

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université de Tébessa  
Faculté des Sciences Exactes et de Sciences de la Nature et de la Vie  
Département de mathématiques et Informatique



# MÉMOIRE

Présenté en vue de l'obtention du diplôme de  
**Magister en informatique**  
Option : Ingénierie des Systèmes Informatiques  
Présenté Par : **KHAMER Lazhar**  
Encadré par : Docteur **KHABABA Abdellah**

## Adaptation des méthodes d'optimisation par les Essaims de particules aux problèmes multi- objectifs

*Devant le jury composé de :*

- Président : Dr. TOUAHRIA MOHAMED  
Maître de conférences .A.  
Univ. FERHAT ABBES - SETIF
- Rapporteur : Dr. KHABABA ABDELLAH  
Maître de conférences .A.  
Univ. FERHAT ABBES - SETIF
- Examineur : Dr. MOUSSAOUI ABELOUAHEB  
Maître de conférences .A.  
Univ. FERHAT ABBES - SETIF
- Examineur : Dr. KAZAR OKBA  
Maître de conférences .A.  
Univ. M KHIDER – BISKRA

**Année : 2011**

# Remerciements

*Mes remerciements les plus vifs , à mon Encadreur Monsieur KHABABA ABDELLAH , Maître de conférence à l'université de Sétif pour son aide , son orientation judicieuse et sa disponibilité , aussi pour la confiance , la patience et la compréhension qu'il m'a toujours manifesté ...*

*je tien également à remercier Monsieur TOUAHRIA MOHAMED , Maître de conférence à l'université de Sétif pour avoir accepté de présider ce jury.*

*je tien également à remercier Monsieur MOUSSAOUI ABDELOUAHAB , Maître de conférence à l'université de Sétif pour avoir accepté de participer à ce jury.*

*je tien également à remercier Monsieur KAZAR OKBA , Maître de conférence à l'université de Biskra pour avoir accepté de participer à ce jury*

*Je tien également à remercier tous mes amis sans exception, ils sont nombreux, je ne peux tous les citer mais je ne les oublis.*

*«Une personne qui n'a jamais commis d'erreur ...*

*...n'a jamais tenté d'innover »*

*Albert Einshtein*

# *Dédicace*

Je dédie ce travail à ma mère pour ces encouragements et ces prières tout au long de mes études. A mon père qui nous a planté l'amour de science.

Je le dédie à mes frères et sœurs, et je les remercie pour leurs encouragements et leurs aides ainsi que tout ma grande famille.

## Abstract

The métaheuristics are a family of stochastic algorithms intended to solve difficult optimization problems. Used in many fields, these methods have the advantage of being generally effective, without modifying the basic structure of the algorithm. Among those, Particle Swarm Optimization (PSO) is a new class of algorithms suggested to solve the problems with continuous variables. The algorithms of PSO take as a starting point the social behavior of the animals evolving/moving in swarm, such as the migratory birds or fish. The " particles " of the same swarm communicate between them in a direct way to build a solution to the arising problem, while being based on their collective experiment.

Recognized since many years for their effectiveness, the Métaheuristics present defects which still discourage certain users. The adjustment of the parameters of the algorithms is one of those problems. It is significant, for each arising problem, to find the set of parameters which leads to optimal performances of the algorithm. However, this task is tiresome and expensive in time, especially for the beginners users. To free the user from this type of adjustment, a researches were carried out to propose algorithms known as " adaptive ". With these algorithms, the values of the parameters are not any more solidified, but are modified, according to the results collected during the process of research. In this optics, Maurice Clerc proposed TRIBES, which is an algorithm of mono-objective PSO without any parameters of control. This algorithm functions like a " black box ", for which the user has only to define the problem to be treated and the stop criterion of the algorithm.

We propose in this subject a study on the adjustment of the parameters for PSO in order to correct some of these defects.

The growing need for methods of resolution of multiobjectifs problems led the designers to adapt their methods to solve this type of problem. The complexity of this operation comes owing to the fact that the objectives to be optimized are often contradictory. We also propose in this subject to apply PSO to some problems of optimization such as image processing (the problem of medical images thresholding, medical images segmentation, medical images aligning).

**Keys Words:** multi-objectif optimisation, particle swarm optimisation, adaptatif algorithm, image segmentation, image aligning.

## Résumé

Les méta-heuristiques sont une famille d'algorithmes stochastiques destinés à résoudre des problèmes d' "optimisation difficile ". Utilisées dans de nombreux domaines, ces méthodes présentent l'avantage d'être généralement efficaces, sans pour autant que l'utilisateur ait à modifier la structure de base de l'algorithme qu'il utilise. Parmi celles-ci, l'Optimisation par Essaim Particulaire (OEP) est une nouvelle classe d'algorithmes proposée pour résoudre les problèmes à variables continues. Les algorithmes d'OEP s'inspirent du comportement social des animaux évoluant en essaim, tels que les oiseaux migrateurs ou les poissons. Les " particules " d'un même essaim communiquent de manière directe entre elles tout au long de la recherche pour construire une solution au problème posé, en s'appuyant sur leur expérience collective.

Reconnues depuis de nombreuses années pour leur efficacité, les méta-heuristiques présentent des défauts qui rebutent encore certains utilisateurs. Le réglage des paramètres des algorithmes est un de ceux-ci. Il est important, pour chaque problème posé, de trouver le jeu de paramètres qui conduise à des performances optimales de l'algorithme. Cependant, cette tâche est fastidieuse et coûteuse en temps, surtout pour les utilisateurs novices. Pour s'affranchir de ce type de réglage, des recherches ont été menées pour proposer des algorithmes dits " adaptatifs ". Avec ces algorithmes, les valeurs des paramètres ne sont plus figées, mais sont modifiées, en fonction des résultats collectés durant le processus de recherche. Dans cette optique-là, Maurice Clerc a proposé TRIBES, qui est un algorithme d'OEP mono-objectif sans aucun paramètre de contrôle. Cet algorithme fonctionne comme une " boîte noire ", pour laquelle l'utilisateur n'a qu'à définir le problème à traiter et le critère d'arrêt de l'algorithme.

Nous proposons dans ce sujet une étude sur le réglage des paramètres pour PSO afin de corriger certains de ces défauts.

Le besoin croissant de méthodes de résolution de problèmes multi-objectifs a conduit les concepteurs à adapter leurs méthodes pour résoudre ce type de problème. La complexité de cette opération provient du fait que les objectifs à optimiser sont souvent contradictoires. Nous proposons aussi dans ce sujet d'appliquer les versions d'algorithmes d'optimisation multi-objectifs sur quelques problèmes d'optimisation tel que traitement d'image (le problème de seuillage d'images médicales, segmentation des images, recalage des images médicales).

**Mots-clés** : optimisation multi-objectif, optimisation par essaim particulaire, algorithme adaptatif, segmentations des images, recalage des images.

## ملخص

المستكشفة هي عائلة خوارزميات عرضية مخصصة لحل مسائل التحسين الصعبة. من إيجابيات هذا الأسلوب أنه فعال إضافة إلى الإمكانية المتاحة للمستعمل لتعديل البنية الأساسية للخوارزمية ولهذا فإن هذا النوع مستثمر في عدة ميادين, و من بين هذه الخوارزميات ما تسمى بـ "التحسين بالمجموعات الجزئية" وهي نوع جديد مقترح لحل المسائل ذات المتغير المستمر. استلهمت خوارزميات التحسين بالمجموعات الجزئية، من النمط المعيشي للحيوانات الاجتماعية كالطيور المهاجرة و الأسماك. يحتك أفراد المجموعة ببعضهم بطريقة مباشرة طول طريق البحث لخلق حل للمشاكل الطارئة انطلاقا من التجربة الاجتماعية. عرفت، الخوارزميات المستكشفة، منذ أعوام عديدة بفعاليتها , لكنها أظهرت نقائص أزجت بعض المستعملين, تعديل معطيات الخوارزمية واحدة من هذه النقائص.

من المهم لكل مشكلة إيجاد مجموعة المعطيات التي تعطي أفضل فعالية للخوارزمية, لكن هذه العملية الدقيقة مكلفة من ناحية الوقت خاصة للمبتدئين. لأجل تحرير هذا النوع من العمل أي، تعديل المعطيات القاعدية للخوارزمية حتى إيجاد المعطيات المثالية، اقترحت الأبحاث نوع من الخوارزميات أطلق عليه اسم "الخوارزميات المتكيفة"، مع هذا النوع قيمة المعطيات لم تعد جامدة بل قابلة للتعديل بالاعتماد على النتائج المتحصل عليها أثناء سير البحث.

على ضوء ما تقدم اقترح، موريس و كلارك، ما يسمى بـ "القبائل"، التي تعد من أصناف خوارزميات التحسين بالمجموعات الجزئية وحيدة الهدف، لكن دون معطيات التحكم, نستطيع تمثيل آلية عملها بـ "العلبة السوداء"، المستعمل ليس عليه سوى تحديد المشكل المراد معالجته و شرط توقف الخوارزمية. لأجل تفادي بعض النقائص طرحنا في هذا الموضوع دراسة لتعديل معطيات من تصحيح بعض الأخطاء.

الحاجة المتزايدة لطرائق حل للمشاكل المتعددة الأهداف قادت المبتكرين لاعتماد هذا النموذج تعقيد هذه العملية يأتي أساسا من أن الأهداف المراد تحسينها غالبا ما تكون متناقضة. اقترحنا أيضا في هذه المذكرة تطبيق خوارزميات التحسين بالمجموعات الجزئية، على بعض معضلات التحسين مثل معالجة الصور، مشكل عزل الصورة الطبية، تجزئة الصورة، ترسيب الصور الطبية.

**الكلمات المفتاحية:** التحسين المتعدد الأهداف, خوارزميات التحسين بالمجموعات الجزئية, الخوارزمية المتكيفة, تجزئة الصور, ترسيب الصور.

# Table des matières

---

## Résumé

<b>Table des matières.....</b>	<b>VII</b>
<b>Listes des Figures.....</b>	<b>XII</b>
<b>Listes des Tableaux.....</b>	<b>XIII</b>
<b>Listes des Algorithmes.....</b>	<b>XIV</b>

<b>Introduction général .....</b>	<b>1</b>
-----------------------------------	----------

## **Chapitre 1: Introduction à l'optimisation par Essaim de particules**

1.1 Introduction .....	5
1.2 OEP standard .....	5
1.2.1 Description informelle.....	5
1.2.2 Formalisation et programmation.....	6
1.3 Graphe d'influence et type de voisinage.....	9
1.3.1 Voisinage statique.....	9
1.3.2 Voisinage dynamique aléatoire.....	10
1.3.3 Voisinage basé sur une distance euclidienne.....	10
1.3.4 Connexité temporelle de la diffusion de l'information et OEP multi- essaim.....	11
1.4 Stratégie de déplacement et distribution de la proximité .....	11
1.4.1 La distribution rectangulaire de l'OEP minimale.....	12
1.4.2 Distribution à secteurs ellipsoïdaux positifs .....	14
1.4.3 Gaussiennes indépendantes.....	15
1.4.4 Locale par gaussiennes indépendantes.....	15
1.4.5 La méthode des pivots.....	15
1.4.6 La méthode des pivots bruitée.....	16
1.5 Sélection des paramètres de l'algorithme.....	16
1.5.1 La vitesse maximale .....	16
1.5.2 L'inertie .....	17
1.5.3 Robustesse et carte de performance .....	17
1.5.4 Stabilité et convergence de l'algorithme .....	17

1.6	Algorithme OEP à convergence garantie.....	18
1.7	Cyber Swarm Algorithmes (CSA) .....	19
1.8	Implémentation séquentielle vis-à-vis à l'implémentation parallèle.....	20
1.9	Domaine d'applications de la méthode OEP.....	20
1.10	La recherche actuelle en matière OEP.....	21
1.10.1	Le projet XPS.....	22
1.10.2	L'axe physique.....	22
1.10.3	L'axe théorique.....	22
1.10.4	L'axe biologique.....	22
1.10.5	L'axe ingénierie.....	23
1.11	Conclusion.....	23

## **Chapitre2: L' optimisation Multi-objectif par Essaim de Particules**

2.1	Introduction .....	25
2.2	Définitions.....	25
2.2.1	Problème multi objectifs.....	25
2.2.2	Le vecteur idéal.....	27
2.2.3	Convexité .....	27
2.3	Multiplicité de solutions .....	27
2.4	Difficulté d'un problème multi objectifs.....	28
2.5	Classification des méthodes d'optimisation multi objectifs.....	29
2.5.1	Les méthodes agrégées .....	29
2.5.2	Avantages et inconvénients des modèles agrégées .....	30
2.5.3	Les méthodes non agrégées non Pareto .....	30
2.5.4	Les méthodes Pareto .....	30
2.5.4.1	La notion de dominances.....	31
2.5.4.2	La notion de Pareto optimal.....	32
2.5.4.3	Le front de Pareto.....	32
2.6	Etat de l'art de l'optimisation par l'essaim des particules en optimisation multi-objectifs .....	32
2.6.1	Principe général de l'OEP multi-objectifs.....	33



2.6.2 Les difficultés d'adaptation.....	35
2.6.3 Leaders dans l'optimisation multi-objectifs .....	35
2.6.4 Estimateur de densité de voisin le plus proche .....	36
2.6.5 Estimateur de densité de grain .....	36
2.6.6 Conservation et propagation des solutions non-dominées .....	37
2.6.7 Maintien de la diversité par création de nouvelles solutions .....	39
2.6.8 Classification des approches OEP pour les problèmes multi-objectifs .....	41
2.6.8.1 OEP multi-objectif pour l'approche agrégative.....	41
2.6.8.2 OEP multi-objectif pour l'approche Non-Pareto.....	43
2.6.8.3 OEP multi-objectif pour l'approche de Pareto.....	45
2.7 OEPMO par Agrégation dynamique DWA basé sur trois techniques d'adaptation du facteur d'inertie.....	48
2.7.1 Etude expérimentale.....	49
2.7.1.1 Fonctions de testes.....	49
2.7.1.2 Résultats numériques et discussion.....	49
2.7.2 Un nouvel algorithme MOOEP agrégatif à base de trois technique d'adaptation.....	58
2.8 Conclusion .....	59
 <b>Chapitre3: L'Optimisation par Essaim des particules adaptatif</b>	
3.1 Introduction .....	62
3.2 Description de TRIBES .....	62
3.2.1 La tribu .....	62
3.2.2 Indicateurs de qualité .....	64
3.2.2.1 La qualité d'une particule.....	64
3.2.2.2 La qualité d'une tribu.....	64
3.2.3 Evolution des tribus .....	64
3.2.3.1 La suppression des particules .....	64
3.2.3.2 Génération des particules .....	65
3.2.3.3 La particule libre .....	66
3.2.3.4 Les particules confinées .....	66
3.2.3.5 Fréquence des adaptations.....	67
3.2.3.6 Evolution de l'essaim.....	67

3.3 Adaptation structurelle et stratégies de déplacement.....	68
3.3.1 Méthode des pivots simple.....	69
3.3.2 Méthode des pivots bruitée.....	70
3.3.3 Locale par gaussiennes indépendantes.....	70
3.4 Algorithme.....	71
3.5 Amélioration de TRIBES.....	72
3.6 Conclusion.....	73

## **Chpître4:Application de la méthode d'essaim particulaire Au recalage des images multimodales**

4.1 Introduction .....	75
4.2 Présentation générale du problème de recalage et notation.....	76
4.3 Principe générale et composants du recalage.....	76
4.3.1 Description formelle .....	77
4.3.2 La technique d'interpolation .....	78
4.3.2.1 Passage au domaine discret.....	78
4.3.2.2 Définition de l'interpolation .....	79
4.3.2.3 L'interpolation dans le processus de recalage d'images.....	80
4.4 Le type de transformation .....	81
4.4.1 Transformation Rigides .....	81
4.4.2 Transformation flexibles.....	81
4.5. Critères de similarité.....	83
4.5.1 Critères de similarité à base des primitives géométriques.....	84
4.5.1.1 Mesures de similarités à base de points.....	84
4.5.1.2 Mesures de similarités à base des ensembles de points.....	84
4.5.2 Critères de similarité iconiques.....	85
4.5.2.1 Relation d'identité.....	85
4.5.2.2 Relation linéaire.....	86
4.5.2.3 Relation fonctionnelle.....	86
4.5.2.4 Relation de dépendance statistique.....	87
4.6 Méthodes d'optimisation .....	87

4.7 utilisation d'OEP dans le recalage des images multimodales.....	88
4.7.1 Une approche OEP avec une orientation initiale de l'utilisateur.....	88
4.7.2 Une approche OEP à base d'une représentation multi-résolutions de l'image.	90
4.8 Une approche OEP à facteur d'inertie adaptatif pour le recalage des images IRM multimodales .....	90
4.8.1 Problématique.....	90
4.8.2 Histogramme d'une image.....	91
4.8.2.1 Histogramme marginal.....	91
4.8.2.2 Entropie d'une image.....	92
4.8.2.3 Histogramme conjoint.....	92
4.8.2.4 Entropie conjointe de deux images .....	92
4.8.3 Information mutuelle.....	93
4.8.4 Normalisation de l'information mutuelle.....	94
4.8.5 Vers un algorithme OEP avec facteur d'inertie adaptatif.....	94
4.8.6 Expérimentation globale.....	97
4.8.7 résultats numériques et discussion.....	98
4.9 conclusion .....	100
<b>Conclusion générale.....</b>	<b>101</b>
<b>ANNEXE A.....</b>	<b>103</b>
<b>ANNEXE B.....</b>	<b>104</b>
<b>BIBLIOGRAPHIE.....</b>	<b>110</b>

# Listes des figures

---

<b>Figure 1.1</b> : Schéma de principe du déplacement d'une particule	7
<b>Figure 1.2</b> : Graphe d'influence statique.	10
<b>Figure 1.3</b> : Les deux distributions élémentaires de la proximité	12
<b>Figure 1.4</b> : Ensemble des positions atteignables résultants de la somme des proximités	13
<b>Figure 1.5</b> : Biais de la distribution	13
<b>Figure 1.6</b> : Dépendance vis-à-vis du système de coordonnées	14
<b>Figure 2.1</b> : Ensembles des actions réalisables dans un problèmes multi-objectifs	26
<b>Figure 2.2</b> : Ensembles des objectifs réalisables dans un problèmes multi-objectifs	26
<b>Figure 2.3</b> : (a) espace convexe et (b) espace non convexe	27
<b>Figure 2.4</b> : Modes de résolution d'un problème multi objectif	28
<b>Figure 2.5</b> : Notion de la dominance	31
<b>Figure 2.6</b> : Le front de Pareto	32
<b>Figure 2.7</b> : Exemple d'estimateur de densité de voisin le plus proche	36
<b>Figure 2.8</b> : Niches de particules	37
<b>Figure 2.9</b> : Exemple d'utilisation de $\epsilon$ dominance dans un archive externe	38
<b>Figure 2.10</b> : La topologie en anneau du schéma de migration utilisé par l'algorithme VEPSO composé de 5 sous essaims	44
<b>Figure 2.11</b> : Résultats des différentes approches d'adaptation du facteur d'inertie pour F1	53
<b>Figure 2.12</b> : Résultats des différentes approches d'adaptation du facteur d'inertie pour F2	54
<b>Figure 2.13</b> : Résultats des différentes approches d'adaptation du facteur d'inertie pour F3	55
<b>Figure 2.14</b> : Résultats des différentes approches d'adaptation du facteur d'inertie pour F4	56
<b>Figure 2.15</b> : Résultats des différentes approches d'adaptation du facteur d'inertie pour F5	58
<b>Figure 3.1</b> : Structure du graphe d'information.	63
<b>Figure 3.2(a et b)</b> : Suppression d'une tribu (monoparticule, pluriparticule)	65
<b>Figure 3.3</b> : Choix d'une nouvelle position	69
<b>Figure 4.1</b> : Les composants principaux d'un processus de recalage des images	77
<b>Figure 4.2 (a)</b> : approche directe d'interpolation	79
<b>Figure 4.2(b)</b> : approche inverse d'interpolation	79
<b>Figure 4.3</b> : Exemple d'interpolation d'un facteur 2 dans les deux dimensions	80
<b>Figure 4.4</b> : Grille des positions de l'image de référence alignées aux positions non régulières de l'image recalé.	80

<b>Figure 4.5 :</b> Un plot de contour illustre l'effet de l'orientation initiale sur les Particules de l'essaim	89
<b>Figure 4.6:</b> structure pyramidale d'une image	90
<b>Figure 4.7 :</b> Schéma d'un histogramme conjoint de deux images A et B	93
<b>Figure 4.8 :</b> Images cérébrales IRM (coupe axiale)	94
<b>Figure 4.9 :</b> L'information mutuelle en fonction d'erreur de translation et l'erreur de rotation	95
<b>Figure 4.10 :</b> Les différents coupe d'un volume cérébrale	96
<b>Figure 4.11 :</b> Nombre d'évaluations pour chaque paire d'image	99

## Liste des tableaux

---

<b>Tableau 1.1:</b> paramètre d'OEP minimale	18
<b>Tableau 2.1:</b> Différentes méthodes d'optimisation fondées sur l'approche agrégative	29
<b>Tableau 2.2:</b> Comparaison des différentes techniques basées sur le nombre des solutions non dominées	51
<b>Tableau 2.3:</b> Comparaison des différentes techniques basée sur la largeur du front de Pareto	51
<b>Tableau 3.1:</b> stratégie de déplacement	70
<b>Table 4.1:</b> taux de succès des paires d'image 1	99
<b>Table 4.2:</b> taux de succès des paires d'image 2	99
<b>Table 4.3:</b> taux de succès des paires d'image 3	100

# Liste des algorithmes

---

<b>Algorithme 1.1 :</b> Algorithme OEP standard	8
<b>Algorithme 2.1 :</b> Algorithme général d'OEP multi-objectifs	34
<b>algorithme 2.2:</b> algorithme de OEPMO basé sur un archive externe	46
<b>Algorithme 2.3:</b> Algorithme OEPMO agrégatif à base de trois méthodes d'adaptation du facteur d'inertie	58
<b>Algorithme 3.1 :</b> Adaptations structurelles	71
<b>Algorithme 3.2 :</b> Algorithme de TRIBES	72

## Introduction générale

Les ingénieurs et les décideurs sont confrontés quotidiennement à des problèmes de complexité grandissante, relatifs à des secteurs techniques très divers, comme dans la conception de systèmes mécaniques, le traitement des images, l'électronique, les télécommunications, les transports urbains, etc. Généralement, les problèmes à résoudre peuvent souvent s'exprimer sous forme de problèmes d'optimisation. Ces problèmes sont le plus souvent caractérisés en plus de leur complexité, d'exigences qui doivent tenir compte de plusieurs contraintes spécifiques au problème à traiter.

Pour décrire un problème, on définit une fonction objectif, ou fonction de coût (voire plusieurs), que l'on cherche à minimiser ou à maximiser par rapport à tous les paramètres concernés. La définition d'un problème d'optimisation est souvent complétée par la donnée de contraintes : tous les paramètres des solutions retenues doivent respecter ces contraintes, faute de quoi ces solutions ne sont pas réalisables.

La plupart des problèmes d'optimisation sont des problèmes NP-difficiles qui ne possèdent pas une méthode exacte (par exemple les méthodes dite d'optimisation globale) efficace permettant de repérer un optimum global à coup sûr et en un nombre fini de calculs. Les méthodes incomplètes, qui sacrifient la complétude pour gagner l'efficacité, deviennent une alternative intéressante.

Dans ce contexte, un nouveau thème de recherche dans le domaine des sciences de l'information a été récemment suggéré. Cette voie regroupe des approches possédant des caractéristiques ou des comportements "intelligents" inspirées de la métaphore biologique (programmation évolutive, stratégie évolutive, programmation génétique, algorithmes génétiques), ou du comportement collectif des insectes (colonies de fourmis, oiseaux migrateurs), etc..., ces nouvelles méthodes dite métaheuristiques sont très utilisés dans le domaine de l'optimisation difficile.

Une métaheuristique est un algorithme d'optimisation visant à résoudre des problèmes d'optimisation difficile (souvent issus des domaines de la recherche opérationnelle, de l'ingénierie ou de l'intelligence artificielle) pour lesquels on ne connaît pas de méthode classique plus efficace.

Les métaheuristiques sont généralement des algorithmes stochastiques itératifs, qui progressent vers un optimum global, c'est-à-dire l'extremum global d'une fonction, par échantillonnage d'une fonction objectif. Elles se comportent comme des algorithmes de recherche, tentant d'apprendre les caractéristiques d'un problème afin d'en trouver une approximation de la meilleure solution (d'une manière proche des algorithmes d'approximation).

Il existe un grand nombre de métaheuristiques différentes, allant de la simple recherche locale à des algorithmes complexes de recherche globale. Ces méthodes utilisent cependant un haut niveau d'abstraction, leur permettant d'être adaptées à une large gamme de problèmes différents.

Dans ce travail, nous nous intéressons à l'étude des capacités de la métaheuristique d'optimisation par essaim particulaire (OEP) pour résoudre des problèmes d'optimisation. Depuis son apparition, l'OEP requiert de plus en plus l'attention de la communauté scientifique vu le succès qu'elle a réalisé. Elle a été appliquée à plusieurs problèmes dans les différents domaines de l'ingénierie.

L'optimisation par essaim particulaire (OEP ou PSO en anglais) 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 le biologiste 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 fonde 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 puisse 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.



La richesse de l'OEP est qu'elles se prêtent à toutes sortes d'extensions. Citons, en particulier :

- L'optimisation multiobjective, où il s'agit d'optimiser simultanément plusieurs objectifs contradictoires ;
- L'optimisation multimodale, où l'on s'efforce de repérer tout un jeu d'optima globaux ou locaux ;
- L'optimisation dynamique, qui fait face à des variations temporelles de la fonction objective.

Dans ce contexte, le travail de ce mémoire présente en premier lieu un état d'art non exhaustif des travaux réalisés, nous nous décrivons les différentes caractéristiques de cette méthode, ainsi que les différentes modifications apportées sur la version originale pour assurer une meilleure convergence. Les différentes implémentations effectuées nécessitent une phase d'adaptation de la méthode adoptée ainsi qu'un bon réglage des paramètres

Le deuxième chapitre est consacré à la présentation des différents formalismes utilisés pour adapté l'OEP à une catégorie particulière des problèmes qui nécessite un traitement spécial, cette catégorie s'appelle les problèmes multi-objectifs. Un nouveau modèle, d'optimisation par essaim particulière pour les problèmes multi-objectifs, basé sur OEP agrégatif dynamique, et trois approches d'adaptation du facteur d'inertie: aléatoire, linéaire, non linéaire est aussi proposé. La nouveauté principale de ce modèle consiste en utilisation d'un mécanisme d'adaptation du facteur d'inertie qui permet de fournir une meilleure diversité des solutions dans l'espace de recherche.

Le troisième chapitre consiste à présenter les différents travaux réalisés dans le cadre de la méthode OEP adaptatif, le but est d'établir des règles intelligentes qui permettent de déterminer automatiquement les paramètres de l'algorithme. Nous avons présenté en détaille un algorithme OEP totalement adaptatif, on appel cet algorithme **TRIBES**

Le dernier chapitre est consacré à l'application de la méthode OEP au recalage des images IRM multimodales, nous avons réalisé une étude comparative sur trois modèles d'OEP à facteur d'inertie adaptatif : le modèle linéaire, le modèle non linéaire et modèle aléatoire le but est de déterminer le meilleur modèle. Les applications de ces approches étant réalisées dans le domaine de l'image et notamment du recalage d'images multimodales issues de données médicales.

# **CHAPITRE 1**

## **INTRODUCTION A L'OPTIMISATION PAR ESSAIM DE PARTICULES**

## 1.1 Introduction

L'Optimisation par Essaim Particulaire (OEP) à laquelle est dédié ce chapitre du présent travail consiste en une méthode de résolution de problèmes dits d'optimisation.

Les problèmes d'optimisation peuvent être décrits en termes mathématiques simples de la manière suivante : étant donné une certaine fonction (dite fonction objectif) définie sur un certain domaine de définition (appelé espace de recherche), trouver le (ou les) point(s) de ce domaine de définition qui minimise (maximise) la valeur de la fonction objectif.

Lorsque la fonction objectif présente certaines propriétés (par exemple, lorsqu'elle est analytique, continue,...) il existe des méthodes de recherche, dites exactes, permettant de trouver le point optimum (un bon exemple de méthode exacte est le bien connu c'est l'algorithme du simplexe).

Cependant, dans bon nombre de cas pratiques, ces méthodes se révèlent inefficaces et ce généralement parce qu'elles demeurent trop lentes vis-à-vis d'espaces de recherche beaucoup trop vastes.

Pour pallier à ce problème, de nouvelles méthodes, dites non-exactes, est connues sous le nom de méta-heuristiques, ont commencé à apparaître. Par non-exactes, il faut comprendre que ces méthodes reposent sur des schémas d'exploration de l'espace de recherche faisant intervenir le hasard et n'assurant pas de retrouver avec certitude la solution cherchée (mais présentant cependant d'excellente performance dans bon nombre de cas pratiques, ce qui motive leurs emplois). Ces schémas de recherche s'inspirent généralement de processus naturels présentant des propriétés d'évolution vers quelque chose de meilleur.

L'OEP est l'une de ces méta-heuristiques. Ce premier chapitre du présent travail sera consacré à en proposer un état de l'art.

## 1.2 OEP standard

### 1.2.1 Description informelle

Au départ J. Kennedy et R. Eberhart [Kennedy et Eberhart, 1995] cherchaient à simuler la capacité des oiseaux à voler de façon synchrone et leur aptitude à changer brusquement de direction tout en restant en une formation optimale. Le modèle qu'ils ont proposé a ensuite été étendu en un algorithme simple et efficace d'optimisation.

Comme nous allons le voir, le fonctionnement de l'OEP fait qu'elle peut être rangée dans les méthodes itératives (on approche peu à peu de la solution) et stochastiques (on fait appel au hasard).

Sous ce terme, on retrouve le comportement suivant: améliorer sa situation en se déplaçant partiellement au hasard et partiellement selon des règles prédéfinies. Chaque particule de l'essaim représente une solution potentielle dans l'espace de recherche. Dans le système OEP chaque particule explore l'espace de recherche multidimensionnel et ajuste sa position selon trois composantes:

- une composante physique: la particule tend à suivre sa courante direction (tendance aventureuse).
- une composant cognitif: cette tendance dite conservatrice représente l'expérience de la particule(le meilleur point par lequel elle est déjà passée).
- une composante sociale: cette composante pousse la particule à aller vers la meilleure des meilleures positions rencontrées par ses informatrices de par le passé.

La particule combine sa vitesse courante, sa meilleure solution et la meilleure solution retrouvée par son voisinage pour ajuster sa position vers une solution optimale (**Figure 1.1**).

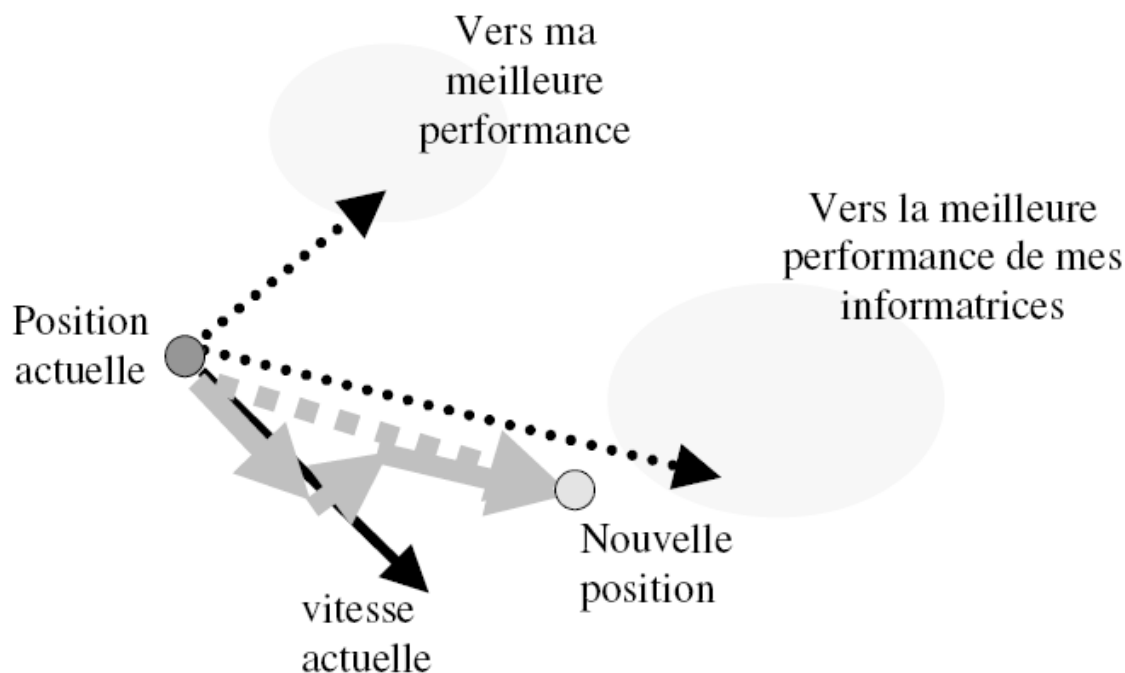
### 1.2.2 Formalisation et programmation

La première étape consiste à initialisé N particule dans l'espace de recherche de dimension D, l'algorithme va évaluer la fonction en ces points de l'espace de recherche, La performance de chaque particule est mesuré on utilisant une fonction de fitness prédéfini qui encapsule les caractéristiques du problème d'optimisation.

Chaque particule  $i$  maintienne une courante position  $X_i$ , une courante vélocité  $V_i$  et sa meilleur position  $XBest_i$  ou :

- $X_i$  représente un point dans un espace de recherche continue.
- Si  $F$  est la fonction de fitness, à chaque pas de temps, chaque particule de l'essaim modifie sa meilleure position et conserve cette position dans sa mémoire de la façon suivante :

$$X_{best_i}(t+1) = \begin{cases} X_{best_i}(t) & \text{si } F(X_i(t+1)) \geq F(X_{best_i}(t)) \\ X_i(t+1) & \text{si } F(X_i(t+1)) < F(X_{best_i}(t)) \end{cases} \quad (1.1)$$



**Figure 1.1 : Schéma de principe du déplacement d'une particule**

D'une autre part, on donne à chacune des particules la capacité de consulter la mémoire de certaines autres particules appelées ses **informatrices** pour déterminer la meilleure position trouvée par celles-ci.

Si la position de la meilleur informatrice d'une particule  $i$  est  $XGbest_i$  et  $Xbest_i$  sa meilleur position alors :

Pour Chaque itération  $t$ , chaque particule  $i$  modifie le  $d^{eme}$  composant du vecteur de vélocité  $V_{i,d}$  selon les équations (1.2) et ajuste sa position selon l'équation (1.3) :

$$V_{i,d}(t+1) = w.V_{i,d}(t) + c_1 r_{1,d} (Xbest_{i,d}(t) - X_{i,d}(t)) + c_2 r_{2,d} (XGbest_{i,d}(t) - X_{i,d}(t)) \quad (1.2)$$

$$\vec{X}_i(t+1) = \vec{X}_i(t) + \vec{V}_i(t+1) \quad (1.3)$$

Où  $w$  représente le coefficient d'inertie,  $c_1$  et  $c_2$  sont des constantes d'accélération,  $r_1$  et  $r_2$  sont des variable aléatoires uniformément distribué sur l'intervalle  $[0,1]$ . Selon l'équation (1.2) le vecteur de vitesse dirige le processus de recherche et reflète la "sociabilité" des particules. Dans cette version standard d'OEP, si l'on considère  $N$  particules, chaque particule compare sa nouvelle position à sa meilleure position obtenue, et la position de la meilleure particule cela donne l'**algorithme 1.1**

### Algorithme 1.1 – Algorithme OEP standard

\*\*\*\*\*

[Les variables et paramètres de l'algorithme]

N nombre de particules

–  $X_i$  position de la particule  $P_i$

–  $V_i$  vitesse de la particule  $P_i$

–  $P_{best_i}$  meilleure fitness obtenue pour la particule  $P_i$

–  $G_{best}$  fitness obtenue pour la meilleure particule.

–  $X_{best_i}$  position de la particule  $P_i$  pour la meilleure fitness

–  $X_{g_{best}}$  position de la meilleure informatrice d'une particule  $P_i$

$C_1, C_2$  valeurs aléatoires positives

[-----]

[Initialisations]

Initialiser aléatoirement la population

...

