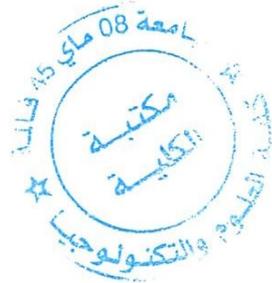


17/621.728

République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université 8Mai 1945 – Guelma
Faculté des sciences et technologies
Département d'électronique et télécommunications



**Mémoire de fin d'étude
pour l'obtention du diplôme de mastère Académique**

Domaine : **Sciences et Techniques**
Filière : **Electronique et Télécommunications**
Spécialité : **Electronique /Systèmes Electroniques**

Optimisation par Essaim Particulaire

Présenté par :
-Hammouda Ammar
-Hallaci Salah Nabil

Sous la direction de :
-Dr:Boudjehem Djalil



JUIN 2012

REMERCIEMENTS

Nos remerciements les plus vifs, vont à notre Encadreur monsieur Boudjehem Djalil , pour son aide, son orientation judicieuse et sa disponibilité, aussi pour la confiance, la patience et la compréhension qu'il nous a toujours manifesté...nous remercions aussi tout nos professeurs qui nous ont toujours aidés.



*« Une personne qui n'a jamais commis d'erreur...
...n'a jamais tenté d'innover »
Albert Einstein*

DEDICACE

Je dédie ce travail à :

Ma Mère,

A mon Père

A mes frères

Sofian, A. Majid, Mehdi et Amis

A mes Sœurs

Et à tous mes Amis

AMMAR HAMMOUDA

06/2012

DEDICACE

Je dédie ce travail à :

Mes Parents

A mes frères,

Et à tous mes Amis

HALLACI SALAH.N

06/2012

Table des Figures

Figure 2.1 Exemple coopération.....	19
Figure 2.2 Schéma de principe du comportement des particules d'un essaim.....	24
Figure 2.3 Le cercle virtuel pour un essaim.....	26
Figure 2.4 (a) anneau (avec $n=2$), (b) rayon, (c) étoile.....	31
Figure 2.5 Organigramme de principe de la méthode des essais particulaires	30
Figure 3.1 Représentation en 3D de la fonction quadratique f	37
Figure 3.2 Déplacements des particules pendant l'opération d'optimisation par l'essai particulaire de la première fonction après chaque 15 itérations...38	
Figure 3.3 Déplacements des particules pendant l'opération d'optimisation par l'essai particulaire pour $C2=3$	39
Figure 3.4 Déplacements des particules pendant l'opération d'optimisation par l'essai particulaire pour $C1=2$	41
Figure 3.5 Déplacements des particules pendant l'opération d'optimisation par l'essai particulaire pour $W=1.2$	42
Figure 3.6 Représente la fonction de Ackley en dimension deux.....	43
Figure 3.7 l'optimisation par essaim particulaire avec la fonction de Ackley à 2 variables.....	44
Figure 3.8 Représente la fonction de Shewefel en dimension deux.....	45
Figure 3.9 l'optimisation par essaim particulaire avec la fonction de Schewefelà 2 variables.....	45

Sommaire :

▪ *Introduction générale.*

▪ *Chapitre 1 : L'Optimisation.*

<i>I-Définition</i>	<i>1</i>
<i>II-Historique</i>	<i>1</i>
<i>III-Domains d'application</i>	<i>3</i>
<i>IV-Techniques d'optimisation</i>	<i>3</i>
<i>IV-1-Algorithmes génétiques</i>	<i>3</i>
<i>IV-2-Le Recuit simulé</i>	<i>9</i>
<i>IV-3- Optimisation par essaim particulaire</i>	<i>13</i>

▪ *Chapitre 1 : L'Optimisation par Essaim particulaire.*

<i>I-Introduction</i>	<i>14</i>
<i>II- METHODES D'OPTIMISATION MULTI-OBJECTIF</i>	<i>17</i>
<i>II -1 Optimisation par Essaims de Particules</i>	<i>18</i>
<i>III- Présentation de la méthode</i>	<i>21</i>
<i>III-1 Principe de la PSO</i>	<i>21</i>
<i>III -2 État et comportement d'une particule</i>	<i>22</i>
<i>III -3 Information utilisée par une particule pour déterminer sa prochaine position</i>	<i>23</i>
<i>III-4 Points essentiels</i>	<i>28</i>

IV- Formalisation29

IV-1 Configuration de la méthode30

*IV-2 L'organigramme de principe de la méthode des essais
particulaires35*

▪ *Chapitre 3 : programmation et*

Simulation.

I-Introduction.....37

II- Synthèse de fonctionnement d'un PSO.....37

III- Optimisation des fonctions test.....43

IV-Conclusion46

▪ *Conclusion générale.*

▪ *bibliographie.*

Introduction générale

Introduction générale :

Dans ce mémoire nous allons développer le thème de l'Optimisation par Essaim Particulaire qui est une technique relativement récente, puisque les premiers articles la concernant datent de 1995, l'appellation d'origine étant, en anglais, Particle Swarm Optimization (*PSO*). Il s'agit d'une méthode faisant appel à une population d'agents, appelés particules, mais, comme on le verra, par rapport à d'autres heuristiques de la même famille, elle présente quelques particularités intéressantes, entre autres la notion que l'efficacité est due à la collaboration plutôt qu'à la compétition.

Et pour ce fait on a tout d'abord commencé par un premier chapitre consacré à l'optimisation d'une manière globale, on définit cet outil d'analyse, et on présente ses différentes techniques comme les algorithmes génétiques et le recuit simulé, ce qui nous mène à entrer dans le vif du sujet les (*PSO*) qui est l'autre technique d'optimisation qu'on a détaillé dans le deuxième chapitre.

Cette partie on l'a entamé par une petite analogie des différentes classes d'optimisation puis on s'est penché sur la présentation des *PSO* (principe, efficacité, comportement d'une particule, voisinage), Et on a terminé par la mise en place d'un organigramme de principe de la méthode essaim particulaire.

Et enfin le dernier chapitre est dédié à la programmation et la simulation

Chapitre 1 :
L'Optimisation.

-L'Optimisation :

I-Définition :

L'optimisation est une branche des mathématiques, cherchant à analyser et à résoudre analytiquement ou numériquement les problèmes qui consistent à déterminer le meilleur élément d'un ensemble, au sens d'un critère quantitatif donné. Ce mot vient du latin optimum qui signifie le meilleur.

L'optimisation joue un rôle important en recherche opérationnelle (donc en économie et microéconomie), dans les mathématiques appliquées (fondamentales pour l'industrie et l'ingénierie), en analyse et en analyse numérique, en statistique pour l'estimation du maximum de vraisemblance d'une distribution, pour la recherche de stratégies dans le cadre de la théorie des jeux, ou encore en théorie du contrôle et de la commande. Aujourd'hui, tous les systèmes susceptibles d'être décrits par un modèle mathématique sont optimisés. La qualité des résultats et des prédictions dépend de la pertinence du modèle, de l'efficacité de l'algorithme et des moyens pour le traitement numérique. [1]

II-Historique :

Les premiers problèmes d'optimisation auraient été formulés par Euclide, au III^e siècle avant notre ère, dans son ouvrage historique *Éléments*. Trois cent ans plus tard, Héron d'Alexandrie dans *Catoptrica* énonce le *principe du plus court chemin* dans le contexte de l'optique. Le plus court chemin pour aller de A à C en passant par un point B de la droite est obtenu lorsque l'angle d'incidence est égal à l'angle réfléchi (sur la figure, il s'agit du chemin vert).

Au XVII^e siècle, l'apparition du calcul différentiel entraîne l'invention de techniques d'optimisation, ou du moins en fait ressentir la nécessité. Newton met au point une méthode itérative permettant de trouver les extrémums locaux d'une fonction en faisant intervenir la notion de dérivée, issue de ses travaux avec Leibniz. Cette nouvelle notion permet de grandes avancées dans l'optimisation de fonctions car le problème est ramené à la recherche des racines de la dérivée.

Durant le XVIII^e siècle, les travaux des mathématiciens Euler et Lagrange mènent au calcul des variations, une branche de l'analyse fonctionnelle regroupant plusieurs méthodes d'optimisation. Ce dernier invente une technique d'optimisation sous contraintes: Les multiplicateurs de Lagrange. Le XIX^e siècle est marqué par l'intérêt croissant des économistes pour les mathématiques. Ceux-ci mettent en place des modèles économiques qu'il convient d'optimiser, ce qui accélère le développement des mathématiques. Depuis cette période, l'optimisation est devenue un pilier des mathématiques appliquées et le foisonnement des techniques est tel qu'il ne saurait être résumé en quelques lignes.

On peut tout de même évoquer l'invention de plusieurs méthodes itératives utilisant le gradient de la fonction, ainsi que l'utilisation du terme *programmation mathématique*, pour désigner des problèmes d'optimisation.

Historiquement, le premier terme introduit fut celui de *programmation linéaire*, inventé par George Dantzig vers 1947. Le terme *programmation* dans ce contexte ne réfère pas à la programmation informatique (bien que les ordinateurs soient largement utilisés de nos jours pour résoudre des programmes mathématiques). Il vient de l'usage du mot *programme* par les forces armées américaines pour établir des horaires de formation et des choix logistiques, que Dantzig étudiait à l'époque. L'emploi du terme *programmation* avait également un intérêt pour débloquer des

crédits en une époque où la planification devenait une priorité des gouvernements. L'expression *programmation mathématique*, qui requiert la longue explication ci-dessus, tend à être abandonnée. Par exemple, en juin 2010, la société savante internationale qui représente cette discipline a vu son nom précédent *Mathematical Programming Society* changé en *Mathematical Optimization Society* ; pour la même raison, on préfère aujourd'hui utiliser les locutions *optimisation linéaire/quadratique/...* au lieu de *programmation linéaire/quadratique/...*

III-Domains d'application : Ils sont extrêmement variés : optimisation d'un trajet, de la forme d'un objet, d'un prix de vente, d'une réaction chimique, du contrôle aérien, du rendement d'un appareil, du fonctionnement d'un moteur, de la gestion des lignes ferroviaires, du choix des investissements économiques, de la construction d'un navire, etc. L'optimisation de ces systèmes permet de trouver une configuration idéale, d'obtenir un gain d'effort, de temps, d'argent, d'énergie, de matière première, ou encore de satisfaction. [1]

IV-Techniques d'optimisation :

VI-1-Algorithmes génétiques :

Les algorithmes génétiques sont un outil d'optimisation dont le but est d'obtenir une solution approchée à un problème d'optimisation, lorsqu'il n'existe pas de méthode exacte (ou que la solution est inconnue) pour le résoudre en un temps raisonnable. Les algorithmes génétiques utilisent la notion de sélection naturelle et l'appliquent à une population de solutions potentielles au problème donné. La solution est approchée par «bonds» successifs, comme dans une procédure de séparation et évaluation, à ceci près que ce sont des formules qui sont recherchées et non plus directement des valeurs.[2]

VI-1-1- Présentation :

a-Analogie avec la biologie :

Les algorithmes génétiques étant basés sur des phénomènes biologiques, il convient de rappeler au préalable quelques termes de génétique.

