

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



N° Réf :.....

Centre Universitaire de Mila

Institut des Sciences et de la Technologie

Département de Mathématiques et Informatiques

Mémoire préparé En vue de l'obtention du diplôme de licence
En: Filière Mathématiques Fondamentales

Thème

Principes des méthodes d'optimisation multi-objectif

Préparé par :

LAIB Roukia

BAOUTA Widad

BOUDAUDI Ouidad

SANGHOUDA Nour el houda

Encadré par :

Baziniar Abdelghafour

Grade: M.A

Année universitaire : 2013 /2014

الإهداء

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

والحمد لله حمدا كثيرا وأشكره شكرا جزيلا
فهو الذي هداني و وفقني إلى ما أنا
فيه. ثمرة نجاحي

→ إلى الشمعة التي احترقت لتنير
طريقي الوردية التي ذبلت لتعطر
حياتي إلى رمز التضحية و
إلى
إلى

إليك

دوما جنبي.

→ إلى من باع عمره ليعيش حلمي إلى
من انتظر هذا اليوم إليك

الذي كان يدعنا إلى وسنأى

إليكم

إلى رفيقات أيامي
ني شغبي شغبي في الحياة

إليكن

إلى زميلاتي في العمل.
إلى كل الأصدقاء و العائلة, إلى

يد

جميع

وداد

باسم الله الرحمن الرحيم

المرسلين و خاتم النبيين محمد عليه أفضل الصلاة

التسليم الذي بلغ الرسالة و أدى الأمانة

الحمد لله حمدا كثيرا لتوفيقه لنا على نجاح هذا العمل المتواضع و المتمثل في مذكرة
التخرج و الذي اهدي ثمرته إلى:

, صاحب الفضل الكبير, ,
فالشكر لك يا رب على هذا العظيم الرائع
الذي أحبه حبا لا يعلمه سواك فأحفظه لنا و أمه

إليك أبي الغالي "نجيب"

الكلمات عن وصف حنانها و عطفها, الحروف لتقصيرها في
مدحها و شكرها,
وحياتنا, فيارب أحفظها و أدمها لـ . فماذا عسايا أقول سوى احبك يا أمي الغالي.

"حبيبة (رشيدة)"

زهرات حياتي الجميلا , إخوتي و شقيقات روعي: أمينة, زينب, خوله.

إلى بلسم الروح و عطر بالمسك يفوح , إلى العزيز على قلبي:

إلى ابن خالتي اللطيف و الجميل: سيف الدين.

عماتي و خالاتي الرائعات و العزيزات: ساسية, حسبية, حياة, ,
, , , مريم, ,
نعيمة.

: , نور الدين, فريد, إسماعيل.

: وسيلة.

: ساسية, أسمهان,

: هاجر, , يحيى, لينده, ميساء,
, رهنف, أريج, أيوب, يونس.

: الحسين, زكية, زليخة.

إلى الصديقات الوفيات: إيمان, , , , فطيمة, , خوله, أميرة,
سامية, .

إلى الاساتذة الكرام الذين رافقوني في مشواري الدر
و كعواش إسماعيل.

المشرف و صاحب التوجيهات القيمة :بزنيار عبد الغفور.

ميع الزملاء و الزميلات و جميع من وقف بجانبي و ساعدني فشكرا لهم.

رقية"

إهداء

تعبيراً عن

هدى هذا

مع تمنياتي الخالصة لهما بدوام الصحة و العافية.

: سيف الدين , إيمان , أسماء متمنية لهم

.

إلى صديقاتي : رقية , ليلي.

و إلى كل من ساعدني في انجاز هذا العمل ولو
بالكلمة الطيبة.

Remerciements

Je tiens particulièrement à remercier mon encadreur Abdalghafour BAZENIAR à son aide, son orientation, judicieuse et sa disponibilité, aussi pour la confiance, la patience et la compréhension.

Qu'il ma toujours manifesté . . . Au terme de ce travail, je tiens à exprimer mes remerciements envers toutes les personnes qui ont contribué au bon déroulement de ce mémoire.

Je tiens particulièrement à remercier mon encadreur Abdalghafour BAZENIAR à son aide, son orientation, judicieuse et sa disponibilité, aussi pour la confiance, la patience et la compréhension.

Qu'il ma toujours manifesté . . .

Table des matières

Introduction générale	1
1 Généralité sur l'Optimisation	3
1.1 Introduction	3
1.2 Les problèmes d'optimisations :	4
1.2.1 Définitions	4
1.3 Choix d'une méthode :	4
1.4 Les méthodes déterministes	4
1.4.1 Méthodes de gradient :	5
1.5 Les méthode stochastiques	7
1.6 Conclusion	7
2 les problèmes d'optimisations multi-objectifs	8
2.1 Introduction	8
2.2 Notions	9
2.2.1 Le vecteur idéal	9
2.2.2 Convexité	10
2.3 Classification des méthodes de résolution	11
2.3.1 Utilisateur	11
2.3.2 Concepteur	12
2.4 La méthode de Pareto	12
2.4.1 Optimum de Pareto	12
2.4.2 la notion de dominances	13
2.4.3 Difficultés des méthodes d'optimisations multi-objectif	14
2.5 conclusion :	15
3 Les méthodes classiques	16
3.1 Introduction	16
3.2 Les méthodes classique	17
3.2.1 Agrégation Pondérée	17
3.2.2 Méthode de ϵ contrainte	17
3.3 Les méthodes exactes	18
3.4 Les méthodes approchés	19

3.4.1	Métaheuristiques à base de solution unique	20
3.4.2	Métaheuristiques à base de population de solutions	20
3.5	Certaines méthodes d'optimisation sont spécifique à l'optimisation multi-objectif	22
3.5.1	Vetor Evaluted Genetic Algorithme(VEGA)	22
3.5.2	Multiple objectif génétic algorithm (MOGA)	23
3.5.3	Recuit simulé (MOSA)	24
3.5.4	Non Dominatet Sorting Génétic algorithm(NSGA)	24
3.6	Conclusion	25
	Conclusion générale	26
	Résumé	27

Table des figures

1.1	plus forte pente de f .	6
2.1	Fonction convexe	11
2.2	Types de résolution d'un problème multiobjectif en fonction de l'intervention du décideur	13

Introduction générale

La plupart de ces problèmes d'optimisation appartiennent à la classe des problèmes NP-difficile : classe où il n'existe pas d'algorithme qui fournit la solution optimale en temps polynomial en fonction de la taille du problème. La plupart des travaux réalisés dans ce domaine étaient dédiés à l'optimisation d'un seul objectif, or la plupart des applications réelles intègrent plusieurs objectifs souvent contradictoires à optimiser simultanément.

Depuis quelques années, l'approche Pareto définie initialement dans des travaux en économie au 19^{ème} siècle a été utilisée dans les sciences pour l'ingénieur. Cette approche a l'avantage de traiter les problèmes multiobjectifs sans transformation, sans favoriser un objectif par rapport à un autre. Tout comme pour l'optimisation mono-objectif, deux classes de méthodes de résolution pour traiter les problèmes multiobjectifs : les méthodes exactes dédiées à résoudre optimalement les petites instances et les méthodes approchées : les heuristiques et en particulier les métaheuristiques (génériques) permettant d'approximer les meilleures solutions sur les plus grandes instances [1].

La notion de solution optimale unique dans l'optimisation uni-objectif disparaît pour les problèmes d'optimisation multi-objectif au profit de la notion d'ensemble de solutions Pareto optimales [5].