[Traitement]

**Répéter**

**Pour**  $i$  de 1 à  $N$  faire

**Si**  $(F(x_i) > P_{best_i})$  Alors

$P_{best_i} := F(x_i)$

$X_{best_i} := x_i$

**Fin Si**

**Si**  $(F(x_i(t)) > G_{best})$  Alors

$G_{best} := F(x_i)$

$X_{g_{best}} := x_i$

**Fin Si**

**Fin Pour**

**Pour**  $i$  de 1 à  $N$  faire

Modification du vecteur  $V_i$  selon l'équation 1.2

Modification  $X_i$  selon l'équation 1.3

**Fin Pour**

**jusqu'à ce que** (le processus converge)

\*\*\*\*\*

Plus une particule est éloignée de la meilleure solution globale et de sa meilleure solution, plus sera importante la variation de sa vitesse afin de faire bouger la particule vers les meilleures solutions. L'algorithme s'exécute tant qu'un critère de convergence n'a pas été atteint. Cela peut être :

- Un nombre fixe d'itérations ;
- En fonction de la fitness ;
- Lorsque la variation de vitesse est proche de 0.

### 1.3 Graphe d'influence et type de voisinage

Le graphe d'influence constitue la structure du réseau social . Les particules à l'intérieur d'un graphe communiquent entre-elles. Différents types de voisinages ont été étudiés pour un graphe d'influence [Kennedy, 1999] ,ils sont considérés en fonction des identificateurs des particules et des informations topologiques comme les distances euclidiennes dans l'espace de recherche. il y a trois types de voisinages à savoir un voisinage statique et un voisinage dynamique aléatoire et un voisinage basé sur la distance euclidienne.

#### 1.3.1 Voisinage statique

Si l'on réfère au premier article qui présente la méthode OEP( optimisation par essaim Particulaire), le voisinage du graphe d'information est statique, il est défini une fois pour toutes avant que le processus d'itération commence.pour un voisinage statique le graphe d'information soit complètement connecté ,la meilleur particule est donc choisie à partir de la population entiere ou bien partiellement connecté ,dans cette version dite locale ,la particule n'est donc plus informée par toutes les particules de l'essaim, mais par elle-même et un voisinage de taille  $k$  défini par l'utilisateur .dans cette version deux topologies sont présentés :

- **Topologie réguliere** :  $N$  particules de l'essaim sont disposées virtuellement autour d' un cercle, ces particules sont numérotées séquentiellement à partir de 1 en parcourant ce cercle. Chaque particule a un ensemble d'informateurs de taille fixe, historiquement appelés son voisinage. Le voisinage de taille  $K$  d'une particule est obtenu à partir de ce cercle en recrutant alternativement des partiules du côté droit et du côté gauche de sa position, jusqu'à ce qu'un total de  $K-1$  voisins soit obtenu, la particule elle-même est inclus dans son voisinage.
- **Topologie aléatoire** : Dans cette topologie les  $K$  voisins d'une prticule sont choisies aléatoirement avant que le processus d'itération commence.

La figure 1.2 illustre la différence entre topologie réguliere et topologie aléatoire.

### 1.3.2 Voisinage dynamique aléatoire

Le principe du voisinage aléatoire dynamique peut être récapitulé brièvement par:

- un nombre  $k$  inférieur ou égal à la taille de l'essaim est défini au début.
- Avant chaque itération, chaque particule choisie aléatoirement  $K$  particules et établit un lien d'information vers elles.

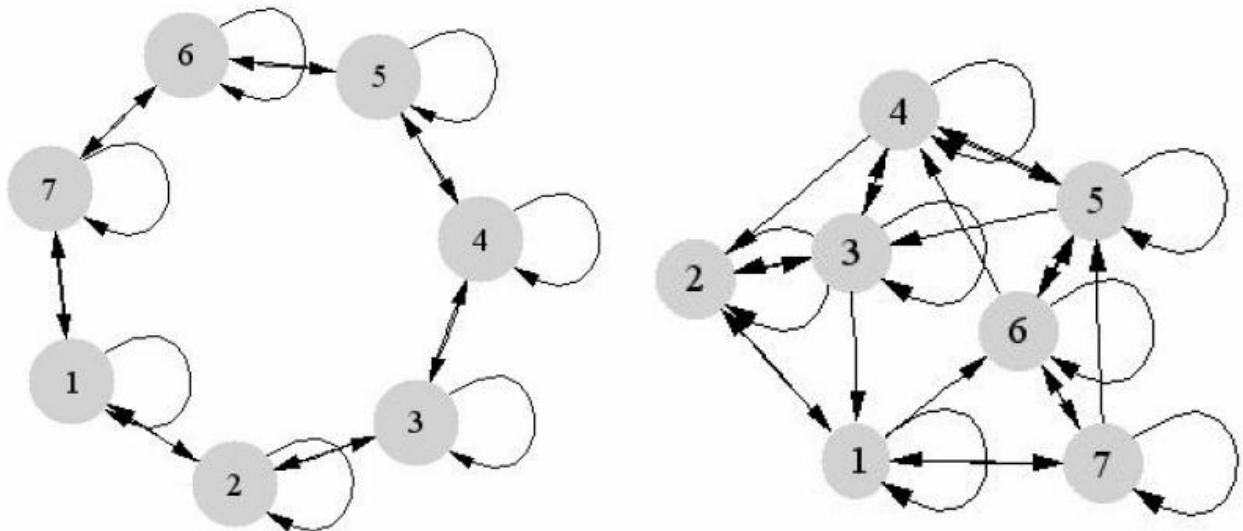
### 1.3.3 Voisinage basé sur une distance euclidienne

Suganthan [Suganthan, 1999] utilise un voisinage spatial entre les particules. Une particule  $P_b$  est voisine d'une particule  $P_a$  si :

$$\frac{\|\vec{x}_a - \vec{x}_b\|}{d_{\max}} < \varepsilon \quad (1.4)$$

avec  $d_{\max}$  la plus grande distance entre deux particules et

$$\varepsilon = \frac{3T + 0.6t_{\max}}{t_{\max}} \quad (1.5)$$



**Figure 1.2 : graphe d'influence statique composé de 7 particules et un voisinage de taille  $K=3$  (a) topologie régulière (b) topologie aléatoire.**

Où  $t$  représente l'itération courante et  $t_{\max}$  le nombre maximal d'itérations. On peut remarquer que la taille du voisinage croît avec le temps.

Cette alternative exige un coût calculatoire très élevés et ne donne pas une meilleure effectiveness vis-à-vis aux autres versions.



Les résultats expérimentaux sur le benchmark des fonctions de test montre qu'en général la topologie régulière est la plus effective que la topologie aléatoire. ainsi bien que le voisinage local converge moins rapidement que la Version du voisinage globale, cette version locale de l'OEP donne de meilleurs résultats, car elle est moins sujette à l'attraction par des minima locaux [Kennedy, 1999], on peut noter aussi que les résultats obtenus avec ces diverses topologies suggèrent des règles de choix pour le nombre de liens par particule.

### **1.3.4 Connexité temporelle de la diffusion de l'information et OEP**

#### **multi- essaim**

Dans le choix d'une topologie, il est important d'assurer ce que l'on appelle la connexité temporelle de la diffusion de l'information. Cela signifie que la topologie mise en place doit être telle qu'à un moment donné, toute particule aura une probabilité tendant vers 1, pour un nombre d'itérations tendant vers l'infini, d'avoir accès à l'information d'une position qui serait meilleure que celle qu'elle conserve en mémoire, et qui aurait été découverte par une autre particule.

À l'inverse, prohiber la connexité temporelle de la diffusion de l'information entre 2 groupes fixes de particules amène la division de l'essaim en 2 sous-essaims distincts de particules, chacun d'entre eux se comportant alors comme un essaim autonome et individuel. Cette alternative, bien évidemment généralisable à une division de l'essaim en  $G$  sous-essaims, peut toutefois s'avérer intéressante dans certains cas où l'on désirerait diviser l'essaim en sous-essaims et ce, afin d'assigner chaque sous-essaim à l'exploration d'une zone particulière de l'espace de recherche (comme l'implémente, par exemple, certains algorithmes OEP développés pour des problèmes d'optimisation multi-objectif).

Dans ce cas, il convient de mettre en place des mécanismes de répulsion entre essaims de manière à éviter que le travail ne soit fait deux fois. On entre alors dans le domaine de ce qui s'appelle l'OEP multi-essaims [Chow et al., 2004].

### **1.4 Stratégie de déplacement et distribution de la proximité**

Comme nous avons déjà vu, la première formulation du OEP montre que le processus d'optimisation est un processus itératif stochastique. la nouvelle position d'une particule est déterminée en fonction de sa propre valeur et celle de son voisinage selon une distribution rectangulaire. l'étude expérimentale sur un benchmark des fonctions de test montre que l'utilisation judicieuse d'autres alternatives tel que une distribution sphérique ou une

distribution gaussienne ,donnent des résultats mieux que celles donnés par la distribution rectangulaire selon la fonction traité.

La présente sous-section dédiée aux distributions des possibles tentera de poser un regard plus approfondi sur la distribution des possibles rectangulaire de l'OEP minimale.

Il s'en suivra la mise en évidence des principaux défauts de cette distribution ainsi que la présentation de certaines variantes.

### 1.4.1 La distribution rectangulaire de l'OEP minimale

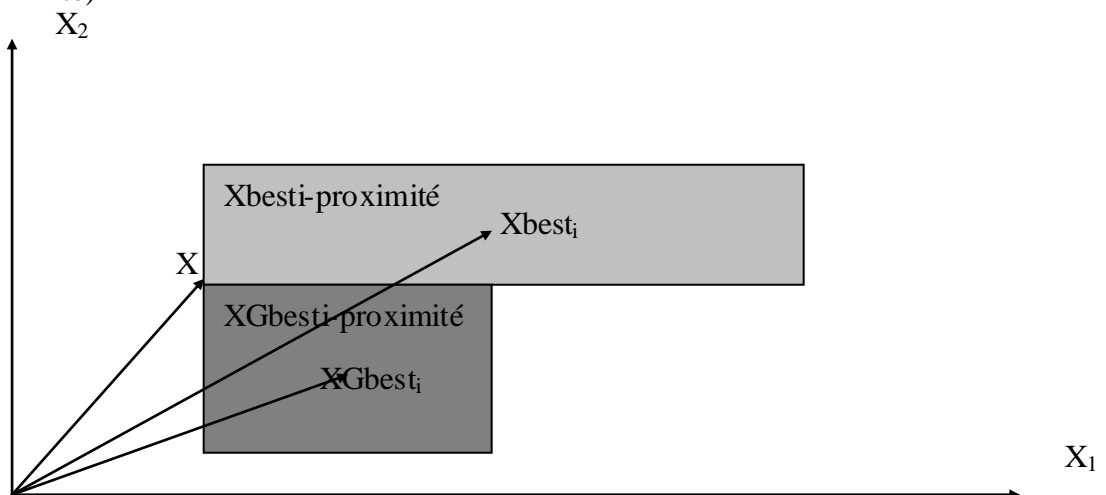
Rappelons-nous les équations de mouvements de l'OEP standard, pour maintenir simplement que chaque élément aléatoire correspond à une distribution uniforme des positions possibles dans un D-rectangle:

$$V_{i,d}(t+1) = w.V_{i,d}(t) + c_1 r_{1,d} (Xbest_{i,d}(t) - X_{i,d}(t)) + c_2 r_{2,d} (XGbest_{i,d}(t) - X_{i,d}(t))$$

$$\vec{X}_i(t+1) = \vec{X}_i(t) + \vec{V}_i(t+1)$$

Nous puvons constaté que la distribution de la possibilité induite par ces équations resulte de la combinaison de trois termes (**figure 1.3**):

- un terme constant relatif à la vitesse courante représente une translation.
- un terme associé à la meilleur position d'une particule Xbest<sub>i</sub> ,induit une distribution uniforme sur un D-rectangle. On nomme cette zone par Xbest<sub>i</sub>-proximité
- un terme associé à la meilleur position retrouvé par le voisinage d'une particule XGbest<sub>i</sub>,induit aussi une distribution uniforme sur un D-rectangle (cette zone est nommée XGbest<sub>i</sub>-proximité)



**Figure 1.3: les deux distributions élémentaires de la proximité Xbest<sub>i</sub>-proximité Et XGbest<sub>i</sub>-proximité dans un espace de deux dimensions.**

Nous pouvons noter que la combinaison par l'addition des deux D-rectangles uniformes ( $X_{best_i}$ -proximité et  $XG_{best_i}$ -proximité) n'est plus uniforme, mais que, néanmoins, son appui demeure lui-même un D-rectangle (**figure 1.4**).

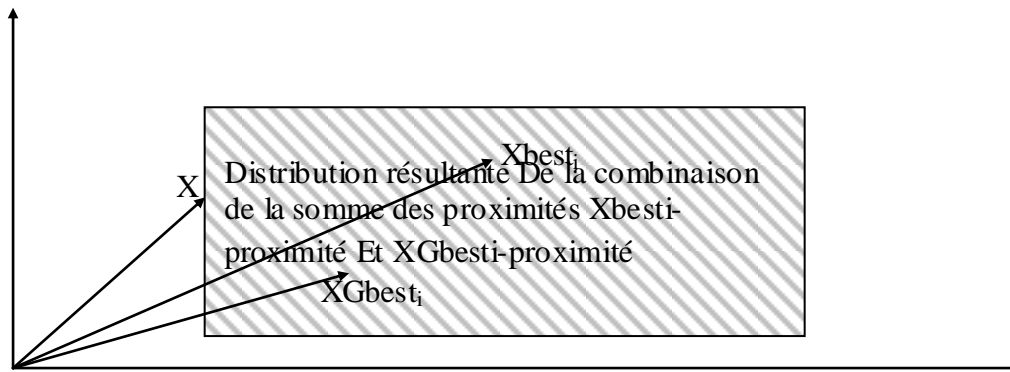


Figure 1.4 Ensemble de positions atteignables résultantes de la somme des proximités  $X_{best_i}$ -proximité et  $XG_{best_i}$ -proximité, pour un espace de recherche a deux dimensions

L'expérience prouve qu'il n'est pas très satisfaisant, dû au fait que cette distribution présentent de defaults majeurs [Clerc, 2005] [Clerc, 2006]:

- **Biais de la distribution** : La proximité de  $X_{best_i}$  (respectivement  $XG_{best_i}$ ) est un simple segment, et la distribution des possibilités pour le prochain déplacement est un D-parallélépipède situé entre ces deux deniers points (ces deux points sont sur sa surface), ce qui limite l'exploration de l'espace de recherche, en particulier lorsqu'un ensemble de points localisés près de  $X_{best_i}$  (respectivement  $XG_{best_i}$ ) n'a aucune chance d'être choisis pour le prochain déplacement (**figure 1.5**).

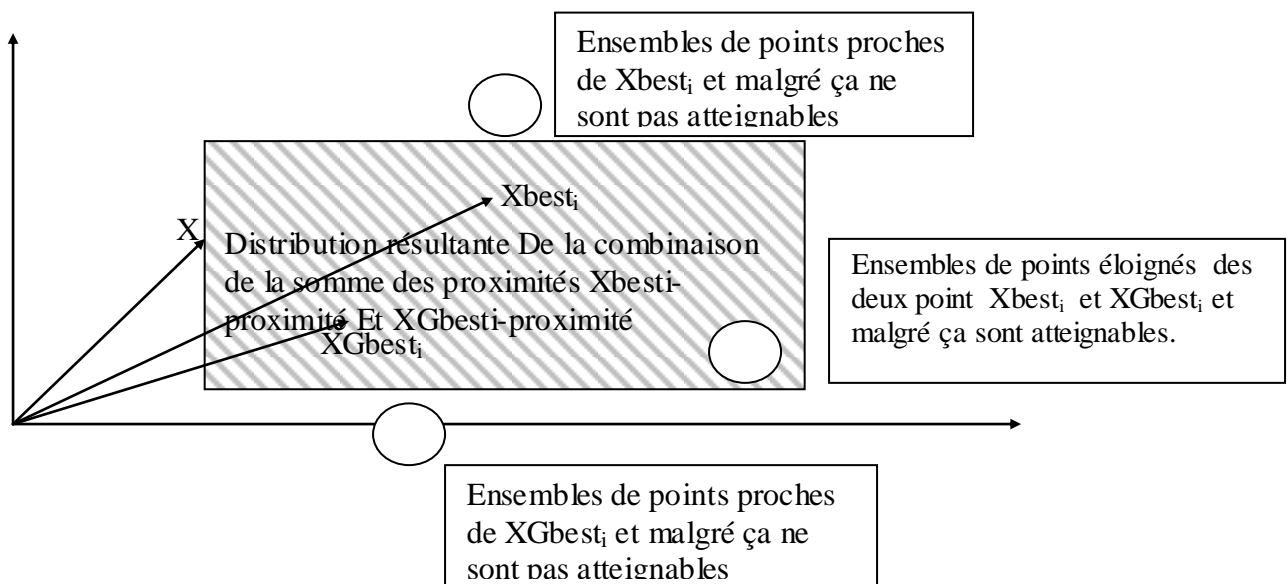
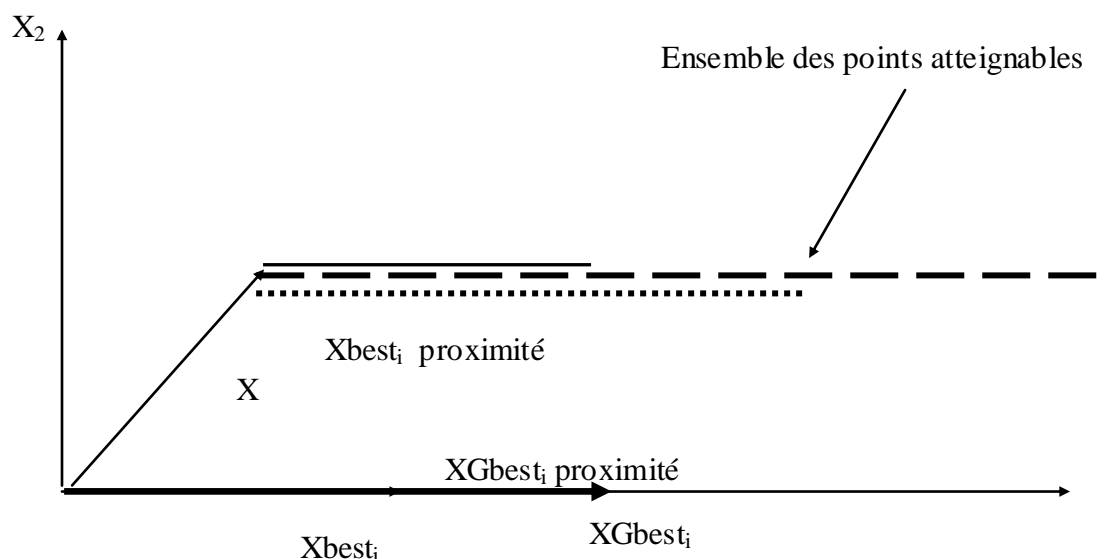


Figure 1.5: Biais de la distribution: certains points ne sont pas atteignable malgré qu'ils sont proche des deux points  $X_{best_i}$  et  $XG_{best_i}$ , d'une autre il y a des points qui sont situés dans coins éloignés des ces deux points et pourtant sont atteignables.

- **Dépendance de la distribution du système de coordonnées** : il convient noter que cette distribution dépend du système de coordonnées (**figure 1.6**). Si par la mauvaise chance le point  $p$  est sur un axe du système de coordonnées, le D-rectangle de sa proximité perd une dimension. Pour un problème avec deux des dimensions, par exemple, il est réduit à un segment.



**Figure 1.6** Dépendance vis-à-vis du système de coordonnées : lorsque  $p$  et  $g$  sont sur le même axe de coordonnées, l'ensemble des positions atteignables perd une dimension

### 1.4.2 Distribution à secteurs ellipsoïdaux positifs

Cette méthode est plus proche au OEP standard, la distribution uniforme est remplacée par une distribution lequel son support est un secteur sphérique positif :

$$V_{id} = w * V_{id} + c1 * alea\_sphere(0, \rho)_d / (X_{best_i,d} - X_{id}) + c2 * alea\_sphere(0, \rho)_d / (X_{Gbest_i} - X_{id}) \quad (1.6)$$

La fonction  $alea\_sphere(0, \rho)$  retourne un point choisi aléatoirement selon une distribution uniforme dans un D-sphere lequel son centre est l'origine des coordonnées et son rayon est  $\rho$ .  $\rho$  est calculé de tel sorte que le volume de D-sphère soit égale à 1.

Notons que cette distribution s'affranchit du problème de biais de distribution mais conserve l'inconvénient de la dépendance vis à vis au système de coordonnées.

### 1.4.3 Gaussiennes indépendantes

Celle-ci est similaire à la méthode précédente. Pour chaque dimension on utilise une distribution normale au lieu de la distribution uniforme, l'équation de mouvement est remplacé par :

$$V_{id} = w * V_{id} + alea\_normal_1(\mu, \sigma) * (X_{best_{i,d}} - X_{id}) + alea\_normal_2(\mu, \sigma) * (XG_{best_{i,d}} - X_{id}) \quad (1.7)$$

Cette méthode introduit deux nouveaux paramètres, le moyen  $\mu$  et l'écart-type  $\sigma$  de la distribution. Si nous voulons rester proche à la version standard, il prévient évident de prendre  $\mu = c_{max}$ ,  $\sigma = c_{max}/4$ .

### 1.4.4 Locale par gaussiennes indépendantes

L'idée est de chercher localement "autour" de la meilleur position connue d' une particule, c'est-à-dire  $X_{best_i}$ . Ainsi, pour chaque dimension  $d$ , une coordonnée plus ou moins proche de  $X_{best_{i,d}}$  est choisie aléatoirement selon une loi gaussienne, si la distribution est centrée à  $X_{best_i}$ , l'équation de déplacement est défini par :

$$x_d = X_{best_{i,d}} - alea\_normal(0, \|X_{best_{i,d}} - x_d\|) \quad (1.8)$$

On peut remplacer cette distribution par une distribution uniforme dans l'intervalle  $[-\|X_{best_{i,d}} - x_d\|, \|X_{best_{i,d}} - x_d\|]$ .

### 1.4.5 La méthode des pivots

La méthode des pivots est une autre alternative à la gestion des déplacements classique suggérée par l'OEP minimale. Elle peut se résumer par :

- On considère deux hypersphères centrés respectivement sur  $X_{best_i}$  et  $XG_{best_i}$ , et de rayon  $|X_{best_i} - XG_{best_i}|$ .
- Dans chacune de ces hypersphères, un point est choisi au hasard.
- On affecte à ces deux points des poids relatifs fonctions de la différence de leur qualité respective.
- La nouvelle position est le barycentre de ces deux points.

Remarquons que, quant à elle, la méthode des pivots présente l'avantage de s'affranchir à la fois du problème de biais de distribution, et de celui de la dépendance vis à vis du système de coordonnées.

### 1.4.6 La méthode des pivots brute

La méthode des pivots brutes s'obtient en rajoutant une étape à la méthode des pivots, une fois le barycentre trouvé, on calcule la nouvelle position en rajoutant à ce barycentre un bruit gaussien de moyenne nulle et d'écart type proportionnel à la différence de qualité entre  $X_{best_i}$  et  $XG_{best_i}$ .

La méthode des pivots brute s'affranchit également des problèmes de biais de distribution et de dépendance vis à vis du système de coordonnées.

## 1.5 Sélection des paramètres de l'algorithme

Six paramètres rentrent en ligne de compte :

- La dimension du problème.
- Le nombre de particules.
- Les valeurs des coefficients :  $\rho_1 = c_1 r_1$ ,  $\rho_2 = c_2 r_2$
- La taille du voisinage.
- La vitesse maximale.
- L'inertie.

Nous allons nous intéresser plus particulièrement aux deux derniers, nous donnons ainsi quelques directives pour le choix des autres paramètres.

### 1.5.1 La vitesse maximale

Pour éviter que les particules se déplacent trop rapidement d'une région à une autre dans l'espace de recherche et limiter la taille du pas d'une particule on fixe une vitesse maximale  $V_{max}$  [Eberhart et al, 1996], de cette façon on peut restreindre un comportement divergent. Ainsi si  $v_{ij}(t)$  est la vitesse de la particule  $P_i$  au temps  $t$  dans la dimension  $j$ ,

$$v_{ij}(t) = V_{max} \text{ si } v_{ij}(t) > V_{max} \text{ et } v_{ij}(t) = -V_{max} \text{ si } v_{ij}(t) < -V_{max}. \quad (1.9)$$

$V_{max}$  est généralement dépendant de l'échelle du problème.  $V_{max}$  n'est pas obligatoire [Clerc and Kennedy, 2002] si on utilise un coefficient de constriction :

$$V_i(t+1) = k(V_i(t) + \rho_1(X_{best_i}(t) - X_i(t)) + \rho_2(XG_{best}(t) - X_i(t))) \quad (1.10)$$

Avec :  $k = 1 - \frac{1}{\rho} + \frac{\sqrt{|\rho^2 - 4\rho|}}{2}$  et  $\rho = \rho_1 + \rho_2 > 4$  (1.11)

### 1.5.2 L'inertie

Le facteur d'inertie est induit par Shi et Eberhart [Shi et Eberhart, 1998]. Pour contrôler l'influence de la vitesse obtenue au pas précédent pour éviter l'élaboration de  $V_{\max}$ .

Un grand facteur d'inertie provoque une grande exploration de l'espace de recherche alors qu'un petit facteur d'inertie concentre la recherche sur un petit espace ce qui provoque une recherche locale. La convergence n'est pas assurée quelques soient les valeurs de  $w$  et également de  $c1$  et  $c2$  [Vandenbergh, 2002], les études expérimentales montrent que:

$$\frac{1}{2}(c1 + c2) < w < 1 \quad (1.12)$$

### 1.5.3 Robustesse et carte de performance

Maurice Clerc démontre qu'une bonne convergence peut être assurée à partir la définition d'une relation de dépendance entre les deux paramètres  $w$  et la valeur maximale  $\rho_{\max}$  de  $\rho1$ ,  $\rho2$ .

La relation entre ces deux paramètres a été définie par l'intermédiaire d'un autre paramètre  $\varphi$  [Clerc et Kennedy, 2002]:

$$w = \frac{1}{\varphi - 1 + \sqrt{\varphi^2 - 2\varphi}} \quad (1.13)$$

$$\rho_{\max} = \varphi w$$

Pour établir la carte de performance, nous procédons à tester un grand nombre de paires ( $w$ ,  $\varphi$ ) et on examine le comportement de l'algorithme pour chaque paire.

Pour limiter l'espace des valeurs possibles voici quelques remarques :

- la taille de l'Essaim sera variée entre 5 et 40 particules.
- $\varphi$  doit être supérieur à 2.
- Pour  $\rho_{\max}$  l'expérimentation montre qu'il doit être supérieur à 1.

### 1.5.4 Stabilité et convergence de l'algorithme

Beaucoup d'efforts ont été investis pour obtenir une meilleure compréhension des propriétés de convergence des systèmes OEP. Ces efforts sont concentrés principalement sur une meilleure compréhension des paramètres de contrôle fondamentaux, à savoir les coefficients d'accélération, le facteur d'inertie, la vitesse maximale, et la taille de l'essaim [Kennedy,1997] [Shi et Eberhart, 1998] [Shi et Eberhart, 1998a] [Shi et Eberhart, 1999]

[Carlisle et Dozier, 2001] [Beielstein et al ,2002] [Naka et al ,2003]. Ces études empiriques, montre que le système OEP est sensible aux choix de paramètres de contrôle, en particulier le facteur d'inertie, les coefficients d'accélération et la vitesse maximale. L'initialisation incorrecte de ces paramètres peut conduire à un comportement divergent ou cyclique.

Les études empiriques d'OEP, cependant, donnent un aperçu du comportement des essais de particules, fournissant des directives pour l'initialisation des paramètres. Par exemple, Eberhart et Shi constaté empiriquement que l'inertie d'un poids de 0,7298 et un coefficient d'accélération de 1,49618 sont des bons paramètres, ce qui conduit à des trajectoires convergentes [Eberhart et Shi ,2000]. Bien que ces valeurs empiriques obtenues fonctionnent bien (en général), ils devraient être considérés avec prudence, car ces études empiriques sont basées uniquement sur un échantillon limité de problèmes. Il convient également de noter que les paramètres de contrôle dépendent au problème d'optimisation.

Le tableau suivant donne un aperçu général sur les différentes valeurs des paramètres avec quelques règles empiriques.

**Tableau1.1 paramètre d'OEP minimale**

paramètre	Nature	Règle empirique	Valeur préférée
W	Facteur d'inertie	Dans ]0,1[	0.7298
$\rho_1, \rho_2$	Coefficients d'accélération	Dans l'environ de 1.5	1,49618
N	Taille de l'ESSAIM	Entre 20 et 40	20
K	Nombre des particules informées par une particule	- Entre 3 et 5 - N pour les problèmes sans minimums locaux	3
Vmax	Vitesse maximale	Utilisé si seulement si $w > 1$	$(X_{max} - X_{min})/2$

## 1.6 Algorithme OEP à convergence garantie

Le modèle Gbest d'OEP conduit à une propriété non souhaité lorsque  $X_i = X_{best_i} = X_{Gbest}$  (pour chaque particule i), dans cette cas la vitesse modifiée par l'équation (1.2) dépend seulement par le terme  $w \cdot v_i(t)$ , laursque une particule approche à la meilleur position de son voisinage ça vélocité approche à zéro ce qui implique que la prticule arrete de se déplacée. ce comportement ne garante pas une convergence vers un optimum global et meme pas à un optimum local mais seulement à la meilleur position retrouvé jusqu' à



ici [Vandenbergh, 2002] [Van et Engelbrecht, 2006]. Van Den Bergh et Engelbrecht [Vandenbergh, 2002] [Van et Engelbrecht, 2002] ont introduit un nouveau algorithme à convergence garantie (GCPSO) pour éviter ce comportement.

Soit  $\tau$  l'index de la meilleure particule de l'ESSAIM, la vitesse et la position de cette particule sont modifiées par :

$$V_{\tau,d}(t+1) = -X_{\tau,d}(t) + XGbest_d + w \cdot v_{\tau,d}(t) + \rho(t)(1-2r_{2,d}) \quad (1.14)$$

$$X_{\tau,d}(t) = XGbest_d + w \cdot v_{\tau,d}(t) + \rho(t)(1-2r_{2,d}) \quad (1.15)$$

Le terme  $-X_{\tau,d}(t)$  remet la position de cette particule à sa meilleure position,  $w \cdot v_{\tau,d}(t)$  signifie une direction de recherche, le terme  $\rho(t)(1-2r_{2,d})$  ajout un comportement aléatoire à l'équation, le paramètre  $\rho(t)$  est dynamiquement adapté pour contrôler l'intervalle de la distribution autour de la meilleure position  $XGbest$  pour forcer le changement de la valeur de  $XGbest$  qui empêche le comportement illustré précédemment.

### 1.7 Cyber Swarm Algorithmes (CSA)

Cette version approfondit l'OEP en intégrant la notion de programmation à mémoire adaptative [Laguna et al, 2008].

- Learning from the Reference Set Members : Création d'un ensemble de référence mémorisant un ensemble des meilleures solutions trouvées. Une particule calcule alors sa nouvelle position depuis la position actuelle des autres particules mais aussi vis-à-vis de cet ensemble de référence.
- Dynamic Social Network : Le voisinage social d'une particule peut changer. On a alors une topologie en étoile et les communications entre les particules se font de manière stochastiques en fonction de tous les individus.
- Diversification Strategy : Les OEP accentuent trop le phénomène d'intensification au détriment de la diversification. Ajout de 2 nouvelles stratégies :
  - Minimum Diversity Strategy : Toutes les particules doivent être éloignées d'une distance minimale (seuil).
  - Exploratory Diversity Strategy : Visiter les régions sous-exploitées trouvées grâce à l'ensemble de référence.
- Conceptual and Algorithmic Description : L'apprentissage social d'une particule n'est pas restreint à l'interaction avec le meilleur de ses précédents voisins, mais l'apprentissage implique plutôt les individus d'un ensemble de référence dynamique contenant les meilleures solutions à travers l'historique de recherche vis à vis de la qualité et de la diversité.

## 1.8 Implémentation séquentielle vis-à-vis à l'implémentation parallèle

Deux types d'implémentation d'OEP sont possibles selon que pour chaque itération, qu'on fasse se déplacer les particules les unes à la suite des autres, ou qu'au contraire, on les fasse se déplacer simultanément. Dans le premier cas on parlera d'implémentation séquentielle, alors que dans le second, on parlera d'implémentation parallèle.

Au niveau de la recherche la différence sera la suivante ; dans le cas séquentiel, la particule qui est en train de se déplacer bénéficiera des éventuelles mises à jour d'informations que les particules ayant déjà effectué leur déplacement auront eues l'occasion de faire, alors qu'à l'inverse, dans le cas parallèle, il n'en sera rien. Des analyses effectuées dans [Clerc, 2005] semblent montrer que pour un algorithme OEP minimal typiquement paramètre, l'implémentation séquentielle présente des résultats sensiblement meilleurs que l'implémentation parallèle. De plus, de par sa nature itérative, l'algorithme OEP semble naturellement propice à une implémentation séquentielle.

L'implémentation parallèle peut cependant se révéler une alternative intéressante dans certains cas. Généralement, il s'agit de situation où l'évaluation de la fonction objective met un temps relativement conséquent (par exemple, dans certains cas où l'évaluation de la fonction objective s'effectue à partir d'un processus réel).

## 1.9 Domaine d'applications de la méthode OEP

Afin de proposer un aperçu des différentes applications traitées par l'OEP, voici les résultats d'une étude menée sur ce sujet et datant de mai 2007. (L'étude dans son intégralité est présentée dans [Poli, 2007]).

L'auteur de l'étude a parcouru la base de données IEEE-Xplore dans laquelle il recensa environ 1100 publications et articles scientifiques ayant attiré à l'OEP.

Parmi ces 1100 publications, environ 400 traitaient de l'OEP en tant qu'outil d'optimisation (en proposant des variantes et analyses), et environ 700 concernaient des applications de l'OEP.

L'auteur s'entreprit ensuite à classifier les 700 articles traitant d'applications de l'OEP en 26 domaines d'application. Cela donna lieu à la classification suivante dans laquelle, comme on peut le voir, bon nombre de grands domaines de l'ingénierie sont présents :

- \_ (9%) Métallurgie
- \_ (9%) Analyse image et vidéo
- \_ (7%) Les applications de contrôle (notamment en automatique)
- \_ (7%) Réseaux de distribution
- \_ (6%) Les antennes et leur design
- \_ (6%) Electronique et électromagnétique
- \_ (6%) L'énergie, génération et système
- \_ (6%) Ordonnancement
- (5%) Design
- \_ (4%) Biomédical
- \_ (4%) Réseaux de communication
- \_ (4%) Clustering et classification
- \_ (4%) Réseaux neuronaux
- \_ (4%) Simulation (notamment des réseaux neuronaux)
- \_ (4%) Signal processing
- \_ (3%) Modélisation
- \_ (3%) Prévision et prédiction
- \_ (3%) Robotique
- \_ (2.4%) Détection et diagnostic d'erreurs
- \_ (2%) L'optimisation combinatoire
- \_ (2%) Modélisation graphique et visuelle
- \_ (1.5%) Machines et moteurs
- \_ (1.4%) Sécurité et militaire
- \_ (1.3%) Divertissement (notamment en génération de musiques et de jeux)
- \_ (1%) Finance et économie

### **1.10 La recherche actuelle en matière OEP**

Depuis son invention en 1995, l'OEP ne cessa de stimuler l'intérêt de la communauté scientifique. Bon nombre de chercheurs se sont en effet penchés sur son étude, si bien que tant les variantes proposées que les analyses généralistes sont à l'heure d'aujourd'hui multiples (pour s'en convaincre, il suffit de considérer l'étude présentée dans [Poli, 2007]).

Cette section est dédiée à la présentation d'un projet de collaboration scientifique internationale en matière d'OEP, nommé projet XPS, et étant le plus grand projet (en terme de

ressources humaines et d'infrastructures) a ce jour consacré à l'OEP en tant qu'outil d'optimisation (c'est-à-dire en adoptant un point de vue le plus généraliste possible). À travers la définition de ses grandes lignes directrices, certaines grandes tendances actuelles de recherches en matière d'OEP seront ainsi évoquées.

