

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud MAMMARI, Tizi-Ouzou



Faculté de Génie Electrique et d'Informatique
Département d'Automatique

MEMOIRE DE FIN D'ETUDES

En vue de l'obtention du diplôme

D'INGENIEUR D'ETAT EN AUTOMATIQUE

Thème

*Classification automatique
par essais de particules*

Proposé par : DIAF Moussa

Présenté par :

AIT MMAMAR Rebiha
BERRICHI Souad
GOUDJIL Lynda

Soutenu le : 12 /10 /2009

Promotion 2009

Avant Propos

Le travail que nous présentons dans ce mémoire a été effectué en vue de l'obtention du diplôme d'ingénieur en Automatique au sein du département Automatique, Faculté de Génie Electrique et d'Informatique, Université Mouloud Mammeri, Tizi-Ouzou.

Nous exprimons nos remerciements et notre gratitude à notre promoteur, Monsieur **Moussa DIAF**, Enseignant au département Automatique, Université Mouloud Mammeri, Tizi-Ouzou, pour l'aide et le soutien qu'il nous a apportés tout au long de ce travail.

Nos plus vifs remerciements vont aussi à Monsieur **DIRAMI Ahmed**, Enseignant au département Automatique, Université Mouloud Mammeri, Tizi-Ouzou, pour nous avoir fait honneur de présider le jury de ce mémoire et l'intérêt qu'il a porté à ce travail.

Nous tenons aussi à remercier vivement Mlle **CHILALI Ouardia**, Enseignante au département Automatique, Université Mouloud Mammeri, Tizi-Ouzou, pour avoir bien voulu faire partie du jury de soutenance de ce mémoire. Qu'il trouve, ici, nos plus vifs remerciements.

Que Mlle **AIT-AIDER Malika**, Enseignante au département Automatique trouve ici l'expression de notre profonde gratitude pour l'intérêt qu'il a porté à notre travail en acceptant de participer au jury de ce mémoire.

Nos remerciements vont aussi au chef du département automatique, Monsieur **BENSIDHOUM M-Tahar** pour sa disponibilité son dévouement pour le bon fonctionnement du département.

Sommaire

Introduction générale	1
------------------------------------	---

Chapitre I : Classification automatique

1.1 Introduction.....	3
1.2 Classification automatique.....	4
1.3 Méthodes de classification.....	5
1.3.1 Classification supervisée.....	6
1.3.2 Classification non supervisée.....	7
1.3.2.1 Classification hiérarchique.....	8
1.3.2.2 Classification non hiérarchique	9
1.4 Conclusion.....	14

Chapitre II : Les métaheuristiques

2.1 Introduction	16
2.2 Généralités sur l'optimisation	17
2.3 Les métaheuristiques	19
2.4 Métaheuristiques en classification.....	24
2.4.1 La recherche tabou en classification	24
2.4.2 Le recuit simulé en classification	26
2.4.3 Les algorithmes évolutionnaires en classification	27
2.5 Conclusion.....	29

Chapitre III : Les PSO en classification automatique

3.1 Introduction.....	30
3.2 Principe de base des PSO	32
3.3 Domaines d'application des PSO	35
3.4 Classification par essaim de particules	35
3.4.1 Choix de la fonction objectif	36
3.4.2 Codification des individus.....	37
3.4.3 Génération de la population initiale.....	37
3.4.4 Critère d'arrêt	37

3.5 Classification par essaims d'abeilles artificielles	38
3.6 Classification par algorithme des mouches	39
3.7 Hybridation des K-means avec les PSO	40
3.8 Conclusion	44
Conclusion générale	46
Références bibliographiques	47

Introduction générale

La classification automatique est une branche de l'analyse des données, a pour objectif de répartir les éléments d'un ensemble en groupes, c'est-à-dire d'établir une partition de cet ensemble. Chaque groupe doit être le plus homogène possible, et les groupes doivent être les plus différents possibles entre eux. Nous nous intéressons, dans ce travail, à la classification *y* non subtilisant les métaheuristiques. En effet, la majorité des problèmes d'extraction de connaissances peuvent s'exprimer comme des problèmes d'optimisation et ne peuvent être souvent résolus de manière exacte dans un temps raisonnable, puisque la capacité de calcul des machines évolue linéairement alors que temps nécessaire à la résolution des problèmes évolue exponentiellement. A cette fin, les méthodes qui supervisent évolution de solution fournies par des heuristiques sont apparues.

Ces méthodes assurent un compromis entre la diversification quand il est possible de déterminer que la recherche se concentre sur les mauvaises zones de l'espace de recherche et l'intensification qui cherche les meilleures solutions dans la région de l'espace de recherche au cours d'analyse. Ces méthodes ont été appelées *métaheuristiques* qui ont pour objectif de trouver des solutions dont la qualité est au-delà de ce qu'il aurait été possible de réaliser par une simple heuristique.

Parmi les métaheuristiques les plus connues de nos jours, on peut citer la recherche tabou, le recuit simulé, les algorithmes de colonies de fourmis, les systèmes immunitaires artificiels, les algorithmes génétiques et les essaims particuliers.

Ainsi notre travail est structuré comme suit. Dans le premier chapitre, nous présenterons les différentes méthodes de classification automatique. Le deuxième chapitre décrira les différentes métaheuristiques et quelques exemples sur leur application en classification non supervisée. Le troisième

Introduction générale

chapitre est consacré à l'étude d'une métaheuristique moderne appliquée à un problème de classification non supervisée et les résultats obtenus.

Enfin, nous terminerons par une conclusion générale.

Chapitre 1

Classification automatique

1. Introduction

La reconnaissance de formes est avant toute la réduction méthodique d'information qui regroupe plusieurs phases à savoir l'acquisition des données, l'apprentissage et de décision. Dans la phase d'apprentissage, il s'agit souvent de répartir les objets en différents classes distinctes. Dans la phase de décision, l'objet à identifier est affecté à l'une de ces classes. L'opération à laquelle nous nous intéressons dans ce travail est la classification automatique qui a pour objectif de répartir des objets décrits par un certain nombre de paramètres en classes homogènes et distinctes. En classification automatique, il existe un grand nombre de méthodes utilisant différentes notions mathématiques et elles peuvent être déterministes ou statistiques et elles varient aussi selon l'information *a priori* sur les données. Ce chapitre est consacré à la classification automatique d'une manière générale et à la méthode des k-means en particulier.

2. Classification automatique [1]

La classification automatique permet de regrouper des objets possédant des propriétés similaires en plusieurs classes homogènes de manière à ce que l'intersection deux à deux des classes formées donne un ensemble vide et que l'union de toutes les classes donne l'ensemble initial des données. Notons que le degré de chevauchement entre les classes et la nature multidimensionnelle constituent les difficultés les plus importantes dans la résolution d'un problème de classification.

Ainsi, soit $X = \{X_1, X_2, \dots, X_N\}$ l'ensemble des N observations à classer. Chaque observation X_i est décrite par P paramètres à savoir $X_i = \{X_{i1}, X_{i2}, \dots, X_{iP}\}$ tel que $X_i \in R^N$.

Soit $C = \{C_1, C_2, \dots, C_K\}$ l'ensemble des K classes.

La classification consiste à répartir l'ensemble des N observations en K classes de telle manière :

1. $C_i \neq \emptyset$ pour $i = 1 \dots k$
2. $C_i \cap C_j = \emptyset \quad \forall i \neq j$
3. $\bigcup_{i=1}^k C_i = N$

La première condition indique qu'aucune classe ne doit être vide, la seconde condition précise qu'aucun chevauchement entre les classes n'est toléré et la troisième vérifie que la cohésion de toutes les classes doit aboutir au nombre initial d'observations.

D'une manière générale, on exige de la classification de vérifier que les points représentant une classe donnée sont plus proches entre eux que des points de toutes les autres classes et que les classes doivent être séparées le plus possible (fig.1.1).

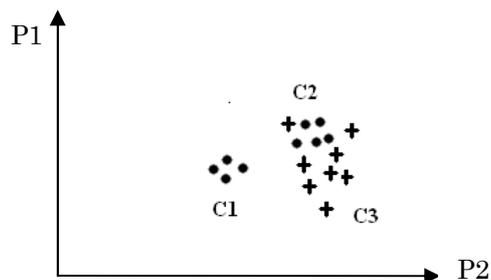


Fig.1.1 Visualisation de classes.

Dans l'exemple de la figure 1 la classe C_1 est distincte alors que les deux classes C_2 et C_3 se chevauchent.

3. Méthodes de classification [2]

D'après certains auteurs, on distingue deux approches importantes de la classification automatique : l'approche supervisée et la proche non supervisée. En effet, Si l'information n'est pas complète, ce qui est souvent le cas, l'utilisateur fait son choix parmi les méthodes supervisées (ou avec professeur) et non supervisées (ou sans professeur). Dans le cas où les données sont classées *a priori*, il s'agit de classification supervisée. L'utilisateur forme alors lui-même des classes pour chaque observation. Dans le cas contraire où l'on ne dispose pas d'information *a priori* sur les données, il s'agira de choisir une méthode non supervisées. Ces méthodes sont encore subdivisées en méthodes paramétrique et non paramétrique compte tenu des informations dont on dispose *a priori* (fig.1.2).

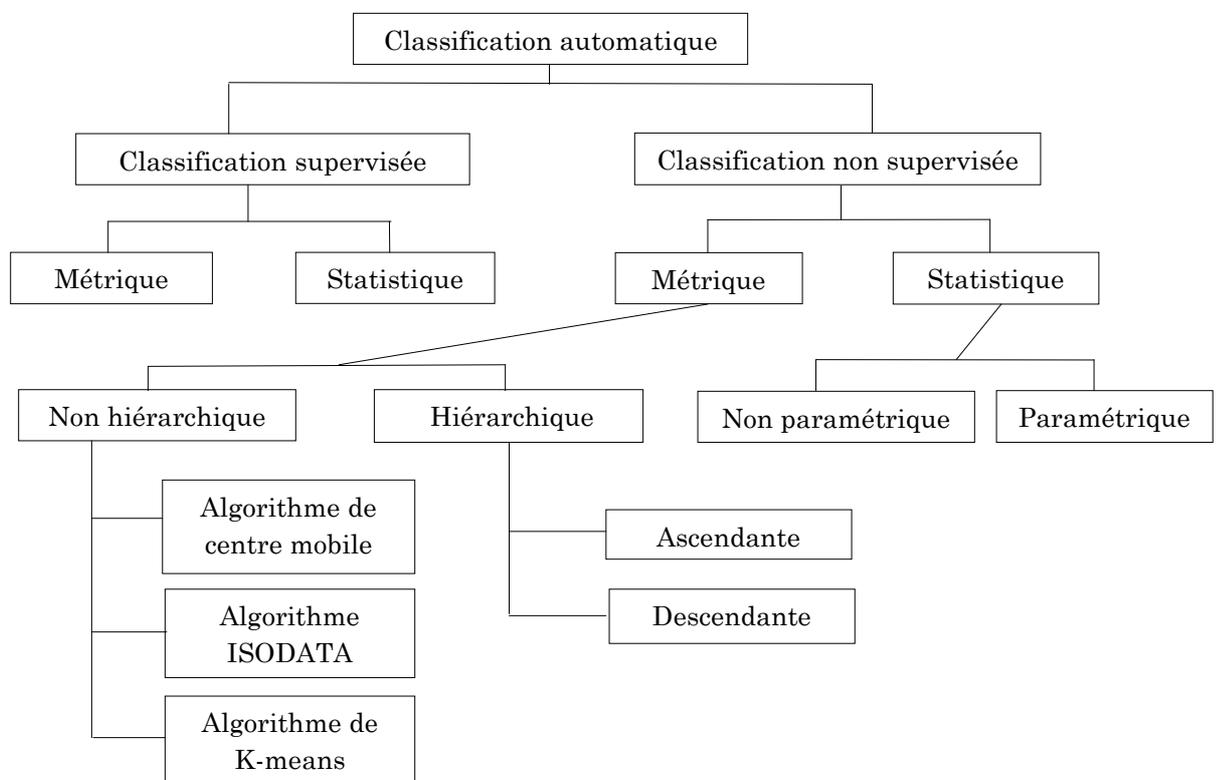


Fig.1.2 Illustration des différentes méthodes de classification automatique.

3.1 Classification supervisée

En Classification supervisée, l'objectif est de concevoir un classifieur capable d'attribuer toute observation inconnue dans une classe qui lui convient. La conception de ce classifieur est basé sur la connaissance *a priori* d'un ensemble d'observations de chaque classe appelé prototype. Le principe de la classification supervisée est de collecter le plus d'informations possibles à partir des groupements connus. Ceci vise à assurer une bonne connaissance des classes et à comprendre, en quelque sorte, pourquoi deux objets distincts sont séparés dans des groupes différents, ou au contraire, rassemblés dans un même groupe. La classification supervisée peut être envisagée par les approches métrique et statistique.

Les méthodes métriques utilisent la notion de distance. Ainsi une observation X_i est affectée à la classe Q_s si seulement si $d_s(X_i) < d_k(X_i)$, $k \neq s$. Les points de l'espace des attributs pour lesquels les fonctions discriminantes $d_s(X)$ et $d_k(X)$ sont égales sont situés sur une surface de dimension $N-1$ appelée *surface de décision* et définie par $d_s(X) = d_k(X)$. Cette dernière partage l'espace en deux régions, l'une pour laquelle $d_s(X) < d_k(X)$, l'autre pour laquelle $d_s(X) > d_k(X)$. Classifier une observation revient à déterminer le domaine de décision auquel elle appartienne. La figure ci-dessous illustre les notions de fonction discriminante et le domaine de décision pour un problème bidimensionnel à deux classes.