Le premier chapitre est un chapitre introductif dans le quel nous présentons une vue sur les notions de l'optimisation. On présente des définitions sur la formulation d'un problème d'optimisation, ainsi la manière de choisir la meilleure méthode de résolution. L'évidence sera de parler sur les deux grandes familles de méthodes d'optimisation : les méthodes déterministes et les méthodes stochastiques. Nous terminons ce chapitre en abordant les méthodes de gradients comme méthode de résolution déterministe.

Dans le deuxième chapitre, on s'attache à étudier les problèmes d'optimisation multi-objectifs. D'abord, on parlera de diverses définitions, ensuite on présentera la classification possible des méthodes de résolution de ces problèmes. A la fin, la méthode de Pareto sera l'enjeu de notre travail.

Le troisième chapitre sera consacré aux méthodes classiques, dont nous décrivons quelques méthodes traditionnellement utilisées pour la résolution des problèmes d'optimisation multi-objectif. Toutes ces approches passent par la transformation du problème initial en un problème d'optimisation mono-objectif. On cite quelques méthodes : *Vector Evaluated Genetic Algorithm* (VEGA) ; *Multiple objective genetic algorithm* (MOGA) ; *Recuit simulé* (MOSA).

1

Généralité sur l'Optimisation

1.1 Introduction

Les problèmes d'optimisation occupent actuellement une place de choix dans la communauté scientifique. L'évolution des techniques informatiques a permis de dynamiser les recherches dans ce domaine. Le monde réel offre un ensemble très divers de problèmes d'optimisation [2] :

- problème combinatoire,
- problème à un ou plusieurs objectif(s),
- problème statique ou dynamique,
- problème dans l'incertain.

Dans les paragraphes suivants, nous présenterons une généralité sur les problèmes d'optimisation.

1.2 Les problèmes d'optimisations :

1.2.1 Définitions

Problème d'optimisation :

Un problème d'optimisation est défini par un espace de recherche des solutions ou une plusieurs fonctions objectifs. Pour modéliser un problème d'optimisation on a trois étapes très importantes sont :

Les variables :

les variables du problème peuvent être de nature diverse (réelle ; entière ; booléenne ; etc). Et exprimer des données qualitatives ou quantitatives.

Une fonction objectif :

représente le but à atteindre pour le décideur (minimisation de coût ; de durée ; d'erreur ; etc.). Elle définit espace de solution potentielle au problème.

L'ensemble des contraintes :

définit des conditions sur l'espace de recherche des solutions que les variables doivent satisfaire. Une personne dispose de 40m de fil barbelé pour clôture une parcelle rectangulaire. Quelles sont les dimensions du rectangle maximisant l'aire de la parcelle en utilisant les 40 m de fil.

Alors :

$$\begin{cases} \max f(x) \\ s/c \quad 2x + 2y = 40 \end{cases}$$

1.3 Choix d'une méthode :

La nature des variables, des domaines de définition et de critères à optimiser va influencer le choix de la méthode de résolution.

Il y a deux grandes familles de méthodes d'optimisation : les méthodes déterministes et les méthodes stochastiques.

1.4 Les méthodes déterministes

Les méthodes déterministes se caractérisent par une exploration systématique de l'espace de recherche. Il existe de nombreuses méthodes d'optimisation déterministes :

Les méthodes locales :

Qui assurent la convergence vers l'optimum de la fonction le plus proche de la solution courante en explorant son voisinage comme la méthode du simplexe de gradient.

D'une manière générale ces méthodes obéissent à l'algorithme suivant :

- a) choisi d'une première solution courante i admissible.
- b) génération d'une solution j dans le voisinage de i .
- c) si $f(j)$ est meilleur que $f(i)$ alors j devient la solution courante puis retour en (b).
- d) L'algorithme se termine lorsqu'il n'y a plus d'amélioration de la solution courante.

Les méthodes globales :

Qui s'attachent à faire converger la solution vers l'optimum globale de la fonction. Comme la méthode de tunneling, la méthode homotypique, méthode de Branch & Bound.

Ces méthodes peuvent être subdivisées en plusieurs sous classe, les méthodes : heuristiques, statistiques, mathématiques, apprentissage et méthode de Branch et Bound.

Nous allons présenter un exemple des méthodes de gradient, qui sont des méthodes de recherche locale.

1.4.1 Méthodes de gradient :

On peut définir le gradient comme :

$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \dots, \frac{\partial f(x)}{\partial x_n} \right)^t$$

Elles permettent de résoudre des problèmes non linéaires et sont basées sur une hypothèse forte la connaissance de la dérivée de la fonction objective en chacun des points de l'espace. Cette famille des méthodes procède de la façon suivante :

On choisit un point de départ x_0 et on calcule le gradient $\nabla f(x_0)$. Comme le gradient indique la direction de plus grande augmentation de f . On se pose déplace d'une quantité λ_0 dans sans apposé au gradient et on a défini le point x_1

$$x_1 = x_0 - \lambda_0 \left(\frac{\nabla f(x_0)}{\|\nabla f(x_0)\|} \right)$$

Cette procédure est répétée et engendre des points x_1, x_2, \dots, x_i . ainsi pas à pas, la distance entre le point d'indice i et l'optimum diminue

$$x_{i+1} = x_i - \lambda_i \left(\frac{\nabla f(x_i)}{\|\nabla f(x_i)\|} \right)$$

λ_i : est le pas de déplacement à chaque itération, $i \in N$.

Si λ_i est fixe ; on parle **méthode de gradient à pas prédéterminé**. L'inconvénient de cette procédure est que la convergence est très dépendant du choix du pas de déplacement. La convergence peut être très lente si le pas est mal choisi. L'intérêt principal de cette méthode est de pouvoir se généraliser aux cas de fonction non partout différentiables.

Actuellement, la méthode la plus usitée de cette famille est **la méthode de la plus forte pente**. Elle permet de se libérer du choix d'un λ_k mais elle introduit **un critère d'arrêt**, le but de cette méthode est de minimiser la fonction de λ .

$$g(\lambda) = f(x_k - \lambda * \nabla f(x_k))$$

Algorithme de la plus forte pente.

- a) Choisir un point de départ x_0 et faire $K=0$
- b) A l'itération K :

$$d_k = -\nabla f(x_k)$$

Recherche λ tel que :

$$f(x_k - \lambda_k * d_k) = \min \{f(x_k - \lambda_k * d_k)\}, \lambda \geq 0$$

$$x_{k+1} = x_k - \lambda_k * d_k$$

- c) Si le test d'arrêt est vérifié alors fin. Sinon $k = k + 1$ et retourner en b.

L'algorithme de la plus forte pente est à base de **l'algorithme de Hill-Colombin** appelé également **algorithme de descente de gradient**.

Exemple Soit $f(x) = \frac{1}{2}x_1^2 + \frac{9}{2}x_2^2$.

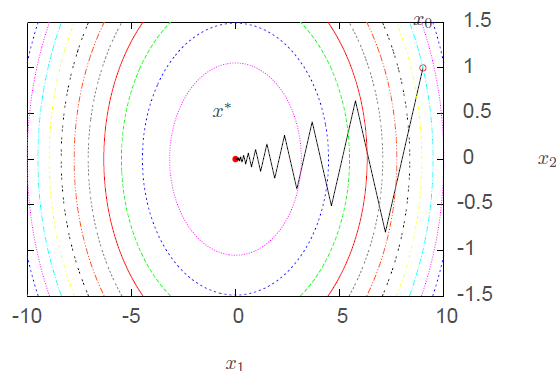


FIGURE 1.1 – plus forte pente de f .