### **1.10.1 Le projet XPS**

Piloté depuis l'université d'Essex (en Angleterre), le projet d'envergure internationale "eXtended Particle Swarm" (XPS) se veut explorer les possibles extensions de l'OEP.

Pour ce faire, ce projet se définit 4 grands axes ; l'axe physique, l'axe biologique, l'axe Théorique, et l'axe ingénierie dont la présentation est faite ci-dessous. Ce projet démarra fin 2004, est actuellement en cours, et présente certains de ses premiers résultats sur son site [site web 1].

### **1.10.2 L'axe physique**

L'axe physique consiste à construire un nouveau modèle amenant la perception de l'essaim particulaire sous un nouvel angle de vue. L'idée est d'envisager les particules comme les éléments d'un certain système, et l'évolution de ce système comme étant régie par la superposition des différents "champs d'influences" qu'émettent chacune des particules et dont résultent leurs interactions. Intuitivement, on perçoit facilement l'analogie inspirée par la physique et qui vaut à cet axe sa dénomination.

### **1.10.3 L'axe théorique**

L'axe théorique a pour but d'accroître les outils théoriques disponibles à ce jour pour la conception d'OEP (typiquement, l'établissement analytique de critères de convergence plus consistants). Pour ce faire, cet axe s'appuie notamment sur l'axe physique en espérant que le nouveau modèle amène par celui-ci permettra l'établissement d'un formalisme mathématique permettant l'établissement des résultats théoriques espérés.

### **1.10.4 L'axe biologique**

L'axe biologique se veut étudier les systèmes biologiques auto-organisés en vue de s'en inspirer pour la conception de possibles extensions de l'OEP. En effet, ces systèmes ont déjà inspiré la conception de bon nombre de méta-heuristiques (Algorithmes Génétiques, Algorithmes ACO, OEP,...), et il est donc légitime de penser que de leur étude pourrait amener à la conception d'extensions performantes de l'OEP. À titre indicatif, précisons la dite

notion d'auto-organisation. J. Dreco et P. SIARRY en proposent [Dreco et Siarry 04] la définition suivante: " L'auto-organisation caractérise un processus au cours duquel une structure émerge au niveau global uniquement d'un grand nombre d'interactions entre les composants de niveau local du système. De plus, les règles spécifiant les interactions entre composants du système sont suivies en utilisant uniquement des informations locales, sans référence au modèle global ".

### **1.10.5 L'axe ingénierie**

Enfin, l'axe ingénierie est consacré à la conception (à proprement parler) d'extensions du schéma de l'OEP en s'appuyant sur les résultats des autres axes.

## **1.11 Conclusion**

Nous avons présenté dans ce chapitre un état de l'art sur la méthode d'optimisation par essaim particulaire .L'OEP est une nouvelle technique d'optimisation qui exploite l'aspect social et l'aspect coopératif entre les agents qui peuvent être vus comme des « animaux » aux capacités assez limitées (Peu de mémoire et de facultés de raisonnement. Sa relative simplicité et son efficacité en font un algorithme le plus utilisé dans le domaine d'optimisation difficile. La plus part des travaux réalisés dans le cadre d'OEP visent l'étude du comportement des particules durant le processus de recherche pour déterminer les meilleures valeurs des paramètres ainsi que l'établissement des heuristiques qui sont utilisés comme des directives pour le choix des bonnes valeurs.

**CHAPITRE 2**

**L' OPTIMISATION MULTI-OBJECTIFS PAR**  
**ESSAIM DE PARTICULES**

## 2.1. Introduction

Dans ce chapitre nous avons présentés quelques notions nécessaires pour aborder les problèmes d'optimisations multi-objectifs, nous nous montrons aussi la difficulté issue de ces problèmes ainsi les différentes approches de résolutions. Pour terminer, nous présentons les différents travaux réalisés dans le cadre d'optimisation multi-objectif par l'essaim particulaire avec une vision critique de chaque travail.

Ce chapitre est consacré aussi à l'évaluation des différentes techniques d'adaptation du facteur d'inertie. Ces techniques sont utilisées dans un algorithme agrégatif d'optimisation multi objectif par essaim particulaire et testé sur des problèmes bi-objectifs de différentes natures, le but est de déterminer la meilleure approche d'adaptation. Un nouveau modèle est aussi proposé pour surmonté le problème de la perte de la diversité à cause de l'utilisation d'une seule technique d'adaptation, ce nouveau modèle est basé sur les trois approches d'adaptation: linéaire, non linéaire et aléatoire.

## 2.2. Définitions

### 2.2.1 Problème multi objectif

Les problèmes d'optimisation multi-objectifs (MOO) correspondent à des problèmes de recherche dont les solutions respectent différents objectifs, il peut être défini comme un problème dont on recherche une solution qui satisfait un ensemble de contrainte et optimise un vecteur de fonctions objectif.

D'une façon générale, un problème multiobjectif est exprimé par :

$$\left\{ \begin{array}{l} \min F(x) = (f_1(x), f_2(x), \dots, f_n(x)), x \in S \subset R^m \\ g_i(x) \leq 0, i = 1..p \\ h_j(x) = 0, j = 1..k \\ x_{k,\min} \leq x_k \leq x_{k,\max}, k = 1..m \end{array} \right\} \quad (2.1)$$

Où  $F(x) = (f_1(x), f_2(x), \dots, f_n(x))$  est le vecteur de l'ensemble de fonctions objectif, le vecteur  $x(x_1, x_2, \dots, x_m)$  et le vecteur des actions de décisions,  $g_i(x)$  sont les contraintes d'inégalités,  $h_j(x)$  représentent les contraintes d'égalités et  $x_{k,\min} \leq x_k \leq x_{k,\max}, k = 1..m$  sont les contraintes de domaine.

Le but de l'optimisation multi-objectif est de minimiser un groupe de fonctions objectifs sans trop dégrader les valeurs des optima obtenus par rapport à ceux obtenus lors d'une optimisation mono-objectif effectuée objectif par objectif.

Donc un problème d'optimisation recherche l'action  $x^*$  telle que les contraintes  $g_i(x^*)$ ,  $i=1..p$ , et  $h_j(x^*)$ ,  $j=1..k$ ,  $x_{k,\min} \leq x_k^* \leq x_{k,\max}$ ,  $k=1..m$  soient satisfaites et qu'il optimise la fonction  $F(x^*) = (f_1(x^*), f_2(x^*), \dots, f_n(x^*))$ .

L'union de définition de chaque variable et les contraintes définies dans l'équation 2.1 forme un ensemble  $E$  qui s'appel l'ensemble des actions réalisables (figure 2.1)

$F$  s'appel l'ensemble des objectifs réalisables voir figure 2.2:

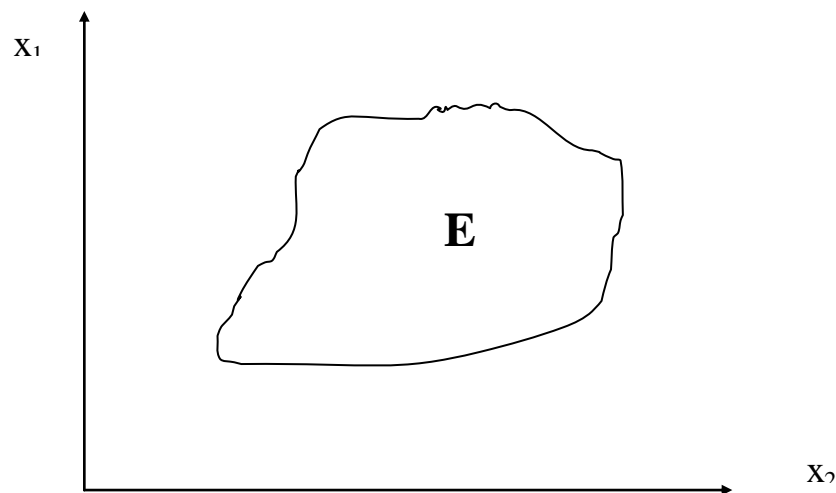


Figure 2.1: Ensembles des actions réalisables dans un problème multi-objectifs  $x=(x_1, x_2)$

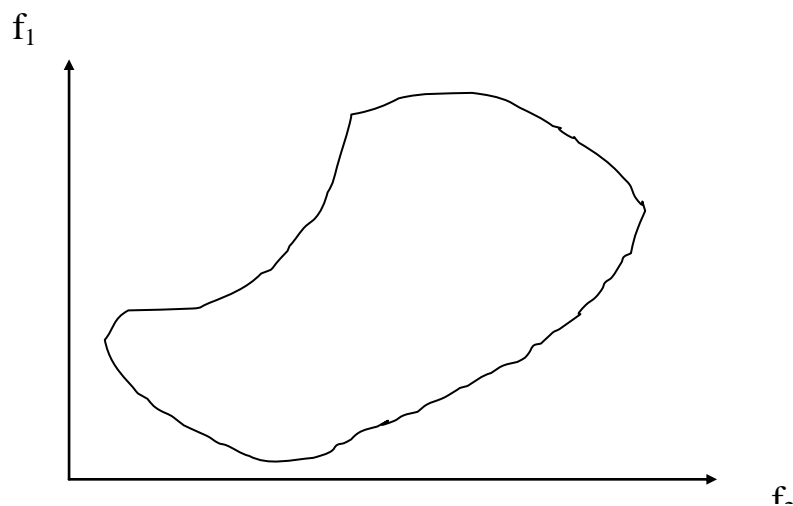


Figure:2.2 Ensembles des objectifs réalisables dans un problème multi-objectifs  $f=(f_1, f_2)$



### 2.2.2 Le vecteur idéal

Le vecteur idéal est le vecteur qui se compose de la solution optimale de chaque fonction objective pris séparément.

$$x^* = (x_1^*, x_2^*, \dots, x_n^*) \quad (2.2)$$

Avec  $x_i^*$  est le vecteur de la solution optimale de la fonction objective  $i$ .

Le vecteur idéal est atteint si seulement si les fonctions objectives soient indépendantes, dans ce cas le problème multiobjectif se transforme en un problème mono-objectif.

### 2.2.3 Convexité

L'ensemble des objectifs réalisables  $F$  est convexe si tout segment joignant de point de  $F$  est inclus dans figure 2.3.

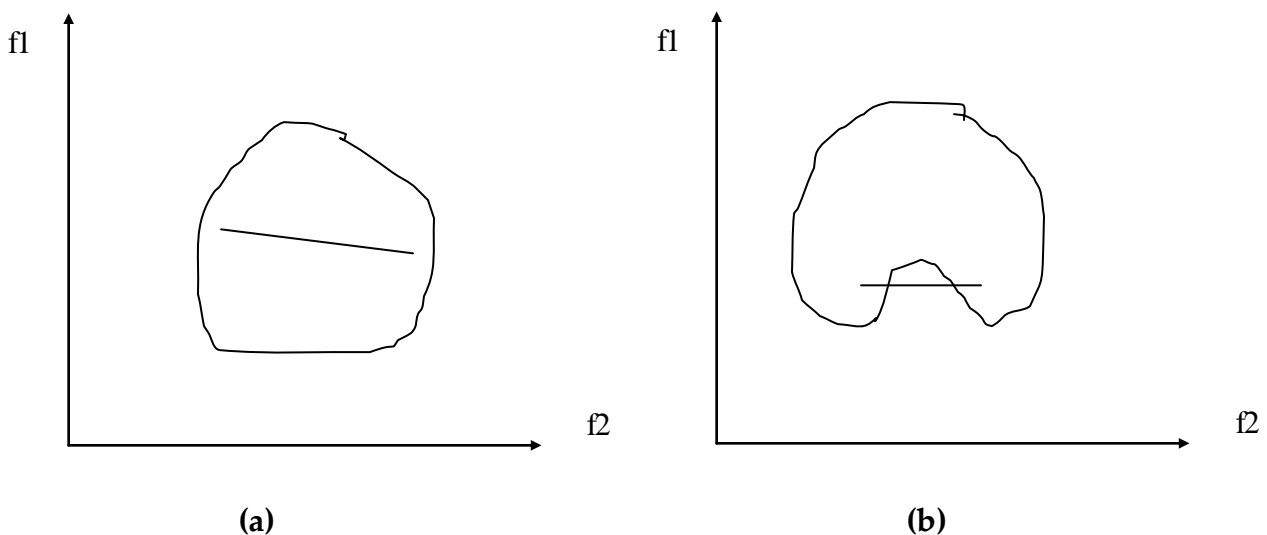


Figure 2.3 : (a) espace convexe

(b) espace non convexe

## 2.3 Multiplicité de solutions

La résolution d'un PMO ne donne pas une solution unique mais plusieurs solutions possibles. Ceci est dû au fait que, souvent, les PMO, présentent des objectifs contradictoires (la diminution d'un objectif entraîne l'augmentation de l'autre).

Donc la résolution d'un PMO, donne un ensemble de solutions non optimales, car elles ne minimisent pas toutes les fonctions objectives. Ce sont des solutions de compromis

## 2.4 Difficulté d'un problème multi objectif

Pour un problème multi objectif il n'existe pas une définition d'une solution optimale, mais simplement on peut exprimer qu'une solution est préférable à une autre et qu'il n'existe pas une solution meilleure que les autres.

Dés lors résoudre un problème multi objectif ne consiste pas à recherché une solution optimale mais une surface de solutions satisfaisantes pour lesquelles on ne peut pas faire un classement.

Pour répondre à ce problème il y a deux modes de résolution. Le premier est de transformer un problème multi objectif à un simple problème mono objectif. Le seconde est de tenter de d'apporter des réponses au problème on prenant en compte l'ensemble des critères. Le schéma 2.4 montre les différences entre les deux approches.

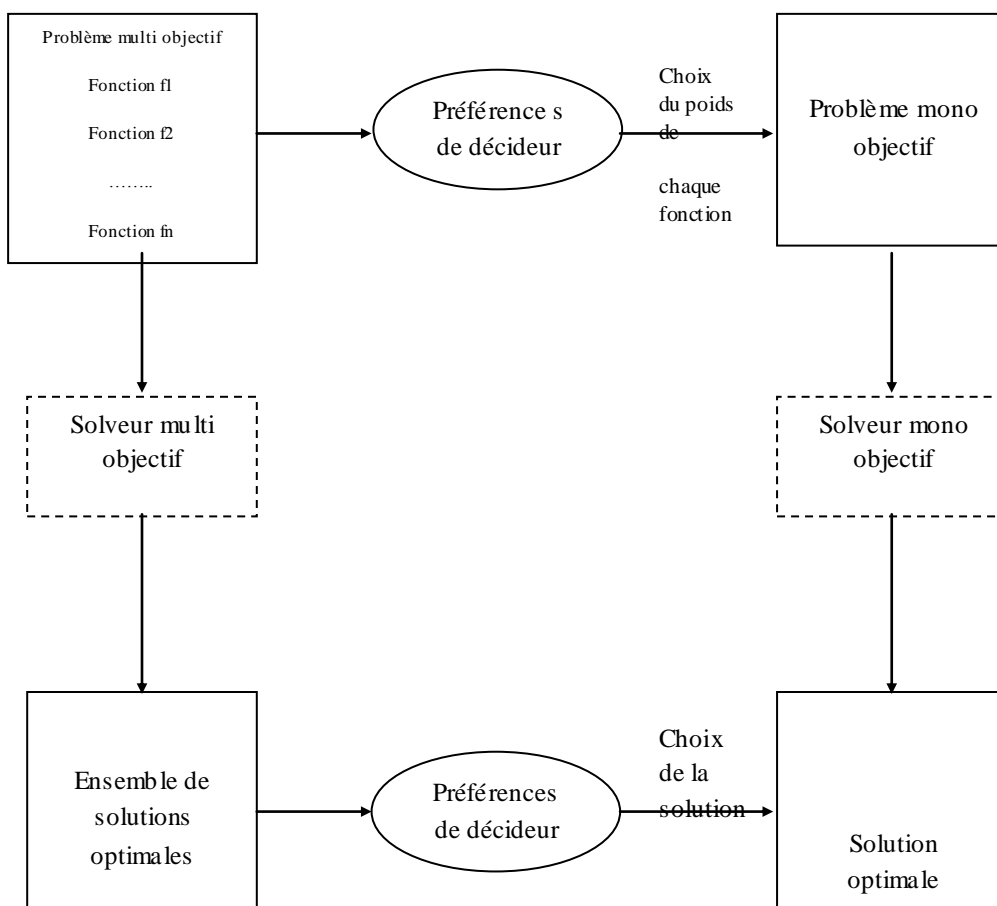


Figure 2.4 Modes de résolution d'un problème multi-objectif

## 2.5 Classification des méthodes d'optimisation multi objectifs

Dans les différents publications nous retrouvant trois classes d'algorithmes d'optimisations multi objectifs : les méthodes agrégées, les méthodes non agrégées non Pareto et les méthodes Pareto.

### 2.5.1 Les méthodes agrégées

Ces méthodes consistes à transformer un problème multi objectif à un problème mono objectif [Collette, et al., 2002b], elles se reposent sur l'axiome suivante : tout décideur essaye d'optimiser une fonction d'utilité  $U=U(f_1, f_2, \dots, f_k)$

Deux modèles sont proposés pour cette approche :

- le modèle additif : c'est le problème le plus utilisé couramment

$$U = \sum_{i=1}^k U_i(f_i) \quad (2.3)$$

- le modèle multiplicatif :  $U = \prod_{i=1}^k U_i(f_i)$  (2.4)

Dans cette approche il existe de nombreuses techniques de résolution [Collette, et al., 2002]. Parmi les techniques qui utilisent cette approche, nous pouvons citer la technique de la moyenne pondérée, les techniques  $\epsilon$ -contraintes, les techniques de programmation par but et min-max [Coello, 1995], goal programming [Coello 1996] [Van Veldhuizen 1999] [Ignizo, 1981] [Van Veldhuizen, 1999]. Le tableau suivant donne les formules utilisées pour chaque technique :

**Tableau 2.1 Différentes méthodes d'optimisation fondées sur l'approche agrégative**

Désignation de la technique	Formule	Description
la moyenne pondérée	$\min \sum_{i=1}^k w_i f_i(x)$	- $w_i$ : poids de chaque fonction $f_i$ avec $\sum_{i=1}^k w_i = \mathbf{1}$
E - contraintes	Minimiser $f_i(x)$ avec $f_j(x) \leq \varepsilon_j \forall j \neq i$	Choisir un objectif $f_i$ , cet objectif doit être optimisé en priorité, considérant que les autres objectif $f_j$ avec $j \neq i$ doivent être inférieurs à $\varepsilon_j$

min-max	$\min \max_i \left( \frac{f_i(x) - T_i}{T_i} \right) i = 1..k$	Minimiser l'écart relatif entre un objectif $f_i$ et son but à atteindre $T_i$
goal programming	$\min \sum_{i=1}^k  f_i(x) - T_i $	Minimiser la somme des écarts entre les résultats et son but à atteindre $T_i$

### 2.5.2 Avantages et inconvénients des modèles agrégées

Malgré que la transformation du problème à un simple problème mono objectif facilite la résolution de ce dernier, mais l'utilisation de cette méthode impose que :

- Les objectifs soient commensurables.
- Tous sous-ensembles doit être préférentiellement indépendantes dans F.
- Le décideur est supposé connaître au priori les poids de chaque fonction objectif afin de les mélanger dans une fonction d'utilité unique.

### 2.5.3 Les méthodes non agrégées non Pareto

Généralement, les méthodes basées sur cette approche possèdent un processus qui traite les objectifs séparément. Deux groupes de méthodes existent dans la littérature : sélection lexicographique et sélection parallèle.

Dans l'approche classique de la sélection lexicographique, la sélection est réalisée suivant un ordre défini a priori. Cet ordre permet de définir les poids des objectifs. Plusieurs métaheuristiques ont été utilisées pour la résolution des problèmes multi-objectifs avec sélection lexicographique [Talbi, 1999]. Dans l'approche par sélection parallèle, le premier travail a été publié par Schaffer [Schaffer, 1985], il est basé sur un algorithme génétique. Son algorithme, appelé VEGA (*vector evaluated genetic algorithm*), sélectionne les solutions courantes du front Pareto suivant chaque objectif, indépendamment des autres (sélection parallèle). L'analyse de cet algorithme a montré que son comportement est le même qu'un algorithme réalisant une agrégation linéaire.

### 2.5.4 Les méthodes Pareto

Ces méthodes s'appuient sur la dominance au sens de Pareto [Goldberg 1989], contrairement aux autres méthodes, elles ne combinent pas les critères et ne les traitent pas

séparément, ces méthodes respectent aussi l'intégrité de chaque critère, car les valeurs des différents critères ne sont pas comparées a priori. Ces méthodes convergent vers une surface de solution (solutions de Pareto), Le choix de la solution finale donc revient à l'utilisateur [talbi 1999].dans les sections suivantes ne donnant des définitions basées sur les travaux de Pareto [Pareto, 1896].

### 2.5.4.1 La notion de dominances

Soient  $x, y$  deux solutions d'un problème multiobjectif définis dans l'espace  $\mathbb{R}^K$ , on dit que  $x$  domine  $y$  ( $x \prec y$ ) si :

$$\forall i, f_i(x) \leq f_i(y) \text{ et } \exists j, f_j(x) < f_j(y) \quad (2.5)$$

Une solution  $x$  est dite faiblement non dominée si :

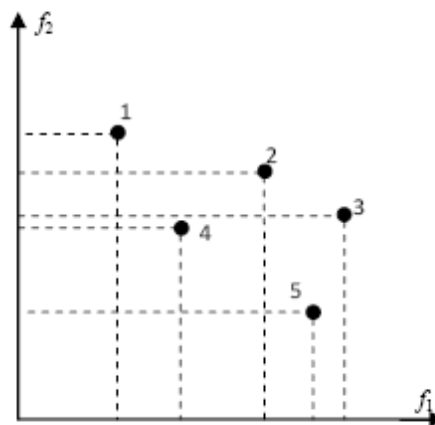
$$\neg \exists y, f_i(y) < f_i(x) \quad (2.6)$$

une solution  $x$  est dite fortement non dominée si :

$$\neg \exists y, f_i(y) \leq f_i(x) \text{ et } \exists i, f_i(x) < f_i(y) \quad (2.7)$$

avec  $i=1..n$  et  $n$  est le nombre des fonctions objectifs.

La figure 2.5 suivante illustre la notion de la dominance dans un problème bi-obejectif :



**Figure 2.5: Notion de la dominance**

Dans cet exemple l'ensemble des points non dominés est :  $\{1, 4, 5\}$

Le point 5 domine le point 3 et le point 4 domine les points 2 et 3, les autres points ne domine aucun point.

### 2.5.4.2 La notion de Pareto optimal

On dit qu'un point  $x \in E \subset R^D$  est un Pareto optimal s'il n'est dominé par aucun point dans E. dans la figure précédente le point 1 n'est pas dominé par aucun point donc il est un Pareto optimal.

L'ensemble des points optimaux de Pareto est défini par :

$$P^* = \{x \in E \mid x \text{ est un Pareto optimal}\} \quad (2.8)$$

### 2.5.4.3 Le front de Pareto

Le front Pareto est l'ensemble de tous les points Pareto-optimaux. Il est défini par :

$$F_{P^*} = \{f(x) = R^k \mid x \in P^*\} \quad (2.9)$$

La figure 2.6 illustre le front de Pareto dans les deux cas convexe et concave dans un problème bi-objectifs.

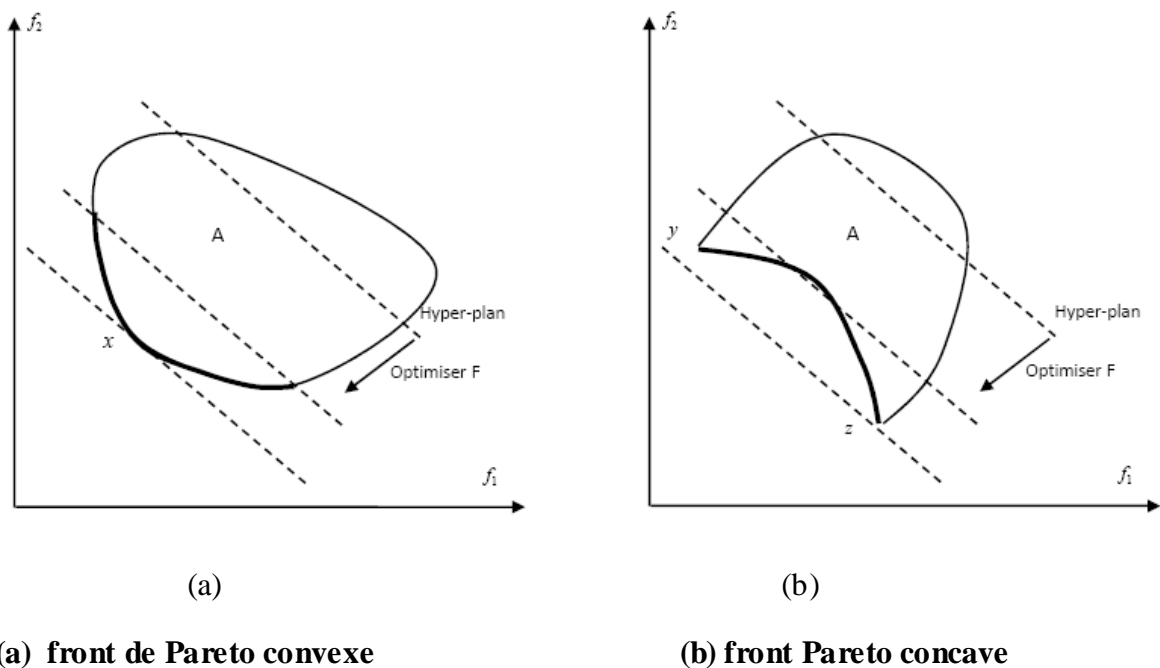


Figure 2.6 : le front de Pareto

## 2.6 Etat de l'art de l'optimisation par l'essaim des particules en optimisation multi-objectifs

Afin d'appliquer la stratégie de OEP pour résoudre des problèmes d'optimisation multi-objectifs, il est évident que le schéma original doit être modifié. Comme nous avons

vu dans la section 2.3 dans ce chapitre, l'ensemble de solutions d'un problème multi-objectifs ne se compose pas d'une solution simple (comme dans l'optimisation mono-objectif). Au lieu de cela, dans l'optimisation multi-objective, nous visons à trouver un ensemble de différentes solutions (ensemble optimal de Pareto). En général pour résoudre un problème multi-objectif, trois buts principaux doivent être prises en compte [Zitzler et al., 2000]:

- Maximiser le nombre des éléments de l'ensemble optimal de Pareto.
- Minimiser la distance entre le front de Pareto trouvé par notre algorithme par rapport au front de Pareto réel.
- Maximiser la distribution des solutions trouvées de telle sorte qu'elle soit uniforme le plus que possible.

Afin d'étendre l'OEP à l'optimisation multiobjectif, il a fallu modifier la formulation originale du schéma d'un problème multiobjectif. Pour cela il a fallu répondre aux trois questions suivantes [Carlos et al, 2002]:

1. Comment sélectionner les particules pour donner la préférence à celle qui est non dominée par rapport aux autres ?
2. Comment retenir les solutions Pareto optimales par rapport à celles déjà vues et rencontrées au cours de la recherche ?
3. Comment maintenir la diversité dans l'essaim pour éviter la convergence vers une seule solution ?

Dans cette section nous présentons le principe général de l'OEP multi-objectifs ainsi que les différents travaux réalisés, ces travaux sont classés selon les trois approches utilisées dans le traitement des problèmes multiobjectif : approches agrégatives, approches non Pareto, approche Pareto. nous avons proposés aussi un algorithme OEPMO à base d'une méthode d'agrégation dynamique qui utilise trois techniques d'adaptation du facteur d'inertie pour préserver mieux la diversité au sein de l'Essaim.

### 2.6.1 Principe général de l'OEP multi-objectifs

Contrairement au OEP mono objectif, dans OEPMO le meilleur point par lequel elle est déjà passée une particule est remplacé par les solutions non dominées générées dans le passé durant le processus de recherche .le résultat d'exécution de l'algorithme est donc un ensemble de solutions non dominées vis-à-vis de toutes les solutions atteintes par les

particules au cours d'exécution, on appelle cette ensemble l'archive externe. Il est donc nécessaire de mettre à jour l'archive externe à la fin de chaque itération.

Une règle est utilisée pour mettre à jour l'archive consiste à ajouter une solution à cet archive si seulement si elle est non dominée vis-à-vis à son contenu et retiré des solutions de l'archive qui sont dominées par cette solution.

L'algorithme 2.1 présente le principe général d'OEP multiobjectif.

\*\*\*\*\*

**Algorithme 2.1 : Algorithme général d'OEP multiobjectif**

\*\*\*\*\*

Initialisation aléatoire des positions et des vitesses de chaque particule

Initialiser la mémoire cognitive de chaque particule.

Déterminer les leaders

Stockage les leaders dans l'archive externe (solution non dominées)

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

**Pour**  $i = 1$  à  $N$  **faire**

    Sélectionner un leader social pour chaque particule

    Calculer la vitesse

    Calculer la nouvelle position

    Déplacer la particule

    Evaluer la position de chaque particule

    Mise à jour la mémoire cognitive

**Fin Pour**

Mise à jour des leaders sociaux

Mise à jour des solutions non-dominées

**Fin Tant que**

Retourner les résultats de l'archive externe

\*\*\*\*\*



Après l'initialisation de la position et la vitesse de chaque particule, un ensemble de leaders est également déterminés avec les particules non-dominées de l'essaim. Comme nous avons déjà mentionné, l'ensemble de leaders est souvent stocké dans des archives externes. Ensuite, un leader social est choisi pour chaque particule de l'essaim. A chaque génération, pour chaque particule, un leader social est choisi et la vitesse est mise à jours selon les trois composants (la vitesse courante, le leader social et la mémoire cognitive). La particule est ensuite évaluée et la valeur de la mémoire cognitive (la meilleure position qu'elle a atteinte jusqu'ici) correspondante est mise à jour. Une nouvelle position de particule remplace sa mémoire cognitive habituellement quand cette position de particule domine sa meilleur position ou si elles sont toutes les deux non-dominée l'une de l'autre. Après la mise à jour de toutes les particules, l'ensemble de leaders est mise à jour aussi. Finalement, la mesure de qualité de l'ensemble de guides est recalculée. Ce processus est répété pour un certain nombre d'itérations.

### 2.6.2 Les difficultés d'adaptation

Pour adapter l'algorithme OEP de base à la résolution des problèmes multi-objectifs, on est confronté à deux difficultés majeures [Pulido, 2005].

La première difficulté rencontrée concerne le choix et la mise des leaders, Pour cela il a fallu répondre aux deux questions suivantes :

- Comment choisir un seul guide de l'ensemble des solutions non-dominées qui sont toutes bonnes, on peut le choisir d'une manière aléatoire ou on doit utiliser un critère additionnel (pour favoriser la diversité, par exemple).
- Comment choisir les particules qui devraient demeurer dans les archives externes d'une itération à l'autre.

### 2.6.3 Leaders dans l'optimisation multiobjectif

Puisque la solution d'un problème multiobjectif se compose d'un ensemble de bonnes solutions, il est évident que le concept de leader traditionnellement adopté dans OEP doit être changé. Afin d'éviter la définition d'un nouveau concept de leader pour des problèmes multi-objectifs, certaines méthodes utilisent des fonctions d'agrégation (sommées pondérées des objectifs) ou approches qui optimisent chaque objectif séparément. Cependant, il est important d'indiquer que la majorité des approches actuellement proposées d'OEPMO redéfinissent le concept de leader.

Comme mentionné plus haut, le choix d'un leader est une composante importante dans la conception d'OEPMO. L'approche la plus directe est de considérer chaque solution non-dominée comme un nouveau leader et puis, un seul leader étant choisi. De cette façon, une mesure de qualité qui indique la qualité d'un leader est très importante. Evidemment, une telle approche peut être définie de différentes manières. Des différentes propositions, pour traiter ce problème, seront présentées plus loin.

Une manière possible de définir une telle mesure de qualité peut être les mesures de densité. La promotion de la diversité peut être faite par ce processus au moyen de mécanismes basés sur quelques mesures de qualité qui indiquent la proximité des particules dans l'essaim. Plusieurs auteurs ont proposé des techniques de choix de leader qui sont basées sur des mesures de densité, nous présentons ici deux des plus important mesures de densité utilisées dans le domaine de l'optimisation multiobjectif:

### 2.6.4 Estimateur de densité de voisin le plus proche

L'estimateur de densité de voisin le plus proche nous donne une idée de la façon dont les voisins les plus proches d'une particule donnée sont distribués, dans l'espace de fonction objectif [Deb et al, 2002]. Cette mesure estime le périmètre du cuboïde formé en employant le plus proche voisin comme sommet (figure 2.7).

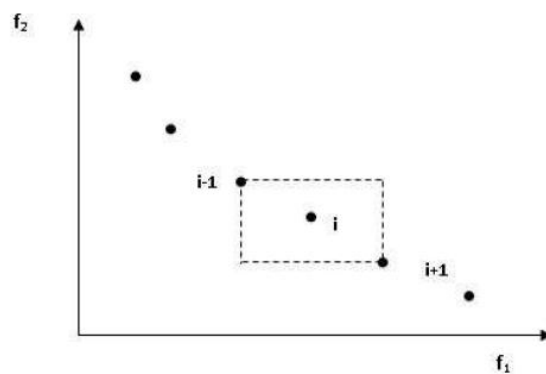


Figure 2.7 : Exemple d'estimateur de densité de voisin le plus proche

### 2.6.5 Estimateur de densité de grain

Quand une particule partage les ressources avec d'autres particules, sa fitness est dégradée proportionnellement au nombre et proximité des particules qui l'entourent avec un certain périmètre seuil [Goldberg et Richardson, 1987] [Deb et Goldberg, 1989]. Un

voisinage d'une particule est défini en termes de paramètre noté  $\sigma_{share}$  qui indique le rayon de voisinage. De tels voisinages s'appellent niches écologiques (figure 2.8).

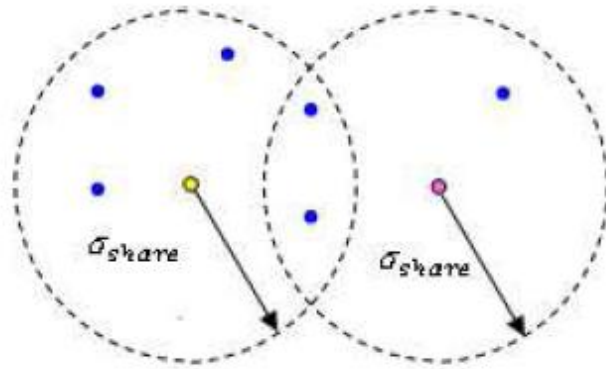


Figure 2.8 : Niches de particules

### 2.6.6 Conservation et propagation des solutions non-dominées

Comme déjà mentionné, il est important de maintenir les solutions non-dominées trouvées le long de tout le processus de recherche et ainsi pouvoir retourner à la fin ces solutions non-dominées en tenant compte de toutes les populations précédentes. Ceci est important non seulement pour des raisons pragmatiques, mais également pour les raisons théoriques [Rudolph, 1998].

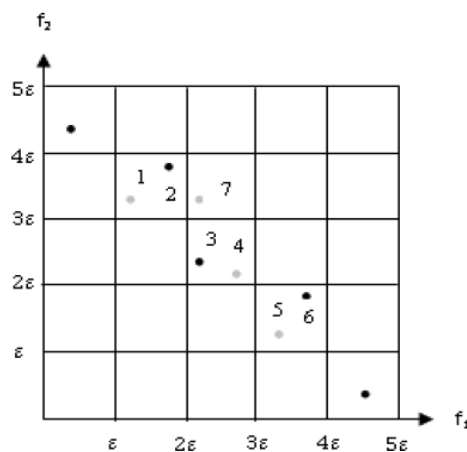
La manière la plus directe de maintenir des solutions non-dominées, en prenant en considérations toutes les populations précédentes (ou essaims), est d'employer des archives externes. De telles archives permettra l'ajout d'une solution seulement si elle est non-dominée par une solution enregistrée dans l'archive ou si elle domine une des solutions de l'archive (dans ce cas, les solutions dominées doivent être supprimés de l'archive).