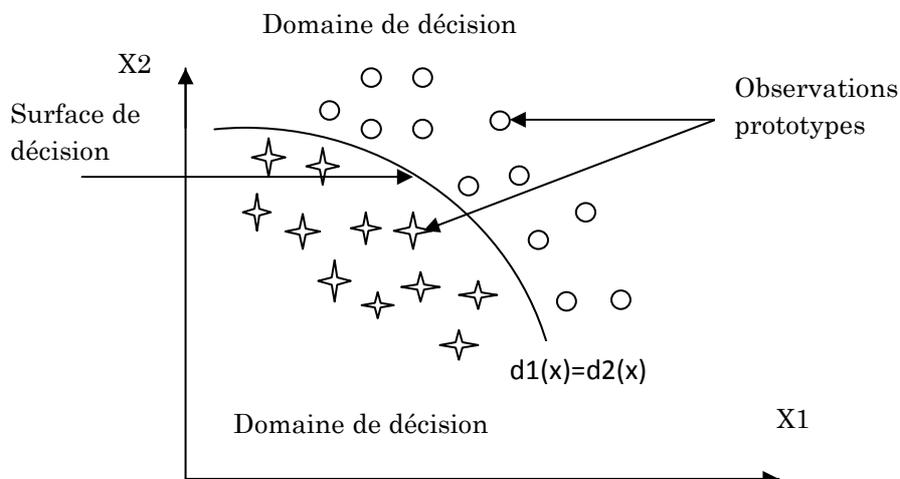


Fig. 1.3. Représentation de surface de décision.

Les méthodes statistiques reposent sur l'étude des caractéristiques statistiques de la distribution des observations ou fonction de densité de probabilité (*fdp*), en effet, toute observation est considérée comme une réalisation particulière d'un vecteur aléatoire continu dont la distribution est définie par une *fdp* multivariable caractéristique de la classe à laquelle appartient l'observation. Ces méthodes utilisent la théorie de la décision qui constitue une approche fondamentale pour résoudre les problèmes de classement et de la classification qui se trouvent posés en terme probabiliste. Cette théorie permet d'effectuer un classement optimal à partir de chaque classe. On rappelle que la règle de décision bayésienne consiste à affecter l'individu à la classe dont la probabilité *a posteriori* calculée par la forme de Bayes est la plus grande.

La formule de Bayes est donnée par

$$P(Q_k/X) = \frac{P_{rk} \cdot f_k(X)}{\sum_{q=1}^Q P_{rq} \cdot f_q(X)} \dots \dots \dots (1.1)$$

où $P(Q_k/X)$ est la probabilité *a posteriori* que l'individu X appartienne à la classe Q_k , P_{rk} , la probabilité *a priori* que l'individu appartienne à la classe Q , $f_k(X)$ la densité de probabilité de X si la classe est Q .

3.2 Classification non supervisée

En classification non supervisée ou « clustering » on ne dispose ni de nombre de classes ni de l'information sur l'appartenance des individus aux classes. C'est une classification sans prototype ou sans professeur. Le regroupement, en classes distinctes, des observations ayant des caractéristiques similaires s'effectue en définissant une mesure de distance entre les observations. Cette notion de distance devra se baser sur des distinctions entre les valeurs prises par les attributs pour les différentes observations. Pour ce mode de classification, on dispose de méthodes statistiques et de méthodes métriques.

Les méthodes statistiques non supervisées présentent l'avantage de rassembler les observations sur lesquelles on ne dispose d'aucune information *a priori* autres que celles qui peuvent être extraites de ces observations. Ces méthodes reposent

sur l'étude de la fonction de densité de probabilité f_{dp} . En effet, toute observation est considérée comme une réalisation particulière d'un vecteur aléatoire continu dont la distribution est définie par une f_{dp} . Pour cette technique on distingue l'approche paramétrique et l'approche non paramétrique. Dans l'approche paramétrique, on ne dispose pas de prototype mais on connaît la forme de f_{dp} de chaque classe que l'on suppose suivre une loi normale. Dans ce cas le problème de classification est amené à celui de la détermination des paramètres d'un mélange d'une f_{dp} représentant les distributions des observations provenant de chacune des classes en présence dans l'échantillon analysé. Dans l'approche non paramétrique, la classification s'apparente, dans ce cas, à la recherche des modes de la fonction de densité de probabilité sous jacente à l'ensemble des observations. Comme la f_{dp} est inconnue, on fait appel à l'estimateur du noyau de Parzen ou l'estimateur des K plus proche voisins. Étant donné que les noyaux des classes correspondent à des régions de l'espace caractérisées par une densité élevée d'observations, leurs localisations sont basées sur la détection des modes de la f_{dp} . Quant aux méthodes métriques non supervisées, elles sont non probabilistes et leur objectif est de montrer le degré de similarité entre les objets en utilisant la notion de distance. On distingue deux principales catégories. La première réalise une hiérarchie des classes tandis que la seconde procède par partitionnement de l'espace des observations.

3.2.1 Classification hiérarchique [3]

Les méthodes hiérarchiques qu'elles soient ascendantes ou descendantes sont parmi les plus utilisées en classification automatique. Dans ces méthodes, on cherche à avoir une hiérarchie ou une suite de partitions "emboîtées" sur l'ensemble des données. La hiérarchie est représentée par un arbre hiérarchique (fig.1.4). Le niveau des nœuds indique le degré de ressemblance entre les objets correspondants. En coupant l'arbre à différents niveaux, on obtient des partitions ascendantes, les méthodes hiérarchiques sont basées sur la fusion d'objets ou groupes d'objets décidée par un critère utilisant la dispersion intraclasse. Notons toutefois que la fusion de deux classes s'effectue non sur un calcul de distances

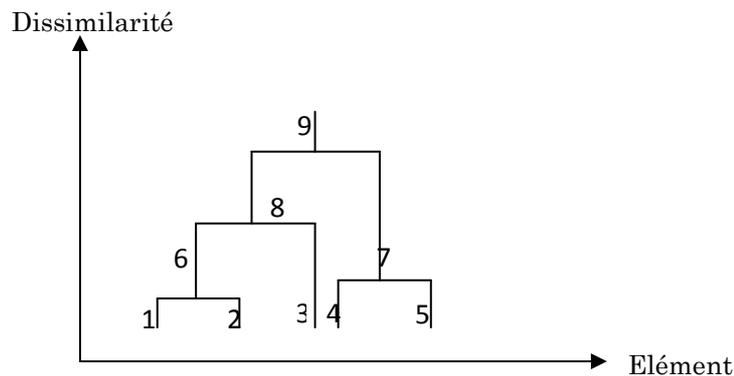


Fig.1.4 Classification hiérarchique.

entre classes mais sur l'augmentation de la dispersion intraclasse. De plus, pour accélérer les calculs, on utilise la notion de voisins réciproques. Descendantes, ces méthodes consistent à diviser d'abord l'ensemble de données, par dichotomie en deux classes, puis chacune de ces classes est elle-même divisée, et ainsi de suite jusqu'à ce que les subdivisions soient homogènes, ou d'effectifs faibles pour se prêter à une partition sûre. Dans notre cas, c'est la dispersion intraclasse qui nous a servi de critère de division. La figure 1.5 donne un exemple d'algorithme hiérarchique ascendant.

1. Initialisation : partition des singletons et calcul de la distance intra classes
2. Tant que le nombre de classes $> K$
 - Regroupement des deux classes les plus proches au sens de D
 - Calcul des distances entre la nouvelle classe et les anciennes classes non regroupées
3. Fin de tant que

Fig.1.5 algorithme hiérarchique ascendant.

3.2.2 Classification non hiérarchique

Contrairement à ces techniques hiérarchiques qui proposent une famille de groupements dont la taille et le nombre de classes varie, il existe d'autres techniques qui produisent une classification en optimisant un critère prédéfini ou une fonction objective. Parmi ces méthodes on peut citer la méthode d'agrégation autour des centres mobiles, la méthode d'ISODATA et l'algorithme des K-means.

Ces algorithmes se ressemblent dans le principe général, mais différent dans la façon de procéder pour aboutir à une partition finale.

- ***L'agrégation autour des centres mobiles***

Cet algorithme peut être considéré comme un cas particulier des techniques connues sous le nom de *nuées dynamiques* proposé par Diday en 1971. Bien qu'elle ne fasse appel qu'à un formalisme limité et que son efficacité soit dans une large mesure attestée par les seuls résultats expérimentaux, il est probablement la technique de partitionnement la mieux adaptée actuellement, produisant des partitions des ensembles étudiés, utilisé aussi bien comme technique de description et d'analyse que comme technique de réduction, généralement en association avec des analyses factorielles et d'autres méthodes de classification.

Son principe consiste à fixer un nombre k de classes et choisir une partition initiale. Cette partition peut être inspirée par une connaissance *a priori* des objets à classer, ou bien, elle peut être obtenue par répartition, au hasard, des objets en k catégories. On exécute alors les opérations de l'algorithme suivant (fig1.6).

1. on détermine q centres provisoires de classes par tirage pseudo-aléatoire. Les q centres induisent une première partition P_0 de l'ensemble des individus 1 en q classes $i=0$
2. tant que le critère d'arrêt n'est pas vérifié faire
3. calculer les centres de gravité des classes de la nouvelle partition P_i
4. affecter chaque objet à la classe dont le centre est plus proche
5. $P_{i+1} \leftarrow P_i$
6. Fin tant que
7. Retourner à 2

Fig.1.6 Algorithme d'agrégation autour des centres mobiles

L'algorithme des centres mobiles, contrairement à de nombreuses méthodes classificatoires, a l'avantage d'optimiser un critère simple de dispersion, savoir le moment d'ordre deux d'une partition.

- **La méthode ISODATA**

Le terme ISODATA est une abréviation de Iterative Self Organizing Data Analysis Techniques la dernière lettre est rajoutée dans le but de faciliter la prononciation du terme ISODATA [4]. La méthode ISODATA se base sur un critère de « qualité » (construction) consistant à minimiser la somme des carrés des écarts (distance) entre chaque objet et le centre du cluster courant qui lui est associé. Comme l'ensemble de ces méthodes, on commence par choisir au hasard K représentant $\{R_1, R_2, \dots, R_K\}$ de K classes $\{C_1, C_2, \dots, C_K\}$ définissant une $R = \{R_1, R_2, \dots, R_K\}$ et $P = \{C_1, C_2, \dots, C_K\}$.

Chacune des observations est assignée à l'une des classes C_K selon une mesure de similarité entre chaque représentation R_k et l'observation considérée. La partition P ainsi obtenue est utilisée pour définir une nouvelle représentation R .

Ce processus itératif, qui inclut les deux phases de définition de la partition P et la réactualisation de la représentation R prend fin lorsqu'un critère mesurant l'adéquation de la représentation R à la partition P est optimisé.

Dans la méthode ISODATA, présentée dans ce qui suit, le représentant de chaque classe C_k est défini par son centre X_k et la mesure de similarité utilisée est la distance euclidienne de chacune des observations aux différents centres X_k $k = 1, \dots, K$. Les étapes d'algorithme de cette méthode sont les suivantes (fig1.7).

1. Choisir arbitrairement le nombre K de classes, ainsi que les centres \bar{X}_k de ces K classes, $k=1,2,\dots, k$
2. Assigner chaque observation à la classe associée au centre le plus proche au sens de la distance euclidienne
3. Si une classe ne contient pas suffisamment d'observations, la supprimer et aller à 2
4. Recalculer les centres \bar{X}_k des classes C_k compte tenu de la nouvelle partition définie à l'étape 2
5. Calculer la norme de la matrice de covariance de chaque classe et la comparer à la valeur autorisée. Si la norme de la matrice de covariance d'une classe C_k est trop grande et la distance moyenne des observations de cette classe à son centre \bar{X}_k est supérieure à la valeur moyenne des distances des observations des différentes classes à leurs centres respectifs, scinder la classe C_k considérée en deux classes distinctes et aller à 2
6. Calculer toutes les distances séparant les différents centres les uns des autres. Si certaines de ces distances sont plus faibles que la distance minimum autorisée séparant deux centres, regrouper les deux classes considérées en une seule et aller à 2.

Fig.1.7 Algorithme ISODATA

Cet algorithme est très simple à implémenter et converge rapidement avec une solution localement optimale. Cependant, son majeur inconvénient est qu'il nécessite de fournir en entrée une partition initiale de bonne qualité ainsi que le nombre possible de classes. Ces contraintes rendent l'utilisation de cet algorithme peu intéressante quand on veut segmenter automatiquement une image.

- **La méthode des k -means** [5]

L'algorithme des k -means est introduit par MacQueen en 1967, est la méthode la plus utilisée dans les applications scientifiques et industrielles de « clustering ». C'est un algorithme de partitionnement déterministe. Le nom dérive du fait que, pour représenter chacune des K classes C_k , on utilise la moyenne (ou la moyenne pondérée) \bar{X}_k de ses points, appelé centre de classe.

