Ministère de l'enseignement supérieur et de la recherche scientifique

Université de Mouloud Mammeri, Tizi-Ouzou

Faculté de Génie Electrique et Informatique

Département d'Automatique

MEMOIRE

De fin d'études

En vue d'obtention du diplôme d'ingénieur d'état

en Automatique

Thème

Hybridation de la méthode des k-means avec le recuit simulé

Proposé par : M^{r}M.Diaf

Présenté par : Allam Farida

Attab Saida

Devant le jury d'examen composé de :

Président : **M^{r}. MASSAR**

Promoteur : **M^{r}. DIAF**

Examinatrice : **M^{me}. KHERAZ**

Examinateur : **M^{r}. HAMASSE**

Promotion 2008-2009

Avant Propos

Le travail que nous présentons dans ce mémoire a été effectué an 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 **MESSAR Y.,** 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 Mme **KHERRAZ K.,** 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 Mr **AHAMASSE M.**, Enseignant 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, MM. **BENSIDHOUM M-Tahar** et **DIRAMI Ahmed** pour leurs disponibilités et leurs dévouements pour le bon fonctionnement du département.

Sommaire

Introduction générale ……...1

Chapitre1 : Le recuit simulé

1.1 Introduction…… ……3

1.2 Généralités sur l'optimisation…… …4

1.3 Les métaheuristiques……6

1.4 Le recuit simulé……11

1.4.1 recuit thermique…….……12

1.4.2analogie entre le recuit thermique et le recuit simulé…….14

1.4.3 Algorithme de recuit simulé……..16

1.5 Conclusion…….……21

Chapitre2 : la classification automatique

2.1 Introduction……22

2.2 Méthode de classification automatique……...23

2.2.1 Décision bayesienne……..24

2.2.2classification paramétrique supervisée…….……..26

2.2.3classification non paramétrique non supervisée …….……...26

2.2.4classification paramétrique non supervisée ……...27

2.2.5 Méthode non paramétrique non supervisée …….…….27

2.2.5.1Méthodes hiérarchiques…….……..28

2.2.5.2 Classification par le k plus proches voisins……..…28

2.3Méthode des k-means…….....29

2.4 Conclusion…….34

Chapitre3 Recuit Simulé et Classification Automatique

3.1 Introduction……..36

3.2 Métaheuristiques en Classification automatique……37

3.3 Hybridation k-means et Recuit Simulé……39

3.3.1 Rappels sur le Recuit Simulé ……...39

3.3.2 Rappels sur la méthode des k-means……40

3.3.3 Hybridation entre la méthode des k-means et le RS…….40

3.4 Tests et Résultats……..43

3.5 Conclusion……45

Conclusion générale…… ……...**46**

Références bibliographiques ……..**48**

Introduction générale

La classification est une étape importante pour l'analyse de données. Elle consiste à regrouper les objets d'un ensemble de données en classes homogènes. Autrement dit former les classes de telle sorte que les objets d'une même classe présentent des caractéristiques identiques mais différentes de celles des objets issus des autres classes.

Lorsque les méthodes utilisées pour la classification sont coûteuses en temps de calcul, on est conduit à les optimiser, puisque un problème de classification peut être vu comme un problème d'optimisation. A cette fin, on a recours à des heuristiques pour la résolution. Cependant, une heuristique ne donne pas forcément une solution optimale. Ceci a poussé les chercheurs à développer des concepts généraux capables de résoudre tous ces problèmes en obtenant un optimum global comme résultat au lieu de l'optimum local. Ceci a conduit au développement des métaheuristiques.

Les métaheuristiques sont des algorithmes stochastiques itératifs qui progressent vers un optimum par échantillonnage d'une fonction objectif. Chaque métaheuristique est inspirée de phénomènes naturels, comme la physique statique et l'évolution de comportement d'être vivants et souvent elle prend le nom du domaine ou elle a été inspirée. Parmi les métaheuristiques les plus connues de nos jours, on peut citer les algorithmes génétiques, la recherche tabou, les recuits simulés, les algorithmes de colonies de fourmis et, plus récemment, les essaims particulaires et les systèmes immunitaires artificiels. Elles sont applicables avec succès dans différents domaines. Parmi les contributions permettant l'amélioration de ces méthodes, on trouve, surtout, des hybridations avec des méthodes de conception différentes. Ceci semble bien donner des résultats encourageants qui privilégient ces pistes. Ainsi, dans ce mémoire composé de trois chapitres, nous consacrerons le premier à l'étude des métaheuristiques, en général, et à la méthode de recuit simulé, d'une manière détaillé Le deuxième chapitre sera consacré à l'une des

méthodes de classification automatique qui est la méthode des k-means Le troisième chapitre décrit l'hybridation de la méthode des k-means avec le recuit simulé suivie de tests et résultats

.

Enfin nous terminerons par une conclusion générale.

Chapitre 1

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 objectif 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 du recuit simulé, 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 1.1 présente les différentes méthodes d'optimisation.

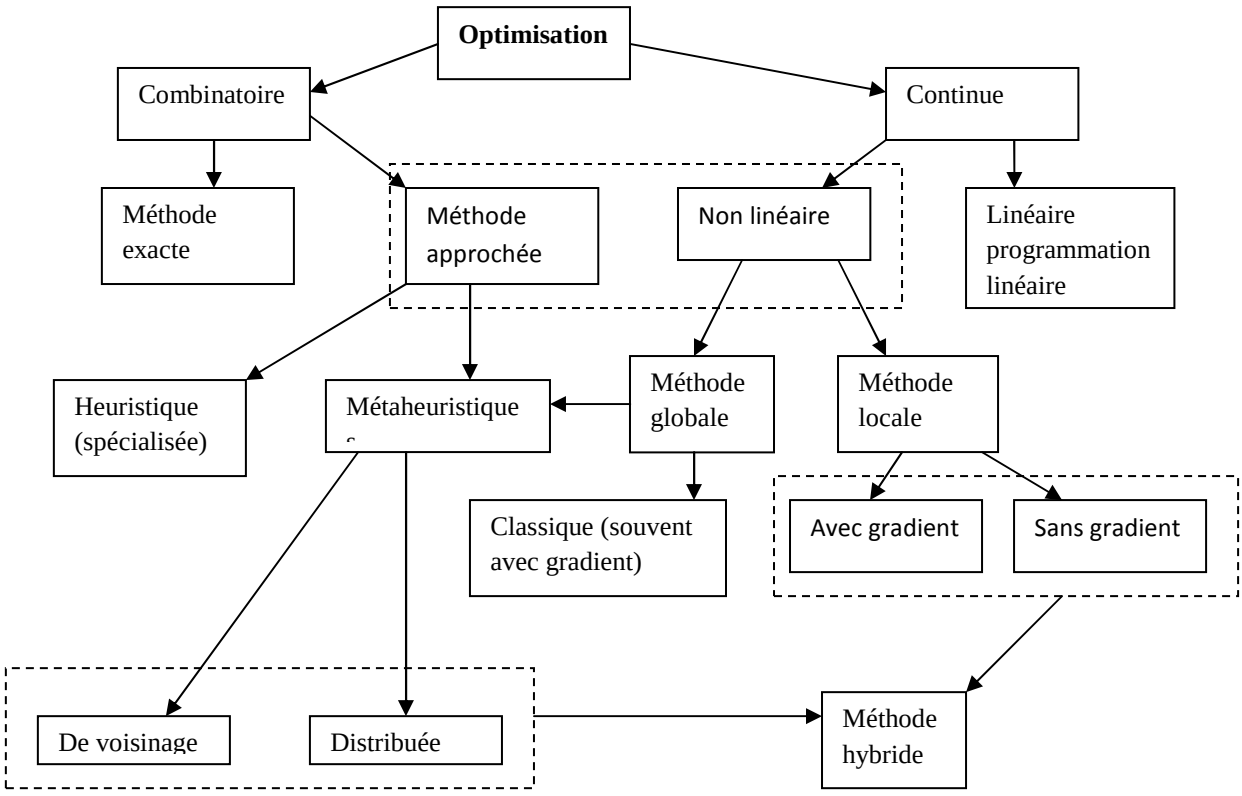

Fig.1.1 Présentation des différents algorithmes d'optimisation [1]

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 piégeage dans les optima locaux. Cependant, pour améliorer les performances d'une recherche, plusieurs auteurs hybrident les deux types d'algorithmes [2].

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: $X \rightarrow \Re \quad x \rightarrow f(x)$

où $X \varepsilon \sigma \tau$ Sur un ensemble de solutions $x / X \subset \Re$ ($\Re$est l'ensemble des réels). Ainsi, on a :

F(x^*) minimum local$\Leftrightarrow$[$\exists\, \varepsilon > 0 \,/\forall\, x \in X$: // $x - x^*$//<$\varepsilon \Rightarrow F(x) \geq$F ($x^*$)] 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 1.2 suivante, il s'agit de trouver l'optimum global G sans être piégé par les autres optima locaux tel que le point L [3].

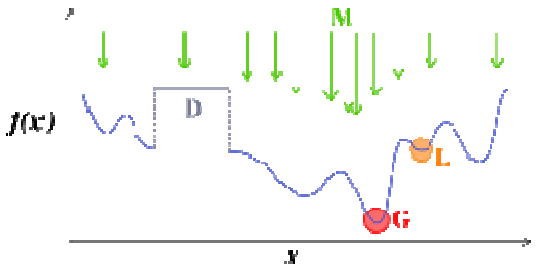

Fig.1.2 Représentation du minimum local et global d'une fonction

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

- 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.

