

N° d'ordre :17/2015-M/MT

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère De L'Enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie Houari Boumediene

Faculté des Mathématiques



MEMOIRE

Présenté pour l'obtention du diplôme de MAGISTER

En : MATHEMATIQUE

Spécialité : Recherche Opérationnelle

Par : DRICI Wassila

Sujet

**Optimisation d'un critère quadratique sur
l'ensemble efficient d'un problème
multi-objectif**

Soutenu publiquement, le 30/09/2015, devant le jury composé de :

M H. AIT HADDADÈNE	Professeur à l'USTHB	Président
M M. MOULAÏ	Professeur à l'USTHB	Directeur de mémoire
M A. BERRACHEDI	Professeur à l'USTHB	Examineur

Table des matières

Introduction Générale	i
1 La Programmation Quadratique	1
1.1 Introduction	1
1.2 Éléments de l’algèbre linéaire	4
1.3 Notion sur l’optimisation convexe	5
1.3.1 Fonctions convexes généralisées	7
1.3.2 Condition d’optimalité en optimisation non linéaire	9
1.4 La Programmation Quadratique	12
1.4.1 Formulation du problème	12
1.4.2 Programmation quadratique semi-définie	13
1.4.3 Résolution d’un programme quadratique semi-défini	13
1.4.4 Programmation quadratique indéfinie	16
1.4.5 Résolution d’un programme quadratique indéfini	17
1.5 Conclusion	19
2 Généralité sur l’Optimisation Multi-objectif	20
2.1 Notions fondamentales	20
2.1.1 Position du problème	21
2.1.2 La relation de dominance	21
2.1.3 Optimalité de Pareto	22
2.1.4 Dominance au sens de Geoffrion	25
2.1.5 Caractérisation des solutions efficaces	26
2.2 Choix de la méthode d’aide à la décision	27
2.3 Approches de résolution d’un problème multi-objectif (MOP)	28
2.3.1 Approche unicritère	28
2.3.2 Approche multi-objectif	32
2.4 Conclusion	33

3	Programmation Linéaire Multi-objectif en Nombre Entiers MOILP	35
3.1	Formulation du problème	36
3.2	Solutions supportées et non supportées	36
3.3	Quelques méthodes de résolution d'un problème <i>MOILP</i>	38
3.3.1	Méthode de D.Klein & E.Hannan :	38
3.3.2	Méthode de Gupta et Malhotra	39
3.3.3	Méthode de M.Abbas et M.Moulaï	41
3.3.4	Méthode de M.Abbas et D.Chaabane	44
3.3.5	Méthode A.Sylva et J.Crema	45
3.4	Conclusion	47
4	Optimisation d'un critère sur un Ensemble Efficient	48
4.1	Introduction	48
4.2	Formulation du problème	49
4.3	Méthode de résolution de (P_E) dans le cas discret	49
4.3.1	Méthode de Jorge	50
4.3.2	Méthode de Chaabane et al.	52
4.4	Conclusion	54
5	Optimisation d'un Critère Quadratique sur l'Ensemble Efficient d'un Problème Multi-objectif Linéaire Discret	55
5.1	Introduction	55
5.2	Optimisation d'un critère quadratique indéfini sur l'ensemble efficient d'un problème <i>MOILP</i>	56
5.2.1	Première approche	57
5.2.2	Deuxième approche	64
5.2.3	Expérience numérique	72
5.3	Optimisation d'un critère quadratique semi-défini sur l'ensemble efficient d'un problème (<i>MOILP</i>)	74
5.3.1	Description de la méthode	75
5.3.2	Exemple numérique	78
5.3.3	Expérience numérique	81
5.4	Conclusion	83
	Conclusion Générale	84
	Bibliographie	85

À **ma mère**,
mon père,
mon frère mes sœurs,
mon fiancé,
À tous ceux qui m'ont témoigné leur soutien,
je dédie le présent mémoire.

Remerciements

Premièrement, je remercie Dieu le Miséricordieux, pour m'avoir donné la volonté et la force pour accomplir ce modeste travail.

Je tiens à exprimer ma profonde gratitude au professeur M. MOULAI, mon directeur de mémoire, et mon guide, pour la confiance qu'il m'a faite en acceptant de diriger mes recherches, et pour ses précieux conseils et orientations, ainsi que pour l'intérêt particulier qu'il a accordé à ce travail. Je ne le remercierai jamais assez pour la grande contribution et l'aide qu'il m'a apporté pour l'aboutissement de ce travail.

J'adresse mes remerciements au professeur H. AIT HADDADÈNE, pour l'honneur qu'il me fait en acceptant de présider le jury de ce mémoire.

Je remercie le professeur A. BERRACHEDI d'avoir accepté de juger ce travail.

Je remercie tous mes collègues et amis pour leurs aides et soutiens et tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Je n'oublierai pas de remercier ma chère famille et mon fiancé qui m'ont toujours encouragé et soutenu.

Introduction Générale

L'une des principales missions pour laquelle la recherche opérationnelle s'est vouée est d'aider à la prise de décision et à la gestion.

Les modèles traditionnels développés dans le cadre des méthodes quantitatives de gestion considéraient en général un critère unique, pour lesquels il existe des solutions optimales. Les algorithmes mis au point consistent alors à définir un moyen d'atteindre, le plus rapidement possible, une telle solution. Cependant, dans de nombreux cas, cette modélisation des problèmes ne traduit pas exactement la réalité à appréhender.

Une autre façon de modéliser les problèmes a vu le jour, il y a maintenant une trentaine d'années, permettant une représentation fidèle de la réalité. La nouveauté consiste à optimiser plusieurs critères éventuellement conflictuels simultanément.

La difficulté principale de tels problèmes est liée à la présence de conflits entre les divers objectifs. En effet, les solutions optimales pour un objectif donné ne correspondent généralement pas à celles des autres objectifs pris indépendamment. De ce fait, il n'existe, la plupart du temps aucun point de l'espace de recherche où tous les critères sont optimales simultanément, on parle dans ce cas de problème d'optimisation multi-objectif (ou multicritère) qui définit un ensemble de solutions acceptables assurant un compromis entre les objectifs considérés, appelées *solutions de compromis* ou *solutions efficaces*.

Dans de nombreuses situations pratiques, il est nécessaire d'utiliser des variables discrètes dans la modélisation du problème. En effet, dans la majorité des applications concrètes de programmation mathématique, la présence des variables entières est inévitable, on parle alors d'optimisation discrète ou programmation en nombres entiers.

De ce fait, la programmation mathématique multi-objectif en nombre entiers représente le cadre général de notre travail dans ce mémoire.