L'inconvénient de cette approche est l'augmentation très rapide de la taille des archives. C'est un point important parce que les archives doivent être mises à jour à chaque génération. Ainsi, cette mise à jour peut devenir très coûteuse en temps de calcul si la taille des archives est importante. Dans le pire des cas, tous les membres de l'essaim peuvent entrer dans l'archive, à chaque génération. Ainsi, le processus de la mise à jour correspondant, à chaque génération, aura une complexité de  $o(kN^2)$ , où  $N$  est la taille de l'essaim et  $k$  le nombre des objectifs. De cette façon, la complexité du processus de mise à jour pour l'exécution complète de l'algorithme est de  $(kMN^2)$ , où  $M$  est le nombre total d'itérations.

Cependant, il est nécessaire d'ajouter un critère pour décider quelles solutions non-dominées doivent être maintenues dans le cas où l'archive est pleine. Dans l'optimisation

multiobjectif évolutionnaire, les chercheurs ont adopté différentes techniques pour réduire la taille des archives. D'autres concepts ont été introduits pour l'utilisation des archives, par exemple, pour ajouter les éléments dans l'archive, la distribution des solutions a été utilisée comme critère additionnel au lieu d'utiliser uniquement le concept de non dominance.

Il faut noter qu'on doit utiliser trois archives pour adapter OEP à l'optimisation multiobjectif : une pour stocker les meilleures solutions globales, une pour les meilleures valeurs pbest et une troisième pour stocker la meilleure solution locale. Cependant, dans la pratique, quelques auteurs rapportent l'utilisation de plus d'une archive dans leur OEPMOs. Plus récemment, d'autres chercheurs ont proposé l'utilisation de formes relaxées de dominance. La plus principale a été la méthode  $\varepsilon$ -dominance [Laumanns et al, 2002]. Le but était de sauvegarder les solutions non dominées dans des archives externes. En utilisant le paramètre  $\varepsilon$ , on définit un ensemble de cases de taille epsilon, une seule solution non-dominée est maintenue pour chaque case (située à la limite gauche et inférieure de chaque case). Comme illustré dans la figure 2.9.



**Figure2.9 : Exemple d'utilisation de  $\varepsilon$  dominance dans un archive externe**

Comme le montre la figure 2.9, la solution 1 domine la solution 2 ; donc la solution 1 est maintenue. Les solutions 3 et 4 sont non dominées l'une de l'autre, mais solution 3 est mieux que 4, puisque solution 3 est le plus proche au coin à gauche inférieur représenté par le point  $(2\varepsilon; 2\varepsilon)$ . Solution 5 domine solution 6, donc solution 5 est maintenue. Solution 7 est non acceptée puisque sa case représentée par le point  $(2\varepsilon; 3\varepsilon)$  est dominée par la case représentée par le point  $(2\varepsilon; 2\varepsilon)$ .

Pour un cas Bi-objectif, l'utilisation du  $\epsilon$ -dominance, comme proposé dans [Laumanns et al, 2002], garantit que les solutions maintenues sont non-dominées en tenant compte de toutes les solutions produites pendant l'exécution. En utilisant  $\epsilon$ -dominance, la taille de l'archive externe final dépend de la valeur  $\epsilon$ , qui est normalement un paramètre défini par l'utilisateur [Laumanns et al, 2002].

### 2.6.7 Maintien de la diversité par création de nouvelles solutions

La convergence rapide est l'une des caractéristiques les plus importantes de l'algorithme OEP. Cependant, il est primordial de maintenir un certain degré de diversité pour éviter que l'algorithme soit piégé.

La convergence prématurée est provoquée par la perte rapide de diversité dans l'essaim. Ainsi, le maintien de la diversité dans OEP est un point très important afin de contrôler sa convergence (normalement rapide). Comme mentionné précédemment, en adoptant OEP pour résoudre des problèmes d'optimisation multi-objectifs, il est possible de favoriser la diversité par le choix de leaders. Cela peut être également fait par les deux principaux mécanismes utilisés pour créer de nouvelles solutions :

#### a. Mise à jour des positions

L'utilisation de différentes topologies de voisinage détermine la vitesse du processus de transfert de l'information à travers l'essaim. Cependant, dans une topologie entièrement reliée, toutes les particules sont reliées les unes avec les autres, l'information est transférée plus rapidement que dans le cas de topologie locale ou d'arbre. Aussi, une topologie spécifique de voisinage détermine également la vitesse de perte de diversité dans l'essaim. Puisque dans une topologie entièrement reliée le transfert d'information est rapide, en employant cette topologie, la diversité dans l'essaim est également perdue rapidement. De cette façon, les topologies qui définissent des voisinages plus petits que l'essaim global pour chaque particule peuvent également préserver la diversité dans l'essaim.

D'autre part, la diversité peut également être favorisée par le facteur d'inertie  $w(t)$  de l'équation (1.2) Le facteur d'inertie est utilisé pour contrôler l'impact des vitesses antérieures sur la vitesse courante. Ainsi, le poids d'inertie influence la différence entre les capacités d'exploration globales et locales [Shi et Eberhart, 1998]. Un grand facteur d'inertie facilite l'exploration globale tandis qu'un plus petit facteur d'inertie tend à faciliter l'exploration locale. La valeur du facteur d'inertie peut varier pendant le processus d'optimisation. Shi [Shi

et Eberhart, 1998] a montré qu'en diminuant linéairement le poids d'inertie d'une valeur relativement grande à une petite valeur durant l'exécution de OEP, l'algorithme favorise une recherche globale au début de son exécution et une recherche locale à la fin. L'addition de la vitesse à la position actuelle pour produire la prochaine position est semblable à l'opérateur de mutation dans des algorithmes évolutionnaires, sauf que la mutation dans OEP est guidée par l'expérience d'une particule et de celle de ses voisines.

### **b. L'utilisation d'un opérateur de mutation (ou turbulence)**

Quand une particule met à jour sa position, une mutation se produit. Parfois, une turbulence est aussi nécessaire. La turbulence reflète le changement du vol des particules qui est hors de son control [Fieldsend et Singh, 2002].

En général, quand un essaim stagne, c-à-d., quand les vitesses des particules sont pratiquement nulles, il devient incapable de produire de nouvelles solutions qui pourraient mener l'essaim hors de cet état. Ce comportement peut mener l'essaim entier à être emprisonné dans un optimum local duquel il est impossible de s'échapper. Puisque le meilleur individu global attire tous les membres de l'essaim, il est possible de mener l'essaim loin d'un endroit courant grâce à la mutation d'une particule simple si la particule mutée devient le nouveau leader. Ce mécanisme permet à la fois de s'échapper des optima locaux et d'accélérer la recherche [Stacey et al, 2003].

De cette façon, l'utilisation d'un opérateur de mutation est très importante afin de s'échapper des optima locaux et d'améliorer les capacités d'exploration de PSO. En fait, différents opérateurs de mutation ont été proposés qui permettent la mutation des composants de la position ou de la vitesse d'une particule.

Le choix d'un bon opérateur de mutation est une tâche difficile qui a un impact significatif sur l'exécution. D'autre part, une fois un opérateur spécifique de mutation est choisi, une autre tâche difficile est de savoir le nombre de mutation à appliquer : avec quelle probabilité, dans quelle étape du processus, dans quel élément spécifique d'une particule, etc. Plusieurs approches proposées ont employé des opérateurs de mutation, néanmoins, d'autres approches qui n'utilisent pas d'opérateurs de mutation ont donné de bonnes performances.

## 2.6.8 Classification des approches OEP pour les problèmes multi-objectifs

On peut classifier les OEPMOs de la manière suivante :

- Approches agrégées.
- Approches basées sur Pareto.
- Approches non Pareto.

Ces différents modèles seront présentés dans les paragraphes suivants.

### 2.6.8.1 OEP multi-objectif pour l'approche agrégative

Les différents travaux réalisés dans cette approche utilisent le modèle additif défini précédemment dans la section 2.5.1 qui transforme le problème en un seul objectif.

#### a- Algorithme de Parsopoulos et Vrahatis

Parsopoulos et Vrahatis [Parsopoulos et al., 2002] proposent d'agréger les objectifs suivant trois méthodes différentes : une agrégation linéaire classique (conventionnl weighted agrégation **CWA**), pour laquelle les poids sont fixés, une agrégation dynamique (dynamic weighted aggregation **DWA**), pour laquelle les poids sont modifiés au cours du traitement et une agrégation dite bang-bang '(bang-bang weighted aggregation **BWA**) pour laquelle les poids sont brutalement modifiés au cours du temps. Dans cette section nous présentons les trois approches en détaille et nous montrons les avantages et les limites de chaque approche.

#### - Agrégation linéaire classique (**MOPSOCWA**)

Dans cette technique les poids sont fixés une fois pour toute avant l'exécution de l'algorithme, en utilisant cet approche seulement un point Pareto optimal peut être obtenu au cours du processus d'optimisation et une connaissance a priori de l'espace de recherche est exigée afin de choisir les poids appropriés. Ainsi, la recherche doit être répétée plusieurs fois pour obtenir un nombre désiré de points optimaux de Pareto (front de Pareto), cependant cette technique n'est pas souhaitée pour la pluparts des problèmes dus aux couts calculatoire très élevés.

Des expérimentations réalisées par Parsopoulos et Vrahatis montre que, MO PSOCWA est capable de détecter la partie convexe du front de Pareto .si la fonction

d'agrégation est concave, cette approche permet de détecter seulement les deux points extrêmes de celui-ci.

**- Agrégation bang-bang (MOPSOBWA)**

Cette approche est proposée pour alléger les limitations du MOPSOBWA . Pour un problème bi-objectif, les poids peuvent être modifiés pendant l'optimisation, selon les équations suivantes:

$$w_1(t) = \text{sign}(\sin(2\pi t/F)) \quad (2.10)$$

$$\text{et } w_2(t) = 1 - w_1(t) \quad (2.11)$$

Où  $t$  est l'indice de l'itération et  $F$  représente la fréquence de changement du poids.

dans cette approche dite bang-bang agrégation, le changement brutale des poids est dus à la fonction  $\text{sign}()$ .des expérimentations sur des fonctions de test de différents natures(concave, convexe, non uniforme) montre que cette technique donne des résultats mieux que les deux autres approches , à l'exception des fonctions concaves où les résultats de l'approche MOPSOBWA sont mieux.

**- Agrégation dynamique (MOPSOBWA)**

Pour un problème bi-objectifs, les poids de chaque fonction sont changés graduellement durant le processus de recherche par:

$$w_1(t) = \left| \sin(2\pi t/F) \right| \quad (2.12)$$

$$w_2(t) = 1 - w_1(t) \quad (2.13)$$

Cette technique détecte bien la forme du front de Pareto dans le cas où la fonction d'agrégation est concave.

**b- L'approche de Baumgartner, Magele et Renhart :**

De même, Baumgartner [Baumgartner et al., 2004] utilisent une OEP avec une topologie complètement connectée combinée avec une agrégation linéaire des objectifs. L'essaim est divisé en  $n$  sous-essaims utilisant chacun un jeu de poids différent et possédant chacun leur propre leader. Les solutions optimales au sens de Pareto sont déterminées à l'aide d'une méthode basée sur l'utilisation du gradient.



### 2.6.8.2 OEP multi-objectif pour l'approche Non-Pareto

Ces approches concernent l'utilisation de plusieurs sous-populations, chaque sous-population traite un seul objectif. Les sous-populations effectuent ensuite un échange d'information ou une recombinaison visant à produire la diversité entre les différentes solutions précédemment produites pour les objectifs qui ont été séparément optimisés, dans cette section nous nous décrivons l'un des algorithmes les plus utilisés dans cette approche, cet algorithme est VEPSO (vector evaluated particle swarm).

#### a- L'algorithme VEPSO (vector evaluated particle swarm)

##### - *Principe général*

L'algorithme de l'optimisation par essaim particulaire évalué par vecteur VEPSO [Parsopoulos et al., 2004] a été inspiré par le concept de l'algorithme génétique évalué par vecteur VEGA [Schaffer, 1985], il est fondé sur l'approche non-Pareto pour détecter la forme du front de Pareto.

VEPSO est un algorithme multi-essaim pour lequel chaque sous-essaim traite un des objectifs, mais chaque sous-essaim échange son information (e.g., la meilleur solution retrouvé par chaque sous essaim) avec les autres sous-essaims. la meilleur position obtenue par une particule et les meilleurs positions obtenues par les sous essaims sont les directives principales utilisées pour le déplacement de l'essaim. Ainsi l'échange de cette information entre les sou-essaims peut mener à l'ensemble des points optimaux de Pareto.

Les auteurs montrent que cette méthode peut mener à des solutions Pareto optimales.

##### - *Description formelle*

Etant donné un problème multiobjectif de  $k$  fonctions objectives:  $f_l(x)$ ,  $l=1\dots k$ , et nous supposons que  $M$  sous essaims,  $S_1, S_2, \dots, S_m$ , de taille  $N$ , sont utilisés pour la résolution de ce problèmes où chaque fonction sera évalué en correspondance avec un sous essaim.

Etant donné aussi  $X_i^{[j]}, V_i^{[j]}, P_i^{[j]}, i=1..N, j=1..M$ , qui représentent respectivement la position courante, la vitesse et la meilleur position obtenue par la particule  $i$  dans le sous essaim  $j$  dans un temps donné. Nous acceptons aussi que  $g_{[j]}$  est l'index de la meilleur position obtenue par le sous essaim  $j$ .

Les particules de chaque sous essaims sont, en général, manipulées selon les équations de mouvements suivantes:

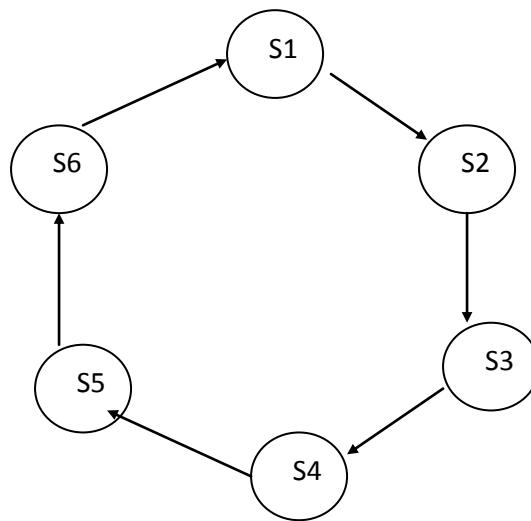
$$V_i^{[j]}(t+1) = w^j.V_i^{[j]}(t) + c_1^j r_1^j (P_i^{[j]} - X_i^{[j]}) + c_2^j r_2^j (P_{g[s]}^{[s]} - X_i^{[j]}) \quad (2.14)$$

$$X_i^{[j]}(t+1) = X_i^{[j]}(t) + V_i^{[j]} \quad (2.15)$$

où  $s$  représente un index appartient à  $\{1, \dots, j-1, j+1, \dots, M\}$

Plusieurs variantes sont proposées pour la sélection du paramètre  $s$ , par exemple on peut choisir  $s$  selon la méthode suivante:  $S=M$  si  $j=1$  et  $S=j-1$  si  $j=2, \dots, M$  [Plagianakos et Vrahatis.,2002 ]

Cette méthode de sélection du paramètre  $s$  correspondante à un schéma de migration des sous essaims en anneau (figure 2.10) .Un autre alternative est de choisir  $s$  aléatoirement dans chaque itération. D'autre méthodes peuvent également être proposées pour le choix de  $s$ , par exemple permet la meilleur particule d'un sous essaim de migré vers un autre sous essaim.



**Figure 2.10 : La topologie en anneau du schéma de migration utilisé par l'algorithme VEPSO composé de 5 sous essaims**

**b- Ordre lexicographique**

Dans cette méthode, l'utilisateur est invité à ranger les objectifs par ordre d'importance.

La solution optimale est alors obtenue par minimisation des fonctions objectifs séparément, commençant par la plus importante et procédant selon l'ordre d'importance assigné aux objectifs [Miettinen, 1999] [Hu et Eberhat, 2002]. L'ordre lexicographique

tend à être utile seulement quand peu d'objectifs sont employés (deux ou trois), et il peut être sensible à l'ordre choisi des objectifs [Coello, 1999]. Cette approche utilise une OEP avec une topologie circulaire de voisinage. Cet algorithme n'introduit pas d'archive externe de stockage. Cependant, dans [Hu et al., 2003], une version de cet algorithme avec utilisation d'une archive externe est proposée.

Un autre travail consiste à prendre en considération l'ordre d'importance déterminé par le décideur durant le processus d'optimisation. L'approche utilise la structure multi-essaims, la population est composée de l'essaim principal et de plusieurs essaims assistants, chaque objectif est optimisé par un essaim assistant correspondant et tous les objectifs sont optimisés simultanément dans l'essaim principal. Une nouvelle équation de la mise à jour de vitesse est introduite afin de partager l'information entre les essaims assistants et l'essaim principal [Wang et Yang, 2008].

### 2.6.8.3 OEP multi-objectif pour l'approche de Pareto

Ces approches utilisent les méthodes de sélection de "leader" basées sur la dominance au sens de Pareto et un archive externe pour stocker les solutions non dominées. Le principe de base de toutes ces approches consiste à considérer des particules comme étant des leaders, si elles sont non dominées dans l'essaim.

#### a- L'Algorithme de Ray et Liew

Cet algorithme utilise la dominance de Pareto et combine le concept de techniques évolutionnaires avec les essaims particulaires. Cette approche utilise l'estimateur de densité de voisin le plus proche pour maintenir la diversité. L'ensemble de leaders maintenus est sauvegarder dans une archive externe [Ray et Liew, 2002].

#### b- MOOEP basé sur l'utilisation d'un archive externe

Dans cette partie nous présentons une approche basé sur une mémoire externe appelée archive et une approche géographique pour maintenir la diversité au sein de l'essaim.

Dans cette technique la meilleur solution retrouvé par une particule est remplacée par des solutions non dominées générées par cette particule .l'utilisation d'un mécanisme d'attraction global et un archive des vecteurs non dominés précédemment trouvés motive la convergence vers des solutions globales non dominées.

La mise à jour de l'archive est réalisée suivant un critère géographique défini en terme des différents valeurs des fonctions objectives pour chaque individu.

L'archive précédemment mentionné est utilisé pour l'identification d'un leader pour guidé la recherche. L'espace de recherche est divisé en hyper-cubes auxquels on donne des notes, qui sont fonction du nombre de solutions non-dominées situées dans chacun d'eux.

Les leaders sont choisis dans les hyper-cubes sélectionnés à l'aide d'un opérateur de sélection par roulette basé sur leurs notes.

L'algorithme de MOOEP est le suivant:

\*\*\*\*\*

### Algorithme 2.2: Algorithme de OEPMO basé sur un archive externe

\*\*\*\*\*

1.initialisation de la population POP

pour i=0 à N faire

initialise POP[i]

fin pour

2.initialisation de la vitesse de chaque particule:

pour i=0 à N faire

initialise V[i]

fin pour

3.évaluation de chaque particule dans la population POP.

4.Stocker les positions des particules non dominées dans l'archive externe ARCH .

5.diviser l'espace de recherche en hyper-cubes et localiser les particules en utilisant ces hyper-cubes comme un système de coordonnées.

6.initialiser la mémoire de chaque particule:

pour i=0 à N faire

PBEST[i]:=POP[i]

fin pour

**Tant que** le nombre maximum de cycles n'est pas atteint **faire**

- Calculer la vitesse de chaque particule :

$$V[i] = w.V[i] + \rho_1(PBEST[i] - POP[i]) + \rho_2(ARCH[h] - POP[i])$$

- calculer la nouvelle position de chaque particule:

$$POP[i] = POP[i] + V[i]$$

- maintenir les particules dans l'espace de recherche en cas où une particule a été quittée cet espace.
- évaluer chaque particule de la population POP.
- mettre à jour l'archive par l'insertion des nouvelles positions non-dominées et la suppression des positions dominées.
- mettre à jour la mémoire de particule par la nouvelle position si celle-ci est meilleur que la meilleur position courante.

### **Fin Tant que**

\*\*\*\*\*

Dans l'algorithme 2.2:  $w$  représente le coefficient d'inertie,  $\rho_1$   $\rho_2$  sont des variable aléatoire, PBEST est la meilleur position trouvé par une particule, ARCH représente l'archive externe qui stocke les positions non dominées.

$h$  est l'indexe de leader utilisé pour guidé la recherche, cet indexe est sélectionné de la façon suivante: on assigne aux hyper-cubes qui contiennent plus qu'une particule un coefficient égale au résultat de la division des nombres de particules dans ce cube sur un nombre  $> 1$ , ceci est utilisé pour diminuer les coefficients des hyper-cubes qui contiennent plus qu'une particules. Après cette étape, on applique une sélection par roulette basé sur ces coefficients pour choisir le hyper-cube qui contient la particule utilisée comme leader, une fois que l'hyper-cube a été choisi, nous choisissons aléatoirement une particule dans un tel hyper-cube.

### **- Recensement des paramètres**

Dans cette paragraphe nous mentionnons brièvement les différents paramètres utilisés dans l'approche Pareto ainsi quelques directives pour déterminer les meilleurs valeurs pour ces paramètres:

- nombre de particules: il est recommandé de choisir un nombre entre 20 et 80 selon le problème traité.
- nombre d'itérations: ce paramètre est lié au nombre de particules, le rapport est inversement proportionnel .le nombre d'itérations recommandé et entre 80 et 120.

- Nombre de divisions : il permet de déterminer le nombre des hyper-cubes doit être générés dans l'espace de recherche des fonctions objectives. la valeur recommandée est entre 30 et 50 divisions.

La taille d'archive: ce paramètre est employé pour délimité le nombre maximum de vecteurs non dominés qui peuvent être stockés dans l'archive. La valeur de ce paramètre déterminera la qualité du front Pareto généré

### c- Approche AMOPSO (Another Multi-objective Particle Swarm

#### Optimization):

Cette approche utilise : le rang de Pareto et une technique de classification qui permet une subdivision de l'espace de recherche en plusieurs sous-essaims, afin de fournir une meilleure distribution des solutions dans l'espace de recherche. Dans chaque sous-essaim, un algorithme d'OEP est exécuté et, au même temps, les différents sous-essaims échangent l'information [Pulido et Coello, 2004].

## 2.7 OEPMO par Agrégation dynamique DWA basé sur trois techniques d'adaptation du facteur d'inertie

Parsopoulos et Vrahatis [Parsopoulos et al., 2002] montre que la méthode d'agrégation dynamique DWA donne une meilleur performance par rapport aux autres approches, le choix de la bonne méthode d'adaptation du facteur d'inertie peut aussi conduire à une meilleur convergence vers le front de Pareto et maintenir la diversité dans l'essaim durant le processus de recherche.

Dans cette section nous avons concentrés sur l'évaluation des trois techniques d'adaptations du facteur d'inertie pour l'approche MOOEP agrégative dynamique DWA.

Les trois techniques d'adaptations sont:

- Linéaire: 
$$w(t) = 1 - \frac{t}{T}(1 - w_{\min}) \quad (2.16)$$

- Non linéaire: 
$$w(t) = (1 - w_{\min}) \frac{(T - t)^2}{T^2} + w_{\min} \quad (2.17)$$

- Aléatoire: 
$$w = (1 - r_1)r_2 \quad (2.18)$$

où  $t$ ,  $T$ ,  $w_{\min}$ , dénotent respectivement l'itération courante, le nombre total d'itérations et le seuil minimal du facteur d'inertie.  $r_1$  et  $r_2$  sont des variables aléatoires définies dans l'équation 1.2 .

## 2.7.1 Etude expérimentale

### 2.7.1.1 Fonctions de testes:

Plusieurs fonctions tests bi-objectifs ont été utilisées pour valider les performances des différents modèles. Ces fonctions ont plusieurs caractéristiques, qui les rendent idéales Pour tester la capacité de ces approches proposées à identifier le front de Pareto.

Les différentes approches d'adaptation du facteur d'inertie sont testées sur le benchmark de test suivant :

- Fonction F1( un front de Pareto uniforme et convexe ):

$$f_1 = \frac{1}{n} \sum_{i=1}^N x_i^2 \quad f_2 = \frac{1}{n} \sum_{i=1}^N (x_i - 2)^2$$

- Fonction F2( un front de Pareto non uniforme et convexe ):

$$f_1 = x_1 \quad g = 1 + \frac{9}{n-1} \sum_{i=2}^N x_i \quad f_2 = g(1 - \sqrt{f_1/g})$$

- Fonction F3( un front de Pareto concave ):

$$f_1 = x_1 \quad g = 1 + \frac{9}{n-1} \sum_{i=2}^N x_i \quad f_2 = g(1 - (f_1/g)^2)$$

- Fonction F4( un front de Pareto ni purement convexe ni purement concave):

$$f_1 = x_1 \quad g = 1 + \frac{9}{n-1} \sum_{i=2}^N x_i \quad f_2 = g(1 - \sqrt[4]{f_1/g} - (f_1/g)^4)$$

### 2.7.1.2 Résultats numériques et discussion

Nous utilisant comme critères de comparaison

- Le nombre de solutions non dominées.
- Largeur du front de Pareto : La largeur de front LF mesure la largeur du front de Pareto. LF est défini par la formule (2.19) :

$$LF = \sqrt{\sum_{k=1}^{nbf} (\max_{i=1}^{nbs}(f_k(s_i)) - \min_{i=1}^{nbs}(f_k(s_i)))} \quad (2.19)$$

Où nbf est le nombre de fonctions du problème, nbs est le nombre de solutions non dominées. Nous avons effectué 3000 itérations pour chaque problème sur l'espace de recherche  $[0,1]^2$ .

Les différentes valeurs des paramètres de l'algorithme sont:

- La taille d'essaim:20 particules.
- La Taille de voisinage:3.
- $c1=c2=0.5$ .

Les résultats des trois techniques sont exhibés sur les schémas 2.11 à 2.15 et eux sont semblables en terme de détection du front de Pareto à ceux obtenus dans [Parsopoulos et al., 2002].

La table 2.2 représente le nombre des solutions non dominées obtenues pour chaque technique d'adaptation ainsi par combinaison des trois approches. La table 2.3 Représente la largeur du front de Pareto.

Les Table 2.2 et 2.3 montrent que la technique d'adaptation non linéaire présente des résultats compétitifs vis-à-vis aux autre techniques en terme de nombre de solutions trouvées et la largeur du front de Pareto détecté. Chaque technique donne aussi des solutions qui ne sont pas dominées par les autres techniques, ça est illustré par le nombre total des solutions non dominées trouvées par les trois techniques.

Toutes les données numériques observées dans les Tables 2.2 et 2.3 sont confirmées par les impressions visuelles données par les Figures 2.11 à 2.15.

On peut conclu que la combinaison des trois approches d'adaptation maximise le nombre des solutions non dominées et préserve la diversité dans l'essaim mieux que l'utilisation d'une seule technique.

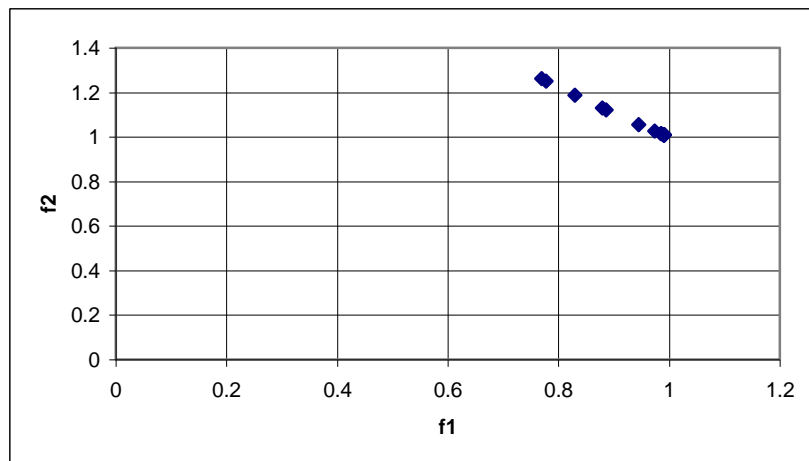


**Table 2.2 Comparaison des différentes techniques basées sur le nombre des solutions non dominées**

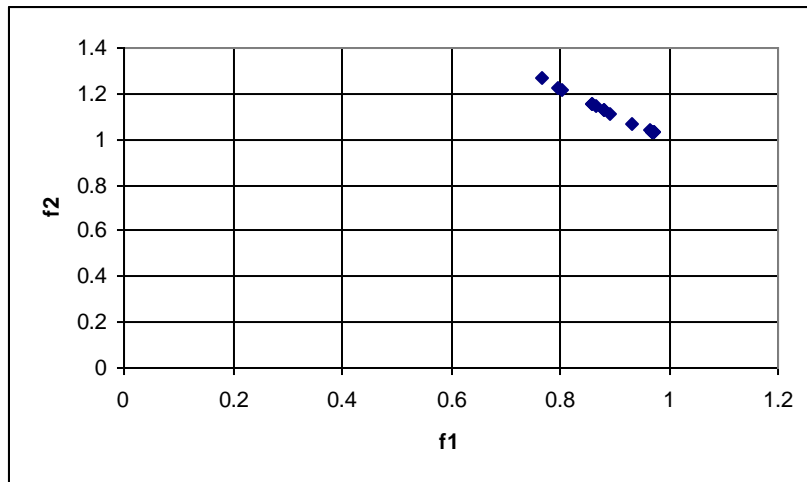
	F1	F2	F3	F4	F5	Moyen
Linéaire	16	6	5	7	3	7.4
Non Linéaire	16	6	7	6	7	8.4
Aléatoire	18	3	8	5	4	7.6
Total des solutions non dominées	48	13	20	17	13	22.2

**Table2.3: Comparaison des différentes techniques basée sur la largeur du front de Pareto**

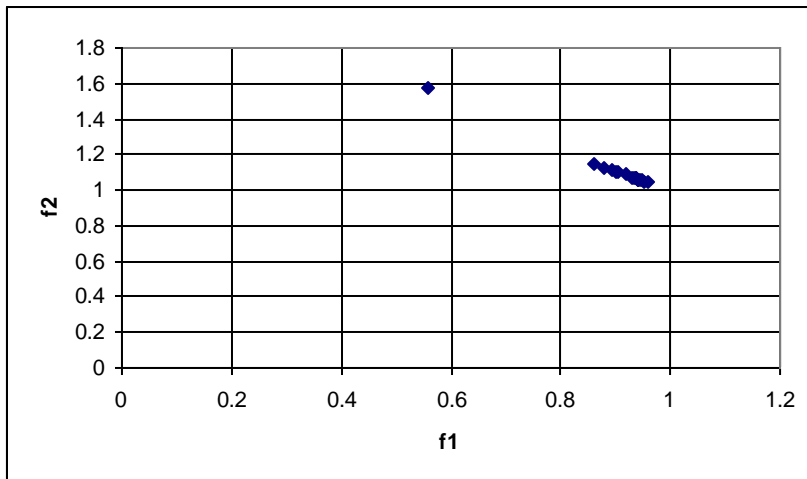
	F1	F2	F3	F4	F5	Moyen
Linéaire	0.68897025	1.16409793	0.8323641	1.40084653	0.98954636	1.01516503
Non Linéaire	0.66542242	1.10141182	1.35453534	1.45432115	1.53909519	1.22295718
Aléatoire	0.96249	0.70654582	1.00327065	0.93134097	1.15939424	0.95260834
Combinaison de trois techniques	0.99721913	1.16409793	1.35817856	1.45432115	1.53909519	1.30258239



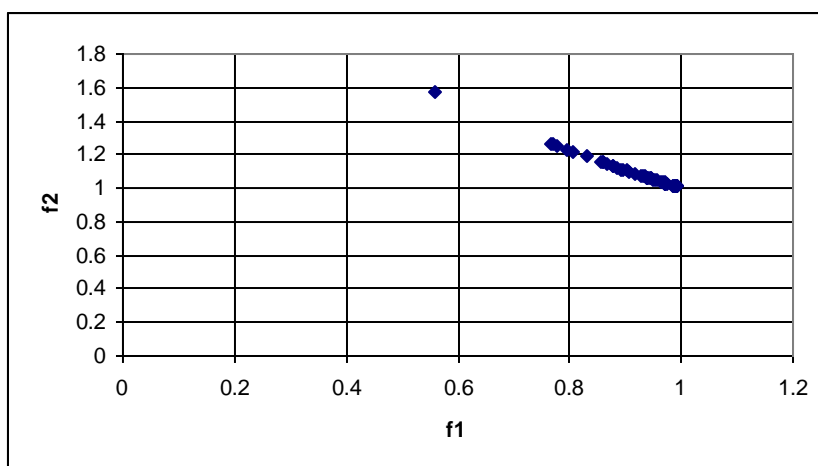
**(a) linéaire**



(b) Non linéaire

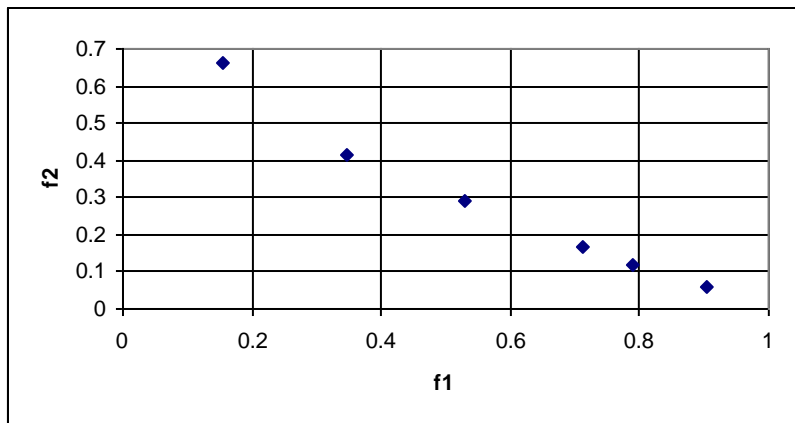


(c) Aléatoire

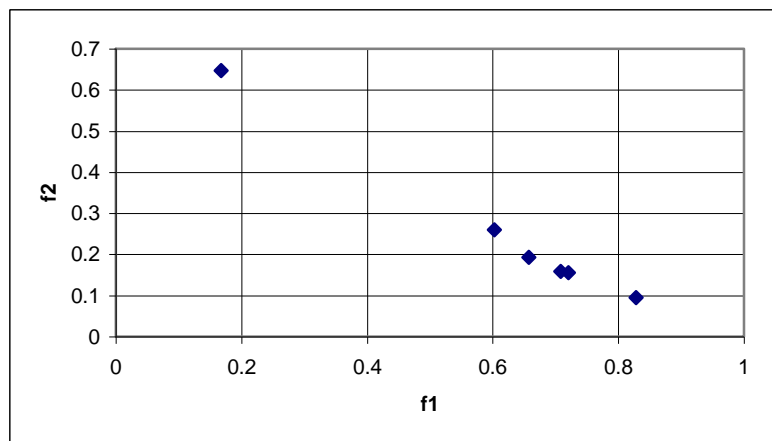


(d) Combinaison de trois approches

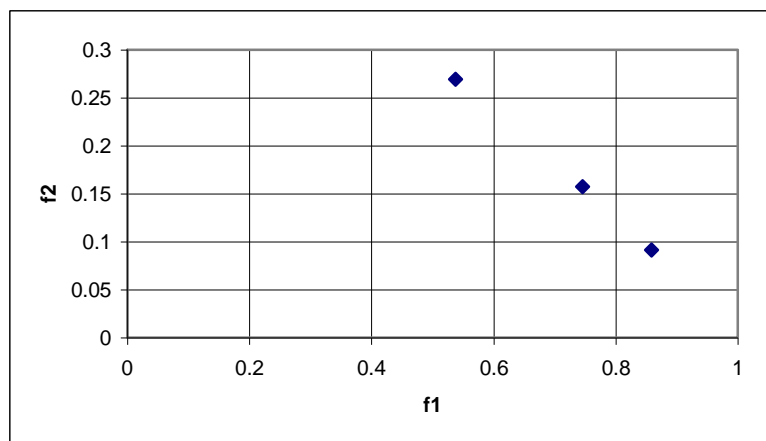
Figure 2.11: Résultats des différentes approches d'adaptation du facteur d'inertie pour F1