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 essaims particulaires 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 (fig.1.3).

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 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

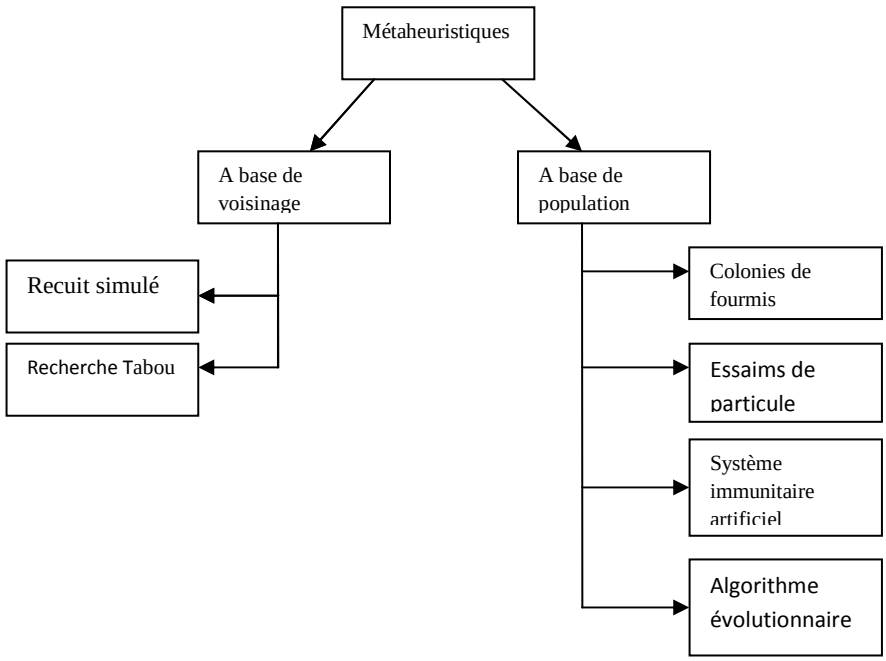

Fig.1.3 Les catégories des métaheuristiques

déclaré intangible [4] [5]. 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. Dés 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. Le recuit simulé, celui-ci, sera décrit dans le prochain paragraphe. 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 etc. Les algorithmes génétiques s'inspirent des principes de la génétique [6]. 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 inexistants 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 [7] [8]. 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 le 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 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. L'optimisation par essaims particulaires (Particle Sarm Optimisation) est issue d'une analogie avec les comportements collectifs de déplacements chez certains groupes d'animaux, comme les bancs de poissons, les nuées de oiseaux, les araignées etc. [9] [10]. 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. Notons que plusieurs autres métaheuristiques n'ont pas été cités dans ce mémoire et d'autres continuent encore de paraître.

4. Le recuit simulé

La méthode du recuit simulé (RS) (*Simulated Annealing* en anglais) est la métaheuristique la plus ancienne publiée par S. Kirkpatrick *et al.* en 1983 [11]. Elle est inspirée de la technique expérimentale du recuit thermique utilisée par les métallurgistes pour obtenir un état solide bien ordonné, d'énergie minimale, en évitant les structures métastables, caractéristiques des minimums locaux d'énergie. Cette métaheuristique permet d'éviter le piégeage dans un minima local. En effet, lorsqu'une fonction d'énergie présente un certain nombre de minima locaux, la tâche de trouver un minimum global devient compliquée. En utilisant, par exemple la méthode classique de descente de gradient dont le principe consiste, pour chaque itération, à faire un pas dans la direction de la plus grande pente vers les basses valeurs de la fonction, si, pour un état initial donné, l'algorithme arrive après quelques itérations à un minimum local, il n'y a plus de direction qui fait diminuer l'énergie et l'algorithme se trouve alors piégé. On peut procéder aussi à une descente stochastique qui, dans un ensemble de solutions, prend une solution au hasard et évalue son efficacité. La méthode évalue, ensuite, chacune des solutions avoisinantes. Si elles sont meilleures que la solution originale, alors celle-ci est choisie comme référence. On effectuera d'autres itérations autour de cette solution, jusqu'à arriver à un optimum local. Dans le RS, on part d'une descente stochastique, mais cette fois, on s'autorise de choisir des solutions

dont le score est moins intéressant que la solution courante, autrement dit, on prend des pas dans une mauvaise direction avec une faible probabilité. La différence que l'on s'autorise entre la solution retenue et la meilleure solution est fonction d'une température, à l'image de la température d'un métal en cours de refroidissement. Cette température diminue avec le temps. Lorsque cette température est suffisamment proche de 0 alors cela revient à terminer l'algorithme de descente stochastique. Le nom de cet algorithme ainsi que la plupart des termes utilisés comme température, équilibre statistique et gel du système sont tous empruntés au domaine de la métallurgie.

4.1 Recuit thermique

Le terme « recuit » utilisé dans l'industrie métallurgique est un processus qui, d'abord, porte la matière à l'état liquide en la réchauffant puis rabaisse la température lentement jusqu'à solidification, contrairement au traitement thermique ou tempe pour lequel le refroidissement est brusque (fig.1.4) [12]. Il est utilisé pour éliminer des impuretés présentes dans l'alliage en fusion et à partir desquelles des cristaux se forment. La forme et la taille de ces cristaux conditionnent les caractéristiques mécaniques de la pièce solide. Ainsi, chauffés au rouge, les métaux refroidis rapidement ou lentement présentent des états complètement différents. En effet, au départ, les atomes sont dans une structure cristalline solide. Lorsque le métal est porté à une certaine température élevée, les atomes se détachent de leurs liaisons et à se déplacent, détruisant, ainsi, la structure originelle du métal. Si la décroissance de température se fait de façon très brusque, on obtient une structure dure mais fragile sous forme de 'verre', caractéristique de la technique de 'trempe'. Si

le métal est refroidi lentement, le temps est laissé aux atomes d'atteindre l'équilibre statistique autrement dit, les atomes forment

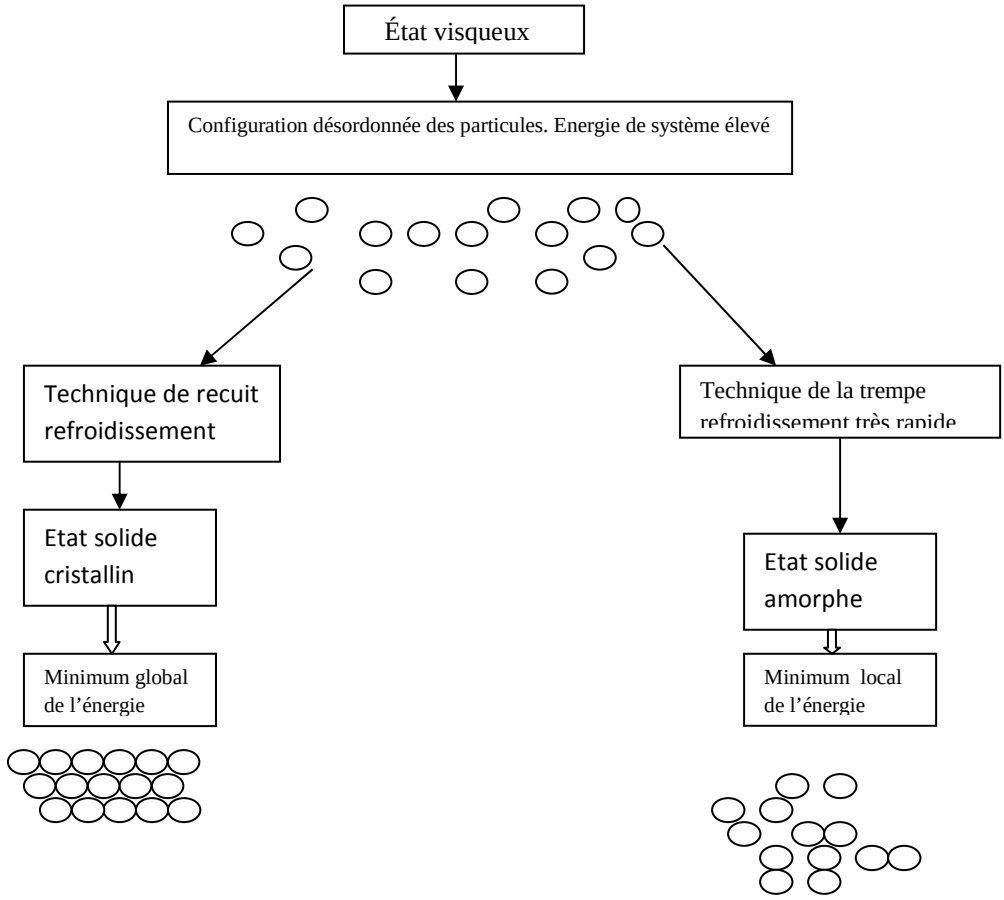

Fig. 1.4.Comparaison des techniques de recuit et de la trempe [13]

de nouvelles liaisons en se répartissant, souvent, de manière plus régulière et le métal devient plus souple, plus flexible et avec moins d'irrégularités conduisant à une structure plus stable. En effet, à l'équilibre thermodynamique à une température T, le temps que passerait le système dans une configuration microscopique d'énergie E est proportionnel à $e^{-\Delta E/K_b T}$. Autrement dit, les configurations d'énergie inférieure seraient favorisées par la nature. Il est dit aussi que la probabilité pour un système physique de posséder une énergie donnée E est proportionnelle à $e^{-\Delta E/K_b T}$où K_b est la constante de Boltzmann égale à $1.3805.10^{-23}$ L'exemple suivant est donné pour mieux comprendre l'effet du refroidissement lent. Imaginons les

atomes comme des petites sphères. Si on jette simplement ces billes dans un récipient, elles s'éparpillent de manière désordonnée. Pour les ordonner, on les secoue plus ou moins fortement, ce qui correspond à une température plus ou moins élevée. Au début, le désordre ne fait qu'augmenter, et les billes voltigent, car le fait de les secouer provoque un réchauffement. Mais si on les secoue de plus en plus lentement, elles s'ordonnent toutes seules. A partir de ce recuit thermique, la méthode du RS a été développée en procédant à une analogie entre les problèmes d'optimisation et ceux de la physique statique.