Chacune des ses composantes du vecteur \bar{X}_k est calculée par

$$\bar{X}_{kj} = \frac{1}{Q_k} \sum_{i=1}^{Q_k} X_{ij} \dots\dots\dots (1.2)$$

L'objectif est alors de minimiser la somme de l'inertie intraclasse sur l'ensemble des classes. L'algorithme procède en deux étapes : Dans la première phase, on réassigne tous les objets au centre le plus proche, et dans la deuxième phase, on recalcule les centres des classes qui ont été modifiées. Pour mesurer la proximité entre un centre et un objet, on calculera une distance entre ces deux vecteurs. On pourra utiliser, par exemple, la distance euclidienne, calculée de la manière suivante :

$$d(X_i, \bar{X}_K) = \|X_i - \bar{X}_K\| = \sqrt{\sum_{j=1}^P (X_{ij} - \bar{X}_{Kj})^2} \dots\dots\dots (1.3)$$

Dans le cas de données numériques, cela donne un sens géométrique et statistique à la méthode. Dans l'algorithme des k-means, l'inertie intraclasse est définie comme la moyenne des carrés des distances des objets de la classe au centre de gravité de celle-ci qui cherche ainsi à construire des classes compactes. L'inertie intraclasse constitue le critère à optimiser, qui se traduit par la minimisation de l'énergie suivante :

$$j(\pi) = \sum_{k=1}^K \sum_{i=1}^{Q_k} d^2(X_i, \bar{X}_K) = \sum_{k=1}^K J_k(\pi) \dots\dots\dots (1.4)$$

Où π est une partition composée de K classes

La partition π qui optimise le critère est définie par

$$j(\pi) = \min j(\pi) \dots\dots\dots (1.5)$$

L'algorithme général des k-means se déroule comme suit (fig.1.8)

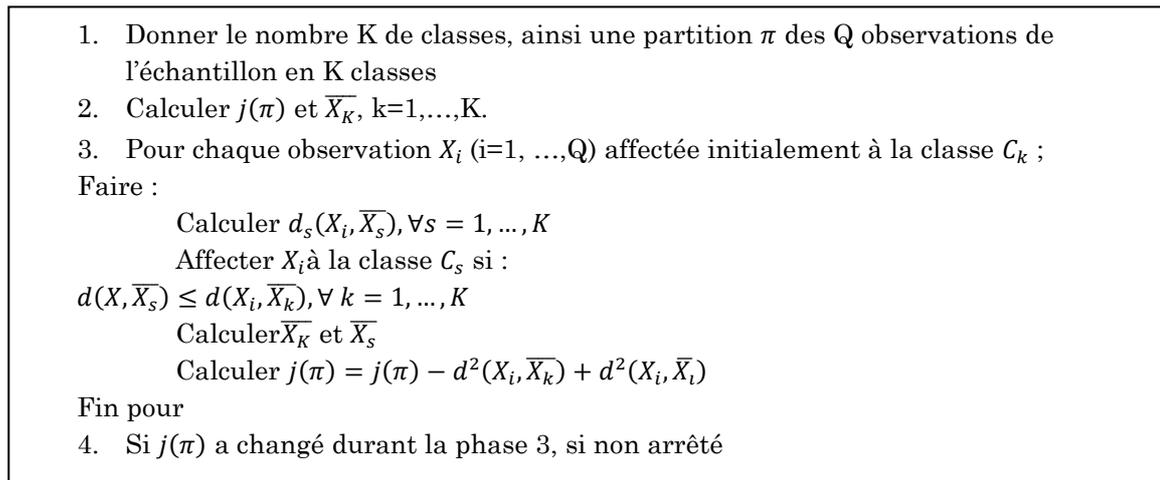


Fig.1.8 Algorithme des k-means.

Les deux phases sont itérativement répétées jusqu'à ce qu'un critère d'arrêt soit atteint, par exemple, si aucune modification n'a eu lieu, ou si le nombre maximum d'itérations a été atteint.

Les principaux problèmes de l'approche des *k-means* comme des autres approches partitionnelles, sont l'influence de la partition initiale (qui est souvent choisie de façon aléatoire), et le choix du paramètre K qui n'est pas toujours évident. En effet, la convergence de l'algorithme vers une solution satisfaisante n'est pas assurée, ce qui impose dans la pratique, de multiplier les initialisations, et augmenter d'autant le temps de calcul.

Il existe d'autres variantes de cette méthode comme celle des fuzzy c-means (FCM) initialement proposé par Dunn en 1973 [6] puis amélioré par Bezdek [7] en 1981. Cette méthode permet d'obtenir un regroupement des éléments par une approche floue avec un certain degré d'appartenance où chaque élément peut appartenir à une ou plusieurs classes, à la différence de k-means, où chaque observation appartient à une seule classe (partition dure).

4. Conclusion

Nous venons d'exposer, dans ce chapitre, un aperçu sur la classification automatique et ses différentes variantes en particulier sur l'approche de classification automatique non supervisée et non hiérarchique. L'objectif de ces méthodes est alors d'obtenir des clusters d'objets homogènes, reposant sur une

mesure de distance qui est basée sur des distinctions entre les valeurs prises par des attributs pour différents objets, suivent un même principe général qui consiste à minimiser le moment intraclasse et maximiser le moment interclasse. Ces méthodes présentent l'avantage d'être rapides, simples et efficace, mais leur inconvénient majeur est que la solution obtenue dépend étroitement des centres initiaux et aboutissent souvent à un optimum local. Cependant, il existe des méthodes permettant de procéder à la détermination automatique de ce nombre de partitions initiales. C'est ainsi que les métaheuristiques d'optimisation ont été introduites pour améliorer les résultats d'une classification. Pour cela, plusieurs travaux dans ce domaine continuent encore de paraître.

Chapitre 2

Les métaheuristiques

1. Introduction

Dans la vie courante, nous sommes, souvent, confrontés à des problèmes qui peuvent être décrits sous forme d'un problème d'optimisation. Cette spécialité a connu un grand développement ces dernières années. Ceci est dû à l'importance accordée par les industriels à cette discipline pour minimiser les coûts de production et accroître les bénéfices et, aussi, à la croissance de la puissance de calcul des ordinateurs produits ces vingt dernières années.

Pour la résolution de tels problèmes, on met souvent en évidence, une fonction objective qu'il s'agit d'optimiser, c'est-à-dire, minimiser ou maximiser. Cependant, pour les problèmes dits difficiles, on ne connaît pas d'algorithmes exacts rapides permettant de les résoudre. Même pour les autres, dits à variables continues, il n'existe pas, non plus, d'algorithmes permettant de repérer un optimum global à coup sûr et en un nombre fini de calculs.

Ainsi, dans ce chapitre, seront présentées, en premier lieu, des notions générales sur l'optimisation et les métaheuristiques. En second lieu et, d'une

manière plus détaillée, nous présenterons la méthode des essaims de particules, objet de ce projet. Ce chapitre sera terminé par une conclusion.

2. Généralités sur l'optimisation

En optimisation, le problème peut se présenter sous différentes appellations, selon sa complexité. Il peut être de classe P, s'il est facile ou résolu par l'application des méthodes classiques. Par contre, dans le cas de convexité stricte, d'existence de discontinuité, de fonction non dérivable et de présence de bruit, on parle de problèmes d'optimisation *difficile*. Ces derniers sont de deux types, à savoir, les problèmes combinatoires (ou problèmes discrets) comme le voyageur de commerce et les problèmes à variables continues. La figure 2.1 présente les différentes méthodes d'optimisation.

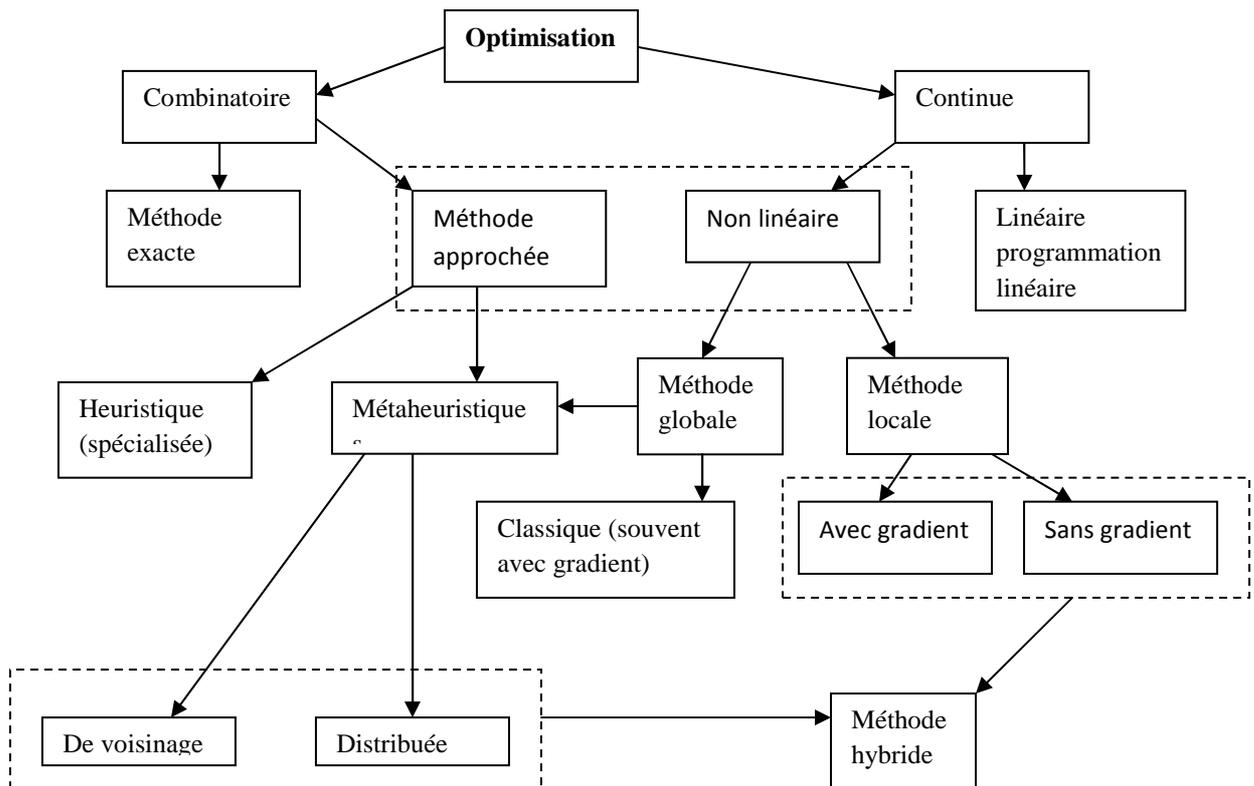


Fig.2.1 Présentation des différents algorithmes d'optimisation [8]

Dans le cas de méthodes linéaires à variables continues, la programmation linéaire est utilisée. Dans le cas où elles sont non linéaires, il s'agit d'une optimisation difficile continue et les méthodes d'optimisation utilisées pour résoudre ce type de problèmes sont classées en deux catégories : les méthodes locales qui permettent de déterminer un minimum local qui est le premier minimum rencontré, et les méthodes de recherche globale qui déterminent un optimum global qui est le minimum absolu. Pour déterminer un minimum global, certaines méthodes doivent éviter le piègeage dans les optima locaux. Cependant, pour améliorer les performances d'une recherche, plusieurs auteurs hybrident les deux types d'algorithmes [9] [10].

Notons, toutefois que, jusqu'à présent, ce sont, plutôt, les *méthodes combinatoires ou discrètes* regroupant les *méthodes exactes* et les *méthodes approchées* qui sont les plus formalisées. Néanmoins, lorsqu'on veut résoudre un problème complexe, ces méthodes prennent un temps de calcul qui croît exponentiellement avec la taille des instances du problème. Ceci conduit souvent à une explosion combinatoire. Comme nous l'avons évoqué, précédemment, pour éviter ce type de problème, on ne cherche pas forcément l'optimum absolu. Une solution très proche de l'optimum absolu montrant l'inexistence d'une solution sensiblement meilleure est largement acceptée. Ceci est obtenu en utilisant les méthodes regroupant les heuristiques et les métaheuristiques. Nous rappelons que la plupart des heuristiques sont spécifiques à un problème donné et ne s'appliquent, en général, qu'à des problèmes discrets. Elles sont de complexité raisonnable, autrement dit, idéalement polynomiales mais efficaces et simples à mettre en œuvre. Quant aux métaheuristiques, elles font l'objet du paragraphe suivant. Dans tous les cas, il est nécessaire de définir une fonction objectif F dont on cherchera les minima. Cette fonction est définie telle que :

$$F: X \rightarrow \mathfrak{R} \quad x \rightarrow f(x)$$

où $X \in \sigma$ Sur un ensemble de solutions $x / X \subset \mathfrak{R}$ (\mathfrak{R} est l'ensemble des réels). Ainsi, on a :

$$F(x^*) \text{ minimum local} \Leftrightarrow [\exists \varepsilon > 0 / \forall x \in X : \|x - x^*\| < \varepsilon \Rightarrow F(x) \geq F(x^*)] \text{ et } x^* \in X;$$

$F(x^*)$ minimum global $\Leftrightarrow \forall x \in X: F(x) \geq F(x^*)$ et $x^* \in X$.

3. Les métaheuristiques

Apparues dans les années 1980, les métaheuristiques forment un ensemble d'algorithmes permettant de trouver la solution la plus rapide et la plus efficace pour une large gamme de problèmes d'optimisation difficile et pour lesquels on ne connaît pas de méthode classique plus efficace. Comme le montre la figure 2.2 suivante, il s'agit de trouver l'optimum global G sans être piégé par les autres optima locaux tel que le point L [11].