Les organismes vivants sont constitués de cellules, dont les noyaux comportent des chromosomes qui sont des chaînes d'ADN. L'élément de base de ces chromosomes (le caractère de la chaîne d'ADN) est un gène. Sur chacun de ces chromosomes, une suite de gènes constitue une chaîne qui code les fonctionnalités de l'organisme (la couleur des yeux ...). La position d'un gène sur le chromosome est son locus. L'ensemble des gènes d'un individu est son génotype et l'ensemble du patrimoine génétique d'une espèce est le génome. Les différentes versions d'un même gène sont appelées allèles.

On utilise aussi, dans les algorithmes génétiques, une analogie avec la théorie de l'évolution qui propose qu'au fil du temps, les gènes conservés au sein d'une population donnée sont ceux qui sont le plus adaptés aux besoins de l'espèce vis-à-vis de son environnement. [2]

b-Principe :

Les algorithmes génétiques, afin de permettre la résolution de problèmes, se basent sur les différents principes décrits ci-dessus. Le problème théorique de la convergence a été résolu par Raphael Cerf, en se basant sur la théorie de Friedlin Weizel des perturbations stochastiques des systèmes dynamiques. La démonstration de R. Cerf montre d'ailleurs que le processus de convergence dépend essentiellement de la mutation, le croisement pouvant être éliminé en théorie. Cependant, la preuve théorique de convergence n'a que peu d'utilité dans la pratique, où l'opérateur de croisement fait bien souvent toute la richesse de l'algorithme génétique par rapport à des méthodes de type recuit simulé.

De manière globale, on commence avec une population de base qui se compose le plus souvent de chaînes de caractères correspondant chacune à un chromosome. Nous reviendrons par la suite sur les différentes structures de données possibles (voir Codage) mais nous retiendrons pour le moment l'utilisation du codage binaire (ex. : 0100110).

Le contenu de cette population initiale est généré aléatoirement. On attribue à chacune des solutions une note qui correspond à son adaptation au problème. Ensuite, on effectue une sélection au sein de cette population.

Il existe plusieurs techniques de sélection. Voici les principales utilisées :

- Sélection par rang

Cette technique de sélection choisit toujours les individus possédant les meilleurs scores d'adaptation, le hasard n'entre donc pas dans ce mode de sélection. En fait, si n individus constituent la population, la sélection

appliquée consiste à conserver les k meilleurs individus (au sens de la fitness) suivant une probabilité qui dépend du rang (et pas de la fitness).

- Probabilité de sélection proportionnelle à l'adaptation (appelé aussi roulette ou roue de la fortune)

Pour chaque individu, la probabilité d'être sélectionné est proportionnelle à son adaptation au problème. Afin de sélectionner un individu, on utilise le principe de la roue de la fortune biaisée. Cette roue est une roue de la fortune classique sur laquelle chaque individu est représenté par une portion proportionnelle à son adaptation. On effectue ensuite un tirage au sort homogène sur cette roue.

- Sélection par tournoi

Cette technique utilise la sélection proportionnelle sur des paires d'individus, puis choisit parmi ces paires l'individu qui a le meilleur score d'adaptation.

- Sélection uniforme

La sélection se fait aléatoirement, uniformément et sans intervention de la valeur d'adaptation. Chaque individu a donc une probabilité $1/P$ d'être sélectionné, où P est le nombre total d'individus dans la population. [2]

c-Cas d'utilisation :

c-1-Les conditions du problème :

Comme cela a été dit plus haut, les algorithmes génétiques peuvent être une bonne solution pour résoudre un problème. Néanmoins, leur utilisation doit être conditionnée par certaines caractéristiques du problème.

Les caractéristiques à prendre en compte sont les suivantes :

- Le temps de calcul de la fonction fitness doit être raisonnablement court. En effet, celle-ci sera évaluée de nombreuses fois.
- Nombre de solutions important : les performances des algorithmes génétiques par rapport aux algorithmes classiques sont plus marquées lorsque les espaces de recherches sont importants. En effet, pour un espace dont la taille est faible, il peut être plus sûr de parcourir cet espace de manière exhaustive afin d'obtenir la solution optimale en un temps qui restera somme toute correct. Au contraire, utiliser un algorithme génétique engendrera le risque d'obtenir une solution non optimale (voir Limites) en un temps qui restera sensiblement identique.
- Pas d'algorithme déterministe adapté et raisonnable.
- Lorsque l'on préfère avoir une solution relativement bonne rapidement plutôt qu'avoir la solution optimale en une durée indéfinie. C'est ainsi que les algorithmes génétiques sont utilisés pour la programmation de machines qui doivent être très réactives aux conditions environnantes.

c-2-Applications industrielles :

Un premier exemple est une réalisation effectuée au sein de l'entreprise Motorola. Le problème pour lequel Motorola a utilisé les algorithmes génétiques concerne les tests des applications informatiques. En effet, lors de chaque changement apporté à une application, il convient de re-tester l'application afin de voir si les modifications apportées n'ont pas eu d'influence négative sur le reste de l'application. Pour cela, la méthode classique est de définir manuellement des plans de test permettant un passage dans toutes les fonctions de l'application. Mais ce type de test nécessite un important travail humain. Le but de Motorola a donc été d'automatiser cette phase de définition de plans de tests. Ils ont pour cela défini un algorithme où chaque individu correspond à un résultat d'exécution d'un programme (l'enchaînement des valeurs passées au programme) et où chaque individu reçoit une valeur qui correspond à son aptitude à passer dans un maximum de parties du code de l'application. Finalement, l'outil développé permet, à l'aide d'un algorithme génétique, de faire évoluer ces programmes de test pour maximiser les zones testées de façon à ce que lors de modifications apportées à l'application on puisse soumettre celle-ci à des tests efficaces. D'autres domaines industriels utilisent aujourd'hui les algorithmes génétiques. On peut retenir entre autres l'aérodynamique où des optimisations sont mises au point à l'aide de ces outils, l'optimisation structurelle, qui consiste à minimiser le poids d'une structure en tenant compte des contraintes de tension admissibles pour les différents éléments, et la recherche d'itinéraires : ces algorithmes ont été utilisés par la NASA pour la mission d'exploration de Mars, dans la gestion des déplacements du robot Pathfinder.

La société Sony les a aussi utilisés dans son robot Aibo. En effet, ce robot a appris à marcher dans un dispositif expérimental où son système de commande a été soumis à une évolution artificielle. Différents modes de commandes ont été testés, les plus performants ont été croisés et le résultat a été très positif. De génération en génération, le robot s'est redressé, puis a commencé à marcher en chutant souvent et a fini par marcher d'un pas assuré. [2]

VI-2-Le Recuit simulé :

Le recuit simulé est une méthode empirique (métaheuristique) inspirée d'un processus utilisé en métallurgie. On alterne dans cette dernière des cycles de refroidissement lent et de réchauffage (recuit) qui ont pour effet de minimiser l'énergie du matériau. Cette méthode est transposée en optimisation pour trouver les extrema d'une fonction.

Elle a été mise au point par trois chercheurs de la société IBM, S. Kirkpatrick, C.D. Gelatt et M.P. Vecchi en 1983, et indépendamment par V. Černý en 1985.

La méthode vient du constat que le refroidissement naturel de certains métaux ne permet pas aux atomes de se placer dans la configuration la plus solide. La configuration la plus stable est atteinte en contrôlant le refroidissement et en le ralentissant par un apport de chaleur externe.

VI-2-1- Déroulement du processus :

Le recuit simulé s'appuie sur l'algorithme de Metropolis-Hastings, qui permet de décrire l'évolution d'un système thermodynamique. Par analogie avec le processus physique, la fonction à minimiser deviendra l'énergie E du système. On introduit également un paramètre fictif, la température T du système.

Partant d'une solution donnée, en la modifiant, on en obtient une seconde. Soit celle-ci améliore le critère que l'on cherche à optimiser, on dit alors qu'on a fait baisser l'énergie du système, soit celle-ci le dégrade. Si on accepte une solution améliorant le critère, on tend ainsi à chercher l'optimum dans le voisinage de la solution de départ. L'acceptation d'une « mauvaise » solution permet alors d'explorer une plus grande partie de l'espace de solution et tend à éviter de s'enfermer trop vite dans la recherche d'un optimum local.

IV-2-2- État initial de l'algorithme :

La solution initiale peut être prise au hasard dans l'espace des solutions possibles. À cette solution correspond une énergie initiale $E = E_0$. Cette énergie est calculée en fonction du critère que l'on cherche à optimiser. Une température initiale $T = T_0$ élevée est également choisie. Ce choix est alors totalement arbitraire et va dépendre de la loi de décroissance utilisée.

IV-2-3- Itérations de l'algorithme :

À chaque itération de l'algorithme une modification élémentaire de la solution est effectuée. Cette modification entraîne une variation ΔE de l'énergie du système (toujours calculée à partir du critère que l'on cherche à optimiser). Si cette variation est négative (c'est-à-dire qu'elle fait baisser l'énergie du système), elle est appliquée à la solution courante. Sinon, elle est acceptée avec une probabilité $e^{-\frac{\Delta E}{T}}$. Ce choix de l'exponentielle pour la probabilité s'appelle règle de Metropolis.

On itère ensuite selon ce procédé en gardant la température constante.

