur de la séquence ininterrompue la plus longue de lettres majuscules.
57	capital_run_length_total.	Somme des longueurs des séquences ininterrompues de lettres majuscules = nombre total de lettres majuscules dans l'e-mail

2. La base de données Breast Cancer

Dans cette base, les échantillons arrivent périodiquement comme rapporte le Dr Wolberg dans ses cas cliniques. La base de données reflète donc ce regroupement chronologique des données.

Nom

Wisconsin Breast Cancer Database

Sources

Créateur:

Dr. William H. Wolberg (physicien), University of Wisconsin Hospitals, Madison, Wisconsin, USA

Donneur:

Olvi Mangasarian

Informations sur les caractéristiques

Numéro de la caractéristique	Nom de la caractéristique	Les valeurs de caractéristiques
1	Numéro de code de l'échantillon	Numéro d'Id
2	Épaisseur de la masse (Clump Thickness)	1 – 10
3	Uniformité de la taille de la cellule	1 – 10
4	Uniformité de la forme de la cellule	1 – 10
5	Adhésion marginale	1 – 10
6	Taille de la cellule unique épithéliale	1 – 10
7	Noyaux nus	1 – 10
8	Chromatine fade (Bland Chromatin)	1 – 10
9	Nucléoles normales	1 – 10
10	Mitoses	1 – 10

3. La base de données German

Dans cette base de données, deux ensembles de données sont fournis :

- L'ensemble de données d'origine, sous la forme définie par Pr Hofmann, contient des caractéristiques catégoriques / symboliques.
- Pour les algorithmes qui ont besoin de caractéristiques numériques, l'Université de Strathclyde a produit une autre base de données, qui a été éditée. Plusieurs caractéristiques indicatrices ont été ajoutées pour la rendre appropriée pour les

algorithmes qui ne peuvent manipuler des caractéristiques catégoriques. Plusieurs caractéristiques qui sont catégoriques (comme la caractéristique 17) ont été codées comme entiers.

Nom

German Credit data

Information sur la Source

Professor Dr. Hans Hofmann . Institut f"ur Statistik und "Okonometrie .Universit"at Hamburg . FB Wirtschaftswissenschaften .Von-Melle-Park 5 .2000 Hamburg 13

Informations sur les caractéristiques

Numéro de la caractéristique	Nom de la caractéristique	Description de la caractéristique
1	Statut de compte existant actuellement	A11 : ... < 0 DM A12 : 0 <= ... < 200 DM A13 : ... >= 200 DM / affectations de salaire pour au moins une année. A14 : pas de vérification de compte
2	La durée en mois	/
3	Historique du crédit	A30: pas de crédits pris / tous les crédits remboursés régulièrement A31: l'ensemble des crédits régulièrement remboursé auprès de cette banque A32: crédits existants remboursés régulièrement jusqu'à maintenant A33: retard dans le remboursement dans le passé A34: compte rendu critique / autres crédits existants (pas dans cette banque)
4	Le but	A40: voiture (nouvelle) A41: voiture (utilisée) A42: mobilier / équipement A43: radio / télévision A44: appareils électroménagers A45: les réparations A46: l'éducation A47: (Vacances - n'existe pas?) A48: recyclage A49: business A410: autres

Numéro de la caractéristique	Nom de la caractéristique	Description de la caractéristique
5	Montant du crédit	/
6	Compte d'épargne / obligations	A61: ... <100 DM A62: 100 <= ... <500 DM A63: 500 <= ... <1000 DM A64: ... >= 1000 DM A65: inconnu / pas de compte d'épargne
7	Emploi actuel depuis :	A71: chômeurs A72: ... <1 années A73: 1 <= ... <4 ans A74: 4 <= ... <7 ans A75: ... >= 7 ans
8	Taux de versement en pourcentage du revenu disponible	/
9	La situation personnelle et le sexe	A91: masculin: Divorcé / séparé A92: féminin: Divorcée / séparée / mariée A93: masculin : célibataire A94: masculin: mari / veuf A95: féminin: célibataire
10	Autres débiteurs / garants	A101 : aucun A102 : membre du groupe A103 : garant
11	Résidence actuelle depuis	/
12	propriété	A121: immobilier A122: sinon A121: contrat d'épargne de la société dans le bâtiment / l'assurance vie A123: sinon A121 / A122: voiture ou autre, pas dans la caractéristique 6 A124: inconnu / pas de propriété
13	L'âge en années	/
14	Autres plans de tranches	A141: banque A142: magasins A143: aucun
15	Logement	A151: location A152: propriété A153: gratuit
16	Nombre de crédits existants auprès de cette banque	/

