

République Algérienne démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique
Université Mentouri – Constantine

Faculté des sciences de l'Ingénieur
Département d'Informatique

N° d'ordre :
Série :

Mémoire

Présenté pour l'obtention du diplôme de Magister en Informatique
Option : Information & Computation

Thème :

L'optimisation multiobjectif et l'informatique quantique

Présentée par :

Mme Mahdi Karima

Dirigé par :

Dr. S. Meshoul

Soutenu le : / /

Devant le jury :

Table des matières

Introduction générale	i
Chapitre 1 : Etat de l'art de l'optimisation multiobjectif	1
1.1 Introduction	1
1.2 Définitions et principaux concepts en optimisation monoobjectif	1
1.3 Une hiérarchie des méthodes d'optimisation monoobjectif	3
1.4 L'optimisation multiobjectif	4
1.5 Les méthodes de résolution des problèmes multiobjectifs	9
1.6 Les Métaheuristiques et l'optimisation multiobjectif	15
1.7 Evaluation des méthodes d'optimisation multiobjectif	29
1.8 Conclusion	31
Chapitre 2 : L'informatique quantique	32
2.1 Introduction	33
2.2 Historique	33
2.3 Principe de base	34
2.4 Formalisme mathématique	34
2.5 L'informatique quantique	35
2.6 Les opérateurs logiques quantiques	39
2.7 Le parallélisme quantique	45
2.8 La cryptographie quantique	46
2.9 Correction d'erreurs quantiques	46
2.10 L'algorithmique quantique	48
2.11 Les algorithmes évolutionnaires quantiques	50
2.12 Conclusion	54
Chapitre 3 : Une Approche Evolutionnaire Quantique Pour Les Problèmes d'Optimisation Combinatoire Multiobjectif	55
4.1 Introduction	56
4.2 Formulation du problème du sac à dos multiobjectif (MOKP)	57

4.3 Définition d'un noyau évolutionnaire quantique	58
4.3.1 La représentation quantique du MOKP	58
4.3.2 Les opérateurs de variations quantiques	59
4.4 L'approche proposée	61
4.4.1 Fonctionnement de l'algorithme	63
4.4.2 Implémentation et évaluation de l'approche proposée	68
4.5 Conclusion	73
Chapitre 4 : Un Algorithme Evolutionnaire Quantique Bi-objectif pour le recalage d'images	74
4.1 Introduction	75
4.2 Le recalage d'images	75
4.2.1 Généralités et concepts de base	75
4.2.2 Caractéristiques d'un recalage	77
4.3 Problème de mise en correspondance	82
4.4 L'approche proposée	83
4.4.1 Formulation bi-objective du problème de recalage d'images.	83
4.4.2 Une représentation quantique du problème	85
4.4.3 Fonctionnement de L'algorithme	86
4.4.4 Résultats expérimentaux	90
4.5 Conclusion	93
Conclusion générale	94
Références Bibliographiques	96

Tables des figures

<i>Fig.2.1</i>	<i>Vers un monde quantique</i>
Fig.2.2	Notation matricielle et notation de Dirac
Fig.2.3	la sphère de Bloch
<i>Fig.2.4</i>	<i>Circuit quantique représentant le Controlled-Not</i>
<i>Fig.2.5</i>	<i>Table de vérité du C_{Not}</i>
<i>Fig.2.6</i>	<i>La porte Controlled-Controlled-Not</i>
Fig.2.7	la porte \sqrt{NOT}
Fig.2.8	circuit quantique pour la somme de deux bits
Fig.2.9	porte de Toffoli pour la conjonction
Fig.2.10	circuit correcteur d'erreur
Fig.2.11	circuit d'un algorithme quantique

Chapitre 1

Etat de l'art de l'optimisation multiobjectif

Dans ce chapitre, nous présentons les principes de base de l'optimisation multiobjectif . Nous donnerons aussi une classification des méthodes d'optimisation multiobjectif. Puis on présentera quelques une de ces méthodes. On finira par définir quelques métriques permettant l'évaluation de ces méthodes.

Sommaire

1.1	Introduction
1.2	Définitions et principaux concepts en optimisation Mono-objectif
1.3	Une hiérarchie des méthodes d'optimisation
1.4	L'optimisation multiobjectif
1.5	Domaines d'applications
1.6	Les méthodes de résolution des problèmes multiobjectif
1.7	Les Métaheuristiques et l'optimisation multiobjectif :
1.8	Evaluation des méthodes d'optimisation multiobjectif
1.9	Conclusion

1.1 Introduction :

La naissance de l'optimisation multi-objectif remonte à un ouvrage de W. Pareto 1906 sur l'économie politique, dans lequel l'auteur définit pour la première fois ce qu'est un optimum multi-objectif.

L'application de l'optimisation multi-objectif aux problèmes d'ingénierie remonte, aux alentours de la seconde guerre mondiale, elle est longtemps restée une science « anecdotique » à cause de son aspect hermétique, dû à la nécessité de maîtriser un bagage mathématique théorique assez important. Un changement radical est apparu récemment avec les métaheuristique (recuit simulé, algorithmes génétiques, etc.) qui sont des méthodes d'optimisation d'un abord plus facile.

Deux approches de l'optimisation multiobjective s'affrontent :

- Résolution vectorielle du problème : on ne modifie pas l'expression du problème
- Transformation du problème en un problème d'optimisation monoobjectif.

1.2 Définitions et principaux concepts en optimisation Mono-objectif:

1.2.1 Qu'est ce qu'un problème d'optimisation :

Un problème d'optimisation se définit comme la recherche du minimum ou du maximum (l'optimum) d'une fonction donnée.

Les variables de cette fonction sont souvent contraintes d'évoluer dans une certaine partie de l'espace de recherche. On aura donc un problème d'optimisation sous contraintes.

Mathématiquement, un problème d'optimisation s'écrit :

Minimiser	$f(\vec{x})$	(fonctions à optimiser)
Avec	$\vec{g}(\vec{x}) \leq 0$	(m contraintes d'inégalités)
Et	$\vec{h}(\vec{x}) = 0$	(p contraintes d'égalités)
Où	$\vec{x} \in \mathfrak{R}^n$, $\vec{g}(\vec{x}) \in \mathfrak{R}^m$ et $\vec{h}(\vec{x}) \in \mathfrak{R}^p$	

1.2.2 Vocabulaire et définitions :

- **Fonction objectif** : c'est la fonction f qu'on cherche à optimiser.
- **Variables de décision** : elles sont regroupées dans le vecteur \vec{x} qui correspond à l'ensemble des variables du problème.

- **Minimum globale :** un « point » \vec{x} est appelé minimum global de la fonction f si : $\forall \mathbf{x}, \mathbf{x} \neq \mathbf{x}^* \Rightarrow f(\mathbf{x}^*) < f(\mathbf{x})$
- **Minimum local fort :** \mathbf{x}^* est un minimum local fort de la fonction f si : $\forall \mathbf{x} \in v(\mathbf{x}^*), \mathbf{x} \neq \mathbf{x}^* \Rightarrow f(\mathbf{x}^*) < f(\mathbf{x})$, Où $v(\mathbf{x}^*)$ représente le voisinage de \mathbf{x}^*
- **Minimum local faible :** \mathbf{x}^* est un minimum local faible de la fonction f si : $\forall \mathbf{x} \in v(\mathbf{x}^*), \mathbf{x} \neq \mathbf{x}^* \Rightarrow f(\mathbf{x}^*) \leq f(\mathbf{x})$

1.2.3 Classification des problèmes d'optimisation :

Cette classification se fait selon :

- Le nombre de variable de décision : problème monovariabale ou multivariabales
- Le type de la variable de décision :
 - Nombre réel continu \Rightarrow problème continu
 - Nombre entier \Rightarrow nombre discret
 - Permutation sur un ensemble fini de nombres \Rightarrow combinatoire

1.3 Une hiérarchie des méthodes d'optimisation monocritères: tirée de [Collette, 02]

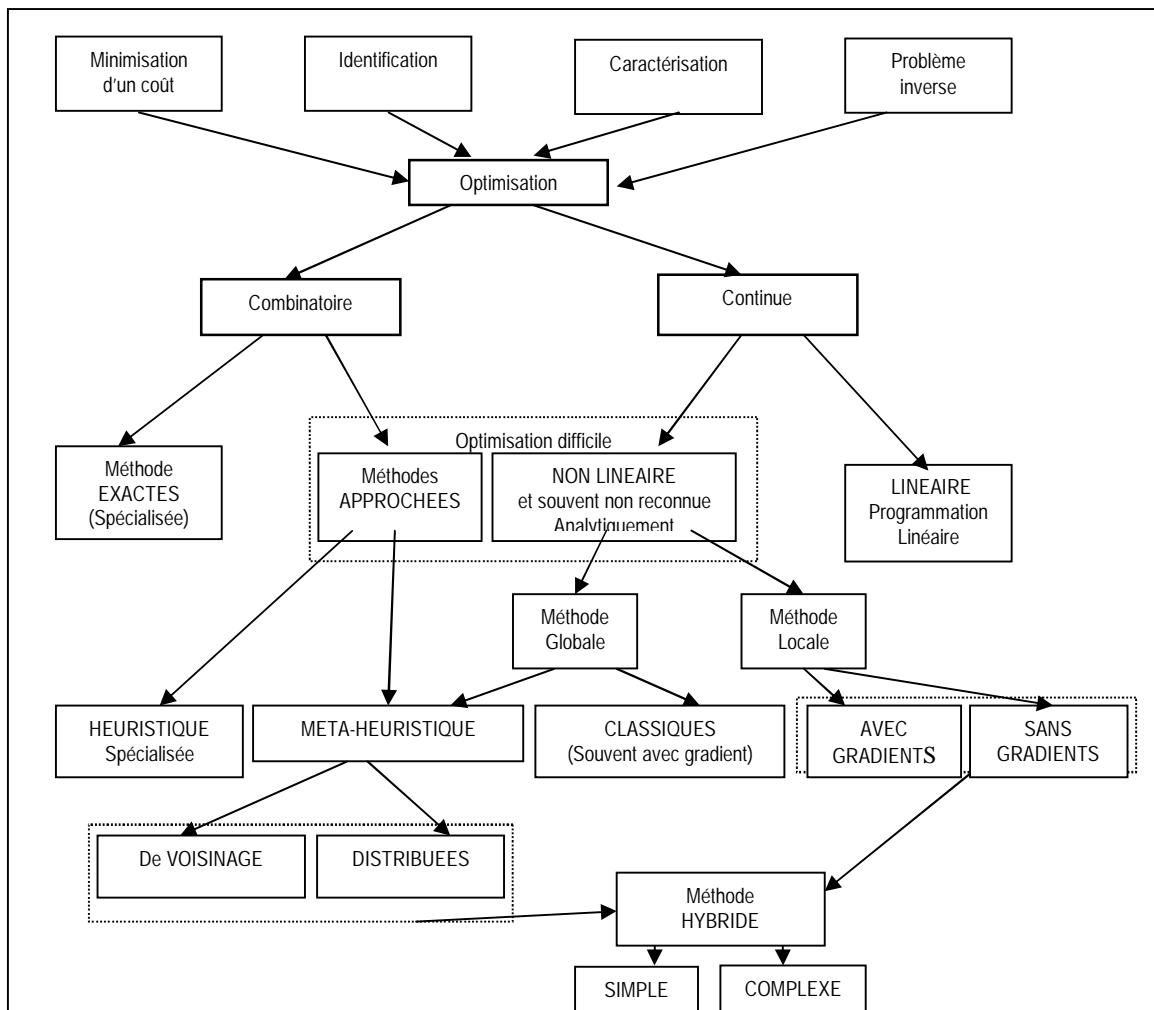


Fig.1.1 Classification des méthodes d'optimisation monocritères

1.4 L'optimisation multi-objectif :

L'optimisation mono-objectif consiste à maximiser (ou minimiser) une seule fonction objectif par rapport à un ensemble de paramètres. Or, la plupart des problèmes d'optimisation réels sont décrits à l'aide de plusieurs objectifs souvent contradictoires devant être optimisés simultanément. La solution optimale correspond alors aux meilleurs compromis possibles permettant de résoudre le problème. Il s'agit de l'optimisation multi-objectif, qui fournit aux décideurs un ensemble de solutions optimales dit Front de Pareto

Mathématiquement, un problème d'optimisation multi-objectif (PMO) est décrit comme suit:

Minimiser	$\vec{f}(\vec{x})$ (k fonctions à optimiser)
Avec	$\vec{g}(\vec{x}) \leq 0$ (m contraintes d'inégalités)
Et	$\vec{h}(\vec{x}) = 0$ (p contraintes d'égalités)
Où	$\vec{x} \in \mathfrak{R}^n$, $\vec{g}(\vec{x}) \in \mathfrak{R}^m$, $\vec{h}(\vec{x}) \in \mathfrak{R}^p$ et $\vec{f}(\vec{x}) \in \mathfrak{R}^k$

La résolution de ce problème consiste à minimiser **au mieux** ces k fonctions objectif. Donc le but de l'optimisation multi-objectif est de minimiser un groupe de fonctions objectif sans trop dégrader les valeurs des optima obtenus par rapport à ceux obtenus lors d'une optimisation mono-objectif effectuée objectif par objectif.

Sauf mention contraire explicite, tous les énoncés et définitions seront donnés dans le cadre de problèmes de minimisation. En effet un problème de maximisation peut être aisément transformé en problème de minimisation en considérant l'équivalence suivante :

$$\text{Maximiser } f(\vec{x}) \Leftrightarrow \text{Minimiser } -f(\vec{x})$$

1.4.1 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)

Si on prend l'exemple de dimensionnement d'une poutre devant supporter une charge donnée, on voudra obtenir une poutre de section la plus petite possible,

produisant la plus petite déformation possible, lorsque la charge repose au milieu de la poutre [Coello, 99]

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*.

1.4.2 La dominance :

Comme la solution optimale est une multitude de points de \mathcal{R}^n , il est vital pour identifier ces meilleurs compromis de définir une relation d'ordre entre ces éléments. Dans le cas des PMO, cette relation est appelée relation de dominance.

La relation de dominance nous permet donc de restreindre l'ensemble des solutions de compromis. Il existe différents types de relations de dominance [Collette, 02], qui permettent d'avoir suffisamment de degrés de liberté dans le choix de la relation qui reproduit au mieux le comportement d'un décideur.

Mais la plus célèbre et la plus utilisée est *la dominance au sens de Pareto*. Elle est définie par :

Le vecteur $\overset{\mathbf{I}}{u}$ domine le vecteur $\overset{\mathbf{I}}{v}$ (on note : $\overset{\mathbf{I}}{u} \mathbf{p} \overset{\mathbf{I}}{v}$) si :

$\overset{\mathbf{I}}{u}$ est au moins aussi bon que $\overset{\mathbf{I}}{v}$ dans tous les objectifs, et,

$\overset{\mathbf{I}}{u}$ est strictement meilleur que $\overset{\mathbf{I}}{v}$ dans au moins un objectif.

Afin de définir formellement la notion de dominance au sens de Pareto, les relations $=$, \leq et $<$ usuelles sont étendues aux vecteurs.

Définition 1 Soient $\overset{\mathbf{I}}{u}$ et $\overset{\mathbf{I}}{v}$ deux vecteurs de même dimension, $\overset{\mathbf{I}}{u} = (u_1, u_2, \dots, u_m)$ et $\overset{\mathbf{I}}{v} = (v_1, v_2, \dots, v_m)$

$$\overset{\mathbf{I}}{u} = \overset{\mathbf{I}}{v} \quad \text{ssi} \quad \forall i \in \{1, 2, \dots, m\}, u_i = v_i$$

$$\overset{\mathbf{I}}{u} \leq \overset{\mathbf{I}}{v} \quad \text{ssi} \quad \forall i \in \{1, 2, \dots, m\}, u_i \leq v_i$$

$$\overset{\mathbf{I}}{u} < \overset{\mathbf{I}}{v} \quad \text{ssi} \quad \overset{\mathbf{I}}{u} \leq \overset{\mathbf{I}}{v} \quad \wedge \quad \overset{\mathbf{I}}{u} \neq \overset{\mathbf{I}}{v}$$

Les relations $>$ et \geq sont définies de manière analogue.

On remarque que les relations qu'on vient de définir ne couvrent pas tous les cas possibles. Par exemple les point (ici vecteurs) $\overset{\mathbf{I}}{u} = (3, 5)$ et $\overset{\mathbf{I}}{v} = (6, 1)$ sont incomparables à l'aide de ces relations. Nous définissons maintenant la relation de dominance au sens de Pareto qui permet de comparer tous les vecteurs de décision possible.

Définition 2 La dominance au sens de Pareto : Soient $\overset{\mathbf{I}}{u}$ et $\overset{\mathbf{I}}{v}$ deux vecteurs de décision, $\overset{\mathbf{I}}{u} = (u_1, u_2, \dots, u_m)$ et $\overset{\mathbf{I}}{v} = (v_1, v_2, \dots, v_m)$,

Définition 5 une solution $\hat{u} \in X$ est dite non dominée par rapport à un ensemble $Y \subseteq X$ si et seulement si :

$$\neg \exists \hat{v} \in Y, \hat{v} \mathbf{p} \hat{u}$$

Définition 6 Ensemble des solutions non dominées : L'ensemble des solutions non dominées de X , est défini par l'ensemble

$$ND(X) = \{ \hat{u} \in X \mid \hat{u} \text{ est non dominé par rapport à } X \}$$

Définition 7 Front de Pareto : Soit F l'image de l'ensemble réalisable X dans l'espace des objectifs. Le Front de Pareto $ND(F)$ de F est défini comme suit :

$$ND(F) = \{ \hat{v} \in F \mid \neg \exists \hat{u} \in F, \hat{u} < \hat{v} \}$$

Le Front de Pareto est aussi appelé l'ensemble des solutions efficaces ou la surface de compromis.

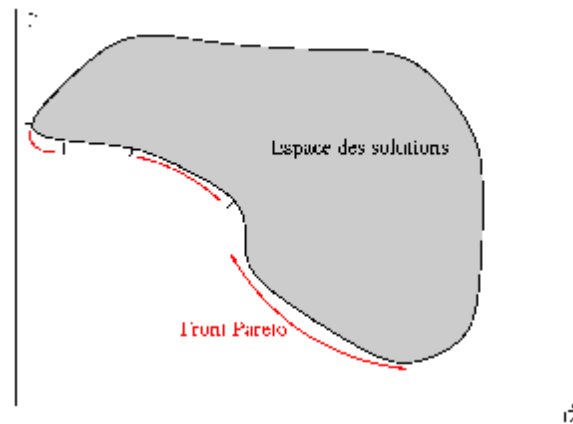


Fig.1.4 Front de Pareto

Algorithme d'Assignment du rang de Pareto

N : nombre de points de l'ensemble sur lequel on effectue les comparaisons

x_i : Un individu

t : la génération t

$Rang(x_i, t)$ Numéro de la surface de compromis auquel est affecté l'individu x_i à la génération t

RangCourant = 1

$m = N$

While $N \neq 0$ do

 For $i=1$ to m do

 If x_i est non dominé

```

Rang( $x_i, t$ ) = RangCourant
End for i
For i=1 to m do
  If Rang( $x_i, t$ ) = RangCourant
    Ranger  $x_i$  dans la population temporaire
    N=N-1
  End for i
  RangCourant = RangCourant +1
  m=N
End while

```

Définition 7 Point Idéal : les coordonnées du point idéal (p_i) correspondent aux meilleures valeurs de chaque objectif des points du front de Pareto. les coordonnées de ce point correspondent aussi aux valeurs obtenues en optimisant chaque fonction objectif séparément.

$$P_i = \min\{y_i \mid \hat{y} \in ND(F)\}$$

Définition 8 Point Nadir : les coordonnées du point Nadir (p_n) correspondent aux pires valeurs de chaque objectif des points du front de Pareto.

$$P_n = \max\{y_i \mid \hat{y} \in ND(F)\}$$

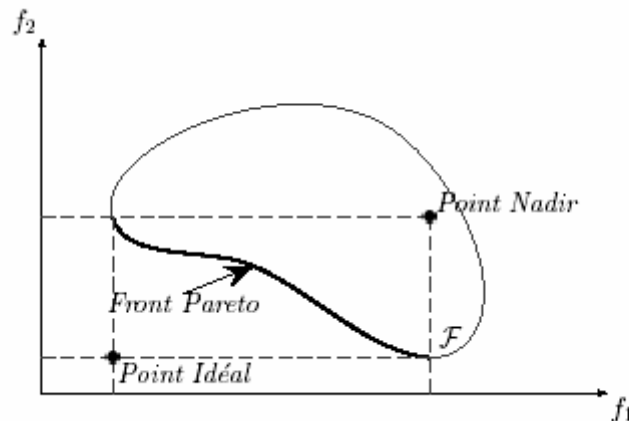


fig.1.5 Point Nadir et point Idéal

Définition 9 La connexité : un ensemble A est connexe, si et seulement si l'équivalence suivante est vérifiée :

$$x \in A \wedge y \in A \Leftrightarrow \text{Segment}(x, y) \subset A$$

La convexité est le premier indicateur de la difficulté du problème. Plusieurs méthodes d'optimisation sont incapables de résoudre d'une façon optimale des problèmes non convexes.

1.5 Domaines d'applications :

1.6 Les méthodes de résolution des problèmes multi-objectif :

Les méthodes d'optimisation multicritères s'intéressent aux problèmes essayant d'optimiser simultanément plusieurs objectifs. Elles peuvent être classées selon le schéma suivant [TAL, 01] :

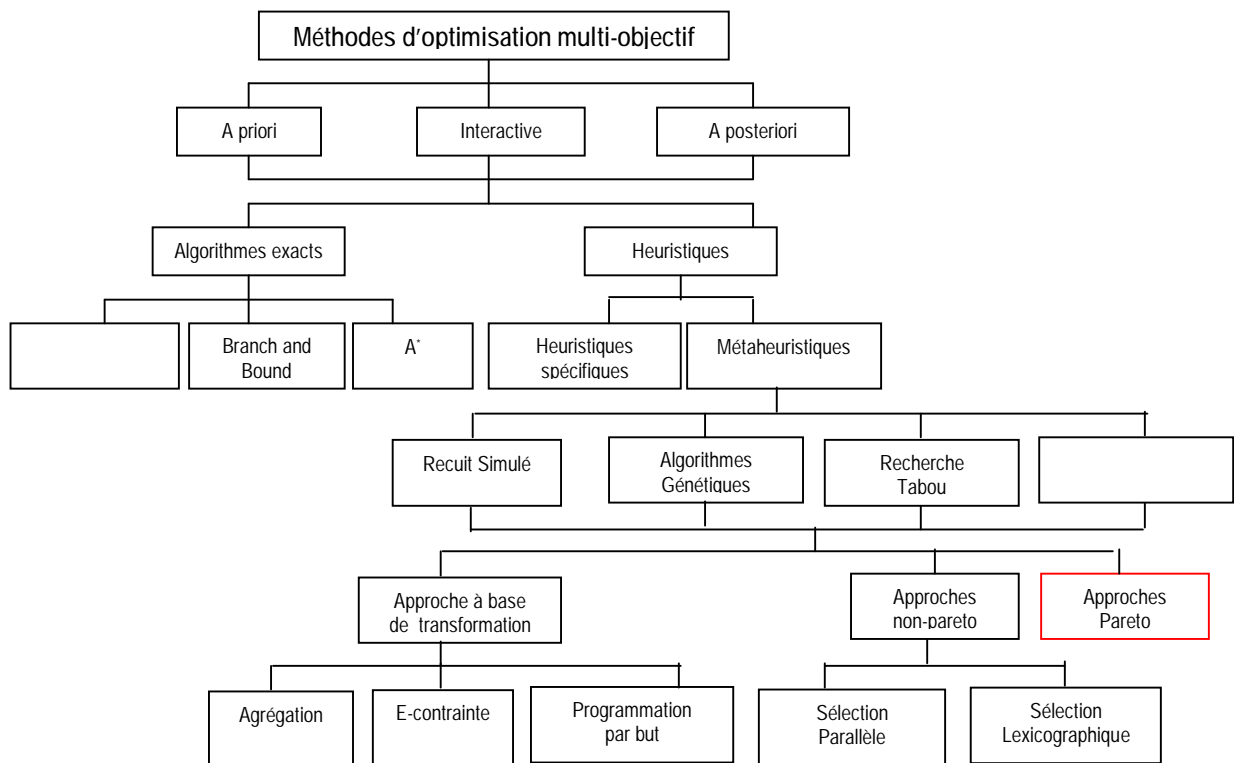


Fig.1.6 Classification des Méthodes d'optimisation multiobjectif

Selon le mode de coopération entre le solveur et le décideur, elles peuvent être rangées en trois familles :

- Les méthodes à préférence à priori :

L'utilisateur définit le compromis qu'il désire réaliser avant de lancer la méthode d'optimisation. On retrouve dans cette famille, toutes les méthodes agrégatives.

- Les méthodes à préférence progressives :

L'utilisateur affine son choix de compromis au fur et à mesure du déroulement de l'optimisation. On retrouve dans cette famille, toutes les méthodes interactives.

- Les méthodes à préférence à posteriori :

L'utilisateur choisit une solution de compromis parmi les solutions de la surface de compromis.

Les algorithmes de résolutions des PMOs se répartissent en deux grandes catégories :

A. LES METHODES EXACTES :

Ces méthodes permettent d'obtenir l'optimum global de manière exacte pour certains types de problèmes. Elles sont efficaces seulement lorsque le PMO à résoudre est bi-objectif et l'espace de recherche est de petite taille.

B. LES HEURISTIQUES :

a. Heuristiques spécifiques :

Elles sont dédiées à la résolution d'un PMO spécifique et ne fonctionnent que sur ce type de problèmes.

b. Métaheuristiques :

Elles sont fondées sur une idée générale. Et peuvent s'adapter à n'importe quel type de problème et donner de bons résultats. On trouve dans cette classe les algorithmes génétiques, la recherche tabou, le recuit simulé,...

Les métaheuristiques résolvant les PMO peuvent être classées en trois approches :

b.1 Transformation des PMO en uni-objectif :

b.1.1 Méthodes d'agrégation :

Les méthodes agrégatives fusionnent les différentes fonctions objectives pour se ramener à un problème d'optimisation mono-objectif.

Parmi ces méthodes on trouve La méthode de pondération des fonctions objectif [Coello, 98], qui est la plus simple des méthodes d'optimisation multi-objectif. Elle

est d'ailleurs appelée « l'approche naïve ». La transformation que l'on effectue est la suivante :

$$\boxed{\begin{array}{l} \text{Minimiser } \mathbf{f}(\vec{x}) \\ \text{Avec } \vec{x} \in S_r \end{array} \Rightarrow \left\{ \begin{array}{l} \text{minimiser } \sum_{i=1}^k w_i \cdot f_i(\vec{x}) \\ \text{Avec } \vec{x} \in S_r, w_i \geq 0 \text{ et } \sum_{i=1}^k w_i = 1 \end{array} \right.$$

Où S_r représente le domaine réalisable.

On peut définir des préférences vis à vis de certaines fonctions objectives en jouant sur les coefficients de pondération.

Discussion :

Cette méthode est très efficace point de vue algorithmique, mais elle ne permet pas de trouver les solutions enfermées dans des concavités.

b.1.2 Méthode ε -contrainte :

Ce n'est pas une méthode d'agrégation des fonctions objectives. Elle est aussi dite méthode du compromis [Miettinen, 99]. Elle transforme un PMO en un problème d'optimisation mono-objectif de la façon suivante :

- Choisir un objectif à optimiser prioritairement,
- Choisir un vecteur de contraintes initiales (\vec{e}),
- transformer le problème en gardant l'objectif prioritaire et en transformant les autres objectifs en contraintes d'inégalités comme suit :

$$\left| \begin{array}{l} \text{Minimiser } \vec{f}(\vec{x}) \\ \text{Avec } \vec{x} \in S_r \end{array} \right. \rightarrow \left| \begin{array}{l} \text{minimiser } \vec{f}_1(\vec{x}) \\ \text{avec } \vec{f}_2(\vec{x}) \leq \varepsilon_2, \dots, \vec{f}_n(\vec{x}) \leq \varepsilon_n \\ \vec{x} \in S_r \end{array} \right.$$

Discussion :

Cette méthode présente plusieurs inconvénients à savoir :

- la formulation des préférences utilisateur est délicate et nécessite une connaissance approfondie du problème de départ.
- Les contraintes rajoutées compliquent la résolution du problème.

b.1.3 Programmation par but :

On définit un ensemble de buts qu'on espère atteindre pour chaque fonction objectif. L'algorithme tente de minimiser l'écart entre la solution courante et ses buts.

Discussion :

Cette méthode est efficace et facile à implémenter mais la définition des buts à atteindre est souvent coûteuse. Pour plus de détails, voir [Miettinen, 99], [Coello, 98].

b.2 Approches non Pareto :

Ces approches traitent séparément les différents objectifs, elles sont sensibles au paysage du front de Pareto (convexité, continuité,...). Elles sont efficaces et faciles à implémenter seulement pour les PMO avec un nombre réduit d'objectifs.

b.2.1 Vector Evaluated Genetic Algorithm (V.E.G.A) :

Cette méthode est décrite dans [Coello, 98] et [Deb, 99]. elle opère comme suit :

Etape 1 : Itération i. initialisation d'une population de taille N.

Etape 2 : Création de k groupes composé chacun de N/k individus. K étant le nombre de fonctions objectif.

Etape 3 : Calcul des efficacité.

Mélange des individus.

Etape 4 : on applique l'algorithme génétique classique (croisement -mutation-sélection) puis on passe à l'itération i+1

Discussion :

Le danger de cette méthode est d'obtenir, en fin d'optimisation, une population d'individus moyens dans tous les objectifs.

b.2.2 L'ordonnancement lexicographique :

C'est une méthode très intuitive qui consiste à considérer les fonctions objectif les unes après les autres et à minimiser à chaque fois un problème mono-objectif, en complétant au fur et à mesure l'ensemble des contraintes [Coello, 98].

Cette méthode procède en k étapes :

§ Etape 1 :

$$\left| \begin{array}{l} \text{Minimiser } f_1(\mathbf{x}) \\ \text{Avec } \mathbf{x} \in S_r \end{array} \right.$$

on note f_1^* la solution de ce problème.

§ Etape2 :

$$\left| \begin{array}{l} \text{Minimiser } f_2(\mathbf{x}) \\ f_1(\mathbf{x}) = f_1^* \\ \text{Avec } \mathbf{x} \in S_r \end{array} \right.$$

On note f_2^* la solution de ce problème.

...

§ **Etape k :**

$$\left| \begin{array}{l} \text{Minimiser } f_k(\mathbf{x}) \\ f_1(\mathbf{x}) = f_1^*, f_2(\mathbf{x}) = f_2^*, \dots, f_{k-1}(\mathbf{x}) = f_{k-1}^* \\ \text{Avec } \mathbf{x} \in S_r \end{array} \right.$$

Discussion :

L'inconvénient majeur de cette méthode est qu'elle requiert un choix de la séquence des objectifs à minimiser. Deux choix différents génèrent deux solutions distinctes.

b.3 Approches Pareto :

Ces approches utilisent la notion de dominance lors de la sélection des solutions. Elles sont capables de générer des solutions Pareto situées dans des concavités. Historiquement, on peut considérer l'existence de deux générations des MOEAs.

La première génération des approches Pareto est caractérisée par l'utilisation :

- § d'un mécanisme de sélection Pareto. En effet la relation de dominance permet d'affecter des rangs aux individus de la population, ce qui fait apparaître la notion de front. C'est la technique de *ranking* [Goldberg, 89].
- § du *sharing (niching)* qui consiste à pénaliser la valeur d'adaptation en fonction du nombre d'individus au voisinage du regroupement cette technique introduite dans [Goldberg, 87] est largement utilisée maintenant.

Les algorithmes correspondants sont relativement simples. Parmi les approches les plus représentatives de cette génération on peut citer :

- § Multi-Objective Genetic Algorithm (M.O.G.A)
- § N.S.G.A (Non dominated Sorting Genetic Algorithm)
- § N.P.G.A (Niche-Pareto Genetic Algorithm)

Après environs dix années de succès, cette génération devient dépassée. La deuxième génération des approches Pareto est maintenant considérée comme l'état d'art de l'optimisation multi-objectif. Ce sont en général des approches élitistes manipulant une population secondaire externe. Parmi les approches les plus représentatives de cette génération on peut citer :

- § SPEA et SPEA2 (Strength Pareto Evolutionary Algorithm)
- § NSGA-II (Non dominated Sorting Genetic Algorithm)

- § PAES (The Pareto Archived Evolution Strategy)
- § PESA et PESA II (The Pareto Envelope-based Selection Algorithm)
- § MOMGA et MOMGA II (Multi-objective Messy Genetic Algorithm)
- § MicroGA pour l'Optimisation multi-objectif.

Comme les métaheuristiques ont permis la réalisation de méthodes de résolutions très performantes, On a préféré les aborder avec un peu plus de détails dans la section qui suit..

1.7 Les Métaheuristiques et l'optimisation multi-objectif :

Les métaheuristiques sont des méthodes générales de recherche dédiées aux problèmes d'optimisation difficile. Elles reprennent des idées que l'on trouve parfois dans la vie courante. Les principales heuristiques sont : le recuit simulé, la recherche tabou et les algorithmes génétiques.

Les métaheuristiques se répartissent en trois familles :

- Les méthodes déterministes de recherche d'optimum local :

Ces méthodes convergent rapidement, mais, la plus part du temps, elles se contentent de trouver l'optimum local.

- Les méthodes déterministes de recherche d'optimum global:

Ces méthodes permettent de trouver un optimum global et ne reposent pas sur un processus stochastique.

- Les méthodes stochastiques de recherche d'optimum global:

Ces méthodes reposent sur un processus stochastique pour la recherche de l'optimum global. Elles sont moins rapides que les méthodes déterministes mais elles peuvent trouver un optimum global difficile à atteindre.

Dans la suite, on ne s'intéressera qu'à cette catégorie de méthodes.

1.7.1 Le recuit simulé :

Le recuit simulé trouve ses origines dans la thermodynamique. Cette méthode est issue d'une analogie entre le phénomène physique de refroidissement lent d'un corps en fusion, qui le conduit à un état solide, de basse énergie. Il faut abaisser la température pour que le corps atteigne l'équilibre thermodynamique.

Ce processus métallurgique a été transposé à l'optimisation et a donné une méthode simple et efficace.