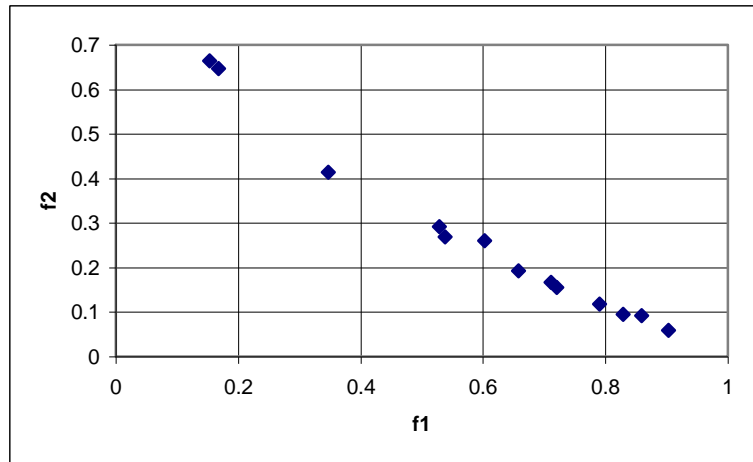
(a) Linéaire



(b) Non linéaire

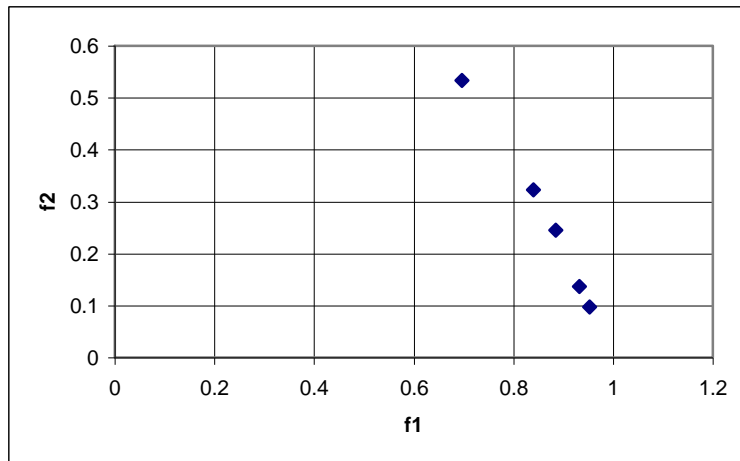


(c) Aléatoire

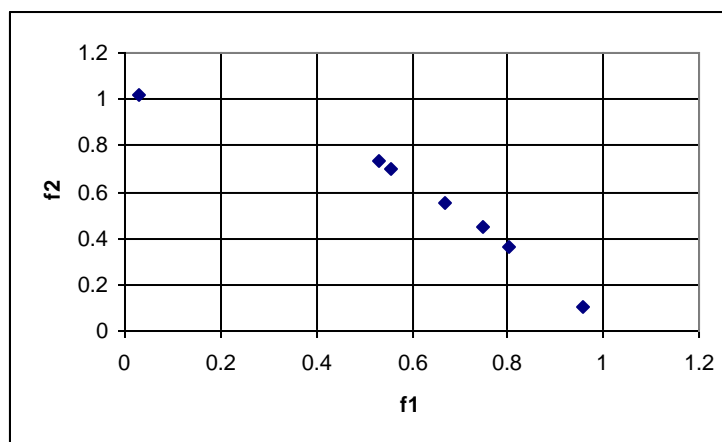


(d) Combinaison de trois approches

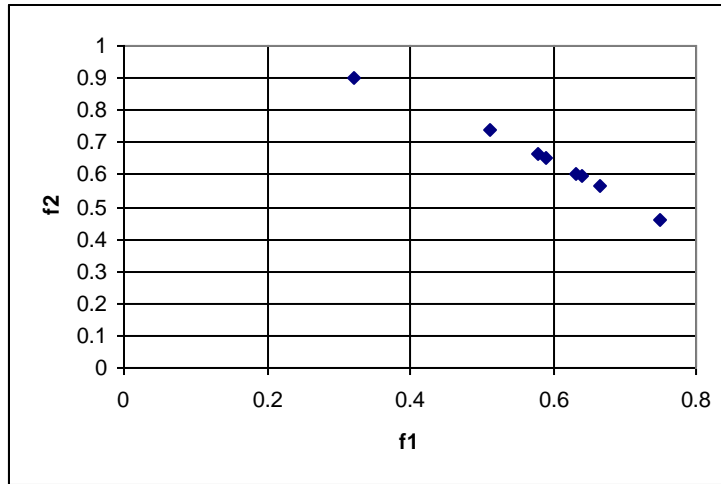
Figure 2.12: Résultats des différentes approches d'adaptation du facteur d'inertie pour F2



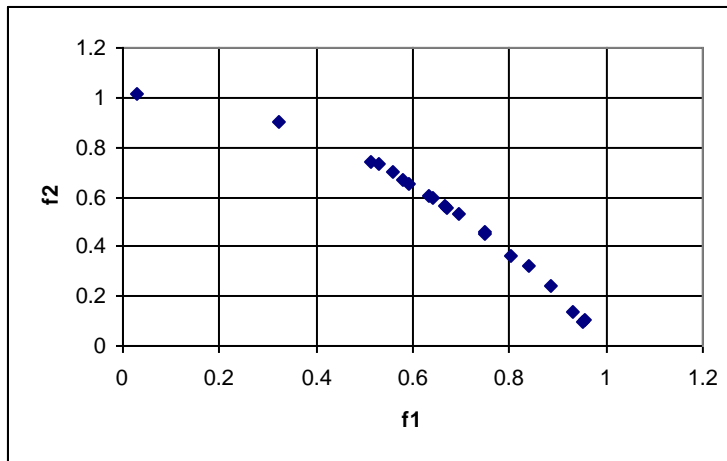
(a) Linéaire



(b) Non linéaire

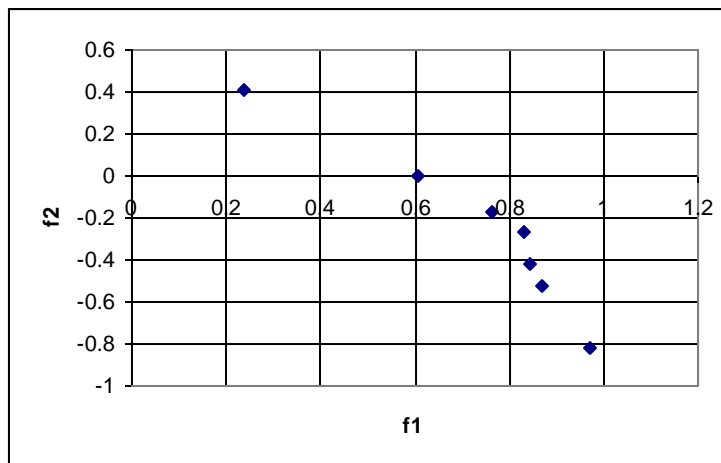


(c) Aléatoire

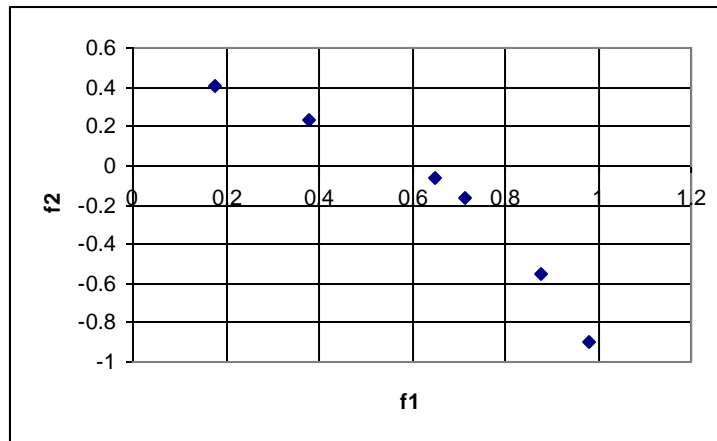


(d) Combinaison de trois approches

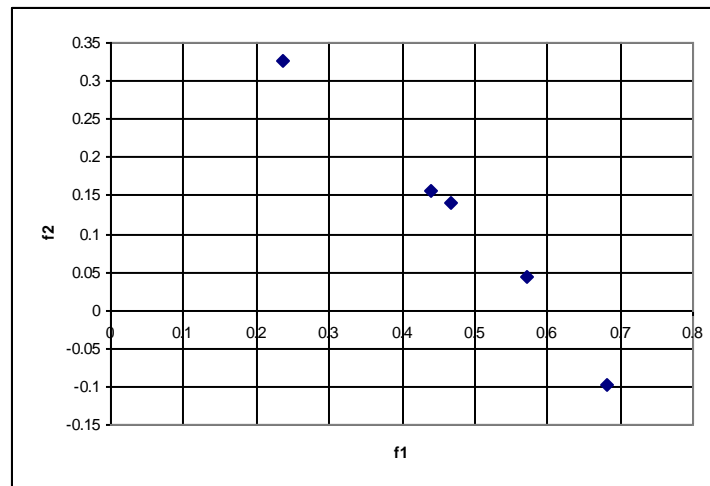
Figure 2.13: Résultats des différentes approches d'adaptation du facteur d'inertie pour F3



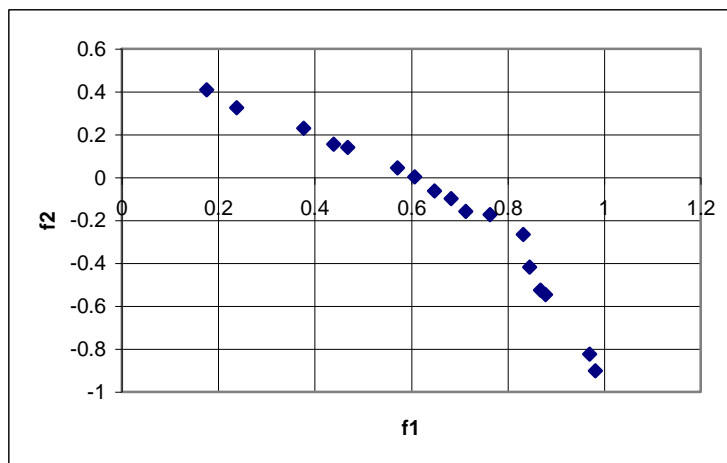
(a) Linéaire



(b) Non linéaire

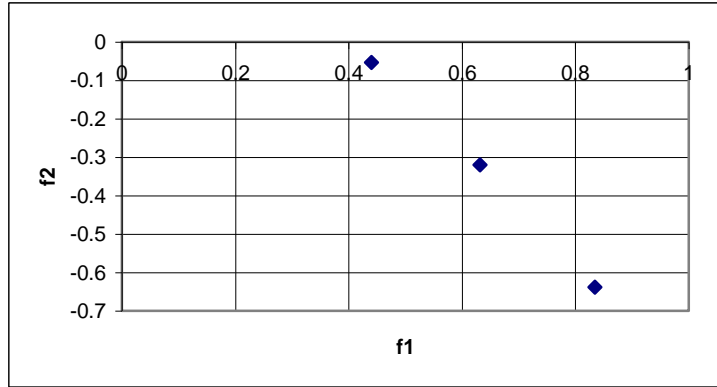


(c) Aléatoire

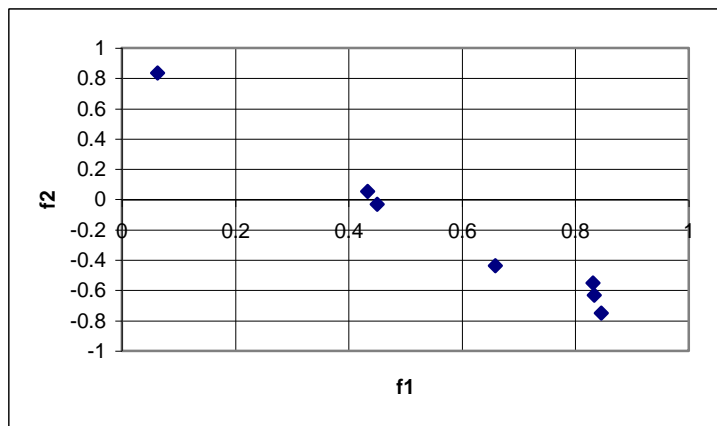


(d) Combinaison de trois approches

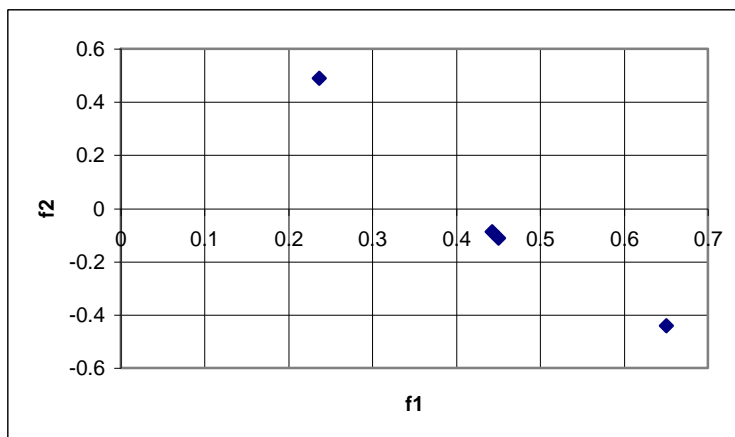
Figure 2.14: Résultats des différentes approches d'adaptation du facteur d'inertie pour F4



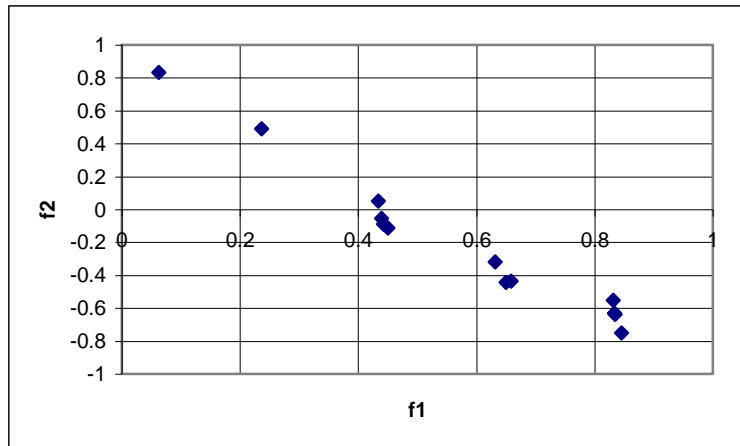
(a) Linéaire



(b) Non linéaire



(c) Aléatoire



(d) Combinaison de trois approches

Figure 2.15: Résultats des différentes approches d'adaptation du facteur d'inertie pour F5

### 2.7.2 Un nouvel algorithme MOOEP agrégatif à base de trois techniques d'adaptation

L'Algorithme 2.3 donne un pseudo-code de la nouvelle approche agrégatif à base de trois méthodes d'adaptation du facteur d'inertie : aléatoire, linéaire et non linéaire. Le nouvel algorithme s'exécute trois fois, dans chaque fois on change la méthode d'adaptation du facteur d'inertie donnée par les équations 2.16 à 2.18 pour obtenir un nombre maximal des solutions non dominées et maintenir la diversité dans l'essaim.

\*\*\*\*\*

#### Algorithme 2.3: Algorithme MOOEP agrégatif à base de trois méthode d'adaptation du facteur d'inertie

\*\*\*\*\*

- 1.initialisation de la population POP
- 2.initialisation de la vitesse de chaque particule:
- 3.évaluation de chaque particule dans la population POP.
- 4.Stocker les positions des particules non dominées dans l'archive externe ARCH .
- 6.initialiser la mémoire de chaque particule:

**Pour**  $i = 1$  à 3 **faire** /\*exécuter l'algorithme trois fois

**Tant que** le nombre maximum d'itération n'est pas atteint **faire**



**Si  $i=1$  alors**

ajuster le facteur d'inertie selon la formule (2.16)

**fin si**

**Si  $i=2$  alors**

ajuster le facteur d'inertie selon la formule (2.17)

**fin si**

**Si  $i=3$  alors**

ajuster le facteur d'inertie selon la formule (2.18)

**fin si**

- Calculer la vitesse de chaque particule selon l'équation
- calculer la nouvelle position de chaque particule selon l'équation
- maintenir les particules dans l'espace de recherche en cas où une particule a été quittée cet espace.
- évaluer chaque particule de la population POP.
- mettre à jour l'archive par l'insertion des nouveaux positions non-dominées et la suppression des positions dominées.
- mettre à jour la mémoire de particule par la nouvelle position si celle-ci est meilleur que la meilleur position courante.

**Fin Tant que**

**Fin Pour**

\*\*\*\*\*

## 2.8 Conclusion

Dans ce chapitre nous avons présentés les différentes notions nécessaires pour décrire un problème multi-objectif, nous avons montrés aussi la difficulté issue de ces problèmes par rapport aux problèmes mono objectifs. Nous nous somme ensuite intéressés aux différentes extensions d'OEP pour manipuler des problèmes multi-objectifs, trois approches sont utilisées par OEP : approche agrégatives, approche non Pareto, approche Pareto.

Nous avons proposés aussi une nouvelle approche d'optimisation multi-objective agrégative par essaim particulaire basé sur trois techniques d'adaptation du facteur d'inertie: aléatoire, linéaire et non linéaire. L'avantage principal de cette approche est qu'elle n'exige pas la définition de la valeur du facteur d'inertie par l'utilisateur, elle permet aussi la détection davantage solutions non dominées que l'utilisation d'une seule technique.

Ce procédure d'adaptation est également implémenté pour maintenir une diversité au sein de l'essaim, et permettant ainsi l'amélioration de la qualité des solutions et le front de Pareto détecté.

**CHAPITRE 3**

**L'OPTIMISATION PAR ESSAIM DES  
PARTICULES ADAPTATIF**

### 3.1 Introduction

Dans l'OEP paramétrique, il n'est pas suffisant de décrire seulement le problème d'optimisation mais aussi il faut indiquer comment on peut faire ça, avec des instructions comme : « commencer par 20 particules », « utilisé un voisinage en anneau »...etc. De ce fait, le traitement d'un problème passe, au préalable, par la définition d'un jeu de paramètres optimal. Or, cette étape, déjà longue et fastidieuse, nécessite une connaissance assez poussée des mécanismes de l'algorithme utilisé. Un utilisateur novice ayant un problème d'optimisation à résoudre risque de ne pas prendre le temps de trouver le jeu de paramètres optimal. Il est donc important d'affranchir l'utilisateur de cette étape en proposant des règles intelligentes qui permettent de déterminer automatiquement les paramètres des algorithmes.

Dans cette section, on présente en détail TRIBES, un algorithme d'optimisation par essaim particulaire adaptatif sans paramètres de contrôle [Clerc, 2003].

### 3.2 Description de TRIBES

TRIBES [Clerc 2003, Clerc 2006] est défini comme une boîte noire, pour laquelle l'utilisateur n'a qu'à spécifier et décrire le problème à résoudre. Proposer une OEP adaptative sans paramètre impose de répondre à deux types de questions : "Comment la structure de l'essaim évolue au cours du temps ?" et "Quel comportement doit adopter une particule ?". La première question est relative à la définition du nombre de particules  $N$  et de la topologie de voisinage. La deuxième question est relative à la définition des paramètres  $w$ ,  $c_1$ ,  $c_2$ ,  $V_{max}$  et la stratégie utilisée pour le déplacement. Deux types d'adaptations sont alors définis : les adaptations structurelles et les adaptations comportementales. Ces deux types d'adaptations sont définis dans les sections suivantes.

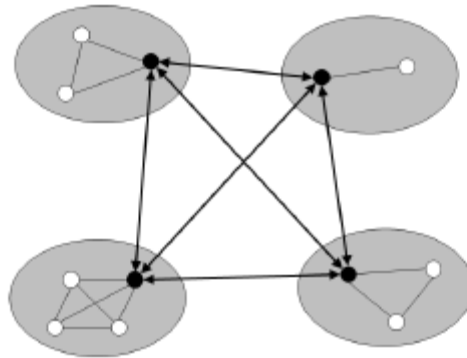
#### 3.2.1 La tribu

Comme nous avons déjà vu dans le chapitre 1, une informatrice d'une particule  $A$  est une autre particule  $B$  pour lequel ça meilleur position peut être lu par  $A$ . Si chaque particule représente un sommet du graphe d'information, on peut représenté la liaison d'information par un arc de  $B$  vers  $A$ . dans la plus part des voisinages déjà présentés et particulièrement le voisinage local, chaque particule a un nombre  $K < N$  d'informatrices, celles-ci permettent la division de l'essaim en sous ensembles de tailles différents, le but est d'apporter une meilleur exploration de l'espace de recherche et d'éviter une convergence prématuré vers un optimum

local résultant de la diffusion totale de l'information à toutes les particules de l'essaim ,de la même manière dans TRIBES, l'essaim est divisé en plusieurs sous-essaims appelés *tribus*. Les tribus sont de tailles différentes, qui évoluent au cours du temps. Le but est d'explorer simultanément plusieurs régions promotrices de l'espace de recherche autour des optima locaux, avant de prendre une décision globale.

Même si chaque tribu parvient à trouver un optimum local, une décision globale de toutes les tribus est nécessaire pour déterminer l'optimum global : les tribus doivent communiquer entre eux .Dans le but de prendre une telle décision, les tribus échangent leurs résultats tout au long du traitement. Deux types de communications sont donc à définir : la communication intra-tribu et la communication inter-tribu. Une telle structure de graphe d'information est similaire à la structure utilisée dans l'OEP multi essaims [Parsopoulos et al., 2004].

Chaque tribu est composée d'un nombre variable de particules. Les relations entre les particules à l'intérieur d'une tribu sont définies par une topologie complètement connectée. Chaque particule connaît les positions pour lesquelles la tribu a trouvé la plus petite et la plus grande valeur de la fonction objectif (communication intra-tribu). Cette structure du graphe d'information offre un bon compromis entre la diffusion et l'exploitation de l'information [WAT 03]. Mais cette structure doit être générée est modifiée par le moyen de la création, l'évolution et la destruction des tribus.



**Figure3.1: Structure du graphe d'information. Communication intra -tribus : les tribus communique entre elle par l'intermédiaire des chamans**

Au fur et à mesure du traitement, chaque tribu va converger vers un optimum local. Il est donc nécessaire que celles-ci communiquent entre elles pour définir lequel de ces optima est à retenir par l'utilisateur. La communication entre les tribus est établie par l'intermédiaire des meilleures particules de chaque tribu (chamans). Ce type de communication est appelé communication inter-tribu.

La figure3.1 montre que chaque particule est informée par elle-même, par toutes les particules de sa tribu et si cette particule est un chaman elle communique avec les chamans des autres tribus.

### 3.2.2 Indicateurs de qualité

Pour des raisons d'adaptation structurelle, on définit deux indicateurs de qualité l'un pour la particule et l'autre pour la tribu.

#### 3.2.2.1 La qualité d'une particule

On dit qu'une particule est *bonne* si elle a amélioré sa meilleure performance au cours de la dernière itération, sinon elle est *neutre*.

Notons que cette définition est qualitative, l'utilisateur est satisfait seulement d'examiner si la qualité est strictement positive ou bien nulle (il n'y a pas d'amélioration).

En plus de cet indicateur, la meilleure et la plus mauvaise position de la tribu sont stockées.

#### 3.2.2.2 La qualité d'une tribu

Pratiquement la qualité d'une tribu est évaluée de la manière suivante :

Soit  $T$  la taille d'une tribu et  $B$  le nombre des meilleures particules dans cette tribu et  $P$  un nombre généré uniformément entre 0 et  $T$ . Si  $B \leq P$  on dit que la tribu est mauvaise, sinon elle est bonne. L'utilisation de ces indicateurs permet de définir les règles de création et de destruction de particules.

### 3.2.3 Evolution des tribus

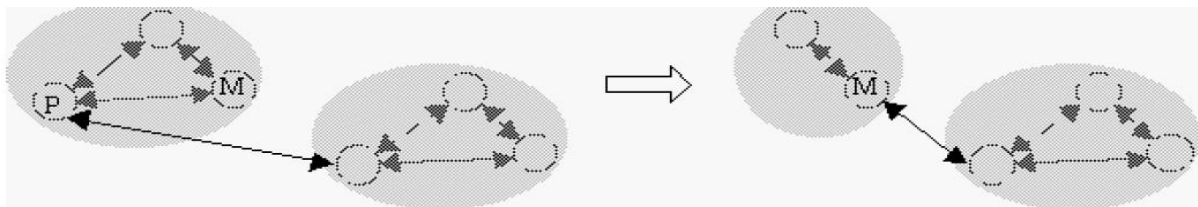
#### 3.2.3.1 La suppression des particules

Dans un souci de gain de temps, il est important d'évaluer la fonction objectif le moins de fois possible. En conséquence Notre intérêt est donc de détruire des particules de l'essaim à chaque fois que cela est possible. Ces destructions sont réalisées en espérant que le résultat final ne soit pas altéré. C'est pourquoi seules les plus mauvaises particules des bonnes tribus sont détruites.

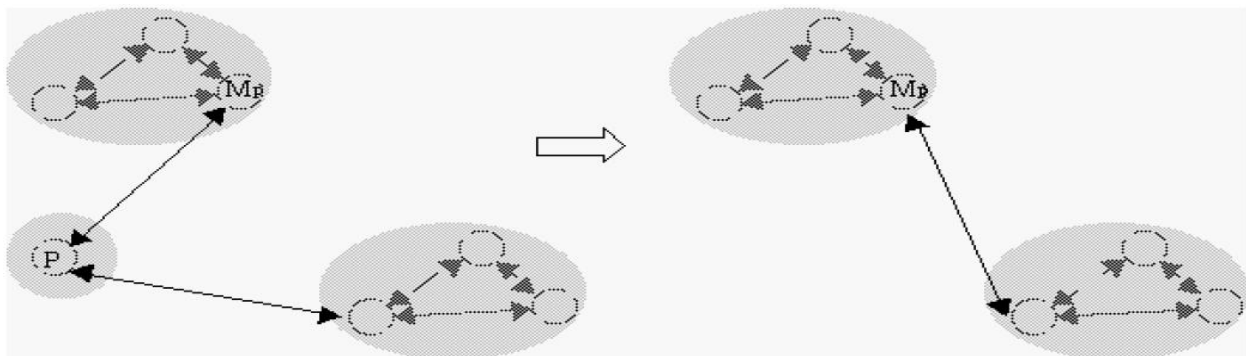
Le but est de trouver l'optimum, si possible à moindre frais, c'est-à-dire en effectuant le moins possible d'évaluations de la fonction. Par conséquent, dès que l'occasion se présente de supprimer à peu près sans risque une particule, il faut la saisir. Notons qu'il vaut mieux conserver à tort une particule (au pire l'on augmentera un peu le nombre d'évaluations stric-

tement nécessaires) qu'en éliminer une à tort (avec le risque de rater complètement la solution). C'est pourquoi seule une bonne tribu pourra éventuellement éliminer une de ses particules et uniquement la pire d'entre elles. Dans le cas d'une tribu mono-particule, l'élimination ne se fera que si un des informateurs a une meilleure performance. En effet, on veut être sûr de conserver au moins une information de meilleure qualité que celle que l'on va éliminer.

Par ailleurs, l'élimination d'une particule implique une redistribution de ses liens d'information, montant et descendant. Dans le cas général, ce report a lieu sur la meilleure particule de la tribu (cf. Figure 1). Dans le cas d'une tribu monoparticule, puisque la suppression de la particule entraîne celle de la tribu tout entière, ils sont reportés sur le meilleur informateur externe de la particule à supprimer (cf. Figure 3.2).



**Figure 3.2(a):** Suppression d'une particule d'une tribu pluriparticule. La particule P est la pire de sa tribu et la tribu a été déclarée « bonne ». Dans ce cas P est supprimé et la redistribution de ses liens externes (ici un seul lien symétrique) se fait sur M, la meilleure particule de la tribu



**Figure 3.2(b):** Suppression d'une tribu monoparticule. La tribu a été déclarée « bonne » et donc la particule unique P, qui est nécessairement la pire de la tribu, même si elle est en même temps « bonne », devrait être supprimée. Mais elle ne le sera effectivement que si son meilleur informateur externe  $M_P$  est meilleur qu'elle. L'hypothèse est en effet que l'information portée par P est alors moins précieuse que celle portée par  $M_P$

**Figure 3.2 : Suppression d'une tribu (mono-particule, pluri-particules)**

#### 3.2.3.2 Génération des particules

Les mauvaises tribus **ont besoins** d'information. Chaque mauvaise particule génère au moins une particule. Le nombre de particules générées par chaque mauvaise tribu est défini

par l'équation **3.1** ci-dessous. Cette équation est inspirée d'une formule identique prouvée pour l'OEP binaire [Clerc, 2005]. Toutes les particules générées par les différentes mauvaises tribus forment une nouvelle tribu.

$$NBparticules = \max\left(2, \frac{9.5 + 0,124(d - 1)}{tribeNB}\right) \quad (3.1)$$

$d$  représente la dimension de l'espace de recherche et  $tribeNB$  le nombre des tribus.

Les mauvaises tribus sont alors reliées à la nouvelle tribu et utilisent la nouvelle information apportée par cette tribu pour améliorer leurs performances. En pratique, le chaman de la nouvelle tribu est un informateur externe des chamans des mauvaises tribus. Deux types de particules sont générés, le type de particule étant sélectionné aléatoirement.

### 3.2.3.3 La particule libres

Pour apporter une meilleure diversité les particules libres sont générées selon les trois cas suivants :

- dans tout l'espace de recherche :  $X_j = U(X_{\min-j}, X_{\max-j}), j \in \{1, \dots, D\}$  (3.2)
- dans la frontière de l'espace de recherche :  $X_j = U(X_{\min-j}, X_{\max-j}), j \in I \subset \{1, \dots, D\}$

$$X_j = U(\{X_{\min-j}, X_{\max-j}\}), j \in J \subset \{1, \dots, D\} \quad (3.3)$$

- dans les sommets de l'espace de recherche:
- $$X_j = U(\{X_{\min-j}, X_{\max-j}\}), j \in I \subset \{1, \dots, D\} \quad (3.4)$$

Où  $U(X_{\min-j}, X_{\max-j})$  est un réel uniformément choisi dans  $[X_{\min-j} ; X_{\max-j}]$  et  $U(\{X_{\min-j}, X_{\max-j}\})$  est un réel uniformément choisi dans la liste  $\{X_{\min-j}, \dots, X_{\max-j}\}$ .  $I$  et  $J$  sont deux sous-espaces qui forment une partition de  $\{1, \dots, D\}$  Ces deux espaces sont définis aléatoirement pour chaque particule générée.

### 3.2.3.4 Les particules confinées

Si  $X$  est la meilleure particule de la tribu génératrice et si  $I_x$  est sa meilleure informatrice,  $P_x$  et  $P_{I_x}$  sont les meilleures performances de  $X$  et de  $I_x$ . La nouvelle particule est générée dans la  $D$ -sphère de centre  $P_{I_x}$  et de rayon  $\|P_x - P_{I_x}\|$  à l'aide de l'équation **3.5** L'objectif est ici d'intensifier la recherche de la tribu génératrice dans des zone promotrice.



$$X_{\text{généré}} = \text{alea\_sphere}(P_{I_x}, \|P_x - P_{I_x}\|). \quad (3.5)$$

Où  $\text{alea\_sphere}(P_{I_x}, \|P_x - P_{I_x}\|)$  est un point uniformément distribué dans le D-sphère de centre  $P_{I_x}$  et du rayon  $\|P_x - P_{I_x}\|$ .

### 3.2.3.5 Fréquence des adaptations

Notons qu'il n'est pas nécessaire, ni souhaitable, de réaliser ces adaptations structurales à chaque itération, car il faut laisser le temps à l'information de se propager entre les particules. Là encore plusieurs règles sont possibles. En théorie, après chaque adaptation, on devrait calculer le diamètre du graphe des relations. Pour cela, il faudrait considérer tous les couples de particules appartenant chacune à deux tribus différentes et trouver le plus court chemin d'information les reliant, en termes de nombre d'arcs. Le plus long des ces plus courts chemins nous donnerait une estimation du nombre d'itérations nécessaires pour être sûr qu'une information possédée par une particule puisse être transmise, plus ou moins directement et plus ou moins déformée, à toutes les autres. Néanmoins, ce calcul est un peu long et l'on pourra se contenter d'utiliser le nombre total de liens d'information. Si, après une adaptation structurale, ce nombre est  $L$ , alors la prochaine aura lieu dans  $L/2$  itérations.  $L$  est calculé à l'aide de:

$$L = \sum_{n=1}^{nbtribus} nb \exp \text{lorer}(n)^2 + nbtribus.(nbtribus - 1) \quad (3.6)$$

Où  $nbtribus$  est le nombre de tribus et  $nbexplorer[n]$  est le nombre de particules de la tribu  $n$ . Le premier terme du membre de droite de l'équation 3.6. Suppose que les particules d'une tribu sont complètement connectées. Donc, pour la tribu  $n$ , le nombre de liens d'information correspondant à la communication intra-tribu est égal à  $nbexplorer[n]^2$ . Le second terme suppose quant à lui que tous les chamans sont reliés entre eux. Chaque chaman est relié à tous les autres, donc il possède  $nbtribus-1$  liens d'informations correspondant à la communication inter-tribu.

### 3.2.3.6 Evolution de l'essaim

Quel est le genre de fonctionnement induit par ces règles ? Au départ, donc, il n'y a qu'une seule particule, représentant une unique tribu. Après la première itération, si sa situation ne s'améliore pas, ce qui est fort probable (et même certain avec les stratégies de déplacement examinées ci-dessous, car elle ne bouge pas du tout au premier pas de temps) une autre particule va être générée, formant une seconde tribu. à l'itération suivante, si aucune des deux

particules n'améliore sa situation, les deux tribus vont générer simultanément chacune une particule : une nouvelle tribu de deux particules sera créée, et ainsi de suite, en notant que plus le nombre de liens augmente, plus le nombre d'itérations entre deux adaptations est important. Ainsi, tant que les choses vont mal, des tribus de plus en plus grandes sont générées, augmentant le pouvoir exploratoire de l'essaim, mais de plus en plus rarement. Entre deux adaptations, l'essaim a donc de plus en plus de chances de trouver une solution.

Mais, inversement, dès qu'une ébauche de solution est trouvée, chaque tribu va progressivement éliminer sa pire particule, éventuellement jusqu'à disparaître complètement. En situation idéale, quand la convergence se confirme, toutes les tribus, sauf éventuellement la dernière créée, se réduisent à une ou deux particules. Globalement, l'essaim a tendance à croître, de plus en plus lentement, de façon asymptotique. Il n'est réellement amené à décroître, au moins temporairement, que dans certains cas simples, quand les tribus sont majoritairement bonnes.

### 3.3 Adaptation structurelle et stratégies de déplacement

Intuitivement, il semble judicieux qu'une particule adopte une stratégie de déplacement fonction de son passé récent. Par exemple, si elle vient d'améliorer deux fois de suite sa position, il n'est sans doute pas nécessaire de la faire trop s'éloigner de la région où elle se trouve et, inversement, aux particules qui évoluent mal de pouvoir diversifier leur recherche.

Il y a trois possibilités d'évolution pour décrire l'historique d'une particule entre deux itérations : amélioration, détérioration ou statu quo, suivant que la particule a amélioré, détérioré ou égalé sa performance précédente. Ces trois états sont notés en utilisant les symboles suivants : + pour amélioration, - pour une détérioration et = pour un statu quo. Le passé récent d'une particule est constitué par ses deux dernières variations. Par exemple, une amélioration suivie d'une détérioration est notée (+ -). Il y a donc neuf possibilités de passés possibles. Cependant, on se contente de les diviser en trois groupes (voir Table 3.1).

(= +) et (+ +) représentent les particules qui ont le meilleur comportement, donc on choisit d'intensifier la recherche dans leur zone à l'aide d'une stratégie locale à gaussiennes indépendantes. (+ =) et (- +) représentent les particules au comportement moyen. Il leur est attribué une stratégie par pivots bruités. Enfin, (- -), (- =), (+ -), (- =) et (= =) représentent les plus mauvaises particules. On leur attribue donc une possibilité de déplacement relativement large, en utilisant une stratégie par pivot.