Numéro de la caractéristique	Nom de la caractéristique	Description de la caractéristique
17	emploi	A171: chômeur /non qualifié - non-résident A172: non qualifié - résident A173: employé qualifié / fonctionnaire A174: gestion / indépendant / employé / employé hautement qualifié/ fonctionnaire
18	Nombre de personnes susceptibles d'assurer l'entretien de	
19	téléphone	A191: aucun A192: oui, enregistré sous le nom de client
20	travailleur étranger	A201 : oui A202 : non

4. La base de données Hepatitis

Les échantillons de cette base contiennent des cas d'hépatite et les classent comme mortels ou non en se basant sur 19 caractéristiques.

Nom

Hepatitis Domain

Sources

Créateur:

Inconnu

Donneur:

G.Gong (Carnegie-Mellon University) via. Bojan Cestnik, Jozef Stefan Institute.
Jamova 39. 61000 Ljubljana. Yugoslavia

Informations sur les caractéristiques

Numéro de la caractéristique	Nom de la caractéristique	Valeur de la caractéristique
1	Âge	10, 20, 30, 40, 50, 60, 70, 80
2	Sexe	Masculin, Féminin
3	Stéroïde	Non, Oui
4	antiviraux	Non, Oui
5	Fatigue	Non, Oui
6	Malaise	Non, Oui
7	Anorexie	Non, Oui
8	Cirrhose	Non, Oui
9	foie dur	Non, Oui
10	rate palpable	Non, Oui
11	spiders	Non, Oui
12	Ascite	Non, Oui
13	Varices	Non, Oui
14	Bilirubine	0.39, 0.80, 1.20, 2.00, 3.00, 4.00
15	phosphatase Alcaline	33, 80, 120, 160, 200, 250
16	SGOT	13, 100, 200, 300, 400, 500,
17	Albumine	2.1, 3.0, 3.8, 4.5, 5.0, 6.0
18	Protime	10, 20, 30, 40, 50, 60, 70, 80, 90
19	Histologie	Non, Oui

5. La base de données Liver

Cette base de données contient des tests sanguins qui sont sensés être sensibles à des troubles du foie qui pourraient résulter de la consommation excessive d'alcool.

Nom

BUPA liver disorders

Source information

Créateurs :

BUPA Medical Research Ltd.

Donneur :

Richard S. Forsyth. 8 Grosvenor Avenue, Mapperley Park, Nottingham NG3 5DX

Informations sur les caractéristiques

Les 5 premières caractéristiques correspondent toutes à des tests sanguins. Chaque ligne dans le fichier bupa.data constitue le dossier d'un individu de sexe masculin.

Il semble que les boissons > 5 est une sorte de sélection sur cette base de données.

Numéro de la caractéristique	Nom de la caractéristique	Explication
1	mcv	mean corpuscular volume : volume globulaire moyen
2	alkphos	alkaline phosphatase : phosphatase alcaline
3	sgpt	alamine aminotransferase : alamine aminotransférase
4	sgot	aspartate aminotransférase : aspartate aminotransferase
5	gammagt	gamma-glutamyl transpeptidase : gamma-glutamyl transpeptidase
6	boissons	nombre d'équivalents à une demi chopine (0.25 litre) de boissons alcoolisées bues par jour
7	selecteur	domaine utilisé pour diviser les données en deux ensembles

6. La base de données Musk

Cette base de données décrit un ensemble de 92 molécules dont 47 sont jugées par des experts humains à être musc (matière naturelle entrant dans la composition des parfums) et les 45 molécules restantes sont jugés non-musc. Le but est d'apprendre à prédire si des nouvelles molécules seront muscs ou non-muscs. Toutefois, les 166 caractéristiques qui décrivent ces molécules dépendent de la forme exacte ou la configuration de la molécule. Une seule molécule peut adopter de nombreuses formes différentes. Pour produire cet ensemble de données, les configurations de faible énergie des molécules

ont été générées et ensuite filtrées pour éliminer les configurations très similaires. Cela a laissé 476 conformations ou configurations et un vecteur de caractéristique a été extrait pour décrire chacune.