Algorithme recuit simulé

Init T (température initiale)
Init x (point de départ choisi au hasard)
Init Δt (température minimale)
While (not(end)) (s'arrêter si température atteint un seuil minimal)
 Y = voisin (x)
 $\Delta c = c(y) - c(x)$
if $\Delta c < 0$ **then** $y = x$ (y: nouveau point de départ)
else **if** $\text{alea}(0,1) < e^{-\Delta c/T}$ **then** $y = x$ (avec une probabilité = $e^{-\Delta c/T}$)
T = $\alpha(T)$ (diminuer la température par palier)
If $T < \Delta t$ **then** **end (while)**
Repeat (while)

Pour appliquer le recuit simulé aux MOPs, il suffit de déterminer comment calculer la probabilité d'acceptation d'un individu $\overset{\cdot}{x}$. La première version du recuit simulé multiobjectif a été proposée par Serafini [Serafini, 92] qui a proposé un ensemble de règles d'acceptation. Dans le cas multiobjectif, trois situations peuvent apparaître en comparant la nouvelle solution $\overset{\cdot}{x}'$ avec la solution courante $\overset{\cdot}{x}$:

- § $\overset{\cdot}{x}'$ domine $\overset{\cdot}{x}$
- § $\overset{\cdot}{x}'$ est dominé par $\overset{\cdot}{x}$
- § $\overset{\cdot}{x}'$ et $\overset{\cdot}{x}$ sont non dominées

Dans le premier cas la probabilité d'acceptation est naturellement égale à un, dans le second cas elle est inférieure à un, mais elle reste indéfinie (inconnue) dans le troisième cas. Actuellement, il existe plusieurs versions du recuit simulé multiobjectif. Parmi lesquels, on va présenter les méthodes MOGA et PASA.

1.7.1.1 La méthode M.O.S.A: (Multiple Objective Simulated Annealing)

Cette méthode a été proposée dans [Ulungu , 99]. Elle utilise le recuit simulé pour rechercher la surface de compromis.

On commence par définir une suite de fonctions donnant la probabilité d'acceptation d'une mauvaise solution, pour chaque fonction objectif :

$$\Pi_k = \begin{cases} \exp(-\Delta f_k / T_n) & \text{si } \Delta f_k > 0 \\ 1 & \text{si } \Delta f_k \leq 0 \end{cases}$$

Avec

$$\Delta f_k = f_k(\mathbf{y}) - f_k(\mathbf{x}_n).$$

T_n : la température du recuit simulé à l'itération n ,

f_k : $k^{\text{ième}}$ fonction objectif,

\mathbf{x}_n : Solution obtenue à l'itération n ,

\mathbf{y}_n : Point considéré lors de l'itération.

Ensuite on agrège ces probabilités. Il existe plusieurs possibilités d'agrégations :

- Produit des probabilités :

$$t(p, \lambda) = \prod_{t=1}^N (p_t)^{\lambda_k}$$

où λ_k correspond à un coefficient de pondération relatif à la $k^{\text{ième}}$ fonction objectif

- La plus petite probabilité :

$$t(\Pi, \lambda) = \min_{k=1, \dots, N} (\pi_k)^{\lambda_k}$$

On agrège, maintenant, les fonctions objectif de la façon suivante :

$$f_{eq}(\mathbf{x}) = \sum_{i=1}^n w_i \cdot f_i(\mathbf{x})$$

et $\Delta f_{eq}(\mathbf{x}) = f_{eq}(\mathbf{x}_{n+1}) - f_{eq}(\mathbf{x})$

Si $\Delta f_{eq} \leq 0$ alors $\mathbf{x}_{n+1} = \mathbf{y}$

Si $\Delta f_{eq} > 0$ alors $\mathbf{x}_{n+1} = \mathbf{y}$ avec la probabilité p et $\mathbf{x}_{n+1} = \mathbf{x}_n$ avec la probabilité $(1-p)$

Cette méthode diffère de la méthode de pondération des fonctions objectif couplée à un recuit simulé par la formule de calcul de la probabilité d'acceptation.

C'est cette agrégation de probabilités individuelles qui a rendu le recuit simulé plus adapté à l'optimisation multi-objectif.

Discussion :

Le couplage de la méthode de pondération des objectifs avec le recuit simulé a surmonté les limitations de la méthode.

Des travaux on montré que cette nouvelle méthode est plus efficace que la méthode d'origine.

L'effet stochastique a permis d'explorer des zones qui étaient auparavant inaccessibles (concavité)

1.7.1.2 La méthode P.A.S.A : (Pareto Archived Simulated Annealing)

Cette méthode, développée à EDF [Engrand , 98], utilise une agrégation des fonctions objectif couplée avec un système d'archivage de solutions non dominées.

On suppose que les fonctions objectif sont positives. On les agrège de la façon suivante :

$$G(\mathbf{x}) = \sum_{i=1}^n \text{Ln}(f_i(\mathbf{x}))$$

On calcule $\Delta(G) = G(\mathbf{x}') - G(\mathbf{x}) = \sum_{i=1}^n \text{Ln}(f_i(\mathbf{x}') / f_i(\mathbf{x}))$

On définit la probabilité d'acceptation p : $p = e^{-\Delta G / T}$

Algorithme de la méthode P.A.S.A

1. Choix de P_0 , une population d'archive initiale, éventuellement réduite à un seul élément x_0 , et choix de T_q initiale.
2. Soit \mathbf{x}'_n un voisin de \mathbf{x}_n ,
3. calcul de $G(\mathbf{x}'_n)$ et $\Delta(G) = G(\mathbf{x}') - G(\mathbf{x})$
4. si $\Delta G < 0$ alors $\mathbf{x}_{n+1} = \mathbf{x}'_n$ avec la probabilité p
5. si $\Delta G > 0$ alors soit $p = e^{-\Delta G / T_q}$
 - $\mathbf{x}_{n+1} = \mathbf{x}'_n$ avec la probabilité p
 - $\mathbf{x}_{n+1} = \mathbf{x}_n$ avec la probabilité $(1-p)$
6. si on sort de la boucle interne (rechercher un autre voisin de \mathbf{x}_n), alors

$$T_{q+1} = \alpha \cdot T_q \quad \text{avec } \alpha < 1$$
7. Principe de non-dominance pour l'archivage de \mathbf{x}_{n+1}
 - si \mathbf{x}_{n+1} domine une des solutions archivées \mathbf{y}_p , \mathbf{x}_{n+1} remplace \mathbf{y}_p dans la population archivées ;
 - Si \mathbf{x}_{n+1} est dominé par au moins une des solutions archivées, alors on ne l'archive pas ;
 - Si \mathbf{x}_{n+1} n'est pas dominé par la population archivée alors on l'ajoute à la population archivée, et on retire de celle-ci les solutions dominées par \mathbf{x}_{n+1} .

8. return to base : lorsque c'est nécessaire, pour que la recherche se fasse dans toute la surface de compromis, $x_{n+1}^i = y_i^i$ (i choisit aléatoirement dans la population archivée)
9. si (non convergence) alors retour en 2.

Discussion :

Dans cette méthode, la population initiale P_0 peut être issue d'une première optimisation multiobjectif. Ceci améliorera la qualité des résultats obtenus.

On peut aussi inclure un « savoir ingénieur » pour la gestion de l'archive.

Malheureusement, cette méthode ne supporte que les valeurs positives de fonctions objectives.

1.7.2 La recherche Tabou :

La recherche Tabou est une métaheuristique développée par Glover ([Glover ,86], [Glover ,90], [Glover, 97]). Elle est basée sur des idées simples, mais elle est néanmoins très efficace. Elle combine une procédure de recherche locale avec un certain nombre de règles et de mécanismes pour surmonter l'obstacle des optima locaux, tout en évitant de cycler. Elle a été appliquée avec succès pour résoudre de nombreux problèmes difficiles d'optimisation combinatoire.

Le principe de la méthode Tabou est simple : On part d'une solution initiale x , puis on cherche le meilleur voisin $s(x)$ dans le voisinage de x [$s(x)$], l'originalité de la méthode réside dans le fait que l'on retient le meilleur voisin de x même si celui-ci est plus mauvais que x .

Ce critère autorisant les dégradations de la fonction objectif évite à l'algorithme d'être piégé dans un minimum local, mais il induit un risque de cyclage. Pour régler ce problème, l'algorithme mémorise la liste des transformations interdites (telle que de $s(x)$ à x) dans la liste Tabou interdisant la transformation inverse d'une transformation faite récemment.

Algorithme Recherche Tabou

Init $L = f$ (liste des points tabous)

Init x (point de départ)

Init k_{fin} (nb d'itérations max total)

Init $k = 0$ et $l = 0$

While (not(end))

$y = \text{voisinage}(x) - L$ (on prend un voisinage sans points tabous)

$\Delta C = C(y) - C(x)$

$x = y$ avec $y \in \text{voisinage}(x) - L$ tel que $\min C(y) - C(x)$

$L = L + \{x\}$

$k = k + 1$

Update L (on retire des mouvements tabou de L suivant des critères d'aspiration)
 If $k = k_{fin}$ then end (while)
 Repeat

Plusieurs versions multiobjectif de la méthode Tabou ont été proposées ([Gandibleux, 97], [Hansen, 98], [Benabdel, 97], [Barichard, 03])

1.7.3 Les algorithmes génétiques :

Les algorithmes génétiques s'inspirent du fonctionnement de l'évolution naturelle, notamment la sélection de Darwin et la procréation selon les règles de Mendel [Goldberg, 94].

La sélection naturelle favorise les individus d'une population qui sont le mieux adaptés à un environnement. La sélection est suivie de la procréation, réalisé à l'aide de croisement et de mutation au niveau du patrimoine génétique des individus (« génotype »). Ainsi deux individus parents, qui se croisent, transmettent une partie de leur patrimoine génétique à leurs descendants.

Le génotype de l'enfant fait que celui-ci est plus au moins adapté à l'environnement. S'il est bien adapté, il a une grande chance de procréer dans la génération future.

- Vocabulaire utilisé :

~ Génotype ou chromosome: individu

~ Individu : codage sous forme de gènes d'une solution potentielle à un problème d'optimisation.

~ Gène : un chromosome est composé de gènes. Dans le codage binaire, un gène vaut 0 ou 1

~ Phénotype : c'est la valeur de la solution représentée par un génotype.

~ Efficacité ou adaptation: correspond à la performance d'un individu dans la résolution d'un problème donné. Par exemple, si l'on considère un problème de maximisation d'une fonction f , l'efficacité de l'individu croîtra avec sa capacité à maximiser cette fonction.

F(x)	10	20	5
Efficacité	2	4	1
Individu	1	2	3

Fig.1.7 Calcul de l'efficacité d'un individu

- Principe des algorithmes génétiques :

On démarre avec une population initiale. Après avoir déterminé l'efficacité de chacun de ses individus, on opère une reproduction. On copie les individus proportionnellement à leur efficacité. Par exemple, l'individu 2 de l'exemple précédent est recopié quatre fois. On va sélectionner les individus suivant leur efficacités, puis les fusionner avec d'autres individus choisis aléatoirement. Cette opération s'appelle « croisement » et consiste à :

- Sélectionner aléatoirement une position de croisement
- A cette position, on sépare le codage puis on échange les morceaux de droite entre les deux individus.

Exemple de croisement :

Individu 1 :

0	1	1	1	0
---	---	---	---	---

Individu 2 :

0	0	0	1	1
---	---	---	---	---

Position de croisement = 3

Résultat :

0	1	1	1	1
---	---	---	---	---

0	0	0	1	0
---	---	---	---	---

Après cette opération de croisement, la population est diversifiée, on sélectionne, au hasard, un certain nombre d'individus auxquels on applique des mutations. La « mutation » consiste à inverser un des gènes de l'individu.

Exemple de mutation :

Individu :

0	0	0	1	1
---	---	---	---	---



0	0	1	1	1
---	---	---	---	---

Algorithme génétique

Initialisation de la population
 Evaluation des fonctions objectif

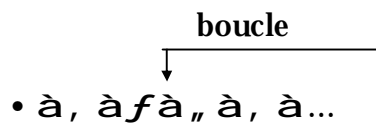
```

Calcul de l'efficacité
For i= 1 to MaxIter
    %Sélection aléatoire,
    Sélection proportionnelle à l'efficacité
    Croisement
    Mutation
    Evaluation des fonctions objectif
    Calcul de l'efficacité
    %Fusion des populations
End For
    
```

1.7.3.1 L'optimisation multiobjectif et les algorithmes génétiques :

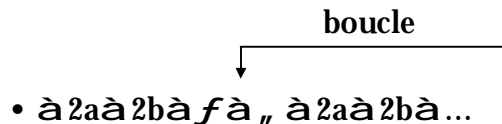
Les algorithmes génétiques sont très bien adaptés au traitement d'un problème d'optimisation multi-objectif.

Le schéma de fonctionnement d'un algorithme génétique pour la résolution d'un problème d'optimisation mono-objectif est le suivant :



- : initialisation de la population
- , : évaluation de l'efficacité des individus de la population
- f* : croisement
- „ : mutation
- ... : sélection

le schéma de fonctionnement d'un algorithme génétique pour la résolution d'un PMO est le suivant :



- : initialisation de la population
- 2a : évaluation de l'efficacité des individus de la population
- 2b : transfert vecteur /efficacité
- f* : croisement
- „ : mutation
- ... : sélection

la plupart des algorithmes utilisent la distance de « Hamming » pour caractériser la différence entre deux individus. Cette distance est calculée de la façon suivante : chaque différence entre les gènes compte pour plus un (+1).

a. La méthode M.O.G.A:(Multiple Objective Genetic Algorithm)

Cette méthode est présentée dans [Deb, 99] et [Fonseca, 93]. Elle utilise la relation de dominance de Pareto pour calculer l'efficacité des individus.

Le rang d'un individu x_i à la génération t est donné par :

$$\text{Rang}(x_i, t) = 1 + p_i^{(t)}$$

Où $p_i^{(t)}$ est le nombre d'individu qui dominant x_i à la génération t .

Pour calculer l'efficacité des individus, on suit les étapes suivantes :

1. classer les individus en fonction de leur rang.
2. affecter une efficacité à un individu en interpolant à partir du meilleur rang au plus mauvais.

Algorithme M.O.G.A

Initialisation de la population
 Evaluation des fonctions objectif
 Assignation d'un rang basé sur la dominance
 Assignation d'une efficacité à partir du rang
 For $i=1$ to $\max G$
 Sélection aléatoire proportionnelle à l'efficacité
 Croisement
 Mutation
 Evaluation des fonctions objectif
 Assignation d'un rang basé sur la dominance
 Assignation d'une efficacité à partir du rang
 Endfor

Discussion :

Cette méthode ne permet pas, dans certain cas, d'obtenir une diversité dans la représentation des solutions.

b. La méthode N.S.G.A:(Non dominated Sorting Genetic Algorithm)

Cette méthode reprend les grandes lignes de la méthode MOGA. La différence principale intervient dans l'efficacité d'un individu [Srinivas, 93].

Dans un premier temps, on affecte à chaque individu un rang de Pareto. Tous les individus de même rang sont classés dans une catégorie. Chaque catégorie aura une efficacité inversement proportionnelle à son rang.

Pour maintenir la diversité de la population, on va partager la valeur de l'efficacité d'une catégorie entre ses individus de la façon suivante :

- On calcule, pour chaque individu, le compte de ses voisins

$$m_i = \sum_{j=1}^K Sh(d(i, j))$$

$$Sh(d(i, j)) = \begin{cases} 1 - \left(\frac{d(i, j)}{s_{share}}\right)^2 & \text{si } d(i, j) < s_{share} \\ 0 & \text{sin on} \end{cases}$$

avec K : nombre d'individus de la catégorie.

s_{share} : permet de définir une zone d'influence pour le calcul de l'efficacité de l'individu

- L'efficacité f_i de l'individu i est donnée par : $f_i = \frac{F}{m_i}$

Discussion :

C'est une méthode efficace, qui permet d'obtenir une diversité dans la représentation des solutions, mais elle reste très sensible au choix du diamètre de la niche s_{share} . Récemment une version améliorée de NSGA appelée NSGA-II est apparue [Deb, 00].

c. La méthode SPEA : (Strength Pareto Evolutionary Algorithm)

La méthode SPEA implémentée par Zitzler et Thiele [Zitzler, 98] est l'illustration même d'un algorithme évolutionnaire élitiste. SPEA maintient un archive externe contenant les meilleurs front de compromis rencontré durant la recherche.

Algorithme SPEA

Initialiser la population P_0 et créer l'archive externe vide $\bar{P} = f$

Mise à jour de \bar{P} à partir des individus non dominés de P_0

Tant que critère- d'arrêt- non- rencontré faire

- Calcul de la valeur d'adaptation pour tous les individus de $P + \bar{P}$

- Sélection dans $P_t + \bar{P}$ en fonction de la valeur d'adaptation
- Croisement
- Mutation
- Mise à jour de \bar{P} à partir des individus non dominés de P

FinTant que

Récemment une version améliorée de SPEA appelée SPEA-2 est apparue [Eckart, 01].

d. La méthode PAES: (The Pareto Archived Evolution Strategy)

La méthode PAES a été introduite par Knowles and Corne [Know, 99]. Elle utilise une stratégie évolutionnaire simple de recherche locale. elle maintient un archive externe des solutions non dominées pour assurer une diversité dans les solutions trouvées

e. La méthode PESA : (The Pareto Envelope-based Selection Algorithm)

La méthode PESA a été proposée par Corne et al (2000). Elle utilise une population interne de taille réduite et une population externe de taille considérable.

f. La méthode MOMGA: (Multi-objective Messy Genetic Algorithm)

Cette méthode a été proposée par Van Veldhuizen and Lamont (2000).

1.7.4 les méthodes hybrides :

Afin d'améliorer les performances d'un algorithme, on essaye de le combiner avec une autre méthode [TAL, 00]. Ce principe général appelé hybridation, peut s'appliquer à plusieurs méthodes. Un cas particulier de l'hybridation entre deux méthodes consiste à combiner un algorithme génétique avec une méthode de recherche locale. Dans une telle hybridation on substitue souvent la mutation par une méthode de recherche locale. Dans le cas des PMOs, on peut citer les méthodes hybrides suivantes :

- § la méthode MOTS [Hansen, 97 ; Gandibleux , 97] combinant une population et une recherche Tabou,
- § La méthode PSA [Czyzak, 98] combinant un algorithme génétique et le recuit simulé,
- § La méthode M-PAES [Corne, 00] intégrant un schéma généralisant l'implémentation d'un grand nombre d'algorithmes hybrides pour l'optimisation multiobjectif.

1.7.5 Autres métaheuristiques :

D'autres Métaheuristiques ont été adapté au cas multiobjectif et ont prouvé leur succès.

§ Les colonies de fourmis [Mariano, 99], [Iredi , 2001], ...

§ L'Optimisation par essaim de particules : [Moore and Champman, 1999], [Ray and Liew, 2002], [Coello and Lechuga, 2002], ...

§ Les réseaux de neurones artificiels : [Yoo and Hajela, 1999], [Kurapati and Azarm,2000], [Cui et al., 2001], [Coello and Cruz, 2002]

§ Les algorithmes culturelles :[Coello and Landa, 2003]

1.8 Evaluation des méthodes d'optimisation multi-objectif:

Vu le nombre important des méthodes d'optimisation multiobjectif, il devient indispensable de disposer des outils nécessaires pour leur comparaison. L'évaluation des surfaces de compromis est un problème délicat qui oblige à utiliser plusieurs mesures différentes, car il est impossible de représenter par une unique valeur réelle la qualité des solutions, la taille du front, la répartition des solutions sur le front, Parmi les métriques les plus utilisées, on trouve :

1.8.1 l'hypervolume et le rapport d'hypervolume:

Cette métrique a été définie dans [Zitzler, 99], elle permet de mesurer le volume compris sous la courbe formée par les points de l'ensemble à évoluer. Ainsi, dans le cas d'un problème bi-objectif, on parle d'hypersurface et non d'hypervolume.

$$H = \left\{ \mathbf{U}_i a_i / v_i \in A \right\},$$

Où a_i désigne le volume occupé par la solution v_i .

On définit aussi un rapport d'hypervolume :

$$RS = \frac{H_S}{H_A}$$

Où H_S est le volume occupé par l'ensemble des solutions non dominées actuel et H_A est le volume occupé par l'ensemble des solutions non dominées théorique.

Plus la valeur de cette métrique est proche de 1, meilleure est la surface de compromis.

1.8.2 la métrique C :

Cette métrique a été introduite par Zitzler [Zitzler, 99]. Soient A et B deux surfaces de compromis, la fonction C suivante transforme les deux surfaces de compromis A et B en un nombre réel compris entre 0 et 1.

C'est une métrique relative qui permet de comparer deux surfaces de compromis A et B

$$C(A, B) = \frac{|\{z'' \in B\} / \exists z' \in A : z' \text{ domine } z''|}{|B|}$$

Ainsi, la valeur $C(A, B) = 1$ signifie que tous les points de B sont dominés ou égaux à ceux de A. et la valeur $C(A, B) = 0$ signifie qu'aucun point de B n'est dominé par un point de A.

La mesure C n'étant pas symétrique, il est nécessaire de considérer les deux valeurs $C(A, B)$ et $C(B, A)$ pour obtenir une mesure plus fiable des deux surfaces de compromis A et B à comparer.

1.8.3 la métrique d'espacement :

Cette métrique a été introduite par Schott [Schott, 95]. Elle permet de mesurer l'uniformité de la répartition des points sur la surface de compromis. Soit A l'ensemble de n solutions non dominées, alors S(A) est définie par :

$$S(A) = \left[\frac{1}{n-1} * \sum_{i=1}^n (\bar{d} - d_i)^2 \right]^{\frac{1}{2}}$$

Avec

$$d_i = \min_j (|f_1^i(\mathbf{x}) - f_1^j(\mathbf{x})| + \dots + |f_m^i(\mathbf{x}) - f_m^j(\mathbf{x})|) \quad \mathbf{i}, \mathbf{j} \in [1, \dots, \mathbf{n}] \text{ et } \mathbf{i} \neq \mathbf{j},$$

m : nombre de fonctions objectif

\bar{d} : moyenne de tous les d_i

Plus le résultat de la mesure est proche de 1, meilleure est la répartition des points sur la surface de compromis.

Nous avons présenté quelques métriques parmi les plus utilisées aujourd'hui, le lecteur est invité à consulter [Collette, 02] pour une présentation d'une variété de métriques.

1.9 Conclusion :

Nous avons présenté dans ce chapitre les concepts de bases de l'optimisation multiobjective, puis une classification des méthodes correspondantes. Les approches agrégatives ainsi que les approches non Pareto semblent peu efficaces, car elles sont très sensibles à la forme du front de Pareto (concavité, non uniformité,...). Par contre les méthodes Pareto issues des métaheuristiques pour l'optimisation multiobjective semblent bien se prêter à ce type de problèmes. Elles ont prouvées leur efficacité pour un grand nombre de problèmes même ceux ayant des fronts de Pareto non convexes et non uniformes. Nous avons aussi présenter trois parmi les métriques les plus utilisées pour mesurer la qualité de l'ensemble des solutions non dominées trouvées.

L'informatique quantique est une nouvelle science caractérisée par un parallélisme massive et une grande diversité dans la représentation des solutions. Comme cette diversité est un atout important pour une méthode d'optimisation. On a choisit d'incorporer les concepts quantique au niveau d'un algorithme génétique multiobjectif pour au chapitre ** donner une nouvelle approche Pareto issue des métaheuristiques multiobjective basée sur les principes de l'informatique quantique.

Références :

- [Barichard, 03] Vincent Barichard, « Approches Hybrides pour les problèmes multiobjectif », , Thèse de Doctorat, université d'Angres, Novembre 2003
- [Benabdel, 97] Ben Abdelaziz F. And Krichen S. (1997), A tabu search heuristic for multiple objective knapsack problems, Ructor Reasearch Report, RR 28-67.
- [Coello and Cruz, 2002]
- [Coello and Landa, 2003]
- [Coello and Lechuga, 2002]
- [Coello, 98] C. A. Coello Coello, An updated survey og G.A. based multiobjective optimization technics, Rapport technique Lania-RD-98-08, Xalapa Veracruz, Mexico, décembre 1998, <http://www.lania.mx/~ccoello/EMOO>.
- [Coello, 99] C. A. Coello Coello, A survey of constraint handling techniques used with evolutionary computation methods, Rapport technique

Lania-RI-99-04, Laboratoire national de l'informatique Avanzada,
Xalapa, Veracruz, Mexico, 1999,
<http://www.lania.mx/~ccoello/EMOO>.

- [Collette, 02] Y. Collette et P. Siarry. Optimisation multiobjectif. Eyrolles, 2002.
- [Corne, 00] D.W. Corne and J. D. Knowles. M-paes: a memetic algorithm for multiobjective optimization. In Proceeding of the 2000 Congress on Evolutionary Computation, pages 325-332, 2000.
- [Cui et al., 2001]
- [Czyzak, 98] P. Czyzak and A.Jaszkiwicz, A Pareto simulated annealing – a metaheuristic for multiple objective combinatorial optimization. Journal of Multi-Criteria Decision Analysis, 7(1):34-47, 1998.
- [Deb, 00] K. Deb, S. Agrawal, A. Pratapand and T. Meyarivan, “A Fast and Elitist multiobjective genetic algorithm: NSGA-II”, Technical Report 20001, Indian Institute of Technology, Kanpur Genetic Algorithms Laboratory(Kan GAL),2000.
- [Deb, 99] K. Deb, An overview of multi-objective evolutionary algorithms, Journée J.E.T., Copie de transparents, mai 1999.
- [Eckart, 01] Eckart Zitzler, Marco Laumanns, and Lothar Thiele, "SPEA2:Improving the Performance of the Strength Pareto evolutionary Algorithm", Technical Report 103, Computer Engineering and Communication Networks lab(Tik), Swiss Federal Institute of Technology(ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, May 2001.
- [Engrand , 98] P. Engrand, X. Mouney, Une méthode originale d'optimisation multiobjectif, Note interne EDF-DER numéro HT-14/97/035/A, mars 1998.
- [Fonseca, 93] C.M. Fonseca, P. J. Fleming, Genetic Algorithms for multiobjective optimization : Formulation, Discussion and Generalization, Proceedings of the Fifth International Conference on Genetic Algorithms, pages 416-423, San Mateo, California, 1993.
- [Gandibleux, 97] Gandibleux X., Mezdaoui N., Freville A. (1997), A multiobjective tabu search procedure to solve combinatorial optimization problems, in : R.Caballero, F. Ruiz, R. Steuer (eds), Advances in

Multiple Objective and Goal Programming. Lecture Notes in Econom

- [Glover , 97] F. Glover, M. Laguna, Tabu search, Kluwer Academic Publishers, 1997.
- [Glover ,86] F. Glover , Future Paths for integer programming and links to artificial intelligence, Comput. and Ops. Res., volume 13, numéro 5, pages 533-549, 1986.
- [Glover ,90] F. Glover , Artificial intelligence, heuristic frameworks and tabu search, Managerial and decision economics, volume 11, pages 365-375, 1990.
- [Glover, 97] F. Glover, M. laguna, Tabu search, Kluwer Academic Publishers, 1997.
- [Goldberg, 87] D.E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In Genetic Algorithms and their applications: Proceedings of the second International Conference on Genetic Algorithms, pages 41-49. Lawrence Erlbaum, 1987.
- [Goldberg, 89] D.E. Goldberg. Genetic algorithms for search, optimization, and machine learning. Reading, MA: Addison-Wesley, 1989.
- [Goldberg, 94] D. Goldberg, Algorithmes génétiques, éditions Addison Wesley, juin 1994.
- [Hansen, 97] M. P. Hansen. Tabu search for multiobjective optimization: Mots. In Proceedings of 13th International conference on MCDM, 1997.
- [Hansen, 98] Hansen M.P.(1998), Metaheuristics for multiple objective combinatorial optimization, Ph.D. Thesis, IMM-PHS-1998-45, Technical University of Danmark, Lyngby.
- [Iredi , 01] S. Iredi , D. Merkle, M. Middendorf, Bi-criterion optimization with multi colony algorithms, in : Proceeding of the First International Conference on Evolutionary Multi-Criterion Optimization. Lecture Notes in Computer Science , Springer, Berlin, 2001.
- [Know, 99] Knowles, J.D. and Corne, D.W., 1999, The Pareto archived evolution strategy: a new baseline algorithm for multiobjective

optimization, in 1999 Congress on Evolutionary Computation , Piscataway, NJ, IEEE Service Center, 98-105.

[Kurapati and
Azarm,
2000]

[Mariano, 99] E. Mariano and E. Morales, 1999, MOAQ and ant-Q algorithm for multiple objective optimization problems, in W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, R. E. Smith (eds), Proceeding of the genetic and evolutionary Computation

[Miettinen, 99] K. M. Miettinen, Nonlinear multiobjective optimization, editions Kluwer academic publisher, 1999.

[Moore and
Champ
man,
1999]

[Ray and Liew,
2002]

[Schott, 95] J. R Schott. Fault tolerant design using single and multicriteria genetic algorithm optimization. Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 1995.

[Serafini, 92] Serafini 1992, Simulated annealing for multiple objective optimization problems, in: Proceeding of the Tenth International Conference on Multiple Criteria Decision Making, Taipei 19-24.07, vol. 1, 87-96.

[Srinivas , 93] N. Srinivas , k, Deb, multiobjective optimization using non-dominated sorting in genetic algorithms, Rapport technique, Department of technical Engineering , Indian Institute of Technology, Kanput, India, 1993.

[TAL, 00] E. G. Talbi. Une taxinomie des métaheuristiques hybrides. Dans ROADEF'2000, 2000.

[TAL, 01] Talbi E., Métaheuristiques pour l'optimisation combinatoire multi-objectif : Etat de l'art, Lille : LIFL, 2001

[Ulungu , 99] E. L. Ulungu , J. Teghem, P. H. Fortemps, D. Tuyttens, MOSA Method: A Tool for Solving Multiobjective Combinatorial

Optimization Problems, Journal of Multicriteria Decision Analysis, Volume 8, numéro 4, pages 221-236, 1999.

[Yoo and
Hajela,
1999]

[Zitzler, 98] E. Zitzler and L. Thiele, An evolutionary algorithm for multiobjective optimization: the Strength Pareto Approach. Technical report, Swiss Federal Institute of technology (Zurich), 1998.

[Zitzler, 99] E. Zitzler, K. Deb and L. Thiele, Comparison of multiobjective evolutionary algorithms: Empirical results, Technique 70, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of technology (ETH) Zurich, December 1999.

Chapitre 1

Etat de l'art de l'optimisation multiobjectif

Dans ce chapitre, nous présentons les principes de base de l'optimisation multiobjectif . Nous donnerons aussi une classification des méthodes d'optimisation multiobjectif. Puis on présentera quelques une de ces méthodes. On finira par définir quelques métriques permettant l'évaluation de ces méthodes.

Sommaire

1.1	Introduction
1.2	Définitions et principaux concepts en optimisation Mono-objectif
1.3	Une hiérarchie des méthodes d'optimisation
1.4	L'optimisation multiobjectif
1.5	Domaines d'applications
1.6	Les méthodes de résolution des problèmes multiobjectif
1.7	Les Métaheuristiques et l'optimisation multiobjectif :
1.8	Evaluation des méthodes d'optimisation multiobjectif
1.9	Conclusion

1.1 Introduction :

La naissance de l'optimisation multi-objectif remonte à un ouvrage de W. Pareto 1906 sur l'économie politique, dans lequel l'auteur définit pour la première fois ce qu'est un optimum multi-objectif.

L'application de l'optimisation multi-objectif aux problèmes d'ingénierie remonte, aux alentours de la seconde guerre mondiale, elle est longtemps restée une science « anecdotique » à cause de son aspect hermétique, dû à la nécessité de maîtriser un bagage mathématique théorique assez important. Un changement radical est apparu récemment avec les métaheuristique (recuit simulé, algorithmes génétiques, etc.) qui sont des méthodes d'optimisation d'un abord plus facile.

Deux approches de l'optimisation multiobjective s'affrontent :

- Résolution vectorielle du problème : on ne modifie pas l'expression du problème
- Transformation du problème en un problème d'optimisation mono-objectif.

1.2 Définitions et principaux concepts en optimisation Mono-objectif:

1.2.1 Qu'est ce qu'un problème d'optimisation :

Un problème d'optimisation se définit comme la recherche du minimum ou du maximum (l'optimum) d'une fonction donnée.

Les variables de cette fonction sont souvent contraintes d'évoluer dans une certaine partie de l'espace de recherche. On aura donc un problème d'optimisation sous contraintes.