1.5 Les méthode stochastiques

Les méthodes stochastiques sont utilisées dans le cas des problèmes où en connais pas un algorithme de résolution en temps polynomial. Evidement la solution recherchée sera approchée de l'optimum.

Dans la pratique, les méthodes stochastiques les plus utilisées sont : la méthode de Monte Carlo, le recuit simulé, les algorithmes évolutionnaires...etc.

1.6 Conclusion

Nous avons parlés dans ce chapitre sur les notions de base des problèmes d'optimisation. Ensuite, Nous avons présenté la classification des méthodes d'optimisation, en particulier la méthode de Pareto.

2

les problèmes d'optimisations multi-objectifs

2.1 Introduction

Ce chapitre est consacré à l'introduction des notions liées à l'optimisation multiobjectif sur lesquelles toute la suite de ce travail va se baser.

On va étudier les problèmes d'optimisations multi-objectif, la classification des méthodes de résolution, la méthode de Pareto dont l'optimum de Pareto et la notion de la dominance. Les difficultés des méthodes d'optimisations multi-objectifs seront cités.

2.2 Notions

Définition 2.1 Un problème multi-objectif ou multicritère peut être définie comme un problème dont on cherche l'action qui satisfait un ensemble de contraintes et optimiser un vecteur de fonctions objectif.

Un action (ou un vecteur de décision) :

On note ;

$$x = (x_1, x_2, \dots, x_n)$$

Avec x_i les variables du problème et n le nombre de variables.

Les contraintes : Une contrainte est une condition que doit satisfaire la solution d'un problème d'optimisation. On distingue deux types de contraintes : les contraintes d'égalité et les contraintes en inégalité :

$$h_i(x) = 0, \quad i \in N. \quad g_j(x) \leq 0, \quad j \in N.$$

Le vecteur de fonctions objectifs : Qui sert de critère pour déterminer la meilleure solution à un problème d'optimisation. On note f tel que :

$$f(x) = (f_1(x), f_2(x), \dots, f_n(x))$$

Avec f_i les objectifs ou critère de décision et n nombre d'objectifs.

Le modèle mathématique est le suivant :

$$\left\{ \begin{array}{l} \max f(x) = (f_1(x), f_2(x), \dots, f_n(x)) \\ h_i(x) = 0, \quad i \in N. \\ g_j(x) \leq 0, \quad j \in N. \\ x: \text{v.a} \end{array} \right.$$

Où :

$g_i(x)$: Les contraintes d'égalités ; $h_j(x)$: Les contraintes d'inégalités.

Lorsqu' un seul objectif (critère) est donné le problème d'optimisation est mono-objectif. Dans ce cas la solution optimal est clairement définie, c'est celle qui a le cout optimal (minimale, maximale).

2.2.1 Le vecteur idéal

Est l'action composée de la solution optimal pour chaque objectif pris séparément .

$$x^0 = (x_1^0, x_2^0, \dots, x_n^0).$$

Avec x_i^0 la valeur qui optimise la $i^{\text{ème}}$ fonction objectif .

La condition nécessaire et suffisante pour que ce vecteur idéal soit atteint est que les fonctions objectifs soient indépendantes. Si cette condition est réalisée alors la résolution du problème multi objectif est transformée en un résolution de plusieurs problèmes uni-objectif.

2.2.2 Convexité

L'ensemble f est dit **convexe** si tout segment joignant deux points quelconques de f est inclus dans f .

Définition 2.2 X est **convexe** : $\forall x, y \in R; \lambda \in [0, 1]$ ssi :

$$\lambda x + (1 - \lambda)y \in X.$$

Définition 2.3 Une fonction $f : x \in R^n \rightarrow R$ définie sur l'ensemble **convexe** si :

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y).$$

Exemple Montrons qu'un intervalle dans R est **convexe** tel que :

$$M = [a, b].$$

$$\text{Soit } x, y \in M : \begin{cases} a \leq x \leq b \dots\dots\dots(1) \\ a \leq y \leq b \dots\dots\dots(2) \end{cases}$$

Montrons que :

$$z = \lambda x + (1 - \lambda)y \in M, \lambda \in [0, 1].$$

(1) * λ : donne

$$\lambda a \leq \lambda x \leq \lambda b \dots\dots\dots(3)$$

(2) * λ : donne

$$\lambda a \leq (1 - \lambda)y \leq \lambda b \dots\dots\dots(4)$$

De (3) et (4) :

$$a \leq \lambda x + (1 - \lambda)y \leq b,$$

alors :

$$\lambda x + (1 - \lambda)y \in M.$$

donc :

$$z \in M.$$

en effet : M est convexe.

Exemple

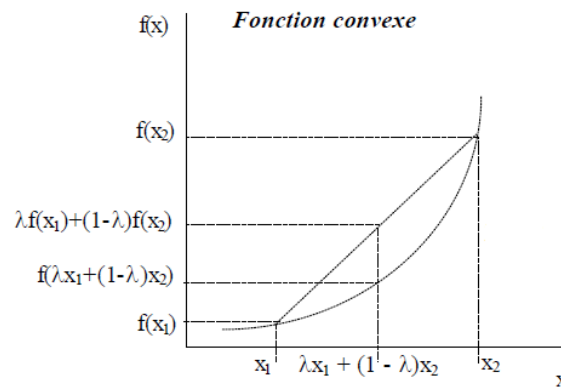


FIGURE 2.1 – Fonction convexe

2.3 Classification des méthodes de résolution

On a deux classifications différentielles des méthodes de résolution des problèmes multiobjectifs [5].

2.3.1 Utilisateur

Cette classification est essentiellement utilisée en recherche opérationnelle. Les décisions étant considérées comme un compromis entre les objectifs et les choix spécifiques du décideur (contraintes des cout, de temps, etc). Un décideur choisit une méthode en fonction de l'aide qu'elle va lui apporter.

Les méthodes a priori (decideur \rightarrow recherche) Les solutions les plus intuitives pour résoudre des problèmes multi-objectifs consistent souvent à combiner les différents fonctions objectifs en une fonction d'utilité suivant les préférences du décideur dans ce cas le décideur est supposé connaître à priori le poids de chaque objectif afin de les mélanger dans une fonction unique. Cela revient à résoudre un problème simple objectif. Cependant dans la plupart des cas, le décideur ne peut pas exprimer, clairement, sa fonction d'utilité. Soit par manque d'expérience ou d'informations, soit parce que les différents objectifs sont non commensurables.

Les méthodes a posteriori (recherche \rightarrow décideur) Le décideur prend sa décision d'après un ensemble de solutions calculées par un solveur. Dans ce cas la qualité de la décision dépend du choix de la méthode de résolution. Car elle –ci va devoir donner un ensemble de résultat le plus représentatif de l'espace des objectifs efficaces.

Les méthodes progressives ou intractives (decideur \leftrightarrow recherche) Dans ces méthodes les processus de décision et l'optimisation sont alternés. Par moment, le décideur intervient de

manière à modifier certaines variables ou contraintes afin de diriger le processus d'optimisation. Le décideur modifie ainsi interactivement le compromis entre ses préférences et les résultats. Ces méthodes exigent une connaissance approfondie, de la part du décideur. Des décideurs, des outils utilisés. [vinck 1988] présente plusieurs méthodes progressives utilisées en recherche opérationnelle. Il trouve que la relation de dominance est trop pauvre pour être utile et les fonctions d'utilité multiattribut trop riche pour être faible. Il tente d'enrichir la relation de dominance par des éléments peu discutables, les préférences solidement ét

2.3.2 Concepteur

Ce classement adopte un point de vue plus théorique articulé autour des notions d'agrégation et d'optimum de Pareto. Ces notions sont développées dans les sections suivantes car nous adoptons cette classification pour présenter les différentes méthodes.