Title

MUSK "Clean1" database

Sources

Créateurs:

AI Group at Arris Pharmaceutical Corporation

contact: David Chapman or Ajay Jain. Arris Pharmaceutical Corporation. 385 Oyster Point Blvd. South San Francisco, CA 94080. 415-737-8600. zvona@arris.com, jain@arris.com

Donneur:

Tom Dietterich, Department of Computer Science, Oregon State University, Corvallis, OR 97331. 503-737-5559.tgd@cs.orst.edu

Informations sur les caractéristiques

Caractéristique	Description
Nom de la molécule	Nom symbolique de chaque molécule. Les muscs ont des noms tels que MUSk-188. Les non-muscs ont des noms tels que NON MUSk-JP13.
Nom de la conformation	Nom symbolique de chaque conformation. Elles ont le format MOL_ISO + CONF. Où : MOL est le nombre de molécule, ISO est le nombre de stéréo (généralement 1), et CONF est le nombre de conformation.
f1 jusqu'à f162	Ce sont des 'caractéristiques de distance' sur le long des rayons. Les distances sont mesurées en centièmes d'angstroms. Les distances peuvent être négatives ou positives, car elles sont effectivement mesurées par rapport à une origine placée le long de chaque rayon. L'origine a été définie par une surface "consensus de musc" qui n'est plus utilisée.
f163	Il s'agit de la distance de l'atome d'oxygène dans la molécule à un point désigné dans trois espaces. Il est également appelé OXY-DIS.
f164	OXY-X: X-déplacement à partir du point désigné.
f165	OXY-Y: Y-déplacement à partir du point désigné
f166	OXY-Z: Z-déplacement à partir du point désigné

ANNEXE 2

LES MESURES UTILISEES POUR LE FEATURE RANKING DANS L'APPROCHE CLASSIQUE PROPOSEE

1. Fréquences d'apparition des termes dans la collection

Pour la fréquence d'apparition des termes dans la collection ou Document Frequency (DF), l'idée est que les mots qui apparaissent dans très peu de documents de la collection (par rapport à un seuil fixé au préalable) ne sont pas informatifs. On peut donc les supprimer du vecteur de caractéristiques et avoir un vecteur réduit. Cette méthode est basée sur le choix du seuil et ne prend pas en compte les classes.

DF est le nombre de documents dans lesquels un terme apparaît. On calcule le DF pour chaque terme unique dans le corpus d'apprentissage et on enlève de l'espace de caractéristiques les termes dont le DF est inférieur à un seuil prédéterminé. L'hypothèse de base est que les termes rares sont soit non informatifs pour la prédiction de catégorie, ou non influents sur la performance globale. Dans les deux cas, la suppression des termes rares réduit la dimensionnalité de l'espace de représentation. L'amélioration de la précision de la catégorisation est également possible si les termes rares sont des termes de bruit.

2. L'information mutuelle

L'information mutuelle est l'un des critères les plus utilisés en statistiques. Le critère de l'information mutuelle est calculé pour une caractéristique w et une classe c de la manière suivante :

$$IM(w, c) = \log \frac{P(w, c)}{P(w) * P(c)}$$

Avec :

- $P(w, c)$: proportion de documents dans la collection appartenant à la classe c et où la caractéristique w est présente
- $P(w)$: DF de la caractéristique w
- $P(c)$: proportion de documents de classe c

Dans notre travail, nous avons calculé la valeur de l'information mutuelle de la classe spam et la classe non spam pour chacune des caractéristiques. Et la valeur globale pour chaque caractéristique est calculée par la formule suivante ($Y = \{\text{spam, légitime}\}$):

$$IM(w) = \sum_{c \in Y} P(c)IM(w, c)$$

3. Le gain d'information

Le gain de l'information mesure la diminution de l'entropie, ou le quantité de connaissance gagnée si une caractéristique est présente ou absente. Le gain de l'information (IG) pour un caractéristique w est défini avec :

$$IG(w) = \sum_{X \subseteq \{w, \bar{w}\}} \sum_{Y \subseteq \{c_i\}} P(X, Y) \log \left(\frac{P(X, Y)}{P(X) * P(Y)} \right)$$

Avec :

- $P(X, Y)$: proportion de documents dans la collection appartenant à la classe Y et où la caractéristique X est présente.
- $P(X)$: DF de la caractéristique X .
- $P(Y)$: proportion de documents de classe Y dans le corpus.