Mathématiquement, un problème d'optimisation s'écrit :

Minimiser	$f(\vec{x})$	(fonctions à optimiser)
Avec	$\vec{g}(\vec{x}) \leq 0$	(m contraintes d'inégalités)
Et	$\vec{h}(\vec{x}) = 0$	(p contraintes d'égalités)
Où	$\vec{x} \in \mathfrak{R}^n$, $\vec{g}(\vec{x}) \in \mathfrak{R}^m$ et $\vec{h}(\vec{x}) \in \mathfrak{R}^p$	

1.2.2 Vocabulaire et définitions :

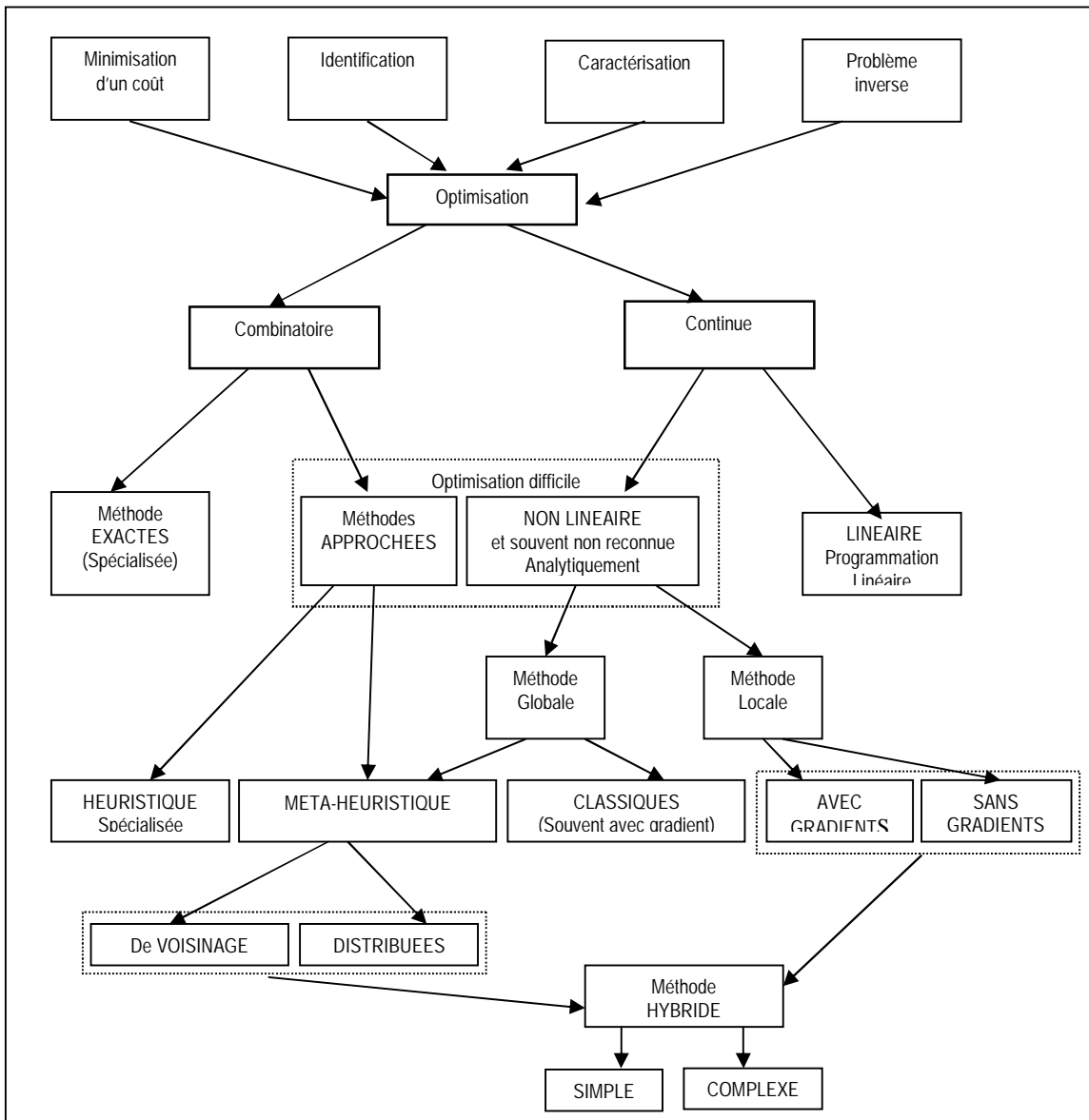
- **Fonction objectif** : c'est la fonction f qu'on cherche à optimiser.
- **Variables de décision** : elles sont regroupées dans le vecteur \vec{x} qui correspond à l'ensemble des variables du problème.
- **Minimum globale** : un « point » \vec{x}^* est appelé minimum global de la fonction f si : $\forall \vec{x}, \vec{x} \neq \vec{x}^* \Rightarrow f(\vec{x}^*) < f(\vec{x})$
- **Minimum local fort** : \vec{x}^* est un minimum local fort de la fonction f si : $\forall \vec{x} \in \nu(\vec{x}^*), \vec{x} \neq \vec{x}^* \Rightarrow f(\vec{x}^*) < f(\vec{x})$, Où $\nu(\vec{x}^*)$ représente le voisinage de \vec{x}^*
- **Minimum local faible** : \vec{x}^* est un minimum local faible de la fonction f si : $\forall \vec{x} \in \nu(\vec{x}^*), \vec{x} \neq \vec{x}^* \Rightarrow f(\vec{x}^*) \leq f(\vec{x})$

1.2.3 **Classification des problèmes d'optimisation :**

Cette classification se fait selon :

- Le nombre de variable de décision : problème monovarié ou multivarié
- Le type de la variable de décision :
 - Nombre réel continu ⇒ problème continu
 - Nombre entier ⇒ nombre discret
 - Permutation sur un ensemble fini de nombres ⇒ combinatoire

1.3 **Une hiérarchie des méthodes d'optimisation : tirée de [Collette, 02]**



1.4 **L'optimisation multi-objectif :**

L'optimisation mono-objectif consiste à maximiser (ou minimiser) une seule fonction objectif par rapport à un ensemble de paramètres. Or, la plupart des problèmes d'optimisation réels

sont décrits à l'aide de plusieurs objectifs souvent contradictoires devant être optimisés simultanément. La solution optimale correspond alors aux meilleurs compromis possibles permettant de résoudre le problème. Il s'agit de l'optimisation multi-objectif, qui fournit aux décideurs un ensemble de solutions optimales dit Front de Pareto

Mathématiquement, un problème d'optimisation multi-objectif (PMO) est décrit comme suit:

Minimiser	$\vec{f}(\vec{x})$	(k fonctions à optimiser)
Avec	$\vec{g}(\vec{x}) \leq 0$	(m contraintes d'inégalités)
Et	$\vec{h}(\vec{x}) = 0$	(p contraintes d'égalités)
Où	$\vec{x} \in \mathcal{R}^n, \vec{g}(\vec{x}) \in \mathcal{R}^m, \vec{h}(\vec{x}) \in \mathcal{R}^p$ et $\vec{f}(\vec{x}) \in \mathcal{R}^k$	

La résolution de ce problème consiste à minimiser *au mieux* ces k fonctions objectif. Donc le but de l'optimisation multi-objectif est de minimiser un groupe de fonctions objectif sans trop dégrader les valeurs des optima obtenus par rapport à ceux obtenus lors d'une optimisation mono-objectif effectuée objectif par objectif.

Sauf mention contraire explicite, tous les énoncés et définitions seront donnés dans le cadre de problèmes de minimisation. En effet un problème de maximisation peut être aisément transformé en problème de minimisation en considérant l'équivalence suivante :

$$\text{Maximiser } f(\vec{x}) \Leftrightarrow \text{Minimiser } -f(\vec{x})$$

1.4.1 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)

Si on prend l'exemple de dimensionnement d'une poutre devant supporter une charge donnée, on voudra obtenir une poutre de section la plus petite possible, produisant la plus petite déformation possible, lorsque la charge repose au milieu de la poutre [Coello, 99]

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*.

1.4.2 La dominance :

Comme la solution optimale est une multitude de points de \mathcal{R}^n , il est vital pour identifier ces meilleurs compromis de définir une relation d'ordre entre ces éléments. Dans le cas des PMO, cette relation est appelée relation de dominance.

La relation de dominance nous permet donc de restreindre l'ensemble des solutions de compromis. Il existe différents types de relations de dominance [Collette, 02], qui permettent d'avoir suffisamment de degrés de liberté dans le choix de la relation qui reproduit au mieux le comportement d'un décideur.

Mais la plus célèbre et la plus utilisée est *la dominance au sens de Pareto*. Elle est définie par :

Le vecteur \vec{u} domine le vecteur \vec{v} (on note : $\vec{u} \mathbf{p} \vec{v}$) si :

\vec{u} est au moins aussi bon que \vec{v} dans tous les objectifs, et,

\vec{u} est strictement meilleur que \vec{v} dans au moins un objectif.

Afin de définir formellement la notion de dominance au sens de Pareto, les relations $=$, \leq et $<$ usuelles sont étendues aux vecteurs.

Définition 1 Soient \vec{u} et \vec{v} deux vecteurs de même dimension, $\vec{u} = (u_1, u_2, \dots, u_m)$ et $\vec{v} = (v_1, v_2, \dots, v_m)$

$$\begin{aligned} \vec{u} &= \vec{v} & \text{ssi} & \quad \forall i \in \{1, 2, \dots, m\}, u_i = v_i \\ \vec{u} &\leq \vec{v} & \text{ssi} & \quad \forall i \in \{1, 2, \dots, m\}, u_i \leq v_i \\ \vec{u} &\leq \vec{v} & \text{ssi} & \quad \vec{u} \leq \vec{v} \quad \wedge \quad \vec{u} \neq \vec{v} \end{aligned}$$

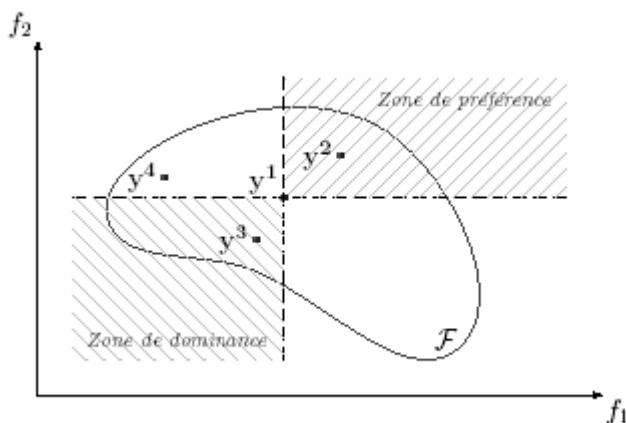
Les relations $>$ et \geq sont définies de manière analogue.

On remarque que les relations qu'on vient de définir ne couvrent pas tous les cas possibles. Par exemple les point (ici vecteurs) $\vec{u} = (3, 5)$ et $\vec{v} = (6, 1)$ sont incomparables à l'aide de ces relations. Nous définissons maintenant la relation de dominance au sens de Pareto qui permet de comparer tous les vecteurs de décision possible.

Définition 2 La dominance au sens de Pareto : Soient \vec{u} et \vec{v} deux vecteurs de décision, $\vec{u} = (u_1, u_2, \dots, u_m)$ et $\vec{v} = (v_1, v_2, \dots, v_m)$,

$$\begin{aligned} \vec{u} &\mathbf{p} \vec{v} \quad (\vec{u} \text{ domine } \vec{v}) & \text{ssi} & \quad \vec{f}(\vec{u}) < \vec{f}(\vec{v}) \\ \vec{u} &\mathbf{p} \vec{v} \quad (\vec{u} \text{ domine faiblement } \vec{v}) & \text{ssi} & \quad \vec{f}(\vec{u}) \leq \vec{f}(\vec{v}) \\ \vec{u} &\sim \vec{v} \quad (\vec{u} \text{ et } \vec{v} \text{ incomparables ou non dominés}) & \text{ssi} & \quad \text{not}(\vec{f}(\vec{u}) \leq \vec{f}(\vec{v})) \wedge \text{not}(\vec{f}(\vec{v}) \leq \vec{f}(\vec{u})) \end{aligned}$$

Ces relations sont définies pour un problème de minimisation.



- § $\vec{z}^i = \vec{f}(\vec{x}^i)$,
- § \vec{x}^1 domine \vec{x}^2 ,
- § \vec{x}^1 est dominé par \vec{x}^3
- § et \vec{x}^1 est non dominé avec \vec{x}^4

Fig. 1 Exemple de dominance

Définition 3 L'optimalité globale au sens de Pareto : Soit F l'image de l'ensemble réalisable X dans l'espace des objectifs. Un vecteur de décision $\vec{u} \in X$ est dit Pareto globalement optimal si et seulement si : $\neg \exists \vec{v} \in X, \vec{v} \mathbf{p} \vec{u}$. Dans ce cas $\vec{f}(\vec{u}) \in F$ est appelé : **solution efficace**.

Définition 4 L'optimalité locale au sens de Pareto : un vecteur de décision $\vec{u} \in X$ est dit Pareto localement optimal si et seulement si, pour un $d > 0$ fixé: $\neg \exists \vec{v} \in X, \vec{f}(\vec{v}) \in B(\vec{f}(\vec{u}), d)$ et $\vec{v} \mathbf{p} \vec{u}$, où $B(\vec{f}(\vec{u}), d)$ représente une boule de centre $\vec{f}(\vec{u})$ et de rayon d .

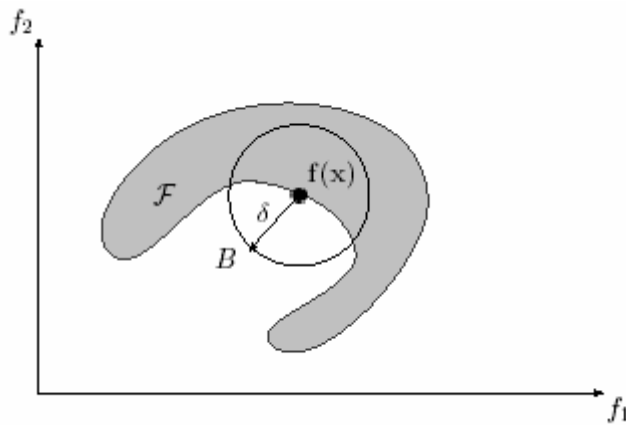


Fig. 2 Optimalité Locale

Définition 5 une solution $\vec{u} \in X$ est dite non dominée par rapport à un ensemble $Y \subseteq X$ si et seulement si :

$$\neg \exists \vec{v} \in Y, \vec{v} \mathbf{p} \vec{u}$$

Définition 6 Ensemble des solutions non dominées : L'ensemble des solutions non dominées de X , est défini par l'ensemble

$$ND(X) = \{ \vec{u} \in X \mid \vec{u} \text{ est non dominé par rapport à } X \}$$

Définition 7 Front de Pareto : Soit F l'image de l'ensemble réalisable X dans l'espace des objectifs. Le Front de Pareto $ND(F)$ de F est défini comme suit :

$$ND(F) = \{ \vec{v} \in F \mid \neg \exists \vec{u} \in F, \vec{u} < \vec{v} \}$$

Le Front de Pareto est aussi appelé l'**ensemble des solutions efficaces** ou la **surface de compromis**.

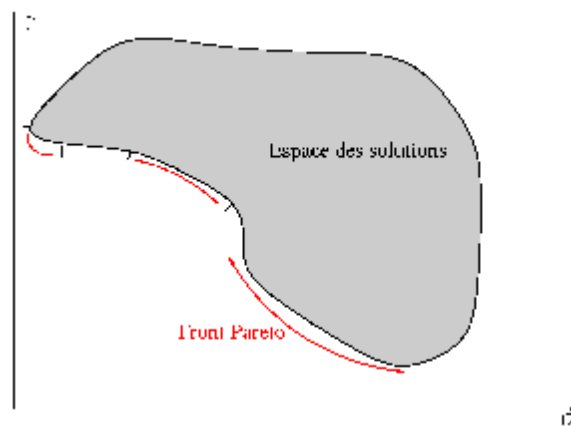


Fig.3 Front de Pareto

Algorithme d'Assignment du rang de Pareto

N : nombre de points de l'ensemble sur lequel on effectue les comparaisons

x_i : Un individu

t : la génération t

$Rang(x_i, t)$ Numéro de la surface de compromis auquel est affecté l'individu X_i à la génération t

RangCourant = 1

$m = N$

While $N \neq 0$ do

 For $i=1$ to m do

 If x_i est non dominé

$Rang(x_i, t) = RangCourant$

 End for i

 For $i=1$ to m do

 If $Rang(x_i, t) = RangCourant$

 Ranger x_i dans la population temporaire

$N=N-1$

 End for i

 RangCourant = RangCourant + 1

$m=N$

End while

Définition 7 Point Idéal : les coordonnées du point idéal (p_i) correspondent aux meilleures valeurs de chaque objectif des points du front de Pareto. les coordonnées de ce point correspondent aussi aux valeurs obtenues en optimisant chaque fonction objectif séparément.

$$P_i = \min \{ y_i \mid \vec{y} \in ND(F) \}$$

Définition 8 Point Nadir : les coordonnées du point Nadir (p_n) correspondent aux pires valeurs de chaque objectif des points du front de Pareto.

$$P_n = \max \{ y_i \mid \vec{y} \in ND(F) \}$$

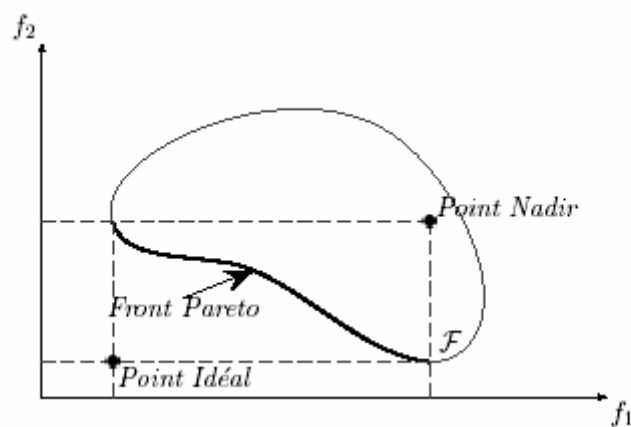


fig.4 Point Nadir et point Idéal

Définition 9 La connexité : un ensemble A est connexe, si et seulement si l'équivalence suivante est vérifiée :

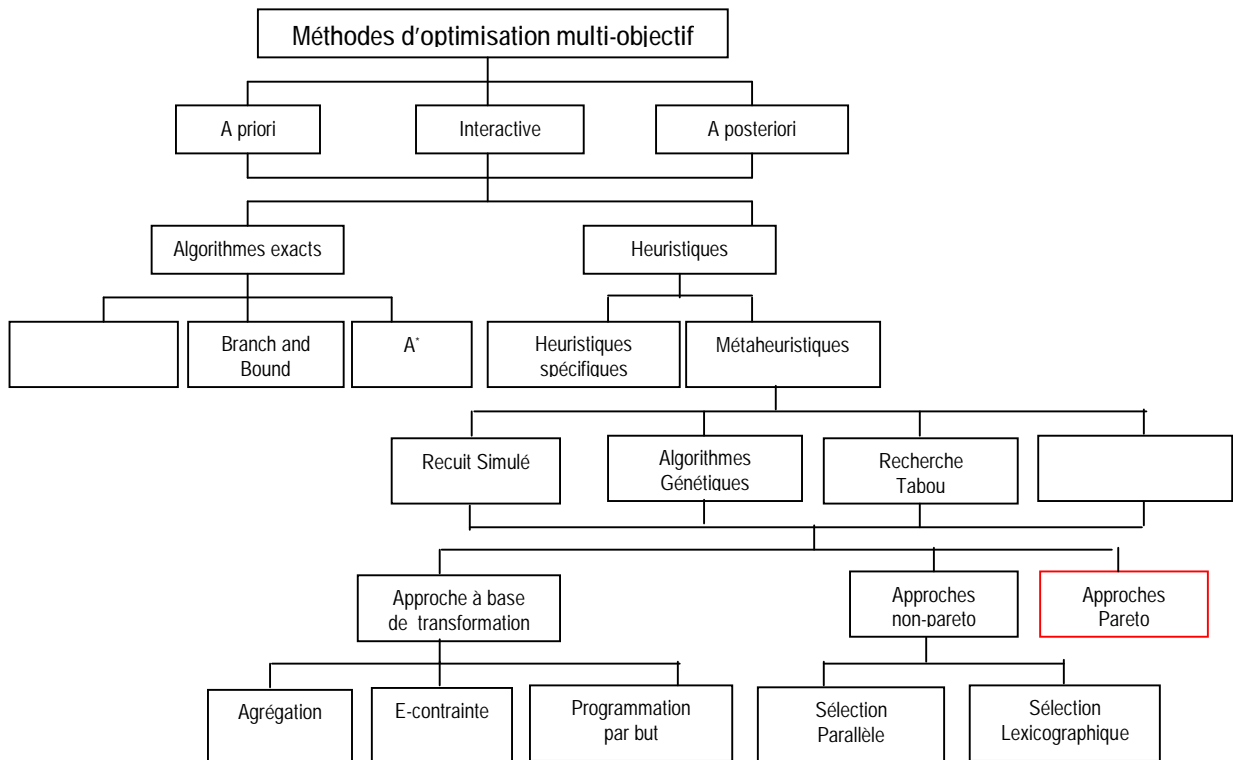
$$x \in A \wedge y \in A \Leftrightarrow \text{Segment}(x, y) \subset A$$

La convexité est le premier indicateur de la difficulté du problème. Plusieurs méthodes d'optimisation sont incapables de résoudre d'une façon optimale des problèmes non convexes.

1.5 Domaines d'applications :

1.6 Les méthodes de résolution des problèmes multi-objectif :

Les méthodes d'optimisation multicritères s'intéressent aux problèmes essayant d'optimiser simultanément plusieurs objectifs. Elles peuvent être classées selon le schéma suivant [TAL, 01] :



Selon le mode de coopération entre le solveur et le décideur, elles peuvent être rangées en trois familles :

- Les méthodes à préférence à priori :

L'utilisateur définit le compromis qu'il désire réaliser avant de lancer la méthode d'optimisation. On retrouve dans cette famille, toutes les méthodes agrégatives.

- Les méthodes à préférence progressives :

L'utilisateur affine son choix de compromis au fur et à mesure du déroulement de l'optimisation. On retrouve dans cette famille, toutes les méthodes interactives.

- Les méthodes à préférence à posteriori :

L'utilisateur choisit une solution de compromis parmi les solutions de la surface de compromis.

Les algorithmes de résolutions des PMOs se répartissent en deux grandes catégories :

A. LES METHODES EXACTES :

Ces méthodes permettent d'obtenir l'optimum global de manière exacte pour certains types de problèmes. Elles sont efficaces seulement lorsque le PMO à résoudre est bi-objectif et l'espace de recherche est de petite taille.

B. LES HEURISTIQUES :

a. Heuristiques spécifiques :

Elles sont dédiées à la résolution d'un PMO spécifique et ne fonctionnent que sur ce type de problèmes.

b. Métaheuristiques :

Elles sont fondées sur une idée générale. Et peuvent s'adapter à n'importe quel type de problème et donner de bons résultats. On trouve dans cette classe les algorithmes génétiques, la recherche tabou, le recuit simulé,...

Les métaheuristiques résolvant les PMO peuvent être classées en trois approches :

b.1 Transformation des PMO en uni-objectif :

b.1.1 Méthodes d'agrégation :

Les méthodes agrégatives fusionnent les différentes fonctions objectives pour se ramener à un problème d'optimisation mono-objectif.

Parmi ces méthodes on trouve La méthode de pondération des fonctions objectif [Coello, 98], qui est la plus simple des méthodes d'optimisation multi-objectif. Elle est d'ailleurs appelée « l'approche naïve ». La transformation que l'on effectue est la suivante :

$$\begin{array}{l} \text{Minimiser } \mathbf{f}(\vec{x}) \\ \text{Avec } \vec{x} \in S_r \end{array} \Rightarrow \begin{cases} \text{minimiser } \sum_{i=1}^k w_i \cdot f_i(\vec{x}) \\ \text{Avec } \vec{x} \in S_r, w_i \geq 0 \text{ et } \sum_{i=1}^k w_i = 1 \end{cases}$$

Où S_r représente le domaine réalisable.

On peut définir des préférences vis à vis de certaines fonctions objectives en jouant sur les coefficients de pondération.

Discussion :

Cette méthode est très efficace point de vue algorithmique, mais elle ne permet pas de trouver les solutions enfermées dans des concavités.

b.1.2 Méthode ϵ -contrainte :

Ce n'est pas une méthode d'agrégation des fonctions objectives. Elle est aussi dite méthode du compromis [Miettinen, 99]. Elle transforme un PMO en un problème d'optimisation mono-objectif de la façon suivante :

- Choisir un objectif à optimiser prioritairement,
- Choisir un vecteur de contraintes initiales (\vec{e}),
- transformer le problème en gardant l'objectif prioritaire et en transformant les autres objectifs en contraintes d'inégalités comme suit :

$$\begin{array}{l} \text{Minimiser } \vec{f}(\vec{x}) \\ \text{Avec } \vec{x} \in S_r \end{array} \rightarrow \begin{array}{l} \text{minimiser } \vec{f}_1(\vec{x}) \\ \text{avec } \vec{f}_2(\vec{x}) \leq \epsilon_2, \dots, \vec{f}_n(\vec{x}) \leq \epsilon_n \\ \vec{x} \in S_r \end{array}$$

Discussion :

Cette méthode présente plusieurs inconvénients à savoir :

- la formulation des préférences utilisateur est délicate et nécessite une connaissance approfondie du problème de départ.
- Les contraintes rajoutées compliquent la résolution du problème.

b.1.3 Programmation par but :

On définit un ensemble de buts qu'on espère atteindre pour chaque fonction objectif. L'algorithme tente de minimiser l'écart entre la solution courante et ses buts.

Discussion :

Cette méthode est efficace et facile à implémenter mais la définition des buts à atteindre est souvent coûteuse. Pour plus de détails, voir [Miettinen, 99], [Coello, 98].

b.2 Approches non Pareto :

Ces approches traitent séparément les différents objectifs, elles sont sensibles au paysage du front de Pareto (convexité, continuité,...). Elles sont efficaces et faciles à implémenter seulement pour les PMO avec un nombre réduit d'objectifs.

b.2.1 Vector Evaluated Genetic Algorithm (V.E.G.A) :

Cette méthode est décrite dans [Coello, 98] et [Deb, 99]. elle opère comme suit :

Etape 1 : Itération i. initialisation d'une population de taille N.

Etape 2 : Création de k groupes composé chacun de N/k individus. K étant le nombre de fonctions objectif.

Etape 3 : Calcul des efficacités.
Mélange des individus.

Etape 4 : on applique l'algorithme génétique classique (croisement –mutation- sélection) puis on passe à l'itération i+1

Discussion :

Le danger de cette méthode est d'obtenir, en fin d'optimisation, une population d'individus moyens dans tous les objectifs.

b.2.2 L'ordonnancement lexicographique :

C'est une méthode très intuitive qui consiste à considérer les fonctions objectif les unes après les autres et à minimiser à chaque fois un problème mono-objectif, en complétant au fur et à mesure l'ensemble des contraintes [Coello, 98].

Cette méthode procède en k étapes :

§ Etape 1 :

$$\left| \begin{array}{l} \text{Minimiser } f_1(\vec{x}) \\ \text{Avec } \quad x \in S_r \end{array} \right.$$

on note f_1^* la solution de ce problème.

§ Etape2 :

$$\left| \begin{array}{l} \text{Minimiser } f_2(\vec{x}) \\ f_1(\vec{x}) = f_1^* \\ \text{Avec } \quad x \in S_r \end{array} \right.$$

On note f_2^* la solution de ce problème.

...

§ Etape k :

$$\left| \begin{array}{l} \text{Minimiser } f_k(\vec{x}) \\ f_1(\vec{x}) = f_1^*, f_2(\vec{x}) = f_2^*, \dots, f_{k-1}(\vec{x}) = f_{k-1}^* \\ \text{Avec } \quad x \in S_r \end{array} \right.$$

Discussion :

L'inconvénient majeur de cette méthode est qu'elle requiert un choix de la séquence des objectifs à minimiser. Deux choix différents génèrent deux solutions distinctes.

b.3 Approches Pareto :

Ces approches utilisent la notion de dominance lors de la sélection des solutions. Elles sont capables de générer des solutions Pareto situées dans des concavités. Historiquement, on peut considérer l'existence de deux générations des MOEAs.

La première génération des approches Pareto est caractérisée par l'utilisation :

- § d'un mécanisme de sélection Pareto. En effet la relation de dominance permet d'affecter des rangs aux individus de la population, ce qui fait apparaître la notion de front. C'est la technique de *ranking* [Goldberg, 89].
- § du *sharing (niching)* qui consiste à pénaliser la valeur d'adaptation en fonction du nombre d'individus au voisinage du regroupement cette technique introduite dans [Goldberg, 87] est largement utilisée maintenant.

Les algorithmes correspondants sont relativement simples. Parmi les approches les plus représentatives de cette génération on peut citer :

- § Multi-Objective Genetic Algorithm (M.O.G.A)
- § N.S.G.A (Non dominated Sorting Genetic Algorithm)
- § N.P.G.A (Niched-Pareto Genetic Algorithm)

Après environs dix années de succès, cette génération devient dépassée. La deuxième génération des approches Pareto est maintenant considérée comme l'état d'art de l'optimisation multi-objectif. Ce sont en général des approches élitistes manipulant une population secondaire externe. Parmi les approches les plus représentatives de cette génération on peut citer :

- § SPEA et SPEA2 (Strength Pareto Evolutionary Algorithm)
- § NSGA-II (Non dominated Sorting Genetic Algorithm)
- § PAES (The Pareto Archived Evolution Strategy)
- § PESA et PESA II (The Pareto Envelope-based Selection Algorithm)
- § MOMGA et MOMGA II (Multi-objective Messy Genetic Algorithm)
- § MicroGA pour l'Optimisation multi-objectif.

Comme les métaheuristiques ont permis la réalisation de méthodes de résolutions très performantes, On a préféré les aborder avec un peu plus de détails dans la section qui suit..

1.7 Les Métaheuristiques et l'optimisation multi-objectif :

Les métaheuristiques sont des méthodes générales de recherche dédiées aux problèmes d'optimisation difficile. Elles reprennent des idées que l'on trouve parfois dans la vie courante. Les principales heuristiques sont : le recuit simulé, la recherche tabou et les algorithmes génétiques.

Les métaheuristiques se répartissent en trois familles :

- Les méthodes déterministes de recherche d'optimum local :

Ces méthodes convergent rapidement, mais, la plus part du temps, elles se contentent de trouver l'optimum local.

- Les méthodes déterministes de recherche d'optimum global:

Ces méthodes permettent de trouver un optimum global et ne reposent pas sur un processus stochastique.

- Les méthodes stochastiques de recherche d'optimum global:

Ces méthodes reposent sur un processus stochastique pour la recherche de l'optimum global. Elles sont moins rapides que les méthodes déterministes mais elles peuvent trouver un optimum global difficile à atteindre.

Dans la suite, on ne s'intéressera qu'à cette catégorie de méthodes.

1.7.1 Le recuit simulé :

Le recuit simulé trouve ses origines dans la thermodynamique. Cette méthode est issue d'une analogie entre le phénomène physique de refroidissement lent d'un corps en fusion, qui le conduit à un état solide, de basse énergie. Il faut abaisser la température pour que le corps atteigne l'équilibre thermodynamique.

Ce processus métallurgique a été transposé à l'optimisation et a donné une méthode simple et efficace.

Algorithme recuit simulé

```

Init T                                (température initiale)
Init x                                (point de départ choisi au hasard)
Init Δt                               (température minimale)
While ( not(end))                    ( s'arrêter si température atteint un seuil minimal)
    Y = voisin (x)
    Δc=c(y) -c(x)
    if Δc < 0 then y = x                ( y: nouveau point de départ )
    else if alea (0,1) < e-Δc/T then y =x ( avec une probabilité = e-Δc/T)
    T = α(T)                           (diminuer la température par palier)
If T < Δt then end (while)
Repeat (while)

```

Pour appliquer le recuit simulé aux MOPs, il suffit de déterminer comment calculer la probabilité d'acceptation d'un individu \underline{x} . La première version du recuit simulé multiobjectif a été proposée par Serafini [Serafini, 92] qui a proposé un ensemble de règles d'acceptation. Dans le cas multiobjectif, trois situations peuvent apparaître en comparant la nouvelle solution \underline{x}' avec la solution courante \underline{x} :

- § \underline{x}' domine \underline{x}
- § \underline{x}' est dominé par \underline{x}
- § \underline{x}' et \underline{x} sont non dominées

Dans le premier cas la probabilité d'acceptation est naturellement égale à un, dans le second cas elle est inférieure à un, mais elle reste indéfinie (inconnue) dans le troisième cas. Actuellement, il existe plusieurs versions du recuit simulé multiobjectif. Parmi lesquels, on va présenter les méthodes MOGA et PASA.

1.7.1.1 La méthode M.O.S.A : (Multiple Objective Simulated Annealing)

Cette méthode a été proposée dans [Ulungu , 99]. Elle utilise le recuit simulé pour rechercher la surface de compromis.

On commence par définir une suite de fonctions donnant la probabilité d'acceptation d'une mauvaise solution, pour chaque fonction objectif :

$$\Pi_k = \begin{cases} \exp(-\Delta f_k / T_n) & \text{si } \Delta f_k > 0 \\ 1 & \text{si } \Delta f_k \leq 0 \end{cases}$$

Avec

$$\Delta f_k = f_k(\mathbf{y}) - f_k(\mathbf{x}_n).$$

T_n : la température du recuit simulé à l'itération n,

f_k : k^{ième} fonction objectif,

\mathbf{x}_n : Solution obtenue à l'itération n,

\mathbf{y} : Point considéré lors de l'itération.

Ensuite on agrège ces probabilités. Il existe plusieurs possibilités d'agrégations :

- Produit des probabilités :

$$t(p, l) = \prod_{t=1}^N (p_t)^{\lambda_k}$$

où λ_k correspond à un coefficient de pondération relatif à la k^{ième} fonction objectif

- La plus petite probabilité :

$$t(\Pi, \lambda) = \min_{k=1, \dots, N} (\pi_k)^{\lambda_k}$$

On agrège, maintenant, les fonctions objectif de la façon suivante :

$$f_{eq}(\mathbf{x}) = \sum_{i=1}^n w_i \cdot f_i(\mathbf{x})$$

et $\Delta f_{eq}(\mathbf{x}) = f_{eq}(\mathbf{x}_{n+1}) - f_{eq}(\mathbf{x})$

Si $\Delta f_{eq} \leq 0$ alors $\mathbf{x}_{n+1} = \mathbf{y}$

Si $\Delta f_{eq} > 0$ alors $\mathbf{x}_{n+1} = \mathbf{y}$ avec la probabilité p et $\mathbf{x}_{n+1} = \mathbf{x}_n$ avec la probabilité (1-p)

Cette méthode diffère de la méthode de pondération des fonctions objectif couplée à un recuit simulé par la formule de calcul de la probabilité d'acceptation.

C'est cette agrégation de probabilités individuelles qui a rendu le recuit simulé plus adapté à l'optimisation multi-objectif.

Discussion :

Le couplage de la méthode de pondération des objectifs avec le recuit simulé a surmonté les limitations de la méthode.

Des travaux on montré que cette nouvelle méthode est plus efficace que la méthode d'origine. L'effet stochastique a permis d'explorer des zones qui étaient auparavant inaccessibles (concavité)

1.7.1.2 **La méthode P.A.S.A :** (Pareto Archived Simulated Annealing)

Cette méthode, développée à EDF [Engrand , 98], utilise une agrégation des fonctions objectif couplée avec un système d'archivage de solutions non dominées.

On suppose que les fonctions objectif sont positives. On les agrège de la façon suivante :

$$G(\mathbf{x}) = \sum_{i=1}^n Ln(f_i(\mathbf{x}))$$

On calcule $\Delta(G) = G(\mathbf{x}') - G(\mathbf{x}) = \sum_{i=1}^n Ln(f_i(\mathbf{x}') / f_i(\mathbf{x}))$

On définit la probabilité d'acceptation p : $p = e^{-\Delta G/T}$

Algorithme de la méthode P.A.S.A

1. Choix de P_0 , une population d'archive initiale, éventuellement réduite à un seul élément x_0 , et choix de T_q initiale.
 2. Soit \mathbf{x}'_n un voisin de \mathbf{x}_n ,
 3. calcul de $G(\mathbf{x}'_n)$ et $\Delta(G) = G(\mathbf{x}') - G(\mathbf{x})$
 4. si $\Delta G < 0$ alors $\mathbf{x}_{n+1} = \mathbf{x}'_n$ avec la probabilité p
 5. si $\Delta G > 0$ alors soit $p = e^{-\Delta G/T_q}$
 $\mathbf{x}_{n+1} = \mathbf{x}'_n$ avec la probabilité p
 $\mathbf{x}_{n+1} = \mathbf{x}_n$ avec la probabilité (1-p)
 6. si on sort de la boucle interne (rechercher un autre voisin de \mathbf{x}_n), alors
 $T_{q+1} = \alpha . T_q$ avec $\alpha < 1$
 7. Principe de non-dominance pour l'archivage de \mathbf{x}_{n+1}
 - si \mathbf{x}_{n+1} domine une des solutions archivées \mathbf{y}_p , \mathbf{x}_{n+1} remplace \mathbf{y}_p dans la population archivées ;
 - Si \mathbf{x}_{n+1} est dominé par au moins une des solutions archivées, alors on ne l'archive pas ;
 - Si \mathbf{x}_{n+1} n'est pas dominé par la population archivée alors on l'ajoute à la population archivée, et on retire de celle-ci les solutions dominées par \mathbf{x}_{n+1} .
 8. return to base : lorsque c'est nécessaire, pour que la recherche se fasse dans toute la surface de compromis,
 $\mathbf{x}_{n+1} = \mathbf{y}_i$ (i choisit aléatoirement dans la population archivée)
 9. si (non convergence) alors retour en 2.
-

Discussion :

Dans cette méthode, la population initiale P_0 peut être issue d'une première optimisation multiobjectif. Ceci améliorera la qualité des résultats obtenus.