Les méthodes agrégées : Ces méthodes transforment un problème multi-objectif en un problème simple objectif.

Les méthodes fondées sur Pareto ; Ces méthodes sont fondées sur la notion de dominance au sens de Pareto qui privilégie une recherche satisfaisant au mieux tous les objectifs.

Les méthodes non agrégées et non Pareto : Certaines méthodes n'utilisent aucun des deux concepts précédant. Alors que l'agrégation ou l'utilisation de la dominance de Pareto traitent les objectifs simultanément, en général, les méthodes dites non agrégées et non Pareto possèdent un processus de recherche qui traite séparément les objectifs.

2.4 La méthode de Pareto

L'idée d'utiliser la dominance au sens de Pareto a été proposée par Goldberg (**Goldberg 1989**) pour résoudre les problèmes proposés par Schaffre (**Schaffre 1985**). Il suggère d'utiliser le concept d'optimalité de Pareto pour respecter l'intégralité de chaque critère car il refuse de comparer à priori les valeurs de différents critères. Les utilisations d'une sélection basée sur la notion de dominance de Pareto vont faire converger la population vers un ensemble de solutions efficace. Ce concept ne permet pas de choisir une alternative plutôt qu'une autre mais il apporte un aide précieuse au décideur.

2.4.1 Optimum de Pareto

La formule de Pareto est : dans un problème multi-objectif il existe un équilibre tel que l'on ne peut pas améliorer un critère sans détériorer au moins un des autres critères. Cette équilibre a été appelé de **Pareto** un point x dit **Pareto-optimal** s'il n'est dominé par aucun autre point appartenant à E ces points également appelés **solutions non inférieures ou non dominées**.

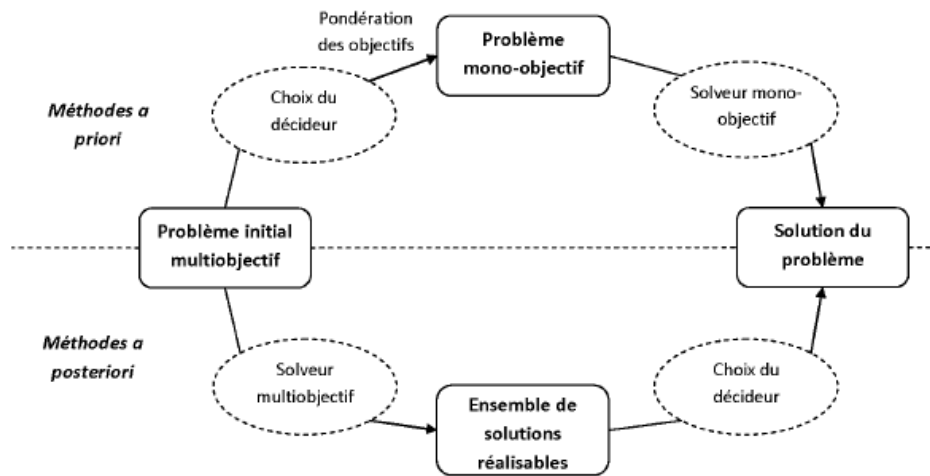


FIGURE 2.2 – Types de résolution d'un problème multiobjectif en fonction de l'intervention du décideur

2.4.2 la notion de dominances

La notion d'optimum de pareto de diviser en deux, l'ensembles des états possibles de la société ; on peut aussi distinguer :

- Ceux qui sont uniformément améliorables : il est possible d'augmenter le bien-être de certains individus sans réduire celui des autres.
- Ceux qui ne sont pas uniformément améliorables : l'augmentation du bien-être de certains individus implique la réduction du bien-être d'au moins un autre individus ce sont les états entrant dans ce deuxième cas de figure que l'on désigne comme optimaux au sens de pareto, on pareto-optimaux.

Définition 2.4 Un point $x \in E$ domine, $x' \in E$ si :

$$\forall i, f_i(x) \leq f_i(x').$$

Un point $x \in E$ est dite **faiblement non dominé** s' il n'existe pas de point $x' \in E$ tel que :

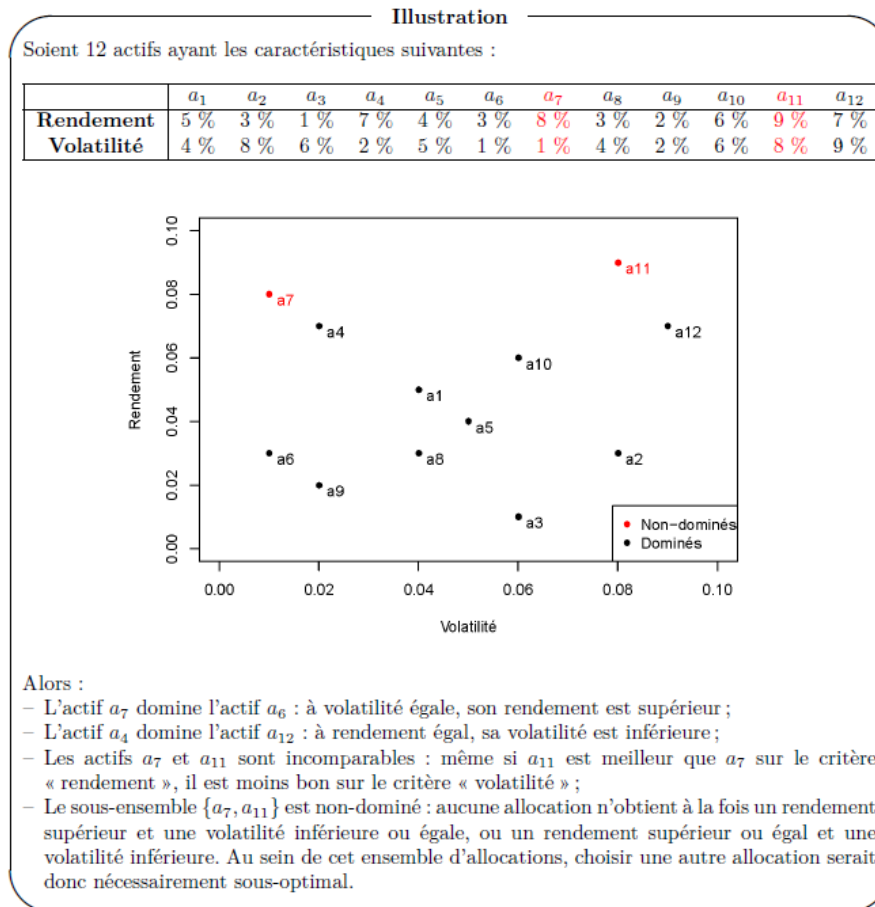
$$f_i(x') < f_i(x) \forall i = 1, \dots, k.$$

Un point $x \in E$ est **dite fortement non dominé** s' il n'existe pas de point $x' \in E$ tel que :

$$f_i(x') \leq f_i(x), \forall i = 1, \dots, k.$$

Avec au moins un i tel que $f_i(x') < f_i(x)$

Exemple



2.4.3 Difficultés des méthodes d'optimisations multi-objectif

Un processus d'optimisation multi-objectif doit résoudre les deux tâches suivantes :

- guider le processus de recherche vers la frontière de Pareto.
- maintenir une diversité des solutions pour assurer une bonne répartition sur la frontière de Pareto.

l'accomplissement de ces tâches est très délicat car les difficultés rencontrées dans un problème multi-objectif

sont identique à celle d'un problème simple objectif mais elles sont implifiées par la présence d'objectif dépendants les uns des autres.