La nature des variables discrètes dans un contexte multi-objectif introduit des difficultés spécifiques qui sont mis en évidence au cours de ce travail.

L'ensemble efficace d'un problème multi-objectif peut être généralement très vaste

et même infini dans le cas continu. Dans certaines situations, les décideurs n'ont pas besoin de toutes les solutions efficaces d'un problème de programmation multi-objectif mais uniquement de solutions efficaces qui réalisent l'optimum d'un objectif différent d'autres objectifs déjà fixés. Ceci nous mène vers la recherche de la solution optimale d'un critère sur l'ensemble des solutions efficaces comme outil pour mesurer les préférences des décideurs et distinguer parmi les nombreuses solutions efficaces.

L'objectif de cette thèse consiste à étudier le problème d'optimisation d'un critère sur l'ensemble efficace d'un problème multi-objectif en variables entières. Ce problème est très difficile à résoudre, ceci est dû principalement à la non convexité de son ensemble réalisable et le problème appartient alors à la classe des problèmes d'optimisation globale ou non convexe. Cependant, nous nous sommes rendus compte que les publications traitant l'optimisation d'un critère linéaire sur l'ensemble des points efficace d'un problème multi-objectif linéaire en variables entières *MOILP* sont relativement très peu nombreuses. Cet état de chose, bien que source de difficulté potentielle, nous a davantage motivé pour explorer cette classe de problèmes.

La plupart des méthodes développées dans la littérature traite le cas (Linéaire- Linéaire) c'est à dire optimiser un critère linéaire sur l'ensemble efficace d'un problème multi-objectif linéaire discret *MOILP* [59], [4], [46], [15].

Très peu de travaux récents ont traité des cas plus généraux, à savoir le cas (Fractionnaire- Linéaire) qui consiste à optimiser un critère fractionnaire sur l'ensemble efficace d'un problème *MOILP* [1], le cas (Linéaire- Fractionnaire) qui optimise un critère linéaire sur un ensemble efficace d'un problème *MOILFP* [83], le cas (Fractionnaire- Fractionnaire) optimisant un critère fractionnaire linéaire sur l'ensemble efficace d'un problème *MOILFP* [84] et le cas (Linéaire- Quadratique) qui, quand à lui, optimise un critère linéaire sur l'ensemble efficace d'un problème multi-objectif quadratique convexe [6].

Ce mémoire est dédié à l'optimisation d'un critère quadratique sur l'ensemble efficace d'un problème multi-objectif discret, nous tenons à souligner qu'à notre connaissance, il n'existe aucun article traitant le problème proposé, ce qui justifie notre intérêt à l'étudier. Nous proposons pour cela le cas de l'optimisation d'un critère quadratique semi-défini sur l'ensemble efficace d'un problème *MOILP* aussi le cas de l'optimisation d'un critère quadratique indéfini sur l'ensemble efficace d'un problème *MOILP*.

Ce mémoire de thèse est organisé de la façon suivante :

- Le chapitre 1 expose une introduction générale à l'optimisation quadratique, nous

nous sommes intéressés à l'optimisation quadratique semi-définie et à l'optimisation quadratique indéfinie, nous présentons les principales approches de résolution proposées dans la littérature pour ces différents problèmes.

- Le chapitre 2 expose le cadre général de notre travail, nous présentons pour cela l'essentiel des définitions et résultats liés à l'optimisation multi-objectif et nous parlerons des différentes approches de résolution.
- Le chapitre 3 se focalise sur la programmation linéaire multi-objectif en variables entières *MOILP* et sur quelques travaux récents existant en littérature.
- Le chapitre 4 expose notre contribution qui consiste à optimiser un critère quadratique sur l'ensemble efficient d'un problème *MOILP*. Nous proposons pour cela deux approches différentes pour le cas d'un critère quadratique indéfini et une approche pour le cas d'un critère quadratique semi-défini.
- Enfin, le mémoire s'achèvera par une conclusion générale et quelques perspectives de recherche.

Chapitre 1

La Programmation Quadratique

1.1 Introduction

Les problèmes de la programmation linéaire se posent lorsque l'on cherche à optimiser (maximiser ou minimiser) une fonction linéaire de plusieurs variables, étant donnée que celles-ci doivent vérifier un certain nombre d'équations et/ou d'inéquations appelées *contraintes*. La forme exacte de ces contraintes peut différer d'un problème à un autre, cependant, tout programme linéaire peut être transformé sous la forme standard suivante :

$$(LP) \begin{cases} \text{optimiser } Z(x) = c^t x \\ sc \\ Ax = b \\ x \geq 0 \end{cases}$$

où c et x sont deux n -vecteurs réels, A est une $m \times n$ -matrice réelle et b un m -vecteur réel.

L'ensemble des contraintes $x \in S = \{x \in \mathbb{R}^n | Ax = b, x \geq 0\}$ représente l'ensemble des solutions réalisables de (LP) .

Si les variables sont astreintes à ne prendre que des valeurs entières, on parle de programme linéaire en nombres entiers (ILP), et si les variables ne peuvent prendre que les valeurs 0 ou 1, le programme est linéaire à variables bivalentes. Il existe aussi des problèmes dans lesquels une partie seulement des variables ne peut prendre que des valeurs entières ; on parle alors, de programme linéaire mixte.

Les éléments de base de la programmation linéaire ont fait l'objet d'une littérature abondante aussi riche que diversifiée. On retiendra essentiellement les ouvrages spécialisés suivants : [52], [57], [67], [71], [75] et [73].

Il existe des algorithmes polynômiaux efficaces pour résoudre un programme linéaire comme ceux dits de points intérieurs initiés par Karmarkar [47] en 1984. Néanmoins l'algorithme du simplexe introduit en 1947 par G.B.Dantzig [25] est le plus célèbre (et le plus efficace dans le cas général) des algorithmes de résolution, bien qu'il ne soit pas polynomial !

L'algorithme du simplexe repose sur le fait qu'une solution optimale d'un programme linéaire peut être prise parmi les sommets du polyèdre de \mathbb{R}^n déterminé par $Ax \leq b$.

Un programme non linéaire est beaucoup plus difficile à résoudre, et la théorie du Simplexe ne peut être utilisée. Une solution optimale d'un programme linéaire lorsqu'elle existe est un point extrême de l'ensemble des solutions réalisables (polyèdre des solutions réalisables). Par contre une solution optimale d'un problème non linéaire peut être un point intérieur ou sur la frontière de l'ensemble des solutions réalisables mais pas nécessairement un point extrême. La principale difficulté de la programmation non linéaire est que l'on peut avoir des optimaux locaux qui ne sont pas globaux.