Où $\{c_i\}_{i=1}^m$ présente l'ensemble de classes, dans notre cas $\{\text{spam, légitime}\}$.

4. Méthode du Chi-2 (χ^2)

La statistique du χ^2 mesure l'écart à l'indépendance entre une caractéristique (mot) w_k (présente ou absente) et une classe c_i (présente ou absente) ;

Le calcul de Chi2 nécessite de construire le tableau de contingence (2×2) pour chaque caractéristique w du corpus et pour chaque classe ci (voir tableau ci dessous). Dans notre cas, nous avons seulement 2 classes, donc il suffit de calculer le tableau pour une seule classe.

	caractéristique w présente	caractéristique w absente	
classe spam présente	a	c	$a + c$
classe spam absente	b	d	$b + d$
	$a + b$	$c + d$	$N = a + b + c + d$

Tableau de contingences pour l'absence ou la présence d'un descripteur dans la classe spam.

La statistique du χ^2 peut se mettre sous la forme :

$$\chi^2_{uni}(w, \text{spam}) = \frac{N(ad - cb)^2}{(a + c)(b + d)(a + b)(c + d)}$$

Cette formule est calculée pour tous les couples (w, ci) . Dans notre cas :

$$\chi^2(w, \text{spam}) = \chi^2(w, \text{légitime})$$

Donc il suffit de calculer la valeur de Chi2 pour la classe spam.

Généralement, la valeur discriminante globale de chaque caractéristique (w) est calculée par la mesure:

$$\chi^2_{\max} = \max_i \chi^2_{uni}(w, c_i)$$

Comme les valeurs de chi2 pour les deux classes sont égales, alors la valeur discriminante globale pour une caractéristique (w) est la valeur de cette caractéristique pour la classe spam.

5. $S2n$

$S2n$, Signal-to-noise ratio coefficient est une méthode basée sur la corrélation entre les caractéristiques. Cette méthode classe les caractéristiques par le rapport de la valeur absolue de la différence des moyennes des classes sur la moyenne des écart-types des classes. Ce critère est similaire au critère de Fisher et au coefficient de corrélation de Pearson.

La formule de $s2n$ pour une caractéristique w est calculée avec:

$$S2n = \frac{|M_w \text{spam} - M_w \text{legitime}|}{\sigma_w}$$

Avec :

- $M_w \text{spam}$ est la moyenne des valeurs des exemples de la classe spams pour la caractéristique w.
- $M_w \text{legitime}$ est la moyenne des valeurs des exemples de la classe légitime pour la caractéristique w.
- $\sigma_w = 0.5 * (\sigma_{\text{spam}} + \sigma_{\text{legitime}})$ avec:
 - σ_{spam} est l'écart-type des valeurs des exemples de la classe spams pour la caractéristique w.
 - σ_{legitime} est l'écart-type des valeurs des exemples de la classe légitime pour la caractéristique w.

6. Relief

La méthode Relief est une méthode qui se base sur l'évaluation individuelle de chacune des caractéristiques, elle a déjà été présentée dans le chapitre 2 (voir section 2.5.2.3).

A PROPOS DE L'AUTEUR

BIOGRAPHIE DE L'AUTEUR

Kamilia MENGHOUR, est née en 1986 à Annaba, où elle a poursuivi ses études jusqu'à l'obtention du baccalauréat (option sciences de la nature et de la vie) en 2004. Elle rejoint l'université d'Annaba la même année pour suivre une formation de Licence Académique en Informatique. Elle obtient sa Licence en 2007, puis accède à une formation de Master en Sciences et Technologies de l'Information et de la Communication (STIC) qu'elle termine en Octobre 2009. A partir de janvier 2010, Melle Menghour est inscrite à une formation doctorale en Informatique affiliée au laboratoire LabGED au département d'informatique de l'université d'Annaba. Depuis 2011, elle enseigne en qualité d'Assistante Associée au département d'Économie de la Faculté de Sciences Économiques et Sciences de Gestion de l'Université d'Annaba.