Notons, par ailleurs, que le confinement de la particule dans l'espace de recherche est réalisé de la même manière qu'en OEP classique, si ce n'est qu'il n'y a pas de vitesse à modi-

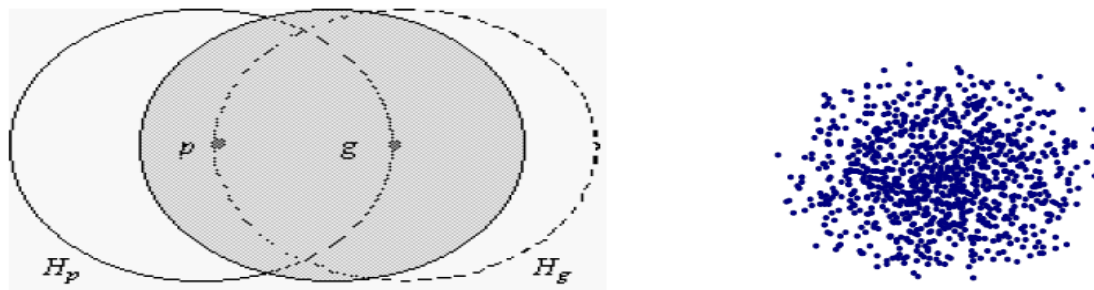
fier. Si une composante de la position tend à sortir des valeurs admissibles, on la ramène donc simplement à celle qui est la plus proche.

#### 3.3.1 Méthode des pivots simple

La méthode des pivots originelle [SER 97], retranscrite dans le vocabulaire de l'OEP, consisterait à n'avoir à chaque itération qu'un nombre pair de particules, à les appairer et, dans chaque couple, à désigner comme pivot la meilleure des deux. Le pivot ne bouge pas et la nouvelle position de l'autre particule est choisie au hasard selon une distribution symétrique centrée sur le pivot, par exemple une gaussienne.

Ici, on procède un peu différemment. Pour chaque particule on considère encore deux points de l'espace de recherche, mais ce sont sa meilleure performance  $p$  et la meilleure performance de ses informateurs,  $g$ . On définit ensuite deux hypersphères,  $H_p$  et  $H_g$ , centrées sur ces points et de même rayon égal à leur distance. Ensuite, dans chaque hypersphère, on choisit un point au hasard selon une distribution uniforme. Un petit programme permettant de le faire est donné en annexe B. Ensuite, on lui affecte un poids fonction décroissante de la valeur de la fonction à minimiser évaluée au centre de l'hypersphère. Enfin la nouvelle position est calculée comme barycentre de ces deux derniers points.

Première remarque, l'affectation d'un poids a un aspect arbitraire a priori assez gênant. Mais, en réalité, la méthode est extrêmement robuste vis-à-vis du choix de la fonction de poids, pourvu que celle-ci respecte quelques conditions très générales : décroissante stricte et valeur finie pour une valeur nulle de la fonction à minimiser (supposée positive, cas auquel on peut, en pratique, toujours se ramener).



**Figure 3.3:** Choix d'une nouvelle position. Le point  $p$  est la meilleure performance de la particule, le point  $g$  la meilleure de celles de ses informateurs. Le rayon commun aux hypersphères  $H_p$  et  $H_g$  est la distance entre ces deux positions. Un point est choisi au hasard uniformément dans  $H_p$ , un autre dans  $H_g$  et les deux combinés linéairement. Le résultat sera d'autant plus proche de  $H_g$  que  $g$  est meilleure que  $p$ . La distribution des nouvelles positions possibles est encore dans une hypersphère, mais n'est plus uniforme, comme on peut le voir sur l'échantillonnage de 1000 positions dans la partie droite de la figure.

Deuxième remarque, il semble que la position actuelle de la particule ne soit pas prise en compte pour calculer la position future, contrairement à la formule de l'OEP classique. Ceci peut sembler curieux, mais, justement, ce n'est qu'une impression. D'une part, si cette position est mauvaise, il n'y a en effet aucun intérêt à l'utiliser, au contraire. D'une certaine manière, on peut donc dire qu'elle est prise en compte précisément en l'ignorant ! D'autre part, si elle est bonne,

**Tableau3.1 stratégie de déplacement**

Passé	Stratégie de déplacement
=+,++	Locale par gaussiennes indépendantes
+,-+	Pivot bruité
--,-,+,-,==	Pivot

Alors en réalité elle coïncide avec sa meilleure performance  $p$  et, de ce fait, elle intervient bien dans le calcul.

### 3.3.2 Méthode des pivots bruitée

Cette seconde stratégie commence exactement comme la précédente. Simplement, une fois déterminée la nouvelle position, on la modifie encore selon un bruitage aléatoire gaussien, de moyenne zéro et d'écart-type d'autant plus faible que la meilleure performance de la particule est proche de celle de ses informateurs. Par exemple, si  $f$  est la fonction à minimiser, on

pourra prendre comme écart-type. 
$$\frac{f(p) - f(g)}{f(p) + f(g)}. \quad (3.7)$$

Ici encore, la formule exacte n'est pas très importante. Il suffit que le résultat soit au plus égal à 1 et strictement décroissant avec la différence des performances  $f(p) - f(g)$ . En pratique, pour chaque composante de la position qui vient d'être calculée, on tire au hasard une valeur  $b$  selon la distribution de bruitage et l'on multiplie la composante par  $(1+b)$ .

### 3.3.3 Locale par gaussiennes indépendantes

Si  $g$  est la meilleure informatrice de la particule, le mouvement est déterminé comme suit:

$$X_d = alea\_normal(g_d - X_d, \|g_d - X_d\|), d \in \{1, \dots, D\} \quad (3.8)$$

où  $alea\_normal(g_d - X_d, \|g_d - X_d\|)$  est un réel choisi aléatoirement selon une distribution gaussienne de moyenne  $g_d - X_d$  et de variance  $\|g_d - X_d\|$ .

## 3.4 Algorithme

L'algorithme de TRIBES est donné par les algorithmes 3.1 et 3.2  $G_i$  est la meilleure informatrice de la particule  $i$ ,  $P_i$  est la meilleure position atteinte par la particule  $i$ .  $n$  est le nombre d'itérations effectuées depuis la dernière adaptation structurelle.

\*\*\*\*\*

### Algorithme 3.1 : Adaptations structurelles

\*\*\*\*\*

test=0

Pour  $i=1$  à nbtribe faire

**Si** tribe[i].status=bad **alors**

**Générer**  $NBparticules = \max(2, \frac{9.5 + 0,124(d-1)}{tribeNB})$  particules

test=1

**Fin Si**

**Si** tribe[i].status=good **alors**

**Supprimer** la plus mauvaise particule de tribe[i]

**Si** tribe[i].size = 1 **alors**

**Rediriger** les liens d'information vers la meilleure particule de la tribu

**sinon**

tribeNb=tribeNb-1

**Rediriger** les liens d'information vers la meilleure informatrice

**Fin Si**

**Fin Si**

**Fin Pour**

**Si** test=1 **alors**

tribeNb= tribeNb+1

Agréger les particules générées à la nouvelle tribu

Etablir un lien entre les différents chamans

**Fin Si**

**Calculer**  $L$

\*\*\*\*\*

### Algorithme 3.2 : Algorithme de TRIBES

\*\*\*\*\*

Initialisation aléatoire de la particule initiale  $X_0$

$P = G_0 = X_0$

**Evaluer** la fonction objectif

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

    Déterminer le statut de chaque particule

    Choisir la stratégie de déplacement

    Déplacement des particules

    Evaluer la fonction objectif

    Mise à jour des  $p_i$  et des  $g_i$

**Si**  $n < L$

    Déterminer la qualité des tribus

    Adaptations structurelles

    Calculer  $L$

**Fin Si**

**Fin Tant que**

\*\*\*\*\*

### 3.5 Amélioration de TRIBES:

TRIBES est un algorithme compétitif qui permet de trouver rapidement des optima locaux (très utile pour l'optimisation dynamique). Cependant les particules ont tendance à rester dans ces optima locaux et ont du mal à en sortir.

Yann Cooren [COOREN, 2008] propose alors des améliorations possibles de TRIBES :

- Insertion d'une phase d'initialisation régulière pour assurer, dès le départ de l'algorithme, une bonne couverture de l'espace de recherche. par les particules. Les particules sont toutes espacées d'une distance fixée.
- Création d'une nouvelle stratégie de déplacement, basée sur une hybridation avec un algorithme à estimation de distribution, pour maintenir la diversité au sein de l'essaim, tout au long du traitement. Une particule a alors une chance de sortir d'un optimum local.

Les résultats sont concluants et apportent une réelle amélioration à l'algorithme initial au niveau des performances.

Puis, il en propose une version multi-objectif, dénommée MO-TRIBES assez performante vis-à-vis de NSGA-II, qui est connu comme le meilleur algorithme d'optimisation multi-objectif, et MOPSO, qui est l'un des meilleurs algorithmes d'OEP multi-objectif.

TRIBES a ensuite été testé sur des applications de segmentation d'images et fournit de bons résultats.

Puis les algorithmes TRIBES ET MO-TRIBES ont été testés sur 2 problèmes d'optimisation de dimensionnement de circuits électroniques. Le premier problème, mono-objectif, a consisté à maximiser le gain d'un amplificateur de courant faible bruit. Le deuxième, multi-objectif, a consisté à simultanément minimiser la résistance d'entrée et maximiser la fréquence de coupure d'un convoyeur de courant de seconde génération. Ces problèmes ont été traités respectivement à l'aide de TRIBES et MO-TRIBES. Les deux problèmes sont correctement résolus.

### 3.6 Conclusion

Il est possible de concevoir des algorithmes d'OEP sous forme de « boîtes noires », l'utilisateur n'ayant à définir que l'espace de recherche, la fonction à minimiser, la précision souhaitée et, par précaution, un nombre maximum d'évaluations.

Le programme TRIBES est un bon exemple de réalisation d'un tel algorithme. Il fonctionne par coopération de tribus de particules. Dans chaque tribu, les liens d'information forment un graphe fortement connexe. Entre tribus, les liens sont plus lâches, mais le graphe total reste toujours connexe. Les stratégies structurelles mises en œuvre automatiquement concernent l'ajout ou la suppression de particules et de leurs liens d'information. Les stratégies de déplacement d'une particule sont fondées sur des distributions de probabilités hypersphériques, bruitées ou non, le choix tenant compte de l'historique à court terme de cette particule.

**CHAPITRE 4**  
**APPLICATION DE LA METHODE D'ESSAIM**  
**PARTICULAIRE AU RECALAGE DES IMAGES**  
**MULTIMODALES**



## 4.1 Introduction

L'imagerie médicale joue désormais un rôle central dans le domaine médical, tant en recherche fondamentale que dans l'aide au diagnostic ou à la décision thérapeutique.

La quantité croissante de données, souvent volumiques, mise à la disposition du praticien, a motivé la conception de nouvelles méthodes automatiques pour l'analyse et l'interprétation des images.

Une des problématiques majeures est de pouvoir analyser et traiter plusieurs images dans un référentiel commun. Ce problème, connu sous le nom de recalage, requiert l'estimation d'une transformation géométrique permettant la superposition spatiale des caractéristiques correspondantes entre les images, il s'agit de déterminer une transformation géométrique caractérisée par un certain nombre de paramètres.

L'estimation des paramètres revient dès lors à résoudre un problème d'optimisation : il s'agit de déterminer les extremas globaux (minimum ou maximum) d'un critère de similarité pour un jeu de paramètres de la transformation. La plus part des critères de similarité ramène à la formulation d'un système d'équations non linéaires qu'il est difficile à résoudre ou bien ne convient pas au système de dérivation d'équation.

Le choix des attributs utilisés pour guider le recalage est crucial. Il est largement conditionné par la nature des images à traiter. Deux approches se distinguent : les méthodes géométriques et les méthodes denses (méthodes iconiques).

Les méthodes géométriques consistent à extraire (manuellement ou automatiquement) des primitives géométriques de l'image (points, courbes ou surfaces) et à les mettre en correspondance, Les approches denses utilisent l'ensemble de l'information portée par les niveaux de gris de l'image, soit directement en comparant les intensités, soit en comparant les valeurs de l'image ayant subi un traitement (par exemple des caractéristiques différentielles de l'image ou sa transformée de Fourier).

Les méthodes denses souffrent de plusieurs inconvénients, parmi ces inconvénients : le coût calculatoire important dû au fait de devoir considérer chacun des pixels ou voxels de l'image, l'autre point négatif concerne le problème d'optimisation correspondant qui est bien plus difficile à résoudre que dans le cas des méthodes géométriques à cause de la présence de nombreux minima locaux. Dans cette section nous avons eu recours à des techniques issues de l'optimisation par les ESSAIM particulières partiellement adaptatives (OEP à facteur

d'inertie adaptatif) pour déterminer ces extrema. Le but est d'affranchir l'utilisateur de définir le facteur d'inertie et de déterminer le meilleur modèle d'adaptation.

## **4.2 Présentation générale du problème de recalage et notation**

Le recalage d'images pourrait se résumer par le déplacement ou la déformation d'une image afin qu'elle ressemble le plus possible à une autre. Des lors, un certain nombre de questions viennent à l'esprit : quelles informations utiliser pour guider le recalage ?

Comment définir la ressemblance entre deux images ? Comment déformer une image ?

Comment trouver la meilleure déformation ? Ces questions mènent aux quatre critères caractérisant une méthode de recalage [Brown, 1992], [Barillot, 1999] :

- les attributs (structures homologues) : ce sont les caractéristiques, extraites des images, qui permettent de guider le recalage. On distingue les attributs extrinsèques (par exemple des marqueurs externes fixés sur le patient) et les attributs intrinsèques (information issue de l'image, comme par exemple les niveaux de gris ou des primitives géométriques extraites).
- le critère de similarité : il définit une certaine distance entre les attributs des images afin de quantifier la notion de ressemblance.
- le type de transformation : il conditionne la manière dont l'image est géométriquement modifiée. Il peut être global ou local et est caractérisé par un certain nombre de degrés de liberté (paramètres de transformation).
- la stratégie d'optimisation : c'est la méthode qui permet de déterminer la meilleure transformation au sens d'un certain critère de similarité dans l'espace de recherche défini par le modèle de déformation.

## **4.3 Principe générale et composant du recalage**

Les composants de recalages et leurs interconnexions sont montrés dans la figure 4.1 Les images Ireca et Iref sont construites grâce aux fonctions Freca et Fref par extraction des attributs des images Ireca et Iref. Ireca est déformée grâce à une transformation  $h$ . Un critère de similarité  $E$  permet de quantifier la ressemblance entre l'image de référence et l'image déformée. La phase d'optimisation consiste à déterminer la transformation optimale qui minimise (maximise)  $E$ .

### 4.3.1 Description formelle

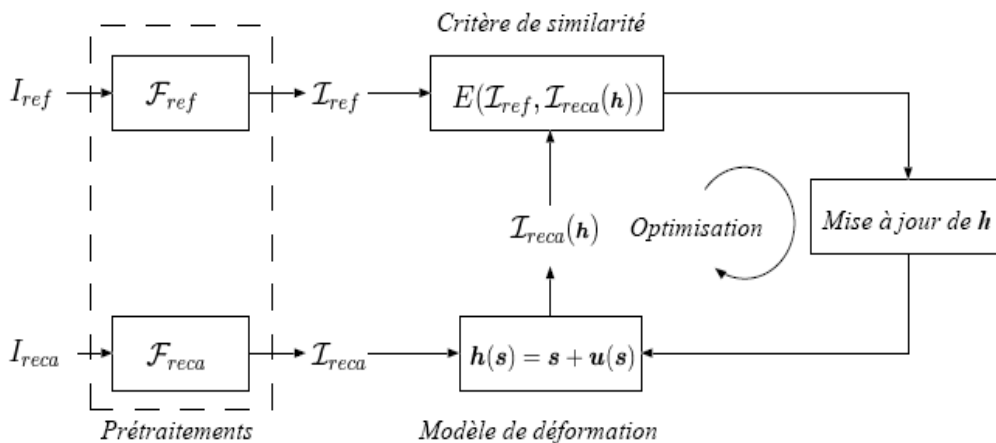
considérons deux images : l'image à recalé  $I_{reca} : \Omega_{reca} \rightarrow \mathbb{R}$  sur une image de référence  $I_{ref} : \Omega_{ref} \rightarrow \mathbb{R}$  tel que :  $\Omega_{reca}, \Omega_{ref}$  sont des sous ensembles de  $\mathbb{R}^d$  ( $d$  représente la dimension de l'image). Soit  $s$  un point de  $\Omega_{reca}$ . le problème de mise en correspondance entre les deux images consiste à l'estimation d'une transformation  $h : \Omega_{reca} \rightarrow \Omega_{ref}$  tel que pour chaque  $s \in \Omega_{reca}$  associé les coordonnées  $h(s)=s+u(s)$  de  $I_{ref}$ ,  $u$  représente le champ de déformation.

Une étape préliminaire à l'estimation de la transformation est l'extraction à partir des images brutes  $I_{reca}$  et  $I_{ref}$  des informations pertinentes (i.e., les attributs) permettant de guider le recalage.  $\mathcal{I}_{reca}$  et  $\mathcal{I}_{ref}$  sont donc construites respectivement à partir de  $I_{reca}$  et  $I_{ref}$  en utilisant les fonctions  $\mathcal{F}_{reca}$  et  $\mathcal{F}_{ref}$ . Ces fonctions peuvent regrouper une chaîne de traitements incluant par exemple l'extraction de primitives géométriques, la suppression d'inhomogénéités de champ, la réduction du bruit, l'application d'une certaine transformation sur les intensités, l'extraction de certaines caractéristiques différentielles de l'image, la normalisation de l'image etc.

Une fois les informations extraites, il s'agit de définir une fonction  $E$  permettant d'associer à un couple  $\mathcal{I}_{reca}(h)$  et  $\mathcal{I}_{ref}$  une valeur permettant de quantifier leur ressemblance. Cette fonction  $E$ , appelée aussi critère de similarité, devrait théoriquement être minimale (ou maximale) lorsque l'image de référence et l'image à recalé sont en parfaite correspondance.

La phase d'optimisation consiste enfin à trouver la transformation optimale  $h$  qui minimise (ou maximise) la fonction  $E$  sur l'espace de recherche  $H$ . Le problème d'optimisation peut ainsi être formulé de la manière suivante :

$$h = \underset{h}{\operatorname{arg\,min}} E(I_{ref}, I_{reca}(h)) \quad (4.1)$$



**Figure 4.1: Les composants principaux d'un processus de recalage des images**

## 4.3.2 La technique d'interpolation

### 4.3.2.1 Passage au domaine discret

Le problème de recalage tel qu'il a été présenté a entièrement été formulé dans le domaine continu. Or, en pratique, la représentation utilise des images est discrète. Les points de la grille de l'image déformée ne correspondent pas nécessairement avec les points de la grille de l'image de référence. Dès lors se pose le problème de l'évaluation des valeurs d'une image pour des coordonnées non entières. Il est donc nécessaire d'avoir recours à des méthodes d'interpolation pour calculer les valeurs des intensités dans les points de la grille de référence. Le choix de la méthode d'interpolation est sujet à un compromis entre qualité et temps de calcul. Deux approches utilisées pour calculer les valeurs des intensités dans la grille de référence après une transformation géométrique :

- **approche directe**: dans cette approche pour chaque point  $p$  de l'image à recalé  $I_{reca}$ , l'image transformé  $I_{trans}$  sera construite à partir la formule suivante:

$$I_{trans}(f(p))=I_{reca}(p) \quad (4.2 a)$$

Où  $f$  est la fonction de la transformation géométrique

- **approche inverse**: contrairement à l'autre approche en utilisant l'inverse de la fonction de la transformation géométrique pour reconstruire l'imagé transformé :

$$I_{trans}(p)=I_{reca}(f^{-1}(p)) \quad (4.2 b)$$

Dès maintenant dans l'espace continue il n'y aucune différence entre les deux méthodes, mais un problème se pose pour la méthode directe dans le cas discret qui nécessite l'interpolation des valeurs des intensités de l'image transformé dans la grille régulière à partir des pixels non entiers (figure 4.2). Tandis qu'en utilisant la transformation inverse, l'interpolation est faite sur les échantillons placés sur la grille régulière de  $I_{reca}$ . En pratique la deuxième approche est plus efficace en terme de cout calculatoire et la qualité de l'interpolation [Noblet, 2006].

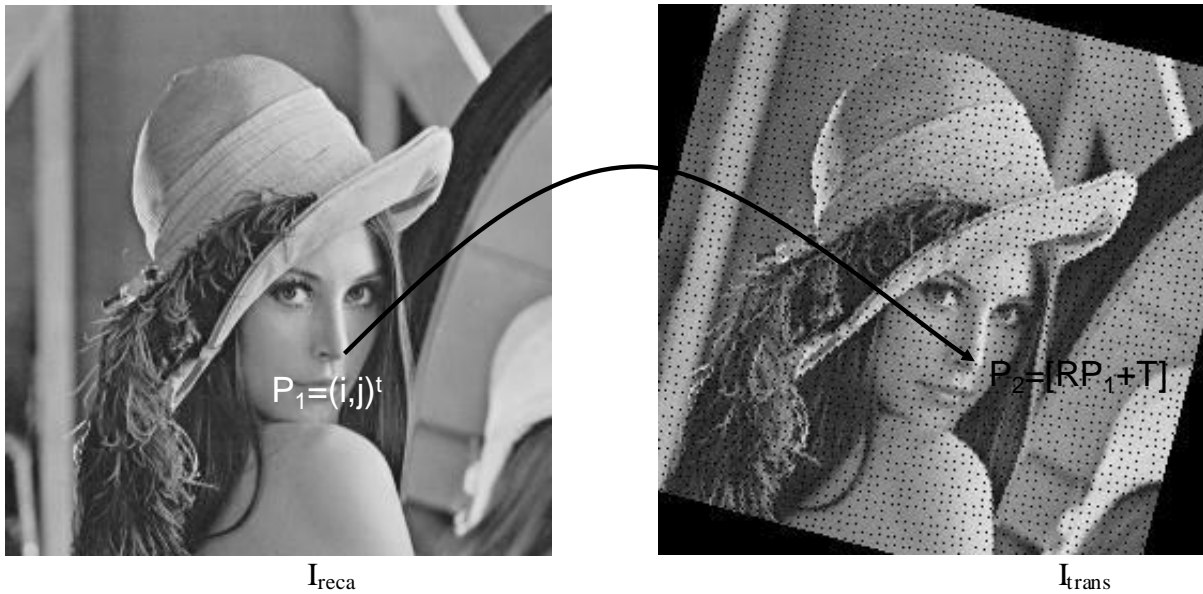


Figure4.2(a) : approche directe d'interpolation pose un problème de qualité d'image construite ( $I_{trans}$ ), les pixels dans la grille régulière sont interpolé à partir des pixels non entiers



Figure4.2(b): approche inverse, on utilisant le transformé inverse, les pixels restaurés par interpolation à partir de la position régulière de la grille

#### 4.3.2.2 Définition de l'interpolation

L'interpolation consiste à calculer les valeurs manquantes (intensité) après une application d'une transformation géométrique sur une image.

Exemple: l'agrandissement d'une image conduit à l'augmentation du nombre de pixels, il est nécessaire de calculer les valeurs manquantes (figure4.3).

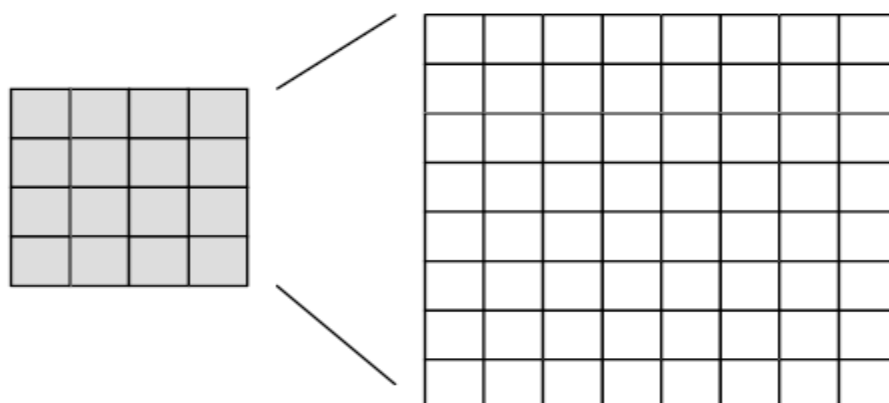


Image source

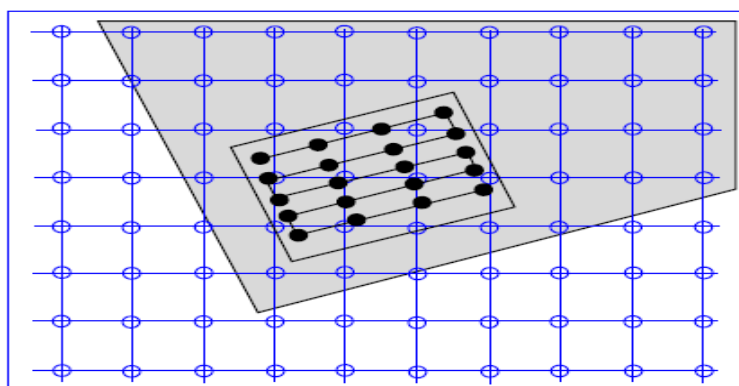
image interpolé

**Figure 4.3 : Exemple d'interpolation d'un facteur 2 dans les deux dimensions**

Plusieurs techniques d'interpolation existent et permettent d'obtenir une image interpolée dont la qualité est souvent croissante avec la complexité de la méthode d'interpolation. Pour un état de l'art des techniques d'interpolation le lecteur pourra se référer à [Lehmann et al, 1999].

#### 4.3.2.3 L'interpolation dans le processus de recalage d'images

Au cours du processus du recalage les valeurs des intensités de l'image de référence et l'image recalée sont comparées typiquement par la fonction de similarité, quand un point de cette image est transformé par certaine fonction de transformation d'un espace à un autre il est ranger dans un point non régulier de la grille de l'image de référence, Par conséquent, l'interpolation est exigée pour évaluer l'intensité d'image transformé à des position du grille de l'image de référence figure 4.4.



**Figure 4.4: Grille des positions de l'image de référence alignées aux positions non régulières de l'image recalée.**

## 4.4 Le type de transformation

Le type de transformation définit la classe des transformations qui régit le recalage entre les deux structures homologues. Elle est de deux types : transformation rigide et transformation flexible.

### 4.4.1 Transformation Rigide

C'est une transformation qui préserve les distances entre les points (isométrie).

La plus simple représentation de la transformation rigide est la translation

$$X' \rightarrow X + b \quad (4.3)$$

où  $b = (\Delta X, \Delta Y)^T$  dans le cas 2D et  $b = (\Delta X, \Delta Y, \Delta Z)^T$  dans le cas 3D.

L'autre type de la transformation rigide est la rotation, cette transformation se définit par :

$$X' \rightarrow A_n X \quad (4.4)$$

A est défini dans le cas 2D par:

$$A = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \quad (4.5)$$

$\theta$  représente l'angle de rotation

et dans le cas 3D :

$$A = \begin{bmatrix} \cos \theta_z & -\sin \theta_z & 0 \\ \sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta_y & 0 & -\sin \theta_y \\ 0 & 1 & 0 \\ \sin \theta_y & 0 & \cos \theta_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ \cos \theta_x & -\sin \theta_x & 0 \\ \sin \theta_x & \cos \theta_x & 0 \end{bmatrix} \quad (4.6)$$

$\theta_x, \theta_y, \theta_z$  sont les angles de rotation autour les axes des coordonnées  $x, y, z$

La transformation rigide générique se compose d'une rotation et une translation.

Ce type transformation est le plus couramment employée dans une situation de recalage multi-modalités (un sujet - plusieurs modalités)

### 4.4.2 Transformation flexible

#### - Transformation affine

La transformation affine autorise, en plus des rotations et des translations, de prendre en compte un facteur d'échelle anisotrope et de modéliser des cisaillements (shear ou skew). Ce type de transformation conserve le parallélisme mais elle ne conserve pas les angles entre les surfaces.

On peut définir la matrice de transformation par:  $A=GSR$

Où  $G$  représente la matrice de cisaillement (shear/warp) :

$$\text{Dans un espace de deux dimensions : } G = G_x G_y = \begin{bmatrix} 1 & g_{xy} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ g_{yx} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.7)$$

Dans un espace de trois dimensions:

$$G = G_x G_y G_z = \begin{bmatrix} 1 & g_{xy} & g_{xz} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ g_{yx} & 1 & g_{yz} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ g_{zx} & g_{zy} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.8)$$

La matrice  $S$  représente la matrice de Zoom :

$$\text{Dans un espace de deux dimensions : } S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.9)$$

$$\text{Dans un espace de trois dimensions: } S = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.10)$$

Le nombre de DDL est de 6 en 2D et de 12 en 3D. Ce modèle est généralement utilisé dans le contexte de l'imagerie médicale pour le recalage inter-individu, bien que son faible nombre de DDL ne lui permette pas d'appréhender avec précision la variabilité anatomique interindividuelle.

### - Transformations projectives

La transformation projective est utilisée principalement pour prendre en compte les effets de perspective dans l'image. Elle ne garantit plus le parallélisme mais impose que l'image d'une droite est une droite. Ce type de transformation est utilisé dans le contexte de l'imagerie médicale pour le recalage d'images 3D sur des images 2D. Ce type de transformation préserve la planéité des surfaces et la rectitude des lignes, par contre elle ne conserve pas le parallélisme.

La transformation projective est définie en fonction de la dimension par:

$$X' = \frac{AX}{P^T X + \alpha} \quad (4.11)$$



**Chapitre 4 : Application de la méthode d'essaim particulière  
au recalage des images multimodales**

---

Où  $P^T = [p_1 \ p_2]$  dans un espace de deux dimensions et  $P^T = [p_1 \ p_2 \ p_3]$  dans espace de trois dimensions

Les constants  $A, p, \alpha$  représentent les paramètres de transformation, le numérateur représente seulement la transformation affine et ses résultats sont divisés par une valeur dépendante de  $X$ .

La transformation projective peut être formulée dans deux étapes:

La première étape représente la transformation linéaire:

$$W = PX, P = \begin{bmatrix} a11 & a12 & a13 & t1 \\ a21 & a22 & a23 & t2 \\ a31 & a32 & a33 & t3 \\ p1 & p2 & p3 & \alpha \end{bmatrix}, X = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (4.12)$$

Suivie par la deuxième étape:

$$X' = g(W) = \begin{bmatrix} W_1 / W_4 \\ W_2 / W_4 \\ W_3 / W_4 \\ 1 \end{bmatrix} \quad (4.13)$$

**- Transformation polynomiale**

La transformation polynomiale peut être considéré come une généralisation de la transformation linéaire. Le vecteur  $X'(x', y', z')$ , dans la version tridimensionnelle, peut être calculé sur la base des composants du vecteur d'entrée  $X$  :

$$X' = \sum_i \sum_k \sum_m \begin{bmatrix} c_x^{ikm} \\ c_y^{ikm} \\ c_z^{ikm} \end{bmatrix} x^i y^k z^m \quad (4.14)$$

Où les constants  $c$  sont les coefficients du polynôme qui représente la transformation.

**4.5 Critères de similarité**

On distingue deux types de critères de similarité utilisées dans le cadre de recalage des images, soit on compare des composants géométriques (points, courbes, surfaces.....) donc on prendre en compte des variations sur les distances euclidiennes pour calculer la fonction de similarité. Soit on utilise des critères qui permettent de comparer toute l'information portée par les niveaux de gris de l'image, on appelle ce type de critères les critères iconiques (denses).

### **4.5.1 Critères de similarité à base des primitives géométriques**

Ce sont des critères qui mesurent la distance entre des primitives géométriques extraites des deux images, on distingue deux catégories :

- les critères qui mesurent la distance entre des points appariés dans les deux images.
- Les critères qui mesurent la distance entre des ensembles de points (courbes, lignes, surfaces).

L'étape d'extraction des primitives géométriques est suivie par une étape d'appariement, une correspondance entre ces informations dans les deux images à recaler est établie dans cette étape.

#### **4.5.1.1 Mesures de similarités à base de points**

Les points sont idéaux pour le recalage d'images car leurs coordonnées peuvent être utilisées directement pour déterminer les paramètres de la fonction de transformation, mais aussi grâce à leurs invariances face à la géométrie des images et leurs facilités de détection par un observateur humain.

Pour mesurer la distance entre les points correspondance, on utilise par exemple la norme euclidienne en minimisant la distance quadratique moyenne :

$$d_T^2(I_1, I_2) = \frac{1}{N} \sum_{i=1}^N \|P_1^i - T(P_2^i)\|^2 \quad (4.15)$$

La distance quadratique moyenne nécessite l'appariement point à point (identifier des paires de points correspondantes).

Plusieurs travaux sont réalisés dans ce cadre, pour plus de détails sur les différents critères à base de points voir [Bookstein, 1989] [Hill et al, 1991] [Rohr et al, 1996] [Ganser et al, 2004].

#### **4.5.1.2 Mesures de similarités à base des ensembles de points**

##### **- les surfaces**

Peuvent être les projections des régions fermées, des réservoirs d'eau, des lacs, bâtiments, forêts, terrains urbains ou des ombres. Elles sont souvent représentées par leurs centres de gravité, qui sont invariants quant à la rotation, dilatation et la déviation, et stables sous un bruit aléatoire et la variation du niveau de gris. Elles sont détectées au moyen des méthodes de segmentation, donc la précision de la segmentation peut influencer significativement le résultat du recalage.

**- les courbes**

Représentations de segments de lignes généraux, contours d'objets, frontières des régions, littoraux, routes, rivières, ou bien des structures anatomiques allongées (dans l'imagerie médicale).

Pour leur détection, des méthodes de détection de bordures standards, comme le détecteur de Canny [Canny, 1986], ou un détecteur basé sur le Laplacienne, sont utilisées [Marr, 1980]. Les lignes sont souvent représentées par des paires de points d'extrémités, ou bien par leurs points de milieu.

### **4.5.2 Critères de similarités iconiques**

Ces critères ne nécessitent pas l'extraction des primitives géométriques mais ils utilisent l'information globale portée par les deux images, l'utilisateur fait une hypothèse sur la nature de dépendance entre les informations portées par les deux images [Althof, 1997], dans ce type il faut différencier entre deux cas : des images issues de même source (même sujet et même dispositif d'acquisition) et des images issues de différentes sources (des images multimodales, images inter-sujets, sujets atlas).

Lorsque les images sont issues de la même source, la relation entre les intensités des deux images est une relation linéaire simple. Par contre dans l'autre cas où les images ne sont pas de la même source, on ne peut pas faire une hypothèse sur la nature de la relation, dans ce cas on utilise des critères issus de la théorie de l'information pour recalibrer les deux images.

Dans ce qui suit, nous allons citer les mesures de similarité les plus fréquentes :

Le choix d'un critère repose sur des hypothèses de dépendance entre les niveaux de gris des images on distingue : la relation d'identité, la relation linéaire, la relation fonctionnelle et la relation statistique :

#### **4.5.2.1 Relation d'identité**

Dans cette catégorie on fait l'hypothèse que les niveaux de gris sont identiques dans les deux images. Le plus simple critère utilisé dans ce cas est le moindre carré :

$$E(I_{ref}, I_{reca}) = \sum_{x \in D} (I_{ref}(x) - I_{reca}(x))^2 \quad (4.16)$$

Cette métrique se fonde sur l'hypothèse que l'intensité qui représente le même point homologue doit être même dans les deux images. Par conséquent, son utilisation est limitée aux images de la même modalité.

#### 4.5.2.2 Relation linéaire

En pratique, l'hypothèse de relation d'identité est rarement vérifiée, même dans le cas monomodal. Il est nécessaire de prendre en compte une remise à l'échelle des intensités grâce à une relation linéaire du type:  $I_0 = aI_1 + b$ . Le coefficient de corrélation linéaire permet prendre en charge cette relation:

$$E(I_{ref}, I_{reca}) = \sum_{x \in D} \frac{(I_{ref}(x) - \mu(I_{ref}))(I_{reca}(x) - \mu(I_{reca}))}{\sigma(I_{ref})\sigma(I_{reca})} \quad (4.17)$$

#### 4.5.2.3 Relation fonctionnelle

Malgré que la relation linéaire soit efficace dans le cas monomodal, elle n'est pas adaptée pour le recalage des images multimodales, dans ce cas on fait l'hypothèse qu'il y a une relation de dépendance fonctionnelle entre les intensités des deux images de types:  $I_{ref} = f(I_{reca})$ .