Nous illustrons la difficulté de la recherche de la solution optimale dans le cas non linéaire par les deux exemples suivants :

Exemple 1 : Considérons l'exemple suivant dont l'illustration graphique est donnée à la figure 1.1, où l'on peut voir que la solution optimale est strictement intérieure.

$$\left\{ \begin{array}{l} \min (x_1 - 3)^2 + (x_2 - 3)^2 \\ sc \\ x_1 \leq 4 \\ 2x_2 \leq 12 \\ 3x_1 + 2x_2 \leq 18 \\ x_1, x_2 \geq 0 \end{array} \right.$$

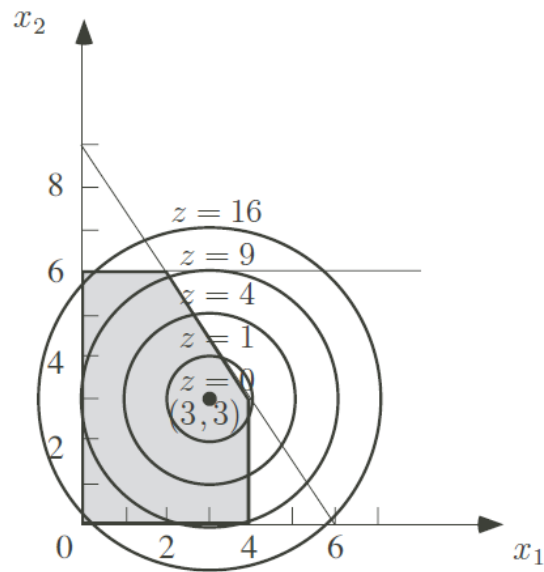


FIGURE 1.1 – Solution intérieure

Exemple 2 :

$$\begin{cases} \max & 3x_1 + 5x_2 \\ \text{sc} & \\ & x_1 \leq 4 \\ & 8x_1 - x_1^2 + 14x_2 - x_2^2 \leq 49 \\ & x_1, x_2 \geq 0 \end{cases}$$

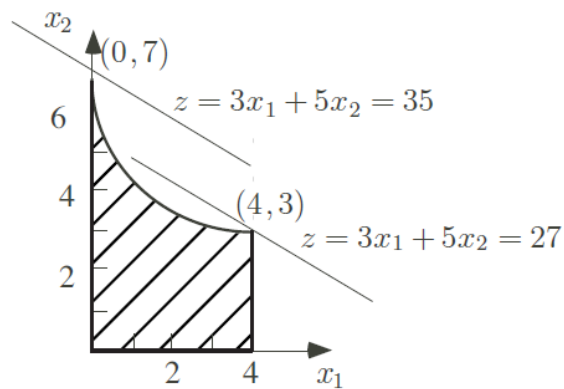


FIGURE 1.2 – Maximum local

La représentation graphique est donnée à la figure 1.2 où l'on peut voir l'existence de deux maximum locaux $(4,3)$ et $(0,7)$ mais $(0,7)$ est le maximum global.

Le problème peut également se produire si la fonction objectif que l'on optimise est non convexe. La détermination de tous les optimaux locaux est un problème très délicat en général. Cependant, il existe une classe de problèmes pour lesquels ce problème ne se produit pas dans le sens que tous les optimums locaux sont des optimum globaux. Il s'agit des problèmes convexes.

Dans ces deux cas, il apparaît alors comme évident que la recherche de la solution optimale est plus difficile que dans le cas linéaire, puisqu'elle doit se faire tout le long du polygone des contraintes, et non seulement en quelques points particuliers. De plus, si la solution cherchée est entière, la complexité est encore augmentée.

Les cas les plus fréquents de la programmation non linéaire est la programmation quadratique dont nous allons détailler dans ce qui suit, les différents outils pour résoudre ces types de programmes.

1.2 Éléments de l'algèbre linéaire

Définition 1.2.1 Soit A une matrice carrée d'ordre n , symétrique.

- A est dite définie positive, si $x^t Ax > 0, \forall x \in \mathbb{R}^n, x \neq 0$.
- A est dite définie négative, si $x^t Ax < 0, \forall x \in \mathbb{R}^n, x \neq 0$.
- A est dite semi-définie positive, si $x^t Ax \geq 0, \forall x \in \mathbb{R}^n$ et $\exists x \in \mathbb{R}^n$ tel que $x^t Ax = 0, x \neq 0$.
- A est dite semi-définie négative, si $x^t Ax \leq 0, \forall x \in \mathbb{R}^n$ et $\exists x \in \mathbb{R}^n$ tel que $x^t Ax = 0, x \neq 0$.
- A est dite indéfinie, si $x^t Ax$ peut prendre un signe ou l'autre selon la valeur de x .

Notons que $x^t Ax = \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j$ est appelé forme quadratique.

Définition 1.2.2 On appelle mineur principal d'ordre j de A , noté $\nabla_j(A)$, le déterminant de la sous-matrice de A obtenue en supprimant les $(n - j)$ dernières lignes et les $(n - j)$ dernière colonnes de A .

Les mineurs principaux de A sont : $\nabla_1(A) = a_{11}, \nabla_2(A) = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}, \dots, \nabla_n(A) = \det(A)$

Définition 1.2.3 Soit $\lambda \in \mathbb{R}$, λ est appelée valeur propre de A si $\det(A - \lambda I) = 0$.

Propriété 1.2.1 Soit A une matrice carrée d'ordre n , symétrique.

- A est définie positive si et seulement si $(-A)$ est définie négatives.

- A est définie positive si et seulement si tous les mineurs principaux de A sont strictement positifs. ie, $\nabla_j(A) > 0 \forall j = \overline{1, n}$.
- A est définie positive si et seulement si toutes les valeurs propres de A sont strictement positives.
- A est définie négative si et seulement si $\forall j = \overline{1, n} (-1)^j \nabla_j(A) > 0$.
- A est définie négative si et seulement si toutes les valeurs propres de A sont strictement négatives.
- A est semi-définie positive si et seulement si toutes les valeurs propres de A sont positives ou nulles et il existe une valeur propre λ telle que $\lambda = 0$.
- A est semi-définie négative si et seulement si toutes les valeurs propres de A sont négatives ou nulles et il existe une valeur propre λ telle que $\lambda = 0$.

1.3 Notion sur l'optimisation convexe

L'importance de la convexité en optimisation peut se résumer en la phrase de Rockafellar [64] : "La vraie ligne de démarcation en optimisation n'est pas entre linéaire et non linéaire mais entre convexe et non convexe".

Définition 1.3.1 (Ensemble convexe) On dit que l'ensemble S est convexe si et seulement si :

$$\forall x, y \in S \text{ et } \forall \lambda \in [0, 1], \lambda x + (1 - \lambda)y \in S$$

D'un point de vue géométrique, un convexe est donc un ensemble qui, lorsqu'il contient deux points, contient nécessairement le segment les reliant.