4.2 Analogie entre le recuit thermique et le recuit simulé

La méthode du RS a été donc inspirée du processus du recuit thermique en introduisant différentes analogies comme el montre la table I suivante.

Table I. Analogie entre des systèmes physiques et un problème d'optimisation [14]

Système physique	Problème d'optimisation
Bain de matériau à l'état liquide	Le problème
L'énergie libre E(X)	La fonction objectif F(X)
L'état stable	Optimum global
L'état métastable	Optimum local
Position des atomes	Configuration du problème
Déplacement d'un atome	Passage d'une configuration à l'autre
Trouver des états de basse énergie	Trouver une bonne configuration (voir la configuration optimale)

Ainsi, la configuration du problème d'optimisation correspond à la position des atomes, ce qui fait que, en optimisation combinatoire, il s'agit de chercher un échantillon parmi toutes les configurations par

application d'une transformation qui garantit le passage d'une configuration à une autre d'une manière aléatoire. La fonction objectif ou de coût du problème est analogue à l'énergie du métal. La température n'a pas d'équivalent en optimisation mais elle a le même effet que la température du système physique. Cependant, en optimisation combinatoire, cette température est un simple paramètre de contrôle de l'algorithme. La fonction objectif est minimisée par l'introduction de cette température qui doit conditionner le nombre d'états accessibles, c'est-à-dire qu'elle doit conduire vers l'état optimal si elle est abaissée de façon lente et bien contrôlée (technique du recuit), et vers un minimum local si elle est abaissée brutalement (technique de la trempe). Ainsi, le minimum local correspond à un état métastable obtenu après une trempe et le minimum global correspond à l'état stable obtenu après le recuit thermique. Puisque que la température n'a pas d'équivalent en optimisation, la distribution de Boltzmann n'est pas directement applicable dans ce cas. Elle est remplacée par le critère de Metropolis (Table II) qui est utilisé pour accepter des configurations de coût supérieur alors que le déplacement s'effectue normalement vers des configurations de coût décroissant. Ainsi, pour simuler l'évolution d'un système physique vers son équilibre thermodynamique à une température donnée T, on fait subir au système une modification élémentaire. Cette transformation a pour effet de diminuer la valeur de la fonction objectif du système. Elle est acceptée dans le cas où elle provoque une augmentation ΔE, *ie*. une diminution de la fonction objectif. Si, par contre, elle conduit à une diminution de ΔE, *ie*. une augmentation de valeur de la fonction objectif, elle est acceptée, toute de même, avec la probabilité $e^{-\Delta E/K_b T}$ Dans la pratique, ceci est réalisée en tirant au hasard un nombre réel compris entre 0 et 1. La fonction objectif est acceptée si

le nombre tiré est inférieur ou égal à $e^{-\Delta E/K_b T}$ En appliquant itérativement cette règle de Metropolis, on engendre une séquence

Table II. Règle de Metropolis**.**

```
 Si   F(X')≤ F(X),
Alors  conserver cette solution X' ; Faire X ← X' ;
Sinon
    Calculer P = Exp ( - ( F(X') – F(X) ) / T ) ;
    Générer un nombre aléatoire ( d'une distribution uniforme ) R compris entre 0 et 1 ;
          Si  R ≤ P,
          Alors
               Accepter la nouvelle solution X' ; Faire  X ← X' ;
          Sinon
               Refuser la solution X' .
```

de configurations qui constitue une chaîne de Markov, en ce sens que chaque configuration ne dépend que de celle qui la précède immédiatement.

Grâce à ce formalisme, on montre que lorsque la chaîne est de longueur infinie (en pratique de longueur suffisante) le système atteint (en pratique se rapproche de) thermodynamique à la température considérée.

Une fois l'équilibre thermodynamique est atteint à une température donnée, on abaisse légèrement la température et on effectue une nouvelle chaîne de Markov à ce nouveau palier de la température. Lorsque la température tend vers zéro, l'algorithme converge vers le minimum absolu de l'énergie. Dans la pratique, lorsque le processus est stoppé dès que le système est figé, *ie.* la température à atteint la valeur nulle ou bien plus aucun mouvement accroissant l'énergie n'a été accepté au cour du palier.

Le déroulement de l'algorithme du RS est décrit ci-après.

4.3 Algorithme du Recuit Simulé

Le RS cherche donc à imiter le processus du recuit thermique en suivant l'algorithme de la table III.

L'idée du RS en optimisation combinatoire est de simuler numériquement une opération de recuit thermique. Le principe en est décrit ci-après.

Table III. Algorithme du RS [15].

1 Choisir aléatoirement, une solution initiale X du système à optimiser et évaluer la valeur de la fonction objectif F = F (X).
2 Choisir une température initiale élevée T.
3 Perturber cette solution pour obtenir une nouvelle solution X' = X+ΔX ;
4 Calculer ΔF = F(X') – F(x) ;
5 Si ΔF ≤ 0,
Alors accepter la solution X'; Faire X ← X' ;
Sinon calculer P = exp (-ΔF/T) ;
Générer un nombre aléatoire R compris entre 0 et 1 ;
Si R ≤ P,
Alors accepter la nouvelle solution X' ; Faire X ← X' ;
Sinon refuser la solution X ;
6 Sauver le meilleur point rencontrer ;
7 Si l'équilibre thermodynamique du système à la température T est atteint,
Alors abaisser légèrement la température T ;
Sinon aller à l'étape 3 ;
8 Si le système est figé,
Alors aller à l'étape 9 ;
Sinon aller à l'étape 3 ;
9 Solution = meilleur point trouvé. Arrêt du programme.

Soit un système à optimiser composé de N éléments et avec un grand nombre d'optima. A chaque configuration du système est associée une fonction F à optimiser. Cette 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 Si l'on recherche la configuration qui minimise la fonction, celle-ci joue le rôle de l'énergie. Si l'on recherche au contraire un maximum, alors on prend comme énergie l'opposée de la fonction. Les configurations du

système peuvent être modifiées par des changements d'états discrets des composants du système.

Initialement, on part d'une configuration aléatoire ou choisie astucieusement en fonction du problème. Au départ de l'itération, on fixe un paramètre de température en relation avec la gamme des énergies accessibles au système. La procédure opère donc sur une solution initiale S_i générée et avec une température T_0 comme une étape initiale. Ensuit. On se déplace vers une nouvelle solution Sj engendrée dans son voisinage de Si défini V(Si). Si F(Sj) ≤ F(Si) alors la solution Sj est acceptée et peut constituer une comme solution du départ pour la prochaine étape. Dans le cas contraire, Sj est acceptée quand même avec la probabilité avec égale à $e^{-\Delta E/T}$ avec ΔE=F(Sj)-F(Si). Ceci est réalisé pratiquement en tirant au hasard un nombre réel compris entre 0 et 1. La solution qui permet de réduire la fonction objectif est acceptée si le nombre tiré est inférieur ou égal à $e^{-\Delta E/T}$ Cette règle de Metropolis appliquée itérativement engendre une suite qui constitue une chaîne de Markov, de sorte que chacune des configurations ne dépende que de celle qui la précède immédiatement. Lorsque la chaîne est de longueur suffisante, le système se rapproche de l'équilibre thermodynamique à la température considérée. La température est ensuite abaissée légèrement suivant une loi de décroissance et une nouvelle chaîne de Markov est engendrée à ce niveau de palier de température. Lorsque la température tend vers zéro, l'algorithme converge vers le minimum absolu de l'énergie. Dans la pratique, le processus est stoppé lorsque la température aura atteint la valeur nulle autrement dit, lorsque, aucune autre configuration qui permet d'accroître l'énergie n'est acceptée durant le palier. A ce niveau, le système est

dit figé. Les paramètres du RS peuvent être fixés de la manière suivante :

Configuration initiale

Initialement, on part d'une configuration aléatoire ou choisie astucieusement en fonction du problème. Le RS est une méthode qui améliore cette solution initiale.

Température initiale T_0

La température est donc un paramètre de contrôle qui doit être suffisamment grande pour ne pas oublier des minimas très proches de cette température. Cette température est diminiée lentement pour permettre au système de rechercher les "bassins d'attraction" dont la préférence est toujours donnée à celui dont le coût est minimal. Cette décroissance est effectuée selon une loi. Cette loi joue un rôle important car le maximum de configurations doit être exploré.
Il existe plusieurs manières de fixer cette température initiale. Une façon de choisir cette valeur consiste à générer un certain nombre de perturbations et à calculer leur moyenne ΔF ainsi que la probabilité d'acceptation *P,* très proche de 1. Ainsi, la température initiale T_0 est déduit de la formulé suivante : $T_0 = \Delta F / Ln(1/P)$.