IV-2-4- Programme de recuit :

Deux approches sont possibles quant à la variation de la température :

1. Pour la première, on itère en gardant la température constante. Lorsque le système a atteint un équilibre thermodynamique (au bout d'un certain nombre de changements), on diminue la température du système. On parle alors de *paliers* de température.
2. La seconde approche fait baisser la température de façon continue. On peut imaginer toute sorte de loi de décroissance, la plus courante étant $T_{i+1} = \lambda T_i$ avec $\lambda < 1$ (assez couramment $\lambda = 0.99$).

Dans les deux cas, si la température a atteint un seuil assez bas fixé au préalable ou que le système devient figé, l'algorithme s'arrête.

La température joue un rôle important. À haute température, le système est libre de se déplacer dans l'espace des solutions ($e^{-\frac{\Delta E}{T}}$ proche de 1) en choisissant des solutions ne minimisant pas forcément l'énergie du système. À basse température, les modifications baissant l'énergie du système sont choisies, mais d'autres peuvent être acceptées, empêchant ainsi l'algorithme de tomber dans un minimum local.

IV-2-5- Étude théorique :

Des études théoriques du recuit simulé ont pu montrer que sous certaines conditions, l'algorithme du recuit convergeait vers un optimum global¹. Ce résultat est important car il nous assure, contrairement à d'autres métaheuristiques, que le recuit simulé peut trouver la meilleure solution, si on le laisse chercher indéfiniment.

Les preuves de convergence reposent sur le principe de construction de l'algorithme du recuit simulé :

- Sous réserve qu'elle respecte la condition de Doeblin² une chaîne de Markov admet une unique distribution stationnaire
- L'algorithme de Metropolis-Hastings permet, étant donnée une distribution de probabilité, de construire une chaîne de Markov dont la distribution stationnaire est cette distribution de probabilité.

Par ailleurs, les mesures de Gibbs, très utilisées en physique statistique, sont une famille de mesures dépendant d'un paramètre température de telle sorte que lorsque la température tend vers 0, la mesure converge vers un Dirac en un point donné (état d'énergie minimale).

En composant les deux, on obtient une chaîne de Markov qui converge vers une mesure invariante qui évolue dans le même temps vers une mesure désignant l'optimum recherché (au sens de la fonction d'énergie associée à la mesure de Gibbs).

Une analyse fine des vitesses de convergence et des écarts asymptotique permet alors d'aboutir à diverses conditions de convergence, dont celle de Hajek : si le schéma de température utilisé vérifie $T_n \geq \frac{K}{\ln n}$ alors l'algorithme converge en probabilité vers l'optimum global.

En pratique, ce schéma converge beaucoup trop lentement et l'on préfère des températures à décroissance linéaire ou quadratique, quitte à n'obtenir qu'un minimum local (qu'on espère proche du minimum global). [3]

IV-3- Optimisation par essaim particulaire :

L'optimisation par essais particuliers (*OEP*) est une métaheuristique d'optimisation, inventée par Russel Eberhart (ingénieur en électricité) et James Kennedy (socio-psychologue) en 1995.

Cet algorithme s'inspire à l'origine du monde du vivant. Il s'appuie notamment sur un modèle développé par Craig Reynolds à la fin des années 1980, permettant de simuler le déplacement d'un groupe d'oiseaux. Une autre source d'inspiration, revendiquée par les auteurs, est la socio-psychologie.

Cette méthode d'optimisation se base sur la collaboration des individus entre eux. Elle a d'ailleurs des similarités avec les algorithmes de colonies de fourmis, qui s'appuient eux aussi sur le concept d'auto-organisation. Cette idée veut qu'un groupe d'individus peu intelligents peut posséder une organisation globale complexe.

Ainsi, grâce à des règles de déplacement très simples (dans l'espace des solutions), les particules peuvent converger progressivement vers un minimum local. Cette métaheuristique semble cependant mieux fonctionner pour des espaces en variables continues.

Conclusion :

Après avoir vu des techniques d'optimisations comme les algorithmes génétiques, les recuit simulé et l'essaim particulaire ce qui nous mène qui passé au vif du sujet les PSO qu'on a bien détaillé dans le deuxième chapitre.

Chapitre 2 :

L'Optimisation par Essaim Particulaire.

I-Introduction:

Les ingénieurs se heurtent quotidiennement à des problèmes technologiques de complexité grandissante, qui surgissent dans des secteurs très divers, comme dans le traitement des images, la conception de systèmes mécaniques, la planification et l'exploitation des réseaux électriques,... etc.

Le problème à résoudre peut fréquemment être exprimé sous la forme générale d'un problème d'optimisation, dans lequel on définit une fonction objective, ou fonction Coût, que l'on cherche à minimiser (ou maximiser) par rapport à tous les paramètres concernés. La définition du problème d'optimisation est souvent complétée par la donnée de contraintes : tous les paramètres (ou variables de décisions) de la solution proposée doivent respecter ces contraintes, faute de quoi la solution n'est pas réalisable.

Il existe de nombreuses méthodes 'classiques' d'optimisation pour résoudre de tels problèmes, applicables lorsque certaines conditions mathématiques sont satisfaites : à savoir, la programmation linéaire traite efficacement le cas où la fonction objective, ainsi que les contraintes s'expriment linéairement en fonction des variables de décision. Dans le cas où la fonction objective et leurs contraintes sont non linéaires ; la programmation non linéaire est applicable. Malheureusement, les situations rencontrées en pratique comportent souvent une ou plusieurs complications, qui mettent en défaut ces méthodes : par exemple, la fonction objective peut être non homogène, ou même ne pas s'exprimer analytiquement en fonction des paramètres ; ou encore, le problème peut exiger la

considération simultanée de plusieurs objectifs contradictoires (Optimisation Multi-objectif).

L'arrivée d'une nouvelle classe de méthodes d'optimisation, nommées méthaheuristiques, marque une grande révolution dans le domaine de l'optimisation. En effet, celles-ci s'appliquent à toutes sortes de problèmes combinatoires, et elles peuvent également s'adapter aux problèmes continus.

Ces méthodes permettent de trouver une solution de bonne qualité en un temps de calcul en général raisonnable, sans garantir l'optimalité de la solution obtenue. Ces méthodes sont avantageusement utilisées pour la résolution des problèmes de grande taille. Les méthodes heuristiques peuvent être divisées en deux classes. Il y a, d'une part, les algorithmes spécifiques à un problème donné qui utilisent des connaissances du domaine, et d'autre part les algorithmes généraux qui peuvent être utilisés pour une grande variété de problèmes [3].

Les méthodes métaheuristiques sont apparues à partir des années 1980 [5], avec une ambition commune : résoudre au mieux les problèmes d'optimisation difficiles. Elles ont en commun les caractéristiques suivantes :

- Elles sont au moins pour une partie stochastiques : cette approche permet de faire face à l'explosion combinatoire des possibilités.
- Elles sont d'origine combinatoire : elles ont l'avantage décisif dans le cas continu d'être directes, c'est-à-dire qu'elles ne recourent pas au calcul souvent problématique des gradients de la fonction objective.

- Elles sont inspirées par des analogies : avec la physique (recuit simulé, diffusion simulée, etc.), avec la biologie (algorithme génétiques, recherche tabou, etc.) ou avec l'éthologie (colonies de fourmis, essaims de particules, etc.).
- Elles sont capables de guider dans une tâche particulière une autre méthode de recherche spécialisée (par exemple, une autre heuristique, ou une méthode d'exploration locale).
- Elles partagent aussi les mêmes inconvénients : les difficultés de réglage des paramètres mêmes de la méthode et le temps de calcul élevé.

L'optimisation mono-objectif se base sur la minimisation (ou la maximisation) d'une seule fonction objective laquelle ne reflète pas la physique réelle du système à optimiser. D'autre part, l'optimisation multi-objectif optimise simultanément plusieurs fonctions objectives qui sont souvent contradictoires.

La solution classique d'un problème d'optimisation multi-objectif consiste à la conversion de tous les objectifs en une seule fonction objective pondérée. La solution optimale de cette fonction reflète un compromis entre tous les objectifs. La conversion de plusieurs objectifs en un seul objectif est habituellement faite en agrégeant tous les objectifs dans une fonction pondérée, ou en optimisant une fonction objective et considérant les autres comme contraintes. Cette approche possède plusieurs limitations résumées dans les points suivants :

- Elle exige la connaissance a priori au sujet de l'importance relative des objectifs et les limites sur les objectives qui sont convertis en contraintes.
- La fonction singulière pondérée mène à une seule solution.
- Le compromis entre les objectifs ne peut pas être évalué facilement.
- L'espace de recherche doit être convexe.

Comparés aux problèmes d'optimisation mono-objective, les problèmes d'optimisation multi-objectif sont plus difficiles à résoudre car ils ne possèdent pas une seule solution mais un ensemble de solutions situé sur la frontière du Pareto. La génération de cette dernière donne plusieurs avantages :

- Elle contient plusieurs solutions optimales à la différence de l'optimisation mono-objectif qui ignore cette solution de compromis.
- L'ensemble de la frontière du Pareto permet aux décideurs de choisir la solution optimale suivant la situation rencontrée [3].

II- METHODES D'OPTIMISATION MULTI-OBJECTIF:

Il y a plusieurs méthodes proposées pour résoudre un problème d'optimisation multi- objectif. les méthodes d'inspiration naturelle sont de plus en plus souvent utilisées pour résoudre des problèmes où les méthodes classiques sont peu efficaces. En voici quelques-unes appartenant à la même famille:

- Optimisation par colonies de fourmis (AntColonyOptimization).
- Optimisation par algorithmes génétiques (AGs).
- Optimisation par essaims particuliers OEP (Particle Swarm Optimization, PSO)[6]

II -1 Optimisation par Essaims de Particules:

Un concept pour l'optimisation de fonctions non linéaires en utilisant la méthodologie des essaims de particules est introduite. L'évolution de plusieurs paradigmes est décrite, et une mise en œuvre de l'un des paradigmes est discutée. Des tests d'évaluation du paradigme sont décrits et les applications y compris l'optimisation de fonctions non linéaires et de la formation réseau de neurones sont proposées. Les relations entre l'optimisation par essaim de particules et les deux la vie artificielle et les algorithmes génétiques sont décrites. [7]

Optimisation par essaim particulaire (PSO) dans sa version historique est un collectif anarchique (au sens originel du terme) méthode itérative mettant l'accent sur la coopération il est en partie aléatoire et sans sélection. L'objectif des premiers chapitres seront au détail de ces caractéristiques et de les formaliser pour obtenir une modèle exploitable qui est particulièrement efficace pour des problèmes fortement non linéaires [8] .