Le critère de woods est l'un des critères qui permet une certaine transition aux critères de similarités multimodales, le rapport de variance d'intensité(VIR) est utilisé dans ce cas pour calculer le critère de woods de la manière suivante:

Soit deux images  $I_{ref}$ ,  $I_{reca}$  composé de N pixels, VIR est un vecteur basé sur le rapport des différents valeurs des pixels des deux images, le  $i^{eme}$  élément du VIR des deux images est calculé par:

$$R_i = VIR_{(I_{ref}/I_{reca})}(i) = \frac{I_{ref}(i)}{I_{reca}(i)} \quad i=1 \dots N \quad (4.18)$$

Où  $I_{ref}(i)$ ,  $I_{reca}(i)$  représentant respectivement le  $i^{eme}$  pixel et N et le nombre de pixels des deux images. Donc le critère de similarité qui évalue la variance relative entre les éléments de ce vecteur est donné par l'expression suivante:

$$E(I_{ref}, I_{reca}) = \frac{\sqrt{\sum_i (R_i - \bar{R})^2}}{\bar{R}} \quad \text{Et} \quad \bar{R} = \frac{1}{N} \sum_i R_i \quad (4.19)$$

#### **4.5.2.4 Relation de dépendance statistique**

Récemment, plusieurs critères issues de la théorie d'information et l'analyse des données ont été proposées pour résoudre le problème du recalage des images multimodales lorsque les intensités se différencient considérablement entre les modalités des images et particulièrement dans certains cas où une ou plusieurs intensités dans l'une des deux images comparées correspondent à plusieurs intensités dans l'autre image se différencient considérablement. L'un de ces schémas utilise la notion d'histogramme conjoint, les critères fondés sur ce schéma sont basés sur la mesure de la dispersion des échantillons de l'histogramme avec pour idée que plus la dispersion est faible, plus forte est la dépendance entre les deux images (et donc meilleur est le recalage).

Le deuxième schéma considère la distance entre la distribution conjointe des images et la distribution conjointe qu'auraient celles-ci sous l'hypothèse d'Indépendance. L'un de ces critères est inspiré de l'entropie de Shannon [Cover et Thomas, 1991], développé par S Collignon [Collignon et al, 1995] l'entropie jointe des deux images comparées mesure la dépendance entre celles-ci, avec l'hypothèse que les deux images sont dépendantes si l'entropie jointe est minimale. L'autre critère qui a connu un grand succès pour le recalage des images multimodales est l'information mutuelle l'information mutuelle [Viola et Wells, 1995] [Collignon et al, 1995] évalue la corrélation statistique entre différents jeux de données.

Une interprétation intuitive possible de ce critère est de considérer qu'il mesure l'aptitude d'une image à expliquer correctement l'autre. Le cas le plus défavorable est celui de deux images indépendantes (information mutuelle nulle) et le cas idéal est celui pour lequel les deux images sont parfaitement recalées (information mutuelle maximum) ce critère est traité en détail dans la section suivante.

### **4.6 Méthodes d'optimisation**

Beaucoup de méthodes ont été proposées pour résoudre le problème de recalage d'images [Zitova, 2003][Brown, 1992]. Si on considère comme critère l'espace des primitives, on peut classer les techniques de recalage d'images en deux grandes catégories. Les approches iconiques basées sur l'intensité et les approches basées sur les primitives.

**Approches iconiques** : Ces approches ne nécessitent aucune segmentation des images. dans ce cas l'espace des primitives est réduit à l'ensemble des pixels de l'image. Une méthode de recalage basée sur cette approche trouve la transformation optimale en cherchant directement les paramètres de la transformation qui optimise une mesure de similarité. Cette technique est

souvent utilisée quant il s'agit de recalage d'images provenant de systèmes d'imageries différents (images multimodales). Parmi les approches iconiques existantes, on trouve :

- Les techniques de descente du gradient [Sharma, 1997],
- Le recuit simulé [Nikou, 1999],
- L'approche multi-résolution [Jenkinson, 2001].

**Approches basées primitives** : Ces approches nécessitent un procédé d'extraction de primitives. Elles tentent de mettre en correspondance ces primitives et déduire la transformation correspondante. Les principales difficultés de cette méthode résident dans le type des primitives, le bruit et les défauts liés à l'extraction des primitives. La façon de coupler la mise en correspondance des primitives et l'estimation de la transformation, nous amène à distinguer :

- Le recalage sans feedback : Dans ce cas, on établit d'abord les correspondances puis on estime la transformation.
- Le recalage avec feedback : Dans ce cas, les processus de mise en correspondance et l'estimation de la transformation sont intégrés dans un processus itératif. On utilise alors une fonction coût qu'on doit minimiser pour assurer la convergence du processus.

## **4.7 Utilisation d'OEP dans le recalage des images multimodales**

Plusieurs travaux sont réalisés dans le cadre d'application d'OEP sur le recalage des images multimodales [Wachowiak et al, 2004] [Li et Sato, 2007] [Bhattacharya1 et Das, 2009] nous constatons selon ces travaux que la méthode d'OEP peut fournir de bons résultats. Dans cette section nous présentons l'un des deux meilleurs travaux réalisés dans le cadre d'utilisation d'OEP dans le recalage des images médicales multimodales. Les deux approches sont basées sur l'orientation initiale des paramètres de recalage.

### **4.7.1 Une approche OEP avec une orientation initiale de l'utilisateur**

Un travail très important [Wachowiak et al, 2004] présente une nouvelle approche pour le recalage des images biomédicales par la méthode OEP, cette approche d'optimisation par l'essaim particulier, a été adaptée pour le recalage des images biomédicales. Les auteurs proposent une nouvelle technique d'essaim de particule qui intègre une orientation initiale de l'utilisateur. L'idée est que le succès du recalage dépend fortement de l'orientation initiale avant que n'importe quelle optimisation soit appliquée. La technique proposée de l'OEP assume cette orientation comme une proximité de la transformation correcte.

Dans la littérature d'OEP (et en littérature d'optimisation en général), des fonctions standard de test, sont utilisées comme un benchmark de test.

Pour ces fonctions, l'initialisation des particules a peu d'importance. Cependant, dans beaucoup d'applications, il y a habituellement quelques connaissances des caractéristiques de la position de l'optimum global. Les utilisateurs des systèmes biomédicaux de recalage d'images sont généralement des professionnels cliniques qu'ils ont une expérience et des indications sur l'orientation correcte du recalage. Ces utilisateurs peuvent choisir une transformation initiale précise avant que le processus de recalage commence, Cependant, si la position "Xinit" qui représente l'orientation initiale est bien placé, les particules de l'essaim peut avoir une probabilité plus élevée de découvrir une région qui contient l'optimum global .

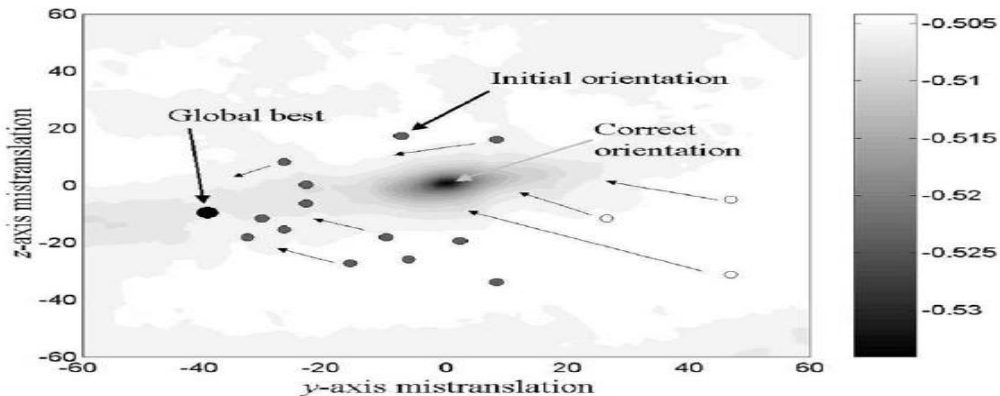
D'une autre manière, dans cette nouvelle approche l'auteur utilise le modèle OEP à facteur de construction déjà présenté dans la formule 1.10 et ajoute à l'équation de mouvement un terme qui représente l'orientation initiale de l'utilisateur finale, l'équation de mouvement a été redéfinie par l'expression suivante :

$$V_i(t+1) = \chi(V_i(t) + \rho_1(Xbest_i(t) - X_i(t)) + \rho_2(XGbest_i(t) - X_i(t)) + \rho_3(X_{init} - X_i(t))) \quad (4.20)$$

Où :

$$\chi = \frac{2k}{\left| 2 - \rho - \sqrt{\rho^2 - 4\rho} \right|} \quad \text{Et } \rho = \rho_1 + \rho_2 + \rho_3 > 4 \text{ et } k \in [0,1]$$

La technique est illustrée dans figure 4.10, cette figure montre un plot de contour de l'information mutuelle pour deux images de différents modalités. La figure montre que la meilleur position globale, réellement représente un minimum local, exerce une attraction sur les autres particules, mais l'effet de la position d'orientation initiale dirige certaines particules dans la direction de l'optimum global.



**Figure 4.5: un plot de contour illustre l'effet de l'orientation initiale sur les particules de l'essaim**

#### 4.7.2 Une approche OEP à base d'une représentation multi-résolutions de l'image

La théorie fondamentale de formation d'image multi-résolutions est d'étudier les images à plusieurs résolutions. Une structure puissante mais simple pour représenter les images à plus d'une résolution est la pyramide d'image . La base de la pyramide contient la représentation de résolution la plus élevée de l'image, l'apex contient une plus basse approximation de résolution (figure).

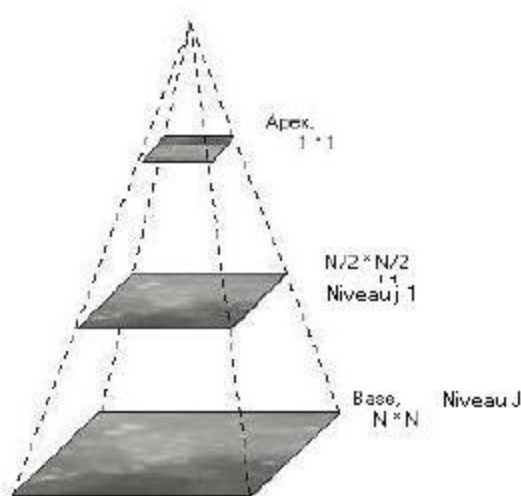


Figure 4.6:structure pyramidale d'une image

Au niveau algorithmique Mahua Bhattacharya et Arpita Das [Bhattacharya1 et Das, 2009] proposent un algorithme OEP avec une stratégie multi-résolution. Elle consiste à utiliser la solution obtenue à une résolution basse comme initialisation à la résolution plus fine suivante. Cela permet d'un côté une réduction drastique du coût global de calcul et d'un autre côté une amélioration de la robustesse de l'algorithme utilisé.

### 4.8 Une approche OEP à facteur d'inertie adaptatif pour le recalage des images IRM multimodales

#### 4.8.1 Problématique

Pratiquement, dans certain cas où le contraste se diffère considérablement entre les images comparées l'utilisation des métriques basées sur la comparaison directe des niveaux de gris sont inapplicables, donc Il est nécessaire de considérer une relation encore moins res-



trictive que la relation directe en introduisant la notion de dépendance, dans cette section nous présentons un critère de similarité qui connu un grand succès pour le recalage d'images médicales multimodales, ce critère s'appelle l'information mutuelle. nous avons réalisés aussi une étude comparative entre trois approches d'adaptation du facteur d'inertie dans le cadre de l'utilisation d'OEP pour maximiser l'information mutuelle des images IRM cérébrales multimodales. le but est de déterminer la meilleur approche.

Laissez nous avant tout présenter quelque notion préliminaire pour la définition de l'information mutuelle.

## 4.8.2 Histogramme d'une image

### 4.8.2.1 Histogramme marginal

Un histogramme marginal d'une image A représentée par q niveaux de gris est le vecteur  $h^A$  tel que le l<sup>ème</sup> élément  $h_l^A$  est calculé par l'expression suivante:

$$h_l^A = \sum_{A_i} 1 \quad \text{E} \{0, 1, \dots, q-1 \mid A_i = \{[i, k] : ((a_{ik} \in A) = l)\} \} \quad (4.21)$$

Où  $h_l^A$  est le nombre des pixels qui ont la valeur l

Pratiquement on utilise souvent l'histogramme normalisé qui est défini par :

$$P_A(l) = NH_l^A = \frac{h_l^A}{N} \quad N \text{ est le nombre de pixel dans A.} \quad (4.22)$$

$NH_l^A$  Représente donc la proportion de pixels de A qui ont la valeur l.

Donc l'image est une réalisation d'un domaine homogène, ergodique et stochastique.

On aussi utilise enfin l'histogramme normalisé cumulé :

$$F_A(l) = CNH_l^A = \sum_{k=0}^l NH_k^A = \frac{1}{N} \sum_{k=0}^l h_k^A \quad (4.23)$$

$CNH_l^A$  représente la proportion des pixels de A qui ont une valeur inférieure ou égale à l.

L'histogramme dans ce cas est une observation statistique, mais il est parfois utilisé comme modèle probabiliste empirique, l'histogramme normalisé correspond à une densité de probabilité et l'histogramme cumulé normalisé correspond à une fonction de répartition.

### 4.8.2.2 Entropie d'une image

Selon la théorie d'information [Cover et Thomas, 1991 ] l'entropie d'une image A est une fonction qui évalue la quantité moyenne de l'information donné par cette image. L'entropie  $E(A)$  se calcul à partir l'histogramme normalisé par :

$$E(A) = -\sum_{k=0}^q P_A(k) \log_2 P_A(k) \quad (4.24)$$

### 4.8.2.3 Histogramme conjoint

D'un point de vue probabiliste, l'histogramme conjoint représente la distribution de probabilité conjointe des niveaux de gris de deux images.

Concrètement, l'histogramme conjoint est un graphe tridimensionnel (de  $R^2$  dans  $R$ ). Chaque point de l'histogramme représente le nombre de fois qu'une combinaison de niveau de gris entre les deux images est rencontrée.

L'histogramme conjoint pour deux images A, B est la matrice  $h^{AB}$  d'une taille  $q \times r$ , où  $q$  et  $r$  sont respectivement le nombre des niveaux de gris dans A et B.

Les éléments de la matrice sont les nombres des pixels de même positions dans A et B qu'ils ont une combinaison particulières des intensités, d'une autre manière le  $(l, m)$  composant de l'histogramme conjoint contient le nombre des différents positions  $x(i, k)$  tel que les deux pixels positionnés à  $x$  dans A et B ont respectivement les intensités  $l$  et  $k$  (figure 4.7). L'histogramme conjoint est calculé par:

$$h^{AB} : h_{l,m}^{AB} = \sum_{D_{l,m}} 1 \quad l=0,1,\dots,q-1 \quad m=0,1,\dots,r-1$$

$$D_{l,m} = \{[i, k] : ((a_{i,k} \in A) = l) \wedge ((b_{i,k} \in B) = m)\} \quad (4.25)$$

L'histogramme conjoint offre une information globale sur la relation entre les intensités des deux images A, B.

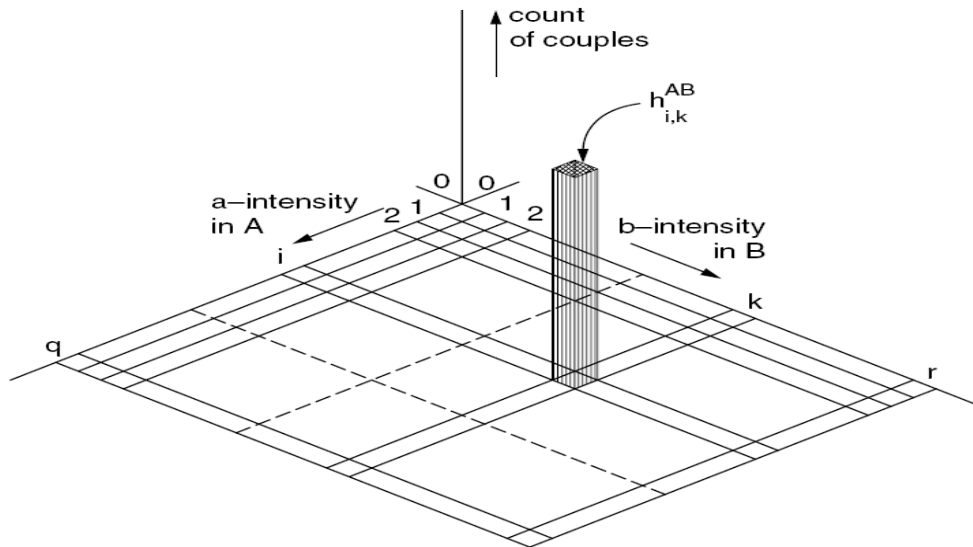
On peut aussi calculer la densité de probabilité des deux images A et B par :

$$P_{AB}(l, m) = \frac{h_{l,m}^{AB}}{N} \quad (4.26)$$

### 4.8.2.4 Entropie conjointe de deux images

L'entropie  $E(A, B)$  se calcul à partir l'histogramme conjoint normalisé par :

$$E(A, B) = -\sum_{l=0}^q \sum_{m=0}^{r-1} P_{AB}(l, m) \log_2 P_{AB}(l, m) \quad (4.27)$$



**Figure 4.7 : Schéma d'un histogramme conjoint de deux images A et B (seule une valeur de l'histogramme est illustré parmi toutes les autres valeurs  $r \times q$ )**

### 4.8.3 Information mutuelle

L'information mutuelle est une mesure qui évalue la corrélation entre deux images est définie en fonction de l'entropie des différentes images  $E(A)$ ,  $E(B)$  et l'entropie conjoint  $E(A, B)$ .

On pratique nous constatant que lorsque les deux images A, B sont différentes :

$$P_{AB}(l,m) = P_A(l)P_B(m) \quad (4.28)$$

$$\text{Et } E(A,B) = E(A) + E(B) \quad (4.29)$$

Mais s'il y a une certaine dépendance entre A et B alors :

$$E(A,B) < E(A) + E(B) \quad (4.30)$$

La différence est appelée l'information mutuelle :

$$I(A,B) = E(A) + E(B) - E(A,B) \quad (4.31)$$

L'information mutuelle évalue la corrélation statistique entre différents jeux de données. Une interprétation intuitive possible de ce critère est de considérer qu'il mesure l'aptitude d'une image à expliquer correctement l'autre. Le cas le plus défavorable est celui de deux images indépendantes (information mutuelle nulle) et le cas idéal est celui pour lequel les deux images sont parfaitement recalées (information mutuelle maximum). Ces critères issus de la théorie de l'information ont montré leur efficacité pour le recalage multimodal. Néanmoins, il est complexe à optimiser car il présente en général un paysage énergétique très cha-

huté (présence de minima locaux, convergence vers des solutions aberrantes)( voir figures 4.8 et 4.9) et requièrent un coût calculatoire élevée, ce qui a principalement limité leurs applications.

#### 4.8.4 Normalisation de l'information mutuelle

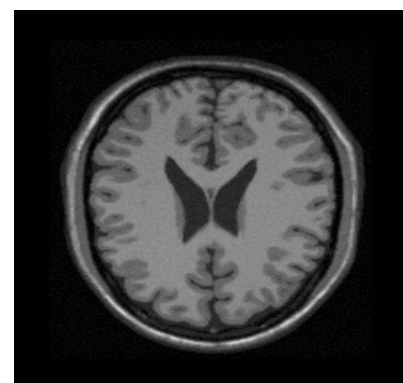
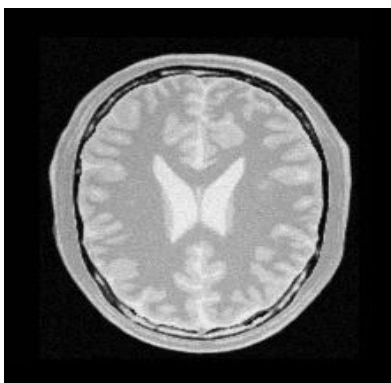
Soit deux images,  $A$  et  $B$ , la normalisation de l'information mutuelle est défini par :

$$NMI(A, B) = 1 + \frac{I(A, B)}{E(A, B)} = \frac{E(A) + E(B)}{E(A, B)} \quad (4.32)$$

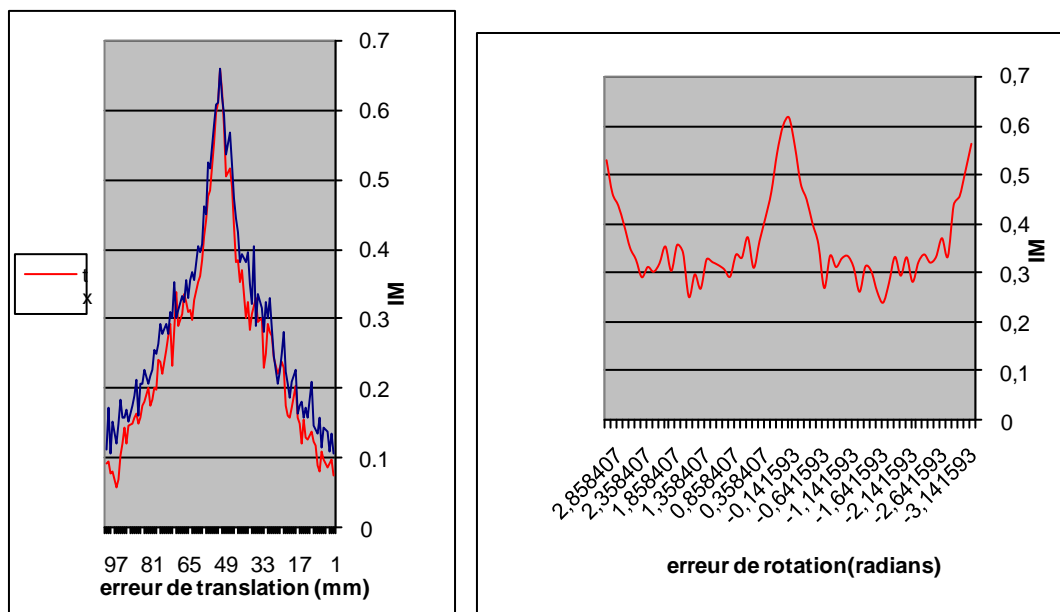
$E(A)$ ,  $E(B)$ ,  $I(A, B)$  et  $E(A, B)$  sont respectivement l'entropie de l'image  $A$ , l'entropie de  $B$ , l'information mutuelle et l'entropie conjointe des deux images.

#### 4.8.5 Vers un algorithme OEP avec facteur d'inertie adaptatif

Beaucoup d'efforts ont été investis pour obtenir une meilleure compréhension des propriétés de convergence des systèmes OEP. Ces efforts sont concentrés principalement sur une meilleure compréhension des paramètres de contrôle fondamentaux, à savoir les coefficients d'accélération, le facteur d'inertie, et la taille de l'essaim [Kennedy J et Eberhart R.C.,1995], [J. Kennedy .,1997], [ Y.Shi et Eberhart R.C .,1998],[ Y.Shi .,1998], [A.Carlisle et G. Dozier., 2001], [T. Beielstein et al.,2002 ]. Ces études empiriques montrent que le système OEP est sensible aux choix des paramètres de contrôle, en particulier le facteur d'inertie, les coefficients d'accélération. L'initialisation incorrecte de ces paramètres peut conduire à un comportement divergent ou cyclique.



(a) modalité avec pondération en densité de protons      (b) modalité avec pondération T1  
Figure 4.8 : images cérébrales IRM (coupe axiale)



**Figure 4.9 : l'information mutuelle en fonction d'erreur de translation et l'erreur de rotation**

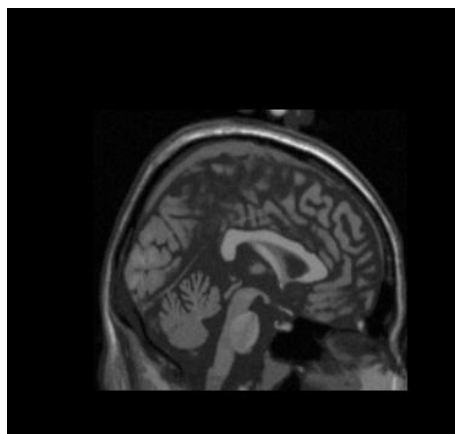
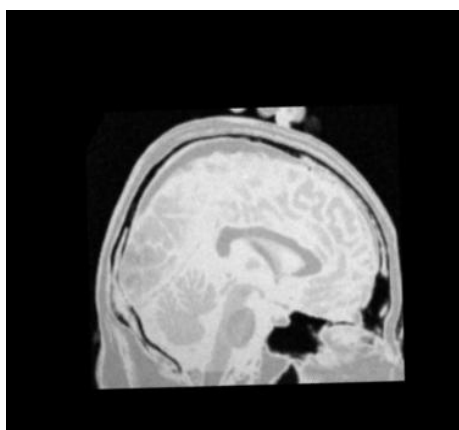
Les études empiriques de OEP donnent un aperçu du comportement des essaims de particules, fournissant des directives pour l'initialisation des paramètres, Par exemple, Eberhart et Shi constaté empiriquement que l'inertie d'un poids de 0,7298 et un coefficient d'accélération de 1,49618 sont des bonnes paramètres, ce qui conduit à des trajectoires convergentes [Beielstein et al,2002 ]. Bien que ces valeurs empiriques obtenues fonctionnent bien (en général), ils devraient être considérés avec prudence, car ces études empiriques sont basées uniquement sur un échantillon limité de problèmes. Il convient également de noter que les paramètres de contrôle dépendent du problème d'optimisation.

De ce fait, le traitement d'un problème passe, au préalable, par la définition d'un jeu de paramètres optimal. Or, cette étape, déjà longue et fastidieuse, nécessite une connaissance assez poussée des mécanismes de l'algorithme utilisé. Un utilisateur novice ayant un problème d'optimisation à résoudre risque de ne pas prendre le temps de trouver le jeu des paramètres optimaux. Il est donc important d'affranchir l'utilisateur de cette étape ou bien au moins réduire les paramètres définis par l'utilisateur en proposant des règles d'adaptation qui permettent de déterminer automatiquement les paramètres de l'algorithme.

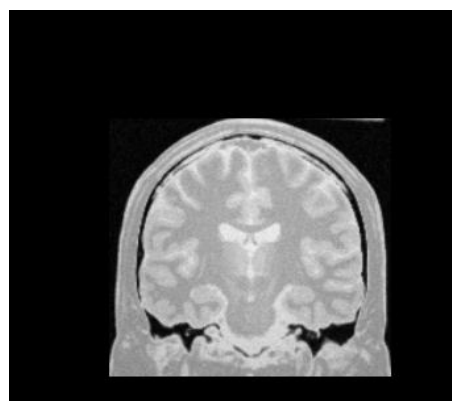
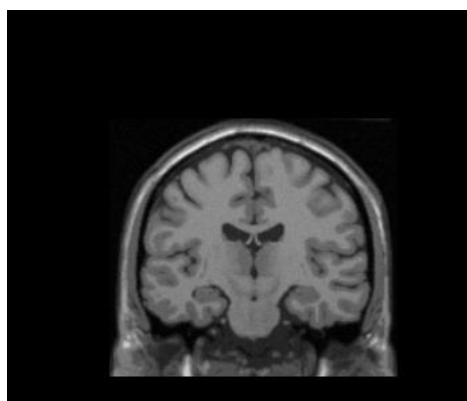
Dans cette section, nous utilisant trois modèles d'adaptation du facteur d'inertie déjà définies dans le chapitre 2 , L'adaptation est de trois types : diminution linéaire, diminution non linéaire et adaptation aléatoire.

## Chapitre 4 : Application de la méthode d'essaim particulaire au recalage des images multimodales

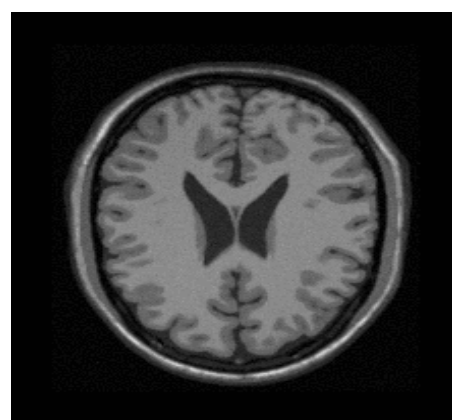
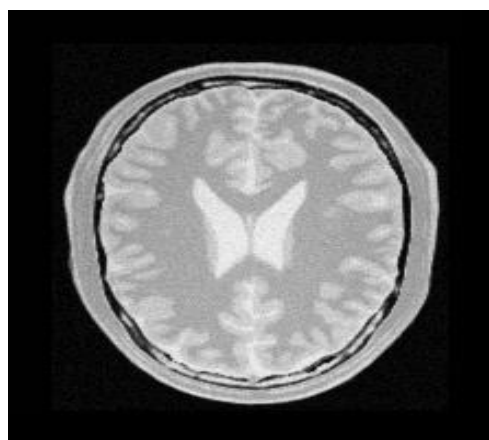
- linéaire:  $w(t) = 1 - \frac{t}{T}(1 - w_{\min})$
- non linéaire:  $w(t) = (1 - w_{\min}) \frac{(T - t)^2}{T^2} + w_{\min}$
- Aléatoire:  $w = (1 - r_1)r_2$



Paire d'images 1 : coupe sagittale taille:374X326



Paire d'images 2 : coupe coronal taille:374X326



Paire d'images 3: coupe axiale taille:221X257

**Figure 4.10 : les différentes coupes d'un volume cérébrale**

### **4.8.6 Experimentation globale**

Pour tester les performances de différentes techniques d'adaptation d'inertie, on applique les différents modèles sur un recalage rigide. Les expérimentations sont réalisées sur 3 paires d'images IRM cérébrales de deux modalités (figure 4.10).

Pour obtenir une meilleure évaluation des performances de chaque technique, la valeur d'IM qui donne une parfaite correspondance entre les deux images est connue au préalable.

#### **- Paramètre de contrôle**

Comme nous avons déjà vu, la performance d'OEP dépend des paramètres de contrôle, pour des propos de comparaisons les différents paramètres ont les mêmes valeurs pour toutes les techniques à savoir :

Les constantes d'accélération  $c_1=c_2=2$ , la taille d'essaim  $N=20$ , le nombre maximal d'itérations : 2000, la précision désiré : 0.01.

Deux types de voisinages sont utilisés, un voisinage local dynamique (LBest), dans ce type chaque particule informe  $K=3$  autres particules choisies aléatoirement durant chaque itération et un voisinage Global (Gbest) : le graphe d'information est complètement connecté.

#### **- Espace de recherche**

L'espace de recherche et de dimension  $D=3$  (degré de liberté de la transformation géométrique 2D-rigide), il se compose de : l'angle de rotation  $\theta$  et les deux paramètres de translation :  $X, Y$ .

#### **- Critères de comparaison**

Nous employons ici deux critères simples pour évaluer la performance de chaque méthode :

- 1) la précision (accuracy): elle est définie par le taux du succès obtenue après  $T$  exécutions de chaque technique.
- 2) Efficacité (Efficiency) : elle est mesurée par le nombre moyen d'évaluations pour obtenir la précision désiré après  $T$  exécutions.

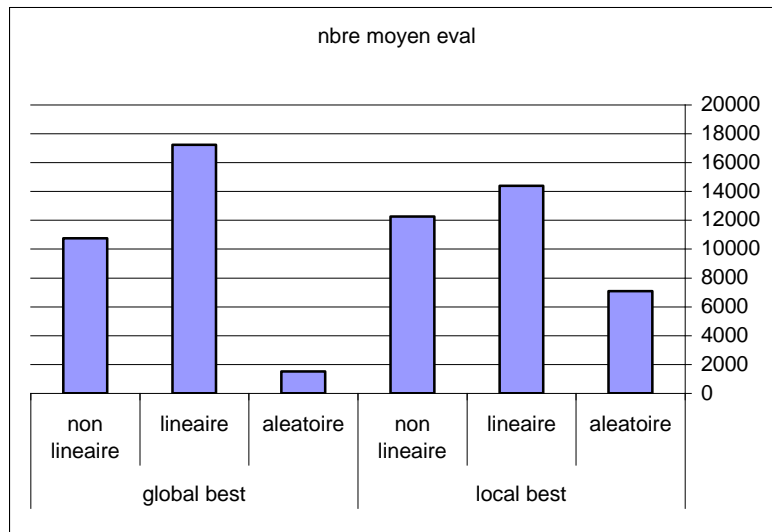
Le taux d'erreur n'est pas utilisé comme critère de comparaison puisque la valeur de l'information mutuelle qui donne une parfaite correspondance entre les deux images est connue au préalable.

### 4.8.7 resultats numeriques et discussion

Les nouvelles techniques d'adaptation sont évaluées sur les différents paires d'images, deux types de voisinages sont utilisés durant le processus de test : un voisinage dynamique local (LBest) et un voisinage global Gbest. Chaque modèle d'adaptation combiné avec l'un des deux types de voisinage est exécuté 10 fois pour obtenir les résultats numériques. L'exécution de chaque algorithme est arrêtée si l'erreur est inférieure à 0.01 ou si le nombre d'évaluations dépasse 40000.

Les tables 4.1, 4.2, 4.3 présentent les taux de succès sur dix exécutions de l'algorithme, et les histogrammes représentés dans la figure 4.11 donnent le nombre moyen d'évaluations de l'information mutuelle.

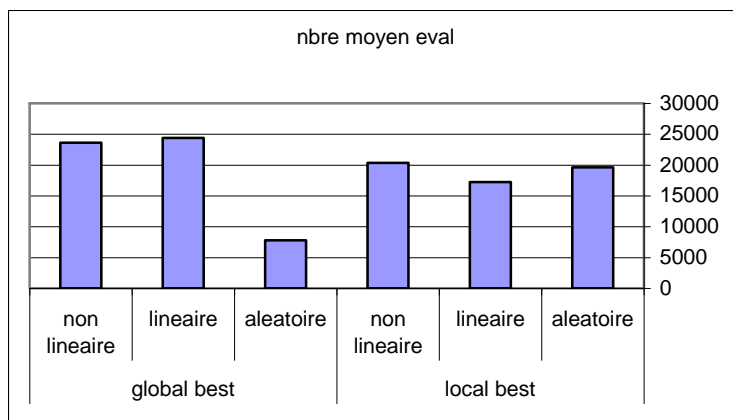
Les résultats calculés montrent que la combinaison de la technique aléatoire d'adaptation et un voisinage global (global best) donne le meilleur taux de succès vis-à-vis des autres techniques (supérieur à 90%). L'utilisation de cette technique apporte aussi un gain en temps appréciable au recalage des images IRM (nécessite un nombre d'évaluations plus faible que les autres méthodes) (voir figure 4.11)



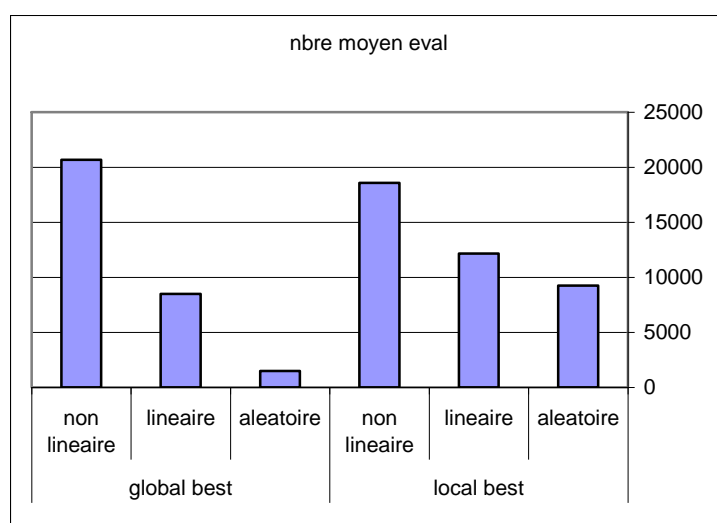
**Paire d'images 1**



## Chapitre 4 : Application de la méthode d'essaim particulière au recalage des images multimodales



**Paire d'images 2**



**Paire d'images 3**

**Figure 4.11: nombre d'évaluation pour chaque paire d'image**

**Table 4.1: taux de succès des paires d'image 1**

type voisinage	Modèle	nombre moyen d'évaluations	taux de succès
local best	Aléatoire	7074	90%
	Linéaire	14392	90%
	non linéaire	12254	100%
global best	Aléatoire	1516	100%
	Linéaire	17232	70%
	non linéaire	10752	100%

**Table 4.2: taux de succès des paires d'image 2**

type voisinage	Modèle	nombre moyen d'évaluations	taux de succès
local best	Aléatoire	19626	70%
	Linéaire	17208	90%
	non linéaire	20370	70%
global best	Aléatoire	7764	90%
	Linéaire	24372	50%
	non linéaire	23596	60%

**Table 4.3: taux de succès des paires d'image 3**

type voisinage	Modèle	nombre moyen d'évaluations	taux de succès
local best	Aléatoire	9246	100%
	Linéaire	12152	100%
	non linéaire	18586	100%
global best	Aléatoire	1472	100%
	Linéaire	8500	100%
	non linéaire	20674	80%

Ce travail a fait l'objet d'une communication [Khamer et khebaba, 2011].