Décroissance de la température

Il existe plusieurs méthodes de décroissance de la température (fig.1.5). La décroissante peut toutefois être fixée selon la loi géométrique qui a l'avantage d'être simple. Cette décroissance utilise, dans ce cas, la formule suivante $T_{k+1} = \alpha T_k$ où α est un facteur (taux de décroissance) de refroidissement choisie dans l'intervalle [0.5, 0.99]. Par défaut, le coefficient de décroissance vaut

0.9 mais peut être modifié dans le dialogue des paramètres du recuit.

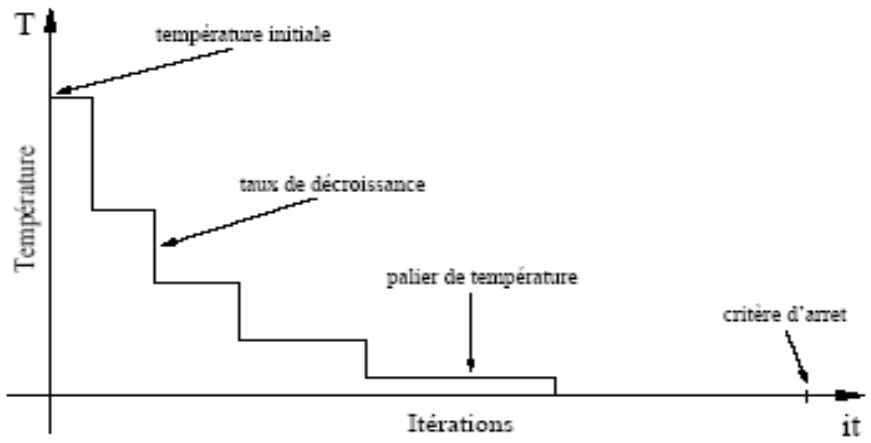

Fig.1.5 Paliers de la température [16]

Changement de pallier de température.

Ceci détermine le moment où l'on va changer de palier de température, c'est à dire le nombre d'itérations que l'on va faire à la même température. ce changement peut s'opérer dés que l'une des conditions suivantes est satisfaites au cours du palier de température, à savoir, 12.N perturbations acceptées ou 100.N perturbations tentées où N désignant le nombre de paramètres du problème. Par défaut, on prend $T_{k+1} = 0{,}9\ T_k$.

Arrêt du système

Ceci signale la fin du traitement ou l'arrêt de l'algorithme. Il correspond au fait qu'aucune autre transformation n'est acceptable. Pour ce faire, il différentes approches comme par exemple lorsque la température est proche de la valeur zéro ou lorsqu'on fixe, par avance, le nombre de paliers de température.

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.

Parmi les métaheuristiques, le RS a prouvé son efficacité à résoudre les problèmes difficile non résolus avec les autres méthodes d'optimisation classique. Il a suscité de nombreux travaux théoriques et pratiques. Parmi les applications, la première a été développe dans le domaine placement de circuit électronique, en traitement d'images, ans l'organisation de réseau informatique du Loto, l'optimisation de la collecte des ordures ménagères et l'organisation d'emploi du temps.

Pour notre part, nous allons contribuer à l'application du RS dans le domaine de la classification automatique.

Chapitre 2

Classification automatique

1. Introduction

Les méthodes de classification automatique visent à répartir des objets décrits par un certain nombre de paramètres en classes homogènes. Les données d'une classification automatique se représentent souvent sous forme d'un tableau objets/caractères. En classification automatique, il existe un grand nombre de méthodes utilisant des notions mathématique très diversifiées. Ainsi, l'utilisateur est souvent confronté au type de méthode à utiliser.

Dans le cas où nous avons toutes les informations en termes de probabilités *a priori* sur les objets, l'application du théorème de Bayes permet de transformer ces probabilités *a priori* en probabilités *a posteriori*, c'est à dire à trouver la classe la plus probable.

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*.

Aujourd'hui, on s'oriente de plus en plus vers des méthodes non paramétrique et non supervisées ou des méthodes de recherche de groupements ou "clustering" [17]. Dans la seconde partie de ce chapitre, nous présenterons succinctement les différentes catégories de la classification automatique. Ensuit, nous présenterons, plus en détail, la méthode des K-means. Ce chapitre sera terminé par une conclusion.

2. Méthodes de classification automatique

La classification est la base de toute connaissance. Il s'agit d'une démarche très courante qui permet de comprendre l'ensemble d'objets à analyser.

La classification, en générale, est définie comme étant l'action de construire une collection d'objet similaires au sein d'un même groupe, dissimilaires quand ils appartiennent à des groupes différents.

Mathématiquement la classification est définie de la manière suivante :

Soit $X=\{x_1, x_2, x_3,\ldots, x_N\}$ l'ensemble des N observation à classer.

Chaque observation x_i est caractérisée par P paramètres.

Soit $C=\{C_1,C_2 ,C_3,\ldots,C_K\}$ l'ensemble de K classes.

La classification consiste à repartir l'ensemble des N observation en K classes de sorte que :

$C_i \neq \emptyset$, i=1,…, K (aucune classe ne doit être vide).

$C_i \cap C_j = \emptyset$, $i \neq j$ (aucun recouvrement entre les classes).

$\coprod\{x_i \in C_k\} = \{X\}$, $i \in \{1,2,\ldots,N\}$ et $k = 1,2,\ldots,K$, l'union des éléments de chaque classe doit être égale à X, le nombre d'observations.

Après la classification, les classes obtenues doivent être compactes et séparables ou éloignées les unes des autres.

Dans l'utilisation d'une classification automatique, généralement,

- on identifie le type de données qui peuvent être qualitatives ou quantitatives,
- on choisit la méthode classification automatique selon le nombre de données, le type de variables et les connaissances *a priori* sur ces données.
- on analyse les résultats obtenus par leur interprétation, leur évaluation etc.

2.1 Décision bayesienne

La décision bayésienne permet de reconnaître si un vecteur de forme x de R^n appartient à une classe parmi les K classes de l'ensemble $C=\{w_i\}$, i=1,2,…,K. Ce vecteur x est supposé aléatoire dont on connaît la densité de probabilité

conditionnelle $p(x/w_i)$ d'avoir x sachant que la forme décrite par le vecteur appartient à la classe w_i. La règle de décision de Bayes permet de déterminer la meilleure classe $w_i(x)$ ou la classe la plus probable en appliquant la formule suivante qui est très pratique du fait que l'on connaît souvent la probabilité à posteriori p(x/wi):

$$\omega^*(x) = w_j \ / \ \vee i, 1 \le i \le K, p(x/w_j)p(w_j) \ge p(x/w_i)p(w_i) \quad \ldots\ldots\ldots (1)$$

On notera de plus que:

$$p(w_i/x) = \frac{p(x/w_i)p(w_i)}{\sum_{i=1}^{K} p(x/w_i)p(w_i)} \quad \ldots\ldots (2)$$

La règle de décision bayésienne nécessite la connaissance de probabilités *a priori* conditionnelles qui ne sont pas directement accessibles à l'expérience. De plus, dès que le nombre de valeurs que peut prendre la variable X devient élevé, l'estimation avec précision de p(x) et $p(x/w_i)$ en chaque point X est impossible.

Dans de telles situations, on est appelé alors à découper l'espace des mesures en blocs pour ensuite appliquer le théorème de Bayes à chacun de ces blocs [18]. Il faut toutefois noter que si le nombre de blocs est trop grand, les quantités p(x) et $p(x/w_i)$ ne peuvent être estimées avec précision. Dans le cas contraire, le bruit de discrétisation imposé aux mesures risque d'introduire des erreurs du fait que les blocs sont très hétérogènes. C'est d'ailleurs pour cette raison que de nombreuses méthodes essaient d'éviter la discrétisation et considèrent l'espace continu.

2.2-Classification paramétrique supervisée

Si l'on ne possède aucune information sur les différents objets à classer, on cherche généralement une modélisation qui obéit à une loi statistique déterminée mais souvent prise gaussienne puisque cette loi suit beaucoup d'exemples naturels de distribution. La classification sera de bonne qualité si effectivement la distribution des objets suit la loi choisie. Rappelons la formule qui définit la fonction de densité de probabilité multivariable associée à une distribution gaussienne :

$$p(X) = \frac{1}{(2\pi)^{\frac{m}{2}} |R|^{\frac{1}{2}}} \exp^{-\frac{1}{2}(X-\overline{X})^T R^{-1} (X-\overline{X})} \quad \text{.....(3)}$$

où X est le vecteur moyenne des vecteurs X et R la matrice de covariance associée à ces vecteurs. Cette matrice est réelle, symétrique et définie positive. !R! est le déterminant de R et m la dimension de l'espace de représentation des observations.

2.3 Classification non paramétrique non supervisée

Dans les méthodes précédentes, on considère la forme des densités de probabilité a priori connues alors que dans la pratique, les observations ne suivent pas souvent ces distributions. De ce fait, les méthodes non paramétriques évitent ce choix a priori en estimant les densités $p(X/w_k)$ à partir des prototypes [19].

L'estimation directe d'une densité de probabilité repose sur le fait que la probabilité pour qu'un objet appartienne à un domaine D est donnée par l'expression:

$$P = \int_{D} p(X)dX$$

Cette formule montre que l'on peut estimer p(X) en estimant P, la probabilité pour que l'objet X appartienne au domaine X. Comme exemple, on peut citer les fenêtres de Parzen.