On peut aussi inclure un « savoir ingénieur » pour la gestion de l'archive.

Malheureusement, cette méthode ne supporte que les valeurs positives de fonctions objectives.

1.7.2 La recherche Tabou :

La recherche Tabou est une métaheuristique développée par Glover ([Glover ,86], [Glover ,90], [Glover, 97]). Elle est basée sur des idées simples, mais elle est néanmoins très efficace. Elle combine une procédure de recherche locale avec un certain nombre de règles et de mécanismes pour surmonter l'obstacle des optima locaux, tout en évitant de cycler. Elle a été appliquée avec succès pour résoudre de nombreux problèmes difficiles d'optimisation combinatoire.

Le principe de la méthode Tabou est simple : On part d'une solution initiale x , puis on cherche le meilleur voisin $s(x)$ dans le voisinage de x [$s(x)$], l'originalité de la méthode réside dans le fait que l'on retient le meilleur voisin de x même si celui-ci est plus mauvais que x .

Ce critère autorisant les dégradations de la fonction objectif évite à l'algorithme d'être piégé dans un minimum local, mais il induit un risque de cyclage. Pour régler ce problème, l'algorithme mémorise la liste des transformations interdites (telle que de $s(x)$ à x) dans la liste Tabou interdisant la transformation inverse d'une transformation faite récemment.

Algorithme Recherche Tabou

```

Init L = f (liste des points tabous)
Init x (point de départ)
Init kfin (nb d'itérations max total)
Init k = 0 et l = 0
While (not(end))
    y = voisinage(x) – L (on prend un voisinage sans points tabous)
    ΔC = C(y) – C(x)
    x = y avec y ∈ voisinage(x) – L tel que min C(y) – C(x)
    L = L + {x}
    k = k + 1
    Update L (on retire des mouvements tabou de L suivant des critères d'aspiration)
    If k = kfin then end (while)
Repeat

```

Plusieurs versions multiobjectif de la méthode Tabou ont été proposées ([Gandibleux, 97], [Hansen, 98], [Benabdel, 97], [Barichard, 03])

1.7.3 Les algorithmes génétiques :

Les algorithmes génétiques s'inspirent du fonctionnement de l'évolution naturelle, notamment la sélection de Darwin et la procréation selon les règles de Mendel [Goldberg, 94].

La sélection naturelle favorise les individus d'une population qui sont le mieux adaptés à un environnement. La sélection est suivie de la procréation, réalisé à l'aide de croisement et de mutation au niveau du patrimoine génétique des individus (« génotype »). Ainsi deux individus parents, qui se croisent, transmettent une partie de leur patrimoine génétique à leurs descendants.

Le génotype de l'enfant fait que celui-ci est plus au mois adapté à l'environnement. S'il est bien adapté, il a une grande chance de procréer dans la génération future.

- Vocabulaire utilisé :

- ~ Génotype ou chromosome: individu
- ~ Individu: codage sous forme de gènes d'une solution potentielle à un problème d'optimisation.
- ~ Gène: un chromosome est composé de gènes. Dans le codage binaire, un gène vaut 0 ou 1
- ~ Phénotype: c'est la valeur de la solution représentée par un génotype.
- ~ Efficacité ou adaptation: correspond à la performance d'un individu dans la résolution d'un problème donné. Par exemple, si l'on considère un problème de maximisation d'une fonction f , l'efficacité de l'individu croîtra avec sa capacité à maximiser cette fonction.

F(x)	10	20	5
Efficacité	2	4	1
Individu	1	2	3

- **Principe des algorithmes génétiques :**

On démarre avec une population initiale. Après avoir déterminé l'efficacité de chacun de ses individus, on opère une reproduction. On copie les individus proportionnellement à leur efficacité. Par exemple, l'individu 2 de l'exemple précédent est recopié quatre fois. On va sélectionner les individus suivant leur efficacités, puis les fusionner avec d'autres individus choisis aléatoirement. Cette opération s'appelle « croisement » et consiste à :

- Sélectionner aléatoirement une position de croisement
- A cette position, on sépare le codage puis on échange les morceaux de droite entre les deux individus.

Exemple de croisement :

Individu 1 :

0	1	1	1	0
---	---	---	---	---

Individu 2 :

0	0	0	1	1
---	---	---	---	---

Position de croisement = 3

Résultat :

0	1	1	1	1
---	---	---	---	---

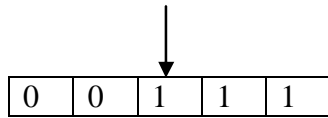
0	0	0	1	0
---	---	---	---	---

Après cette opération de croisement, la population est diversifiée, on sélectionne, au hasard, un certain nombre d'individus auxquels on applique des mutations. La « mutation » consiste à inverser un des gènes de l'individu.

Exemple de mutation :

Individu :

0	0	0	1	1
---	---	---	---	---



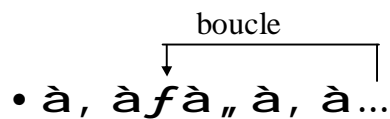
Algorithme génétique

Initialisation de la population
 Evaluation des fonctions objectif
 Calcul de l'efficacité
 For i= 1 to MaxIter
 % Sélection aléatoire,
 Sélection proportionnelle à l'efficacité
 Croisement
 Mutation
 Evaluation des fonctions objectif
 Calcul de l'efficacité
 % Fusion des populations
 End For

1.7.3.1 L'optimisation multiobjectif et les algorithmes génétiques :

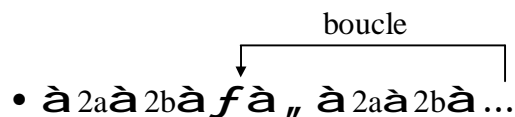
Les algorithmes génétiques sont très bien adaptés au traitement d'un problème d'optimisation multi-objectif.

Le schéma de fonctionnement d'un algorithme génétique pour la résolution d'un problème d'optimisation monobjectif est le suivant :



- : initialisation de la population
- , : évaluation de l'efficacité des individus de la population
- f** : croisement
- „ : mutation
- ... : sélection

le schéma de fonctionnement d'un algorithme génétique pour la résolution d'un PMO est le suivant :



- : initialisation de la population
- 2a : évaluation de l'efficacité des individus de la population
- 2b : transfert vecteur /efficacité
- f** : croisement
- „ : mutation

... : sélection

la plupart des algorithmes utilisent la distance de « Hamming » pour caractériser la différence entre deux individus. Cette distance est calculée de la façon suivante : chaque différence entre les gènes compte pour plus un (+1).

a. **La méthode M.O.G.A.**(Multiple Objective Genetic Algorithm)

Cette méthode est présentée dans [Deb, 99] et [Fonseca, 93]. Elle utilise la relation de dominance de Pareto pour calculer l'efficacité des individus.

Le rang d'un individu x_i à la génération t est donné par :

$$\text{Rang}(x_i, t) = 1 + p_i^{(t)}$$

Où $p_i^{(t)}$ est le nombre d'individu qui dominent x_i à la génération t .

Pour calculer l'efficacité des individus, on suit les étapes suivantes :

1. classer les individus en fonction de leur rang.
2. affecter une efficacité à un individu en interpolant à partir du meilleur rang au plus mauvais.

Algorithme M.O.G.A

```

Initialisation de la population
Evaluation des fonctions objectif
Assignation d'un rang basé sur la dominance
Assignation d'une efficacité à partir du rang
For i=1 to maxG
    Sélection aléatoire proportionnelle à l'efficacité
    Croisement
    Mutation
    Evaluation des fonctions objectif
    Assignation d'un rang basé sur la dominance
    Assignation d'une efficacité à partir du rang
Endfor

```

Discussion :

Cette méthode ne permet pas, dans certain cas, d'obtenir une diversité dans la représentation des solutions.

b. **La méthode N.S.G.A.**(Non dominated Sorting Genetic Algorithm)

Cette méthode reprend les grandes lignes de la méthode MOGA. La différence principale intervient dans l'efficacité d'un individu [Srinivas, 93].

Dans un premier temps, on affecte à chaque individu un rang de Pareto. Tous les individus de même rang sont classés dans une catégorie. Chaque catégorie aura une efficacité inversement proportionnelle à son rang.

Pour maintenir la diversité de la population, on va partager la valeur de l'efficacité d'une catégorie entre ses individus de la façon suivante :

- On calcule, pour chaque individu, le compte de ses voisins

$$m_i = \sum_{j=1}^K Sh(d(i, j))$$

$$Sh(d(i, j)) = \begin{cases} 1 - \left(\frac{d(i, j)}{\sigma_{share}}\right)^2 & \text{si } d(i, j) < \sigma_{share} \\ 0 & \text{sin on} \end{cases}$$

avec K : nombre d'individus de la catégorie.

σ_{share} : permet de définir une zone d'influence pour le calcul de l'efficacité de l'individu

- L'efficacité f_i de l'individu i est donnée par : $f_i = \frac{F}{m_i}$

Discussion :

C'est une méthode efficace, qui permet d'obtenir une diversité dans la représentation des solutions, mais elle reste très sensible au choix du diamètre de la niche σ_{share} . Récemment une version améliorée de NSGA appelée NSGA-II est apparue [Deb, 00].

c. **La méthode SPEA :** (Strength Pareto Evolutionary Algorithm)

La méthode SPEA implémentée par Zitzler et Thiele [Zitzler, 98] est l'illustration même d'un algorithme évolutionnaire élitiste. SPEA maintient un archive externe contenant les meilleurs front de compromis rencontré durant la recherche.

Algorithme SPEA

Initialiser la population P_0 et créer l'archive externe vide $\bar{P} = f$

Mise à jour de \bar{P} à partir des individus non dominés de P_0

Tant que critère- d'arrêt- non- rencontré faire

- Calcul de la valeur d'adaptation pour tous les individus de $P + \bar{P}$
- Sélection dans $P_t + \bar{P}$ en fonction de la valeur d'adaptation
- Croisement
- Mutation
- Mise à jour de \bar{P} à partir des individus non dominés de P

FinTant que

Récemment une version améliorée de SPEA appelée SPEA-2 est apparue [Eckart, 01].

d. **La méthode PAES:** (The Pareto Archived Evolution Strategy)

La méthode PAES a été introduite par Knowles and Corne [Know, 99]. Elle utilise une stratégie évolutionnaire simple de recherche locale. elle maintient un archive externe des solutions non dominées pour assurer une diversité dans les solutions trouvées

e. **La méthode PESA :** (The Pareto Envelope-based Selection Algorithm)

La méthode PESA a été proposée par Corne et al (2000). Elle utilise une population interne de taille réduite et une population externe de taille considérable.

f. **La méthode MOMGA:** (Multi-objective Messy Genetic Algorithm)

Cette méthode a été proposée par Van Veldhuizen and Lamont (2000).

1.7.4 **les méthodes hybrides :**

Afin d'améliorer les performances d'un algorithme, on essaye de le combiner avec une autre méthode [TAL, 00]. Ce principe général appelé hybridation, peut s'appliquer à plusieurs méthodes. Un cas particulier de l'hybridation entre deux méthodes consiste à combiner un algorithme génétique avec une méthode de recherche locale. Dans une telle hybridation on substitue souvent la mutation par une méthode de recherche locale. Dans le cas des PMOs, on peut citer les méthodes hybrides suivantes :

- § la méthode MOTS [Hansen, 97 ; Gandibleux , 97] combinant une population et une recherche Tabou,
- § La méthode PSA [Czyzak, 98] combinant un algorithme génétique et le recuit simulé,
- § La méthode M-PAES [Corne, 00] intégrant un schéma généralisant l'implémentation d'un grand nombre d'algorithmes hybrides pour l'optimisation multiobjectif.

1.7.5 **Autres métaheuristiques :**

D'autres Métaheuristiques ont été adapté au cas multiobjectif et ont prouvé leur succès.

- § Les colonies de fourmis [Mariano, 99], [Iredi , 2001], ...
- § L'Optimisation par essaim de particules : [Moore and Champman, 1999], [Ray and Liew, 2002], [Coello and Lechuga, 2002], ...
- § Les réseaux de neurones artificiels : [Yoo and Hajela, 1999], [Kurapati and Azarm,2000], [Cui et al., 2001], [Coello and Cruz, 2002]
- § Les algorithmes culturelles :[Coello and Landa, 2003]

1.8 Evaluation des méthodes d'optimisation multi-objectif:

Vu le nombre important des méthodes d'optimisation multiobjectif, il devient indispensable de disposer des outils nécessaires pour leur comparaison. L'évaluation des surfaces de compromis est un problème délicat qui oblige à utiliser plusieurs mesures différentes, car il est impossible de représenter par une unique valeur réelle la qualité des solutions, la taille du front, la répartition des solutions sur le front, Parmi les métriques les plus utilisées, on trouve :

1.8.1 l'hypervolume et le rapport d'hypervolume:

Cette métrique a été définie dans [Zitzler, 99], elle permet de mesurer le volume compris sous la courbe formée par les points de l'ensemble à évoluer. Ainsi, dans le cas d'un problème bi-objectif, on parle d'hypersurface et non d'hypervolume.

$$H = \left\{ \mathbf{U} a_i / v_i \in A \right\},$$

Où a_i désigne le volume occupé par la solution v_i .

On définit aussi un rapport d'hypervolume :

$$RS = \frac{H_s}{H_A}$$

Où H_s est le volume occupé par l'ensemble des solutions non dominées actuel et H_A est le volume occupé par l'ensemble des solutions non dominées théorique.

Plus la valeur de cette métrique est proche de 1, meilleure est la surface de compromis.

1.8.2 la métrique C :

Cette métrique a été introduite par Zitzler [Zitzler, 99]. Soient A et B deux surfaces de compromis, la fonction C suivante transforme les deux surfaces de compromis A et B en un nombre réel compris entre 0 et 1.

C'est une métrique relative qui permet de comparer deux surfaces de compromis A et B

$$C(A, B) = \frac{|\{z \in B\} / \exists z' \in A : z' \text{ domine } z\}|}{|B|}$$

Ainsi, la valeur $C(A, B) = 1$ signifie que tous les points de B sont dominés ou égaux à ceux de A. et la valeur $C(A, B) = 0$ signifie qu'aucun point de B n'est dominé par un point de A.

La mesure C n'étant pas symétrique, il est nécessaire de considérer les deux valeurs $C(A, B)$ et $C(B, A)$ pour obtenir une mesure plus fiable des deux surfaces de compromis A et B à comparer.

1.8.3 la métrique d'espacement :

Cette métrique a été introduite par Schott [Schott, 95]. Elle permet de mesurer l'uniformité de la répartition des points sur la surface de compromis. Soit A l'ensemble de n solutions non dominées, alors $S(A)$ est définie par :

$$S(A) = \left[\frac{1}{n-1} * \sum_{i=1}^n (\bar{d} - d_i)^2 \right]^{\frac{1}{2}}$$

Avec

$$d_i = \min_j (|f_1^i(\mathbf{x}) - f_1^j(\mathbf{x})| + \dots + |f_m^i(\mathbf{x}) - f_m^j(\mathbf{x})|) \quad i, j \in [1, \dots, n] \text{ et } i \neq j,$$

m : nombre de fonctions objectif

\bar{d} : moyenne de tous les d_i

Plus le résultat de la mesure est proche de 1, meilleure est la répartition des points sur la surface de compromis.

Nous avons présenté quelques métriques parmi les plus utilisées aujourd'hui, le lecteur est invité à consulter [Collette, 02] pour une présentation d'une variété de métriques.

1.9 Conclusion :

Nous avons présenté dans ce chapitre les concepts de bases de l'optimisation multiobjective, puis une classification des méthodes correspondantes. Les approches agrégatives ainsi que les approches non Pareto semblent peu efficaces, car elles sont très sensibles à la forme du front de Pareto (concavité, non uniformité,...). Par contre les méthodes Pareto issues des métaheuristiques pour l'optimisation multiobjective semblent bien se prêter à ce type de problèmes. Elles ont prouvées leur efficacité pour un grand nombre de problèmes même ceux ayant des fronts de Pareto non convexes et non uniformes. Nous avons aussi présenter trois parmi les métriques les plus utilisées pour mesurer la qualité de l'ensemble des solutions non dominées trouvées.

L'informatique quantique est une nouvelle science caractérisée par un parallélisme massive et une grande diversité dans la représentation des solutions. Comme cette diversité est un atout important pour une méthode d'optimisation. On a choisit d'incorporer les concepts quantique au niveau d'un algorithme génétique multiobjectif pour au chapitre ** donner une nouvelle approche Pareto issue des métaheuristiques multiobjective basée sur les principes de l'informatique quantique.

Références :

- [Barichard, 03] Vincent Barichard, « Approches Hybrides pour les problèmes multiobjectif », , Thèse de Doctorat, université d'Angres, Novembre 2003
- [Benabdel, 97] Ben Abdelaziz F. And Krichen S. (1997), A tabu search heuristic for multiple objective knapsack problems, Ructor Reasearch Report, RR 28-67.
- [Coello and Cruz, 2002]
- [Coello and Landa, 2003]
- [Coello and Lechuga, 2002]
- [Coello, 98] C. A. Coello Coello, An updated survey og G.A. based multiobjective optimization technics, Rapport technique Lania-RD-98-08, Xalapa Veracruz, Mexico, décembre 1998, <http://www.lania.mx/~ccoello/EMOO>.
- [Coello, 99] C. A. Coello Coello, A survey of constraint handling techniques used with evolutionary computation methods, Rapport technique Lania-RI-99-04, Laboratoire national de l'informatique Avanzada, Xalapa, Veracruz,

-
- Mexico, 1999, <http://www.lania.mx/~ccoello/EMOO>.
- [Collette, 02] Y. Collette et P. Siarry. Optimisation multiobjectif. Eyrolles, 2002.
- [Corne, 00] D.W. Corne and J. D. Knowles. M-paes: a memetic algorithm for multiobjective optimization. In Proceeding of the 2000 Congress on Evolutionary Computation, pages 325-332, 2000.
- [Cui et al., 2001]
- [Czyzak, 98] P. Czyzak and A.Jaszkiwicz, A Pareto simulated annealing – a metaheuristic for multiple objective combinatorial optimization. Journal of Multi-Criteria Decision Analysis, 7(1):34-47, 1998.
- [Deb, 00] K. Deb, S. Agrawal, A. Pratapand and T. Meyarivan, “A Fast and Elitist multiobjective genetic algorithm: NSGA-II”, Technical Report 20001, Indian Institute of Technology, Kanpur Genetic Algorithms Laboratory(Kan GAL),2000.
- [Deb, 99] K. Deb, An overview of multi-objective evolutionary algorithms, Journée J.E.T., Copie de transparents, mai 1999.
- [Eckart, 01] Eckart Zitzler, Marco Laumanns, and Lothar Thiele, ”SPEA2:Improving the Performance of the Strength Pareto evolutionary Algorithm”, Technical Report 103, Computer Engineering and Communication Networks lab(Tik), Swiss Federal Institute of Technology(ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, May 2001.
- [Engrand , 98] P. Engrand, X. Mouney, Une méthode originale d'optimisation multiobjectif, Note interne EDF-DER numéro HT-14/97/035/A, mars 1998.
- [Fonseca, 93] C.M. Fonseca, P. J. Fleming, Genetic Algorithms for multiobjective optimization : Formulation, Discussion and Generalization, Proceedings of the Fifth International Conference on Genetic Algorithms, pages 416-423, San Mateo, California, 1993.
- [Gandibleux, 97] Gandibleux X., Mezdaoui N., Freville A. (1997), A multiobjective tabu search procedure to solve combinatorial optimization problems, in : R.Caballero, F. Ruiz, R. Steuer (eds), Advances in Multiple Objective and Goal Programming. Lecture Notes in Econom
- [Glover , 97] F. Glover, M. Laguna, Tabu search, Kluwer Academic Publishers, 1997.
- [Glover ,86] F. Glover , Future Paths for integer programming and links to artificial intelligence, Comput. and Ops. Res., volume 13, numéro 5, pages 533-549, 1986.
- [Glover ,90] F. Glover , Artificial intelligence, heuristic frameworks and tabu search, Managerial and decision economics, volume 11, pages 365-375, 1990.
-

- [Glover, 97] F. Glover, M. Laguna, Tabu search, Kluwer Academic Publishers, 1997.
- [Goldberg, 87] D.E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In Genetic Algorithms and their applications: Proceedings of the second International Conference on Genetic Algorithms, pages 41-49. Lawrence Erlbaum, 1987.
- [Goldberg, 89] D.E. Goldberg. Genetic algorithms for search, optimization, and machine learning. Reading, MA: Addison-Wesley, 1989.
- [Goldberg, 94] D. Goldberg, Algorithmes génétiques, éditions Addison Wesley, juin 1994.
- [Hansen, 97] M. P. Hansen. Tabu search for multiobjective optimization: Mots. In Proceedings of 13th International conference on MCDM, 1997.
- [Hansen, 98] Hansen M.P.(1998), Metaheuristics for multiple objective combinatorial optimization, Ph.D. Thesis, IMM-PHS-1998-45, Technical University of Denmark, Lyngby.
- [Iredi , 01] S. Iredi , D. Merkle, M. Middendorf, Bi-criterion optimization with multi colony algorithms, in : Proceeding of the First International Conference on Evolutionary Multi-Criterion Optimization. Lecture Notes in Computer Science , Springer, Berlin, 2001.
- [Know, 99] Knowles, J.D. and Corne, D.W., 1999, The Pareto archived evolution strategy: a new baseline algorithm for multiobjective optimization, in 1999 Congress on Evolutionary Computation , Piscataway, NJ, IEEE Service Center, 98-105.
- [Kurapati and Azarm, 2000]
- [Mariano, 99] E. Mariano and E. Morales, 1999, MOAQ and ant-Q algorithm for multiple objective optimization problems, in W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, R. E. Smith (eds), Proceeding of the genetic and evolutionary Computation
- [Miettinen, 99] K. M. Miettinen, Nonlinear multiobjective optimization, éditions Kluwer academic publisher, 1999.
- [Moore and Champman, 1999]
- [Ray and Liew, 2002]
- [Schott, 95] J. R Schott. Fault tolerant design using single and multicriteria genetic algorithm optimization. Master's thesis, Department of Aeronautics and

-
- Astronautics, Massachusetts Institute of Technology, 1995.
- [Serafini, 92] Serafini 1992, Simulated annealing for multiple objective optimization problems, in: Proceeding of the Tenth International Conference on Multiple Criteria Decision Making, Taipei 19-24.07, vol. 1, 87-96.
- [Srinivas , 93] N. Srinivas , k , Deb, multiobjective optimization using non-dominated sorting in genetic algorithms, Rapport technique, Department of technical Engineering , Indian Institute of Technology, Kanput, India, 1993.
- [TAL, 00] E. G. Talbi. Une taxinomie des métaheuristiques hybrides. Dans ROADEF'2000, 2000.
- [TAL, 01] Talbi E., Métaheuristiques pour l'optimisation combinatoire multi-objectif : Etat de l'art, Lille : LIFL, 2001
- [Ulungu , 99] E. L. Ulungu , J. Teghem, P. H. Fortemps, D. Tuyttens, MOSA Method: A Tool for Solving Multiobjective Combinatorial Optimization Problems, Journal of Multicriteria Decision Analysis, Volume 8, numéro 4, pages 221-236, 1999.
- [Yoo and Hajela, 1999]
- [Zitzler, 98] E. Zitzler and L. Thiele, An evolutionary algorithm for multiobjective optimization: the Strength Pareto Approach. Technical report, Swiss Federal Institute of technology (Zurich), 1998.
- [Zitzler, 99] E. Zitzler, K. Deb and L. Thiele, Comparison of multiobjective evolutionary algorithms: Empirical results, Technique 70, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of technology (ETH) Zurich, December 1999.

Chapitre 2

L'informatique quantique

Dans ce chapitre, nous présentons les principes de base de l'informatique quantique. Ainsi qu'un état des travaux proposés dans ce cadre.

Sommaire

1.1 Introduction	33
1.2 Historique	33
1.3 Principe de base	34
1.4 Formalisme mathématique	34
1.5 L'informatique quantique	35
1.6 Les opérateurs logiques quantiques	39
1.7 Le parallélisme quantique	45
1.8 La cryptographie quantique	46
1.9 Correction d'erreurs quantiques	46
1.10 L'algorithmique quantique	48
1.11 Les algorithmes évolutionnaires quantiques	50
1.12 Conclusion	54

2.1 Introduction :

Depuis ses débuts, l'informatique électronique n'a été qu'une course effrénée à la rapidité, due à la miniaturisation des transistors. En analysant l'évolution de la miniaturisation des processeurs et des moyens de stockage de données, une extrapolation, même approximative, nous montre que vers 2020, il nous faudra que quelques atomes pour stocker un bit. Sur un tel système, les règles classiques ne peuvent plus s'appliquer, nous sommes alors dans un autre monde où les comportements sont dictés par d'autres lois, celles de la mécanique quantique.

En attendant 2020, l'informatique quantique, née de la rencontre de physiciens et de théoriciens de en, a déjà jeté les bases d'un nouvel espace technique. On est encore loin de réalisations de taille industrielle, mais ce domaine est suffisamment prometteur pour que l'on doive dès aujourd'hui s'y intéresser, car cela risque bien de bouleverser le paysage informatique d'ici 10 à 20 ans et avec lui quelques-unes de nos certitudes.

2.2 Historique :

Au cours des années 70, Richard Feynman, prix Nobel de physique 1965, émet l'idée d'un ordinateur quantique. Son projet ne retient pas l'attention car il ne propose pas d'application concrète. En 1994 Peter Shor [Shor, 94], des laboratoires AT et T Reasearch, conçoit un algorithme quantique de factorisation des grands nombres. Shor souligne qu'avec un ordinateur classique, aboutir au résultat prendrait des millions d'années, tandis qu'avec un ordinateur quantique, cela ne prendrait qu'une année. Or, la factorisation des grands nombres est l'un des principes de base du décryptage. La publication de Shor fait grand bruit.

En 1996, Lov Grover[Grover, 97], de Bell Labs, publie un algorithme quantique de recherche d'un élément dans un ensemble de n objets. Une application intéressante de cet algorithme est la recherche dans une base de données.

En 1996, Neil Gershenfeld, du MIT, et Isaac Chuang, aujourd'hui à IBM, inventent une toute nouvelle approche réalisable de l'ordinateur quantique. L'information y est représentée par les spins nucléaires des atomes contrôlés par des techniques de résonance magnétique nucléaire.

En 1998 l'équipe de Berkely, dirigée par Isaac Chuang, annonce la réalisation du premier ordinateur quantique 2-qbits, dans un dé à coudre de chloroforme (CHCl_3). L'ordinateur retrouve une donnée parmi quatre en une seule étape. Il s'agit de la première mise en œuvre de l'algorithme de Lov Grover.

En 1999, fût crée l'ordinateur à 3-qbits. Une équipe du centre de recherche IBM d'Almaden, dirigée par Isaac Chuang a fait la démonstration de l'algorithme quantique de recherche dans une base de données à huit éléments.

Août 2000, IBM a annoncé la mise en oeuvre de l'ordinateur quantique à 5-qbits. L'ordinateur d'Isaac Chuang utilise les cinq atomes de fluor d'une molécule complexe spécialement conçue. Il permet de trouver, en une seule étape, la bonne

combinaison parmi deux éléments. Il faudrait jusqu'à quatre étapes dans une approche classique. Deux des 5-qbits sont utilisés dans une recherche de la solution, les trois autres étant réservés au calcul du résultat.

Bien qu'il va falloir beaucoup de temps pour que ce domaine atteigne un certain degré de maturité, cette évolution historique ne peut être qu'un indice important de son avenir prometteur.

2.3 Principe de base:

La compréhension, même superficielle de l'informatique quantique, ne peut se faire sans quelques notions de bases sur la théorie quantique, c'est pourquoi, nous tenteront dans la section suivante, de dégager les notions de base de cette théorie.

La physique classique s'appuie sur deux notions fondamentales et différentes : l'onde (champs) et la particule (corpuscule). Au début du vingtième siècle, certaines expériences ont ébranlé ces concepts classiques, mettant en avant la particularité que, sous certaines conditions, un champ avait certains aspects corpusculaires, et vice versa. Ces deux notions perdirent donc leur caractère distinct. Nous devons abandonner l'idée que tout objet est soit un champ soit une particule. Il faut donc reconnaître qu'on a affaire à d'autres objets, proprement quantique, appelé quantons. Les inégalités de Heisenberg, l'un des piliers de la théorie quantique, prouvent qu'il faut abandonner l'idée naturelle qui affirme que chaque système a des valeurs de grandeurs bien déterminées. En effet, ces grandeurs ont le plus souvent une dispersion. Un quanton peut par exemple avoir une localisation spatiale spectrale, signifiant ainsi, que la notion de position n'a pas de sens pour lui, c'est comme s'il était à toutes les positions (de son spectre) en même temps. Cependant, si l'on effectue une mesure expérimentale sur cette position, nous obtiendront une valeur unique avec une certaine probabilité, nous aurons, en quelque sorte rendue sa position classique. Le quanton, ne sera plus en superposition d'états, mais bien à l'état dans lequel nous l'avons trouvé lors de la mesure. Nous avons irrémédiablement modifié l'état du quanton, l'ayant fait interagir avec le monde extérieur, nous avons cassé son caractère quantique. Il s'agit du phénomène de décohérence [Zurek, 91].

2.4 Formalisme mathématique :

2.4.1 Espace de Hilbert :

En théorie classique, les diverses grandeurs physiques d'un système sont mathématiquement décrites par des nombres, ou plus précisément des fonctions du temps à valeurs numériques. Les inégalités d'Heisenberg ont montré qu'il est par contre impossible, en règle générale, de caractériser un système quantique par une

valeur bien déterminée de toutes ses grandeurs physiques. La description d'un système de quantons exige un nouveau formalisme faisant appel à des concepts mathématiques radicalement différents de ceux utilisés par la physique classique. Ce formalisme repose sur les notions mathématiques d'espace de Hilbert et de ses opérateurs. Contrairement à l'espace Euclidien, l'espace de Hilbert repose sur le corps des nombres complexes.

L'état d'un système quantique est représenté par un vecteur, normé à l'unité, dans un espace de Hilbert H , appelé « espace des états » du système.

2.4.2 La notation de Dirac :

Soit 0 et 1 les valeurs classiques de l'état d'un bit. On associe à chacune de ces valeurs le vecteur d'une base orthogonale $\left\{ \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\}$ de l'espace vectoriel \mathbb{C}^2 , en utilisant les notations de Dirac [Dirac, 58] :

Valeur	Notation vectorielle	Notation de Dirac
0	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$ 0\rangle$
1	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$ 1\rangle$

Figure 2.1 Notation matricielle et notation de Dirac

Voici quelques interprétations de notations :

- § $|x\rangle$ dénote un vecteur colonne utilisé pour décrire un état quantique,
- § $\langle x|$ désigne le vecteur transposé de $|x\rangle$,
- § $|x\rangle\langle y|$ désigne le produit des deux vecteurs $|x\rangle$ et $|y\rangle$

$$|0\rangle\langle 1| = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \quad (2.1)$$

2.5 L'informatique quantique :

2.5.1 Le Qbit :

Un ordinateur, qu'il soit classique ou quantique, ne fait que manipuler des bits : systèmes physiques pouvant prendre deux valeurs logiques distinctes 0 ou 1. Dans le cadre classique, on utilise des transistors pour obtenir une présence ou non du courant qui représente les valeurs logiques. Le bit quantique, ou plus simplement qbit (quantum bit en anglais) [Shumacher, 95] est un système microscopique (atome, électron, photon, etc ...) pouvant prendre deux valeurs distinctes 0 ou 1. L'ordinateur

quantique n'existe pas encore. Plusieurs voies d'étude sont lancées pour la réalisation du qbit. Le qbit peut être dans l'état 0 avec une probabilité $|a|^2$, l'état 1 avec une probabilité $|b|^2$. C'est là toute la puissance de l'informatique quantique. Ainsi, un qbit vit dans un espace de Hilbert à deux dimensions et peut être dans une superposition de ces deux états orthogonaux.

$$|\Psi\rangle = a|0\rangle + b|1\rangle \quad (2.2)$$

Où a et b sont deux nombres complexes vérifiant la condition $|a|^2 + |b|^2 = 1$.

Il est intéressant de mentionner que selon le théorème de Holevo [Holevo, 79], seulement un bit classique peut être extrait d'un bit quantique.

Sous leur apparente ressemblance, les bits classiques offrent des possibilités totalement différentes, dues aux étranges propriétés des qbits, que nous décrirons ci-après.

Le tableau suivant propose une comparaison entre les bits classiques et les bits quantiques.

Bit Classique	Bit Quantique
Un bit a toujours une valeur définie.	Pas de valeur définie pour le qbit tant qu'on ne l'observe pas.
Un bit vaut seulement 0 ou 1.	Un qbit peut être dans une superposition de 0 et 1 simultanément.
Un bit peut être copié sans être affecté.	Un qbit dans un état inconnu ne peut être copié sans erreur.
Un bit peut être lu sans affecter sa valeur.	Lire un qbit qui est initialement dans une superposition changera sa valeur.
Lire un bit n'affecte pas un autre.	Si un qbit lu est enchevêtré avec un autre qbit, en lire un affectera l'autre.

2.5.2 Registre quantique :

Un registre quantique est composé d'un ensemble de qbits. De part sa complexité et les interactions de ses qbits, un registre quantique confère à l'informatique quantique de nombreuses propriétés. Ces propriétés sont à l'origine des cinq grands principes de l'informatique quantique.

Un registre quantique Ψ est un ensemble de qubits, c'est une superposition arbitraire de n qubits tel que:

$$\Psi = \sum_{x=0}^{2^n-1} C_x |X\rangle \quad (2.3)$$

Où les amplitudes C_x satisfont la règle de normalisation suivante :

$$\sum_{x=0}^{2^n-1} |C_x|^2 = 1 \quad (2.4)$$

Ce phénomène de superposition permet de stocker un volume important d'informations comparé à la capacité de stockage dans un système classique. Le tableau suivant donne une idée sur le nombre de bits classiques requis pour représenter le même volume d'information qu'un nombre donné de qbits.