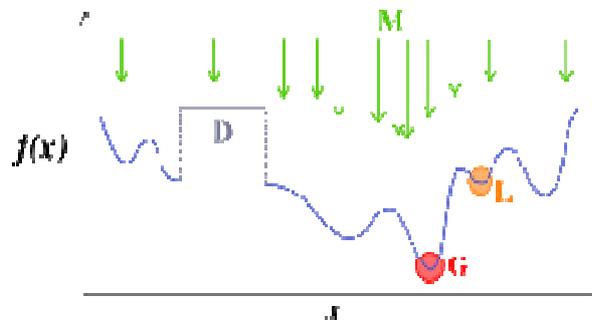


Fig.2.2 Représentation du minimum local et global d'une fonction

Pour ce faire, les métaheuristiques, généralement, s'articulent autour des trois notions suivantes [12] :

- la diversification (exploration)
- l'intensification (exploitation)
- la mémorisation (apprentissage)

La diversification ou exploration désigne le processus qui dirige la procédure pour récolter de l'information sur le problème à optimiser. La stratégie de diversification la plus simple consiste à redémarrer périodiquement le processus de recherche à partir d'une solution générée aléatoirement ou choisie judicieusement dans une région non encore visitée de l'ensemble des solutions admissibles. Pour sa part, l'intensification ou exploitation utilise l'information déjà récoltée pour explorer, en détail, les zones jugées prometteuses dans l'espace de recherche. Sa mise en œuvre réside, le plus souvent, dans

l'élargissement temporaire du voisinage de la solution courante. Quant à la mémorisation, elle est le support de l'apprentissage qui permet à l'algorithme de ne tenir compte que des zones où l'optimum global est susceptible de se trouver, évitant ainsi, les optima locaux qui sont de bonnes solutions, mais qui ne sont pas les meilleures des solutions possibles [13].

En alternant l'intensification, la diversification et la mémorisation, le fonctionnement des métaheuristiques est progressif et itératif. L'étape initiale est souvent choisie de façon aléatoire et l'étape d'arrêt est souvent fixée à l'aide d'un critère d'arrêt.

Toutes les métaheuristiques s'appuient sur l'équilibre entre l'intensification et la diversification de la recherche. Sinon, on assistera à une convergence trop rapide vers des minima locaux par manque de diversification ou à une exploration trop longue par manque d'intensification.

Le fonctionnement des métaheuristiques est généralement inspiré à partir des systèmes physiques comme le recuit simulé, biologiques comme les algorithmes évolutionnaires, ethnologiques comme les algorithmes de colonies de fourmis ou de l'optimisation par essais particuliers etc. Ainsi, elles peuvent être réparties en deux catégories : les méthodes à base de voisinage correspondant à la recherche locale et les méthodes à base de population correspondant à la recherche globale [14] (fig.2.3).

Les métaheuristiques à base de voisinage sont les méthodes de recherche locale. Elles sont itératives : à partir d'une solution unique x_0 considérée comme point de départ, la recherche consiste à passer d'une solution à une solution voisine par déplacements successifs dans un voisinage constitué de l'ensemble des solutions. Souvent, les opérateurs de recherche locale s'arrêtent quand une solution localement optimale est trouvée. Mais accepter uniquement ce type de solution n'est pas toujours satisfaisant. Il est alors

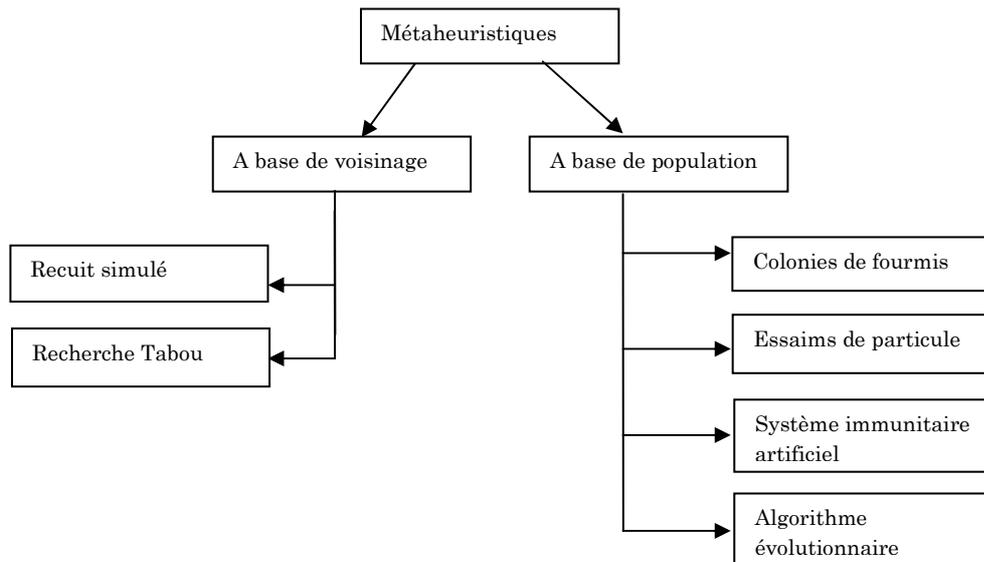


Fig.2.3 Les catégories des métaheuristiques

important de sortir de ces minima locaux en permettant à l'opérateur de recherche locale de trouver des points pour lesquels la nouvelle solution retenue sera de qualité meilleure que la précédente. C'est le principe adopté pour la recherche tabou et le recuit simulé. Dans la recherche tabou développée par Glover, le mot tabou est défini comme une sorte d'interdiction religieuse par laquelle un objet est déclaré intangible [15]. Le principe de cette méthode est de passer à la meilleure solution voisine de la solution initiale même si celle-ci est plus mauvaise que la solution initiale. Ce critère autorise la dégradation de la fonction objectif et évite ainsi le blocage dans un minimum local. Cette méthode incorpore une mémoire adaptative qui permet de prendre avantage de l'histoire de la recherche et, l'exploitation stratégique dérive de l'hypothèse qu'un mauvais choix stratégique peut produire plus d'information qu'un bon choix aléatoire. Dès qu'une solution plus mauvaise que la solution courante est retenue, un risque de cycle existe. Pour éviter ce phénomène une liste tabou T de solutions déjà visitées est mise à jour. La procédure est arrêtée lorsqu'un nombre donné d'itérations est effectuées sans qu'il y ait d'amélioration de la solution.

La méthode de recuit simulé est une technique d'optimisation publiée en 1982 par Kirk Patrick [16]. Elle trouve ses origines dans la thermodynamique. Cette méthode issue d'une analogie entre le phénomène physique de refroidissement

lent d'un corps en fusion, qui le conduit à un état solide de basse énergie. Il faut abaisser lentement la température pour que le corps atteigne l'équilibre thermodynamique. Pour les matériaux, cette basse énergie se manifeste par l'obtention d'une structure régulière, comme dans les cristaux et l'acier. Ainsi la fonction coût correspond à l'énergie du système et la température sera représentée par un paramètre T dans la procédure. L'objectif est de minimiser la fonction coût suivant le paramètre T . la procédure opère sur une solution initiale si avec une température T_0 comme une étape initiale. Ensuite, on se déplace vers une solution s_j choisie dans le voisinage de s_i défini pour $V(s_i)$. La solution s_j est choisie comme solution de départ pour prochaine étape selon sa qualité. Ainsi si $F(s_j) \leq F(s_i)$ alors la solution s_j sera choisie comme solution de départ, si non s_j est acceptée avec une probabilité P_j qui est égale à $\exp(-d/T)$ où $d = f(s_j) - f(s_i)$. On réitère cette règle un certain nombre de fois fixé au départ. Ensuite on diminue la température T et on effectue une nouvelle série d'itérations à cette température. Le processus global s'arrête lorsqu'on atteint une température minimale. On note que le rôle du paramètre T est déterminant pour le choix de la solution courante car, à haute température, $\frac{d}{T}$ tend vers 0. Cela implique $\exp\left(\frac{-d}{T}\right)$ tend vers 1. Donc la plupart des solutions trouvées sont acceptées et ainsi le processus de recherche est pratiquement aléatoire. A basse température $\frac{d}{T}$ est très supérieur à 0, cela implique $\exp\left(\frac{-d}{T}\right)$ tend vers 0 donc la plupart des solutions augmentant la fonction coût sont systématiquement rejetées. Ce principe général a conduit à une transposition de ces états physiques au domaine de l'optimisation en définissant une équivalence entre les des deux domaines.

Contrairement aux méthodes précédentes, les méthodes de recherche à base de population travaillent sur un ensemble de points de l'espace de recherche ou population de solutions. Ces métaheuristiques sont inspirées de la biologie et utilisant des phénomènes d'auto-organisation. Parmi ces algorithmes de population on trouve, particulièrement, les algorithmes génétiques, les colonies de fourmis, les essaims de particules et les systèmes immunitaires artificiels

etc. Les algorithmes génétiques s'inspirent des principes de la génétique. Pour cette raison, ces algorithmes accordent une grande importance à la distinction entre la représentation génétique d'un individu (*génotype*) et sa représentation réelle (*phénotype*). L'opérateur principal utilisé par les algorithmes génétiques pour la construction de nouvelles solutions et l'opérateur de *recombinaison* (appelé aussi *croisement*). Cet opérateur récupère des parties du génotype de deux ou plusieurs solutions (parents), qu'il combine pour construire un (ou plusieurs) nouveau génotype (enfant), héritant ainsi de certaines de leurs caractéristiques. L'utilisation de la recombinaison, seule, ne permet pas d'introduire du matériel génétique nouveau, puisque cet opérateur combine le matériel déjà présent dans la population. Pour remédier à ce problème, les algorithmes génétiques utilisent *la mutation*, comme opérateur secondaire permettant d'introduire de nouveaux gènes inexistant dans la population.

Le principe de l'algorithme des colonies de fourmis est basé sur la manière dont les fourmis cherchent leurs nourritures et retrouvent leur chemin pour retourner dans la fourmilière. Initialement, les fourmis explorent les environs de leur nid de manière aléatoire [17]. Sitôt qu'une source de nourriture est repérée par une fourmi, son intérêt est évalué (quantité et qualité) et la fourmi ramène un peu de nourriture au nid. 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. Les traces laissées s'accumulent au fur et à mesure que la piste est rejointe par plus de congénères. Les phéromones ont comme caractéristique l'évaporation en fonction du temps. Les pistes les plus longues seront donc abandonnées au profit de la plus courte. C'est la piste la plus courte. Pour transposer ce comportement à un algorithme général d'optimisation combinatoire, un modèle relatif à cette évaporation a été mis au point. Une analogie est faite entre l'aire de recherche de nourriture et l'ensemble de solutions admissibles du problème, entre la quantité ou la qualité de la nourriture et la fonction objectif à optimiser et, enfin, entre les traces et une mémoire adaptative. Dans le cas de Systèmes immunitaires artificiels (Artificial Immune Systems), le terme système immunitaire artificiels (AIS) s'applique à une large gamme de

systemes différents, notamment aux métaheuristiques d'optimisation, sont apparus dans les années 90 et sont inspirées du fonctionnement du système immunitaire humain qui est mécanisme de défense capable d'apprendre. Les systèmes immunitaires artificiels sont bien adaptés à l'extraction de connaissance dans plusieurs domaines différents comme la robotique, la détection d'anomalies ou l'optimisation et donnent des résultats intéressants. L'approche utilisée dans les algorithmes AIS est très voisine de celle des algorithmes évolutionnaires, mais a également été comparée à celle des réseaux de neurones. On peut, dans le cadre de l'optimisation difficile, considérer les AIS comme une forme d'algorithme évolutionnaire présentant des opérateurs particuliers. Pour opérer la sélection, on se fonde par exemple sur une mesure d'affinité entre le récepteur d'un lymphocyte et un antigène ; la mutation (exemple de rétroaction négative) s'opère quant à elle via un opérateur d'hyper-mutation directement issu de la métaphore. Parmi les autres métaheuristiques à base de population, nous allons intéresser, particulièrement, ci-après, à l'optimisation par essaims de particules (Particle Swarm Optimisation), est publiée en 1995 est issue d'une analogie avec les comportements collectifs de déplacement d'animaux. Le principe de cet algorithme est qu'une particule ne décide de son prochain mouvement qu'en fonction de la meilleure position qu'elle a rencontrée et en fonction de son meilleur voisin. Cette méthode est plus détaillée dans le prochain chapitre.

4. Métaheuristiques en classification

4.1 La recherche tabou en classification [18]

La recherche tabou a été appliquée au problème de la classification non supervisée par Al-Sultan en 1995. Le programme démarre avec un partitionnement aléatoire en k classes. Pour chaque donnée appartenant à la classe k' , on génère un nombre aléatoire R tel que $0 \leq R \leq 1$. Si $R \geq p_t$ (p_t est un paramètre de l'algorithme qui détermine l'étendue du voisinage), alors la donnée est ôtée de sa classe originale k' et affectée aléatoirement à une classe

k'' telle que $k' \neq k''$. Si non, la donnée reste dans sa classe d'origine. On procède ainsi pour générer un certain nombre de solutions voisines (chaque parcours de toutes les données génère une solution). On applique une fonction objectif à chaque solution et on les trie de la plus performante à la pire. Puis en commençant par la meilleure solution et jusqu'à ce qu'une solution soit choisie, on vérifie si elle contient une affectation qui est dans la liste taboue (en effet, ici la liste contient des affectations de type $o_i : k' \rightarrow \text{classe } k''$) si ce n'est pas le cas, elle est prise comme nouvelle solution courante. Si par contre, elle contient une affectation taboue, on peut la choisir uniquement si c'est la meilleure solution qu'on ait trouvée jusqu'à présent. Et si on parcourt toutes les solutions sans en choisir aucune (toutes taboues et pas les meilleures) on repart avec une autre solution choisie aléatoirement (fig.2.4).

Une des améliorations les plus importantes qui ont été apportées à cet algorithme est son hybridation avec le k-means. En effet, au lieu de générer la première solution aléatoirement, on le fait avec le k-means. Les études ont montré que cette approche était plus efficace que l'algorithme du k-means et celui de la recherche tabou seule.