L'origine de cette méthode vient des observations faites lors des simulations informatiques de vols groupés d'oiseaux et de bancs de poissons de REYNOLD, HEPPNER et GRENANDER. Ces simulations ont mis en valeur la capacité des individus d'un groupe en mouvement à conserver une distance optimale entre eux et à suivre un mouvement global par rapport aux mouvements locaux de leur voisinage.

D'autre part, ces simulations ont également révélé l'importance du mimétisme dans la compétition qui oppose les individus à la recherche de la nourriture. En effet, les individus sont à la recherche de sources de nourriture qui sont dispersés de façon aléatoire dans un espace de recherche, et dès lors qu'un individu localise une source de nourriture, les autres individus vont alors chercher à le reproduire.

Ce comportement social basé sur l'analyse de l'environnement et du voisinage constitue alors une méthode de recherche d'optimum par l'observation des tendances des individus voisins. Chaque individu cherche à optimiser ses chances en suivant une tendance qu'il modère par ses propres vécus [9].

Pour illustrer ce que coopération signifie en OEP, voici un exemple très simple. J'utilise ici volontairement une autre métaphore que celle des insectes sociaux, pour insister sur le fait que ce sont les principes qui comptent et non pas leur habillage, biologique ou autre.

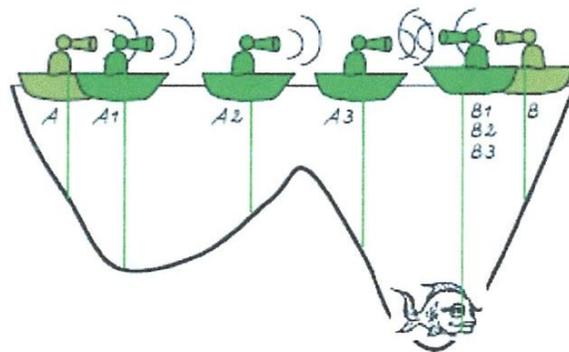


Figure 2.1 Exemple coopération

Comme d'habitude, le gros poisson est au plus profond de l'étang. A chaque pas de temps, chaque pêcheur indique à son collègue l'endroit le plus profond qu'il a déjà trouvé.

Au début, pas de différence, et chacun suit sa propre route indépendamment.

Ensuite, le pêcheur B a trouvé un meilleur emplacement, alors le pêcheur A tend à aller vers lui, assez vite. Ce faisant, il détériore sa position. Dès lors, il a un compromis à faire entre la meilleure position qu'il a trouvée jusque-là et celle trouvée par B. Il continue d'aller vers B, mais plus lentement, et de même le pas de temps suivant. Le point important est que, ce faisant, il échappe au minimum local.

Naturellement, c'est un exemple un peu caricatural, mais on y trouve les notions essentielles : position, vitesse (ou, plus précisément, un opérateur modificateur de position), échange d'informations, mémoire (très limitée), capacité rudimentaire à combiner les informations pour prendre une décision [10].

Nous avons choisi la méthode de l'OEP pour les raisons suivantes :

- Elle donne un bon résultat dans un temps de calcul plus court qu'avec d'autres méthodes.
- Il y a peu de paramètres à ajuster.
- Elle permet une large marge d'applications.
- Elle permet de trouver l'optimum de fonctions non-linéaires.
- Contrairement aux (AGs) et à d'autres algorithmes, l'OEP a une grande flexibilité de contrôle qui permet de gérer l'exploration globale et l'exploration locale de l'espace de recherche [6].

III- Présentation de la méthode :

III-1 Principe de la PSO:

L'optimisation par essaim de particules repose sur un ensemble d'individus originellement disposés de façon aléatoire et homogène, que nous appellerons dès lors des particules, qui se déplacent dans l'hyper-espace de recherche et constituent chacune une solution potentielle.

Chaque particule dispose d'une mémoire concernant sa meilleure solution visitée ainsi que la capacité de communiquer avec les particules constituant son entourage. À partir de ces informations, la particule va suivre une tendance faite, d'une part, de sa volonté à retourner vers sa solution optimale, et d'autre part, de son mimétisme par rapport aux solutions trouvées dans son voisinage.

À partir d'optimums locaux et empiriques, l'ensemble des particules va, normalement, converger vers la solution optimale globale du problème traité [9] .

- Elle a des agents avec des capacités de perception, mémorisation et calcul limitées.
- Elle est dynamique induite par des interactions locales.
- Observation de comportements globaux « émergents ».
- Une population d'agents (ici particules).
- Coopération plutôt que Compétition.
- Efficace sur un espace de recherche continu.
- Dynamiques définis sur un espace temps discret.

III -1.1 La PSO est efficace dans des domaines variés :

- Régulation de systèmes électriques.
- Conception d'ailes d'avions.
- Analyse d'images.

III -2 État et comportement d'une particule:

Modèle social simplifié:

- Mime le comportement d'une nuée d'oiseaux (ou banc de poissons).
- Basé sur les règles qui permettent de soudain changement de direction, dispersion, regroupement, etc.
- État (instantané) d'une particule:
- position (représente une solution du problème).
- vitesse (direction pour un futur déplacement).

Comportement :

- Se déplacer dans l'espace de recherche.
- dans le but de se positionner sur des solutions optimales.

III -3 Information utilisée par une particule pour déterminer sa prochaine position:

- Sa vitesse actuelle.
- Sa meilleure performance.
- La meilleure performance de ses voisines.

D'où trois types de comportement :

- égoïste : suivre sa propre voie.
- conservateur : revenir en arrière.
- panurgien : suivre aveuglement le meilleur de tous.

III -3.1 Compromis entre les trois types de comportement:

- Un déplacement est finalement une combinaison.
- Pondérée des trois types de mouvement ; pris en compte de :
- l'état instantané : position & vitesse (mémoire propre à court terme).
- performance personnelle (mémoire propre à long terme).
- performance des voisins (mémoire partagée).

Compromis psycho-social, entre confiance en soi et influence de ses relations sociales.

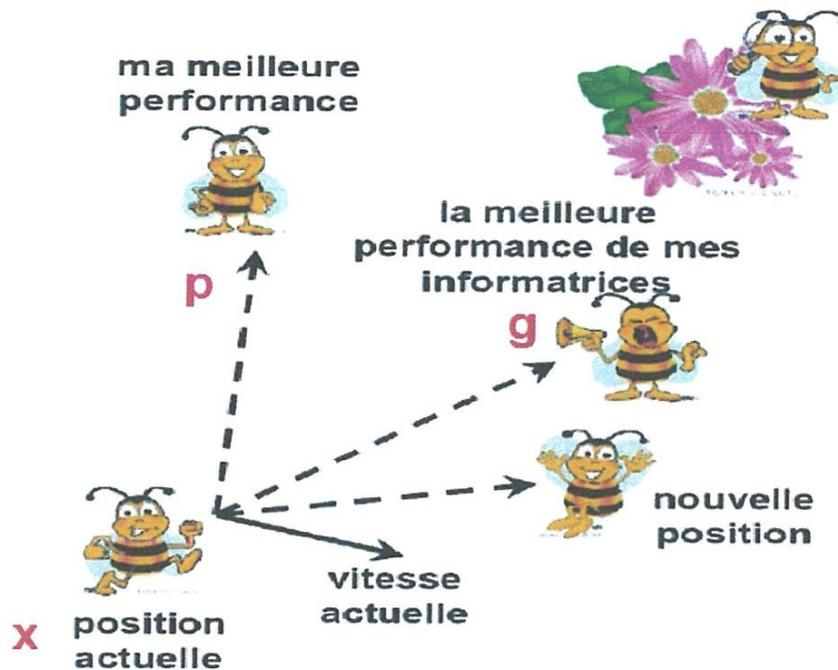
III -3.2 Trois types de comportement à pondérer:

Figure.2.2 Schéma de principe du comportement des particules d'un essaim.

Pour chaque particule et chaque dimension:

- Équation 1:

$$v(t+1) = w*v(t) + R(c)*(p(t)-x(t)) + R(c)*(g(t)-x(t))$$

- Équation 2:

$$x(t+1) = x(t) + v(t+1)$$

Où

- $v(t)$:= vitesse au temps t .
- $x(t)$:= position au temps t .
- $p(t)$:= la meilleure position précédente de la particule.
- $g(t)$:= la meilleure position parmi les meilleures positions précédentes des informateurs de la particule.
- $R(c)$:= un nombre venant d'une distribution aléatoire, qui dépend de « c » dans cette norme, la distribution est uniforme sur $[0, c]$.

III-3.2.1 Initialisation:

- 1 Particule = 1 solution du problème d'optimisation.
- Pour chaque particule :
 1. Fixer sa position au hasard dans l'espace de recherche.
 2. Fixer sa vitesse au hasard.
 3. Définir son voisinage:
 - géographique (distance), et donc dynamique.
 - Social, et donc fixe.

III-3.2.2 Définition du voisinage:

- Géographique
 - Nécessite une notion de distance.
 - Doit être mis à jour à chaque déplacement donc dynamique.
- Social
 - Fixé une fois pour toute à l'initialisation.

Fixé une fois pour toute à l'initialisation sur un cercle virtuel. En fait converge vers un

voisinage géographique.

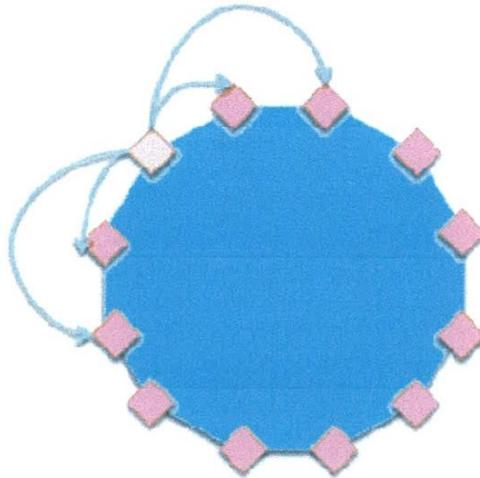


Figure.2.3 Le cercle virtuel pour un essaim.