2.4 Classification paramétrique non supervisée

Les méthodes paramétriques non supervisées ressemblent aux méthodes paramétrique supervisées à la seule restriction que l'on ne dispose pas de prototypes à priori. Parmi ces méthodes, on peut citer l'estimation par le maximum de vraisemblance [20], l'estimation des densités de probabilités composées, l'estimation de la moyenne par apprentissage, etc. Avec ces méthodes, on risque d'imposer aux données une structure particulière sans trouver leur véritable structure. On leur préfère alors les méthodes non paramétriques pour détecter les régions de forte densité des points.

2.5 Méthode non paramétrique non supervisée [21]

Sans information à priori sur la nature des classes, ces méthodes non paramétriques et non supervisées sont très nombreuses et englobent particulièrement les méthodes hiérarchiques, les méthodes agrégatives, les "clusterings" et les méthodes permettant la détection des modes par calcul du gradient, la détermination de la concavité ou convexité des fonctions de densité de probabilité etc...

2.5.1 Méthodes hiérarchiques

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. 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 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'effectif faible pour se prêter à une partition sûre[22]. Dans notre cas, c'est la dispersion intraclasses qui nous a servi de critère de division.

2.5.2 -Classification par le k plus proches voisins

Trois étapes distinctes composent cette méthode non supervisée et non paramétrique. La première étape consiste à recenser pour chaque observation les k observations qui lui sont les plus proches. L'algorithme de Kittler basé sur la distance euclidienne a été utilisé. A la seconde étape, pour

chaque observation x_i, on calcule l'estimateur. A la troisième étape de la procédure, on applique le filtre médian pour raffiner davantage les estimateurs pour que les points situés dans un proche voisinage aient des estimateurs semblables pour aboutir à la bonne classification.

3. Méthode des k-means

La méthode d'agrégation autour des centres mobiles est la méthode la plus utilisée de par sa simplicité et la qualité des résultats qu'elle offre. On la retrouve sous d'autres appellations ou d'autres variantes à savoir la méthode des k-means, des c-means, la méthode ISODATA, abréviation de Itérative Self Organizing Data Analysis[23], la méthode des fussy c-means utilisant la logique floue, Ces algorithmes se ressemblent dans le principe général, mais différent dans la façon de procéder pour aboutir à une partition finale. Elles procèdent par groupement de données selon une distance comme la distance euclidienne. Chaque classe est représentée par son centroïde. La procédure de partitionnement s'effectue par rapport aux centroïdes des classes. La méthode d'agrégation autour des centres mobiles ou k-means à laquelle nous nous intéressons particulièrement, permet de construire une partition finale qui est censée être la meilleure à partir d'une partition initiale qui peut être fixée par l'utilisateur d'une manière aléatoire. Le choix de cette partition initiale reste toutefois délicat. Cependant, il existe des méthodes qui permettent la déterminer automatiquement. Comme nous le verrons plus bas, les classes obtenues peuvent caractérisées à l'aide des moments interclasse et interclasse qui nous

renseignent sur la qualité de la partition finale. On se réfère, donc, au moment d'ordre deux d'un ensemble I de N points par rapport à leur centre de gravité G. Ce moment est donné par la formule générale suivante :

$$M^2(I,G) = \sum_{i=1}^{N} m_i d^2(i,G) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

où m_i sont des masses ponctuelles ou des pondérations, et $d(i,G)$ les distances des points i au centre de gravité. Lorsque les observations sont décrites par M variables (espace à M dimensions), ces distances, lorsqu'elles ont euclidiennes, sont données par la formule suivante:

$$d^2(I,G) = \sum_{j=1}^{M} (x_{ij} - xG_j)^2 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Ainsi l'expression donnant le moment centré d'ordre 2 devient :

$$M^2(I,G) = \sum_{i=1}^{N} \sum_{j=1}^{M} m_i (x_{ij} - xG_j)^2 \ \ldots\ldots\ldots (3)$$

Dans le repère à M dimension, la j^{ème} coordonnée du centre de gravité G du nuage des N observations est donnée par:

$$xG_j = \frac{\sum_{i=1}^{N} m_i x_{ij}}{\sum_{i=1}^{N} m_i} \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

Si les observations ne sont pas pondérées, (m_i=1), nous aurons :

$$xG_j = \frac{1}{N} \sum_{i=1}^{N} x_{ij} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

et

$$M^2(I,G) = \sum_{i=1}^{N} \sum_{j=1}^{M} (x_{ij} - xG_j)^2 \ \ldots\ldots\ldots\ldots (6)$$

Ainsi, le moment centré d'ordre deux $M^2(I,G)$ correspond à l'écart type d'une distribution gaussienne. L'écart type étant un paramètre de dispersion, on remarque que si les points sont très concentrés autour du centre de gravité, la valeur de $M^2(I,G)$ devient faible. Dans le cas où ces points sont très dispersés, $M^2(I,G)$ prend une valeur plus élevée.

Le moment centré d'ordre deux peut ainsi nous renseigner sur la qualité des classes obtenues en classification.

Le nuage *I* peut être décomposé en *L* sous-nuages ou sous-ensembles disjoints *In* n=1,2,...L. Chaque sous-nuage *In* correspond à un nombre K_n d'éléments tel que K_n <= N, n=1,2,...L.

Pour chaque sous-nuage I_n, n=1,2,...,L, on calcule le moment centré d'ordre deux. En notant G_n les barycentres de ces sous-nuages, le moment du sous-nuage In par rapport à son centre de gravite G_n est calculé par la formule suivante:

$$M^2(I,G) = \sum_{i=1}^{Kn} \sum_{j=1}^{M} (x_{ij} - xG_{nj})^2 \dots\dots\dots\dots\dots\dots\dots\dots (7)$$

Par rapport au barycentre G de l'ensemble des points ou du nuage global I_n, ce moment est:

$$M^2(I_n,G) = M^2(I_n,G_n) + M_n d^2(G_n,G) \dots\dots\dots\dots\dots\dots\dots (8)$$

En appliquant le théorème de Huyghens pour le sous-ensemble I_n, on obtient:

$$M^2(I,G) = \sum_{n=1}^{L} \sum_{i=1}^{Kn} \sum_{j=1}^{M} \left(x_{ij} - xG_{nj}\right)^2 + \sum_{n=1}^{L} \sum_{j=1}^{M} M_n \left(xG_{nj} - xG_j\right)^2 \dots (9)$$

Le terme $\sum_{n=1}^{L} \sum_{i=1}^{Kn} \sum_{j=1}^{M} (x_{ij} - xG_{nj})^2$ représente la dispersion intraclasse et $\sum_{n=1}^{L} \sum_{j=1}^{M} M_n (xG_{nj} - xG_j)^2$ représente la dispersion interclasses.

La formule donnant $M^2(I,G)$ montre que le moment centré d'ordre deux du nuage *I* est la somme des moments centrés d'ordre deux des sous-nuages I_n augmentée du moment interclasse. En d'autres termes, la dispersion totale est décomposée en dispersion à l'intérieur des sous-ensembles ou des classes.

La dispersion interclasse est de valeur élevée si les barycentres des classes formées sont éloignés du barycentre du nuage. Quant aux dispersions intraclasses, elles sont de valeurs faibles si les points constituant chaque classe sont très rapprochés du centre de gravite correspondant. En classification automatique, on cherche à minimiser la dispersion intraclasse et à maximiser la dispersion interclasse.
En effectuant le rapport entre le moment intraclasse et le moment interclasse, on peut constituer l'indice R qui est le rapport entre ces deux moments Cet indice R est calculé par la formule suivante :

$$R = \frac{\sum_{n=1}^{L} \sum_{i=1}^{Kn} \sum_{j=1}^{M} (x_{ij} - xG_{nj})^2}{\sum_{n=1}^{L} \sum_{j=1}^{M} M_n (xG_{nj} - xG_j)^2} \ldots\ldots\ldots\ldots (10)$$

où L est le nombre de classes ou sous-nuages disjoints I_n de barycentres G_n , K_n, le nombre d'éléments de la classe I_n, M le nombre de variables décrivant chaque observation et x_{ij}, xG_{nj}, xG_j, respectivement les coordonnées des objets dans l'espace à M dimensions, le centre de gravité du sous-nuage I_n e la classe k et le centre de gravité de tout le nuage de points.
On note que dans cet algorithme, le choix des L classes ou centres initiaux s'effectue sur la base d'un tirage aléatoire sans remise à partir de la population à classifier. La partition des classes est modifiée à chaque affectation d'un individu i à l'une des classes après le calcul des distances euclidiennes par rapport à chaque centre G_n des classes initiales. L'individu est affecté à la classe correspondant à la distance la plus faible. Cette opération est effectuée pour chaque élément du nuage de point. Les centres de nouvelles classes ainsi formées sont recalculés. Le calcul à nouveau de toutes

les distances de chaque élément à ces nouveaux centres est effectué encore et l'affection de ces éléments est aussi effectuée et de nouvelles classes ont ainsi formées. L'opération se poursuit ainsi jusqu'à l'arrêt de la procédure.

Le déroulement de la méthode des k-means s'effectue selon l'algorithme de la figure 2.1.

Etape 0- On choisit par un tirage aléatoire sans remise k individus parmi n individus composant l'ensemble I. Ces K centres notés $\{c^0_1, c^0_2, \ldots\ldots, c^0_k\}$ sont provisoires.

Etape 1- Chaque individu i de I est affecté à une classe et une seule. Chacune de ces classes est localisée par son centre. La procédure d'affectation est la suivante : l'élément i est affecté à la classe notée p^0_1 de centre c^0_1 si et seulement si :

$d(i, c^0_1) < d(i, c^0_k),\ k\#1\ ;\ k=1,2,\ldots,K$

Après avoir affecté tous les individus on obtient K classes notées $\{p^0_1, p^0_2, \ldots, p^0_K\}$ de centres respectifs $\{c^0_1, c^0_2, \ldots, c^0_K\}$.