Cette méthode présente deux avantages dont le premier, sort plus facilement des optima locaux que le k-means et le second est elle évite la plupart des cyclages grâce à la liste taboue. Ses principaux inconvénients sont ; le nombre de classes qui doit être connu à l'avance, et un cyclage dont la longueur est supérieure à la taille de la liste taboue peut avoir lieu. Et si on prend une liste trop grande pour éviter ce problème, l'algorithme devient lent et la mémoire est saturée.

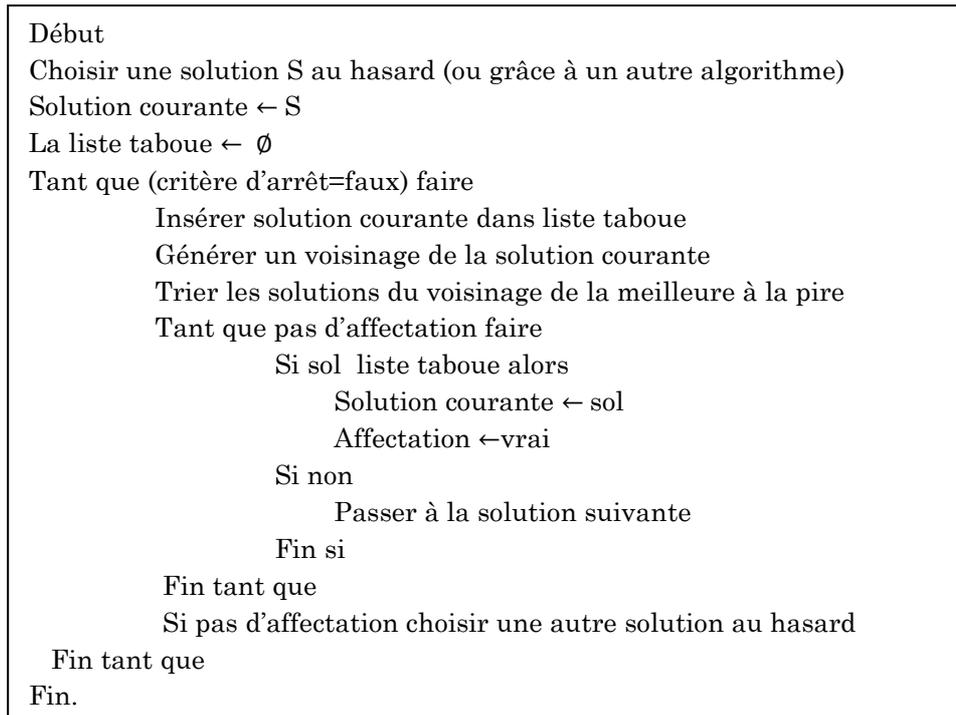


Fig.2.4 Algorithme de la recherche tabou en classification

4.2 Le recuit simulé en classification

L'algorithme de recuit simulé a été utilisé pour résoudre de très nombreux problèmes combinatoires. On peut citer le problème de voyageur de commerce, l'organisation de réseaux, le placement de composants électroniques, etc. le point fort de cet algorithme est sa capacité à sortir des optima locaux, et ce grâce à sa fonction d'acceptation. En effet, même si une solution plus mauvaise que la solution courante, on peut quand même l'explorer selon une certaine probabilité. La convergence du recuit simulé vers la solution optimale dans un problème d'optimisation a été démontrée par Aarts et al en 1985. Cet algorithme a été adapté pour la classification non supervisée, il se résume en quatre étapes (fig.2.5).

La méthode de recuit simulé présente aussi deux avantages ; la grande capacité à sortie des optima locaux et la convergence est garantie vers la bonne solution. Ses principaux inconvénients résume dans le nombre de classe doit également être connu à l'avance et d'une manière générale, cet algorithme est plus lent que les autres métaheuristiques.

Etape 1 : partitionner les n données en k classes aléatoirement. Initialiser le compteur de boucle $nb=0$. Et calculer les centres de gravités des classes.

Etape 2 : perturber la partition actuelle en choisissant aléatoirement une donnée et en l'affectant aléatoirement à une autre classe. Calculer les nouveaux centres de gravité.

Etape 3 : soit E l'erreur quadratique de l'ancienne partition et E' celle de la nouvelle. $\Delta E = E' - E$.

La nouvelle perturbation est acceptée si $\exp\frac{-\Delta E}{T_{nb}} > \gamma$ où γ est un nombre choisi aléatoirement dans l'intervalle $[0,1]$ et T_{nb} la température correspondant à nb^* .

Etape 4 : si $nb \geq maximum$ ou E assez petit alors fin du programme, si non incrémenter nb et aller à l'étape 2.

Fig.2.5 Algorithme du recuit simulé en classification.

4.3 Les algorithmes évolutionnaires en classification [19]

De très nombreux travaux ont été menés pour la classification par algorithmes évolutionnaires, et cela pour réaliser différentes tâches : classification de zones géographiques, classification d'images médicales, etc. chacune avec un codage et des opérateurs spécifiques.

Dans un algorithme évolutionnaire pour la classification, la génération des individus composant la population de base est soit aléatoire, soit résultat de l'application du k-means à des individus générés aléatoirement, comme cela est fait pour d'autres métaheuristiques. Un individu présente le plus souvent une certaine partition des objets à classer. La difficulté réside alors dans le codage des partitions en individus. Ce codage consiste à garder uniquement les centres de gravité des classes dans un individu. La partition est ensuite construite sur le même principe que le k-means (l'objet o_i appartient à la classe dont le centre est le plus proche, selon la distance euclidienne). Ici, on est proche du formalisme génotype/phénotype des algorithmes génétiques. Le génotype étant les centres de gravité et les phénotypes étant les partitions construites. Pour ce type de codage des individus en centres de gravité, le croisement est plus délicat. On peut échanger des centres de gravités selon la méthode des points de coupure, ou alors faire la moyenne des valeurs de centres de gravités. En ce qui concerne la mutation, l'ajout d'une réalisation

d'une loi normale centrée $N(0,\sigma)$ permet de sortir des optima locaux et de pouvoir explorer l'espace de recherche. On peut également utiliser l'algorithme du k-means comme opérateur de mutation, mais cela risque d'accélérer la convergence vers un optimum local. La sélection pour le remplacement se fait selon la valeur de la fonction de fitness appliquée à chaque individu. Son avantage est la petite taille des individus, ce qui permet de manipuler des populations d'assez grande taille. Mais d'un autre côté, il comporte plusieurs inconvénients ; Le premier est la difficulté de définir les opérateurs de variation, le second étant de ne construire que des classes convexes (comme le k-means), nous y reviendrons plus en détail dans la section suivante consacrée au codage par anticorps. Cette codification est empruntée aux systèmes immunitaires artificiels pour représenter une partition. En effet, on démarre d'un nombre fixe de données (anticorps) qui sont générées aléatoirement. On fera évoluer ces anticorps pour qu'ils convergent vers les emplacements où ils maximiseront la fonction objectif. On considérera que deux anticorps font partie de la même classe si la distance les séparant est inférieure à un certain seuil. L'intérêt de cette méthode est qu'une classe est représentée non pas par un point (centre de gravité) mais par plusieurs points (anticorps). Le centre de gravité ne permet de former une classe si elle est incluse dans la sphère qui l'entoure, la classe sera donc toujours convexe (fig.2.6b). Alors qu'avec plusieurs points, on peut former des classes de formes variées composées de l'union des sphères autour des anticorps (fig.2.6c).

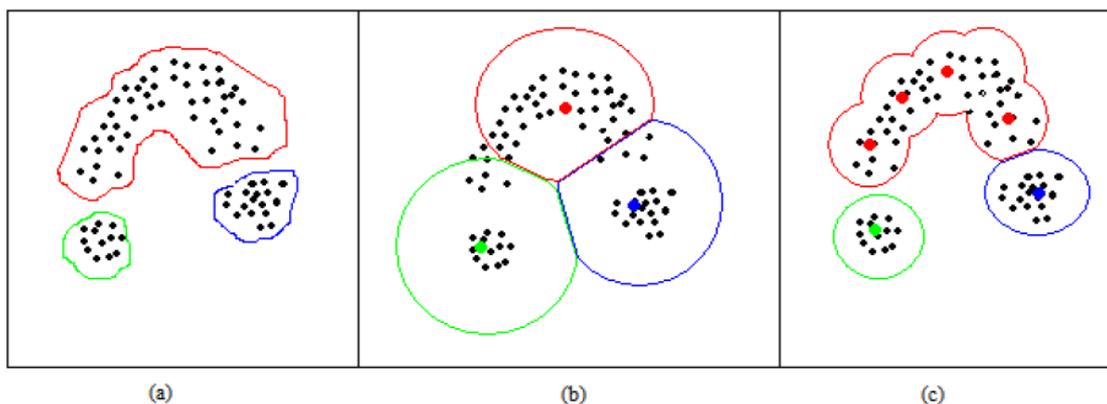


Fig.2.6 Exemple de classification de données avec algorithme évolutionnaire.

(a) trois classes de points sur un plan entourées par des couleurs différentes. Trois classes obtenues avec un centre de gravités (point coloré) pour chaque classe (b), on remarque qu'il ya des points mal classés. Trois classes obtenues en fusionnant les cercles autour des anticorps (points colorés). On est proche de la classification optimale en (a).

5. Conclusion

Devant l'inefficacité des méthodes d'optimisation classiques ou déterministes à résoudre des problèmes de NP-difficiles, plusieurs métaheuristiques ont été développées sur la base d'observation de phénomènes physiques, biologiques sociaux pour apporter des solutions à ce type de problèmes. Méthodes approchées, dans la pratique, elles ont toutefois montré leur grande efficacité pour fournir des solutions approchées de bonne qualité pour un grand nombre de problèmes d'optimisation classiques et d'applications réelles de grande taille. Les métaheuristiques sont adaptables et applicables à une large classe de problèmes, représentées essentiellement par des *méthodes de voisinage*, et les *méthodes de population*. Cependant, les difficultés principales rencontrées sont, la conception d'une mémoire échantillonnant correctement le problème et dont il est aisé d'extraire l'information pertinente pour orienter la recherche, l'équilibrage de la balance entre les techniques d'intensification et de diversification ainsi le maintien de la flexibilité de l'algorithme de façon à ce qu'il s'adapte au problème. Afin de résoudre des instances de taille et de difficultés croissantes, il faut mettre au point des méthodes toujours plus puissantes. Pour atteindre cet objectif, au moins deux voies privilégiées se développent : l'hybridation de la méthode et la parallélisation.

Chapitre 3

Les PSO en Classification Automatique

1. Introduction

Plusieurs différentes méthodes ont été dédiées à la classification automatique non supervisée. Cependant, l'orientation vers les métaheuristiques a donné, dans certains cas difficiles, de très bons résultats. Les métaheuristiques constituent une sorte de modèle ou de squelette d'algorithme qu'il faudra adapter à un problème donné. Leur utilisation pour résoudre le problème de la classification non supervisée a donné lieu à de nombreux travaux. La recherche tabou et le recuit simulé ont été utilisés seuls ou hybridés avec des variantes du k-means. Mais le domaine le plus florissant actuellement est celui des algorithmes biomimétiques ; les algorithmes évolutionnaires parmi lesquels les algorithmes génétiques, les algorithmes multi-agents mimant le comportement social des animaux et plus particulièrement des insectes. Ce comportement collectif des animaux a toujours été une source d'inspiration pour résoudre des problèmes d'optimisation. On a pu constater que certains groupes comme

certaines espèces de poissons, d'oiseaux, insectes sociaux, etc. avaient un comportement très complexe alors que les individus qui les composent n'ont accès qu'à des informations limitées et réagissent selon un schéma assez simple. L'observation des groupes en essaim (fig.3.1) a permis l'élaboration d'une métaheuristique appelée PSO qui est une méthode née en 1995 aux Etats-Unis sous le nom de *Particle Swarm Optimization* (PSO) par Russel Eberhat et James Kennedy [20].



Fig.3.1 Essaim d'oiseaux en vol [21]

Ainsi, initialement, ces deux concepteurs à l'origine de cette méthode cherchaient à modéliser des interactions sociales entre des « agents » devant atteindre un objectif donné dans un espace de recherche commun, chaque agent ayant une certaine capacité de mémorisation et de traitement de l'information. La règle de base était qu'il ne devait y avoir aucun chef d'orchestre, ni aucune connaissance par les agents de l'ensemble des informations, seulement des connaissances locales. Dès les premières simulations, le comportement collectif de ces agents évoquait celui d'un essaim d'êtres vivants convergeait vers des sites intéressants qui se trouvent dans bien d'autres modèles inspirés des systèmes naturels. Ici, la métaphore la plus pertinente est probablement celle de l'essaim d'abeilles, du fait

qu'une abeille ayant trouvé un site prometteur sait en informer certaines de ses consœurs et que celles-ci vont tenir compte de cette information pour leur prochain déplacement. Ces algorithmes sont aussi étudiés par M. Clerc qui a publié son premier ouvrage en français entièrement consacré aux PSO expliquant les principes de base [22]. A partir de 1998, ces principes ont été appliqués pour la première fois à un problème de classification [23].

L'algorithme PSO est une méthode itérative stochastique pour résoudre de nombreux problèmes d'optimisation. Cette méthode a connu beaucoup de succès auprès de la communauté d'optimisation, ses bonnes performances dans différentes applications, et la possibilité d'hybridation avec d'autres métaheuristiques ont contribué à cet engouement et malgré son "jeune âge" en comparaison des autres métaheuristiques, un nombre important de travaux ont été publiés.