Guider le processus de recherche vers la frontière de Pareto :

Le processus de recherche est souvent relenti ou totalement dèrouté par des fonctions possédant une des caractéristiques suivantes : **multimodalité, isolation d'un optimum ou tromperie.**

la multimodalité :

Un fonction est dite multimodale si elle possède plusieurs optimaux globaux. Dès lors, chaque optimum exerce sur les individus d'une population une attraction différentes qui peut piéger le processus de convergence de l'olgorithme. Ce problème peut être évité en utilisant une technique de répartition des individus de type sharing ou crowding.

l'isolation d'un optimum :

Il existe des problèmes dans lesquels un optimum peut être entouré de grandes zones pratiquement plate. Cet optimum se trouve alors isolé car l'espace de recherche qui l'entoure ne peut pas guider vers lui les individus de la population.

le tromperie :

Un problème est trompeur lorsqu'il guide la convergence vers une zone non optimal de la fonction.

Maintenir la diversité sur la frontière de Pareto :

La difficulté à maintenir une bonne répartition des solutions sur la frontière de Pareto résulte principalement des caractéristiques suivantes : **convexité ou non convexité de la frontière de Pareto, discontinuité de cett frontière et non uniformité de la distribution.**

convexité ou non convexité :

Certains problème ont une frontière de Pareto non convexe. Les méthodes dans le calcul de la fitness est basé sur le nombre d'individus dominé vont être moins efficaces.

discontinuité :

Si une frontière de Pareto est discontinuité, nous trouvons le mémé principe que pour une fonction multimodale, les différentes parties de cette frontière vont exercer proportionnellement à leur taille, une attraction plus ou moins importante sur les individus d'une population. Certaines d'entre elles pourront donc ne pas être découvertes.

Les méthodes basées sur une population génétique sont plus sensibles à ce phénomène que les méthodes utilisant des stratégies d'évolution.

la non uniformité de répartition sur la frontière :

La solution sur la frontière de Pareto peuvent ne pas être réparties uniformément. La raison principale vient du choix des fonctions objectifs. Par exemple, si une des fonctions objectifs est multimodale, elle va influencer de manière très différente la répartition des solutions sur la frontière de Pareto

2.5 conclusion :

Dans le deuxième chapitre nous avons défini les problèmes d'optimisation multi-objectif et la classification des méthodes de résolution. La méthode de pareto, l'optimum de pareto et la notion de la dominance été expliqués. Enfin nous avons abordés quelques difficultés sur l'application des méthodes d'optimisation multi-objectif.

3

Les méthodes classiques

3.1 Introduction

Dans ce chapitre nous avons connus les méthodes classique qui transformer les problèmes multi-objectifs au problèmes mono-objectif , les méthodes exactes, les méthodes approchées etcertaines méthodes d'optimisation sont spécifique à l'optimisation multi-objectif.

3.2 Les méthodes classique

Dans cette section, nous décrivons quelques méthodes traditionnellement utilisées pour la résolution des problèmes d'optimisation multi-objectif. Toutes ces approches passent par la transformation du problème initial en un problème d'optimisation mono-objectif. Nous détaillerons, dans ce qui suit, les deux méthodes les plus connues et utilisées : l'optimisation par agrégation pondérée des objectifs, et l'approche par ε -contraintes [3].

3.2.1 Agrégation Pondérée

Cette méthode consiste à ramener le problème multi-critère au problème de l'optimisation d'une combinaison linéaire des objectifs initiaux. Les coefficients sont généralement choisis en fonction de l'importance relative des objectifs. Soit

$$f_w(x) = \sum_{m=1}^M w_m f_m(x)$$

Où les poids $w_m \geq 0$ sont tels que $\sum_{m=1}^M w_m = 1$. La méthode "de la somme pondérée" pour la résolution du problème consiste donc à résoudre le problème d'optimisation suivant :

$$\text{Minimiser } f_w(x)$$

La figure ; illustre le processus de minimisation de la combinaison linéaire de deux critères. En effet, la minimisation de la somme pondérée de deux objectifs correspond au "déplacement" d'une droite définie par la ratio $-w_1/w_2$ (tracée dans l'espace des critères). Ce processus s'arrête au moment où la droite atteint la position tangente à la frontière du domaine de recherche (dans l'espace des critères).

3.2.2 Méthode de ε contrainte

Cette méthode consiste à retenir une des fonctions qu'il faut optimiser comme unique objectif et transformer les critères restants en contraintes. Le problème d'optimisation peut être donc reformulé sous la forme suivante :

$$\begin{cases} \min_{x \in S} f_\mu(x) \\ f_m(x) \leq \varepsilon_m, \forall m = 1, \dots, M \text{ telque } m \neq \mu \end{cases}$$

Où $\mu \in \{1, \dots, M\}$ et les paramètres ε_m sont à définir par l'utilisateur.

Supposons, par exemple, que lors de la résolution d'un problème de minimisation de deux objectifs, nous ayons décidé de retenir f_2 comme fonction à optimiser sous la contrainte $f_1(x) \leq \varepsilon_1$. Le domaine de la recherche est divisé dans l'espace des critères par la droite verticale $f_1 = \varepsilon_1$. En deux parties : la partie faisable ($f_1(x) \leq \varepsilon_1$), et la partie infaisable ($f_1(x) > \varepsilon_1$). Résoudre le problème (1) revient donc à trouver un point faisable tel que la valeur de $f(x)$ soit la plus petite possible .

3.3 Les méthodes exactes

Les méthodes exactes cherchent à trouver de manière certaine la solution optimale en examinant de manière explicite ou implicite la totalité de l'espace de recherche. Elles ont l'avantage de garantir la solution optimale néanmoins le temps de calcul nécessaire pour atteindre cette solution peut devenir très excessif en fonction de la taille du problème (explosion combinatoire) et le nombre d'objectifs à optimiser. Ce qui limite l'utilisation de ce type de méthode aux problèmes bi-objectifs de petites tailles. Ces méthodes génériques sont : Branch & Bound, Branch & Cut, Branch & Price, Branch, Cut & Price. D'autres méthodes sont moins générales, comme : La programmation dynamique, simplexe, La programmation linéaire en nombre entiers, L'algorithme A' . D'autres méthodes sont spécifiques à un problème donné comme l'algorithme de Johnson pour l'ordonnement.

Branch & Bound : Est une méthode générique de résolution exacte de problèmes d'optimisation, et plus particulièrement d'optimisation combinatoire. C'est une méthode constructive de recherche arborescente qui utilise l'énumération implicite basée sur la notion de bornes afin d'éviter l'énumération de larges classes de mauvaises solutions. Elle utilise la stratégie diviser pour régner, en se basant sur deux concepts : le branchement (séparation) qui consiste à partitionner ou diviser l'espace des solutions en sous problèmes pour les optimiser chacun individuellement ; et l'évaluation qui consiste à déterminer l'optimum de l'ensemble des solutions réalisables associé au nœud en question ou, au contraire, de prouver mathématiquement que cet ensemble ne contient pas de solution optimale, la méthode la plus générale consiste à borner le coût des solutions contenues dans l'ensemble. Les techniques les plus classiques pour le calcul de bornes sont basées sur l'idée de relaxation de certaines contraintes : relaxation continue, relaxation lagrangienne. ...