On suppose toutes les particules disposées (symboliquement) en cercle et, pour la particule étudiée, on inclut progressivement dans ses informatrices, d'abord elle-même, puis les plus proches à sa droite et à sa gauche, de façon à atteindre le total

requis. Il y a bien sûr de nombreuses variantes, y compris celle consistant à choisir les informatrices au hasard.

Mais celle-ci est à la fois simple et efficace.

Une fois la meilleure informatrice détectée, la modification de la vitesse est une simple combinaison linéaire de trois tendances. A l'aide de coefficients de confiance :

- La tendance «aventureuse», consistant à continuer selon la vitesse actuelle.
- La tendance «conservatrice », ramenant plus ou moins vers la meilleure position déjà trouvée.
- La tendance « panurgisme», orientant approximativement vers la meilleure informatrice.

Les termes « plus ou moins» ou « approximativement» font référence au fait que le hasard joue un rôle, grâce à une modification aléatoire limitée des coefficients de confiance, ce qui favorise l'exploration de l'espace de recherche.

La figure II-2 présente un schéma de principe résumant les explications ci-dessus.

Naturellement, pour pouvoir être programmé, tout ceci est formalisé dans des équations de mouvement.

III-4 Points essentiels:

Inspiré par les interactions sociales ou les déplacements d'un groupe d'animaux.

Pour chaque particule :

- Persistance : durée de vie « illimitée ».
- Capacités de mémorisation de son état propre:
 - court terme.
 - long terme.
- Coopérer plutôt que concurrencé.
- Chacun est producteur et consommateur d'information.
- Capacité de perception de son environnement proche:
 - au sens géographique.
 - au sens social.
- Pas de mécanisme de sélection:
 - Une particule même actuellement médiocre est conservée.
 - elle permettra peut-être le succès futur du fait qu'elle « sort de sentiers battus».
- Chacune est « teacher et learner» [11] .

IV- Formalisation:

Un essaim de particule est caractérisé par:

- a) le nombre de particules de l'essaim, noté nb
- b) la vitesse maximale d'une particule, noté \vec{v}_{max}
- c) la topologie et la taille du voisinage d'une particule qui définissent son réseau social
- d) l'inertie d'une particule, notée Ψ
- e) les coefficients de confiance, notés ρ_1 et ρ_2 , qui pondèrent le comportement conservateur (ie. la tendance à retourner vers la meilleure solution visitée) et le panurgisme (ie. la tendance à suivre le voisinage)

Une particule est caractérisée, à l'instant t , par :

$\vec{x}_i(t)$: sa position dans l'espace de recherche

$\vec{v}_i(t)$: sa vitesse

\vec{x}_{pbest_i} : la position de la meilleure solution par laquelle elle est passée

\vec{x}_{vbest_i} : la position de la meilleure solution connue de son voisinage

$pbest_i$: la valeur de fitness de sa meilleure solution

$vbest_i$: la valeur de fitness de la meilleure solution connu du voisinage [9]

IV-1 Configuration de la méthode:

IV-1.1 Nombre de particules :

La quantité de particules allouées à la résolution du problème dépend essentiellement de deux paramètres :

la taille de l'espace de recherche et le rapport entre les capacités de calcul de la machine et le temps maximum de recherche. Il n'y a pas de règle pour déterminer ce paramètre, faire de nombreux essais permet de se doter de l'expérience nécessaire à l'appréhension de ce paramètre.

IV-1.2 Topologie du voisinage :

La topologie du voisinage défini avec qui chacune des particules va pouvoir communiquer. Il existe de nombreuses combinaisons dont les suivantes sont les plus utilisées :

- a) topologie en étoile : chaque particule est reliée à toutes les autres, ie. L'optimum du voisinage est l'optimum global ;
- b) topologie en anneau : chaque particule est reliée à n particules (en général, $n = 3$), c'est la topologie la plus utilisée ;
- c) topologie en rayon : les particules ne communiquent qu'avec une seule particule centrale.

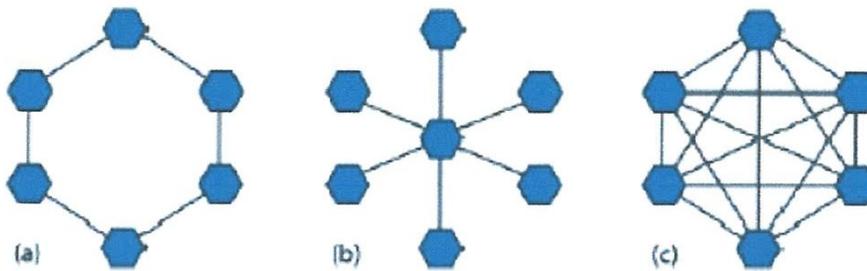


Figure. 2.4 (a) anneau (avec $n=2$), (b) rayon, (c) étoile

Le voisinage géographique auquel nous sommes amenés à penser en premier lieu n'est pas nécessairement pertinent car, d'une part, il s'agirait d'un voisinage trop local, et d'autre part car la sociabilisations des particules tend à rendre tout voisinage social en voisinage géographique.

Enfin, c'est un voisinage très lourd en terme de calculs car nécessitant de recalculer le voisinage de chaque particule à chaque itération.

IV-1.3 Coefficients de confiance:

Les variables de confiance pondèrent les tendances de la particule à vouloir suivre son instinct de conservation ou son panurgisme.

Les variables aléatoires r_1 et r_2 peuvent être définis de la façon suivante :

$$\begin{cases} p_1 = r_1 \cdot c_1 \\ p_2 = r_2 \cdot c_2 \end{cases}$$

Où r_1 et r_2 suivent une loi uniforme sur $[0;1]$ et c_1 et c_2 sont des constantes positives déterminées de façon empirique et suivant la relation $c_1 + c_2 \leq 4$.

IV-1.4 Vitesse maximale et coefficient de constriction:

Afin d'éviter que les particules ne se déplacent trop rapidement dans l'espace de recherche, passant éventuellement à côté de l'optimum, il peut être nécessaire de vitesse maximale (notée V_{\max}) pour améliorer la convergence de l'algorithme. Cependant, on peut s'en passer si on utilise un coefficient de constriction k - introduit par Maurice CLERC [12] et qui permet de resserrer l'hyper-espace de recherche.

$$K=1-\frac{1}{p}+\frac{\sqrt{|p^2-4p|}}{2}$$

Avec $p = p_1 + p_2 > 4$.

$$\begin{aligned}\vec{V}_i(t) &= k \cdot (\vec{V}_i(t-1)) \\ &+ p_1 \cdot (\vec{x}_{pbest_i} - \vec{x}_i(t)) \\ &+ p_2 \cdot (\vec{x}_{vbest_i} - \vec{x}_i(t))\end{aligned}$$

Les études de SHI et EBERHART indiquent que l'utilisation d'un coefficient de constriction donne généralement un meilleur taux de convergence sans avoir à fixer de vitesse maximale.

Cependant, dans certains cas, le coefficient de constriction seul ne permet pas la convergence vers la solution optimale pour un nombre d'itérations donné. Pour résoudre ce problème, il peut être intéressant de fixer $\vec{V}_{\max} = \vec{x}_{\max}$ en plus du coefficient de constriction ce qui, selon les études de SHI et EBERHART, permet d'améliorer les performances globales de l'algorithme.

IV-1.5 Facteur d'inertie:

Le facteur d'inertie Ψ — introduit par SHI et EBERHART — permet de définir la capacité d'exploration de chaque particule en vue d'améliorer la convergence de la méthode. Une grande valeur de $\Psi (>1)$ est synonyme d'une grande amplitude de mouvement et donc infinie d'exploration globale. Au contraire, une faible valeur de $\Psi (< 1)$ est synonyme de faible amplitude de mouvement et donc, d'exploration locale. Fixer ce facteur, revient donc à trouver un compromis entre l'exploration locale et l'exploration globale.

Le calcul de la vitesse est alors défini par :

$$\begin{aligned}\vec{V}_i(t) &= \Psi \cdot (\vec{V}_i(t-1)) \\ &+ p_1 \cdot (\vec{x}_{pbest_i} - \vec{x}_i(t)) \\ &+ p_2 \cdot (\vec{x}_{gbest_i} - \vec{x}_i(t))\end{aligned}$$

La taille du facteur d'inertie influence directement la taille de l'hyper -espace explore et aucune valeur de Ψ ne peut garantir la convergence vers la solution optimale.

Les études menées par SHI et EBERHART indiquent une meilleure convergence pour $\Psi \in [0.8; 1.2]$. Au delà de 1.2, l'algorithme tend à avoir certaines difficultés à converger.

D'après les recherches prouvées il est également possible de faire diminuer le facteur d'inertie au cours du temps, un peu à la manière de la température dans un algorithme de recuit simulé (SimulatedAnnealing). De bons résultats ont été trouvés pour une valeur décroissant linéairement de 0.9 à 0.4.

Pour de plus amples informations sur le réglage de ce paramètre, veuillez vous référer à la thèse de VAN DEN BERGH [13] .

IV-1.6 Initialisation de l'essaim:

La position des particules ainsi que leur vitesse initiale doivent être initialisées aléatoirement selon une loi uniforme sur $[0::1]$. Cependant, en ce qui concerne la position des particules, il est préférable d'utiliser un générateur de séquence de SOBOL qui est plus pertinent dans la disposition homogène des particules dans un espace de dimension n .

IV-1.7 Critères d'arrêt:

Comme indiqué précédemment, la convergence vers la solution optimale globale n'est pas garantie dans tous les cas de figure même si les expériences dénotent la grande performance de la méthode. De ce fait, il est fortement conseillé de doter l'algorithme d'une porte de sortie en définissant un nombre maximum d'itération (que nous noterons $nbiter_{max}$).

L'algorithme doit alors s'exécuter tant que l'un des critères de convergence suivant n'a pas été atteint :

- $nbiter_{max}$ a été atteint ;
- la variation de la vitesse est proche de 0 ;
- le fitness de la solution est suffisant.