Etape 2- En considérant les K classes obtenues à l'étape1, on recalcule les centres de gravité. On obtient donc K nouveaux centres notés $\{c^1_1, c^1_2, \ldots, c^1_K\}$. On utilise la même règle d'affectation qu'à l'étape 1. On obtient K nouvelles classes $\{p^1_1, p^1_2, \ldots, p^1_K\}$ de centres respectifs $\{c^1_1, c^1_2, \ldots, c^1_K\}$.

Etape h On détermine K nouvelles classes en calculant les centres de gravité des classes obtenues à l'étape (h-1). La règle d'affectation reste la même qu'à l'étape précédente et on obtient, par la suite, une nouvelle répartition $\{p^h_1, p^h_2, \ldots, p^h_K\}$ de centres respectifs $\{c^h_1, c^h_2, \ldots, c^h_K\}$.

Arrêt de l'algorithme

L'arrêt de l'algorithme de la méthode des "k-means" se fait

- Lorsque deux itérations successives conduisent à une même partition.
- Lorsqu'on fixe un critère d'arrêt tel que le nombre maximal d'itérations.

Fig.2.1 Algorithme des k-means

La figure 2.2 illustre un exemple de déplacement de deux entres centres d'inertie de deux classes de données et la formation des classes.

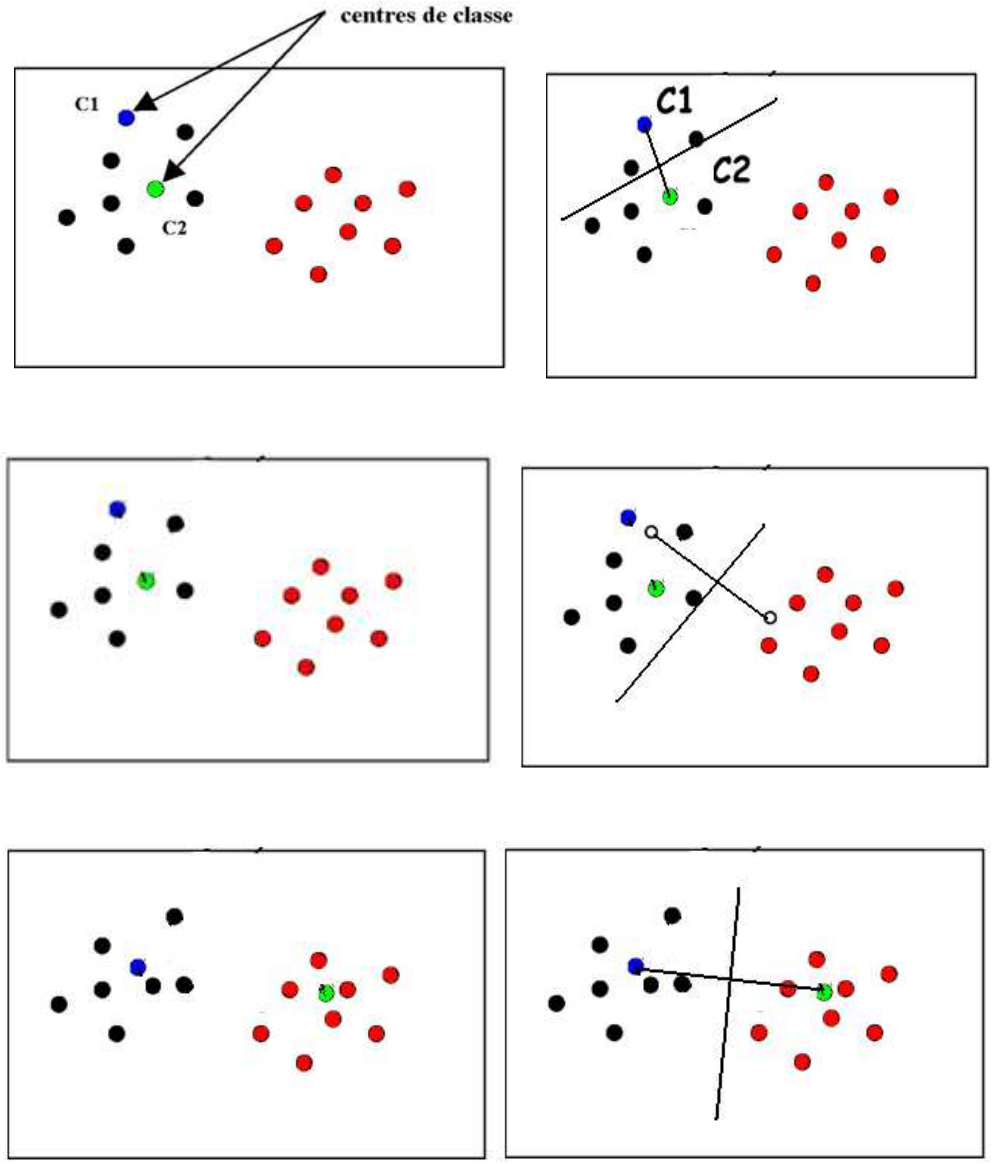

Fig.2.2 exemple de déplacement des centres des classes

4.Conclusion

Ce présent chapitre a donné un aperçu sur la diversité et l'étendue des méthodes de classification automatique. Plusieurs autres méthodes n'ont pu être rappelées notamment les approches telles que les surfaces séparatrices, la classification séquentielle, la classification avec rejet, la classification par regroupement ou clusterings etc. D'autres méthodes dites interactives associent l'utilisateur à la

classification en lui proposant des visualisations graphiques des regroupements. Encore faut-il pour cela que cette visualisation ne présente pas de distorsions.

Vu que l'apprentissage est très important et constitue la classification automatique proprement dite, deux situations peuvent se présenter, à savoir lorsque l'on dispose d'une partition a priori ou non. C'est l'une des caractéristiques qui différencient les méthodes de classification automatique.

Notons que pour la méthode des k-means, fortement liée au nombre de k classes fixées *a priori*, il est intéressant de déterminer automatiquement ce nombre. Par ailleurs ; on peut choisir une autre distance autre que l'euclidienne parmi les distances existant en analyse de données. Pour l'arrêt de la l'algorithme, il peut être décidé, aussi par l'évaluation de différents indices qui peuvent constituer une fonction objective à optimiser. Ceci fera l'objet du prochain chapitre.

Chapitre 3

Recuit Simulé et Classification Automatique

1. Introduction

Rappelons que l'objectif de la classification est de regrouper des données en classes distinctes telles que les données appartenant à la même classe soient les plus proches possibles, au sens d'une distance prédéfinie, et que les données appartenant à des classes différentes soient les plus éloignées possible au sens de cette même distance. Ceci revient donc à trouver une partition de l'ensemble des données qui minimise un indice de validité ou une erreur à choisir parmi tant d'autres. En ce qui nous concerne, il s'agit de minimiser la variance intraclasse et maximise la variance interclasses. Par ailleurs, en classification automatique, différentes méthodes ont été proposées. Cependant certains travaux ont montré que l'orientation qui a donné les meilleurs résultats jusqu'à présent est celle des métaheuristiques, vu la complexité du problème. C'est ainsi que l'introduction des métaheuristiques en classification automatique a donné lieu à de nombreux travaux. Dans le présent travail, nous allons nous intéresser à l'hybridation de la méthode des k-means avec le recuit simulé.

2- Métaheuristiques en Classification automatique

Plusieurs méthodes d'optimisation basées sur les techniques diverses ont été proposées en classification automatique et continuent de faire sujet de nouveaux travaux dans le domaine. Parmi ces métaheuristiques, celles à base de population comme les essaims de particules, les algorithmes évolutionnaires parmi lesquels les algorithmes génétiques, les algorithmes biomimétiques imitant le comportement social des animaux et insectes sont actuellement en plein expansion [24]. On notera aussi la classification automatique utilisant les fourmis artificielles dont le premier algorithme a été proposé par Lumer et Faietaen 1992 [25]. La classification par essaim de particules ou PSO où le comportement collectif des animaux a toujours été source d'inspiration pour résoudre des problèmes d'optimisation ont aussi fait objet d'application en classification automatique [26]. C'est ainsi que nous pouvons trouver de nombreux algorithmes biomimétiques, à savoir, les approches évolutives basés sur l'évolution et la sélection d'individus. On peut notamment citer les algorithmes évolutionnaires qui s'inspirent de la théorie de l'évolution darwinienne [27], les systèmes immunitaires artificiels qui s'inspirant du fonctionnement des systèmes immunitaires des vertébrés [28] et l'algorithme des mouches, variante des algorithmes évolutionnaires [29]. Parmi les métaheuristiques à base de voisinage utilisées en classification automatique, nous pouvons citer la recherche tabou et le recuit simulé. La recherche tabou est une métaheuristique introduite par Glover 1986 [30]. Elle est efficace dans la résolution de nombreux problèmes NP-difficiles. Le principe de cette méthode est de doter la recherche d'une mémoire pour éviter le cyclage. L'algorithme (fig.3.1) utilise donc une liste dite « liste taboue » dans laquelle les dernières solutions parcourues sont enregistrées. Ainsi,

le rebouclage sur la même solution indéfiniment est évité. En classification automatique, la Recherche Tabou a été appliquée par Al-Sultan en 1995 [31]. Le programme débute avec un partitionnement aléatoire en *K* classes. Pour chaque donnée appartenant à la classe *k',* on génère un nombre aléatoire *R* tel que 0<=R<= 1. Si R>=Pt où *Pt* est un paramètre de l'algorithme qui détermine l'étendue du voisinage alors la donnée est ôtée de sa classe original k' et affectée aléatoirement à une classe k" telle que k'><k". Sinon, la donnée est gardée dans sa classe d'origine. On procède ainsi pour générer un certain nombre de solutions voisines de sorte que chaque parcours de toutes les données génère une solution. La fonction objectif qui peut être l'indice VCR ou l'erreur quadratique est utilisée 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. Cette métaheuristique a été hybridée avec la méthode des 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 des k-means et celui de la Recherche Tabou seule.

```
Recherche Tabou :
Début :
Choisir une solution S au hasard (ou grâce à un autre algorithme.)
Solution courante ← S
La liste taboue ← %
Tant que (critère d'arrêt = faux) faire
        Insérer solution courante dans liste taboue
        Générer un voisinage de la solution courante
        Trier les solutions du voisinage de la meilleure à la pire*
        Tant que pas d'affectation faire
                Si sol . liste taboue alors
                        Solution courante ←Sol
                        Affectation ←vrai
                Sinon
                         Passer à la solution suivante
                Fin si
        Fin tant que.
        Si pas d'affectation choisir une autre solution au hasard
Fin tant que
Fin
```