Branch & Cut : Padberg et Rinaldi [Padberg, 1991] ont amélioré l'idée du B&B basé sur la programmation linéaire en décrivant une méthode utilisant des inégalités renforçant la relaxation par programmation linéaire. Le principe de base est le même que pour le B&B, mais au lieu de brancher sur une variable pour descendre dans l'arbre de recherche, on recherche des plans de coupes qui permettent de restreindre l'espace des solutions réalisables.

Branch & Price Les algorithmes de B&B résolvant des programmes linéaires en générant les variables dynamiquement quand cela est nécessaire sont appelés algorithmes de Branch & Price. Les inégalités d'un programme linéaire en nombre entier peuvent être vues comme une matrice, chaque colonne correspondant à une variable, et chaque ligne correspondant à une inégalité. Comme pour les plans de coupes, les colonnes de la matrice correspondant aux inégalités prises en compte peuvent être définies implicitement si le nombre de variables est grand. Si une colonne n'est pas présente dans la matrice courante, alors la variable correspondante prend implicitement la valeur zéro. Le processus de générer dynamiquement les variables est appelé 'pricing'.

Branch, Cut & Price Lorsque les variables et les plans de coupes sont générés dynamiquement durant l'algorithme de B&B, on appelle cette technique Branch, Cut & Price. Il existe une certaine symétrie entre la génération de coupes et de variables. Cependant, même si les méthodes de B&C et de B&P sont proches, combiner les deux approches requiert la mise en place de méthodes sophistiquées. Un aperçu des différents travaux répertoriés utilisant cette approche est consultable à l'adresse :

Méthode à deux phases La méthode deux-phases a initialement été proposée par Ulungu et Teghem pour la résolution d'un problème d'affectation bi-objectif [Ulungu, 1995]. Comme son nom l'indique, cette méthode est décomposée en deux étapes : la première consiste à trouver toutes les solutions supportées du front Pareto, puis la deuxième phase cherche de façon indépendante les solutions non supportées situées entre tous les couples de solutions supportées adjacentes. Cette méthode travaille donc essentiellement dans l'espace objectif. Durant la première phase de la méthode, les deux solutions extrêmes (solutions optimisant chacune un des deux objectifs) sont recherchées. Puis, de façon récursive, dès que deux solutions supportées r et s sont trouvées, la méthode recherche d'éventuelles autres solutions supportées entre r et s , à l'aide de combinaisons linéaires bien choisies des objectifs. A la fin de la première phase l'ensemble des solutions supportées est donc trouvé. Cette première phase rappelle la méthode dichotomique. La deuxième phase consiste alors en la recherche des solutions non supportées appartenant au front Pareto. Ces solutions ne peuvent être obtenues par combinaisons d'objectifs. Ulungu et Teghem proposent alors d'utiliser les solutions supportées trouvées pour réduire l'espace de recherche en argumentant que les solutions Pareto non supportées restantes sont forcément dans les triangles rectangles basés sur deux solutions supportées consécutives. Ainsi, une recherche de type deuxième phase est exécutée entre chaque couple de solutions supportées adjacentes. La méthode de recherche au sein de ces triangles dépend du problème étudié. A la fin de la deuxième phase, toutes les solutions Pareto sont trouvées.

3.4 Les méthodes approchées

Méthodes souvent inspirées de mécanismes d'optimisation rencontrés dans la nature. Elles sont utilisées pour les problèmes où on ne connaît pas d'algorithmes de résolution en temps polynomial et pour lesquels on espère trouver une solution approchée de l'optimum global. Elles cherchent à produire une solution de meilleure qualité possible dictée par des heuristiques avec un temps de calcul raisonnable en examinant seulement une partie de l'espace de recherche. Dans ce cas l'optimalité de la solution n'est pas garantie ni l'écart avec la valeur optimale. Parmi ces heuristiques, on trouve les métaheuristiques qui fournissent des schémas de résolution généraux permettant de les appliquer potentiellement à tous les problèmes. Plusieurs classifications des métaheuristiques ont été proposées, la plupart distinguent globalement deux catégories : celles se basant sur une solution unique et celles se basant sur une population de solution.

3.4.1 Métaheuristiques à base de solution unique

Travaillent sur un seul point de l'espace de recherche à un instant donné en commençant avec une solution initiale puis de l'améliorer itérativement en choisissant une nouvelle solution dans son voisinage. Voisinage de taille ϵ : $N(X') = \{X \in S / \|X - X'\| < \epsilon\}$. Minimum local : $\forall X \in N(X')$ on a $F(X') \leq F(X)$

On définit la notion d'optimalité locale au sens Pareto : un sous-ensemble de solutions non dominées dans un voisinage déterminé.

Algorithmes de recherche locale pour une optimisation locale

Tout algorithme basé sur la notion de voisinage, piégé par le premier optimum local. Le principe, simple, à partir d'une configuration initiale X_0 on applique successivement des transformations à la solution courante (mouvement). La recherche consiste à passer d'une solution à une solution voisine. De manière générale, les opérateurs de recherche locale s'arrêtent quand une solution localement optimale est trouvée, c'est à dire quand il n'existe pas de meilleure solution dans le voisinage.

Les algorithmes de descente : - Descente simple « simple descent »

Le principe est simple, à partir d'une solution existante, chercher aléatoirement une solution dans le voisinage et accepter cette solution si elle améliore la solution courante.

- Plus grande descente « Deepest descent »

Est une version plus agressive de l'algorithme de descente. Au lieu de choisir une solution X' dans le voisinage de X , on choisit toujours la meilleure solution X' du voisinage de X . et on se trouve très rapidement bloqué dans un optimum local.

- Multi-start descent

Dans les deux algorithmes précédents, l'équilibre souhaité entre intensification et diversification n'existe donc plus. Un moyen très simple de diversifier la recherche peut consister à réexécuter un des algorithmes en prenant un autre point de départ. Comme l'exécution de ces algorithmes est souvent très rapide, on peut alors inclure cette répétition au sein d'une boucle générale. On obtient alors un algorithme de type « Multi-start descent ». Dans ces algorithmes, il est clair que si la fonction comporte beaucoup de minima locaux, on est certain de rester bloqué sur l'un d'eux. Il serait alors intéressant de pouvoir s'échapper de ces minima locaux : la stratégie consiste à favoriser les descentes, mais sans interdire tout à fait les remontées.

3.4.2 Métaheuristiques à base de population de solutions

Travaillent sur un ensemble de points de l'espace de recherche en commençant avec une population de solution initiale puis de l'améliorer au fur et à mesure des itérations. L'intérêt de ces méthodes est d'explorer un très vaste espace de recherche et d'utiliser la population comme facteur «de diversité» de plus elle sont très adaptées et très largement utilisées pour l'optimisation multiobjectifs.