IV-2 L'organigramme de principe de la méthode des essais particulières:

Schéma de principe de l'algorithme est donné par la figure suivante:

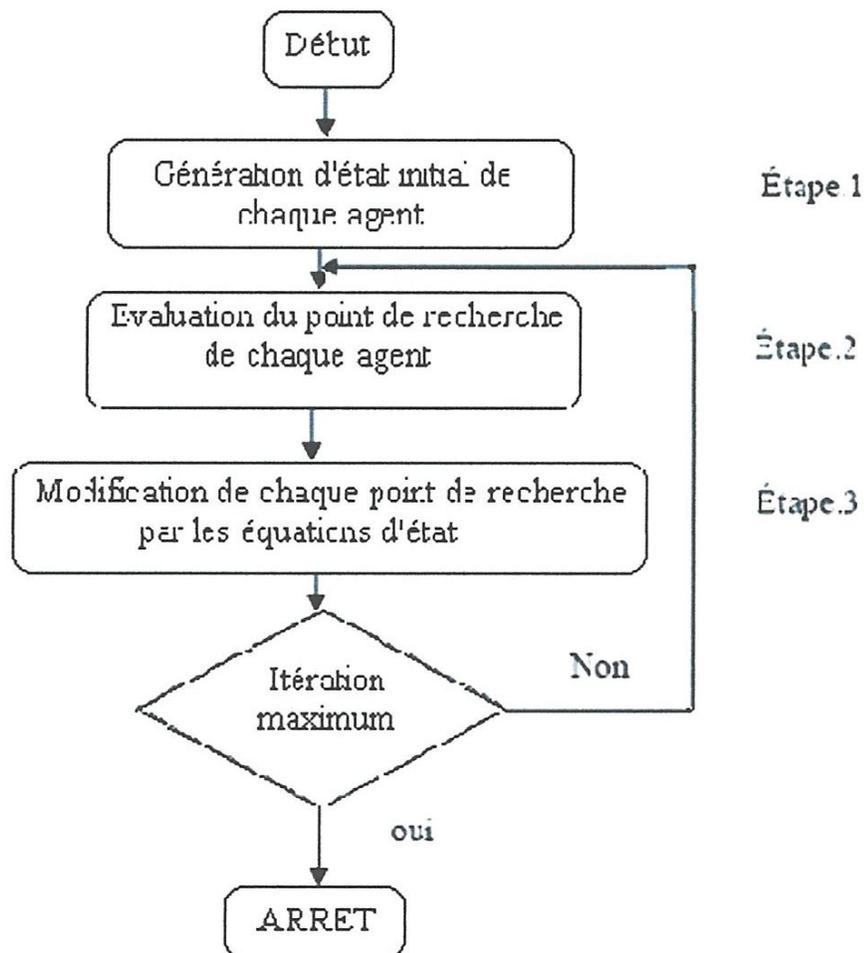


Figure.2.5 Organigramme de principe de la méthode des essais particulières.

Etape 1 : Génération d'état initial de chaque agent.

Etape2 : Evaluation du point de recherche de chaque agent

Les points de recherche courants sont évalués au moyen de la fonction objective relative au système d'étude. Si la valeur d'un agent est meilleure que son *pbest* courant, *pbest* prend cette nouvelle valeur. Si la meilleure valeur de *pbestest* meilleure que *gbest* courant, *gbest* est remplacé par cette meilleure valeur et l'agent qui correspond à cette meilleure valeur est ainsi stocké.

Etape 3 : Modification du point de recherche.

Etape 4 : Critère d'arrêt.

La procédure de recherche est arrêtée lorsque le nombre courant d'itération devient égal à la valeur maximale d'itérations préalablement déterminée. Dans ce cas de figure, la dernière valeur *gbest* peut être considérée comme solution. Dans l'autre cas, la recherche continue en revenant à l'étape 2.

CONCLUSION :

Après avoir présentés le principe, la formalisation et l'algorithme de notre (PSO) on déduit que c'est un outil important qui peut être utilisé pour l'optimisation de nombreux problèmes.

Et dans ce contexte on a proposé un programme matlab dans le chapitre qui suit avec résultats et simulation.

Chapitre 3: Résultats et
Simulation.

I-Introduction:

Dans ce chapitre nous allons présenter les résultats de simulation d'application des PSO avec quelque fonction tests de littérature.

Premièrement nous présenterons une synthèse du mécanisme de fonctionnement d'un PSO avec une fonction convexe, puis on présentera les résultats de l'application des PSO avec deux fonction test la fonction *Schwefel* et la fonction de *Ackley*.

II- Synthèse de fonctionnement d'un PSO:

Pour faire étudier le mécanisme de fonctionnement d'un PSO nous utilisons la fonction quadratique (convexe) suivante dans un intervalle de recherche initial $-50 < x_{1,2} < 50$

$$f(x) = x_1^2 + x_2^2 \quad 3.1$$

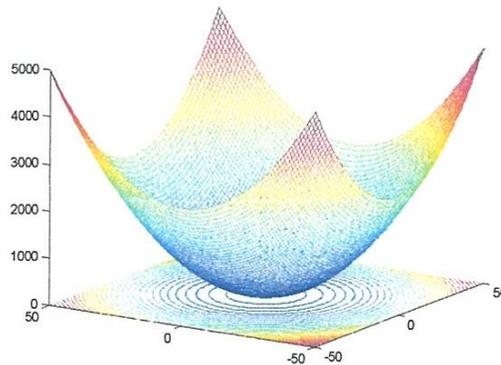


Figure 3.1 Représentation en 3D de la fonction quadratique f

Cette fonction a un seul minimum qui se trouve au point $(0, 0)$.

Pour l'algorithme utilisé nous avons adoptées les paramètres suivants:

$c1 = 0.12$, $c2 = 1.2$, $w = 0.9$. après l'application de l'algorithme pour cette fonction on a obtenu les valeurs suivantes : $f_{\min} = 3.1488e-007$, la solution optimale se trouve au point $x^* = (-0.0000 \quad -0.0054)$.

Les déplacements des particules pendant l'opération d'optimisation sont montrés dans la figure 3.2

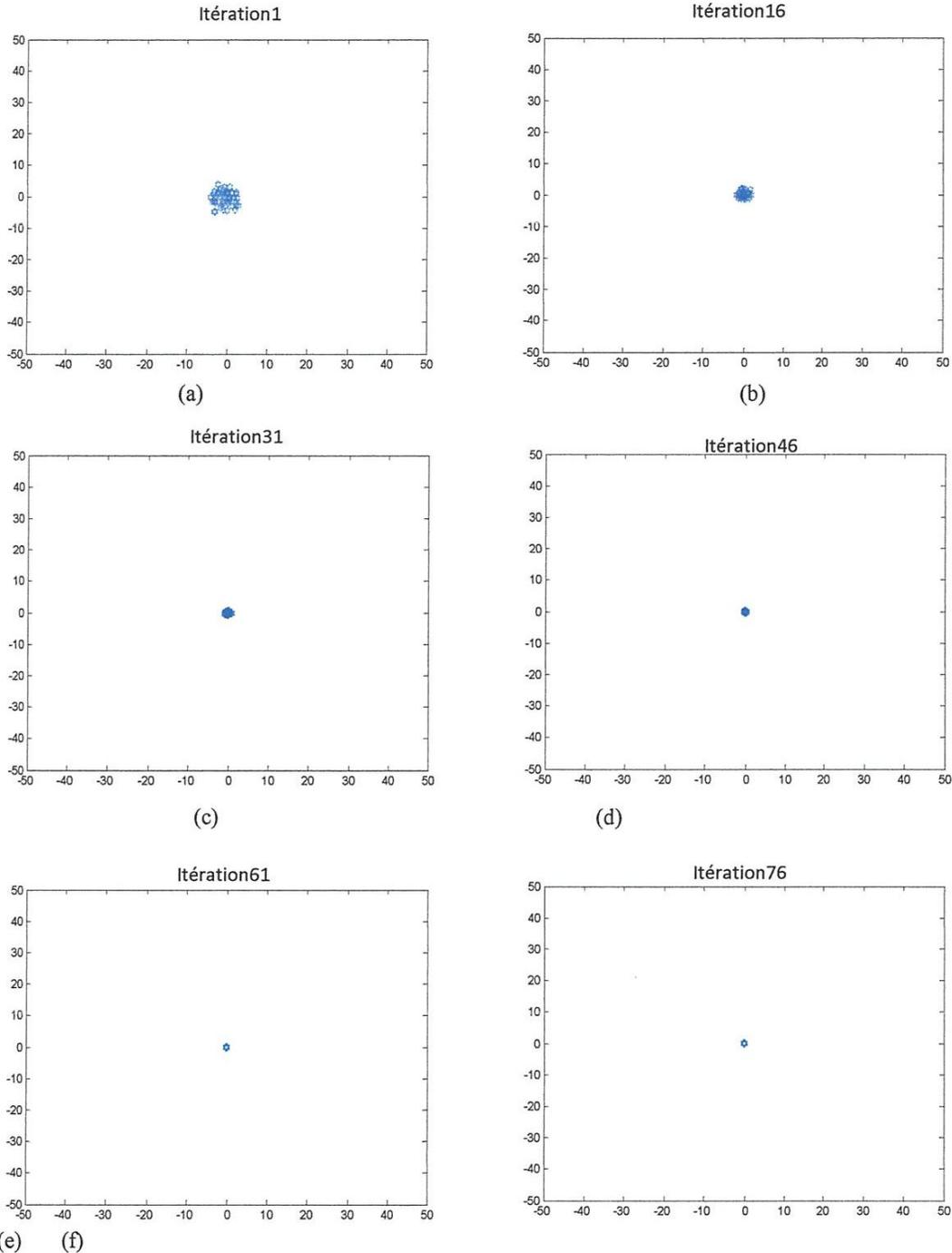


Figure 3.2 Déplacements des particules pendant l'opération d'optimisation par l'essai particulaire de la première fonction après chaque 15 itérations.

1. Pour voir l'effet des paramètres du PSO on a changé la valeur de c_2 comme suit :

$$c_1 = 0.12; c_2 = 3; w = 0.9.$$

L'exécution de l'algorithme nous donne les figures suivantes.