Définition 1.3.2 (Combinaison linéaire convexe) Un vecteur $y \in S$ est une combinaison linéaire convexe des points $\{x_1, \dots, x_p\}$ s'il existe des coefficients réels λ_i ; $i \in \{1, \dots, p\}$, tels que :

$$y = \sum_{i=1}^p \lambda_i x_i, \text{ avec } \lambda_i \geq 0, \forall i \in \{1, \dots, p\} \text{ et } \sum_{i=1}^p \lambda_i = 1.$$

Définition 1.3.3 (Enveloppe convexe) L'enveloppe convexe d'un ensemble $S \subset \mathbb{R}^n$; est l'ensemble des points de \mathbb{R}^n qui s'écrivent comme combinaisons convexes des points de S . Elle est notée par :

$$\text{conv}(S) = \left\{ x \in \mathbb{R}^n / x = \sum_{i=1}^p \lambda_i x_i, x_i \in S, \lambda_i \geq 0 \text{ } i \in \{1, \dots, p\} \text{ et } \sum_{i=1}^p \lambda_i = 1 \right\}$$

C'est le plus petit ensemble convexe contenant S .

- L'ensemble $\{x \in \mathbb{R}^n / a'x = b\}$ représente un hyperplan de \mathbb{R}^n ;
- L'ensemble $\{x \in \mathbb{R}^n / a'x \leq b\}$ représente un demi-espace fermé de \mathbb{R}^n dont l'hyperplan correspondant constitue la frontière.

Définition 1.3.4 (Polyèdre) – *Un polyèdre S est l'intersection d'un nombre fini de demi-espaces fermés et/ou d'hyperplans.*

- *Un polyèdre est un ensemble convexe fermé.*
- *Un polyèdre S est borné s'il existe une valeur β finie et positive telle que :*

$$|x_j| \leq \beta \quad \forall j \in \{1, \dots, n\} \text{ et } \forall x \in S.$$

- *Un polytope est un polyèdre borné et non vide.*

Définition 1.3.5 (Point extrême) *Soit S un convexe non vide de \mathbb{R}^n . x est dit point extrême ou sommet de S si :*

$$x = \lambda x_1 + (1 - \lambda)x_2 \quad \forall x_1, x_2 \in S \text{ et } \lambda \in]0, 1[\text{ alors } x = x_1 = x_2.$$

Définition 1.3.6 (Fonction convexe) *La fonction $f : S \rightarrow \mathbb{R}$ est dite convexe sur l'ensemble S non vide de \mathbb{R}^n si et seulement si :*

$$\forall x, y \in S \text{ et } \forall \lambda \in [0, 1] \text{ on a } f(x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$$

Définition 1.3.7 (Fonction concave) *La fonction $f : S \rightarrow \mathbb{R}$ est dite concave sur l'ensemble S non vide de \mathbb{R}^n si et seulement si :*

$$\forall x, y \in S \text{ et } \forall \lambda \in [0, 1] \text{ on a } f(\lambda x + (1 - \lambda)y) \geq \lambda f(x) + (1 - \lambda)f(y)$$

- Les cas stricts correspondent aux mêmes définitions avec des inégalités strictes pour $0 < \lambda < 1$ et $x \neq y$.
- La fonction f est concave (resp. strictement concave) $\Leftrightarrow (-f)$ est convexe (resp. strictement convexe).

Définition 1.3.8 (Épigraphe) *On appelle épigraphe de f noté $\text{épi}(f)$ le sous ensemble de \mathbb{R}^{n+1} défini par :*

$$\text{épi}(f) = \{(x, y) \in S \times \mathbb{R} / y \geq f(x)\}$$

Définition 1.3.9 (Hypographe) On appelle hypographe de f noté $\text{hyp}(f)$ le sous ensemble de \mathbb{R}^{n+1} défini par :

$$\text{hyp}(f) = \{(x, y) \in S \times \mathbb{R} / y \leq f(x)\}$$

Théorème 1.3.1 1. f est convexe si et seulement si $\text{épi}(f)$ est convexe.
2. f est concave si et seulement si $\text{hyp}(f)$ est convexe.

Lemme 1.3.1 Si f est une fonction convexe alors l'ensemble niveau $L(f, \alpha) = \{x \in S : f(x) \leq \alpha\}$ est convexe pour tout $\alpha \in \mathbb{R}$.

Définition 1.3.10 Soit $S \subseteq \mathbb{R}^n$, $S \neq \emptyset$. L'ensemble $V_\epsilon(x) = \{x \in S : \|x - \bar{x}\| < \epsilon\}$ est appelé voisinage de $\bar{x} \in S$, où $\epsilon > 0$ et $\|\cdot\|$ désigne la norme euclidienne de \mathbb{R}^n .

Définition 1.3.11 On dit que \bar{x} appartient à l'intérieur de S , noté $\text{int}(S)$, s'il existe $\epsilon > 0$ tel que $V_\epsilon(\bar{x}) \subset S$.

Théorème 1.3.2 Soit $f : S \subset \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction convexe sur S ensemble convexe non vide. Alors f est continue sur l'intérieure de S .

1.3.1 Fonctions convexes généralisées

Les fonctions convexes généralisées jouent un rôle très important dans la théorie de l'optimisation. Soit une fonction $f : S \subset \mathbb{R}^n \rightarrow \mathbb{R}$ avec S un convexe non vide de \mathbb{R}^n .

Définition 1.3.12 (Fonction quasiconvexe) La fonction f est dite quasiconvexe sur S si et seulement si :

$$\forall x, y \in S \text{ et } \forall \lambda \in [0, 1] \text{ on a } f(\lambda x + (1 - \lambda)y) \leq \max\{f(x), f(y)\}$$

f est quasiconvexe sur S est ainsi équivalente à :

$$\forall x, y \in S \text{ et } \forall \lambda \in [0, 1], f(x) \leq f(y) \Rightarrow f(\lambda x + (1 - \lambda)y) \leq f(y)$$

Définition 1.3.13 (Fonction quasiconcave) La fonction f est dite quasiconcave sur S si et seulement si :

$$\forall x, y \in S \text{ et } \forall \lambda \in [0, 1] \text{ on a } f(\lambda x + (1 - \lambda)y) \geq \min\{f(x), f(y)\}$$

La fonction f est quasiconcave $\Leftrightarrow (-f)$ est quasiconvexe.

Une fonction qui est à la fois quasiconvexe et quasiconcave est dite *quasimonotone* ou *quasilinéaire*.