Algorithmes Evolutionnaires :

Un algorithme évolutionnaire est typiquement composé de trois éléments fondamentaux :

- **une population** constituée de plusieurs individus représentant des solutions potentielles (configurations) du problème donné, permettant de mémoriser les résultats à chaque étape du processus de recherche.
- **un mécanisme d'évaluation** (fitness) des individus permettant de mesurer la qualité de l'individu,
- **un mécanisme d'évolution** de la population permettant, grâce à des opérateurs prédéfinis (tels que la sélection, la mutation et le croisement), d'éliminer certains individus et d'en créer de nouveaux. Ces méthodes sont applicables dans la plupart des problèmes d'optimisation (multimodaux, non continu, contraints, bruités, multiobjectif, dynamiques, etc.). On peut distinguer quatre grandes classes d'algorithmes évolutionnaires qui se différencient par leur manière de représenter l'information et par leur façon de faire évoluer la population d'une génération à l'autre :

Programmation évolutive (Evolutionary Programming) Ce modèle évolutionniste accente l'utilisation de la mutation et n'utilise pas dans sa version originale la recombinaison des individus par croisement. Développé à l'origine pour l'évolution d'automates à état fini, ce modèle est souvent appliqué à la résolution de problèmes d'optimisation à variables réelles. Dans ce cas, il utilise une mutation qui consiste à ajouter une perturbation Gaussienne à chaque composante du vecteur à variables réelles constituant l'individu. Cette perturbation est basée sur la performance de l'individu : l'idée consiste à faire subir des mutations importantes aux mauvais individus et inversement des mutations faibles aux bons individus. L'opérateur de sélection est de type probabiliste : ils'agit de la méthode du tournoi basée sur une compétition entre individus choisis aléatoirement.

Stratégies d'évolution Elles ont été développées [Schwefel, 1995] pour résoudre des problèmes d'optimisation industriels à variables réelles dans lesquels il n'existe pas de fonction objective analytique. Ce modèle utilise le principe de mutation sur les réels du modèle de la Programmation Evolutive. Cependant, ce principe a été affiné de sorte que la fonction de perturbation Gaussienne est contrôlée par l'ensemble de la population courante. Si la proportion de mutation réussie est élevée, l'espace de recherche exploré est restreint autour d'un optimum local, il faut donc diversifier la population en augmentant le taux de mutation. Ces approches utilisent un opérateur de sélection de type déterministe : les solutions dont la fitness est mauvaise sont éliminées de la population. En outre, dans le modèle original, les populations des parents et de leurs descendants sont généralement de taille différente.

Algorithmes génétiques (Genetic Algorithms) En 1809 J.B Lamarck émit l'hypothèse de l'évolution (les organismes s'adaptent en fonction de leurs besoins). En 1859 C. Darwin émit l'idée de la sélection naturelle (dans toute espèce les meilleurs sont sélectionnés). Les bases de l'évolution étaient posées. En 1901 De Vries exposa sa théorie du mutationnisme (les variations responsables de l'évolution ne se faisaient pas dans le temps mais de façon soudaine).

ine et se produisaient dans l'œuf). Les bases de la génétique étaient posées. Ces concepts vont être utilisées en Informatique : **En 1975 Jhon Holland** proposa l'Algorithme Génétique. En 1989 Goldberg exposa les fondements mathématiques des AGs.

Programmation génétique (Genetic Programming) Est une extension du modèle d'apprentissage des algorithmes génétiques à l'espace des programmes. Les individus formant une population sont donc des programmes candidats à la résolution d'un problème. Ces programmes sont exprimés sous la forme d'arbres sur lesquels les opérateurs génétiques produisent des transformations en vue d'obtenir un programme qui satisfait la résolution du problème choisi.

3.5 Certains méthodes d'optimisation sont spécifique à l'optimisation multi-objectif

3.5.1 Vector Evaluated Genetic Algorithm (VEGA)

En 1985 **schaffer** propose une extension d'un algorithme génétique simple pour la résolution d'un problème multi-objectif. Cette méthode est appelée **vector evaluated genetic algorithm**. La seule différence avec un algorithme génétique simple est la manière dont s'effectue la sélection. L'idée est simple. Si nous avons **K** objectifs et une population de **N** individus, une sélection de **N/K** individus est effectuée pour chaque objectif. Ainsi **K** sous-populations vont être créées, chacune d'entre elles contenant les **N/K** meilleurs individus pour un objectif particulier. Les **K** sous-populations sont ensuite mélangées afin d'obtenir une nouvelle population de taille **N**. Le processus se termine par l'application des opérateurs génétiques de modification (croisement et mutation).

Discussion :

La méthode **VEGA** a tendance à créer des sous-populations dont les meilleurs individus sont spécialisés pour un objectif particulier. L'évolution de la population favorise l'apparition des espèces. En effet, comme la méthode de sélection ne tient compte que d'un seul objectif, elle privilégie les individus qui obtiennent une bonne performance pour cet objectif. Dès lors ces individus ne seront sélectionnés que lorsqu'on effectuera la sélection sur cet objectif. Les individus que **schaffer** appelle les individus "**milieu**", parce qu'ayant une performance générale acceptable mais ne possédant aucun critère fort, vont être éliminés car ils ne seront sélectionnés dans aucune sous-population. Cette disparition entraîne la spécialisation des individus pour chaque objectif. Ce résultat est contraire au but initial de la méthode qui était de trouver un compromis entre différents critères.

schaffer propose deux heuristiques améliorer sa méthode :

-la première est un croisement restreint qui ajoute une préférence pour sélectionner les parents non dominés. Cette méthode a tendance à éviter la disparition des individus "**milieu**" mais elle a tendance également à accentuer la convergence.

-la seconde encourage le croisement entre individus spécialisés sur les objectifs. Mais les effet sont identiques à la première heuristique.

Critique :

Malgré ces imperfections, cette méthode est très souvent utilisé car facilement implémentable dans un algorithme génétique classique. L'utilisateur peut associer à **VEGA** n'importe quel mode de sélection (tournoi, roulette, rang). Mais comme le tournoi est une technique de sélection plus élitiste que les deux autre méthodes, son utilisation accentue le phénomène de spécialisation.

3.5.2 Multiple objectif génétic algorithm (MOGA)

En 1993 Fonesca et Fleming on proposé une méthode dans laquelle chaque individu de la population est rangé en fonction du nombre d'individus que le dominant. Ensuite, ils utilisent une fonction de notation permettant de prendre en compte le rang de l'individu et le nombre d'individus ayant même rang.

soit un individu x_i à la génération t . Dominé par $p_i(t)$ individus. le rang de cet individus est :

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

Tous les individus non dominés sont de rang 1.

Fonesca et Fleming en 1993 calculent la fitness de chaque individu de la façon suivantes :

- a) calcul du rang de chaque individu.
- b) affectation de la fitness de chaque individu par application d'une fonction de changement d'échelle sur la valeur de son rang. Cette fonction est en général linéaire. Suivant le problème d'autres types de fonction pourront être envisagé afin d'augmenter ou diminuer l'importance des meilleurs rangs ou d'atténuer la largeur de l'espace entre les individus de plus fort rang et de plus bas rang.

Discussion :

L'utilisation de selection par rang a tendance à répartir la population autour d'une même optimum. Or cela n'est pas satisfaisant pour un décideur car cette méthode ne lui proposera qu'une seul solution. Pour éviter cet dérivé, les natures utilisent une fonction de **sharing**. Ils espèrent ainsi répartir la population sur l'ensemble de la frontière de Pareto.

Critique :

Le **sharing** utilisé dans cette méthode agit sur l'espace des objectifs. Cela suppose que deux actions qui ont le même résultat dans l'espace des objectifs ne pourront pas être représentés dans la population.