2. Principe de base des PSO [24]

Le principe général d'algorithme de l'essaim de particules repose sur un ensemble d'individus initialement disposés de façon aléatoire et homogène. Appelés particules, ces individus se déplacent dans l'hyper-espace de recherche et chacune constitue une solution potentielle. Assimilé à un système dynamique, dans l'essaim, les positions et les vitesses des particules constituant l'état du système. Cependant chaque particule est modélisée par sa position dans l'espace de recherche et par sa vitesse. A chaque instant, toutes les particules ajustent leurs positions et vitesses, donc leurs trajectoires par rapport à leurs meilleures positions, à la particule ayant la meilleure position dans l'essaim et à leurs positions actuelles. En réalité, chaque particule est influencée, non seulement par sa propre expérience, mais aussi par celle des autres particules. La position et la vitesse d'une particule dans un espace de recherche à N dimensions sont définies par : $P_i = (P_{i,1}, \dots, P_{i,N})$ et $V_i = (V_{i,1}, \dots, V_{i,N})$ respectivement, chaque particule est caractérisée par sa meilleure position $L_i = (L_{i,1}, \dots, L_{i,N})$ à l'itération t . La meilleure position qu'atteint l'essaim est sauvegardée dans le vecteur $G = (g_1, \dots, g_N)$.

La vitesse de chaque particule est mise à jour selon l'expression suivante

$$V_{ij}(t + 1) = K \left[\omega \cdot V_{ij}(t) + c_1 \cdot r_1 (L_{ij} - V_{ij}(t)) + c_2 \cdot r_2 (g_j - V_{ij}(t)) \right] \dots \dots \dots (3.1)$$

$$K = \frac{2}{|2 - Q - \sqrt{Q^2 - 4Q}|} \dots \dots \dots (3.2)$$

La position à l'itération t+1 est alors calculée

$$P_{ij}(t + 1) = P_{ij}(t) + V_{ij}(t + 1) \dots \dots \dots (3.3)$$

$$Q = c_1 + c_2 \dots \dots \dots \text{et } Q > 4 \dots \dots \dots (3.4)$$

Dans ces expressions,

- ω est le facteur d'inertie qui contrôle l'influence de l'ancienne vitesse sur la vitesse courante, afin de permettre aux particules d'éviter les minima locaux.
- c_1, c_2 : sont des coefficients d'accélération tels que c_1 contrôle le comportement de la particule dans sa recherche autour de sa meilleure position et c_2 contrôle l'influence de l'essaim sur le comportement de la particule.
- r_1, r_2 : sont des nombres aléatoires uniformément distribués dans l'intervalle [0 1]

Ces particules parcourent l'espace et collaborent afin de trouver un optimum. Contrairement aux algorithmes génétiques, chaque particule possède une faible mémoire contenant le meilleur objectif trouvé et partage cette information avec ses particules voisines.

Le voisinage constitue la structure du réseau social. Il définit entre les particules où il existe un échange d'information. Il est rarement défini géométriquement. Cependant un voisinage social tend à devenir un voisinage géométrique. Le voisinage définit la topologie de l'essaim qui peut être sous différentes formes (fig.3.2):

- *Topologie en étoile* : le réseau social est complet, chaque particule est attirée vers la meilleure particule notée f_{best} et communique avec les autres.
- *Topologie en anneau* : chaque particule communique avec n voisines immédiates. On prend en général $n=3$ chaque particule tend à se déplacer vers le meilleur dans son voisinage local noté l_{best} .
- *Topologie en rayon* : une particule centrale est connectée à toutes les autres. Seule cette particule centrale ajuste sa position vers la meilleure, si cela provoque une amélioration et l'information est propagée aux autres.

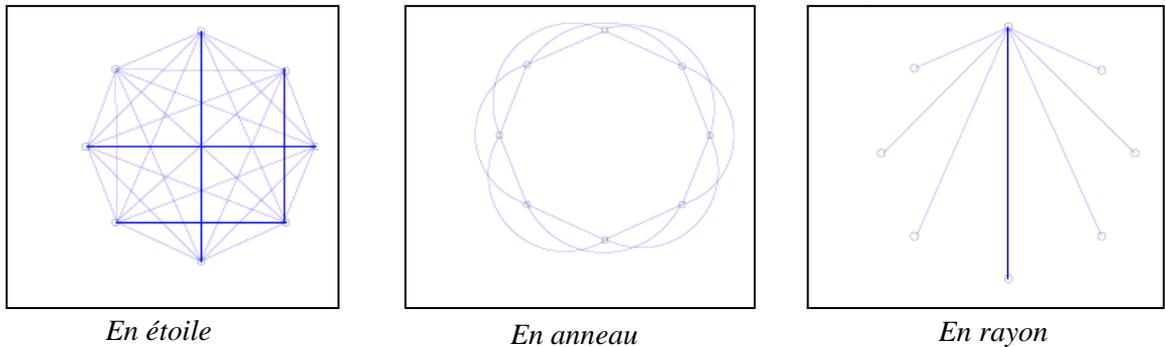


Fig.3.2 Topologie d'un voisinage social.

Le paramétrage est un outil qui sert à limiter la vitesse V_{max} pour éviter de sortir de l'espace de recherche

- L'ordre de grandeur est d'empêcher de traverser tout l'espace en une itération. Ce paramètre est sensible au problème d'échelle.
- De plus un bon réglage des paramètres assure une convergence de particules (fig.3.3).

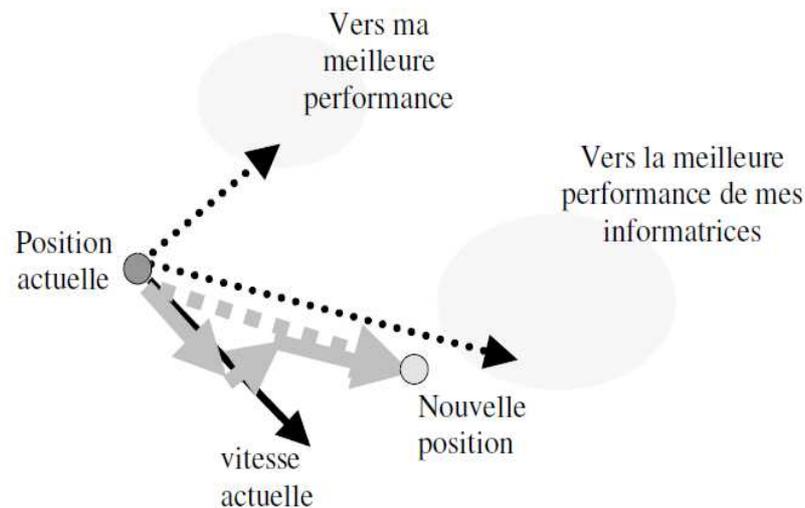


Fig.3.3 Schéma de principe du déplacement d'une particule [25].

Pour réaliser son prochain mouvement, chaque particule combine trois tendances à savoir suivre sa vitesse propre, revenir vers sa meilleure performance et aller vers la meilleure performance de ses informatrices.

3. Domaines d'application des PSO

Mêmes récents, l'algorithme d'essaim de particule trouve des applications dans divers domaines comme dans l'industrie pour, par exemple, la conception du profil d'une aile d'avion, le calcul de trajectoire pour le forage, la conception des filtres optiques et le timing d'ouverture des soupapes d'un moteur à explosion. Cet algorithme est aussi utilisé dans la segmentation d'image, la restauration d'image, qui constitue une étape très importante dans le processus de traitement d'images, alignement de photos satellites et la reconnaissance de suspects et utilisé aussi dans la télécommunication.

4. Classification par essaim de particules [26]

L'objectif principal des différentes méthodes de classification automatique est de répartir les éléments d'un ensemble en groupes homogènes, jusqu'à présent, un grand nombre de méthodes basées sur des techniques diverses ont été proposées et continuent de faire sujet de nouveaux travaux de recherche dans

le domaine. Quelque soit la technique utilisée, chacune de ces méthodes utilisées présente des avantages et des inconvénients par rapport au mode de fonctionnement et, surtout, par rapport à l'efficacité à séparer les classes en présence. La méthode avec laquelle nous allons faire une classification non supervisée des données brutes est celle l'essaim de particules (PSO) avec détermination du nombre de classes d'une manière automatique. Ce dernier est déterminé en utilisant la méthode de k-means qui réalise la classification des données. Puis une étape d'évaluation de cette dernière est effectuée à l'aide de l'indice de validation qui est le VCR. Ce processus est répété en changeant le nombre de classes, et la meilleure configuration est celle qui correspond à la valeur de VCR la plus petite. Après cette étape l'algorithme du PSO commencera l'exécution en prenant le nombre de classe celui qui correspond à la meilleure configuration trouvée. L'algorithme de PSO se compose d'un essaim des particules dans l'espace de recherche. La position de chaque particule est une solution potentielle au problème. La vitesse de chaque particule est modifiée sur sa distance de sa meilleure position personnelle et de la meilleure position globale. En d'autres termes les particules se déplacent selon leur expérience.

4.1 Choix de la fonction objectif

Pour juger de l'homogénéité des classes, il est nécessaire de disposer d'un critère de comparaison des objets (distance, inertie, moment, etc.). À partir de ce critère, on peut construire une partition. D'une manière générale, le signe d'une bonne classification est caractérisée par les valeurs d'indice de validation appelé indice VRC (Variance Ratio Criterion), qui permet d'obtenir la meilleure partition parmi toutes les partitions possibles de l'ensemble de données. Cet indice est basé sur la variance intra-classe, c'est-à-dire la dispersion des points dans chaque classe et la variance inter-classes [27]. Il est donné par l'expression suivante :

$$R = \frac{\text{dispersion inter-classe}}{\text{dispersion intra-classe}} \dots \dots \dots (3.5)$$

$$R = \frac{\sum_{k=1}^K \sum_{i=1}^{Q_k} \sum_{j=1}^M (x_i^j - g_j^k)^2}{\sum_{k=1}^K \sum_{j=1}^M (g_j^k - G_j)^2} \dots \dots \dots (3.6)$$

La valeur minimale de cet indice est le signe d'une bonne classification et le nombre K correspondant représente le nombre correct de classes.

4.2 Codification des individus

Dans notre cas, les particules doivent être des solutions au problème de classification, donc des partitions. La question est de savoir comment coder une partition, la codification consiste à représenter une partition par les centres de gravité de ses classes. Chaque particule de l'essaim est donc composée de k centres de gravité, où chaque centre de gravité est composé lui-même de m attributs réels.

4.3 Génération de la population initiale

Pour un essaim de particules, on génère aléatoirement les centres de gravité, chaque attribut soit une réalisation d'une loi normale centrée et l'écart type égal à celui de l'attribut.

4.4 Critère d'arrêt

Il y a plusieurs critères d'arrêt d'un algorithme basé sur l'essaim de particules tel que le nombre maximum d'itérations fixé à l'avance par l'utilisateur, le processus peut aussi être arrêté si la fonction objective ne s'améliore pas au bout d'un certain nombre d'itérations. Dans notre application, la procédure de recherche s'arrête et fournit la meilleure solution dès que le nombre maximal d'itérations est atteint. Les étapes de l'algorithme PSO pour la classification non supervisée (fig.3.4).

```
Début :
Initialiser toutes les particules avec k centres de gravités aléatoires chacune.
Pour nb=1 à max nb faire
  Pour toutes les particules i faire
    Pour tous les objets o faire
      Calculer la distance entre o et tous les centres de gravités.
      Affecter o à la classe dont le centre est le plus proche.
    Fin pour
  Calculer la fonction objective
Fin pour
Calculer la meilleure position locale pour chaque particule ainsi que la meilleure
position globale.
Mettre à jour la position et la vitesse des particules selon les formules des PSO.
Fin pour
Fin.
```

Fig.3.4 Algorithme PSO pour la classification non supervisée.

5. Classification par Essaims d'Abeilles Artificielles

Le comportement de fourragement des abeilles a été étudié de près et s'est révélé intéressant pour les problèmes d'optimisation [28]. Quand la nourriture est abondante et proche de la ruche, les abeilles circulent aléatoirement dans un petit rayon et s'attardent sur les sites où la nourriture est abondante et facile à collecter. Mais durant la saison de moissonnage, les abeilles doivent explorer une surface beaucoup plus grande, et c'est ce comportement qui se révèle intéressant pour l'optimisation. En effet, un certain nombre d'abeilles exploratrices sont envoyées explorer les endroits lointains. Puis à leur retour, chacune dépose une quantité de pollen et exécute une danse pour indiquer la direction et la distance qui sépare la ruche de l'endroit où elle a trouvé le plus de nourriture. La ruche choisit un certain nombre de ces sites à explorer par les abeilles, le nombre d'abeilles affectées à un site est proportionnel à sa proximité et à la quantité de nourriture qu'il contient. Si après la moisson d'un site par des abeilles, celui-ci contient encore beaucoup de nourriture, les abeilles exécutent encore une danse à leur retour et le nombre d'abeilles affectées à ce site augmentera encore [29].

Pour l'utilisation de cette méthode pour la classification non supervisée, on démarre avec n abeilles exploratrices. Chaque abeille est en réalité un ensemble

de k centres de gravités. Pour chaque abeille, on calcule la distance entre chaque donnée et chaque centre de gravité, la donnée est affectée à la classe dont le centre de gravité est le plus proche, une partition par abeille est ainsi construite. Puis les centres de gravité sont recalculés en fonction des partitions. On calcule la fonction de fitness de chaque abeille. On sélectionne les m meilleurs sites (représentés par les abeilles) et on leur affecte de nouvelles abeilles pour explorer le voisinage de ces m sites. Parmi ces sites, e sites sont appelés élites et ont beaucoup plus abeilles que les $m - e$ sites restants. Après cette recherche de voisinage, on sélectionne l'abeille qui a le meilleur fitness pour chaque site, on envoie les abeilles restantes faire une recherche aléatoire. A la fin, on sélectionne les n meilleures abeilles qui seront considérées comme la nouvelle population d'abeilles exploratrices, puis on réitère l'opération jusqu'à atteindre un critère d'arrêt. L'algorithme suivant résume cette approche.