Les fonctions quasiconvexes, quasiconcaves et quasilinéaires sont caractérisées par la convexité de leurs ensembles niveaux.

Remarque 1.3.1 Une fonction convexe sur S convexe est quasiconvexe.

Définition 1.3.14 (Fonction strictement quasiconvexe) La fonction f est strictement quasiconvexe sur S si :

$$\forall x, y \in S \text{ avec } f(x) \neq f(y) \forall \lambda \in]0, 1[\text{ on a } f(\lambda x + (1 - \lambda)y) < \max\{f(x), f(y)\}$$

f strictement quasiconcave si $(-f)$ est strictement quasiconvexe.

Définition 1.3.15 (Fonction différentiable) Soit S un ensemble non vide de \mathbb{R}^n et $f : S \rightarrow \mathbb{R}$. f est dite différentiable en $\bar{x} \in S$ s'il existe une application linéaire $L : \mathbb{R}^n \rightarrow \mathbb{R}$ et une fonction $\alpha : \mathbb{R}^n \rightarrow \mathbb{R}$ tels que :

$$f(x) = f(\bar{x}) + L(x - \bar{x}) + \|x - \bar{x}\| \alpha(\bar{x}; x - \bar{x}); \forall x \in S$$

Dans ce cas L est spécifiée par la matrice C d'ordre $1 \times n$ telle que $C = \nabla f(\bar{x})$ où $\lim_{x \rightarrow \bar{x}} \alpha(\bar{x}; x - \bar{x}) = 0$ et $\nabla f(\bar{x}) = \left(\frac{\delta f(\bar{x})}{\delta x_1}, \dots, \frac{\delta f(\bar{x})}{\delta x_n} \right)^t$

Définition 1.3.16 (Fonction pseudoconvexe) Soit f une fonction différentiable sur S un ouvert non vide de \mathbb{R}^n . f est pseudoconvexe sur S si :

$$\forall x, y \in S, (y - x)' \nabla f(x) \geq 0 \Rightarrow f(y) \geq f(x)$$

Définition 1.3.17 (Fonction pseudoconcave) Soit f une fonction différentiable sur S un ouvert non vide de \mathbb{R}^n . f est pseudoconcave sur S si :

$$\forall x, y \in S, (y - x)' \nabla f(x) \leq 0 \Rightarrow f(y) \leq f(x)$$

Théorème 1.3.3 Soit $f : S \rightarrow \mathbb{R}$ différentiable et pseudoconvexe sur S un convexe ouvert non vide de \mathbb{R}^n . Alors, f est à la fois quasiconvexe et strictement quasiconvexe.

Proposition 1.3.1 1. Si f est convexe alors f est quasiconvexe. Si de plus f est différentiable alors elle est pseudoconvexe.

2. Si f est concave alors f est quasiconcave. Si de plus f est différentiable alors elle est pseudoconcave.

1.3.2 Condition d'optimalité en optimisation non linéaire

Considérons le problème de minimisation (P_m) suivant :

$$(P_m) \begin{cases} \min f(x) \\ sc \\ x \in S = \{x \in \mathbb{R}^n | Ax \leq b, x \geq 0\} \end{cases}$$

Définition 1.3.18 (Minimum global) Une solution admissible \bar{x} est dite minimum global si et seulement si $f(\bar{x}) \leq f(x)$, $\forall x \in S$.

Définition 1.3.19 (Minimum local) Une solution admissible \bar{x} est dite minimum local si et seulement si il existe un voisinage $V_\epsilon(x)$ de x tel que $f(\bar{x}) \leq f(x)$, $\forall x \in S \cap V_\epsilon(x)$.

Tout minimum global est évidemment un minimum local.

En général, on est beaucoup plus intéressé par la recherche de minima globaux (car ils sont les seuls à garantir que la valeur de leur fonction objectif ne peut être améliorée), mais ceux-ci sont malheureusement également beaucoup plus difficiles à calculer (intuitivement, la raison en est qu'il suffit pour prouver qu'un minimum est local de vérifier qu'il n'existe pas de meilleure solution dans un voisinage restreint autour de ce minimum, tandis que prouver qu'un minimum est global requiert l'analyse de la fonction objectif sur l'entièreté du domaine admissible). Cependant, à l'aide de la notion de convexité, de quasiconvexité et de pseudoconvexité, nous allons décrire une classe de problèmes pour laquelle la situation est bien plus favorable.

Théorème 1.3.4 (Conditions nécessaire d'optimalité) Soit $f : S \subset \mathbb{R}^n \rightarrow \mathbb{R}$ est une fonction convexe sur S un ensemble convexe non vide. Si x est un minimum local alors il est un minimum global pour le problème (P_m) .

Théorème 1.3.5 Si $f : S \subset \mathbb{R}^n \rightarrow \mathbb{R}$ est une fonction convexe et différentiable sur S convexe non vide et $\bar{x} \in S$ tel que $\nabla f(x) = 0$, alors x est un minimum global de f sur S .

Théorème 1.3.6 Soit $f : S \rightarrow \mathbb{R}$ une fonction quasiconcave (ou quasiconvexe) et continue sur S polyèdre borné, alors au moins une solution optimale \bar{x} , est un point extrême de S .

Les fonctions strictement quasiconvexes et strictement quasiconcaves, sont particulièrement importantes dans la programmation non linéaire, car elles assurent qu'un minimum local et un maximum local sur un ensemble convexe soit, respectivement, un minimum global et un maximum global.

Théorème 1.3.7 Soit $f : S \rightarrow \mathbb{R}$ strictement quasiconvexe sur S et S un convexe non vide de \mathbb{R}^n . Si x est un minimum local alors x est un minimum global.

Les fonctions pseudoconvexes partagent, la propriété importante du théorème 1.3.5, avec les fonctions convexes.

Théorème 1.3.8 Si $f : S \subset \mathbb{R}^n \rightarrow \mathbb{R}$ est une fonction pseudoconvexe et différentiable sur S convexe non vide et $\bar{x} \in S$ tel que $\nabla f(x) = 0$, alors x est un minimum global de f sur S .

Conditions de Kuhn-Tucker

Considérons le problème de minimisation (P) :

$$(P) \begin{cases} \min f(x) \\ sc \\ x \in S \end{cases}$$

où S est l'ensemble de solutions réalisables donné par :

$$S = \{x \in \mathbb{R}^n \mid g_i(x) \leq 0, i = 1, \dots, m; x \geq 0\}$$

On suppose que f et g_i ; $i = 1, \dots, m$, sont différentiable sur un domaine de \mathbb{R}^n .