Fig.3.1 Pseudo-code de la Recherche Tabou

Quant au recuit simulé, nous allons l'utiliser plus en détail en l'hybridant avec la méthode des k-means.

3- Hybridation k-means et Recuit Simulé

3.1 Rappels sur le Recuit Simulé

Rappelons que l'algorithme de Recuit Simulé décrit dans le chapitre a comme point fort, sa capacité à sortir des optima locaux et ce, grâce à sa fonction d'acceptation. En effet, même si une solution est plus mauvaise que la solution courante, on peut quand même l'explorer selon une certaine probabilité. Par ailleurs, grâce au formalisme des chaines de Markov, la convergence du Recuit Simulé vers la solution dans un problème d'optimisation est garantie. La figure 3.2 rappelle le principe de l'algorithme du RS :

1 Choisir aléatoirement, une solution initiale X du système à optimiser et évaluer la valeur de la fonction objectif F = F (X).
2 Choisir une température initiale élevée T.
3 Perturber cette solution pour obtenir une nouvelle solution X' = X+ΔX ;
4 Calculer ΔF = F(X') – F(x) ;
5 Si ΔF ≤ 0,
Alors, accepter la solution X'; Faire X ← X' ;
Sinon, calculer P = exp (-ΔF/T) ;
Générer un nombre aléatoire R compris entre 0 et 1 ;
Si R ≤ P,
Alors accepter la nouvelle solution X' ; Faire X ← X' ;
Sinon refuser la solution X ;
6 Sauver le meilleur point rencontrer ;
7 Si l'équilibre thermodynamique du système à la température T est atteint,
Alors abaisser légèrement la température T ;
Sinon aller à l'étape 3 ;
8 Si le système est figé,
Alors aller à l'étape 9 ;
Sinon aller à l'étape 3 ;
9 Solution = meilleur point trouvé. Arrêt du programme.

Fig.3.2 Pseudo-code du RS.

3.2 Rappels sur la méthode des k-means

Présentée dans le chapitre précédent, la méthode des k-means converge rapidement mais, le plus souvent, vers un optimum local. L'algorithme des k-means est, toutefois, conditionné par le choix des centres initiaux des classes. Rappelons que, globalement, son principe consiste à fixer, initialement, les centres des classes aléatoirement, puis à calculer les distances euclidiennes de toutes les observations par apport aux centres des classes. Chaque observation est affectée à la classe pour laquelle la distance euclidienne est la plus petite. On aura donc un nouveau partitionnement. Le processus est répété jusqu'à ce qu'il converge, autrement dit, jusqu'à ce que les centres de gravité ne varient plus.

Algorithme K-Means
Entrée : un ensemble *O* d'objets __ et le nombre de classe K.
Sortie : partition de *O* en K classes.
Début :
Générer aléatoirement K objets et les considérer comme les centres de gravités initiaux.
Tant que (condition d'arrêt = faux)
    Pour i allant de 1 à Nombre d'objets :
        Affecter __ à la classe __ dont le centre de gravité lui est le plus proche.
        Calculer les nouveaux centres de gravité des classes (l'ancienne et la nouvelle classe en cas de réaffectation.)
    Fin pour.
Fin tant que.
Fin.

Fig.3.3 Pseudo-code de la méthode des k-means

3.3 Hybridation entre la méthode des k-means et le RS

La méthode de k-means réalise une bonne classification de données ce qui revient à trouver une bonne configuration qui permet de bien distinguer les classes. Le problème peut être représenté par une suite indéfinie d'itérations qui calcule les centres de gravité et une fonction objective comme l'indice VCR qui assigne une valeur à chaque itération. L'objectif est de trouver l'optimum global qui permet de trouver une partition de l'ensemble des données qui

minimise la variance intra-classe et maximise la variance interclasses. Pour trouver cet optimum, l'algorithme du recuit simulé peut être utilisé conformément à la figure 3.4.

k-means et recuit simulé

Étape 1

Partitionner les n données en K classes aléatoirement. Initialiser le compteur de boucle nb=0 et calculer les centres de gravité des classes.

Étape 2

Perturber la partition en cours en choisissant aléatoirement une donnée et en l'affectant aléatoirement à une autre classe. Calculer les nouveaux centres de gravité.

Étape 3

Soit *E* l'erreur quadratique de l'ancienne partition et *E'* celle de la nouvelle.
ΔE=E'-E
La nouvelle perturbation est acceptée si Exp $(-\Delta E/T_{nb})>\gamma$.
où γ est un nombre choisi aléatoirement dans l'intervalle]0,1[et T_{nb}.la température correspondant à l'itération nb

Étape 4

Si nb ≥ maximum ou E assez petit alors fin du programme, sinon incrémenter *nb* et aller à l'étape 2.

Fig 3.4 hybridation entre les k-mean et le RS

La température initiale T_0 est un paramètre de l'algorithme. Elle est régie par deux contraintes : $T_{nb-1} \geq T_{nb}$.et $lim_{nb\to\infty}(T_{nb}) = 0$.
En général, on choisit une décroissance géométrique de la température, c'est-à-dire $T_{nb} = \alpha T_{nb-1}$ où $\alpha \in]0,1[$.

Dans Trejos et autres (1998) et Piza *et al.* [32], a été présentée l'application de métaheuristiques au problème de partitionnement tel que le recuit simulé, dans un contexte euclidien. Les résultats sont sensiblement meilleurs que ceux des k-means.
Ainsi, selon l'algorithme ci-dessus, nous commençons par choisir une première partition aléatoire. Pour chaque paramètre T_{nb} de « température », nous réitérons avec le procédé suivant :

- à chaque étape, nous choisissons, au hasard un objet, disons x_i. Nous choisissons également, au hasard, l'index l du nouveau groupe vers lequel l'objet x_i pourrait être transféré.
  Le transfert est fait réellement avec la probabilité $\{1, e^{-\Delta E/T_{nb}}\}$ (règle de Métropolis), où ΔE est le changement produit dans la fonction objective.
  Après quelques itérations, nous changeons le paramètre de température de refroidissant $T_{nb+1}<T_{nb}$ du système et répétons le processus de transfert, jusqu'à ce que les critère d'arrêt soit atteint.

Le processus de refroidissement utilisé est:

1. *Température initiale.* La temperature initiale T_0 est calculée de telle manière que, au début, la probabilité approximative d'accepter de nouvelles « mauvaises partitions » (ceux qui augmentent E, est environ ζx100%. Ceci est effectué en choisissant T_0 =ΔW+/(ln 1/ζ) où ΔE+ est la moyenne de variation dans la fonction objective E, pour les partitions plus mauvaises que la cloison initiale. Le paramètre ζ est fixé à 0.7.

2. <u>*Refroidissement de la température*</u> :
Nous calculons $T_{k+1} = \lambda T_k$, où $\lambda=0.92$.

3. <u>*Processus de transfert itératif :*</u>
Pour chaque température T_k, nous employons l'approche homogène, dans laquelle la largeur maximale de la chaîne de Markov est de N_{over} pas. Cependant, si de nouvelles N_{limit} « mauvaises partitions » étaient déjà acceptées, alors le processus de la température s'arrête. on emploie $N_{over}=M_{in}$ $(100n^2(k-1), 20000)$ et $N_{limit} = Min\ (10n^2\ (k-1), 4000)$.

4. *Température finale* : l'algorithme s'arrête à la température finale. Cependant, nous arrêtons l'algorithme si aux dernières valeurs de température, aucun transfert n'est effectué.

4. Tests et Résultats

Trois ensembles de données sont présentés : un fichier de variables binaires crée artificiellement, le fichier Iris et un fichier de grande dimension.

Fichier de variables binaires

L'ensemble de données artificielles est composés de 20 objets décrits par 6 variables binaires (figure 3.5). Ces données sont réparties en les 4 classes suivantes: {1.2.3.4.5}, {6.7.8.9.10}, {11.12.13.14.15} et {16.17.18.19.20}.
Les paramètres du recuit simulé sont décrits dans la section précédente. L'application du RS donne les mêmes classes, sans erreur.