Nombre de qbits	Nombre de bits classiques nécessaires pour une description complète
10	1000
20	1000000
30	1000000000
300	Plus que le nombre d'atomes dans l'univers visible

On peut bien constater la puissance de calcul de l'ordinateur quantique qui dépasse largement celle de l'ordinateur classique. C'est la conclusion à laquelle est arrivé le physicien Richard Feynman en 1982 et à qui revient le mérite de l'émergence de ce domaine fascinant qu'est l'informatique quantique.

2.5.3 Les cinq principes de l'informatique quantique :

L'informatique quantique repose sur cinq principes de base :

2.5.3.1 Superposition :

Comme nous l'avons vu précédemment, un qbit peut être dans un état 0 ou 1, mais également dans les deux en même temps. Un registre quantique de n qbits pourra donc, comme un registre classique, contenir 2^n valeurs distinctes. Mais de part sa nature quantique, il pourra aussi contenir à la fois toutes ces valeurs, sous la forme d'une superposition quantique des 2^n états. Un ordinateur qui traiterait cette information sans perdre la superposition quantique serait capable de calculer à la fois les 2^n résultats correspondant à toutes les valeurs d'entrées.

Exemple :

Imaginez un système à 3 bits, il peut se trouver dans une configuration parmi 8 états possibles (000, 001, ..., 111). Avec un système quantique à 3 qbits, les 8 configurations sont mémorisées simultanément.

$$\frac{1}{\sqrt{8}}(|000\rangle + |001\rangle + \dots + |111\rangle) \quad (2.5)$$

Un système à n qbits travaillera donc sur 2^n nombres à la fois. Là où l'ordinateur classique va répéter 2^n fois l'opération ou faire 2^n calculs en parallèle, avec l'ordinateur quantique en une seule étape on pourra appliquer la même opération sur 2^n nombres distincts.

Ce parallélisme quantique massif [Deutsch, 92] rendrait l'ordinateur quantique infiniment plus efficace qu'un ordinateur classique. Il sera donc possible de développer de nouveaux algorithmes qui permettent la résolution de problèmes jusqu'à présent inaccessibles pour l'informatique classique.

2.5.3.2 Interférence :

L'interférence [Nielsen, 00] est une porte quantique spéciale permettant soit d'augmenter (Interférence Constructive) ou de diminuer (interférence destructive) la probabilité d'obtenir un état.

2.5.3.3 Enchevêtrement :

Une des propriétés les plus intrigantes de la théorie quantique est le concept d'enchevêtrement et la non localité qu'elle implique. L'enchevêtrement semble être responsable de la puissance du calcul quantique ([Ekert, 98], [Fitzgerad, 01]). D'un point de vue mathématique, l'état global d'un système quantique est dit enchevêtré s'il est impossible de l'écrire sous forme d'un produit tensoriel de deux de ses sous systèmes.

$$|\Phi_{AB}\rangle \neq |\Phi_A\rangle \otimes |\Phi_B\rangle \quad (2.6)$$

En d'autres termes, deux bits quantiques sont dits enchevêtrés si l'un dépend de l'autre et toute modification de l'un affecte l'autre.

Exemple :

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \text{ est un état enchevêtré.}$$

Dans ce cas d'enchevêtrement, les deux qbits sont dits enchevêtrés car chacun dépend de l'autre. A partir de la formule précédente, on peut déduire que les deux quantons ne peuvent être que dans des états identiques, donc si on mesure un des deux qbits et l'on obtient l'état h alors forcément le deuxième qbit est à l'état h .

La non localité ou téléportation : Dans certains cas, l'observation d'une particule à un endroit peut influencer instantanément l'état d'une autre particule à un autre endroit, aussi lointain qu'on le veut du premier.

2.5.3.4 Non déterminisme :

Contrairement à la physique classique, la mécanique quantique n'est pas déterministe. En d'autres termes, les mêmes causes ne produisent pas nécessairement les mêmes effets. Le non déterminisme signifie que si on prépare un qbit dans une superposition d'états et qu'on le mesure, on ne peut pas prédire si on va obtenir 0 ou 1. C'est ce qui confère à l'informatique quantique toute sa magie et sa puissance théorique.

2.5.3.5 Non clonabilité :

En 1982, W. Wootters, W. Zurek et D. Dieks énoncent le principe de non-duplication qui affirme que la copie n'est pas toujours possible dans le monde quantique. Comme la mesure détruit l'état quantique d'un photon, on ne peut jamais connaître avec certitude cet état. De là, on ne peut pas copier ce que l'on ne connaît pas. Ce principe de non clonabilité est très utile pour la cryptographie quantique.

2.6 Les opérateurs logiques quantiques :

Pour l'informatique classique, la possibilité de réaliser les portes logiques NOT et NAND sur un ensemble de bits classiques permet de réaliser n'importe quel calcul

classique. On dit alors que (NOT et NAND) forment un ensemble universel (complet) pour le calcul classique.

Pour le calcul quantique, un ensemble universel doit aussi être défini. Avant de présenter un tel ensemble, il est toutefois nécessaire de bien comprendre ce qu'est un calcul quantique. Dans le paradigme standard, ce n'est rien de plus que l'évolution unitaire contrôlée d'un système quantique

$$|y(t)\rangle = U(t)|y(0)\rangle, \quad (2.7)$$

Où $|y(0)\rangle$ est l'état initial de l'ordinateur et l'état final qui, une fois mesuré, sera la réponse au calcul. L'opérateur unitaire d'évolution $U(t)$ représente le programme quantique : la dynamique du système est choisie de façon à correspondre au calcul à effectuer.

2.6.1.1 Les portes logiques quantiques

Le principe du calcul quantique est simple, on commence par préparer n qbits à l'état $|0\rangle$. Ensuite ils subissent une évolution unitaire dans l'espace H^{2^n} de l'instant de départ t_0 jusqu'à l'instant t . cette évolution est décrite par un opérateur unitaire $U(t, t_0)$ agissant dans H^{2^n} . Une mesure des qbits est effectuée au temps t .

L'évolution unitaire U doit être réversible c'est-à-dire $U^{-1}(t, t_0) = U(t, t_0)$

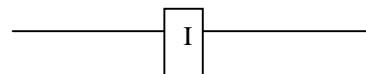
Voici quelques portes quantiques simples :

L'identité I : $|0\rangle \rightarrow |0\rangle$

$|1\rangle \rightarrow |1\rangle$

- La matrice correspondante est $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

- Le circuit correspondant est

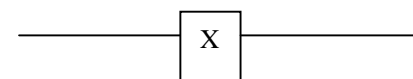


La négation X : $|0\rangle \rightarrow |1\rangle$

$|1\rangle \rightarrow |0\rangle$

- La matrice correspondante est $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

- Le circuit correspondant est



La porte Y : $|0\rangle \rightarrow -|1\rangle$
 $|1\rangle \rightarrow |0\rangle$

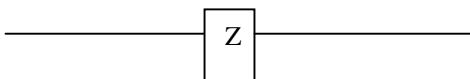
- La matrice correspondante est $\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$

- Le circuit correspondant est 

La transformation de décalage de phase Z :

Z : $|0\rangle \rightarrow |0\rangle$
 $|1\rangle \rightarrow -|1\rangle$

- La matrice correspondante est $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$

- Le circuit correspondant est 

Toutes ces portes sont unitaires. Si on prend l'exemple de la porte Y on aura :

$$YY^* = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} = I$$

La négation contrôlée C_{Not} :

C_{Not} : $|00\rangle \rightarrow |00\rangle$
 $|01\rangle \rightarrow |01\rangle$
 $|10\rangle \rightarrow |11\rangle$
 $|11\rangle \rightarrow |10\rangle$

- La matrice correspondante est $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

- Le circuit correspondant est

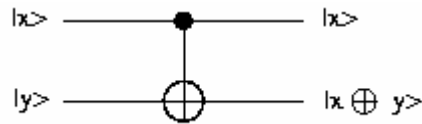


Figure 2.2 Circuit quantique représentant le Controlled-Not

Le cercle noir indique le bit de contrôle et le cercle ouvert indique la négation conditionnelle du bit sujet.

La porte controlled-Not, C_{Not} , change le deuxième bit si le premier est à 1, et le laisse inchangé sinon.

I. Entrée	II. Sortie
00	00
01	01
10	11
11	10

Figure 2.3 Table de vérité du C_{Not}

En général, on peut avoir plusieurs bits de contrôle. Ainsi le Non Contrôlé- Contrôlé (Controlled-Controlled-Not) inverse le troisième bit si et seulement si les deux premiers sont tous les deux à 1. Le Controlled-Controlled-Not a la représentation graphique suivante :

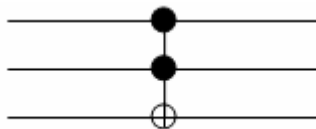


Figure 2.4 La porte Controlled-Controlled-Not

Porte Racine Carré de NOT :

La porte $U_{\sqrt{NOT}}$ appliquée deux fois de suite à un qbit, l'inverse.

$$U_{\sqrt{NOT}}U_{\sqrt{NOT}}|0\rangle = U_{NOT}|0\rangle = |1\rangle,$$

$$U_{\sqrt{NOT}}U_{\sqrt{NOT}}|1\rangle = U_{NOT}|1\rangle = |0\rangle$$

L'opérateur \sqrt{NOT} est définie par :

$$U_{\sqrt{NOT}}|0\rangle = \left(\frac{1}{2} + \frac{i}{2}\right)|0\rangle + \left(\frac{1}{2} - \frac{i}{2}\right)|1\rangle$$

$$U_{\sqrt{NOT}}|1\rangle = \left(\frac{1}{2} - \frac{i}{2}\right)|0\rangle + \left(\frac{1}{2} + \frac{i}{2}\right)|1\rangle$$

Ceci peut être représenté graphiquement par :

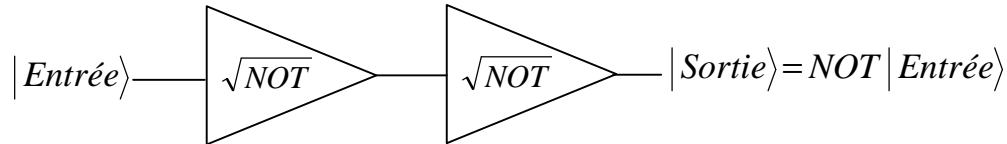


Figure 2.5 la porte \sqrt{NOT}

La porte Walsh-Hadamaard

Une autre transformation à un seul qubit qui est très importante est la transformation de Hadamard. Cette porte permet une rotation de l'état d'un qubit :

$$U_{WH}|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad U_{WH}|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \quad (2.8)$$

Cette porte permet de créer des superpositions et de représenter 2^n valeurs dans n qubits comme suit :

$$\begin{aligned} U_{WH}|0\rangle \otimes U_{WH}|0\rangle \otimes \mathbf{K} \otimes U_{WH}|0\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \mathbf{K} \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ &= \frac{1}{\sqrt{2^n}}(|00\mathbf{K}0\rangle + |00\mathbf{K}1\rangle + \mathbf{K} + |11\mathbf{K}1\rangle) \\ &= \frac{1}{\sqrt{2^n}}(|0\rangle + |1\rangle + |2\rangle + \mathbf{K} + |2^n - 1\rangle) \end{aligned}$$

2.6.1.2 Circuits quantiques :

On dispose d'un ensemble complet de portes logiques réversibles qui peuvent accomplir tous les calculs classiques sur un ordinateur quantique.

Si U_1 et U_2 sont deux transformations unitaires alors la transformation conditionnelle

$$|0\rangle\langle 0| \otimes U_1 + |1\rangle\langle 1| \otimes U_2 \quad (2.9)$$

est unitaire.

D'où

$$C_{not} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X \text{ et } T = |0\rangle\langle 0| \otimes I \otimes I + |1\rangle\langle 1| \otimes C_{not}$$

La transformation T représente la porte de Toffoli. Cette porte peut être utilisée pour construire les opérateurs AND et NOT de la façon suivante :

$$T|1,1,x\rangle = |1,1,-x\rangle$$

$$T|x,y,0\rangle = |x,y,x \wedge y\rangle$$

Par exemple le circuit quantique suivant implémente la somme de deux bits x et y , en utilisant les portes de Toffoli et C_{not}

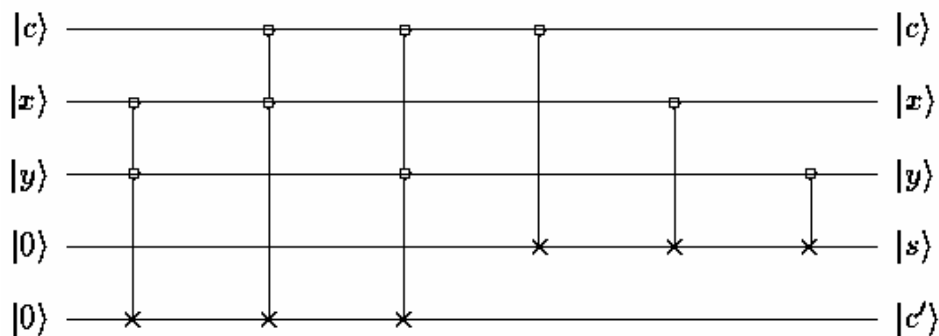
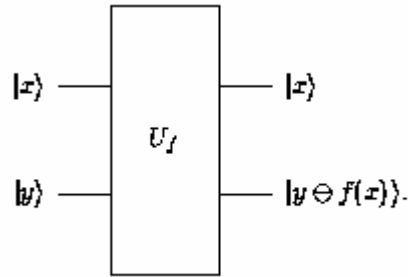


Figure 2.6 circuit quantique pour la somme de deux bits

Pour une fonction classique arbitraire f avec m bits d'entrées et k bits de sorties, on a le circuit quantique suivant :

Figure 2.7 circuit quantique pour une fonction f

2.7 Le parallélisme quantique

Si U_f est appliquée à des qbits en superpositions, elle génère une superposition de résultats. De cette façon on peut calculer $f(x)$ pour n valeurs de x en une seule application de U_f . C'est le parallélisme quantique.

La puissance des algorithmes quantiques provient essentiellement du parallélisme quantique et de l'enchevêtrement. La majorité des algorithmes quantiques commencent par le calcul de la fonction considérée pour une superposition de toutes les valeurs possibles comme suit :

- Commencer avec l'état initial $|0000\dots0\rangle$
- Appliquer la transformation de Walsh-Hadamard pour avoir une superposition.

Exemple : On considère la porte de Toffoli qui calcule la conjonction de deux valeurs.

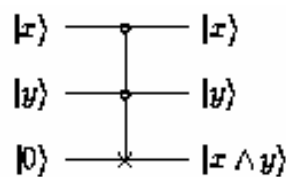


Figure 2.8 porte de Toffoli pour la conjonction

On prend comme entrée une superposition de toutes les combinaisons de bits possibles pour x et y avec la valeur zéro

$$\begin{aligned} H|0\rangle \otimes H|0\rangle \otimes |0\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle \\ &= \frac{1}{\sqrt{2}}(|000\rangle + |010\rangle + |110\rangle). \end{aligned}$$

En Appliquant la transformation de Toffoli à cette superposition de données en entrées pour avoir une superposition de sorties.

$$T(H|0\rangle \otimes H|0\rangle \otimes |0\rangle) = \frac{1}{\sqrt{2}}(|000\rangle + |010\rangle + |111\rangle).$$

On remarque que le résultat trouvé n'est autre que la table de vérité de la conjonction.

2.8 Cryptographie quantique :

La cryptographie quantique est basée sur le fait qu'un bit d'information est codé sur un seul objet quantique [Bellac, 03], comme par exemple un seul photon. Or d'après les lois de la mécanique quantique, il n'est pas possible de mesurer un système quantique inconnu sans le modifier. Un espion essayant d'intercepter une communication entre deux parties (traditionnellement nommées Alice et Bob), ne pourra alors pas extraire d'information utile sans être détecté par Alice et Bob. Ces derniers pourront partager une clé aléatoire avec une sécurité absolue basée sur les lois de la mécanique quantique.

Donc il ne s'agit pas de chiffrer un message à l'aide de l'information quantique, mais, de s'assurer que la transmission de clé (dite publique) n'a pas été espionnée.

Alice envoie vers Bob quatre types de photons polarisés suivant $|\rightarrow\rangle, |\uparrow\rangle, |j\rangle$ et $|k\rangle$ de même Bob analyse ses photons selon les quatre directions verticale, horizontale et inclinées ($\pm 45^\circ$). Exemple Bob enregistre 1 si le photon est polarisé $|\uparrow\rangle$ ou $|j\rangle$ et 0 sinon. Après enregistrement d'un nombre suffisant de photons, Bob annonce publiquement la suite des analyseurs qu'il a utilisés, mais non ses résultats. Alice fait des comparaisons et lui donne toujours publiquement la liste des polarisateurs compatibles avec ses analyseurs, les bits correspondants forment la clé.

Pour vérifier que le message n'a pas été intercepté, Alice et Bob échange une partie de la clé et la compare publiquement, s'ils aperçoivent une réduction de corrélation entre les valeurs de leurs bits, ils déduisent que le message a été intercepté.

2.9 Correction d'erreurs quantiques

Parmi les problèmes rencontrés lors de la construction des ordinateurs quantiques, on trouve la nécessité d'isoler un état quantique. L'interaction des particules, représentant un état quantique, avec l'environnement externe perturbe celui ci et cause sa décohérence.

Les principaux types d'erreurs sont les suivants:

$$a|0\rangle + b|1\rangle \quad \hat{a} \quad \begin{array}{l} a|1\rangle + b|0\rangle \quad Z \\ a|0\rangle - a|1\rangle \quad X \\ a|1\rangle - b|0\rangle \quad Y \end{array}$$

Toutes les autres erreurs peuvent être décrites par des combinaisons de X,Y, et Z. On doit uniquement corriger deux de ces trois types d'erreurs car $Y = XZ$.

Afin d'illustrer les techniques de QEC (Quantum Error Correction) par un exemple simple, on considère le cas où un qbit reverse spontanément sa valeur (« bit flip ») au cours du calcul. Il est possible de protéger l'information quantique contre ce type d'erreurs en encodant chaque qbit logique à l'aide de trois qbits. Une superposition d'états donnera

$$a|0\rangle + b|1\rangle = a|000\rangle + b|111\rangle$$

Une erreur de « bit flip » sur le premier qbit de cette superposition donnera :

$$a|000\rangle + b|111\rangle \quad \hat{a} \quad a|100\rangle + b|011\rangle$$

Afin de détecter ces erreurs, on ajoute aux trois qbits de code, deux qbits auxiliaires préalablement initialisés dans l'état $|0\rangle$, on applique ensuite le circuit suivant :

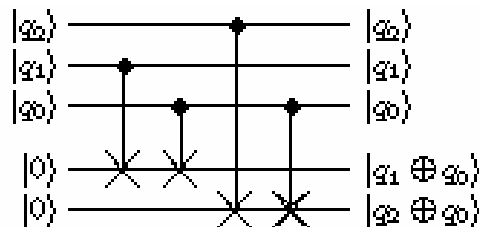


Figure 2.9 circuit correcteur d'erreur

On mesure les deux qbits auxiliaires, pour obtenir le syndrome qui nous renseigne sur le bit sur lequel le « bit flip a été effectué. On applique alors la transformation

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

sur le bit erroné pour le corriger.

Les erreurs de phase (« phase flip ») sont de la forme :

$$a|0\rangle + b|1\rangle \quad \hat{a} \quad a|0\rangle - a|1\rangle$$

et peuvent être corrigées par des techniques semblables à celles du « bit flip ».

2.10 L'algorithmique quantique :

La puissance potentielle d'un ordinateur quantique a été reconnue par Richard Feynman [Feynman, 82]. David Deutsch a montré ensuite que les problèmes faciles pour les ordinateurs classiques seraient aussi faciles pour tous les ordinateurs quantiques [Deutsch, 85]. A présent, La conception d'algorithmes quantiques est une tâche très difficile. Au cours des traitements quantiques, on ne doit pas perdre les superpositions quantiques car elles sont responsables de l'efficacité des algorithmes quantiques.

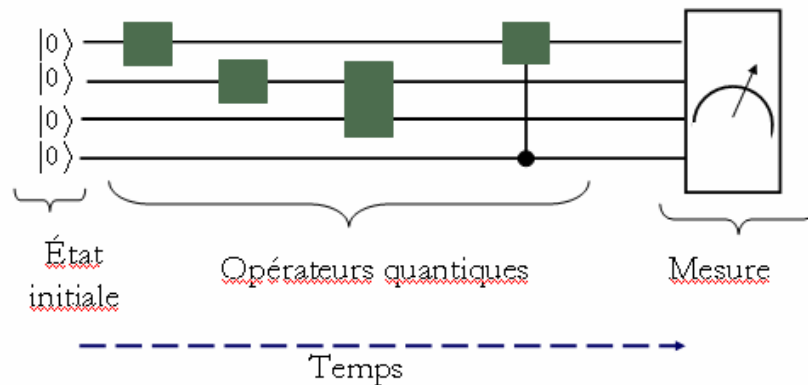


Figure 2.10 circuit d'un algorithme quantique

Nous allons, à présent, présenter brièvement les deux premiers algorithmes quantiques.

2.10.1 L'algorithme de Shor

Shor a inventé un algorithme quantique, pour factoriser un entier de n bit, dont la complexité est nettement inférieure à celle des algorithmes classiques de factorisation [Shor, 94].

La majorité des algorithmes de factorisation, y compris celui de Shor, ramène les problèmes de factorisation à des problèmes de recherche de période de fonctions.

Shor utilise le parallélisme quantique pour obtenir une superposition de toutes les valeurs de la fonction en une seule étape. Puis il calcule la transformation de Fourier de la fonction. Celle-ci ramène toutes les amplitudes de la fonction à des multiples de l'inverse de sa période avec une probabilité élevée. En mesurant le nouvel état, on obtient la période de la fonction, qui à son tour sert à factoriser l'entier à factoriser.

La transformation de Fourier ramène la fonction d'un domaine temporel vers un domaine fréquentiel, et de là elle transforme toute fonction de période r en une fonction qui ne s'annule qu'à des multiples de sa fréquence $\frac{2\pi}{r}$.

Les étapes de l'algorithme de Shor :

Déroulant l'algorithme de Shor à travers l'exemple suivant : $M=21$

Etape 1 : Parallélisme quantique

- § Choisir un entier a quelconque
- § Si a et M ne sont pas premiers entre eux alors a est un facteur de M Sinon dérouler le reste de l'algorithme
- § Soit $M^2 \leq m < 2M^2$.

On utilise le parallélisme quantique pour calculer

$$\frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} |x, f(x)\rangle \quad (2.10)$$

où $f(x) = a^x \bmod M$

Etape 2 : Recherche d'un état avec une amplitude égale à celle de f:

On mesure les derniers $\lceil \log_2 M \rceil$ qbits de (1), On obtient une valeur u
L'état après la mesure est :

$$C \sum_x g(x) |x, u\rangle \quad (2.11)$$

$$\text{Avec } g(x) = \begin{cases} 1 & \text{if } f(x) = u \\ 0 & \text{otherwise.} \end{cases}$$

$g(x)$ est donc la fonction qu'on recherche.

Etape 3 : Appliquons la Transformation de Fourier Quantique (QFT)

La QFT est appliquée aux m premiers qbits de (2) (la valeur de u ne nous intéresse pas) on obtient :

$$U_{QFT} = \sum_x g(x) |x\rangle \rightarrow \sum_c G(c) |c\rangle \quad (2.12)$$

Etape 4 : Extraction de la période

On mesure l'état (3), on obtient $v = j \frac{2^m}{r}$.

Si v est un multiple de $2^m/r$, il est facile d'extraire la période, on réduit la fraction $v/2^m$, son dénominateur q donnera la période r.

Etape 5 : Trouver un facteur de M

Si q est pair, on peut affirmer que $a^{q/2} + 1$ et $a^{q/2} - 1$ ont des facteurs non triviaux communs avec M

Etape 6 Répéter l'algorithme si nécessaire

Dans les cas suivants, on n'aboutit pas à une factorisation de M :

- la valeur v n'est pas un multiple de $2^{m/r}$
- le dénominateur q peut être un facteur de la période et non la période elle-même
- l'étape 5 donne M comme un facteur de M
- la période de $f(x) = a^x$ est impair

2.10.2 L'algorithme de Grover

L'algorithme de Grover [Grover, 97] recherche dans une liste non structurée de taille N , tous les x vérifiant une certaine propriété P .

Soit U_p la porte quantique qui implémente la fonction classique $P(x)$

$$U_p : |x, 0\rangle \leftarrow |x, P(x)\rangle$$

Premièrement, on va évaluer P pour toutes les valeurs d'entrées possibles x_i

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{n-1} |x, P(x)\rangle \quad (2.13)$$

La probabilité de mesurer l'état $|x_0, 1\rangle$ est égale à 2^{-n}

On applique une transformation à l'état (1) pour augmenter l'amplitude du vecteur $|x_0, 1\rangle$ et diminuer celle de $|x_0, 0\rangle$, puis on mesure le dernier qbit, représentant $P(x)$, le résultat sera 1 avec une probabilité élevée

Si c'est le cas, le nouvel état sera

$$\frac{1}{\sqrt{2^k}} \sum_{i=1}^k |x_i, 1\rangle,$$

où k représente le nombre de solutions.

Si on mesure maintenant les bits restants, on trouve une de ces solutions

Si la mesure de $P(x)$ donne zéro, on reprend dès le début.

2.11 Les algorithmes évolutionnaires quantiques :

Les algorithmes évolutionnaires (EAs) sont des méthodes d'optimisation et de recherche stochastiques basées sur les principes de l'évolution biologique naturelle. Comparés aux méthodes d'optimisation traditionnelles, les EAs sont robustes et peuvent être appliqués à différents problèmes réels. Le calcul évolutionnaire inclut trois grandes catégories :

- § Les algorithmes génétiques,
- § La programmation évolutionnaire et
- § Les stratégies d'évolutions.

Les EAs manipulent une population de solutions potentielles. Ils appliquent une fonction d'évaluation et un mécanisme de sélection pour générer itérativement les meilleures approximations d'une solution.

Les EAs sont caractérisés par la représentation des individus, la fonction d'évaluation et la dynamique de la population.

Les recherches dans le domaine de l'hybridation évolutionnaire quantique ont commencé vers la fin des années 90. Elles peuvent être classées en deux grandes catégories. Les approches de la première catégorie, s'intéressent à la génération de nouveaux algorithmes purement quantiques basés sur les techniques de programmation automatique [Spector, 99]. Cependant, les approches de la deuxième catégorie, visent la génération d'algorithmes évolutionnaires inspirés du calcul quantique [Han, 02], [Narayanan, 96], [Han, 00], [Han, 01], [Kim, 03], [Jang, 03], [Han, 03] et [kim, 04]. Ces derniers sont destinés à être exécutés sur des ordinateurs classiques.

Comme tout algorithme évolutionnaire, un algorithme évolutionnaire inspiré du calcul quantique (QEA) est caractérisé par :

- § la représentation des individus,
- § la fonction d'évaluation et
- § la dynamique de la population.

Nous allons maintenant détailler ces différents concepts en se basant sur les travaux de Han et Kim [Han, 02].

A. La représentation :

Les QEAs utilisent une nouvelle représentation des individus basée sur le concept du qubit. Un chromosome quantique est une chaîne de qubits. Un qubit étant la plus petite unité d'information définie par une paire de nombres (a, b) comme suit :

$$|y\rangle = \begin{pmatrix} a \\ b \end{pmatrix} \text{Où } |a|^2 + |b|^2 = 1 \quad (2.14)$$

$|a|^2$ est la probabilité que le qubit se trouve dans l'état '0' et $|b|^2$ est la probabilité que le qubit se trouve dans l'état '1'. Un qubit peut être à l'état '1', à l'état '0' ou dans une superposition des deux.

Un chromosome quantique est une chaîne de m qubits défini par :

$$\begin{pmatrix} a_1 & a_2 & \dots & a_m \\ b_1 & b_2 & \dots & b_m \end{pmatrix} \text{Où } |a_i|^2 + |b_i|^2 = 1, i=1,2,\dots, m. \quad (2.15)$$

Cette représentation quantique a l'avantage de pouvoir représenter une superposition linéaire des états.

Exemple : Soit le système à trois qubits avec trois paires d'amplitudes suivant :

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{2} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & \frac{\sqrt{3}}{2} \end{pmatrix}$$

Les états de ce système peuvent être représentés par :

$$\frac{1}{4}|000\rangle + \frac{\sqrt{3}}{4}|001\rangle - \frac{1}{4}|010\rangle - \frac{\sqrt{3}}{4}|011\rangle + \frac{1}{4}|100\rangle + \frac{\sqrt{3}}{4}|101\rangle - \frac{1}{4}|110\rangle - \frac{\sqrt{3}}{4}|111\rangle$$

Le résultat précédent implique que les probabilités de représenter les états $|000\rangle$, $|001\rangle$, $|010\rangle$, $|011\rangle$, $|100\rangle$, $|101\rangle$, $|110\rangle$ et $|111\rangle$. Sont respectivement : $\frac{1}{16}$, $\frac{3}{16}$, $\frac{1}{16}$, $\frac{3}{16}$, $\frac{1}{16}$, $\frac{3}{16}$, $\frac{1}{16}$ et $\frac{3}{16}$. Par conséquent, le système à trois qubits précédent contient l'information de 8 états.

Donc le calcul évolutionnaire utilisant une représentation quantique a une meilleure diversité que les autres représentations, du fait qu'il peut représenter une superposition linéaire des états d'une manière probabiliste. On voit bien, à travers l'exemple précédent, qu'un seul chromosome quantique est suffisant pour représenter 8 états. Cependant dans une représentation binaire au moins 8 chaînes (chromosomes) : (000), (001), (010), (011), (100), (101), (110) et (111) sont nécessaires.

B. La fonction d'adaptation :

Elle est aussi appelée fonction d'évaluation. Elle permet d'affecter à chaque individu de la population une valeur numérique permettant de distinguer les chromosomes proches de la solution optimale. Elle représente donc un outil indispensable pour la sélection des meilleurs individus de la population.

Donc chaque individu possède une valeur d'adaptation. L'idée clé est de créer un état physique enchevêtré stockant ce couple de valeur : un indice identifiant l'individu et sa valeur d'adaptation [Giraldi, 04].

C. L'observation :

Elle est aussi appelée mesure. Elle permet d'obtenir un chromosome classique à partir d'un chromosome quantique et ceci par la mesure de ses états un à un. Par exemple, si on a le chromosome :

$$C = \begin{bmatrix} 0.9501 & 0.2311 & 0.6068 & 0.4860 & 0.8913 \\ 0.0973 & 0.9466 & 0.6317 & 0.7638 & 0.2056 \end{bmatrix}$$

Sa mesure donne le vecteur binaire suivant : $[0 \ 1 \ 1 \ 1 \ 0]$

D. La dynamique de la population :

Une porte quantique est utilisée comme un opérateur de variation dans les QEAs. Elle permet la mise à jour des différents qbits représentant les chromosomes. Les qbits mis à jour doivent satisfaire la condition de normalisation: $|a'|^2 + |b'|^2 = 1$, où α' et β' sont les valeurs des qbits mis à jour.

Selon la nature du problème à traiter, cet opérateur de variation quantique peut être la porte Not, la porte controlled-Not ou la porte d'Hadamard. Mais généralement, la porte quantique la plus utilisée dans les QEAs est la rotation quantique définie par :

$$U(\Delta q_i) = \begin{bmatrix} \cos(\Delta q_i) & -\sin(\Delta q_i) \\ \sin(\Delta q_i) & \cos(\Delta q_i) \end{bmatrix} \quad (2.16)$$

Où $\Delta(q_i)$, $i=1, \dots, m$, est un angle de rotation qui fait dévier chaque qbit vers l'état 0 ou 1 dépendant de son signe. La valeur de $\Delta(q_i)$ dépend de la nature du problème à traiter.

E. La structure général d'un QEA :

Nous allons, à présent, présenter d'une façon simplifiée les principales étapes d'un QEA :

Etape 1 : Initialisation de la population quantique.

Etape 2 : Mesure de la population quantique et génération d'une population binaire.

Etape 3 : Restauration de la faisabilité des solutions binaires.

Etape 4 : Sauvegarde de la meilleure solution.

Etape 5 : Evaluation des chromosomes de la population binaire.

Etape 6 : Mise à jour de la population quantique.

Etape 7 : Si la condition d'arrêt est non satisfaite, aller à l'étape 2 sinon fin.

Il existe à présent de algorithmes génétiques quantiques qui ont enrichi l'algorithme précédemment défini avec des opérateurs de reproduction quantiques tels que la mutation et le croisement quantique ([Draa, 04] et [Talbi, 04]).

F. Les avantages des QEAs:

Les algorithmes évolutionnaires inspirés du calcul quantique (QEA) ont été récemment proposés. Ils peuvent assurer une balance entre l'exploration et l'exploitation plus efficacement que les algorithmes génétiques conventionnels. En effet, QEA peut explorer l'espace de recherche avec un nombre très réduit d'individus et exploite cet espace pour rechercher une solution globale dans un temps très court. QEA est basé sur les principes et concepts du calcul quantique comme le bit quantique et la superposition des états. Cependant QEA n'est pas un algorithme quantique, mais c'est un nouvel algorithme évolutionnaire.

2.12 Conclusion :

Afin de se familiariser avec le domaine quantique, nous avons présenté dans ce chapitre les concepts de bases de l'informatique quantique, ainsi que les principes des algorithmes quantiques. Nous avons aussi présenté l'algorithme de shor [Shor, 94] et celui de Grover[Grover, 97], les deux pionniers de l'algorithmique quantique. Après ce bref aperçu de l'informatique quantique, on remarque que ce domaine a fait appel à plusieurs spécialités tel que la physique, les mathématiques et l'informatique. L'objectif visé par cette intégration de connaissances est la réalisation d'un ordinateur quantique et son utilisation pour effectuer certains calculs beaucoup plus rapidement qu'avec un ordinateur classique.

Quelque soit l'angle disciplinaire sur lequel on l'envisage, les défis dans le domaine quantique sont de taille. Par exemple très peu d'algorithmes quantiques efficaces sont connus. Les concepts quantiques ne pourront toutefois être applicables qu'une fois un ordinateur quantique dûment réalisé. En attendant cette réalisation, et afin de tirer profit de ces nouveaux concepts quantiques, les chercheurs ont pensés à les intégrer dans des approches classiques. Un nouveau paradigme de calcul inspiré des principes de l'informatique quantique a vu le jour. Dans ce même contexte, on présentera au chapitre suivant une nouvelle approche évolutionnaire inspirée du calcul quantique.