Définition 1.3.20 On appelle Lagrangien associé au problème (P) la fonction $L : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ définie par :

$$L(x, \lambda) = f(x) + \sum_{i=1}^m \lambda_i g_i(x)$$

Les λ_i pour $i = 1, \dots, m$ sont appelés multiplicateurs de Lagrange.

Définition 1.3.21 – Les contraintes $g_i(x) = 0$ sont dites actives (ou saturées) en x .

– Les contraintes $g_i(x) < 0$ sont dites inactives (ou insaturées) en x .

Définition 1.3.22 Soit $z \in \mathbb{R}^n$ $z \neq 0$ et soit $x \in S$.

On dit que z est une direction admissible en x s'il existe $\bar{\alpha} > 0$ tel que $x + \alpha z$ soit une solution réalisable pour tout $\alpha \in [0, \bar{\alpha}]$.

Remarque 1.3.2 Si x est un minimum local du problème (P) et z une direction admissible alors $f(x + \alpha z) \geq f(x)$ pour $\alpha > 0$ suffisamment petit.

On définit les ensembles suivants :

$I(x) = \{i \in \{1, \dots, m\} | g_i(x) = 0\}$: ensemble des indices des contraintes actives en x .

$Z(x) = \{z \in \mathbb{R}^n | \exists \bar{\alpha} > 0, (x + \alpha z) \in S, 0 < \alpha < \bar{\alpha}\}$: ensembles des directions admissibles en x .

$\overline{Z(x)}$: fermeture de $Z(x)$.

$Y(x) = \{z \in \mathbb{R}^n | z \neq 0, z^t \nabla g_i(x) \leq 0, \forall i \in I(x)\}$.

On a alors :

Lemme 1.3.2 *Soit $x \in S$, on a $\overline{Z(x)} \subset Y(x)$.*

Définition 1.3.23 *On dit que S satisfait l'hypothèse de qualification des contraintes (QC) en $x \in S$ si $\overline{Z(x)} = Y(x)$*

La vérification de l'hypothèse de qualification des contraintes en x , signifie que les conditions $z^t \nabla g_i(x) \leq 0 \forall i \in I(x)$ sont nécessaires et suffisantes pour que z soit une direction admissible en x .

La vérification directe de l'hypothèse de qualification des contraintes (QC) peut être très difficile en pratique, c'est pourquoi on a recherché des conditions suffisantes pour que (QC) soit réalisée. Les résultats les plus importants sont rassemblés dans le lemme suivant :

Lemme 1.3.3 1. *Pour que l'hypothèse de (QC) soit vérifiée en un point x_0 de S il suffit que $\nabla g_i(x_0)$ pour $i \in I(x_0)$ soient linéairement indépendants*

2. *Pour que l'hypothèse de (QC) soit vérifiée en tout point $x \in S$, il suffit que l'une des deux conditions suivantes soit vérifiée :*

(a) *toutes les fonctions g_i sont linéaires,*

(b) *toutes les fonctions g_i sont pseudoconvexes et il existe $\bar{x} \in S$ tel que $g_i(\bar{x}) < 0; \forall i = 1, \dots, m$ (cette condition est connue sous le nom de condition de Slater)*

Le théorème de Kuhn-Tucker s'énonce alors comme suit :

Théorème 1.3.9 *Soit $x^* \in S$ et supposons que l'hypothèse de qualification des contraintes soit vérifiée en x^* , ie $\overline{Z(x^*)} = Y(x^*)$.*

– *Si x^* est un optimum local pour le problème P_m , alors il existe $\lambda \in \mathbb{R}^m$ vérifiant le système de Kuhn-Tucker suivant :*

$$\begin{cases} \nabla f(x^*) + \sum_{i=1}^m \lambda_i \nabla g_i(x^*) = 0 \\ \lambda_i g_i(x^*) = 0 \quad i = 1, \dots, m \\ \lambda_i \geq 0, \quad i = 1, \dots, m \end{cases} \quad (\text{KT})$$

- Si f est pseudoconvexe, g_i quasiconvexe pour $i = 1, \dots, m$ une condition nécessaire et suffisante pour que x^* soit une solution optimale pour le problème (P) est qu'il existe $\lambda \in \mathbb{R}^m$ vérifiant les conditions du système KT.

1.4 La Programmation Quadratique

Les programmes quadratiques sont les programmes non linéaires les plus proches de la programmation linéaires, ils représentent une transition naturelle de la théorie de la programmation linéaire vers la théorie de la programmation non linéaire, cependant, il existent des différences importantes entre les solutions des problèmes de programmation linéaire et celles des problèmes de programmation quadratique. Une solution optimale d'un programme linéaire lorsqu'elle existe est un point extrême de l'ensemble des solutions réalisables (polyèdre des solutions réalisables). Par contre une solution optimale d'un problème quadratique peut être un point intérieur ou sur la frontière de l'ensemble des solutions réalisables mais pas nécessairement un point extrême.

La programmation quadratique est une classe importante de programmation mathématique, de part sa vague utilisation dans différents domaines comme les probabilités [85], la production efficace [28], la sélection de portefeuille [53] ...

1.4.1 Formulation du problème

Le problème de la *programmation quadratique* (QP) consiste à déterminer un élément x^* optimisant la fonction $\frac{1}{2}x^t G x + g^t x$ sur un domaine réalisable S défini par le système de contraintes linéaires. Il a donc la forme suivante :

$$(QP) \begin{cases} \text{optimiser } \frac{1}{2}x^t G x + g^t x \\ sc \\ S = \{x \in \mathbb{R}^n / Ax \leq 0, x \geq 0\} \end{cases}$$

où G est une matrice réelle symétrique d'ordre $n \times n$, g et x deux vecteurs de \mathbb{R}^n , A une matrice réelle d'ordre $m \times n$ et b un vecteur de \mathbb{R}^m . L'ensemble S est supposé borné non vide.

Lorsque la matrice G possède certaines propriétés, la fonction f peut prendre un nom particulier. En effet, lorsque G est semi-définie (positive ou négative) le problème

est appelé *programme quadratique semi-défini*, sinon le problème est dit *programme quadratique indéfini*.

Si de plus, les variables sont astreintes à ne prendre que des valeurs entières, le programme (QP) est appelé programme quadratique en nombres entiers noté (IQP) .

1.4.2 Programmation quadratique semi-définie

Un problème de programmation quadratique semi-définie est un programme qui peut se mettre sous la forme suivante :

$$(QP_D) \begin{cases} \min(\max) f(x) = \frac{1}{2}x^t Gx + g^t x \\ sc \\ S = \{x \in \mathbb{R}^n / Ax \leq b, x \geq 0\} \end{cases}$$

où G est une matrice réelle symétrique d'ordre $n \times n$ qui a la particularité d'être semi-définie positive (resp négative), g et x deux vecteurs de \mathbb{R}^n , A une matrice réelle d'ordre $m \times n$ et b un vecteur de \mathbb{R}^m . L'ensemble S est supposé borné non vide.