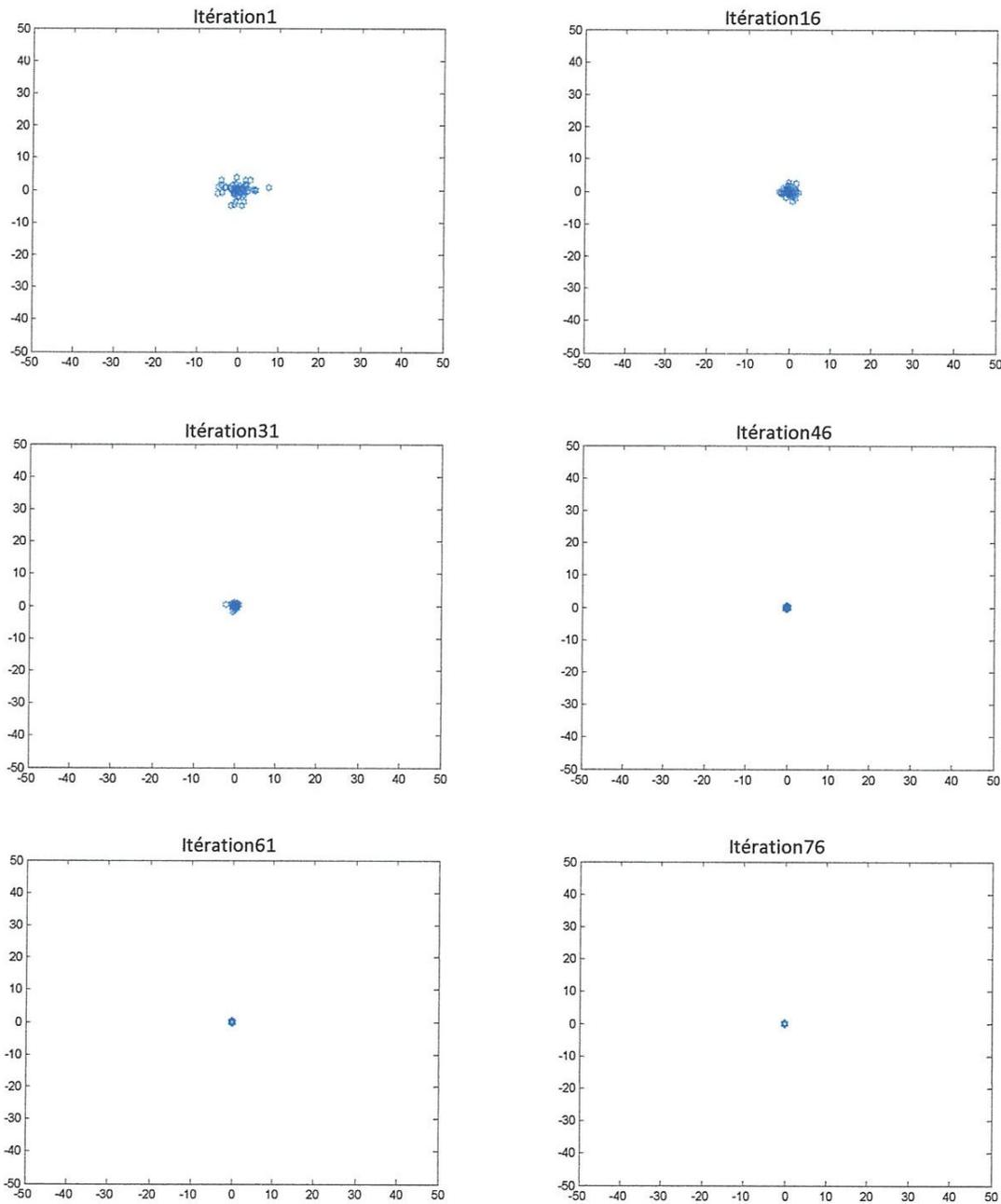


Figure 3.3 Déplacements des particules pendant l'opération d'optimisation par l'essai particulaire pour $C_2=3$.

D'après ces figures et surtout les deux figures après la 31^{ème} itération, on voit qu'une augmentation de la valeur de $c2$ affectera d'une façon négative la vitesse de convergence et la qualité de la solution obtenu. Et on obtenu les valeurs suivantes : $x^* = (-0.0125 \quad -0.0008)$ avec $f_{min} = 4.4366e-007$.

2. Maintenant, pour voir l'effet de $c1$ sur les performances de notre PSO nous changera ca valeur à $c1 = 2$; $c2 = 1.2$; $w = 0.9$. L'exécution de l'algorithme nous donne les résultats montrés dans la figure 3.4 et les valeurs suivantes : $x^* = (-0.0014 \quad 0.0008)$ avec $f_{min} = 1.3263e-007$

avec

$$f_{min} = 3.4251 \times 10^6$$

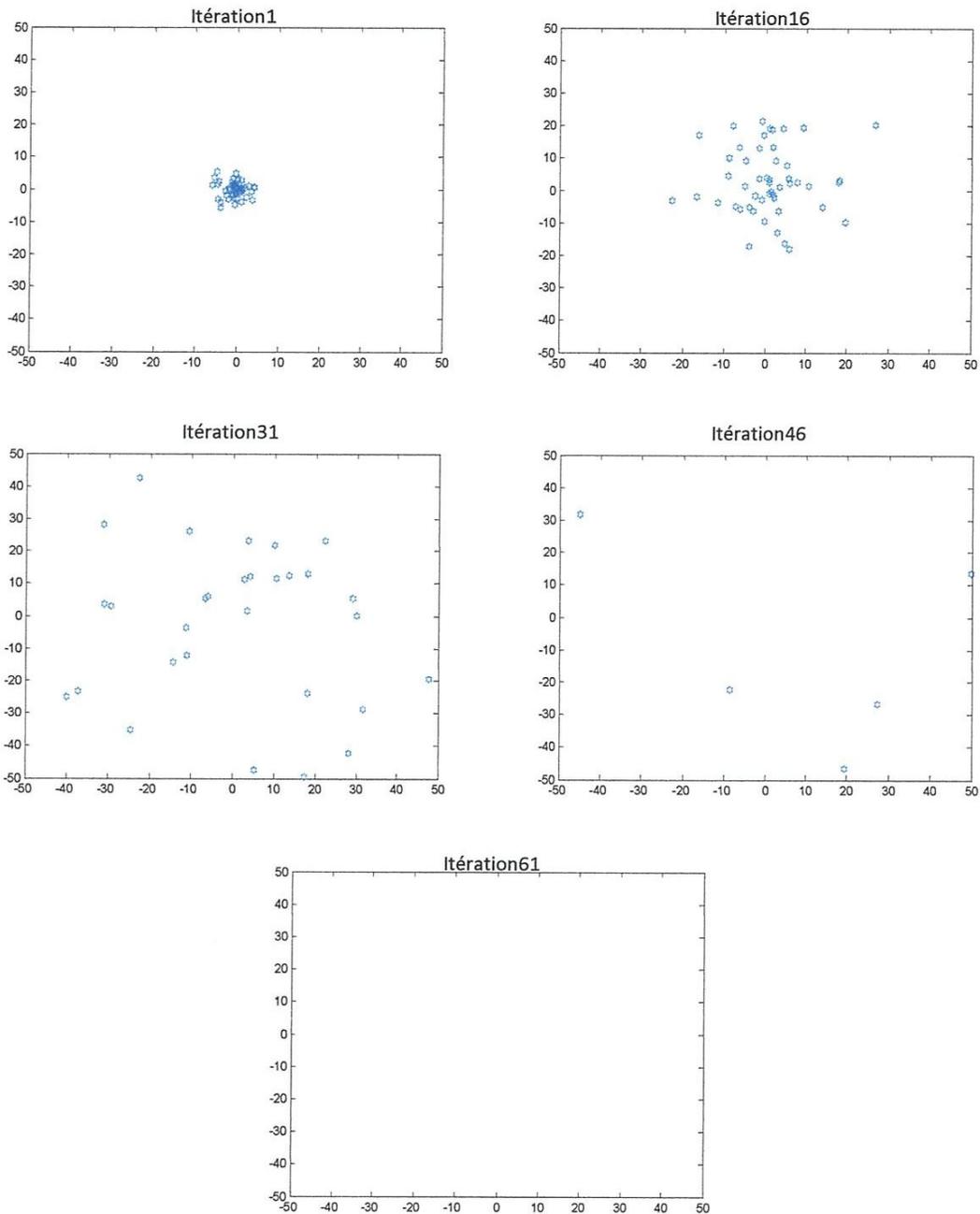


Figure 3.5 Déplacements des particules pendant l'opération d'optimisation par l'essaim particulaire pour $W=1.2$.

D'après ces figures on peut dire que l'augmentation du paramètre w amènera à une divergence globale de l'algorithme d'optimisation ce qui permet de conclure que ce paramètre augmente la sensibilité de l'algorithme.

III-Optimisation des fonctions test:

Basant sur les résultats de la partie précédente on a adopté les paramètres suivants:

$$c2 = 1.2;$$

$$c1 = 0.12;$$

$$w = 1.2;$$

- La première fonction est la fonction de **Ackley** a 2 variables donnée par :

$$f(x_i)_{i=1,2} = -2 \exp \left[-0.2 \sqrt{\frac{1}{2} \sum_{i=1}^2 x_i^2} \right] - \exp \left[\frac{1}{2} \sum_{i=1}^2 \cos(2.\pi.x_i) \right] + 2 + e \quad (3.2)$$

Où : $e=2.7183$

L'optimum global de cette fonction est situé au point : $x_i = 0$ avec $f(x^*) = 0$ et l'espace de recherche initial est choisi tel que :

$$\alpha_0 = \{-100 \leq x_i \leq 100\}$$

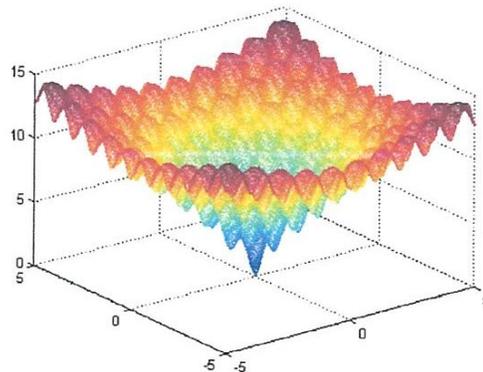


Figure 3.6 Représente la fonction de Ackley en dimension deux.