Chapitre 3

Une Approche Evolutionnaire Quantique Pour Les Problèmes d'Optimisation Combinatoire Multiobjectif

Dans ce chapitre, nous allons présenter une nouvelle approche pour l'optimisation multiobjectif. Cette nouveauté réside dans la définition d'une représentation quantique des problèmes à traiter. Les concepts quantiques tels que le qbit, la superposition des états et les opérateurs quantiques ont été adaptés au problème du sac à dos multiobjectif. Une dynamique évolutionnaire inspirée de ces concepts est également définie pour trouver une approximation du Front de Pareto théorique.

Sommaire

3.1 Introduction	56
3.2 Formulation du problème du sac à dos multiobjectif (MOKP)	57
3.3 Définition d'un noyau évolutionnaire quantique	58
3.3.1 La représentation quantique du MOKP	58
3.3.2 Les opérateurs de variations quantiques	59
3.4 L'approche proposée	61
3.4.1 Fonctionnement de l'algorithme	63
3.4.2 Implémentation et évaluation de l'approche proposée	68
3.5 Conclusion	73

3.1 Introduction

Les problèmes d'optimisation multiobjectif combinatoire sont difficiles à résoudre et nécessitent l'utilisation de méthodes très performantes. Ainsi, malgré le nombre important de méthodes d'optimisation multiobjectif disponibles, il n'existe pas encore de méthodes puissantes pouvant surpasser les difficultés des MOPs. Ce domaine reste toujours ouvert à de nouvelles idées et philosophies de résolutions. Dans ce chapitre nous allons proposer une nouvelle approche pour la résolution des MOPs utilisant le paradigme évolutionnaire quantique. Comme tout algorithme évolutionnaire, l'approche proposée est caractérisée par la représentation des individus, la fonction d'évaluation et la dynamique de la population.

Pour un souci de clarté, nous allons commencer par définir la structure générale d'un noyau quantique qu'on va développer et décortiquer par la suite.

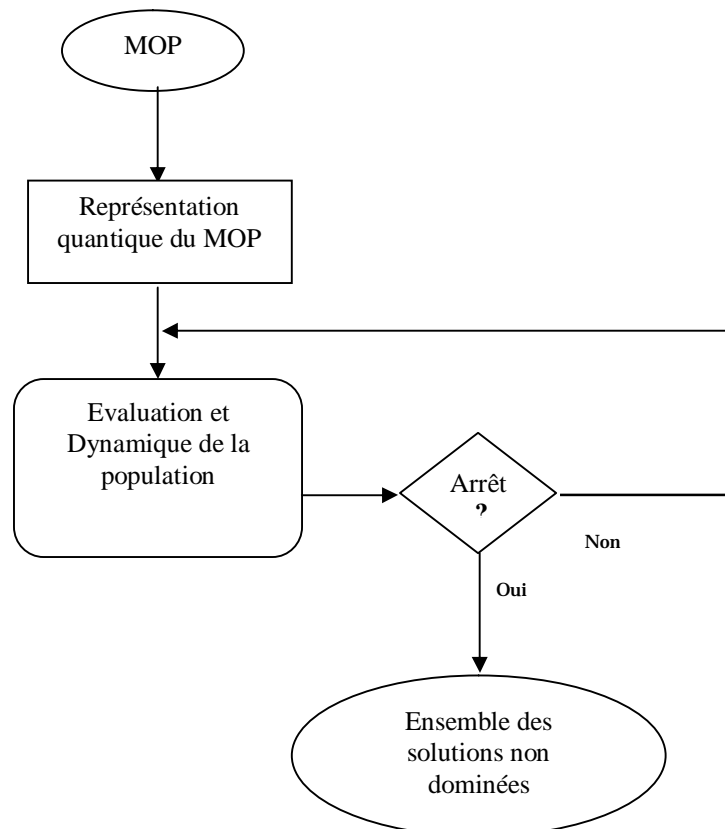


Figure 3.1 Noyau quantique évolutionnaire pour un MOP

On commencera tout d'abord par la formulation du problème du sac à dos multiobjectif (MOKP), l'un des problèmes combinatoires les plus étudiés dans la communauté multiobjectif, qu'on a choisit pour valider l'approche proposée puis on définira les différents éléments composants le noyau quantique et on finira par présenter et discuter les résultats obtenus.

3.2 Formulation du problème du sac à dos multiobjectif (MOKP)

Le sac à dos multidimensionnel multiobjectif (MOKP) est l'un des problèmes les plus réputés dans la communauté des problèmes d'optimisation combinatoire multiobjectif, il peut modéliser de nombreux problèmes réels, comme la répartition de budget et l'allocation de ressources. Nous avons donc choisi ce problème pour tester notre approche.

Plusieurs algorithmes heuristiques ont été développés pour résoudre le MOKP. On peut citer à titre d'exemple l'algorithme génétique de Zitzler SPEA [Zitzler, 98], la recherche tabou de Barichard [Barichard, 03a] et celle de Abdelaziz [Abdelaziz, 97], etc....

Le problème du MOKP est défini de la façon suivante : étant donné un ensemble d'objets, à chacun d'entre eux sont associés des valeurs de profit ainsi que des poids. Le problème du sac à dos multiobjectif consiste à sélectionner un sous ensemble d'objets maximisant une fonction multiobjectif, exprimée en fonction des valeurs de profit, tout en respectant un ensemble de contraintes du type sac à dos [Barichard, 03a]. Formellement, ce problème peut être décrit comme suit :

$$\begin{array}{l}
 \max \quad z^j(x) = \sum_{i=1}^m c_i^j x_i, j = 1, \dots, k \\
 \text{telle que} \quad \sum_{i=1}^m w_i^l x_i \leq b_l, l = 1, \dots, q \\
 x_i \in \{0, 1\}, i = 1, \dots, m
 \end{array} \tag{3.1}$$

Où

m : désigne le nombre total d'objets,

$x_i, i=1, \dots, m$: désignent les variables de décision,

k : le nombre de fonctions objectif,

z^j : la $j^{\text{ème}}$ composante de la fonction multiobjectif z ,

c_i^j : désigne le profit de l'objet i par rapport à l'objectif j

w_i^l : désigne le poids de l'objet i par rapport à la contrainte l .

et q : le nombre de contraintes de type sac à dos du problème.

3.3 Définition d'un noyau évolutionnaire quantique

Après avoir défini le problème du MOKP, on va essayer maintenant de le résoudre en exploitant les concepts de l'informatique quantique. Pour cela, on doit :

- § définir une représentation quantique d'un individu solution du MOKP,
- § puis, définir l'ensemble des opérateurs génétiques quantiques, qui permettent de faire évoluer la population quantique d'une génération à l'autre.

3.3.1 La représentation quantique du MOKP

Une solution potentielle du MOKP est donnée par le vecteur de décision

$$X = [x_1, x_2, \dots, x_m]^t. \quad (3.2)$$

Chaque variable x_i prend la valeur 1 si l'objet i correspondant est présent dans le sac à dos et prend la valeur 0 sinon. Pour trouver une représentation quantique adéquate à ce vecteur solution, on est amené à définir des probabilités de présence ou d'absence pour chaque objet dans le sac à dos. Chaque objet peut être naturellement représenté par une paire de nombres (a, b) permettant ainsi de définir le qbit :

$$|\Psi\rangle = a|0\rangle + b|1\rangle \quad (3.3)$$

Où a et b sont deux nombres vérifiant la condition $|a|^2 + |b|^2 = 1$.

$|a|^2$ donne la probabilité pour que le qbit prennent la valeur 0, et donc la variable de décision prend aussi la valeur 0. Tandis que la valeur $|b|^2$ donne la probabilité pour que le qbit prennent la valeur 1, et donc la variable de décision prend aussi la valeur 1. Le vecteur de décision X peut être représenté par un registre quantique q contenant m qbits de la façon suivante :

$$q = \begin{pmatrix} a_1 & a_2 & \dots & a_m \\ b_1 & b_2 & \dots & b_m \end{pmatrix} \quad (3.4)$$

Cette représentation est avantageuse car elle permet de représenter une superposition linéaire de tous les états possibles. Toutes les combinaisons possibles des valeurs du vecteur de décision X peuvent être dérivées à partir de cette simple représentation. Pour plus de diversité, on maintient une population de solutions quantiques. Donc à chaque génération de notre approche évolutionnaire, on manipule une matrice d'individus quantiques notée $Q(t)$:

$$Q(t) = \{q_1^t, q_2^t, \dots, q_n^t\} \tag{3.5}$$

Où n représente la taille de la population, et $q_j^t, j=1..n$ est l'un individu quantique définit par :

$$\left(\begin{array}{c|c|c|c} a_{j1}^t & a_{j2}^t & \dots & a_{jm}^t \\ b_{j1}^t & b_{j2}^t & \dots & b_{jm}^t \end{array} \right) \tag{3.6}$$

3.3.2 Les opérateurs de variations quantiques :

Afin de faire évoluer la population quantique de générations en générations, l'ensemble suivant des opérateurs quantiques est appliqué.

§ L'opération de mesure quantique :

Une fois la population quantique initialisée, on doit la mesurer pour extraire la population binaire correspondante. Malgré la superposition des états présente dans un qbit, la mesure ne produit qu'une seule valeur binaire dépendant des valeurs de ses probabilités $|a^2|$ et $|b^2|$. La figure 3.2 montre le passage d'une population quantique à une population binaire correspondante.

Chromosome 1	0,88	0,1	0,66	0,56	0,647
	0,47497368	0,99498744	0,7512656	0,82849261	0,76249
Chromosome 2	0,22	0,89	0,96	0,55	0,789
	0,97549987	0,45596052	0,28	0,83516465	0,6143932
Chromosome 3	0,887	0,265	0,56	0,11	0,22
	0,46176942	0,96424841	0,82849261	0,99393159	0,97549987

(a)

Chromosome 1	0	1	1	1	1
Chromosome 2	1	0	0	1	0
Chromosome 3	0	1	1	1	1

(b)

Figure 3.2 Exemple de mesure, (a) population quantique, (b) population binaire correspondante.

§ L'interférence quantique :

L'interférence quantique peut être défini comme étant une rotation spéciale (Figure 3.3) qui peut être utilisé comme une porte quantique dans les algorithmes évolutionnaires. C'est un opérateur d'intensification, qui permet de ramener progressivement les chromosomes quantiques au voisinage de la solution trouvée. Cette porte peut être définie par :

$$U(dq_i) = \begin{bmatrix} \cos(dq_i) & -\sin(dq_i) \\ \sin(dq_i) & \cos(dq_i) \end{bmatrix} \quad (3.7)$$

Où : $dq_i, i = 1, 2, \dots, m$ est un angle de rotation faisant dévier chaque qubit vers l'état 0 ou 1 dépendant de son signe. dq_i est un paramètre de réglage devant être choisis expérimentalement. Le signe de dq_i change selon la table présentée dans la figure 3.4.

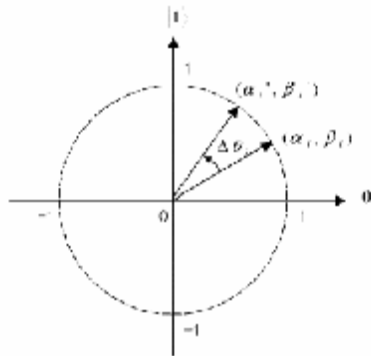


Figure 3.3 Rotation d'un qubit

Angle	Valeur du bit de référence	β	A
$+\delta\theta$	1	> 0	> 0
$-\delta\theta$	0	> 0	> 0
$-\delta\theta$	1	< 0	> 0
$+\delta\theta$	0	< 0	> 0
$-\delta\theta$	1	> 0	< 0
$+\delta\theta$	0	> 0	< 0
$+\delta\theta$	1	< 0	< 0
$-\delta\theta$	0	< 0	< 0

Figure 3.4 Lookup table.

§ La mutation quantique :

La mutation classique opère comme une petite perturbation qui inverse le bit muté. Dans ce travail on a utilisé un type de mutation appelé *mutation quantique*. Cette mutation est aussi une perturbation mais elle opère sur les probabilité d'un qubit du chromosome de la façon suivante: si ce qubit est représenté par: $\alpha|0\rangle + \beta|1\rangle$., la mutation génère le qubit $\beta|0\rangle + \alpha|1\rangle$. Dans la figure qui suit le chromosome quantique

est muté au niveau de point 3. On remarque que la mutation quantique est le même opérateur que le NOT abordé dans le chapitre précédent.

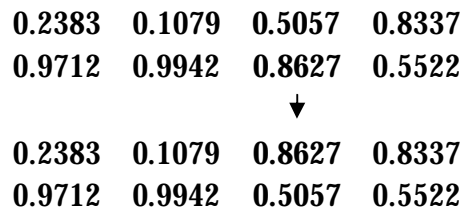


Figure 3.5 Mutation quantique

§ Le croisement quantique :

Le croisement quantique est similaire à un croisement classique dans le principe. Cependant il opère sur des chromosomes quantiques. Donc c'est un croisement qui opère sur des matrices de probabilités et génère des matrices de probabilités. Par exemple le croisement de la figure 3.6 aura comme entrée les deux chromosomes quantiques *a* et *b* et produira en sortie les deux chromosomes quantiques *c* et *d*. Le point de croisement est choisi par l'une des techniques connues, pour cet exemple le croisement est fait au point 2.

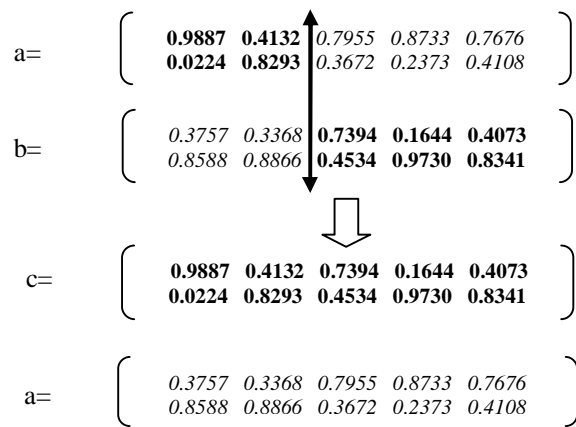


Figure 3.6 Croisement quantique.

3.4 L'approche proposée : QMOKP (a Quantum approach for Multiobjective Knapsack Problem)

Maintenant, on va décrire comment le noyau quantique défini précédemment, peut être exploitée et décortiqué afin de définir une nouvelle approche évolutionnaire quantique pour la résolution du problème multiobjectif MOKP. Les résultats d'une première version de cette approche ont été acceptables et très prometteurs [Meshoul, 05a]. Après les étapes de tests intensifs et de validation, nous avons amélioré cette

première version et généré une seconde plus élaborée [Meshoul, 05b]. Les principales différences de la version2 de QMOKP par rapport à sa version précédentes peuvent être résumées en :

- § Une amélioration au niveau du calcul des valeurs d'adaptation des individus,
- § Une nouvelle technique d'archivage.

QMOKP est un algorithme évolutionnaire quantique élitiste. En effet, l'approche proposée maintient un archive externe contenant le meilleur ensemble des solutions non dominées rencontré durant le processus de recherche. QMOKP appartient à la classe des approches Pareto car elle utilise un mécanisme de sélection basé sur la relation de dominance Pareto.

En plus de l'archive externe $A(t)$, QMOKP maintient, une population quantique $Q(t)$, une population binaire $B(t)$ et une population temporaire dénotée par $Temp(t)$. On a utilisé la relation de dominance de Pareto pour assigner des valeurs d'adaptation aux différents individus de la population. Pour plus de clarté, on commence par donner le schéma général de notre algorithme, puis on détaillera les différentes étapes par la suite.

QMOKP Algorithm:

INPUT: Problem specification

- Step 1: Generate an empty archive $A(0)$.
- Step 2: Generate initial population $Q(0)$. Set $t=0$.
- Step 3: If ($t > \max_gen$) Output the dominated set $A(t)$. Stop.
- Step 4: Make $P(t)$ by observing the states of $Q(t)$.
- Step 5: Repair $P(t)$ for restoring the feasibility of the solutions in $P(t)$.
- Step 6: Copy non dominated solutions of $P(t)$ and $A(t)$ in $A(t+1)$.
- Step 7: Compute the fitness of each individual in $P(t)$ and $A(t+1)$.
- Step 8: Select individuals from $P(t)$ and $A(t+1)$ and store them in $Temp(t)$.
- Step 9: Interfere $Q(t)$ toward $Temp(t)$.
- Step 10: Apply quantum crossover and quantum mutation.
Set $t=t+1$ and go to step 3.

OUTPUT: the set $A(t)$ of non dominated solutions.

3.4.1 Fonctionnement :

La figure 3.7 illustre le schéma général de fonctionnement de l'algorithme QMOKP.

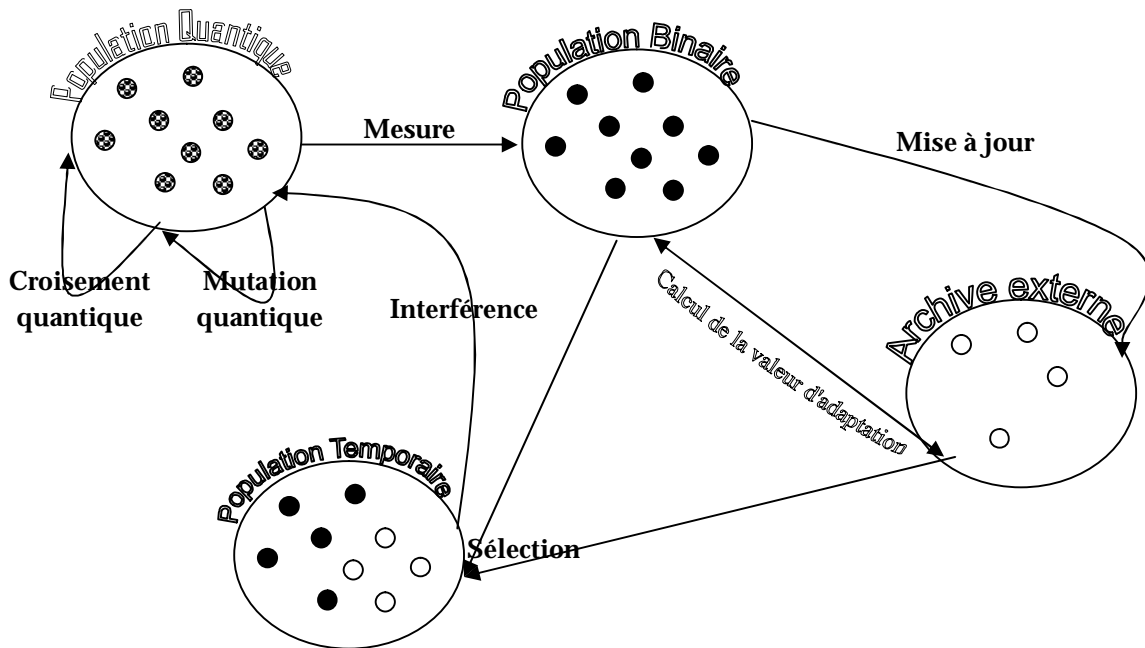


Figure 3.7 Fonctionnement général de QMOKP.

La spécification du problème inclut la définition des fonctions objectives, les contraintes du problème et l'ensemble des paramètres du problème tel que la taille m du vecteur de décision et le nombre k des fonctions objectives.

La première étape de QMOKP permet d'initialiser à l'ensemble vide l'archive externe $A(0)$. Ensuite, la population quantique $Q(0)$ est générée. Tous les a_i^0 et b_i^0 pour $i=1,2,\dots,m$ de tous les $q_j^0 = q_j^t|_{t=0}$, $j=1,2,\dots,n$ sont initialisés à $1/\sqrt{2}$. Ceci implique qu'un individu q_j^0 représente une superposition linéaire de tous les états possibles avec la même probabilité :

$$|y_{q_j^0}\rangle = \sum_{k=1}^{2^m} \frac{1}{\sqrt{2^m}} |X_k\rangle \quad (3.8)$$

Où X_k est le $k^{\text{ième}}$ état représenté par la chaîne binaire $(b_1 b_2 \dots b_m)$, avec x_i , $i = 1,2,\dots, m$ est soit '0', soit '1' selon la probabilité de $|a_i^0|^2$ ou $|b_i^0|^2$ respectivement.

Une fois $Q(0)$ initialisé, l'opération mesure quantique est appliquée. Elle consiste à dériver une image binaire de la population quantique $Q(t)$. Cette représentation binaire est dénotée par $P(t)$ et elle est défini par :

$$P(t) = \{x_1^t, x_2^t, \dots, x_n^t\} \quad (3.9)$$

Une solution binaire, $x_j^t, j=1,2,\dots,n$ est une chaîne binaire de longueur m , qui est construite par la sélection de '0' ou '1' pour tout bit en utilisant la probabilité $|a_i^0|^2$ ou $|b_i^0|^2$. Par la suite, la faisabilité des solutions obtenues est vérifiée. Si un individu binaire de $P(t)$ viole la contrainte de capacité, l'algorithme tente de réparer et restaurer la faisabilité des solutions dans $P(t)$. On applique ensuite la relation de dominance de Pareto à tous les individus de la population binaire et de l'archive externe courant. On extrait ensuite l'ensemble des solutions non dominées pour mettre à jour l'archive externe. Les nouvelles solutions non dominées rencontrées sont rajoutée à l'archive et les anciennes solutions de l'archive dominées par ces nouveau individus sont supprimées. Si la taille de l'archive excède une certaine limite, une procédure de réduction appropriée est déclenchée. Dans l'étape suivante, on calcule les valeurs d'adaptation des individus dans $A(t)$ et $P(t)$ selon le principe de la méthode SPEA2 [Eckart, 01]. Les n meilleurs individus de $A(t+1)$ sont sélectionnés et stockés dans la population temporaire $Temp(t)$. si la taille de $A(t+1)$ est inférieure à n , les individus manquants dans $Temp(t)$ sont complétés à partir de $P(t)$. $Temp(t)$ et $P(t)$ ont la même longueur n .

La population temporaire est donc construite à partir des meilleurs individus de $P(t)$ et $A(t+1)$. Elle est utilisée pour générer une nouvelle population quantique $Q(t)$. et ceci en appliquant une opération quantique spéciale appelée interférence. C'est une porte de rotation spéciale, qui vise à intensifier la recherche dans le voisinage des solutions stockées dans $Temp(t)$.

Finalement, les individus quantiques dans $Q(t)$ sont sélectionnées pour des opérations de croisement et de mutation quantique.

Toutes les étapes décrites précédemment sont appliquées itérativement, et le processus évolue jusqu'à atteindre un nombre maximum d'itérations. La population externe est présentée comme étant la solution du problème considéré.

Nous présentant maintenant quelques détails concernant la fonction d'évaluation, les techniques d'archivage et d'interférence utilisées dans QMOKP.

§ Fonction d'évaluation :

Pour chaque individu i de $P(t)$ et $A(t+1)$, la valeur d'adaptation est calculé de la façon suivante :

1. Soit S_i la force de l'individu i . S_i est le nombre des individus de $P(t)$ et $A(t+1)$ dominés par l'individu i .

$$\forall i \in P(t) \cup A(t+1) \quad S_i = |\{j \in P(t) \cup A(t+1) / i \mathbf{f} j\}| \quad (3.10)$$

Où la notation $|\cdot|$ dénote la cardinalité d'un ensemble et le symbole \mathbf{f} représente la relation de dominance. Ainsi $i \mathbf{f} j$ se lit i domine j .

2. calculer la valeur d'adaptation brute d'un individu i . elle est dénotée par R_i et représente la somme des forces des individus dominant i :

$$R_i = \sum_{\substack{j \in P(t) \cup A(t+1) \\ j \mathbf{f} i}} S_j \quad (3.11)$$

Il est important de noter que dans ce cas les meilleurs individus sont ceux dotés des valeurs d'adaptations les plus petites.

Ainsi, un individu ayant une valeur nulle de R_i correspond à un individu i non dominé. Une valeur élevée de R_i signifie que l'individu i est dominé par plusieurs éléments.

Cette méthode adoptée pour le calcul de la valeur d'adaptation brute R_i d'un individu i procure une sorte de niche (le sharing) basée sur le concept de la dominance de Pareto. Malheureusement, on peut constater que ce mécanisme devient inefficace quand la majorité des individus ne se dominant pas entre eux. La solution à ce problème consiste à introduire une technique d'estimation de densité discriminant entre des individus ayant des valeurs d'adaptation brutes égales.

3. calculer la densité D_i définit par :

$$D_i = \frac{1}{S_i^k + 2} \quad \text{Where } k = \sqrt{|P(t)| + |A(t+1)|} \quad (3.12)$$

S_i^k est la distance entre l'individu i et sont $k^{\text{ième}}$ voisin, qui est sélectionner à partir d'une liste de distance, triée dans l'espace des objectives, entre l'individu i et tous les autres individus de $P(t)$ et $A(t+1)$.

4. finalement, la valeur d'adaptation d'un individu i est donnée par :

$$F_i = R_i + D_i \quad (3.13)$$

On peut constater qu'un individu I est non dominé si sa valeur d'adaptation F_i est inférieure à 1.

§ Technique de réduction de l'archive externe :

Le nombre d'individus à stocker dans l'archive externe doit être constant. Soit A_r cette constante. L'algorithme débute avec un archive externe vide. Puis à travers les différentes itérations, cet archive est mis à jour continuellement. En premier temps, l'ensemble des solutions non dominées rencontrées au cours du processus de recherche vient s'ajouter à l'archive et ses anciennes solutions dominées par les nouveaux arrivants sont supprimées. Lors de l'insertion des solutions dans l'archive deux situations peuvent se présenter : soit l'archive est trop petit ($|A(t+1)| < A_r$), soit très grand ($|A(t+1)| > A_r$). Dans le premier cas, les meilleurs $A_r - |A(t+1)|$ individus de la population $P(t)$ sont sélectionnés pour compléter l'archive. Dans le deuxième cas, la taille de l'archive dépasse la constante A_r . On fait appel, alors, à une procédure de réduction spécifique qui permet de le tronquer. Ainsi, à chaque itération de cette procédure, l'individu le plus proche d'un autre individu est éliminé. La figure (3.8) illustre cette opération.

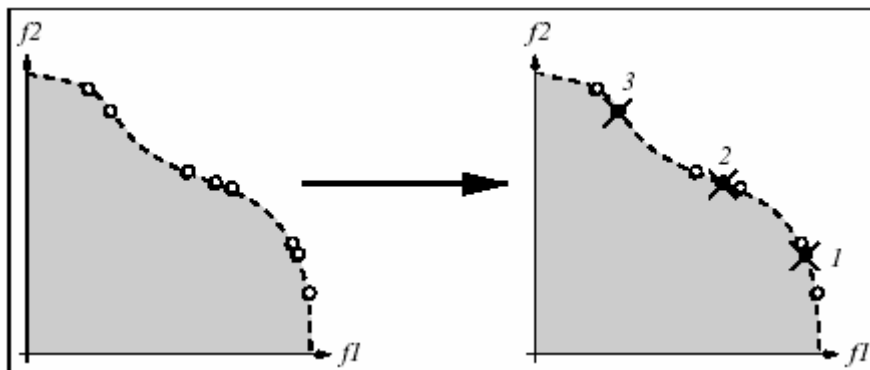


Figure 3.8 Exemple de réduction d'archive. A droite, l'archive initial. A gauche, l'archive après sa réduction. Les individus sont éliminés dans l'ordre indiqué (1-2-3). On suppose $A_r=5$.

§ Techniques de la mise à jour de la population quantique :

Plusieurs techniques d'interférences ont été testées. La première technique opère comme suit. Pour chaque chromosome q'_j , $j=1\dots,n$ de $Q(t)$ va tenter de dévier graduellement vers le chromosome a'_i de $Temp(t)$ le moins distant. La distance

utilisée est celle de Hamming, elle est calculée comme suit : chaque différence entre q_j^t et a_i^t compte pour plus 1. la déviation du chromosome quantique q_j^t vers le chromosome binaire a_i^t est réalisé à l'aide d'opération d'interférence détaillée précédemment dans le noyau quantique.

Cependant, la deuxième technique tente de faire dévier la nouvelle population quantique $Q(t+1)$ vers la population temporaire $Temp(t)$. Cette déviation est réalisée en utilisant une porte quantique spécifique. Cette dernière est notée U et est appliquée à chaque qbit j de chaque chromosome i , $q_{ij}^t = \begin{bmatrix} a_{ij} \\ b_{ij} \end{bmatrix}$ à l'itération t . Elle est définie de la façon suivante :

$$\left\{ \begin{array}{l} U(q_{ij}^{t+1}) = \begin{bmatrix} \beta_{ij} \\ \alpha_{ij} \end{bmatrix} \text{ Si } [(\alpha_{ij}^2 < \beta_{ij}^2) \wedge Temp(i,j) = 0] \vee [(\alpha_{ij}^2 > \beta_{ij}^2) \wedge Temp(i,j) = 1] \\ \text{et} \\ U(q_{ij}^{t+1}) = q_{ij}^t \text{ Sinon} \end{array} \right. \quad (3.14)$$

Pour plus de clarté, prenons l'exemple suivant : on suppose que le chromosome i de la population temporaire à l'itération i est :

$$Temp^t(i) = [0 \quad 1 \quad , \quad \dots \quad , \quad 0]$$

et le $i^{\text{ème}}$ chromosome quantique q_i^t de la population quantique $Q(t)$ est :

$$q_i^t = \begin{bmatrix} 0.2383 & 0.1079 & , & \dots & , & 0.8337 \\ 0.9712 & 0.9942 & , & \dots & , & 0.5522 \end{bmatrix}$$

On remarque que le premier bit de $Temp^t(i)$ a la valeur 0 et $0.2383^2 < 0.9712^2$. Dans ce cas, on inverse les deux probabilités α et β du premier qbit de q_i^t . Et on procède de même pour tous les autres bits et qbits de $Temp(t)$ et $Q(t)$ respectivement. On aura à la fin de l'opération d'interférence :

$$q_i^{t+1} = \begin{bmatrix} 0.9712 & 0.1079 & , & \dots & , & 0.8337 \\ 0.2383 & 0.9942 & , & \dots & , & 0.5522 \end{bmatrix}$$

Après des tests expérimentaux des deux techniques proposées, il s'est avéré que la deuxième technique est plus intéressante et fournit de meilleurs résultats pour la

résolution du MOKP. Dans la section suivante, on ne présentera donc que les résultats relatifs à cette technique.

3.4.2 Implémentation et évaluation de l'approche proposée :

Pour l'implémentation de l'approche proposée, on a choisi le langage MATLAB. Cet outil facile a été spécialement conçu pour les différents calculs matriciels. Il offre aussi une variété de possibilités graphiques. Nous avons donc tout le bagage nécessaire pour l'implémentation de QMOKP.

Nous allons, à présent, présenter les résultats expérimentaux obtenus par l'approche proposée. Pour une étude comparative nous avons choisi une des méthodes les plus performantes de l'optimisation multiobjectif SPEA. Plusieurs études comparatives avec d'autres algorithmes évolutionnaires multiobjectifs ont été établies et prouvent l'efficacité de SPEA [Zitzler, 99], elle a aussi été un point de référence dans une variété de nouvelles investigations.

Le premier jeu de données a été défini par Eckart Zitzler et Marco Laumanns dans [Eckart, 03]. On a un MOKP bi-objectif avec deux contraintes et 100 objets par sac à dos.