Dans le cadre de ses activités scientifiques dans le domaine de la classification et des systèmes bio-inspirés, Kamilia MENGHOUR a participé à plusieurs conférences internationales et a publié un article dans un journal international en 2014, elle a aussi une deuxième publication soumise en cours d'évaluation après révision. Depuis 2013, Melle Menghour est membre du Laboratoire d'Ingénierie des Systèmes Complexes (LISCO) de l'université d'Annaba.

CONTRIBUTIONS SCIENTIFIQUES

1. Publications dans des journaux internationaux

- 1 Menghour K., Souici-Meslati L., “Classical and swarm based approaches for feature selection in spam filtering”, International Journal of Advanced Intelligence Paradigms, IJAIP, InderScience Publishers, Vol. 6, No. 3, pp. 214-234, 2014.

Informations sur le journal IJAIP

ISSN:1755-0386 (print) .1755-0394 (online)

Fréquence de publication:4 numéros par an (depuis 2008).

Processus de reviewing: peer reviewed.

Publié par: InderScience Publishers

Pays: United Kingdom

Site Web: <http://www.inderscience.com/jhome.php?jcode=ijaip>

Indexé par :SCOPUS (Elsevier), Academic OneFile (Gale), ACM Digital Library, DBLP Computer Science Bibliography, Expanded Academic ASAP (Gale), Google Scholar, Inspec (Institution of Engineering and Technology), Scirus (Elsevier)

H Index: 4 (<http://www.scimagojr.com/>, 2008-2013)

- 2 Menghour K., Souici-Meslati L., “Hybrid ACO-PSO approaches for feature selection”, **Soumis pour publication dans un journal international, en cours de reviewing après révision.**

2. Communications avec comité de lecture international

1. Menghour K., Souici-Meslati L., “Sélection de Caractéristiques pour le Filtrage de Spams”, **CORIA’2010**, Conférence sur la Recherche d’Information et ses Applications, pp. 349-360, Sousse, Tunisie, 18-20 Mars 2010, pp. 349-360, **ISBN : 978-9973-37-581-0**, indexée **DBLP** et **SCOPUS**
2. Menghour K., Souici-Meslati L., “Une approche de sélection d’attributs pour la reconnaissance de chiffres manuscrits”, **EGC-M’2010**, Conférence Internationale Maghrébine sur l’Extraction et la Gestion de Connaissances, Alger, Algérie, 13-14 Décembre 2010.
3. Menghour K., Souici-Meslati L., “Sélection de caractéristiques basée sur les colonies de fourmis pour le filtrage de spams”, **EGC-M’2011**, Conférence Internationale Maghrébine sur l’Extraction et la Gestion de Connaissances, pp. 168-179, Tanger, Maroc, 23-25 Novembre, 2011.
4. Menghour K., Souici-Meslati L., “Ant colony optimization for feature subset selection: An application to spam filtering”, **CISC’2011**, Conférence Internationale sur les Systèmes Complexes, Jijel, Algérie, 6-8 Décembre, 2011.

Résumé

Le travail présenté dans ce manuscrit se situe dans le domaine de la classification et, plus précisément, celui de la sélection d'attributs ou de caractéristiques qui se rapporte au problème du choix des attributs les plus informatifs pour la réalisation d'une tâche de classification. L'objectif principal de cette sélection est de réduire le nombre de caractéristiques, en éliminant les attributs redondants et non pertinents, tout en essayant de maintenir ou d'améliorer la performance du classifieur. Dans le cadre de notre travail, nous nous sommes intéressés aux méthodes bio-inspirées évolutionnaires, en particulier, celles issues de l'intelligence en essaims (swarm intelligence) pour la sélection de caractéristiques.

Après une étude approfondie des méthodes classiques et bio-inspirées de sélection d'attributs, nous avons commencé par la proposition d'une approche basée sur des méthodes classiques d'ordonnancement de caractéristiques puis deux approches de sélection d'attributs issues de la swarm intelligence, l'une est basée sur les variantes de l'algorithme d'optimisation par les colonies de fourmis (ACO) et l'autre est basée sur l'algorithme d'optimisation par les essaims particulaires (PSO).