<p>Etape1 : initialiser la population d'abeilles exploratrices. Etape2 : calculer la fonction fitness de chaque solution. Etape3 : sélectionner les sites pour une recherche de voisinage. Etape4 : affecter le plus d'abeilles aux sites d'élite puis calculer le fitness. Etape5 : sélectionner la meilleure abeille pour chaque site. Etape6 : envoyer les abeilles restantes faire une recherche aléatoire et calculer leur fitness. Etape7 : si le critère d'arrêt n'est pas atteint, aller à l'étape 2.</p>

Fig.3.5 Algorithme des abeilles pour la classification.

6. classification par algorithme des mouches [30]

L'algorithme des mouches est un algorithme évolutif particulier. Il présente les mêmes opérateurs de variation que les algorithmes évolutionnaires (croisement et mutation), et y ajoute deux autres : la *migration* et le *partage*.

La migration est un pourcentage de nouveaux individus créés aléatoirement qu'on introduit à chaque étape dans la population courante, ceci est utile à la diversité de la population, mais aussi si l'algorithme est dynamique. En effet, si la configuration ou la fonction objective évoluent en temps réel, cet opérateur permet d'ajouter de nouveaux individus qui vont prendre en compte cette

évolution, nous allons le voir avec un exemple concret dans l'application à la vision stéréo. Le partage, quant à lui, est un opérateur qui réduit la fonction fitness des individus se trouvant dans une zone trop dense (où il y a déjà beaucoup d'autres individus). Cet opérateur permet d'éviter que tous les individus ne convergent vers le même point.

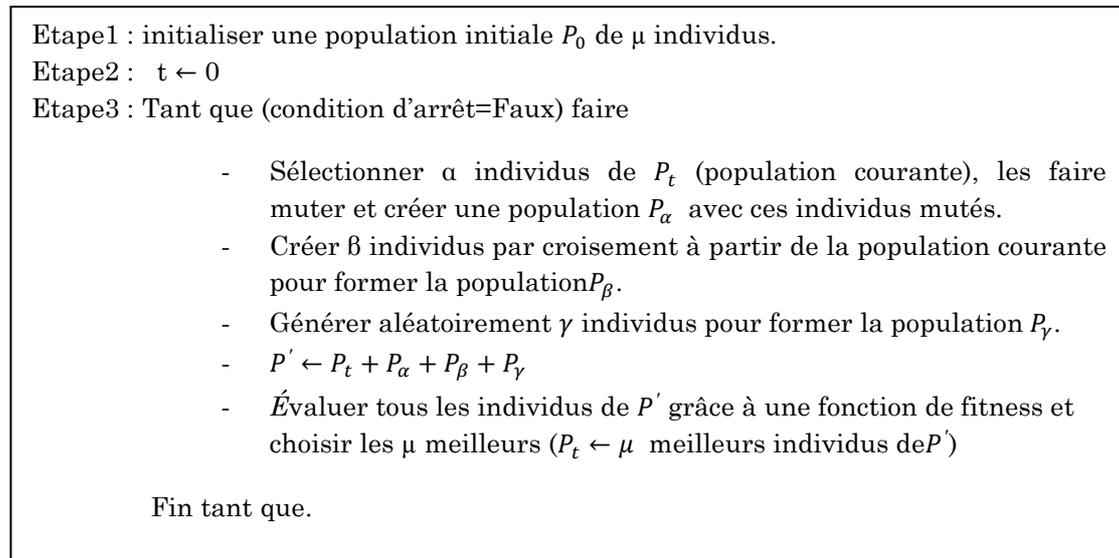


Fig.3.6 Algorithme des mouches.

Quand le critère d'arrêt est atteint, l'algorithme retourne la population P_t comme solution. α, β, γ et μ sont des paramètres de l'algorithme.

Cet algorithme a été utilisé jusque là uniquement en robotique. Leur principaux intérêts sont la représentation réduite des individus et une convergence rapide, d'où leur utilité et leur popularité en traitement temps réel que nécessite certaines applications robotiques. Nous allons voir qu'il est cependant possible de les utiliser pour résoudre le problème de classification non supervisée.

7. Hybridation des k-means avec les PSO

La segmentation d'image joue un rôle essentiel dans l'interprétation de divers genres d'images. Des techniques de segmentation d'image peuvent être groupées dans plusieurs catégories telles que la segmentation edge-based, la segmentation region-oriented, l'histogramme thresholding, et les algorithmes groupant (Gonzalez et Woods, 1992). Le but d'un algorithme groupant est

d'agréger des données dans des groupes tels que les données dans chaque groupe partagent les dispositifs semblables tandis que les classes de données sont distinctes de l'une à l'autre. L'algorithme k-means est une méthode largement répandue employée pour trouver la structure des données. L'algorithme PSO a été adapté pour la classification non supervisée par Omran et al en 2002. Les centres de classes assignés aux particules ont été initialisés aléatoirement. Chaque pixel a été distribué à la classe avec la distance euclidienne minimale. Alors PSO a été employé pour raffiner les centres de classes en utilisant une fonction fitness. Dans (van der Merwe et Engelbrecht, 2003) l'algorithme k-means était appliqué pour alimenter une particule de l'essaim initial et le reste de l'essaim ont été initialisés aléatoirement. Nous proposons une nouvelle hybridation d'algorithme de PSO-K means où k-means est appliqué à toutes les particules. L'algorithme proposé de PSO-K-means est présenté comme suit:

<p>Étape 1: Initialisez le nombre de classes à K et le nombre de particules à m.</p> <p>Étape 2 : Initialisez les ensembles de m de centres de classes aléatoires de K à employer par des m particules.</p> <p>Étape 3: Pour chaque particule, laissez chaque Pixel X appartenir à la classe dans lequel il a la plus petite distance euclidienne au centre de surface.</p> <p>Étape 4: Calculez les nouveaux centres de classes ; Si les nouveaux centres de classes convergent à les vieux, passez à la prochaine étape. Sinon, passez à l'étape 3.</p> <p>Étape 5: sauvegarder la meilleure solution trouvée jusqu'ici a exécuté par chaque particule. Appelez-la meilleure solution pbest ou personnelle.</p> <p>Étape 6: sauvegarder la meilleure solution parmi les meilleures solutions personnelles de m trouvées. Appelez-la meilleure solution gbest ou globale.</p> <p>Étape7: Mettez à jour les centres de classe de chaque particule selon les valeurs de centre de classe de la solution pbest et gbest, en utilisant Eqs. (3.1) et (3.3).</p> <p>Étape 8: Si le critère d'arrêt est satisfait allez faire un pas après. Sinon, passez à l'étape 3.</p> <p>Étape 9: Produisez la solution optimale.</p>

Fig.3.7 Algorithme PSO-K-means.

L'algorithme de PSO-K-means inclut également un ensemble de paramètres à déterminer empiriquement. Les paramètres ont été choisis comme suggérés par Van Der Merwe et Engelbrecht [31], ce qui a eu comme conséquence la bonne convergence. Ces paramètres ont été placés comme suit: $c1, C2 = 1,49$ et $w = 0,72$.

Nous avons examiné les algorithmes hybrides proposés et comparons les résultats à ceux de l'algorithme du K-means dans les figures ci-dessous. Les images utilisées incluent le flamant, les cubes, l'aurore et le fleuve. Le nombre de classes à trouver dans toutes les images est 3 excepté des cubes qui est 4, les résultats obtenus sont présentés. L'amélioration du PSO sur l'algorithme k-means est évident en tout des images examinées. Dans les cubes, le flamant et les images d'Aurore on peut voir que l'algorithme k-means a des résultats instables et dans certains cas il manque quelques classes tandis que l'algorithme PSO-K-means est plus stable et ils identifient clairement les classes. Dans l'image de fleuve les résultats montrent que l'algorithme k-means peut produire des résultats stables et semble être moins stable, mais il est évident que même pour que cette image étudiée plus loin le comportement des algorithmes décrits, nous avons obtenu le pourcentage d'exactitude de classification des résultats sur l'image de fleuve. Puis, en comparant les résultats de la classification avec les vraies données de base (fig.3.11d).

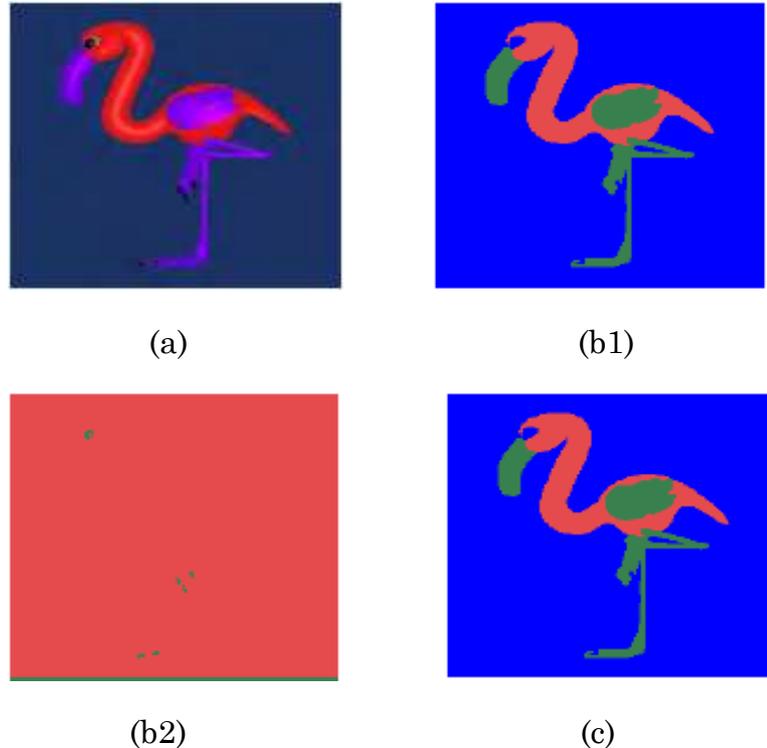


Fig.3.8 Les résultats obtenus ;(a) image originale, (b1et b2) K-means, (c) l'hybride PSO-K-means.

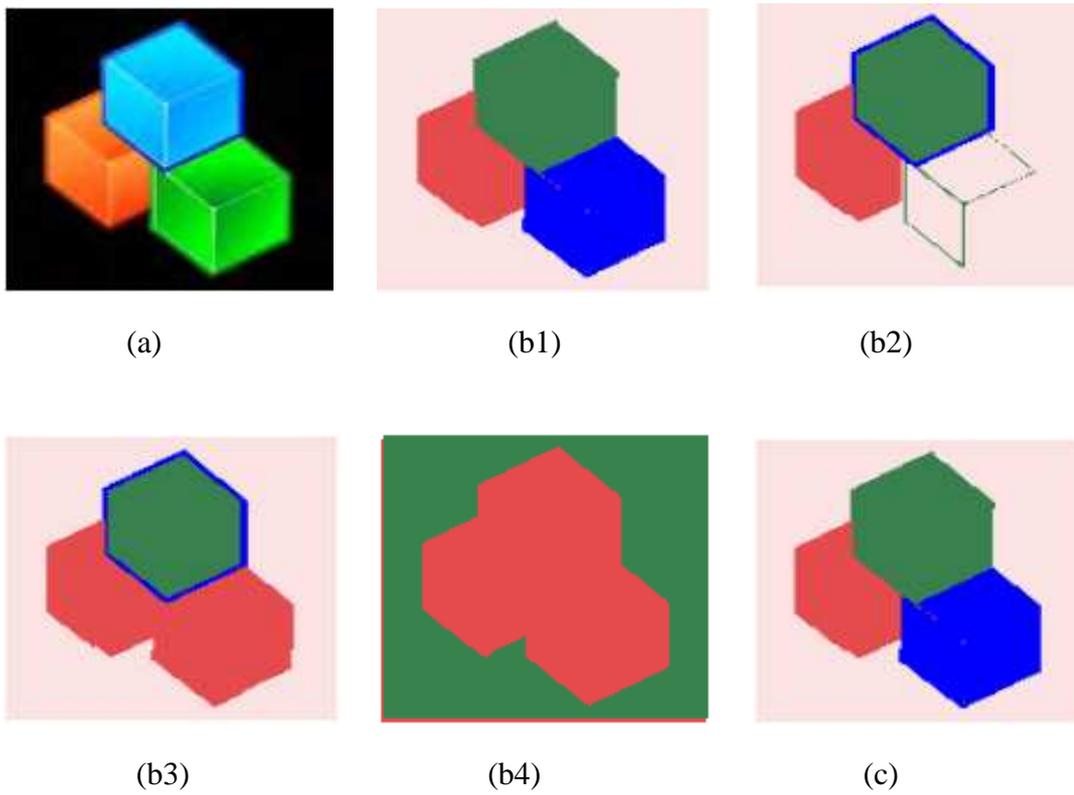


Fig.3.9 Les résultats obtenus ;(a) image originale, (b1, b2, b3, b4) K-means, (c) l'hybride PSO-K-means.