Table 1 : les valeurs obtenues pour la couverture C et le rapport d'hypervolume RH à travers les itérations.

	itérations	N	RH	C(S,SPEA)	C(SPEA,S)
Figure1	1000	10	1.0200	0.4681	0.36454
Figure2	800	10	1.0247	0.3404	0.566
Figure3	700	10	1.0134	0.3191	0.6304
Figure4	500	10	1.0110	0.2979	0.6000
Figure5	500	20	1.0245	0.5745	0.3300
Figure6	1000	20	1.0356	0.5868	0.3182

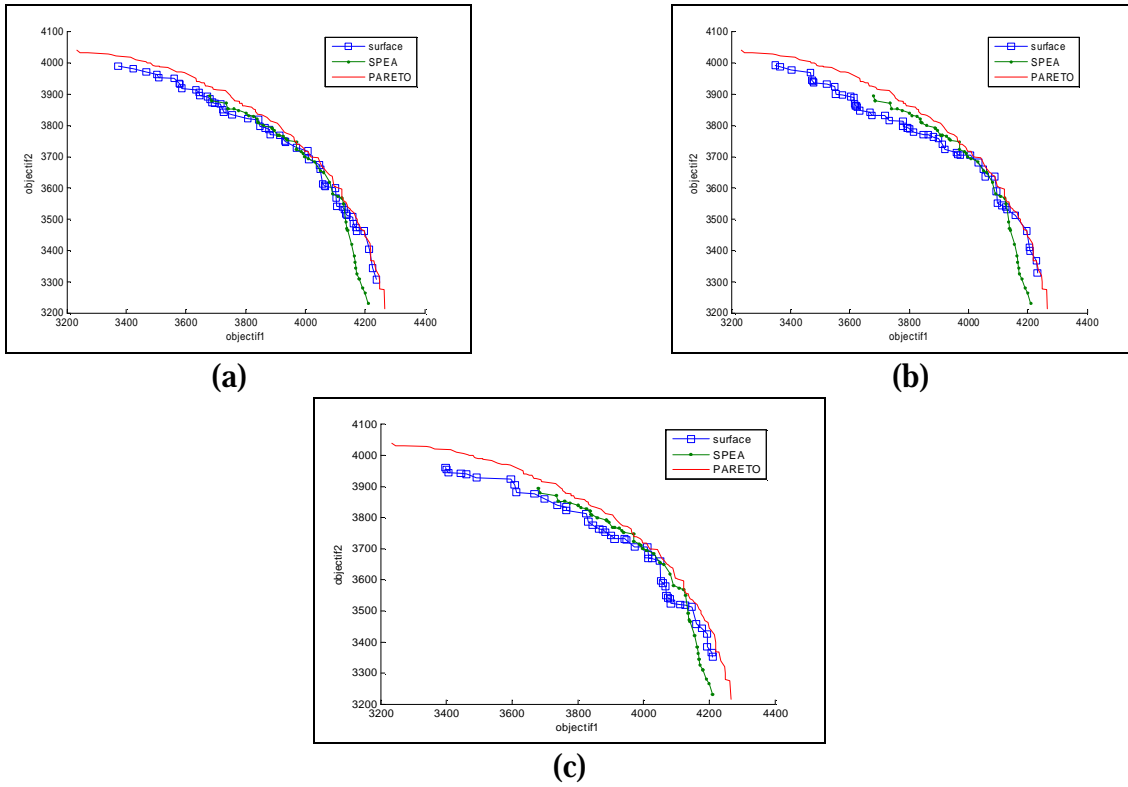


Figure 3.9 : Evolution de l'algorithme proposé à travers : (a) 1000 (b) 800 (c) 700 itérations

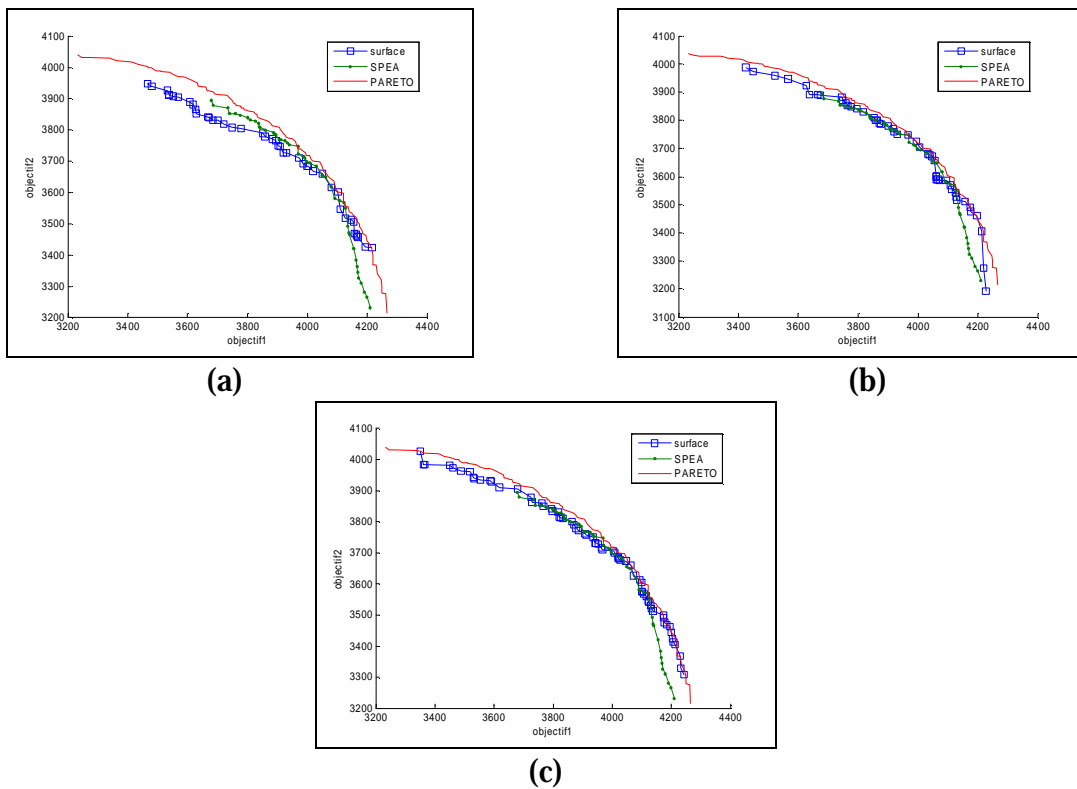


Figure 3.10 : Evolution de l'algorithme proposé à travers : (a) 500-10 (b) 500-20 (c) 1000-20

La figure 3.9 montre l'évolution de l'approche proposée à travers les itérations. Elle illustre les valeurs obtenues des deux objectives pendant le processus de recherche. Une comparaison avec les résultats de la méthode SPEA est élaborée. On peut voir clairement que la méthode proposée se rapproche de plus en plus du front de Pareto théorique et peut compléter l'ensemble des solutions non dominées de SPEA. On doit encore prendre en considération le fait que les algorithmes évolutionnaires classiques nécessitent beaucoup plus d'individus et un grand nombre d'itérations pour aboutir à des solutions acceptables. On peut remarquer qu'une bonne approximation du front de Pareto est obtenue après un nombre d'itérations inférieur à 1000. Pour mieux étudier les performances de cette approche, on s'est intéressé à l'influence de la taille de la population. On n'a pas considéré un seul chromosome quantique mais plusieurs puisqu'on travaille sur un ordinateur classique et non quantique. La figure 3.10 présente les résultats obtenus pour une taille de population égale à 10 et 20 individus avec 500 et 1000 itérations. Dans la table 1, on présente les valeurs obtenues pour les deux mesures définies dans l'état de l'art. Dans cette table S désigne l'approche QMOKP, RH l'hypervolume et C la métrique proposée par Zitzler. Pour plus de clarté, on va essayer de donner quelques interprétations aux valeurs numériques des métriques RH et C regroupées dans Table 1.

§ Interprétation des résultats de la Table1 :

a./ La métrique RH :

L'hypervolume a été défini dans le chapitre1 au niveau de la section (1.7.1). Il permet de mesurer la surface occupée par un ensemble de solutions non dominées. Le rapport d'hypervolume est une métrique relative permettant de comparer les surfaces de deux ensembles de solutions non dominées. La figure 3.11 présente un exemple simple pour faciliter l'interprétation des résultats obtenus. Il a été tiré de [Collette, 02]. En effet, pour l'ensemble de solutions non dominées E1, l'hypersurface est calculée comme suit :

$$H_{E1} = S_1 + S_2 + S_3 + S_4 = 4 * 4 + 3 * (6 - 4) + 2 * (8 - 6) + 1.5 * (10 - 8) = 29.$$

Pour l'ensemble E2 :

$$H_{E2} = A_1 + A_2 + A_3 = 5 * 4 + 3 * (6 - 4) + 2.5 * (9 - 6) = 33.5.$$

La valeur de l'hypervolume est donc : $RH = \frac{H_{E1}}{H_{E2}} = \frac{29}{33.5} = 1.155$

La valeur RH dépasse légèrement 1, On conclut donc que les deux ensembles E1 et E2 occupent des surfaces presque égales. La surface occupée par E1 est légèrement supérieure à celle de E2.

Après cet exemple illustratif, on est maintenant capable d'interpréter les résultats du rapport de l'hypervolume obtenus (voir table 1). En effet, toutes les valeurs de RH obtenus (1.0200, 1.0247, 1.0134, 1.0110, 1.0245, 1.0356) sont très proches de l'unité. On peut conclure que QMOKP et SPEA ont produit deux ensembles de solutions non dominées occupants des surfaces presque égales.

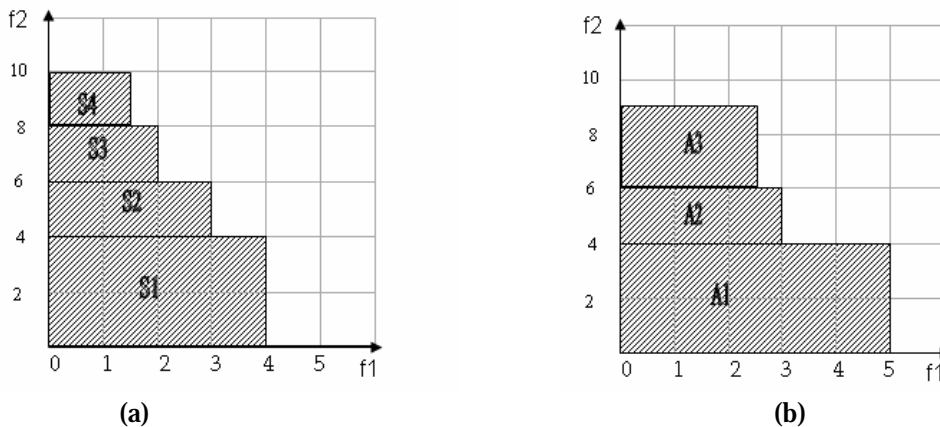


Figure 3.11 Exemple illustratif de l'hypervolume. (a) la surface de l'ensemble de solutions non dominées E1. (b) la surface de l'ensemble de solutions non dominées E2

b./ La métrique C :

Comme elle a été présentée dans la section 1.7.2, la métrique C est une métrique relative qu'on a utilisé pour comparer la méthode proposée est SPEA. En effet, la valeur $C(S,SPEA)$ donne un nombre compris entre 0 et 1, indiquant la proportion de la surface de compromis trouvée par SPEA qui est dominée par la surface de compromis trouvée par QMOKP. Ainsi, lors d'une exécution de QMOKP avec 1000 itérations et 10 individus, on a trouvé $C(S,SPEA)= 0.4681$ et $C(SPEA,S) =0.36454$. Ses résultats sont lus de la façon suivante : 46.81% des solutions de SPEA sont dominées par les solutions de QMOKP et 36.454% des solutions de QMOKP sont dominées par SPEA. On doit remarquer que la somme $C(S,SPEA)+ C(SPEA,S)$ n'est pas nécessairement égale à 1. En effet, $1- C(S,SPEA)+ C(SPEA,S)$ mesure le nombre de solutions de SPEA et QMOKP qui sont non dominées entre elles.

Maintenant, on est en mesure d'interpréter et analyser les résultats présentés dans la table 1. En effet, on remarque que parmi six exécutions de QMOKP, dans trois cas

$C(\text{SPEA},S) > C(S,\text{SPEA})$ et dans les trois autres cas $C(\text{SPEA},S) > C(S,\text{SPEA})$. Ceci montre la complémentarité de notre méthode avec SPEA.

Une analyse plus poussée de ses résultats montre que les meilleurs résultats trouvés correspondent soit à une taille de population égale à 10, soit à un nombre d'itérations égale à 1000 (figure 3.12). Un réglage de ses paramètres peut donc fournir de meilleurs résultats.

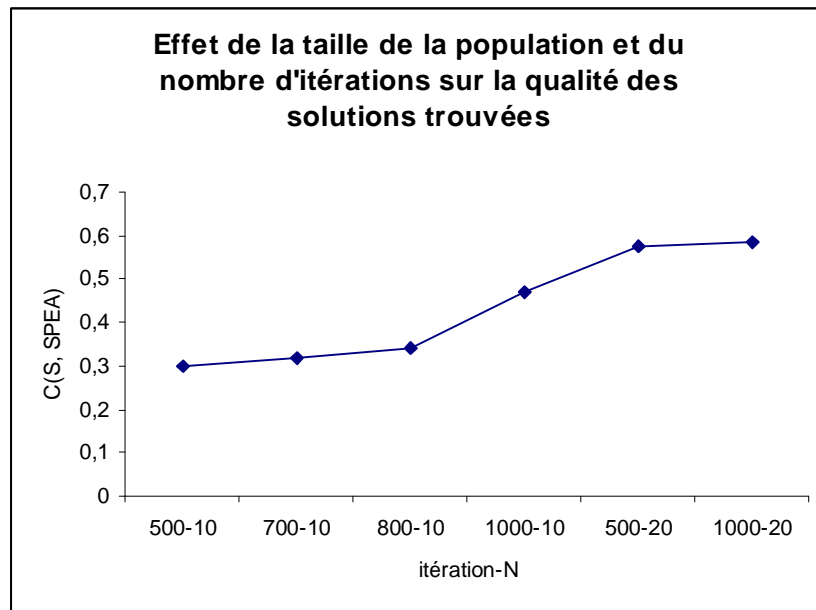


Figure 3.12 Effet de la taille de la population et du nombre d'itérations sur la qualité des solutions trouvées.

On a aussi testé notre algorithme avec les jeux de données défini dans MCDM [MCDM, 02]. On a un problème MOKP bi-objectif, une contrainte et 50 objets par sac à dos. Les résultats étaient très prometteurs.

Nous allons, maintenant, résumer les avantages de QMOKP.

Les avantages de l'approche proposée :

Cette approche tire profit des trois aspects qu'elle utilise, l'aspect évolutionnaire, quantique et multiobjectif. En effet, c'est une approche Pareto élitiste quantique présentant les avantages suivants :

- § Elle est flexible : la formulation du problème à résoudre peut être facilement modifié ou étendu.
- § Elle est multiobjective : l'espace des solutions peut être exploré en une seule exécution.

- § La résolution multiobjective du MOKP ne nécessite l'introduction d'aucun nouveau paramètre dont le réglage sera difficile.
- § Elle offre au décideur une multitude de solutions.
- § Elle n'utilise qu'un nombre très réduit d'individus dans la population manipulée. Tout en assurant une bonne diversité.

3.5 Conclusion

Dans ce chapitre, on a proposé une approche pour l'optimisation multiobjectif. Les grandes lignes de cette approche consistent essentiellement dans la représentation de la population manipulée en utilisant un codage quantique et en définissant des opérateurs spécifiques tels que la mesure et l'interférence. En comparaison avec un algorithme évolutionnaire, l'approche basée quantique a l'avantage d'avoir plus de diversité. Toutes les solutions existent avec seulement un seul chromosome. Les résultats obtenus sont très encourageant et montre la faisabilité et aussi l'efficacité de l'approche proposée.

Donc on peut espérer son utilisation pour d'autres problèmes complexes tels que les problèmes du traitement d'images et de vision artificielle. Dans le prochain chapitre nous allons décrire une application de cette approche au problème de recalage d'images, l'un des problèmes cruciales dans le domaine du traitement d'image.

Chapitre 4

Un Algorithme Evolutionnaire Quantique Bi-objectif pour le recalage d'images

Dans ce chapitre, on s'intéresse à la résolution d'un problème réel en vision artificielle : le recalage d'images. Dans un premier temps, nous proposons une formulation de ce problème en termes d'optimisation multiobjectif. Dans un second temps, nous adoptons l'approche proposée dans le chapitre précédent à ce cas d'étude.

Sommaire

4.1 Introduction	75
4.2 Le recalage d'images	75
4.2.1 Généralités et concepts de base	75
4.2.2 Caractéristiques d'un recalage	77
4.3 Problème de mise en correspondance	82
4.4 L'approche proposée	83
4.4.1 Formulation bi-objective du problème de recalage d'images.	83
4.4.2 Une représentation quantique du problème	85
4.4.3 Fonctionnement de L'algorithme	86
4.4.4 Résultats expérimentaux	90
4.5 Conclusion	93

4.1 Introduction

Le recalage d'images est un problème fondamental en vision par ordinateur et plus précisément en traitement et analyse d'images. Il est requis dans plusieurs domaines tel que le traitement et l'analyse d'images médicales, l'inspection d'images aériennes et satellitaires, la robotique, la biologie moléculaire, la conception assistée par ordinateur. Malgré l'abondance des travaux de recherche, le problème de recalage d'image suscite toujours un intérêt croissant. On précise que principalement toutes les définitions et les images illustratives ont été prise de [Meshoul, 04].

4.2 Le recalage d'images

4.2.1 Généralités et concepts de base

Intuitivement, le recalage d'images est le processus qui permet de superposer la plus grande partie commune entre deux images. C'est-à-dire, il consiste à trouver la transformation optimale qui appliquée à une des images que nous appelleront, image source, permet de la rendre en correspondance spatiale avec l'autre image référence (figure 4.1).

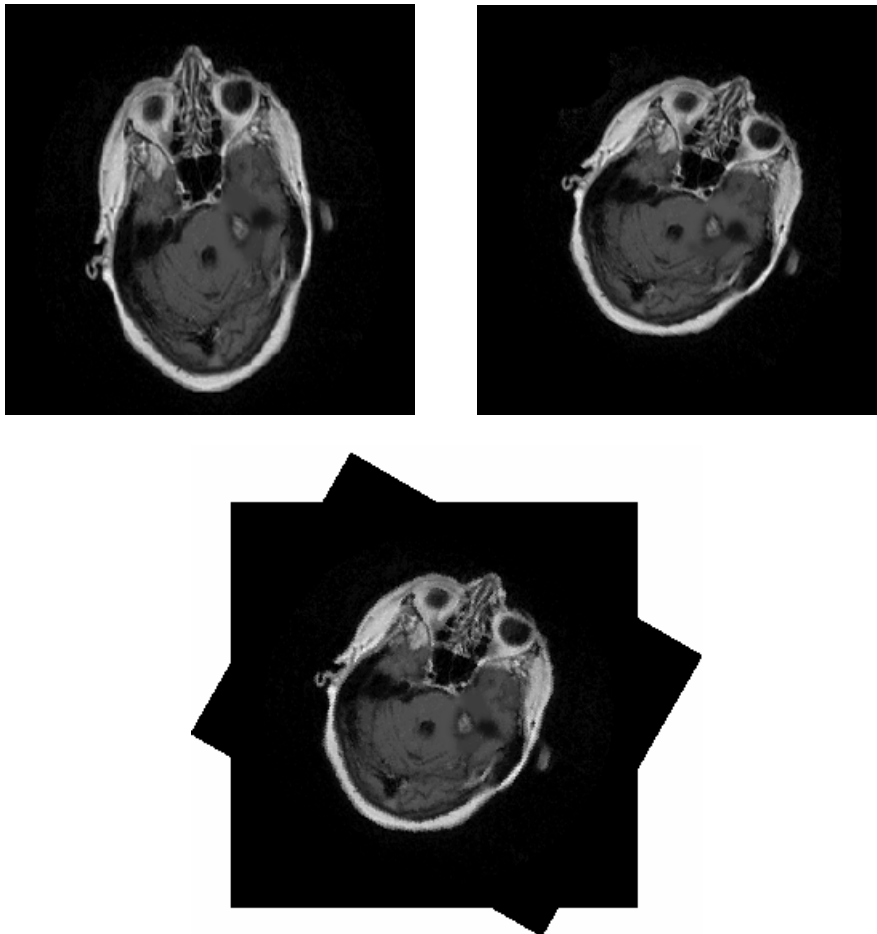


Figure 4.1. Superposition d'une image référence et d'une image source.

D'une manière formelle [Meshoul, 04], soient deux images à recaler I_R et I_S qui dénotent respectivement l'image référence et l'image source. I_R et I_S sont considérées comme des applications d'une partie finie de \mathbb{R}^2 ou \mathbb{R}^3 à valeurs dans \mathbb{R} :

$$I_R : \Omega_{I_R} \subset \mathbb{R}^d \rightarrow \mathbb{R} \quad \text{et} \quad I_S : \Omega_{I_S} \subset \mathbb{R}^{d'} \rightarrow \mathbb{R} \quad \text{où} \quad (d, d') \in \{2, 3\}^2$$

Les éléments de Ω_{I_R} et Ω_{I_S} sont les vecteurs de dimension 2 ou 3 des coordonnées spatiales associés aux points images (pixels ou voxels). Recaler I_R et I_S consiste à trouver la transformation géométrique G et la transformation radiométrique F qui permettent d'apparier des points physiquement homologues selon la relation :

$$I_R(p) = F(I_S(G(p)), q) + B \quad (4.1)$$

où

p : est le vecteur des coordonnées spatiales d'un point d'une image,

q : est un vecteur qui représente tous les facteurs exogènes qui influent sur les valeurs des pixels tel que la lumière par exemple.

B : est une mesure d'une variable aléatoire modélisant le bruit du processus d'imagerie.

De ces définitions, il ressort que le désalignement des images est causé par les distorsions radiométriques et les distorsions géométriques.

- § les distorsions radiométriques sont corrigées par la transformation F . Parmi les sources de ces distorsions on peut citer la lumière et les conditions atmosphériques.
- § les distorsions géométriques sont corrigées par la transformation G . Parmi les sources de ces distorsions on peut citer l'imperfection du procédé d'imagerie.

Dans la suite, on s'intéressera aux distorsions géométriques. Le lecteur peut consulter [Brown, 92] pour une étude détaillée des distorsions radiométriques.

4.2.2 Caractéristiques d'un recalage

Dans une revue de référence, Brown a identifié quatre critères pour caractériser un recalage [Brown, 92]. Il s'agit de la nature de la transformation, l'espace des primitives, les mesures de similarité et la stratégie de recherche.

4.2.2.1 Nature de la transformation

Le recalage d'images nécessite la détermination de la transformation géométrique G qui relie tout point p_1 de l'image source à un point p_2 de l'image référence. Le choix du type de transformation dépend des distorsions à corriger par le recalage. Il existe deux types de transformations :

a. les transformations paramétriques

Cette classe englobe toutes les transformations linéaires et non linéaires. Nous considérons que tout point p_1 de l'image source a pour coordonnées homogènes $(x,y,1)$ et tout point p_2 de l'image référence a pour coordonnées homogènes $(u,v,1)$.

a.1 Translation

C'est une transformation à deux degrés de liberté à savoir t_x , déplacement en x et t_y , déplacement en y .

$$p_2 = G(p_1) = p_1 + T, \quad (4.2)$$

Ce qui s'écrit sous forme matricielle :

$$\begin{bmatrix} u \\ v \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} x \\ y \\ \mathbf{1} \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ \mathbf{1} \end{bmatrix}$$

a.2 Translation et rotation

La combinaison d'une translation T de paramètres (t_x, t_y) et d'une rotation R d'angle q , peut s'exprimer sous la forme suivante :

$$p_2 = G(p_1) = R(p_1) + T,$$

$$\begin{bmatrix} u \\ v \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \cos q & -\sin q & t_x \\ \sin q & \cos q & t_y \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} x \\ y \\ \mathbf{1} \end{bmatrix} \quad (4.3)$$

a.3 Translation et homothétie

C'est une combinaison d'une translation T de paramètres (t_x, t_y) et d'une homothétie H de facteur le scalaire s :

$$p_2 = G(p_1) = s.p_1 + T, \quad (4.4)$$

Ce qui s'écrit sous forme matricielle :

$$\begin{bmatrix} u \\ v \\ \mathbf{1} \end{bmatrix} = s. \begin{bmatrix} x \\ y \\ \mathbf{1} \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ \mathbf{1} \end{bmatrix}$$

a.4 Similitude

Une similitude englobe une rotation R d'angle q , une translation T de paramètres (t_x, t_y) et une homothétie H de facteur le scalaire s :

$$p_2 = G(p_1) = s.R(p_1) + T, \quad (4.5)$$

$$\begin{bmatrix} u \\ v \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} s.\cos q & -s.\sin q & t_x \\ s.\sin q & s.\cos q & t_y \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} x \\ y \\ \mathbf{1} \end{bmatrix}$$

a.5 Etirement

C'est une généralisation directe de la similitude dans le sens où c'est une combinaison d'un changement d'échelle sur chaque axe représenté par les paramètres S_x et S_y avec un déplacement rigide (une rotation et une translation) :

$$p_2 = G(p_1) = R.D.X + T, \quad (4.6)$$

Où D est une matrice diagonale $D = \begin{bmatrix} S_x & \mathbf{0} \\ \mathbf{0} & S_y \end{bmatrix}$,

Sous forme matricielle, on écrit :

$$\begin{bmatrix} u \\ v \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} S_x \cdot \cos q & -S_y \cdot \sin q & t_x \\ S_x \cdot \sin q & S_y \cdot \cos q & t_y \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} x \\ y \\ \mathbf{1} \end{bmatrix}$$

a.6 Transformation affine

Elle a la forme générale suivante :

$$p_2 = G(p_1) = L(p_1) + T, \quad (4.7)$$

Où L est une transformation linéaire,

Sous forme matricielle, on écrit :

$$\begin{bmatrix} u \\ v \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} x \\ y \\ \mathbf{1} \end{bmatrix}$$

a.7 Transformation projective ou homographie plane

Elle permet de décrire un mouvement 3D d'une caméra autour d'un objet 3D où tous les points sont coplanaires. La transformation projective ajoute deux degrés de liberté à la transformation affine pour traiter les effets de projection. On obtient alors une transformation non linéaire qui a pour forme (en coordonnées homogènes) :

$$\begin{bmatrix} u \\ v \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ \mathbf{1} \end{bmatrix} \quad (4.8)$$

a.8 Transformation bilinéaire

C'est une généralisation directe de la transformation affine mais elle présente des propriétés différentes.

$$\begin{aligned} u &= a_1x + a_2y + a_3xy + a_0 \\ v &= b_1x + b_2y + b_3xy + b_0 \end{aligned} \quad (4.9)$$

$$\begin{bmatrix} u \\ v \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 & a_0 \\ b_1 & b_2 & b_3 & b_4 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} x \\ y \\ xy \\ \mathbf{1} \end{bmatrix}$$

a.9 Transformation polynomiale d'ordre supérieure

la transformation polynomiale d'ordre p est spécifiée sous forme :

$$u = \sum_{i=0}^p \sum_{j=0}^p a_{ij} x^i y^j \quad \text{et} \quad v = \sum_{i=0}^p \sum_{j=0}^p b_{ij} x^i y^j \quad (4.10)$$

b. les transformations non paramétriques

Les transformations paramétriques décrites précédemment ne sont pas bien adaptées pour traiter des distorsions locales. Récemment une seconde alternative consiste à exploiter les modèles élastiques. Plusieurs transformations non paramétriques ont été alors définies [Zitova, 03].

4.2.2.2 Espace des primitives

Une primitive est une structure d'intérêt extraite d'une image et qui véhicule une information de type photométrique, géométrique, statistique ou toute autre sémantique de haut niveau. Selon qu'elles nécessitent des procédés d'extraction ou non, les primitives peuvent être iconiques ou géométriques.

La primitive iconique est de l'ordre de la sensation. Elle consiste en l'élément le plus élémentaire dans une image à savoir le pixel ou le voxel caractérisé par son intensité et sa position. Son identification ne nécessite aucune tâche d'extraction.

Les primitives géométriques sont de l'ordre de la perception voire même cognition. Elles peuvent être des primitives points d'intérêt, des lignes, des surfaces et des régions. Elles nécessitent un procédé d'extraction.

4.2.2.3 Mesures de similarité

Plusieurs mesures de similarités ont été proposées dans la littérature [Zitova, 03]. Une mesure de similarité permet d'évaluer quantitativement le degré de ressemblance entre les deux images à recalcr. Formellement, c'est une fonction notée C à valeurs réelles dont les arguments sont les images à comparer et la transformation considérée.

$$C : (I_S, I_R, G) \rightarrow R \quad (4.11)$$

Les valeurs retournées par la fonction de similarité donnent une indication sur la ressemblance entre I_R et la transformée par G de I_S .

Plusieurs mesures de similarités ont été détaillées dans [Meshoul, 04].

4.2.2.4 Stratégie de recherche

Beaucoup de méthodes ont été proposées pour résoudre le problème de recalage d'images [Zitova, 03][Brown, 92]. 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 intensité et les approches basées 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. Parmi les approches iconiques existantes, on trouve :

- § Les techniques de descente du gradient [Sharma, 97],
- § Le recuit simulé [Nikou, 99],
- § L'approche multi-résolution [Jenkinson, 01].

§ **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.

Parmi les approches basées primitives existantes, on trouve :

- § La relaxation [Kittler, 00],
- § La programmation dynamique [Ohta, 85],
- § Alignement ou prédiction [Alter, 93],
- § Transformé de Hough généralisée [Samal, 97],
- § Hachage géométrique [Lamiroy, 96],
- § L'arbre d'interprétation [Grimson, 87],
- § Le plus proche voisin itéré [Besl, 92],
- § Isomorphisme de graphes [Depiario, 96],
- § Les méthodes connexionnistes pour la mise en correspondance [Hopfield, 85],
- § Les méthodes floues [Xue, 01].

4.3 Problème de mise en correspondance

Les points extraits des images à recaler peuvent être :

- § Points pertinents : Un point d'une image est dit pertinent s'il a un correspondant extrait dans l'autre image et qu'il contient une information utile pour l'estimation.
- § Points bruités : Un point est dit bruité s'il contient peu d'informations pertinentes.
- § Points aberrants : Un point d'une image est dit aberrant s'il n'a pas de correspondant dans l'autre image.

4.4 L'approche proposée

Cette approche a été proposée et acceptée dans TISR'2005 [Mahdi, 05]. Pour la présenter, nous commençons d'abord par proposer une formulation bi-objective du problème de recalage d'images. Par la suite on lui associe une représentation quantique adéquate au biais de laquelle on expliquera le fonctionnement général de l'approche proposée et on finira par présenter une étude expérimentale.

4.4.1 Formulation bi-objective du problème de recalage d'images.

Dans le cadre de ce travail, on s'intéresse à l'alignement des images mono modalités issues des mêmes systèmes d'imagerie. Pour résoudre ce problème, la principale difficulté réside dans la mise en correspondance de deux ensembles de points, extraits des images source et référence, avec la présence de bruit et de points aberrants (outliers).

Soient P_1 et P_2 deux ensembles de primitives points extraits de deux images I_R et I_S par le biais d'un détecteur tel que Harris [Trajkovic, 98](voir figure 4.2). On peut considérer P_1 et P_2 comme des abstractions des images I_R et I_S . On pose :

$$P_1 = \{X_i\}_{i=1}^{n_1} \text{ et } P_2 = \{X'_j\}_{j=1}^{n_2} \quad (4.12)$$



Figure 4.2 Extraction de primitives utilisant le détecteur de Harris.(a) image avant extraction, (b) image après extraction de points.

Soit G la relation géométrique qui lie les deux ensembles de points. Dans le cadre de ce travail on considère le cas d'une transformation affine.

Le problème de mise en correspondance pour le recalage basé primitive peut être décrit par le quadruplet $(S, Cost, Corr, C)$ où :

§ S dénote l'ensemble des solutions potentielles. Une solution potentielle consiste en un état de correspondance à partir duquel une transformation peut être estimée :

$$S = \{s_1, s_2, \dots, s_n\}$$

§ $Cost$: une application de $S \rightarrow R$ qui désigne la fonction coût. Elle définit la qualité de chaque solution.

$$Cost : S \rightarrow R \text{ qui associe à chaque } s_k \in S \text{ un coût } f(s_k) \in R.$$

§ $Corr$: une application de $S \rightarrow R$ qui désigne la fonction corrélation. Elle définit le degré de similarité entre l'image source et l'image référence.

$$Corr : S \rightarrow R \text{ qui associe à chaque } s_k \in S \text{ un degré de similarité } f(s_k) \in R.$$

§ C : l'ensemble des contraintes qui dérivent la faisabilité d'une solution. Nous considérons notamment les contraintes d'unicité.

Donc le problème de recalage d'image peut être défini comme un problème d'optimisation bi-objectif de la façon suivante :

$$\left\{ \begin{array}{ll} Max & Corr(s) \\ Min & Cost(s) \\ \text{Vérifiant} & C \\ \text{Où} & s \in S \end{array} \right. \quad (4.13)$$

Une solution potentielle, dans notre cas, est une matrice de correspondances contenant des points pertinents et des points aberrants. Elle est définie par :

$$s = \{(X_i, X'_j) / X_i \in P_1 \cup \{null\}, X'_j \in P_2 \cup \{null\}\} \quad (4.14)$$

Où le terme null est utilisé pour désigner les outliers. Si on suppose que P_1 et P_2 sont liés par une transformation géométrique affine G , la fonction coût est définie par :

$$Cost(s) = \sum e_{ij}^2 \quad (4.15)$$

Où $e_{ij} = \|X'_j - G(X_i)\|$ est le terme résiduel correspondant de la mise en correspondance de la paire de points (X_i, X'_j) appartenant à la solution courante s . le facteur α est utilisé pour renforcer les correspondances correctes et rejeter les outliers. La fonction de corrélation est définie comme suit :

$$Corr(s) = average(cor(X_i, X'_j)), \forall (X_i, X'_j) \in s \quad (4.16)$$

Où $cor(X_i, X'_j)$ est la corrélation entre la paire de points (X_i, X'_j) .

Enfin, la contrainte C exprime le fait que chaque point x_i de P_1 doit avoir un et un seul correspondant x'_j appartenant à $P_2 \cup \{null\}$.

4.4.2 Une représentation quantique du problème

Une solution potentielle du problème de recalage d'image est donnée par la matrice de correspondance $M = (m_{ij})_{i=1..n_1, j=1..n_2}$, où n_1 et n_2 représentent respectivement les dimensions des deux ensembles de points P_1 et P_2 . La représentation des points aberrants se fait facilement en rajoutant à la matrice M une extra-ligne et une extra-colonne. Chaque élément m_{ij} de M prend la valeur 1 si le point x_i de P_1 correspond au point x'_j de P_2 , et prend la valeur 0 sinon. Trouver une représentation quantique pour cette matrice solution nécessite la définition des probabilités pour qu'une paire de points soit en correspondance.

Quant un point de P_1 ou P_2 est supposé aberrant, l'élément correspondant dans la ligne externe ou la colonne externe est mis à 1.

Donc, chaque élément m_{ij} peut être naturellement représenté par une paire de nombres (a, b) définissant le qubit:

$$|y\rangle = \begin{pmatrix} a \\ b \end{pmatrix} \text{ où } |a|^2 + |b|^2 = 1 \quad (4.17)$$

$|a|^2$ donne la probabilité pour que le qbit prenne la valeur 0, et donc la variable de décision prend aussi la valeur 0. Tandis que la valeur $|b|^2$ donne la probabilité pour que le qbit prenne la valeur 1, et donc la variable de décision prend aussi la valeur 1. Donc la matrice de correspondance M peut être représentée comme une matrice de registres quantiques q contenant $2^{*(n_1+1)} * (n_2+1)$ qbits de la façon suivante :

$$q = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1(n_2+1)} \\ b_{11} & b_{12} & b_{13} & \dots & b_{1(n_2+1)} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2(n_2+1)} \\ b_{21} & b_{22} & b_{23} & \dots & b_{2(n_2+1)} \\ \dots & \dots & \dots & \dots & \dots \\ a_{[(n_1+1)]1} & a_{[(n_1+1)]2} & a_{[(n_1+1)]3} & \dots & a_{[(n_1+1)](n_2+1)} \\ b_{[(n_1+1)]1} & b_{[(n_1+1)]2} & b_{[(n_1+1)]3} & \dots & b_{[(n_1+1)](n_2+1)} \end{bmatrix} \quad (4.18)$$

Cette représentation est avantageuse car elle permet de représenter une superposition linéaire de tous les états possibles. Toutes les combinaisons possibles des valeurs des variables de décision peuvent être dérivés à partir de cette simple représentation. Pour plus de diversité, on maintient une population de solutions quantiques. Donc à chaque génération de notre approche évolutionnaire, on manipule une matrice d'individus quantiques notée $Q(t)$:

$$Q(t) = \{q_1^t, q_2^t, \dots, q_n^t\} \quad (4.19)$$

Où n représente la taille de la population, et q_j^t , $j=1..n$ est l'un individu quantique définit dans (1)

Avec cette nouvelle représentation, on peut maintenant formuler les contraintes d'unicité comme suit :

$$\sum_{k=1}^{n_2+1} m_{ik} = 1, \quad \forall i \in [1..n_1] \quad \text{et} \quad \sum_{k=1}^{n_1+1} m_{kj} = 1, \quad \forall j \in [1..n_2] \quad (4.20)$$

Enfinement le problème de recalage d'images bi_objectif peut être formulé comme suit :

$$\begin{array}{l} \text{Min } Cost(s) = \sum \|X'_j - G(X_i)\|^2 \\ \text{Max } Corr(s) = average(cor(X_i, X'_j)), \quad \forall (X_i, X'_j) \in s \\ \text{vérifiant } \sum_{k=1}^{n_2+1} m_{ik} = 1, \quad \forall i \in [1..n_1] \\ \text{et } \sum_{k=1}^{n_1+1} m_{kj} = 1, \quad \forall j \in [1..n_2] \\ \text{avec } X_i \in P_1, X'_j \in P_2 \end{array} \quad (4.21)$$

4.4.3 Fonctionnement de l'algorithme

Maintenant, on va décrire comment la représentation quantique, décrite précédemment, peut être manipulée et exploitée par notre algorithme évolutionnaire

quantique afin de trouver une approximation du Front de Pareto pour le problème de recalage d'image bi-objectif.