## 4.9 conclusion

Dans ce chapitre, nous avons présenté trois techniques d'optimisation par essaim particulière partiellement adaptatif (OEP à facteur d'inertie adaptatif). Ces techniques ne nécessitent aucune intervention humaine pour la définition du facteur d'inertie et permettent donc à l'utilisateur novice une prise en main plus facile de l'algorithme.

Appliqué sur un problème de recalage des images IRM cérébrales multimodales, la technique d'adaptation aléatoire combinée avec un voisinage global donne des résultats compétitifs en termes de taux de succès et le nombre d'évaluations nécessaire pour maximiser l'information mutuelle.

## Conclusion générale

Dans ce mémoire nous avons investigué la capacité de l'optimisation par essaim particulière (OEP) pour l'optimisation mono-objectif et multi-objectifs, nous avons présentés aussi une version adaptative de la méthode appelée TRIBES. Dans ce travail nous sommes concentrés aussi sur l'évaluation de trois techniques d'optimisation globale par des essais particuliers à facteur d'inertie adaptatif, ses techniques sont appliquées à un recalage rigide des images IRM multimodales de cerveau ainsi que sur des problèmes bi-objectifs.

une étude sur l'état de l'art de l'optimisation par essaim particulière montre que l'application d'OEP, pour l'optimisation globale, n'était pas une tâche évidente. En effet, elle a nécessité une phase préliminaire de réglage des paramètres pour chaque problème étudié. Dans le premier chapitre nous avons montrés que les différentes modifications de l'algorithme de base de cette méthode sont réalisées dans le cadre d'établir des règles et des heuristiques génériques pour aidé l'utilisateur à l'initialisation correcte de ces paramètres.

Nous avons passé ensuite en revue l'état de l'art concernant l'adaptation de l'OEP aux problèmes multiobjectifs. Nous avons commencé par fournir une introduction sur les différentes notions nécessaires pour traité un problème multiobjectifs puis le principe de MOOEP dans lequel nous avons décrit son algorithme de base et les issues principales qui doivent être considérées en adaptant OEP à l'optimisation multi-objective, et alors nous avons analysé chacun d'eux en plus détail. Nous avons également proposé une taxonomie pour classifier les techniques courantes rapportées dans la littérature spécialisée, et nous avons fourni un aperçu des approches basées sur une telle taxonomie.

Une partie du présent travail a consisté en conception d'un nouveau modèle d'optimisation multi-objectif par essaims particulière basée sur OEP à facteur d'inertie adaptatif et une approche OEP agrégatif dynamique. La nouveauté principale de ce modèle consiste en utilisation de trois mécanismes d'adaptation du facteur d'inertie qui permettent de fournir davantage solutions sur l'ensemble Pareto-optimal.

L'un des inconvénients de l'OEP est que son comportement est très fortement dépendant des valeurs données à ses différents paramètres de contrôle. Un temps non négligeable est donc nécessaire au préalable pour régler le jeu de paramètres en fonction du problème posé. Pour pallier ce problème, de nombreuses recherches ont été menées, afin de proposer des méthodes adaptatives. Une méthode adaptative est une méthode pour laquelle les valeurs des paramètres sont calculées au cours de l'exécution de l'algorithme, en fonction du

comportement des particules, et non imposées par l'utilisateur. dans ce cadre nous avons présentés une version d'OEP appelée TRIBES . Cette version présente la particularité d'être totalement adaptative, l'utilisateur n'a qu'à définir son problème et son critère d'arrêt, sans se préoccuper des paramètres de l'algorithme. Tous les paramètres de contrôle sont calculés de manière autonome par l'algorithme, aucune intervention humaine n'étant nécessaire à la bonne résolution d'un problème.

Les performances de trois techniques d'adaptation du facteur d'inertie ont été aussi validées sur un problème de recalage des images IRM multimodales, les trois techniques sont : l'adaptation linéaire, non linéaire et l'adaptation aléatoire .Le critère de similarité utilisé dans l'expérimentation s'appel l'information mutuelle, Ce critère issu de la théorie de l'information a montré leur efficacité pour le recalage multimodal. Néanmoins, il est complexe à optimiser (présence de maxima locaux, convergence vers des solutions aberrantes, non-régularité, non-convexité).les trois approches utilisées pour maximiser l'information mutuelle ont été évaluées selon deux critères, la précision définie par le taux du succès obtenue après T exécutions de chaque techniques et l'efficacité qui mesure le nombre moyen d'évaluations pour obtenir la précision désiré après T exécutions. Les résultats montrent que la technique aléatoire est capable de pallier le problème du blocage aux optimas locaux.

## Perspectives

De nombreuses perspectives peuvent être envisagées. Dans cette section nous présentons succinctement celles qui nous paraissent être les plus intéressantes :

Pour les problèmes multi-objectifs on peut tester aussi la nouvelle technique d'adaptation du facteur d'inertie sur des problèmes plus de deux objectifs, particulièrement dans le cas d'OEP basé sur l'approche de Pareto.

Pour le recalage des images multimodales, on peut aussi élaborer une version totalement adaptative basé sur la version TRIBES et le différents stratégies de déplacements illustrées dans la littérature d'OEP (OEP sans paramètres) .

## ANNEXE A

---

### Code source d'une distribution sphérique implémenté en c

---

```
struct position alea_sphere(int D, struct position center, double radius,
double unif)
{
/* fournit aléatoirement un point dans une D-sphère ou selon une distribution
normale isotropique

unif = 1 = > distribution uniforme
unif > 0 but # 1 = > distribution non uniforme
unif < 0 = > Gaussian, avec deviation standard = abs(unif)
*/
int j;
double l;
double pw;
double r;
double sigma;
struct position x;
x.size = D;
pw = 1/(double)D;
// Stage 1. direction aléatoire
l= 0;
for (j = 0;j < D;j++)
{x.x[j] = alea_normal(0,1); l= l+ x.x[j]*x.x[j];}
l= sqrt(l);
// Stage 2. rayon aléatoire
if (unif > 0)
{r = alea(0,1); r = pow(r, pw*unif);}
else
{sigma = -unif; r = fabs(alea_normal(0, sigma)); r = pow(r, pw);}
for (j = 0;j < D;j++)x.x[j] = (radius*r*x.x[j]/l)+center.x[j];

return x;
}
```

## ANNEXE B

---

### Implémentation en c les différentes fonctions utilisées pour l'algorithme OEP basé sur l'approche agrégative dynamique et trois techniques d'adaptation du facteur d'inertie

---

```
#if defined(_MSC_VER)
#pragma warning ( disable : 4786 )
#endif

#include "stdafx.h"

#include <stdio.h>
#include <tchar.h>
#include "math.h"
#include <stdlib.h>
#include <time.h>

//-----
# define D_max 100 //la dimimension maximale de l'espace de recherche
# define N_max 100 //la taille maximale de l'espace de recherche
# define two_pi 6.283185307
# define E 2.718281828
# define ulong unsigned long //pour la génération des nombres aléatoires
# define RAND_MAX_KISS ((unsigned long) 4294967295)
//Structures
struct position { int size;double x[D_max]; double F,f1,f2;};// la position
d'une particule dans l'espace de recherche
struct vector {int size;double v[D_max]; };// le vecteur vitesse d'une
particule
struct archive{struct position p;int non_dom;};// archive externe des
solutions no dominées
//Sous routines de générations des nombres aléatoires
double alea(double has, double b);
int alea_entier(int a, int b);
int alea_whole(int has, int b);
ulong rand_kiss();
void seed_rand_kiss(ulong seed);

double ma_fonction(struct position X, int fonction);// calcule de la valeur de
la fonction de fitness
// routine de l'algorithme OEP
void pso();
double w1,w2;// poids des fonction objectives
double f1,f2;// variables mémorisés les valeurs des fonction objectives
FILE * f_run;

int _tmain(int argc, _TCHAR* argv[])
{
    pso();

    return 0;
}
```

## Annexe

```
//-----  
void pso()  
{  
    double c1, cmax; //Coefficients de confidences  
    int D; // dimension de l'espace de recherche  
    int d; //la dimension courante  
    double eps; //desired Precision  
    double min_f; // la meilleur performance  
  
    int function; //Code de la fonction à minimisée  
    int G,g; //index de la meilleur informatrice  
    int k;  
    int K; //maximum nombre des particules informées  
    int LIENS[N_max][N_max]; //liaisons d'information  
    int m;  
    struct position better; //Memorisation de la meilleur position obtenue  
    struct position P[N_max]; //positions  
    struct position P_m[N_max]; //les meilleurs positions obtenues  
    struct archive arch_ext[20]; //archives des solutions non dominées  
    int N; //taille d'essaim  
    int n; //indice de la particule courante  
    int nb_solnd; //nombre de solutions no dominées  
    int n_exec, n_exec_max; //Nombre d'exécutions  
    struct vector V[D_max]; // vecteurs des vitesses  
  
    int numb_iteration, compt_arch; //nombre d'itération et compteur  
d'archive  
    int num_max_it; // nombre maximal d'itérations  
    f_run = fopen( "f_run.txt", "w" );  
    K = 3; //nombre des particules informées par une particule  
  
    cmax = 0.5; // valeur maximale du coefficient d'accélération  
    N = 20; //taille d'essaim  
    D=2;  
    int num_max_it=3000; //initialisation du nombre maximal d'itération  
    compt_arch=0; //initialisation du compteur d'archive  
    seed_rand_kiss(1); //Initialization du générateur des nombres aléatoire  
    // le probleme traité  
    fonction = 5; //Code de la fonction.  
  
    //Initialisation des différents variables  
    n_exec = 0;  
    nb_solnd=0;  
    for (int k=0;k<20;k++) arch_ext[k].non_dom=0; //initialisation de  
l'archive  
    do{  
  
        numb_iteration=0;  
        //initialisations des variables d'grégations suivant l'approche DWA  
        w1=fabs(sin((two_pi)/200));  
        w2=1-w1;  
        //initialisation aléatoire de la position et la vitesse de chaque  
particule  
        for (n= 0;n < N;n++)  
        {
```

## Annexe

```
P[n].size= D;
V[n].size = D;

for (d = 0;d < D;d++)
{
    P[n].x[d] = alea(0, 1);
    V[n].v[d] = alea(-0.5, 0.5);
}
P[n].F= ma_fonction(P[n], fonction);//evaluation de la position
P[n].f1=f1;
P[n].f2=f2;
P_m[n] = P[n]; //meilleur position = position initiale
}
//stocker le meilleur résultat atteint jusqu'ici
better = P_m[0];
for (n = 0;n < N;n++) if (P_m[n].F < better.F) better = P_m[n];
numb_iteration=numb_iteration+1;

do
{ //modifier les valeurs de poids selon l'approche DWA
    w1=fabs(sin((two_pi*numb_iteration)/200));
    w2=1-w1;
    //modification d'inertie selon l'approche linéaire
    if (n_exec==0) ( c1=1-(numb_iteration*0,6)/num_max_it;
    //modification d'inertie selon l'approche non linéaire
    if (n_exec==1)c1=(0,6*(num_max_it-numb_iteration)*(num_max_it-
numb_iteration)
                    / (numb_iteration*numb_iteration)+0,4;
for (n = 0;n < N;n++) // Initialisation des liens d'information
    for (m = 0;m < N;m++)if(m!=n) LIENS[m][n] = 0;
                        else LIENS[n][n] = 1;
for (m = 0;m < N; m++)
    for (k= 0;k < K;k++){n = alea_entier(0, N-1);LIENS[m][n] = 1;}

//Diplacement de la particule
for (n= 0;n < N;n++) // pour chaque particule...
{
    // chercher la meilleur informatrice
    for (G= 0;G < N;G++) {if (LIENS[G][n] == 1)min_f =
P_m[G].F;break;}
    for (m = G + 1;m < N;m++)
        if (LIENS [m][n] == 1)
            if (P_m[m].F < min_f){ G = m;min_f = P_m[m].F;}

// calculer la nouvelle vélocité
for (d= 0;d < D;d++)
{ double alea1=alea(0, cmax);
  double alea2=alea(0, cmax);
  if (n_exec==1) c1=(1-alea1)*alea2;
  V[n].v[d] = c1*V[n].v[d] + alea1*(P_m[n].x[d]-P[n].x[d]);
V[n].v[d] = V[n].v[d] + alea2*(better.x[d]-P[n].x[d]);
  // déplacement
  P[n].x[d] = P[n].x[d] + V[n].v[d];
  // interval confinement

  if (P[n].x[d] < 0) {P[n].x[d] = alea(0,1);V[n].v[d] = 0;}
```



## Annexe

```
        if (P[n].x[d] > 1){ P[n].x[d] = alea(0,1);V[n].v[d] = 0;}}

// evaluation de la nouvelle position
P[n].F = ma_fonction(P[n], fonction);
P[n].f1=f1;
P[n].f2=f2;

// mise à jour de la meilleur position
if (P[n].F < P_m[n].F) P_m[n] = P[n];
// enregistrement de la meilleur résultat obtenue jusqu'ici
if (P_m[n].F < better.F) better = P_m[n];

}
// teste de la non dominance
int i,sol_dom;
sol_dom=0;

for ( i=0;i<20; i++)
    if (arch_ext[i].non_dom==1)
        if((better.f1>arch_ext[i].p.f1) && (better.f2>arch_ext[i].p.f2))
{sol_dom=1;break;}
//mise à jour de l'archive
if (sol_dom==0)

{
    nb_solnd++;
    for ( int i=0;i<20; i++)
        if (arch_ext[i].non_dom==0)
            { arch_ext[i].p=better;
              arch_ext[i].non_dom=1;
              break;}
}

//suppression des solutions dominées de l'archive externe
for (int i=0;i<20; i++)
if (arch_ext[i].non_dom==1)
    if((better.f1<arch_ext[i].p.f1) && (better.f2<arch_ext[i].p.f2))
    {
        arch_ext[i].non_dom=0;
        nb_solnd--;
    }

numb_iteration ++;
}
while ( numb_iteration < num_max_it);
    n_exec = n_exec + 1;

}
while ((n_exec< 3));
int l=0;
// enregistrement des solutions non dominées dans le fichier texte
for ( l=0;l<20;l++)
```

## Annexe

```
    {if (arch_ext[1].non_dom==1)
      {fprintf(f_run,"%f      %f      %f      %f",
arch_ext[1].p.f1,arch_ext[1].p.f2,arch_ext[1].p.x[0],arch_ext[1].p.x[1]);
      fprintf(f_run,"\n");}}
```

```
}
```

```
//-----
```

```
double alea(double a, double b)
```

```
{
//donner un nombre pseudo-aléatoire entre a et b
//selon une distribution pseudo uniforme
```

```
double r;
```

```
r = (double)rand_kiss()/RAND_MAX_KISS;
```

```
return a + r*(b-a);
```

```
}
```

```
//=====
```

```
int alea_entier(int a, int b)
```

```
{
```

```
// donner un nombre entier aléatoire entre a et b
```

```
int ir;
```

```
double r;
```

```
r = alea(0,1); ir = (int)(a + r*(b + 1-a)); if (ir > b) ir = b;
```

```
return ir;
```

```
}
```

```
//=====
```

```
double ma_fonction(struct position x, int function)
```

```
{
```

```
// évaluation de la fonction à minimisé dans la position x
```

```
//Ajouter votre fonction
```

```
int D,d;
```

```
double f,g;
```

```
D = 2;
```

```
switch (function)
```

```
{
```

```
case 1:
```

```
    f1= 0; for(d = 0;d < D;d++) f1 = f1 + x.x[d]*x.x[d];f1=f1/D;
```

```
    f2=0;for(d = 0;d < D;d++) f2 = f2 + (x.x[d]-2)*(x.x[d]-2);f2=f2/D;
```

```
    f=w1*f1+w2*f2;
```

```
break;
```

```
case 2:
```

```
    f1=x.x[0];
```

```
    g=0;for(d = 1;d < D;d++) g=g+x.x[d];g=1+(9/(D-1))*g;
```

```
    f2=g*(1-(f1/g)*(f1/g));
```

```
    f=w1*f1+w2*f2;
```

```
break;
```

```
case 3:
```

```
    f1=x.x[0];
```

```
    g=0;for(d = 1;d < D;d++) g=g+x.x[d];g=1+(9/(D-1))*g;
```

```
    f2=g*(1-sqrt(f1/g));
```

## Annexe

```
f=w1*f1+w2*f2;

break;
case 4: //
    f1=x.x[0];
    g=0;for(d = 1;d < D;d++) g=g+x.x[d];g=1+(9/(D-1))*g;
    f2=g*(1-sqrt(sqrt(f1/g))-(f1/g)*(f1/g)*(f1/g)*(f1/g));
    f=w1*f1+w2*f2;

break;
case 5: //
    f1=x.x[0];
    g=0;for(d = 1;d < D;d++) g=g+x.x[d];g=1+(9/(D-1))*g;
    f2=g*(1-sqrt(f1/g)-(f1/g)*sin(5*two_pi*f1));
    f=w1*f1+w2*f2;

break;

}
return f;
}
// random generator
static ulong kiss_x = 1;
static ulong kiss_y = 2;
static ulong kiss_z = 4;
static ulong kiss_w = 8;
static ulong kiss_carry = 0;
static ulong kiss_k;
static ulong kiss_m;

void seed_rand_kiss(ulong seed)
{
    kiss_x = seed | 1;
    kiss_y = seed | 2;
    kiss_z = seed | 4;
    kiss_w = seed | 8;
    kiss_carry = 0;
}

ulong rand_kiss()
{
    kiss_x = kiss_x * 69069 + 1;
    kiss_y ^= kiss_y << 13;
    kiss_y ^= kiss_y >> 17;
    kiss_y ^= kiss_y << 5;
    kiss_k = (kiss_z >> 2) + (kiss_w >> 3) + (kiss_carry >> 2);
    kiss_m = kiss_w + kiss_w + kiss_z + kiss_carry;
    kiss_z = kiss_w;
    kiss_w = kiss_m;
    kiss_carry = kiss_k >> 30;
    return kiss_x + kiss_y + kiss_w;
}
```

# Bibliographie

---

- [**Althof , 1997**] R.J Althof ., (1997)., " A rapid and automatic image registration algorithm with subpixel accuracy". IEEE Transactions on Medical Imaging, vol.16 .
- [**Barillot, 1999**] C. Barillot.,(1999), " Fusion de données et imagerie 3D en médecine. Habilitation à diriger des recherches", Université de Rennes I, 1999.
- [**Baumgartner et al, 2004**] U Baumgartner., Ch Magele., W Renhart., (2004). "Pareto optimality and particle swarm optimization". IEEE Transactions on Magnetics,40(2) :1172-1175, March 2004.
- [**Beielstein et al,2002** ] T. Beielstein, K.E. Parsopoulos, M.N. Vrahatis,(2002), "Tuning pso parameters through sensitivity analysis",. in Technical Report, Reihe Computational Intelligence CI 124/02, Department of Computer Science, University of Dortmund, 2002.
- [**Bhattacharya1 et Das, 2009**] M. Bhattacharya1, A. Das:"Registration of Multimodality Medical Imaging of Brain using Particle Swarm Optimization", Springer explorer 2009.
- [**Bookstein, 1989**] F.L. Bookstein.,(1989), " Principal warps : Thin-plate splines and the decomposition of deformations". IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, no. 6, pages 567-585, juin 1989.
- [**Brown, 1992**] L.G. Brown.,(1992), " A survey of image registration techniques". ACM Computing Surveys, vol. 24, no. 4, pages 325-376, décembre 1992.
- [**Canny,1986**] Canny. (1986). "A computational approach to edge detection". IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8.
- [**Carlisle et Dozier, 2001**] A. Carlisle, G. Dozier ,(2001), "An off-the-self pso",. in: Proceedings of the Workshop on Particle Swarm Optimization, Indianapolis, USA, 2001.
- [**Carlos et al,2002**] A Carlos. C. A.C Coello, A David. Van Veldhuizen, and B Gary.,(2002), " Lamont. Evolutionary Algorithms for Solving Multi-Objective Problems". Kluwer Academic Publishers, New York, May 2002.
- [**Charnes 1961**] Charnes A., Cooper W. Management Models and Industrial Applications of Linear Programming, vol 1, John Wiley, New-york.
- [**Chow et al, 2004**] C. Chow, H. Tsui., (2004)., "Autonomous agents response learning by multispecies particle swarm optimization",. Proceedings of the IEEE Congress on Evolutionary Computation., Vol. 1, pp. 778-785, 2004, IEEE Press.

- [Clerc et Kennedy, 2002]** M. Clerc., J. Kennedy.,(2002), "The Particle Swarm-Explosion, Stability, and Convergence in a Multidimensional Complex Space", IEEE Transactions on Evolutionary Computation, vol. 6, p. 58-73, 2002.
- [Clerc, 2003]** M. Clerc. (2003). TRIBES ,2003, "Un exemple d'optimisation par essaim particulaire sans paramètres de contrôle", Conférence OEP'03, 2 Octobre , 2003, Paris, France.
- [Clerc, 2005]** M. Clerc. (2005). "Binary Particle Swarm Optimisers: toolbox, derivations, and mathematical insights"
- [Clerc, 2005]** M.Clerc, (2005), livre, "L'optimisation par essais particulaires. Versions paramétriques et adaptatives ", Edition Lavoisier.
- [Clerc, 2006]** M. Clerc, (2006), article, "Confinements and Biases in PSO", Centre pour la communication scientifique directe (document librement consultable sur Internet : <http://hal.archives-ouvertes.fr>).
- [Coello, 1995]** C. A.C.Coello, (1995). " Multiobjective design Optimization of Counterweight Balancing of a Robot Arm using genetic Algorithm", Seventh International Conference on Tools with Artificial Intelligence, p. 20-23.
- [Coello, 1996]** C. A.C Coello ,(1996)," An Empirical study of Evolutionary Techniques for Multiobjective Optimization in Engineering Design", Ph.D. Thesis, Department of Computer sciences, Tulane University, New Orleans 1996.
- [Collette et Siarry, 2002]** Y. Collette, P. Siarry. ,(2002),"Optimisation multiobjectif", 2002, Edition Eyrolles.
- [Collignon et al, 1995]** A Collignon, D Vandermeulen, P Suetens., and G Marchal, (1995).,"3D multi-modality medical image registration using feature space clustering"., In Proceedings of the 1st International Conference on Computer Vision, Virtual Reality and Robotics in Medicine, pp. 195-204, Nice, France.
- [Cover et Thomas, 1991 ]** T.M Cover., J.A Thomas.," Elements of Information Theory". John Wiley & Sons, New York (1991).
- [Deb et al, 2002]** K Deb., S Jain., (2002). "Running Performance Metrics for Evolutionary Multiobjective Optimization", Tech. rep, KanGAL Report,2002.
- [Deb et Goldberg, 1989]** K Deb., Goldberg D. E. (1989). "An investigation of niche and species formation in genetic function optimization"., In J. David Schaffer, editor, Proceedings of the Third International Conference on Genetic Algorithms, pp. 42-50, San Mateo, California, June. George Mason University, Morgan Kaufmann Publishers,1989.

- [Dreo et Siarry 04]** J.Dreo, P. Siarry, (2004), "Metaheuristiques pour l'optimisation et l'autoorganisation dans les systemes biologiques ", resultat du Laboratoire d'Etude et de Recherche en Instrumentation, Signaux et Systèmes, Université de Paris
- [Eberhart et al,1996]** R.C. Eberhart, P.K. Simpson, R.W. Dobbins, "Computational Intelligence PC Tools", first ed.,Academic Press Professional, 1996.
- [Eberhart et Shi ,2000]** R.C. Eberhart, Y. Shi,(2000), "Comparing inertia weights and constriction factors"., in particle swarm optimization, in: Proceedings of the IEEE Congress on Evolutionary Computation, San Diego, USA, 2000, pp. 84–88.
- [Fieldsend et Singh, 2002]** J. E Fieldsend., S Singh., (2002)., "A multiobjective algorithm based upon particle swarm optimisation, an efficient data structure and turbulence"., In Proceedings of the 2002 U.K. Workshop on Computational Intelligence, pp. 37-44, Birmingham, UK, September 2002.
- [Ganser et al, 2004]** K.A. Ganser, H. Dickhaus, R. Metzner & C.R. Wirtz.,(2004)., "A deformable digital brain atlas system according to Talairach and Tournoux". Medical Image Analysis, vol. 8, no. 1, pages 3-22, mars 2004.
- [Goldberg et Richardson, 1987]** D. E Goldberg., J Richardson., (1987)., "Genetic algorithms with sharing for multimodal function optimization". Proceedings of the Second International Conference on Genetic Algorithms, pp. 41-49,1987.
- [Goldberg, 1989]** D.E. Goldberg. (1989). "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley, 1989.
- [Hill et al, 1991]** D.L. Hill, D.J. Hawkes, J.E. Crossman, M.J. Gleeson, T.C. Cox, E.E. Bracey, A.J. Strong & P Graves.,(1991)., "Registration of MR and CT Images For Skull Base Surgery Using Point-Like Anatomical Features". British Journal of Radiology, vol. 64, no. 767, pages 1030-1035, novembre 1991.
- [Hu et Eberhat, 2002]** X Hu. , R Eberhart, (2002). "Multiobjective optimization using dynamic neighborhood particle swarm optimization"., In Congress on Evolutionary Computation (CEC'2002), volume 2, pp. 1677-1681, Piscataway, New Jersey, May 2002.
- [Ignizo, 1981]** J.P Ignizio.,(1981), "The Determination of a Subset of Efficient Solutions via Goal Programming", Computing and Operations Research 3, pp. 9-16,1981.
- [Jenkinson, 2001]** M. Jenkinson and S. Smith, "A global optimization method for robust affiner registration of brain images", Medical Image Analysis, 5:143-156, 2001.
- [Khamer et khebaba, 2011]** L Khamer et A khebaba ,(2011), "L'optimisation par l'essaim des particules à facteur d'inertie adaptatif, application au recalage des images IRM multimodales",

Proceeding of the 1<sup>st</sup> International conférence on information systems and technologies-ICIST 2011, Université of Tebessa, Algeria.

**[Kennedy et Eberhart, 1995].** J. Kennedy ., R.C.Eberhart., (1995)., " Particle Swarm Optimization"., IEEE International Conférence on Neural Networks., Perth, Australia., p. 1942-1948, 1995.

**[Kennedy, 1999]** J. Kennedy., (1999)., "Small worlds and mega-minds : effects of neighborhood topology on particle swarm performance"., Proceedings of the 1999 IEEE Congress on Evolutionary Computation., pp. 1931-1938, 1999, IEEE Press.

**[Kennedy,1997]** J. Kennedy,(1997), "The particle swarm: Social adaptation of knowledge", in Proceedings of the IEEE International Conference on Evolutionary Computation, Indianapolis, USA, 1997, pp. 303–308.

**[Laguna et al, 2008]** M .Laguna., P.Y Yin., F .Glover. and J.X Zhu.,(2008)., "Cyber swarm algorithms - improving particle swarm optimization using adaptive memory strategies". 2008.

**[Laumanns et al, 2002]** M Laumanns., L Thiele., K Deb., E Zitzler, (2002)., "Combining convergence and diversity in Evolutionary multi-objective optimization"., Evolutionary Computation, 10(3), pp. 263-282,2002.

**[Lehmann et al, 1999]** T.M. Lehmann, C. Gonner and K. Spitzer.,(1999)., "Survey : interpolation methods in medical image processing." IEEE Transactions on Medical Imaging, vol. 18, no. 11, pages 1049-1075, novembre 1999.

**[Li et Sato, 2007]** Q. Li , I.Sato ," Multimodality Image Registration by Particle Swarm Optimization of Mutual Information", spring explorer,2007

**[Marr , 1980]** D.H.E Marr., (1980). "Theory of edge detection". Proceedings of the Royal Society London 207.

**[Miettinen, 1999]** K Miettinen., (1999). "Nonlinear Multiobjective Optimization". Kluwer Academic Publishers, Boston, Massachusetts 1999.

**[Naka et al ,2003]** S. Naka, T. Genji, T. Yura, Y. Fukuyama,(2003), "A hybrid particle swarm optimization for distribution state estimation", IEEE Transactions on Power Systems 18 (1) (2003) 60–68.

**[Nikou, 1999]** C. Nikou, "Contributions au recalage d'images médicales multimodales: approches par fonctions de similarités robustes et modèles déformables sous contraintes statistiques", thèse de doctorat Université Louis Pasteur, Strasbourg, 1999.

**[Noblet, 2006]** V. Noblet.,(2006)., "Recalage non rigide d'images cérébrales 3D avec contrainte de conservation de la topologie. PhD thesis", Universit é Strasbourg-I, march 2006.

- [Pareto, 1896]** V. Pareto. (1896). "Cours d'économie politique", Rouge, 1896.
- [Parsopoulos et al, 2002]** K. E Parsopoulos., M. N Vrahatis., (2002)., "Particle swarm optimization method in multiobjective problems"., In Proceedings of the 2002 ACM Symposium on Applied Computing (SAC'2002), pp. 603-607, Madrid, Spain,2002.
- [Parsopoulos et al, 2004]** K. E Parsopoulos., D Tasoulis., M. N Vrahatis., (2004), "Multiobjective optimization using parallel vector evaluated particle swarm optimization"., In Proceedings of the IASTED International Conference on Artificial Intelligence and Applications (AIA 2004), volume 2, pp. 823-828, Innsbruck, Austria, February 2004.
- [Plagianakos et Vrahatis.,2002 ]** V. P. Plagianakos and M. N. Vrahatis. Parallel evolutionary training algorithms for "hardware-friendly" neural networks. Natural Computing, 1(2-3):307- 322, 2002.
- [Poli, 2007]** R. Poli, (2007), article, "An Analysis of Publications on Particle Swarm Optimisation Applications", Swarm Intelligence, Volume 1 Number1
- [Pulido et Coello, 2004]** G. T Pulido, C. A Coello., (2004). "Using Clustering Techniques to Improve the Performance of a Particle Swarm Optimizer", In Kalyanmoy Deb et al., editor, Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2004), pp. 225-237, Seattle, Washington, USA, Springer-Verlag, Lecture Notes in Computer Science Vol. 3102.
- [Pulido, 2005]** G. T Pulido., (2005)., "On the Use of Self-Adaptation and Elitism for Multiobjective Particle Swarm Optimization. PhD thesis", Computer Science Section, Department of Electrical Engineering, CINVESTAV-IPN, Mexico, September 2005.
- [Rohr et al , 1996]** K. Rohr, H.S. Stiehl, R. Sprengel, W. Beil, T.M. Buzug, J. Weese and M.H.Kuhn.,(1991).," Point-Based Elastic Registration of Medical Image Data Using Approximating Thin-Plate Splines"., In Visualization in Biomedical Computing,LNCS-1131, pages 297-306, septembre 1996.
- [Rudolph, 1998].** G Rudolph., (1998), "On a multi-objective evolutionary algorithm and its convergence to the Pareto set". In Proceedings of the 5th IEEE Conference on Evolutionary Computation, pp. 511-516, Piscataway, New Jersey,1998.
- [Schaffer, 1985]** J.D Schaffer;r, (1985). "Multiple objective optimization with vector evaluated genetic algorithms", In J.J. Grefenstette ed., Proceedings of the First International Conference on Genetic Algorithms and Their Applications, Pittsburgh, PA, pp. 93-100.
- [Schaffer, 1985]** J.D. Schaffer., (1985)., "Multiple objective optimization with vector evaluated genetic algorithm", Proceedings of the 1rst International Conference on Genetic Algorithm, pp. 93-100, Lawrence Erlbaum Associates,1985.



[SER 97] Serra P., Stanton A. F., Kais S., "Pivot

[Sharma, 1997] R. Sharma, M. Pavel, "Multisensor image registration, in proceeding of the Society for Information Display", XXVIII pp: 951-954, 1997.

[Shi et Eberhart, 1998] Y. Shi, R.C. Eberhart,(1998), "Parameter selection in particle swarm optimization", in Proceedings of the Seventh Annual Conference on Evolutionary Programming, New York, USA, 1998, pp.591–600.

[Shi et Eberhart, 1998a] Y. Shi, R.C. Eberhart.,(1998)., "A modified particle swarm optimizer"., in Proceedings of the IEEE Congress on Evolutionary Computation., Piscataway., USA, 1998, pp. 69–73.

[Shi et Eberhart, 1999] Y. Shi, R.C. Eberhart.,(1999)., "Empirical study of particle swarm optimization", in Proceedings of the IEEE Congress on Evolutionary Computation, IEEE Press, 1999, pp. 1945–1950.

[Shi et Eberhart, 1998] Shi Y., Eberhart R., (1998). "Parameter selection in particle swarm optimization". In Evolutionary Programming VII : Proceedings of the Seventh annual Conference on Evolutionary Programming, pp. 591-600, NewYork, USA. Springer-Verlag.

[site web 1] <http://xps-swarm.essex.ac.uk/>

[Stacey et al, 2003] A Stacey., M Jancic., I Grundy., (2003)., "Particle swarm optimization with mutation". In Proceedings of the Congress on Evolutionary Computation, pp. 1425-1430,Camberra, Australia,2003.

[Suganthan, 1999] P .Suganthan., (1999). "Particle Swarm Optimizer with Neighborhood Operator"., In IEEE Congress on Evolutionary Computation., volume III., pages 1958–1961.

[Talbi, 1999] E.G. Talbi. (1999). "Métaheuristiques pour l'optimisation combinatoire multiobjectif : Etat de l'art", CNET, PE :98-757.33, 1999.

[Van et Engelbrecht, 2002] F. Van den Bergh, A.P. Engelbrecht,(2002), "A new locally convergent particle swarm optimizer", in: Proceedings of the IEEE Conference on Systems, Man and Cybernetics, October 2002, pp. 96–101.

[Van et Engelbrecht, 2006 ] F. Van den Bergh, A.P. Engelbrecht,(2006), "A study of particle swarm optimization particle trajectories", Information Science 176 (8) (2006) 937–971.

[Van Veldhuizen, 1999] D.A.Van Veldhuizen.,(1999)," Multiobjective, evolutionary algorithms : classification, analyses and new innovation", air force institute of Technology, United States 1999.

[Vandenbergh, 2002] F .Vandenbergh., (2002).," An Analysis of Particle Swarm Optimizers"., PhD thesis, Department of Computer Science, University of Pretoria.

**[Viola et Wells, 1995]** P. Viola, W.M. Wells.,(1995), "Alignment by maximization of mutual information". International Conference on Computer Vision (E. Grimson, S. Shafer, A. Blake and K. Sugihara, eds.), IEEE Computer Society Press, Los Alamitos, CA, pp. 16-23,1995.

**[Wachowiak et al, 2004]** M.P Wachowiak, , R Smolikova, , Y.F Zheng, J.M Zurada, A.S Elmaghraby,..(2004).. "An Approach to Multimodal Biomedical Image Registration Utilizing Particle Swarm Optimization". IEEE Trans. Evol. Comput. 8 (2004) 289-301

**[Wang et Yang, 2008]** V Wang., Y Yang., (2008)." Handling Multiobjective Problem with a Novel Interactive Multi-swarm PSO", Proceedings of the 4th International Conference on Intelligent Computing : Advanced Intelligent Computing Theories and Applications 5227, pp. 575-582.

**[Watts,2003]** D.J Watts., "Six Degrees: The Science of a Connected Age", Norton, New York,2003 method for global optimization", Physical Review, vol. 55, 1997, p. 1162-1165.

**[Zitzler et al ,2000]** E Zitzler, D Kalyanmoy, and T Lothar.,(2000) "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results"., Evolutionary Computation, 8(2):173–195, Summer 2000.