3.5.3 Recuit simulé (MOSA)

La méthode de recuit simulé est basée sur une analogie avec le processus physique de recuit des matériaux cristallins. Cette méthode du recuit simulé appliquée aux problèmes d'optimisations. Considère une solution initial et recherche dans son voisinage une autres solution de façon aléatoires. L'originalité de cette méthode est qu'il est possible de diriger vers une solution voisine optima locaux. Au début de l'algorithme, un paramètre T , apparenté à la température, est déterminé et décroît tout long de l'algorithme pour tendre vers 0. La valeur de ce température va dépendre la probabilité P d'acceptation des solutions inappropriées (plus la température T est élevée, plus cette sera forte).

La performance du recuit simulé dépend, entre autre, de la règle de refroidissement (c'est-à-dire la décroissance du paramètre T) que l'on utilise. Un refroidissement trop rapide mènerait vers optimum local pouvant être de très mauvaise qualité. un refroidissement trop lent serait très coûteux en temps de calcul. Le règle des différents paramètres (température initiale, nombre d'itération pour le température, décroissance de la température,...) peut être long et difficile.

Algorithme :

- a) A partir d'un point initial x_0 on effectue un déplacement aléatoire (changement d'état).
- b) Si le déplacement mène à x_1 tel que $f(x_1) \leq f(x_0)$ alors x_1 est accepté, sinon, il est accepté avec une probabilité

$$p = \exp\left(\frac{-|\nabla(f)|}{KT}\right)$$

$\nabla(f)$: Représente la distance de déplacement : $x_1 - x_0$

T : Est assimilé à une température décroissante au cours du temps.

K : Est une constante.

3.5.4 Non Dominatet Sorting Génétic algorithm(NSGA)

Dans la méthode proposée par [srivinas.and.deb.1993] le calcul de la fitness s'affecte en séparant la population en plusieurs groupes en fonction du degré de domination au sens de pareto de chaque individu [4].

Algorithme de la fonction notation

- a) Dans la population entière on recherche les individus non dominés ces derniers constituent la première frontière de pareto.
- b) On leur attribue une valeur de fitness factice. Cette valeur est supposée donner une chance égale de reproduction à tous ces individus. Mais pour maintenir la diversité dans la population, il est nécessaire d'appliquer une fonction de sharing sur cette valeur.
- c) Ensuite, ce premier groupe d'individus est supprimé de la population.

- d) On recommence cette procédure pour déterminer la seconde frontière de Pareto. La valeur factice de fitness attribuée à ce second groupe est inférieure la plus petite fitness après application de la fonction de **sharing** sur le premier groupe. Ce mécanisme est répété jusqu'à ce que l'on ait traité tous les individus de la population.

critique :

Cette méthode paraît moins efficace en temps de calcul que la méthode **MOGA** car le temps de calcul de la notation (tri de la population et "sharing") est important. Mais l'utilisation de "sharing" sur l'espace d'état et le tri de solution en différentes frontières semblent plus appropriés à maintenir une grande diversité de la population et à répartir plus efficacement les solutions sur la frontière de Pareto. De plus, cette méthode est applicable dans les problèmes avec un nombre quelconque d'objectifs.

Trois critiques ont été soulevées pour cette méthode, à savoir :

- Sa complexité de calcul de $O(K.N^3)$ avec K le nombre d'objectifs et N la taille de la population, essentiellement due au processus de tri de la population et d'application de heuristique de partage.
- Son approche non élitiste. Le tri de la population est une heuristique intéressante pour distribuer la population sur l'ensemble de la frontière de Pareto mais cette procédure ralentit le processus de convergence de l'algorithme. De plus cet effet est accentué par l'utilisation de la méthode de sélection par reste stochastique.
- La nécessité de spécifier un paramètre de "sharing".

3.6 Conclusion

Dans ce chapitre nous sommes intéressés en général aux méthodes classiques de résolution des problèmes d'optimisation multi-objectif.

Conclusion générale

L'objectif de ce travail était de présenter les travaux de recherche sur les méthodes d'optimisation multiobjectif. nous avons vu que cette problématique est abordée par la communauté scientifique suivant deux approches. La première approche tente de ramener un problème multiobjectif à un problème mono-objectif. La deuxième approche adopte un point de vue plus global en utilisant la notion de dominance au sens de pareto.

Dans le premier chapitre. Nous avons abordés les différentes notions et concepts d'optimisation. On a présenté la manière de choisir la méthode de résolution adéquate.

Dans le deuxième chapitre, on a consacré toute notre intention sur les problèmes multiobjectif. La méthode de pareto, l'optimum de Pareto et la notion de la dominance ont été abordés. Ensuite nous avons fournis certaines informations sur les difficultés rencontrés dans l'application des méthodes d'optimisation multi-objectif.

Dans le troisième chapitre nous avons présentés les méthodes classiques, dont nous avons décrit quelques méthodes utilisées pour la résolution des problèmes d'optimisation multi-objectif. Toutes ces approches passent par la transformation du problème initial en un problème d'optimisation mono-objectif. À la fin nous sommes penchés à décrire quelques méthodes : *Vector Evaluated Genetic Algorithm (VEGA)* ; *Multiple objectif génétic algorithm (MOGA)* ; *Recuit simulé (MOSA)*.

Le choix de la méthode de résolution à mettre en oeuvre dépendra souvent de la complexité du problème. Si le problème est de petite taille, alors un algorithme exact permettant de trouver la solution optimale peut être utilisé. Il est nécessaire de faire appel à des (méta)heuristiques permettant de trouver de bonnes solutions approchées.

Bibliographie

- [1] M. Basseur. « *Conception D'algorithmes Coopératifs Pour L'optimisation Multiobjectif : Application Aux Problèmes D'ordonnancement De Type Flow-Shop* ». Université des sciences et technologies de Lille U.F.R. D'I.E.E.A. thèse pour obtenir le grade de Docteur de l'U.S.T.L. 2005.
- [2] O. Hajji. « *contribution au développement des méthodes d'optimisation stochastiques, Application à la conception des dispositifs électronique* ». Thèse de Doctorat, Université des sciences et technologies de Lille. 2003.
- [3] M. Merdjiaoui Brahim. « *optimisation multi-objectif par algorithmes génétiques et approche Pareto des paramètres d'usinage sous contraintes des limitations de production* ». Mémoire de majester, Université M'hamed Bougara Boumardes. 2006.
- [4] A. Berro. « *optimisation multi-objectif et stratégies d'évolution en envirement dynamique* ». Thèse de Doctorat, Université des sciences sociale Toulouse 1. 2001.
- [5] A. BEN ABDALLAH. « *Optimisation multi-objectif évolutionnaire* ». Mémoire de Mastère d'Ingénierie Mathématique. Ecole de polytechnique de tunisie. 2004.

Résumé

L'objet de ce mémoire est présenté les travaux de recherche sur les méthodes d'optimisation multiobjectif. Cette problématique est abordée par la communauté scientifique suivant deux approches. La première approche tente de ramener un problème multiobjectif à un problème mono-objectif. La deuxième approche utilise la notion de dominance au sens de Pareto. On présente quelques méthodes comme : *Vector Evaluated Genetic Algorithm (VEGA)*; *Multiple objective genetic algorithm (MOGA)*; *Recuit simulé (MOSA)*. Le choix de la méthode dépendra souvent de la complexité du problème.

Mots clés : Optimisation, Optimisation Multiobjectif, Méthode de Pareto, Complexité du Problème.