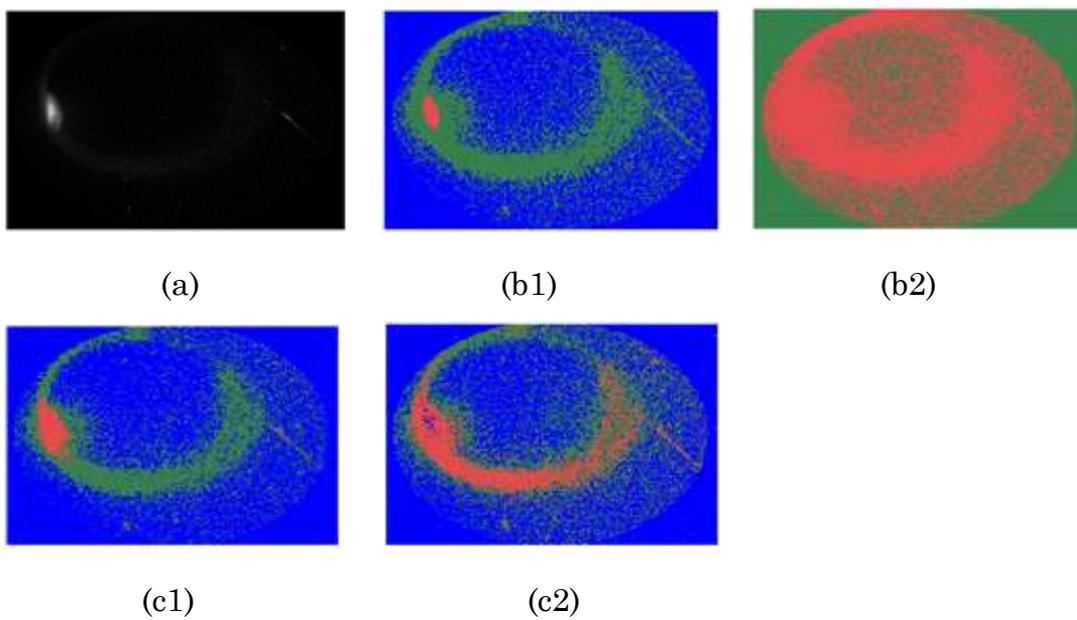


Fig.3.10 Les résultats obtenus; (a) Image originale, (b1, b2) K-means, (c1, c2) Hybridation PSO-K-means.

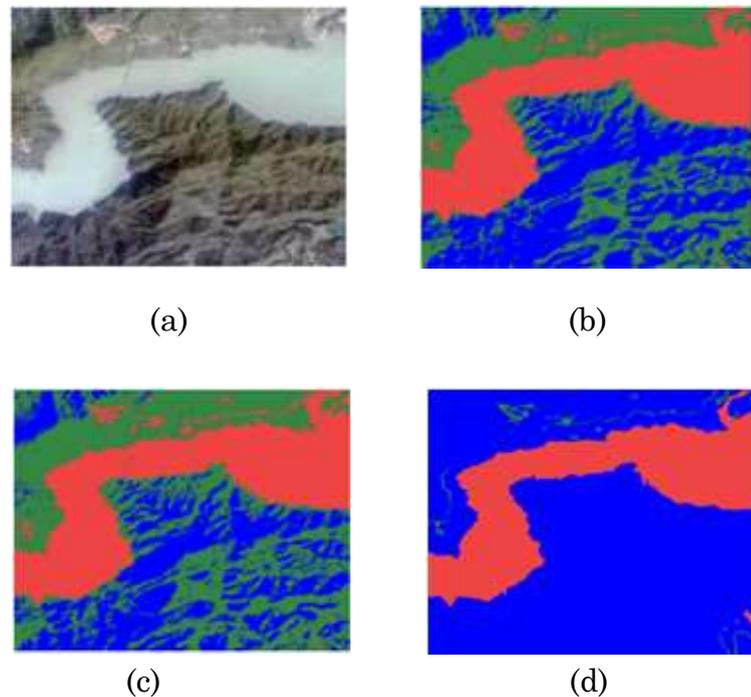


Fig.3.11 Les résultats obtenus; (a) image originale, (b) K-means, (c) Hybride PSO-K-means, (d) les vrais données de bases.

Les résultats expérimentaux ont prouvé que les techniques de Swarm Intelligence peuvent améliorer l'algorithme k-means en identifiant les classes. L'algorithme k-means souvent ne réalise pas les classes puisque ce dépend fortement des centres initiaux de classes. L'algorithme de PSO-K-means fournit un plus grand espace de recherche comparé à l'algorithme k-means. En utilisant ces algorithmes pour grouper, l'influence des centres initiaux incorrectement choisis de classe sera diminuée au-dessus d'un certain nombre d'itérations. Par conséquent, ces algorithmes dépendent moins des graines initiales aléatoirement choisies et sont pour trouver la solution optimale globale.

8. Conclusion

Dans ce chapitre nous avons fait un tour d'horizon sur l'algorithme du PSO appliqué à la classification non supervisée seul ou hybridé avec la méthode des k-means, il est apparu que cet algorithme est un outil très puissant pour l'optimisation. En plus du fait de s'inspirer de l'auto-organisation des insectes

sociaux, il dispose d'une base théorique solide qui permet son mise en application dans divers domaines. Pour cela, les chercheurs sont de plus en plus intéressés par ce type d'algorithme (algorithmes biomimétiques et particulièrement ceux mimant le comportement des insectes sociaux). L'approche multi-agents se révèle en effet très efficace et peu complexe à mettre en œuvre (le comportement d'un individu étant assez simple).

Conclusion générale

Dans ce mémoire, nous avons abordé un travail sur la classification non supervisée de données par les essaims de particules. L'étude est basée sur l'hybridation méthode du K-means et aux essaims de particules. La classification automatique et ses différentes variantes ont fait l'objet du premier chapitre, ensuite nous nous sommes intéressés à la méthode du K-means de manière plus formelle. Cette méthode présente l'avantage d'être rapide, simple et efficace, mais son inconvénient majeur est sa grande sensibilité aux conditions initiales. De plus, elle aboutit souvent à un optimum local.

Les métaheuristiques qui ont été développées sur la base d'observation de phénomènes physiques, biologiques et les comportements sociaux de certains insectes constituent un outil puissant d'optimisation, ont déjà été appliquées à la classification non supervisée à de nombreuses reprises. Ainsi, nous avons rappelé les différentes métaheuristiques qui sont représentées essentiellement par des méthodes de voisinage comme la recherche tabou, le recuit simulé et les méthodes de population comme les algorithmes génétiques, colonies de fourmis, systèmes immunitaires artificiels et les essaims de particules. Bien que les travaux utilisant ces métaheuristiques soient nombreux, nous avons vu quelques applications. Il nous est apparu que certaines améliorations étaient introduites. Nous avons également vu que les résultats du nouvel algorithme évolutionnaire utilisant la codification par anticorps se sont révélés globalement meilleurs que ceux d'un algorithme évolutionnaire utilisant des codifications classiques.

La méthode du PSO est considérée comme la métaheuristique qui a connu beaucoup de succès auprès de la communauté d'optimisation malgré son jeune âge. Cette métaheuristique est utilisée pour détecter les centres des classes qui optimisent mieux une fonction d'adaptation prédéfinie. Les tests réalisés ont montré l'efficacité et la puissance à classer des données et à segmenter des images couleurs. Cependant, cette méthode nécessite un choix judicieux des paramètres pour réaliser une bonne classification. Il est important d'effectuer

Conclusion générale

une étude comparative des résultats avec ceux obtenus avec d'autres métaheuristiques pour juger leurs efficacités et leurs performances avec les mêmes fichiers.

Enfin, ce travail nous a permis d'approfondir et d'améliorer nos connaissances théoriques et pratiques concernant le domaine de la reconnaissance de formes et de l'optimisation.

Références bibliographiques

1. M. DIAF, Méthodes de décision en reconnaissance des formes, en traitement d'image et en ophtalmologie, Thèse de doctorat d'état, spécialité Automatique, Université Mouloud Mammeri de TIZI OUZOU, 1994.
2. Törn, A.A., « Cluster analysis as a tool in a global optimization model», Proceedings of Third International Congress of Cybernetics and Systems, Bucharest , 1975, Springer Verlag, 249-260.
3. Diaf M. and K.K. Chopra, « A Mixed Method for Automatic Classification », Journal of Studies in Informatics and Control. vol.3, n°.1, pp.33-42, 1994
4. Belaïd. « Implémentation d'une méthode de classification non supervisée de données à l'aide d'algorithmes génétiques évolutionnaires ». Mémoire d'ingénieur. INI. 2008.
5. J. B. MacQueen. « Some methods for classification analysis of multivariate observations. » Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. Pages 281-297. University of California Press. 1967.
6. A.Lachkar, O. Ammor et N. Rais. « Détermination du nombre de classes par le principe du maximum d'entropie pour des classes en chevauchement. » Rapport de recherche. Universités de Meknès et de Fès, Maroc. 2006.
7. L. I. Kuncheva et J. C. Bezdek. « Selection of cluster prototypes from data by a genetic algorithm. » 5th European Congress on Intelligent Techniques and Soft Computing, Aachen, Allemagne. Pages 1683-1688. 1997
8. Petrowski J.A., P. Siarry and E.D. Taillard, « Métaheuristiques pour l'optimisation difficile » Eyrolles, 2003
9. Xia L., Xuehui L., & Jihong, Z. (2003, Dec.). Codebook design by a hybridization of ant colony with improved LBG algorithm. *Proceedings of the International Conference on Neural Networks and Signal Processing*, Vol.1, 2003 (pp. 469- 472). China : Shenzhen Univ.

Références bibliographiques

10. P. SIARRY, *les méthodes d'optimisation globale*. Ecole centrale Paris, France. Octobre 1997.
11. Dorigo, M., & Stützle, T. (2002). The Ant Colony Optimization metaheuristic: Algorithms, Applications and Advances. Kluwer Academic Publishers. Glover, F. & Kochenberger, G. (Eds.). *Handbook of Metaheuristics* (pp. 250-285). New York : Springer.
12. Johann Dréo – Patrick Siarry « métaheuristiques pour l'optimisation et auto-organisation, dans les systèmes biologique ».
13. Chelouah R. ; Siarry, P., "Tabu Search applied to global optimization", *European Journal of Operational Research*, vol. 123, pp. 256-270, 2000.
14. Patrick Siarry « les métaheuristiques et leur adaptation pour l'optimisation continue ». Université de paris 12, Val de Marne.
15. F Glover, M. Laguna «Tabu Search » Kluwer Academic Publishers, Boston, 1997.
16. S. KIRK PATRICK, C.D. GELATT, M.P. VECCHI, *Optimisation by simulated annealing*, *Science*, Vol. 220, n° 4598, 1983.
17. Moussa Diaf , Kamal Hammouche , Patrick Siarry From the real ant to the artificial ant: application in combinatorial optimization, data clustering, collective robotics and image processing. *Nature-Inspired Informatics for Intelligent : Applications and Knowledge Discovery: Implications in Business, Science and Engineering*, p.298-322. ISF - IGI Eds. USA, ISBN978-160566705-8, 2009, <http://www.info-sci-ref.com>
18. Shu-Chuan Chu et J. F. Roddick. « A clustering algorithm using the Tabu Search approach with Simulated Annealing. » *Data Mining II-Proceedings of Second International Conference on Data Mining Methods and Databases*. A Adelaïde, Australie. 2000.
19. Shi Y. & Eberhart R., Particle Swarm Optimization: Developments, Applications and Resources, *Proceedings of the Congress on Evolutionary Computation*, vol. 1, 2001, pp.81-86

Références bibliographiques

20. J. Kennedy et R. C. Eberhart. « Particle Swarm Optimisation. » Proc. IEEE Int'l Conf. on neuronal networks. Tome 4. Pages 1942-1948. 1995.
21. T. Ohira. « Immune pattern recognition system. » Proc. of International Workshops on Biologically Inspired Evolutionary Systems. Tokyo. Japon. 1995
22. M. Omran, A. Salman et A. P. Engelbrecht. « Image classification using Particle Swarm Optimization. » Conference in Simulated Evolution and Learning. Vol. 1, pages 370-374. 2002.
23. Abraham, S. Das et S. Roy. « Swarm Intelligence Algorithms for Data Clustering. » Soft Computing for Knowledge Discovery and Data Mining, Oded Maimon and Lior Rokach (Eds.), Springer Verlag, Allemagne. 2007.
24. Clerc, M., L'optimisation par essaim particulière : principe, modèles et usages. Techniques et Science Informatiques. Vol.21, 2002, p.941-964.
25. Venayagamoorthy G. K., Smith S. C. & Singhal G. (2007), Particle swarm-based optimal partitioning algorithm for combinational CMOS circuits, *Engineering Applications of Artificial Intelligence*, vol. 20, 2007, pp.177-184.
26. Clerc M. & Kennedy J. (2002), The Particle Swarm – Explosion, Stability and Convergence in a Multidimensional Space, IEEE Transactions on Evolutionary Computation, vol. 6, No. 1, Feb 2002, pp. 58-73.
27. E. DIDAY, J. LEMAIRE, J. POUGET, F. TESTU, *Eléments d'analyse de données*, Dunod, 1982. (IA471).
28. M. Fathian, B. Amiri et A. Maroosi. « Application of honey-bee mating optimization algorithm on clustering. » Applied math. and computation. Vol. 190, n°2, pages 1502-1513. 2007.
29. D. T. Pham, S. Otri, M. Muhmuddin, H. Al-Jabbouli et A. Haj Darwish. « Application of the bees algorithm to fuzzy clustering . » I-PROMS : Innovative Production Machines and Systems. 2008.

Références bibliographiques

30. P. Collet, E. Lutton et F. Rayna. « Polar IFS + Parisian Genetic Programming – Efficient IFS Inverse Problem Solving ». Genetic Programming and Evolvable Machines 1(4). Pages 339-361. 2000.
31. Swarm Intelligence, Focus on Ant and Particle Swarm Optimization Edited by Felix T. S. Chan and Manoj Kumar Tiwari Published by the I-Tech Education and Publishing, Vienna, Austria © 2007 I-Tech Education and Publishing. www.i-techonline.com.