Théorème 1.4.1 *Une forme quadratique semi-définie positive (resp semi-définie négative) est une fonction convexe (resp concave).*

Comme la fonction $f(x)$ est convexe (resp concave), les contraintes sont linéaire alors, D'après le théorème 1.4.1, les sections 1.2 et la sous section 1.3.2, des conséquences pour un problème de programmation quadratique (PQ_D) peuvent être cités :

Conséquences

- Un minimum (resp maximum) local du problème (QP_D) est un minimum (resp maximum) global.
- Une condition nécessaire et suffisante pour que x soit une solution optimale du problème (QP_D) est qu'il existe $\lambda \geq 0$ tel que (x, λ) vérifie les conditions du système KT.

1.4.3 Résolution d'un programme quadratique semi-défini

Méthode d'activation de contraintes (ASM)

La méthode d'activation des contraintes (Active-set method, ASM) est une méthode classique, développée au début des années soixante-dix pour la résolution des problèmes

de programmation linéaire et quadratique. Elle s'applique pour des problèmes d'optimisation avec des contraintes linéaires de type inégalités ou mixtes (égalités et inégalités).

Le principe général de la méthode consiste à écarter temporairement un certain nombre de contraintes d'inégalités et de résoudre à chaque itération un problème avec uniquement des contraintes d'égalité, correspondant aux contraintes actives. Par la suite, l'ensemble des indices actifs est ajusté en ajoutant ou/et en supprimant une contrainte à la fois jusqu'à l'obtention de la solution optimale.

La méthode ASM pour le cas linéaire est facile à appliquer par rapport au cas quadratique puisque cela dépend du nombre de contraintes actives à l'optimum. D'après la théorie de la programmation linéaire, on sait à l'avance que la solution optimale correspond à un sommet du polyèdre du domaine admissible, contrairement à un problème quadratique où la solution peut être un sommet, une face ou un point de l'intérieur du polyèdre.

Méthode du point intérieur

Les méthodes de points intérieurs forment une classe d'algorithmes qui permettent de résoudre des problèmes d'optimisation convexe (linéaires ou non). Les méthodes de points intérieurs se répartissent en plusieurs familles :

- La méthode "affine scaling" (optimisation sur des ellipsoïdes).
- La méthode de réduction du potentiel (notion de barrière, chemin central, relaxation).

La méthode du chemin central est le représentant le plus important de cette famille.

Toutes ces méthodes trouvent leur origine (historique) dans les travaux d'un élève du mathématicien soviétique Andreï Kolmogorov : Dikin. C'est, en effet, à Dikin que l'on doit la méthode des ellipsoïdes sur laquelle se fondent plus ou moins toutes les méthodes de points intérieurs. Arkadi Nemirovski, David B. Yudin, Shor développent, en 1972, la méthode des ellipsoïdes pour des problèmes d'optimisation (non linéaires) convexe. En 1979, Leonid Khachiyan démontre que la méthode des ellipsoïdes, appliquée à la *PL* a une complexité, dans le pire des cas, polynomiale. Cependant, l'algorithme qu'il propose est beaucoup plus lent que le simplexe. Cette approche, qui s'étend de façon très élégante à des problèmes non linéaires convexes peut être considéré comme l'idée germinale des méthodes de points intérieurs développées par la suite. Les algorithmes développés par la suite ont été inspirés par l'Algorithme de Karmarkar, développé en 1984 par Narendra Karmarkar [47] pour l'optimisation linéaire. L'idée de base de la méthode est d'utiliser des fonctions barrières pour décrire l'ensemble des solutions qui est convexe par définition du problème. A l'opposé de l'algorithme du simplexe, cette méthode atteint l'optimum

du problème en passant par l'intérieur de l'ensemble des solutions réalisables. Toutefois, on trouve quelques unes des plus importantes contributions dans les travaux de Nesterov et Nemirovski [58], qui stipulent que la méthode peut être appliquée sur d'autres classes de problèmes comme les problèmes quadratiques convexes.

Méthode du simplexe quadratique de Wolfe (1959)

Durant ces dernières années beaucoup d'algorithmes ont été conçus pour résoudre des problèmes quadratiques convexe, la méthode classique pour la résolution de ce type de problème est celle de Wolfe [38],[79], [80] qui est une modification légère de la méthode du simplexe [23], [24], [38].

Son principe consiste à résoudre le système d'optimalité de "Karush-Kuhn-Tucker" KT , en ajoutant la condition de complémentarité. l'algorithme de la méthode nécessite une solution réalisable de départ. Elle est obtenue en utilisant la première phase du simplexe.

Considérons le problème de programmation quadratique sous la forme standard suivante :

$$\begin{cases} \min \frac{1}{2}x^t Gx + g^t x \\ sc \\ S = \{x \in \mathbb{R}^n / Ax = b, x \geq 0\} \end{cases} \quad (\text{QP}')$$

où G est une matrice réelle semi-définie positive d'ordre $n \times n$, g et x deux vecteurs de \mathbb{R}^n , $A \in \mathbb{R}^{m \times n}$ et $b \in \mathbb{R}^m$. L'ensemble S est supposé borné non vide.

Le Lagrangien associé au problème QP' précédent est :

$$L(x, \lambda, \gamma) = \frac{1}{2}x^t Gx + g^t x + \lambda^t (Ax - b) - \gamma^t x$$

Le m-vecteur λ représente le multiplicateur de Lagrange associé à la contrainte d'égalité, et le n-vecteur γ représente le multiplicateur de KT associé à la contrainte d'inégalité du problème QP'.

Comme le problème QP' est convexe alors les contraintes d'optimalité de KT sont à la fois nécessaires et suffisantes, formulées de cette manière :

x^* est une solution optimale du problème QP' si et seulement si il existe deux vecteurs

$\lambda^* \in \mathbb{R}^m$ et $\gamma^* \geq 0$, appartenant à \mathbb{R}^n vérifiant les relations suivantes :

$$\left\{ \begin{array}{ll} \frac{\partial L}{\partial x^*} = Gx + g + A^t \lambda^* - \gamma^* = 0 & L_1 \\ \frac{\partial L}{\partial \lambda^*} = Ax^* - b = 0, x^* \geq 0 & L_2 \\ \gamma^{*t} x^* = 0 & L_3 \\ \lambda^* \in \mathbb{R}^m, \gamma^* \geq 0 & L_4 \end{array} \right. \quad (\text{KT})$$

- la relation L_1 est appelée condition de stationnarité,
- la relation L_2 est la faisabilité de la solution,
- la relation L_3 est la condition de complémentarité,
- la relation L_4 est la non négativité des multiplicateurs deKT.