Objet			Objet	
1	1 1 1 1 1 1		11	0 0 0 1 1 1
2	1 1 1 1 1 1		12	0 0 0 1 1 1
3	1 1 1 1 1 1		13	0 0 0 1 1 1
4	1 1 1 1 1 1		14	0 0 0 1 1 1
5	1 1 1 1 1 1		15	0 0 0 1 1 1
6	1 1 1 0 0 0		16	0 0 0 0 0 0
7	1 1 1 0 0 0		17	0 0 0 0 0 0
8	1 1 1 0 0 0		18	0 0 0 0 0 0
9	1 1 1 0 0 0		19	0 0 0 0 0 0
10	1 1 1 0 0 0		20	0 0 0 0 0 0

Fig.3.5 Fichier de données binaires

Fichier Iris :

Le fichier Iris est utilisé par un grand nombre de chercheurs en classification automatique afin de tester les performances des méthodes proposées. Il constitué d'un fichier de 150 fleurs d'iris composé de trois espèces avec 50 fleurs pour chacune des espèces. Sur chacune de ces fleurs, ont été effectuées 4 mesures : la longueur et la largeur des sépales, la longueur et la largeur de pétales. Un tableau de 150 x 4 est ainsi constitué. Ainsi ce fichier présente les caractéristiques suivantes :

Nombre d'objets 150
Nombre de classes 3
Nombre d'attributs 4
Nombre d'objets dans la Classe 1, 50
Nombre d'objets dans la Classe 2, 50
Nombre d'objets dans la Classe 3; 50
L'application de la méthode donne les résultats suivants :
Nombre de classes obtenues : 03
Nombre de fleurs dans la clase1 : 50
Nombre de fleurs dans la clases 2 : 47
Nombre de fleurs dans la classe 3, 53

La figure 3.6 montre une visualisation de ce fichier d'iris sur deux paramètres..

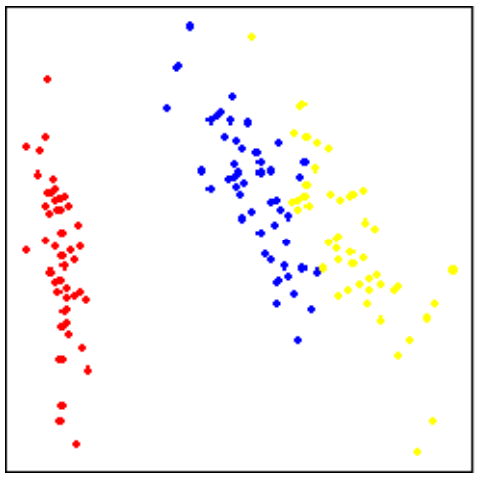

Fig.3.6 Visualisation des données du fichier de données iris

Fichier de grandes données

La méthode a aussi été appliquée à un fichier de données de très grande dimension. Il s'agit d'un fichier de données relatives à 1336 abeilles décrites par six paramètres. La figure 3.6 montre quatre classes.

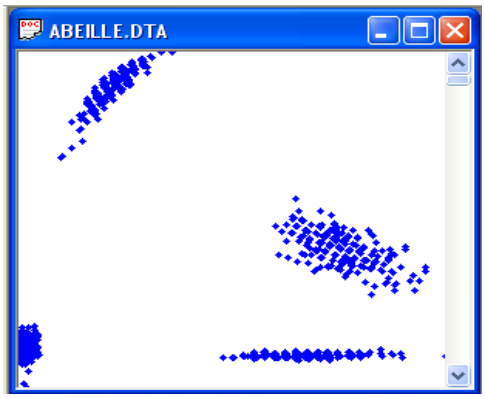

Fig.3.7 Fichier de données réelles de grande dimension

5- Conclusion :

Nous avons présenté un chapitre sur les métaheuristiques appliquées à la classification non supervisée. Il est apparu que le résultats obtenus sont plus performants avec cette hybridation. Le Recuit Simulé, en particulier, est dans cette même logique. Cependant, les chercheurs sont de plus en plus intéressés par les algorithmes biomimétiques et particulièrement ceux mimant le comportement des insectes sociaux. L'approche multi-agent se révèle en effet très efficace et peu complexe à mettre en œuvre.

Les métaheuristiques constituent un ensemble de méthodes approchées adaptables à un très grand nombre de problèmes combinatoires et de problèmes d'affectation sous contraintes. Elles ont publié 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. L'étude de ces méthodes est actuellement en plein développement.

Les contextes décisionnels rencontrés dans les situations réelles ont montré les limites des méthodes classiques de la recherche opérationnelle. Résoudre exactement des problèmes d'optimisation combinatoire est très dur. En effet, pour de tels problèmes, les méthodes exactes requièrent un effort calculatoire qui croît exponentiellement avec la taille des instances du problème (explosion combinatoire) et, rapidement, les heuristiques ou métaheuristiques deviennent l'unique moyen d'obtenir une bonne solution en un temps raisonnable. Dans Ce cas on trouve le recuit simulé comme première métaheuristiques utilisé dans de nombreux travaux et application pour des raisons de faciliter d'implémentation et pour avoir permis de résoudre de nombreux problèmes. Nous avons étudié les méthodes classe est de classification non supervisée de donnée avec l'algorithme des k-means. L'application des k-means à montré son rôle important d'apporter des solutions proches de l'optimum global, ainsi qu'une rapidité de traitement. Pour atteindre l'optimum global, on fait l'hybridation de la méthode des k-means avec le recuit simulé qui est l'objet de notre travail. Les résultats obtenus sur des fichiers de données divers ont montré la puissance de la méthode. L'application de

cette méthode sur un corpus réel montre objectivement l'amélioration observée dans les résultats obtenus.

On peut conclure que l'hybridation conduit à de meilleurs résultats par rapport à ceux fournis par la méthode des k-means seule. Il serait cependant intéressant d'approfondir la présente étude en utilisant différents fichiers et en testant l'ensemble des paramètres de la méthode. Il serait aussi intéressant de programmer, dans la même démarche, les autres métaheuristiques, surtout à base de population, inspirées du comportement animal, qui paraissent encore plus prometteuses.

Références bibliographiques

[1] P. SIARRY, *les méthodes d' «optimisation globale.* Ecole centrale Paris, 1France. Octobre 1997.

[2] Petrowski J.A., P. Siarry and E.D. Taillard, « Métaheuristiques pour l'optimisation difficile » Eyrolles, 2003

[3] 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.

[4] F Glover, M. Laguna «Tabu Search » Kluwer Academic Publishers, Boston, 1997.

[5] 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.

[6] 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.

[7] 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

[8] 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.

[9] Clerc, M., L'optimisation par essaim particulaire : principe, modèles et usages. Techniques et Science Informatiques. Vol.21, 2002, p.941-964.

[10] J. Kennedy et R. C. Eberhart. « Particle Swarm Optimisation. » Proc. IEEE Int'l Conf. on neuronal networks. Tome 4. Pages 1942-1948. 1995

[11] S. KIRKPATRICK, C.D. GELATT, M.P. VECCHI, *Optimisation by simulated annealing*, Science, Vol. 220, n^0 4598, 1983.

[12] E. H. L. Aarts et P. J. M. Van Laarhoven. « Statistical cooling : a general approach to combinatorial optimisation problems. » Philips J. of Research, tome 40, pages 193-226. 1985

[13] Boudjemaa. F « application des metaheuristiques pour optimiser une méthode de classification automatique » mémoire de magister en automatiqueUniversité Mouloud Mammeri de TIZI-OUZOU 2005

[14] 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

[15] E. Triki , Y. Collette , P. Siarry, «A theoretical study on the behavior of simulated annealing leading to a new cooling schedule», European Journal of Operational Research 166 2005

[16] Törn, A.A. Cluster analysis as a tool in a global optimization model, In Proceedings of Third International Congress of Cybernetics and Systems, Bucharest 1975, Springer Verlag, 249-260.

[17] 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.

[18] Jain *et al.*; « Data Clustering: A review », ACM Computing Surveys, Vol. 31, No 3, 1999

[19] A. Belaïd., Implémentation d'une méthode de classification non supervisée de données à l'aide d'algorithmes génétiques évolutionnaires ». Mémoire d'ingénieur. INI. 2008.

[20] P. Brucker. « On the complexity of clustering problems. » Lecture Notes in Economics and Mathematical Systems, Springer, Berlin. 1978

[21] . 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.

[22] 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.

[23] A. K. Jain et R. C. Dubes. « Algorithms for clustering data.» Editions Prentice Hall Advanced Reference Series. 1988.

[24] J. Kennedy et R. C. Eberhart. « Particle Swarm Optimisation. » Proc. IEEE Int'l Conf. on neuronal networks. Tome 4. Pages 1942-1948. 1995.

[25] 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.

[26] A. 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.

[27] J. Louchet, M. Guyon, M-J. Lesot et A. Boumaza. « L'algorithme des mouches dynamiques : guider un robot par évolution artificielle en temps réel ». Extraction des Connaissances et Apprentissage : Apprentissage et Evolution, vol. 1 – N.3. Pages 115-130. 2001

[28] T. Ohira. « Immune pattern recognition system. » Proc. of International
Workshops on Biologically Inspired Evolutionary Systems. Tokyo. Japon. 1995

[29] 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.

[30] Chelouah R. ; Siarry, P., *"Tabu Search* applied to global optimization", European Journal of Operational Research, vol. 123, pp. 256-270, 2000.

[31] Al-Sultan. « A tabu search approach to the clustering problem. » Pattern
Recognition, volume 28, no. 9, pages 1443-1451. 1995

[32] P. Borne, Métaheuristiques pour l'optimisation difficile, Revue REE, avril pp. 1-23, 2009