Suite à l'étude des travaux relatifs aux méthodes bio-inspirées hybrides, nous avons constaté le manque de propositions d'hybridation de méthodes bio-inspirées pour la sélection d'attributs. Dans ce cadre, nous proposons et comparons trois approches d'hybridation d'algorithmes d'optimisation par les colonies de fourmis et les essaims particulaires pour la sélection d'attributs.

Les expérimentations ont été effectuées en utilisant six bases de données publiques, et leurs résultats ont montré, qu'avec toutes les approches proposées et évaluées, on peut réduire le nombre de caractéristiques et améliorer les taux de classification correcte.

Mots-clés : Sélection d'attributs ou de caractéristiques, approches bio-inspirées, intelligence en essaims, optimisation par les colonies de fourmis (ACO), optimisation par les essaims particulaires (PSO).

Abstract

The work presented in this manuscript can be placed within the classification field and, more specifically, attribute or feature selection which relates to the problem of selecting the most informative attributes for achieving a classification task. The main objective of this selection is to reduce the features number, by eliminating redundant and irrelevant attributes, while trying to maintain or improve the classifier performance. In our work, we are interested in the bio-inspired evolutionary methods, in particular those of swarm intelligence for feature selection.

After a detailed study of classical and bio-inspired attribute selection methods, we have started by the proposition of an approach based on classical feature ranking methods, then two swarm intelligence feature selection approaches. The first one is based on variants of ant colony optimization (ACO) algorithm, and the other based on particle swarm optimization algorithm (PSO).

After study on the bio-inspired hybrid methods, we found the lack of proposals in hybridization of bio-inspired attribute selection methods. In this context, we propose and compare three hybridization approaches of ant colonies and particle swarm optimization algorithms for attribute selection.

The experiments were performed using six public databases, and the results showed that, with all the proposed and evaluated approaches, we can reduce the features number and improve the correct classification rate.

Keywords: Attribute or feature selection, bio-inspired approaches, swarm intelligence, ant colony optimization (ACO), particle swarm optimization (PSO).

الملخص

إن العمل المقدم في هذا المخطوط يندرج في إطار مجال التصنيف، وبصفة أدق اختيار أو تحديد الخصائص أو الميزات، الذي يتعلق بمشكلة اختيار الخصائص التي تعطي أكبر قدر من المعلومات لتحقيق مهمة التصنيف. إن الهدف الرئيسي من هذا الاختيار هو تقليل عدد الخصائص، من خلال إزالة الخصائص الزائدة عن الحاجة و الغير ذات صلة، محاولين بذلك الحفاظ على مردود المصنف أو تحسينه. في إطار عملنا، ينصب اهتمامنا نحو الطرق التطورية المستوحة من البيولوجيا، وأسماها المستوحة من ذكاء الأسراب من أجل تحديد الخصائص. بعد الدراسة المفصلة للطرق الكلاسيكية والمستوحة من البيولوجيا من أجل تحديد الخصائص، بدأنا أولاً باقتراح منهجية قائمة على الطريقة الكلاسيكية التي ترتكز على ترتيب الخصائص، ثم اقتربنا منهاجيتين مستوحيتين من ذكاء الأسراب، حيث أن الأولى ترتكز على متغيرات خوارزمية استمثال مستعمرات النمل (ACO)، والأخرى على خوارزمية استمثال سرب الجسيمات (PSO). استناداً إلى دراسة الأعمال ذات الصلة بالطرق الهجينية المستوحة من البيولوجيا، استنتجنا عدم وجود مقتراحات تهجين بين طرق تحديد الخصائص المستوحة من البيولوجيا. وفي هذا السياق، قمنا باقتراح و مقارنة ثلاثة منهاجيات تهجين بين خوارزميات استمثال مستعمرات النمل و سرب الجسيمات من أجل تحديد الخصائص.

أجريت التجارب باستخدام ستة قواعد بيانات عامة، وأظهرت النتائج أنه مع كل منهاجيات المقترحة والتي تم تقييمها، يمكننا تقليل عدد الخصائص وتحسين معدل التصنيف الصحيح.

الكلمات المفتاحية : تحديد الخصائص، الطرق المستوحة من البيولوجيا، ذكاء الأسراب ، استمثال مستعمرات النمل (ACO)، استمثال سرب الجسيمات (PSO).