Optimisation de la fonction de *Ackley* à 2 variables:

L'utilisation des PSO dans l'optimisation de la fonction de *Ackley* à 2 variables donne les résultats indiqués par la figure 3.7 et les valeurs suivants :

$$x^* = (0.0251 \quad 0.1341) * 1.0^e-009 \text{ avec } f_{min} = 1.8172^e-005$$

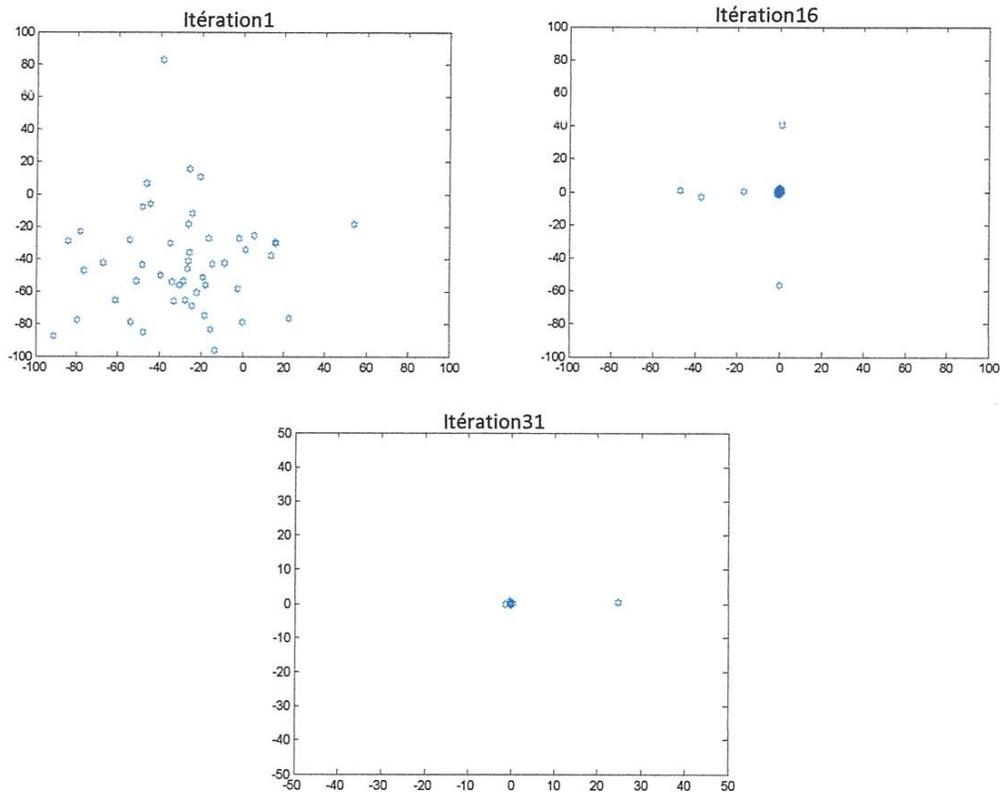


Figure 3.7 l'optimisation par essaim particulaire avec la fonction de Ackley à 2 variables.

- La deuxième fonction est la fonction de *Schwefel* a 2 variables donnée par :

$$f(x_i)_{i=1..2} = V + \sum_{i=1}^2 -x_i \sin(\sqrt{|x_i|}) \quad (3.3)$$

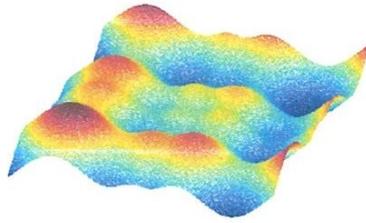


Figure 3.8 Représente la fonction de Shewefel en dimension deux.

L'optimum global de cette fonction est situé au point : $x_i = 65.5479$

avec $f(x^*) = -127.27$.

L'espace de recherche initial est choisi tel que :

$$\alpha_0 = \{-100 \leq x_i \leq 100\}$$

L'utilisation des PSO dans l'optimisation de la fonction de *Shewefel* à 2 variables donne les résultats indiqués par la figure 3.9 et les valeurs suivantes :

$x^* = (65.5479 \quad 65.5479)$ avec $f_{min} = -127.2700$

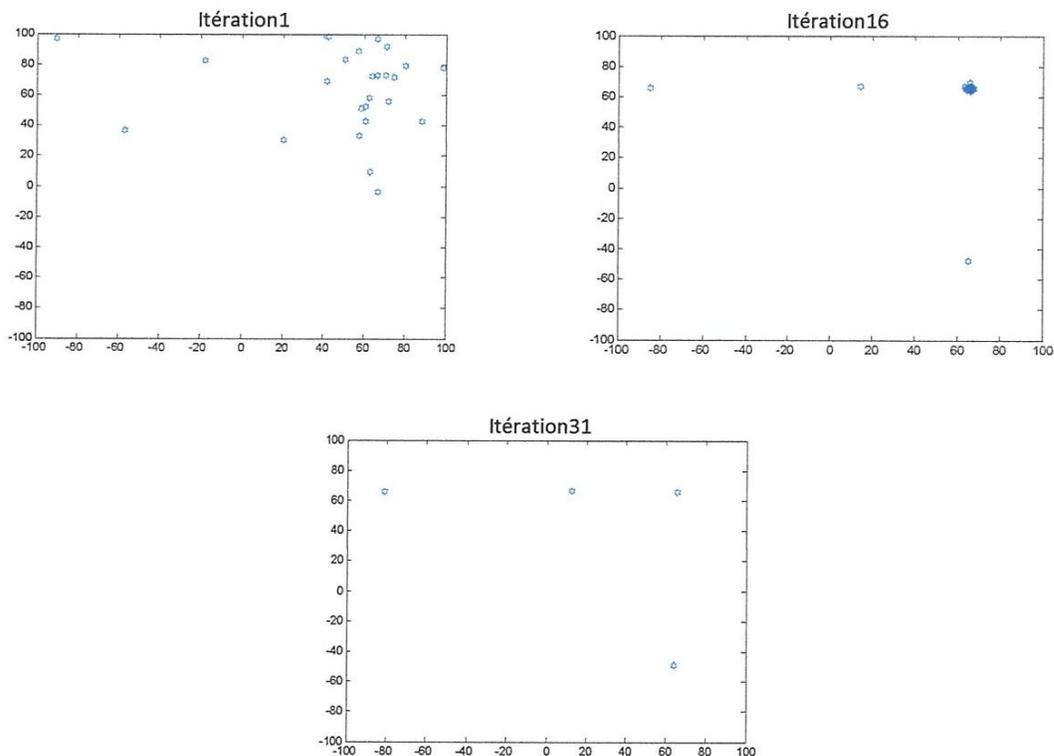


Figure 3.9 l'optimisation par essaim particulaire avec la fonction de Schewefel à 2 variables

IV-Conclusion :

Dans ce chapitre nous avons présentés une synthèse du mécanisme de fonctionnement d'un PSO avec une fonction convexe puis on a présenté les résultats de cette application avec deux fonction test la fonction *Schwefel* et la fonction de *Ackley*. On a conclu que tout changement d'un de leurs paramètres peut avoir un effet sur le fonctionnement de l'algorithme tout comme sur la solution obtenue.

Conclusion générale

Conclusion générale :

Dans ce travail, on a fourni un aperçue sur l'optimisation par essais particuliers qui est une nouvelle approche pour le contrôle et la coordination de systèmes multi-objectif. On a choisi la méthode de la PSO et on a discuté ce choix. Par la suite on a présenté le principe, la formalisation, la configuration et l'organigramme de la PSO. Cette méthode est reconnue comme un outil efficace pour les optimisations dans de nombreux problèmes.

On peut déduire que l'approche d'estimation de paramètres en utilisant la PSO a été prouvée comme performante pour sa précision et son temps de calcul très court ce qui la rend très puissante.

Bibliographie

Bibliographie :

- [1]-Yann COOREN « Dr en sciences » « Perfectionnement d'un algorithme adaptatif d'Optimisation par Essaim Particulaire.Applications en génie médical et en électronique.
- [2]-Hela BOUKEF BEN OTHMAN « doctorat Sur l'ordonnancement d'ateliers job-shop flexibles et flow-shop en industries pharmaceutiques Optimisation par algorithmes génétiques et essais particuliers».
- [3]-Rabah BENABID «Optimisation Multi objectif de la Synthèse des FACTS par les Particules en Essaim pour le Contrôle de la Stabilité de Tension des Réseaux Electriques».
- [4]-Projet d'ingénieur d'Institut Supérieur d'Informatique de Modélisation et de leurs Applications L'optimisation par essaim particulaire pour des problèmes d'ordonnancement.
- [5]- Y. Collette et P. Siarry (2002). " Optimisation Multi-objectif," EYROLLES.
- [6]- Eid Abdelbaki Ahmed GOUDA. (2011). Thèse présentée en vue de l'obtention du grade de Doctorat: "Transmission Planétaire Magnétique Etude, Optimisation ET Réalisation "l'Université Henri Poincaré,Nancy I pp 78.
- [7]- Kennedy, J.; Eberhart, R. (1995). "Particle Swarm Optimization".Proceedings of IEEE International Conference on Neural Networks. IV. pp. 1942–1948.
- [8]- Maurice Clerc(2006) ." Particle Swarm Optimization" . ISTE (International Scientific and Technical Encyclopedia) ,p. 19
- [9]- Guillaume CALAS. " Optimisation par essaim de particules " .http://guillaume.calas.free.fr/data/Publications/PSO-Overview_v2.pdf
- [10]- Maurice Clerc (2003). " L'optimisation par essaim particulaire". France Télécom R&D ,Tutorial pour OEP
- [11] BOUZID Allal El Moubarek (2008-2009), " Optimisation par la méthode des essais particuliers D'une fonction trigonométrique ". Ecole Normale Supérieure D'enseignement Technique , Oran .

Bibliographie

[12] Maurice Clerc and James Kennedy (2002). " The particle swarm - explosion, stability, and

convergence in a multidimensional complex space ". IEEE Trans. Evolutionary Computation,6(1) :58–73,.

[13] F. van den Bergh (2002) ." An Analysis of Particle Swarm Optimizers ". PhD thesis, Department of Computer Science, University of Pretoria, November.