En plus de la population quantique $Q(t)$, on maintient, dans notre algorithme, une population temporaire dénotée par $Temp(t)$ et un archive externe $A(t)$, qui permet de stocker l'ensemble des solutions non dominées trouvées au cours du processus de recherche. On a utilisé la relation de dominance de Pareto pour assigner des valeurs d'adaptation aux différents individus de la population. Pour plus de clarté, on commence par donner le schéma général de notre algorithme, puis on détaillera les différentes étapes par la suite.

INPUT: Problem specification

Begin

$t \leftarrow 0.$

i. Create empty external non dominated set $A(t)$.

ii. Initialize $Q(t)$.

iii. Extract sets of points P_1 and P_2 from I_R and I_S .

iv. Compute correlation matrix.

While ($t \leq \text{max_gen}$)

v. Make $P(t)$ by observing the states of $Q(t)$.

vi. Repair $P(t)$ for restoring the feasibility of the solutions in $P(t)$.

vii. Copy non dominated solutions of $P(t)$ and $A(t)$ in $A(t+1)$.

viii. Compute the fitness of each individual in $P(t)$ and $A(t+1)$.

ix. Select individuals from $P(t)$ and $A(t+1)$ and store them in $Temp(t)$.

x. Interfere $Q(t)$ toward $Temp(t)$.

xi. Apply quantum crossover and quantum mutation.

$t \leftarrow t + 1$

End

End

OUTPUT: the set $A(t)$ of non dominated solutions.

La spécification du problème inclut la définition des fonctions objectives, les contraintes du problème, l'ensemble des paramètres du problème et les deux images à recalculer I_R et I_S .

Dans l'étape ii. $Q(0)$ est généré. Tous les a_{ij}^0 et b_{ij}^0 pour $i=1,2,\dots,(n_1+1)$, $j=1,2,\dots,(n_2+1)$ de tous les $q_k^0 = q_k^t|_{t=0}$, $k=1,2,\dots,n$ sont initialisés à $1/\sqrt{2}$. Ceci implique qu'un

individu q_k^0 représente une superposition linéaire de tous les états possibles avec la même probabilité :

$$|\Psi_{q_k^0}\rangle = \sum_{l=1}^{2^{(n_1+1)(n_2+1)}} |X_l\rangle \quad (4.22)$$

Où X_l est le $l^{\text{ième}}$ état représenté par la matrice binaire

$$\begin{bmatrix} b_{11} & b_{12} & b_{13} & \dots & b_{1(n_2+1)} \\ b_{21} & b_{22} & b_{23} & \dots & b_{2(n_2+1)} \\ \dots & & & & \\ b_{(n_1+1)1} & b_{(n_1+1)2} & b_{(n_1+1)3} & \dots & b_{(n_1+1)(n_2+1)} \end{bmatrix} \quad (4.23)$$

Pour laquelle chaque b_{ij} est soit '0', soit '1' selon la probabilité de $|a_{ij}^0|^2$ ou $|b_{ij}^0|^2$ respectivement.

Au cours de l'étape iii. On fait appel au détecteur Harris pour extraire les ensembles de points P_1 et P_2 à partir des images référence I_R et source I_S . Puis on calcule la matrice de corrélations entre ces deux ensembles de points. A l'étape iv, un opérateur quantique est appliqué. Il consiste à dériver une image binaire de la population quantique $Q(t)$. Cette représentation binaire est dénotée par $P(t)$ et elle est défini par :

$$P(t) = \{b_1^t, b_2^t, \dots, b_n^t\} \quad (4.24)$$

Une solution binaire, b_k^t for $k=1,2,\dots,n$ est une matrice binaire de longueur $(n_1+1)(n_2+1)$, qui est construite par la sélection de '0' ou '1' pour tout bit en utilisant la probabilité $|a_{ij}^0|^2$ et $|b_{ij}^0|^2$. La figure 4.3 présente un exemple de chromosome appartenant à $P(t)$.

	X'_1	X'_2	X'_3	X'_4	X'_5	...	X'_{n_2}	Slack
X_1	1	0	0	0	0		0	0
X_2	0	0	0	0	0		0	1
X_3	0	0	0	1	0		0	0
X_4	0	0	0	0	0		1	0
...								
X_{n_1}	0	0	1	0	0		0	0
Slack	0	1	0	0	1		0	

Figure 4.3. Exemple de chromosome de $P(t)$

Dans l'étape suivante, la faisabilité des solutions obtenues est vérifiée. Si un individu binaire de $P(t)$ viole la contrainte d'unicité, l'algorithme tente de réparer et restaurer la faisabilité des solutions dans $P(t)$. Dans l'étape vii. la population externe $A(t)$,

initialisé à 0 au début de l'algorithme, est régulièrement mise à jour en utilisant la relation de dominance Pareto. Pendant chaque itération, les nouvelles solutions non dominées rencontrées sont rajoutée à l'archive et les anciennes solutions de l'archive dominées par ces nouveau individus sont supprimées. Dans l'étape viii, on calcule les valeurs d'adaptation des individus dans $A(t+1)$ et $P(t)$ selon le principe de la méthode QMOKP détaillée dans le chapitre précédent. Pour des raisons de clarté, nous présentons les grandes lignes de calcul des valeurs d'adaptations des différents chromosomes. Pour chaque individu i de $P(t)$ et $A(t+1)$, la valeur d'adaptation est calculé de la façon suivante :

1. Soit S_i la force de l'individu i . S_i est le nombre des individus de $P(t)$ et $A(t+1)$ dominés par l'individu i .

$$\forall i \in P(t) \vee A(t+1) \quad S_i = \{j \in P(t) \vee A(t+1) / i \mathbf{f} j\} \quad (4.25)$$

Où le symbole $i \mathbf{f} j$ se lit i domine j .

2. calculer R_i pour l'individu i . c'est la somme des forces des individus dominant i :

$$R_i = \sum_{\substack{j \in P(t) \vee A(t+1) \\ j \mathbf{f} i}} S_j \quad (4.26)$$

3. calculer la densité D_i définit par :

$$D_i = \frac{1}{S_i^k + 2} \quad \text{Where} \quad k = \sqrt{|P(t)| + |A(t+1)|} \quad (4.27)$$

où S_i^k est la distance entre l'individu i et sont $k^{\text{ième}}$ voisin, qui est sélectionner à partir d'une liste de distance, triée dans l'espace des objectives, entre l'individu i et les autres individus.

4. finalement, la valeur d'adaptation d'un individu est donnée par :

$$F_i = R_i + D_i \quad (4.28)$$

Dans l'étape ix, La population temporaire est construite à partir des meilleurs individus de $P(t)$ et $A(t+1)$. Les n meilleurs individus de $A(t+1)$ sont sélectionnés et

stockés dans $\text{Temp}(t)$. si la taille de $A(t+1)$ est inférieure à n , les individus manquants dans $\text{Temp}(t)$ sont complétés à partir de $P(t)$. $\text{Temp}(t)$ et $P(t)$ ont la même longueur n . Cette population temporaire est utilisée pour générer une nouvelle population quantique $Q(t)$. Et ceci en appliquant une opération quantique spéciale appelée interférence. C'est une porte de rotation spéciale, qui vise à intensifier la recherche dans le voisinage des solutions stockées dans $\text{Temp}(t)$.

Plusieurs techniques d'interférences ont été testées. Ces techniques ont été détaillées dans le chapitre précédent.

Finalement, les individus quantiques dans $Q(t)$ sont sélectionnés pour des opérations de croisements et mutations quantiques. On doit noter, à ce niveau, qu'il s'agit de croisement de matrices de correspondances.

Toutes les étapes décrites précédemment sont appliquées itérativement, et le processus évolue jusqu'à atteindre un nombre maximum d'itérations. La population externe est présentée comme étant la solution du problème considéré. En effet chaque solution de l'archive externe représente une matrice de correspondance à laquelle est associée une transformation, une valeur de corrélation et un coût.

4.4.4 Résultats expérimentaux

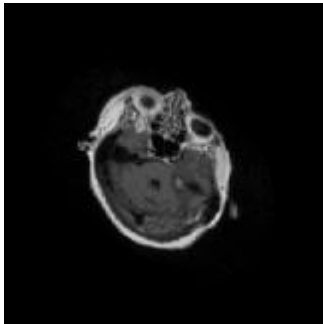
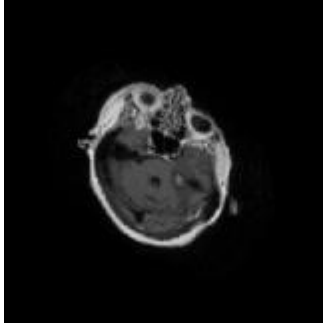
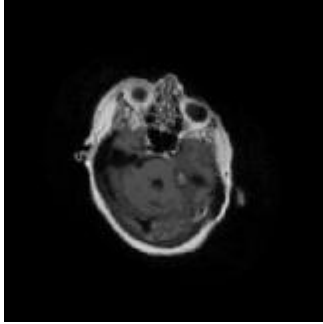
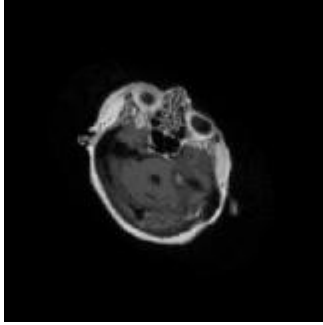
Dans cette section, nous présentons les résultats expérimentaux obtenus par l'approche proposée. Cette approche a été proposée et acceptée dans TISR'2005 [Mahdi, 05].

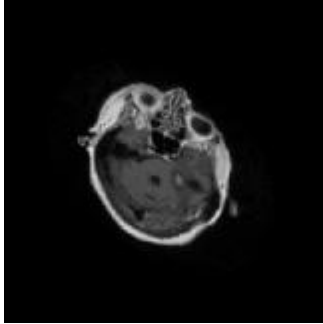
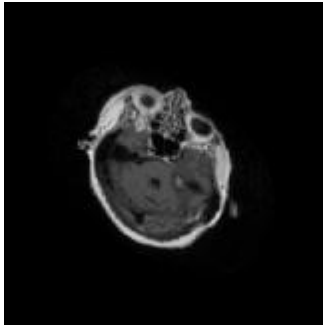
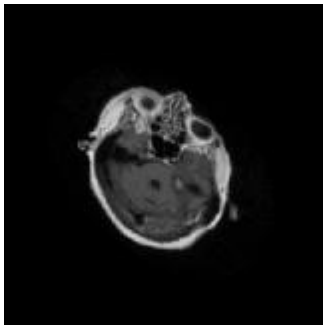
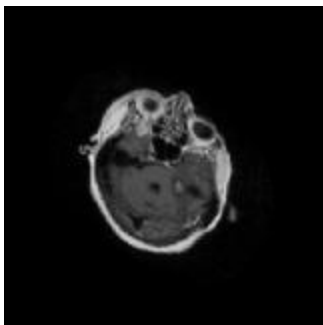
Dans cette exécution, on a utilisé 1000 générations et 4 individus. La figure 4.4 donne les deux images référence et source qu'on a utilisé lors des expérimentations.



Figure 4.4. exemple de paires d'images utilisées dans les expériences. (a) image de référence et (b) image source à transformer.

Le tableau suivant présente l'ensemble des solutions non dominées trouvées par la méthode proposée. Pour chaque solution, on a donné l'image correspondante, ses valeurs de coût et de corrélations.

N°	Image	Coût e+003	Corrélation
1		4.1633	0.6388
2		4.1260	0.6195
3		4.3561	0.7204
4		4.0976	0.5617

5		4.0094	0.5321
6		4.2632	0.6493
7		4.1165	0.6002
8		4.0279	0.5372

La figure 4.5 donne une représentation graphique de l'approximation du front de Pareto trouvée lors du processus de recherche.

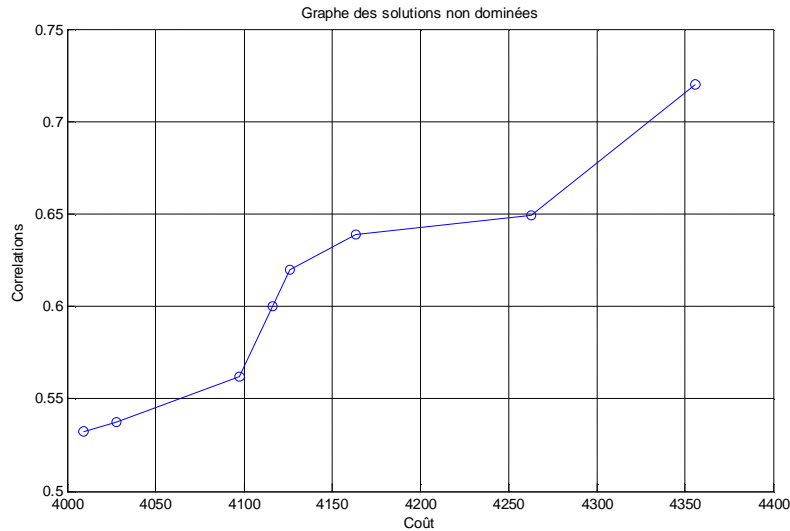


Figure 4.5. L'ensemble des solutions non dominées

4.5 Conclusion

On a présenté dans ce chapitre une approche bi-objective pour le recalage d'images. L'approche proposée consiste fondamentalement à adopter une représentation probabiliste de la population entière. Des opérateurs spéciaux sont utilisés pour manipuler cette population quantique. On peut citer à titre d'exemple la mesure quantique et l'interférence. Comparer à un algorithme évolutionnaire, l'approche basée quantique fournit l'avantage de donner une plus grande diversité. Toutes les solutions existent dans seulement un chromosome. Cette résolution bi-objective du problème de recalage d'images a surpassé toutes les difficultés liées au réglage de certains paramètres utilisés dans la fonction à optimiser. En effet, le problème de recalage d'images n'est pas de nature mono-objectif. Pour sa résolution mono-objectif, il a fallu introduire un nouveau paramètre de calcul dont le réglage est très difficile. Par conséquent la nouveauté de cette approche est premièrement les principes quantiques qu'elle utilise, et deuxièmement sa formulation du problème. L'idée est de formuler le problème de l'alignement d'images comme un problème d'optimisation bi-objectif. Nous avons choisi d'optimiser deux fonctions objectives, la fonction Coût et les fonctions corrélation. Dans cette résolution multiobjective, nous avons à la fin du processus de l'optimisation un ensemble de solutions de compromis dans une seule exécution. Cette multiplicité de solutions est très utile dans un système d'imagerie. Les résultats obtenus sont très encourageant et montrent la faisabilité et aussi l'efficacité de l'approche proposée. Afin d'améliorer les performances de cette approche, on peut prévoir une étape d'initialisation intelligente, exploiter le parallélisme intrinsèque à la méthode et probablement considérer d'autres fonctions d'adaptation.

Références bibliographiques

[Alter, 93]

T. Alter and W. Grimson, *Fast and robust 3D recognition by alignment*, in proceedings of International Conference on Computer Vision, pp. 113-120, 1993.

[Ausiello, 99]

G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti Spaccamela, M. Protrasi, *Complexity and Approximation Combinatorial Optimization Problems and their Approximability Properties*, Springer Verlag, 1999.

[Barichard, 03a]

V. Barichard and J. Hao, *Genetic Tabu Search for Multi-objective Knapsack Problem*, Angres 2003.

[Barichard, 03b]

V. Barichard, *Approches Hybrides pour les problèmes d'optimisation multiobjectifs*, Thèse de doctorat, Université d'Angers, 2003.

[Bellac, 03]

Michel le Bellac, *Introduction à l'informatique quantique*, cours donné à l'Ecole Supérieure de Sciences Informatique (ESSI), Institut non linéaire de Nice, 2003.

[Benabdel, 97]

F. Ben Abdelaziz And S. Krichen, *A tabu search heuristic for multiple objective knapsack problems*, Ructor Reasearch Report, RR 28-67, 1997.

[Besl, 92]

P. Besl and N. Mckay, *A method for registration of 3D shapes*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(2): 238-256, 1992.

[Blais, 02]

Alexandre Blais, *Algorithmes et architectures pour ordinateurs quantiques supraconducteurs*, Sherbrooke, Québec, Canada, thèse de doctorat en physique, 2002.

[Brown, 92]

L. Brown, *A survey of image registration techniques*, ACM surveys, 24(4): 325-376, 1992.

[Carraway, 90]

R. Carraway, T. Morin and H. Moskowitz. *Generalized dynamic programming for multicriteria optimisation*, European Journal of Operational Research, p95-104, 1990.

[Coello, 98]

C. Coello Coello, *An updated survey on G.A. based multiobjective optimization techniques*, Rapport technique Lania-RD-98-08, Xalapa Veracruz, Mexico, décembre 1998.<http://www.lania.mx/~ccoello/EMOO>

[Coello, 99]

C. Coello Coello, *A survey of constraint handling techniques used with evolutionary computation methods*, Rapport technique Lania-RI-99-04, Laboratoire national de l'informatique Avanzada, Xalapa, Veracruz, Mexico, 1999.
<http://www.lania.mx/~ccoello/EMOO>

[Coello, 01]

C. Coello, A. Carlos and G. Pulido, *Multiobjective Optimization using a Micro-Genetic Algorithm*, In Lee Spector, Erik D. Goodman, Annie Wu, W. B. Langdon, Hans-Michael Voigt, Mitsuo Gen, Sandip Sen, Marco Dorigo, Shahram Pezeshk, Max H. Garzon and Edmund Burke, editors, Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001), pages 274-282, San Francisco, California. Morgan Kaufmann Publishers, 2001.

[Coello, 02]

C. Coello, A. Carlos and N. Cortés, *An approach to solve multiobjective Optimization Problems Based on an artificial Immune System*, In Jonathan Timmis and Peter J. Bentley, editors, First International Conference on Artificial Immune Systems (ICARIS'2002), pages 212-221. University of Kent at Canterbury, UK. ISBN 1-902671-32-5, September 2002.

[Coello, 03]

C. Coello and R. Becerra, *Evolutionary Multiobjective Optimization using a cultural Algorithm*, In IEEE Swarm intelligence Symposium Proceedings, IEEE Service Center, pp. 6-13, Indianapolis, Indiana, USA, April 2003.

[Collette, 02]

Y. Collette et P. Siarry, Optimisation multiobjectif. Eyrolles, 2002.

[Corne, 00a]

D. Corne and J. Knowles. *M-paes: a memetic algorithm for multiobjective optimization*. In Proceeding of the 2000 Congress on Evolutionary Computation, pages 325-332, 2000.

[Corne, 00b]

D. Corne , J. Knowles and M. Oates, *The Pareto Envelope-bases Selection Algorithm for Multiobjective Optimization*, In Marc Schoenauer, Kalyanmoy Deb, Gunter Rudolph, Xin Yao, Evelyne Lutton, J. J. Merelo and Hans-Paul Schwefel (editors), Proceedings of the Parallel Problem Solving from Nature VI Conference, pages 839-848. Springer, 2000.

[Cui, 01]

Cui, Xunxue, M. Li and T. Fang, *Study of Population Diversity of Multiobjective Evolutionary Algorithm Based on Immune and Entropy Principles*. In Proceedings of the Congress on Evolutionary Computation 2001, volume 2, pages 1316-1321, Piscataway, New Jersey. IEEE Service Center, May 2001.

[Czyzak, 98]

P. Czyzak and A.Jaszkiewicz, *A Pareto simulated annealing – a metaheuristic for multiple objective combinatorial optimization*. Journal of Multi-Criteria Decision Analysis, 7(1):34-47, 1998.

[Deb, 99]

K. Deb, *An overview of multi-objective evolutionary algorithms*, Journée J.E.T., Copie de transparents, mai 1999.

[Deb, 00]

K. Deb, S. Agrawal, A. Pratapand and T. Meyarivan, *A Fast and Elitist multiobjective genetic algorithm: NSGA-II*, Technical Report 20001, Indian Institute of Technology, Kanpur Genetic Algorithms Laboratory (Kan GAL), 2000.

[Depiario, 96]

F. Depiario, M. Trivedi and S. Serbin, *Graph matching using a direct classification of nodes attendance*, Pattern Recognition Letters, vol. 13, pp.529-539, 1996.

[Deutsch, 85]

D. Deutsch, *Quantum theory, the Church-Turing principle and the universal quantum computer*, Proc. R. Soc. London Ser. A 400, 97-117, 1985.

[Deutsch, 92]

D. Deutsch, R. Jozsa, *Rapid solution of problems by quantum computation*. Proceedings of the Royal Society of London SER. A439, 553-558, 1992.

[Dirac, 58]

Dirac, *The principles of Quantum Mechanisms 4th ed*. Oxford University Press. 1958.

[Draa, 04]

A. Draa, *Une nouvelle approche pour le recalage d'images multimodales, basée sur l'informatique quantique et les algorithmes évolutionnaires*, Mémoire de Magistère en informatique, université Mentouri Constantine , 2004.

[Eckart, 01]

Eckart Zitzler, Marco Laumanns, and Lothar Thiele, *SPEA2: Improving the Performance of the Strength Pareto evolutionary Algorithm*, Technical Report 103, Computer Engineering and Communication Networks lab(Tik), Swiss Federal Institute of Technology(ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, May 2001.

[Eckart, 03]

Eckart Zitzler and Marco Laumanns, *Problems and Test Data for Multiobjective Optimizers*. <http://www.tik.ee.ethz.ch/~zitzler/testdata.html>. Last updated on October 6, 2003

[Ekert, 98]

A. Ekert et R. Jozsa. *Quantum algorithms: entanglement enhanced information processing*. Phil. Trans. Roy. Soc. Lond. A, 356 :1779, 1998

[Engrand , 98]

P. Engrand, X. Mouney, *Une méthode originale d'optimisation multiobjectif*, Note interne EDF-DER numéro HT-14/97/035/A, mars 1998.

[Feynman, 82]

R. P. Feynman, *Simulating Physics with Computers*, Int. J. Theor. Phys. 21, 467-488, 1982.

[Fitzgerad, 01]

R. Fitzgard. *What really gives a quantum computer its power? Physics Today*, page 20, January 2001.

[Fonseca, 93]

C.M. Fonseca, P. J. Fleming, *Genetic Algorithms for multiobjective optimization : Formulation, Discussion and Generalization*, Proceedings of the Fifth International Conference on Genetic Algorithms, pages 416-423, San Mateo, California, 1993.

[Gandibleux, 97]

X. Gandibleux, N. Mezdaoui, A. Freville, *A multiobjective tabu search procedure to solve combinatorial optimization problems*, in : R.Caballero, F. Ruiz, R. Steuer (eds), *Advances in Multiple Objective and Goal Programming*. Lecture Notes in Econom, 1997.

[Giraldi, 04]

G. Giraldi, R. Poertugal, R. Thess, *Genetic Algorithms and Quantum Computation, National Laboratory for Scientific Computing, Petropolis, RJ, Brazil, 2004.*

[Glover ,86]

F. Glover, *Future Paths for integer programming and links to artificial intelligence*, Comput. and Ops. Res., volume 13, numéro 5, pages 533-549, 1986.

[Glover , 97]

F. Glover, M. Laguna, *Tabu search*, Kluwer Academic Publishers, 1997.

[Glover ,90]

F. Glover , *Artificial intelligence, heuristic frameworks and tabu search*, Managerial and decision economics, volume 11, pages 365-375, 1990.

[Goldberg, 87]

D.E. Goldberg and J. Richardson, *Genetic algorithms with sharing for multimodal function optimization*. In Genetic Algorithms and their applications: Proceedings of the second International Conference on Genetic Algorithms, pages 41-49. Lawrence Erlbaum, 1987.

[Goldberg, 89]

D.E. Goldberg, *Genetic algorithms for search, optimization, and machine learning*. Reading, MA: Addison-Wesley, 1989.

[Goldberg, 94]

D. Goldberg, *Algorithmes génétiques*, édition Addison Wesley, juin 1994.

[Grimson, 87]

E. Grimson and T. Lozano-Perez, *Localizing overlapping parts by searching the interpretation tree*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 9(5): 468-482, 1987.

[Grover, 97]

L. Grover, *Quantum Mechanics Helps in Searching for Needle in a Haystack Phys. Rev. Lett.* 79, 325-328, 1997.

[Han, 00]

K. Han, J. Kim, *Genetic quantum Algorithm and its application to Combinatorial Optimization problem*, in Proc. 2000 Congr. Evolutionary Computation, vol. 2, La Jolla, CA, pp. 1354-1360, 2000.

[Han, 01]

K H. Han, K H Park, C. H. Lee and J. H. Kim, *Parallel quantum-inspired genetic algorithm for combinatorial optimization problem*, in Proc. 2001 Congr. Evolutionary Computation, vol. 2, Seoul, Korea, pp. 1422-1429, 2001.

[Han, 02]

K. Han, J. Kim, *Quantum-Inspired Evolutionary Algorithm for a Class of Combinatorial Optimization*, IEEE Transactions on Evolutionary Computation, vol. 6, pp. 580-593, December 2002.

[Han, 03]

K . H. Han and J. H. Kim, *On setting the parameters of quantum-inspired evolutionary algorithm for practical applications*, in Proc. Congr. Evolutionary Computation, Canberra, Australia, pp. 178-184, Dec. 2003.

[Han, 04]

K.H. Han and J.H Kim, *Quantum inspired evolutionary algorithms with a new termination criterion, H_c gate two phase schemes*, IEEE transaction on evolutionary computation, 2004.

[Hansen, 97]

M. P. Hansen. *Tabu search for multiobjective optimization*, In Proceedings of 13th International conference on MCDM, 1997.

[Hansen, 98]

M. P. Hansen, *Metaheuristics for multiple objective combinatorial optimization*, Ph.D. Thesis, IMM-PHS-1998-45, Technical University of Denmark, Lyngby, 1998.

[Hillier, 95]

F. S. Hillier, G. J. Lieberman, *Introduction to operations Research*, McGraw-Hill International Editions, 1995.

[Holevo, 79]

A. S. Holevo, *Capacity of quantum communication channel*. Problems of Inf. Transm., 5 : 247, 1997.

[Hopfield, 85]

J. Hopfield and D. Tank, *Neural computations of decisions in optimization problems*, Biol. Cybernetics vol. 52, pp.141-152, 1985.

[Iredi , 01]

S. Iredi, D. Merkle, M. Middendorf, *Bi-criterion optimization with multi colony algorithms*, in: Proceeding of the First International Conference on Evolutionary Multi-Criterion Optimization. Lecture Notes in Computer Science, Springer, Berlin, 2001.

[Jang, 03]

J. Jang, K. Han, J. Kim, *quantum-inspired evolutionary algorithm-based face verification*, In Lecture Notes in Computer Science. Berlin, Germany: Springer-Verlag, Proc. Genetic Evolutionary Computation Conf. (2724), pp.2147-2156, 2003.

[Jenkinson, 01]

M. Jenkinson and S. Smith, *A global optimization method for robust affine registration of brain images*, Medical Image Analysis, 5:143-156, 2001.

[Kim, 03]

K. Kim, J. Hwang, K. Han, J. Kim and K. Park, *A quantum-inspired evolutionary computing algorithm for disk allocation method*, IEICE Trans. Inform. Syst., vol. E86-D, pp. 645-649, 2003.

[Kim, 04]

J. H. Kim and K. Han , *Quantum-Inspired Evolutionary Algorithms with New Termination Criterion, He Gate, and Two-Phase Scheme*, IEEE Transactions on Evolutionary Computation, Vol. 8, N°2, April 2004.

[Kittler, 00]

J. Kittler, *Probabilistic relaxation and the Hough transform*, Pattern Recognition, 33(4):705-714, 2000.

[Know, 99]

J. Knowles and D. Corne, *The Pareto archived evolution strategy: a new baseline algorithm for multiobjective optimization*, in 1999 Congress on Evolutionary Computation, Piscataway, NJ, IEEE Service Center, 98-105, 1999.

[Kurpati, 00]

A. Kurpati and S. Azarm, *Immune Network Simulation with Multiobjective Genetic Algorithms for Multidisciplinary Design Optimization*, Engineering Optimization, 33:245-260, 2000.

[Lamiroy, 96]

B. Lamiroy and P. Gros, *Rapid object indexing and recognition using enhanced geometric hashing*, in Proceedings of European Conference on Computer Vision, pp.578-589, 1990.

[Lechuga, 02]

M. Lechuga and C. Coello, *MOPSO: A proposal for multiple objective particle swarm optimization*, IEEE Congress on Evolutionary Computation, 2002 Honolulu, Hawaii USA, 2002.

[Mahdi, 05]

K. Mahdi, S. Meshoul, M. Batouche, *Bi-objective Quantum optimization for image alignment*, in proceedings of text, image recognition, workshop, TISR'2005.

[Mariano, 99]

E. Mariano and E. Morales, *MOAQ and ant-Q algorithm for multiple objective optimization problems*, in W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, R. E. Smith (eds), Proceeding of the genetic and evolutionary Computation, 1999

[MCDM, 02]

International Society on Multiple Criteria Decision Making. MCDM Numerical Instances Library. <http://www.univ-valencienne.fr/ROAD/MCDM/>
Created: October 1998, Last update: 25 September 2002.

[Meshoul, 04]

S. Meshoul, *Optimisation par les systèmes complexes pour le recalage et la mise en correspondance en analyse d'images*, Thèse de doctorat, Constantine, 2004.

[Meshoul, 05a]

S. Meshoul, K. Mahdi, M. Batouche, *Evolutionary Multi-objective Optimization using a Quantum-based Approach*, in proceedings of International Conference on Computer Systems and Technology, ICSIT'2005, pp. 748-753, volume 2, Algiers, July 2005.

[Meshoul, 05b]

S. Meshoul, K. Mahdi, M. Batouche, *A Quantum Inspired Evolutionary Framework for Multi-Objective Optimization*, in proceedings of EPIA 2005, *Lecture notes in Artificial intelligence*, pp. 190-201, C. Bento, A. Cardoso and G. Dias (Eds.), Springer Verlag, 2005.

[Miettinen, 99]

K. M. Miettinen, *Nonlinear multiobjective optimization*, editions Kluwer academic publisher, 1999.

[Narayanan, 96]

Narayanan and M. Moore, *Quantum-inspired genetic algorithms*, in Proc. IEEE Int. Conf. Evolutionary Computation, Nagoya, Japan, pp. 61-66, 1996.

[Nielsen, 00]

M. A. Nielsen et I. L. Chuang, *quantum Computation and quantum Information*, Cambridge University Press, 2000.

[Nikou, 99]

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.

[Ohta, 85]

Y. Ohta and T. Kanade, *Stereo matching by intra and inter scanline search*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 7(2), pp.139-154, 1985.

[Pirlot, 03]

M. Pirlot and J. Teghem, *Résolution de problèmes de RO par des métaheuristiques (Traité IC2, série Informatique et systèmes d'information)*. Hermès Sciences, 2003.

[Ray, 02]

T. Ray and K. M. Liew, *A swarm methaphor for multi-objective design optimization*. Engineering Optimization. 34(2):141-153, March 2002.

[Samal, 97]

A. Samal and J. Edwards, *Generalized Hough transform for natural shapes*. Pattern Recognition Letters, 18(5): 473-480, 1997.

[Schott, 95]

J. R Schott. *Fault tolerant design using single and multicriteria genetic algorithm optimization*. Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 1995.

[Serafini, 92]

Serafini, *Simulated annealing for multiple objective optimization problems*, in: Proceeding of the Tenth International Conference on Multiple Criteria Decision Making, Taipei 19-24.07, vol. 1, 87-96, 1992.

[Sharma, 97]

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

[Shor, 94]

P. Shor, *Algorithms for Quantum Computation: Discrete Logarithms and Factoring*, Proceedings on the 35th Annual Symposium of Computer Science, IEEE Computer Society, Los Alamitos, CA, 1994.

[Shumacher, 95]

B. Schumacher, *Quantum coding*, Phys. Rev. A, 51 :2738, 1995.

[Spector, 99]

L. Spector, H. Bernstein and N. Swamy, *Finding a better-than-Classical Quantum AND/OR Algorithm using Genetic Programming*, in Proceedings of the 1999 Congress on Evolutionary Computation, IEEE Press, Piscataway, NJ, pp. 2239-2246, Jul 1999.

[Srinivas , 93]

N. Srinivas, K. Deb, *multiobjective optimization using non-dominated sorting in genetic algorithms*, Rapport technique, Department of technical Engineering , Indian Institute of Technology, Kanput, India, 1993.

[Talbi, 00]

E. G. Talbi. *Une taxinomie des métaheuristiques hybrides*. Dans ROADEF'2000, 2000.

[Talbi, 01]

E. Talbi, *Métaheuristiques pour l'optimisation combinatoire multi-objectif : Etat de l'art*, Lille : LIFL, 2001

[Talbi, 04]

H. Talbi, A. Draa et M. Batouche, *A quantum-inspired genetic algorithm for multi source affine image registration*, ICIAR'2004, LNCS 3211, pp.147-154, 2004.

[Trajkovic, 98]

M. Trajkovic and M. Hedley, *Fast corner detection, Image and Vision Computing*, 16(2) :75-87, 1998.

[Ulungu , 99]

E. Ulungu , J. Teghem, P. Fortemps, D. Tuyttens, *MOSA Method: A Tool for Solving Multiobjective Combinatorial Optimization Problems*, Journal of Multicriteria Decision Analysis, Volume 8, numéro 4, pages 221-236, 1999.

[Van, 00]

D. Van Veldhuizen, Jesse B. Zydallis and Gary B. Lamont, *Messy Genetic Algorithm Based Multi-Objective Optimization: A Comparative Statistical Analysis*, In PPSN/SAB Workshop on Multiobjective Problem Solving from Nature (MPSN), Paris, France, September 2000.

[Xue, 01]

Z. Xue, D. Shen and E. Teoh, *An efficient fuzzy algorithm for aligning shapes under affine transformations*, *Pattern Recognition*, 34(6) : 1171-1180, 2001.

[Yoo, 99]

J.Yoo and P. Hajela, *Immune network simulations in multicriterion design*. *Structural Optimization*, 18:85-94, 1999.

[Zitova,03]

B. Zitova and J. Flusser, *Image registration methods: A survey*, *Image and Vision Computing*, 21:977-1000, 2003.

[Zitzler, 98]

E. Zitzler and L. Thiele, *An evolutionary algorithm for multiobjective optimization: the Strength Pareto Approach*. Technical report, Swiss Federal Institute of technology (Zurich), 1998.

[Zitzler, 99]

E. Zitzler, K. Deb and L. Thiele, *Comparison of multiobjective evolutionary algorithms: Empirical results*, Technique 70, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of technology (ETH) Zurich, December 1999.

[Zurek, 91]

W. Zurek. *Decoherence and the transition from quantum to classical*. *Physics Today*, page 36, Octobre 1991.