Le système KT contient $(n+m)$ équations linéaires à $(2n+m)$ inconnues, définies par L_1 et L_2 , et n équations non linéaires définies par L_3 :

$$\gamma_j^{*t} x_j = 0 \quad \forall j = \overline{1, n}. \quad (\text{L3})$$

Pour trouver une solution au système tel que L3 soit vérifiée, il suffit d'obtenir une solution réalisable de base du système linéaire $\{(L_1), (L_2)\}$, tout en s'assurant que x_j^* soit basique et γ_j^* est non basique, ou vice versa.

Pour appliquer la méthode du simplexe, il faut alors écrire le système $\{(L_1), (L_2)\}$ sous forme standard, ce qui veut dire que le second membre doit être positif ou nul ainsi que le vecteur λ^* , on peut dans ce cas écrire le vecteur λ^* sous la forme suivante :

$$\lambda_i^* = \alpha_i^* - \alpha_{m+i}^*, \quad \alpha_{m+i}^* \geq 0, \quad \alpha_i^* \geq 0, \quad i = 1, \dots, m.$$

1.4.4 Programmation quadratique indéfinie

Un problème de programmation quadratique indéfini est un programme dont la fonction objectif quadratique n'est ni convexe ni concave, c'est à dire que la matrice G n'est plus définie positive ou définie négative. Dans ce cas le problème (QP) devient plus difficile à résoudre puisque l'optimum local ne sera pas forcément un optimum global.

Nous nous intéresserons dans cette thèse à l'optimisation d'un critère quadratique indéfini qui peut être écrit sous forme d'un produit de deux fonctions linéaires positives. Une application pour ce type de programme peut être trouvée dans la théorie microéconomique [20], [41]; étant donné un consommateur rationnel souhaitant maximiser son utilité de consommer deux produits soumis à une contrainte budgétaire, la fonction d'utilité étant le produit des quantités des deux produits consommés, il est clair que le

consommation est limitée par les niveaux de production des produits et par conséquent il est utile d'inclure des contraintes linéaire supplémentaires relatives aux ressources et aux capacités.

Ainsi, le problème (QP) peut être écrit sous la forme suivante :

$$(QP_I) \begin{cases} \max f(x) = (p^t x + \alpha)(q^t x + \beta) \\ sc \\ S = \{x \in \mathbb{R}^n / Ax \leq b, x \geq 0\} \end{cases}$$

On suppose que les facteurs $(p^t x + \alpha)$ et $(q^t x + \beta) > 0 \forall x \in S$, $p, q \in \mathbb{R}^n$ et $\alpha, \beta \in \mathbb{R}$. L'ensemble S est supposé borné non vide.

Théorème 1.4.2 *La fonction $\phi(x) = (p^t x + \alpha)(q^t x + \beta)$ est strictement quasiconcave sur S convexe.*

Conséquences

Les programmes quadratique quasiconcave partagent quelques propriétés importantes avec les programmes de programmation linéaire.

Comme la fonction $f(x)$ est quasiconcave, les contraintes sont linéaire alors, D'après les théorèmes 1.3.6, 1.3.7, des conséquences pour un problème de programmation quadratique PQ_I peuvent être cités :

- Puisque la fonction f est strictement quasiconcave sur S alors d'après le théorème 1.3.7 maximum local du problème (QP_I) est un maximum global.
- Puisque f est quasiconcave et continue sur S alors, d'après le théorème 1.3.6, une solution optimale du problème (QP_I) est atteinte en un point extrême de S .

1.4.5 Résolution d'un programme quadratique indéfini

De nombreuses méthodes exactes ont été proposées pour la résolution des problèmes modélisés sous forme de (QP_I) , nous pouvons citer en particulier les méthodes proposés par K.swarup[74], S.D. Sharma[70], C.R. Bector et M. Dahl [7] et P. Anand [5].

L'extension de l'algorithme du simplexe au problème QP_I est due à S.D. Sharma [70] et K. Swarup [74].

La méthode Simplexe [25] fournit un algorithme systématique qui consiste à passer d'une solution réalisable de base correspondant à un sommet du polyèdre S vers une

autre solution de base de telle sorte que la fonction objectif soit meilleure. Cette procédure est alors répétée jusqu'à obtention d'une solution qui soit optimale pour notre problème. Si la fonction objectif est améliorée à chaque itération, donc on ne peut jamais passer par un même sommet deux fois. Étant donné que le nombre de sommets est fini, le procédé atteint le point optimale en un nombre fini d'étapes.

On considère le problème (QP_I) suivant :

$$(QP_I) \begin{cases} \max & f(x) = (p^t x + \alpha)(q^t x + \beta) = f_1(x).f_2(x) \\ & sc \\ & S = \{x \in \mathbb{R}^n / Ax \leq b, x \geq 0\} \end{cases}$$

Où $f_1(x) = (p^t x + \alpha) > 0$ et $f_2(x) = (q^t x + \beta) > 0 \forall x \in S$, $p, q \in \mathbb{R}^n$ et $\alpha, \beta \in \mathbb{R}$, L'ensemble S est supposé borné non vide.

La méthode proposée pour résoudre le problème (QP_I) est celle de K. Swarup est une méthode exactement semblable à la technique du simplexe utilisé pour la programmation linéaire. Les différentes itérations de la méthode sont réalisées dans un tableau du simplexe augmenté qui comprend $m+3$ lignes. Les m première lignes correspondent aux contraintes d'origine, les lignes $m+1$ et $m+2$ correspondent aux deux facteurs $f_1(x)$, $f_2(x)$ de la fonction objectif quadratique du problème (QP_I) et la ligne $m+3$ correspond au vecteur de croissance de la fonction $\phi(x)$, noté γ_j , calculés pour les variables hors base par la formule suivante :

$$\bar{\gamma}_j = f_1 \bar{\gamma}_j^2 + f_2 \bar{\gamma}_j^1 + \bar{\mu}_j \bar{\gamma}_j^1 \bar{\gamma}_j^2. \quad (1.1)$$

Où

B : représente l'ensemble des indices de base.

N : l'ensemble des indices hors base.

x^* : est la solution de base du problème (QP_I) .

et $\bar{\gamma}_j^1 = \bar{p}_j - \bar{p}_B x_j, j \in N$;

$\bar{\gamma}_j^2 = \bar{q}_j - \bar{q}_B x_j, j \in N$;

$f_1 = p^t x^* + \alpha, i \in B$;

$f_2 = q^t x^* + \beta, i \in B$;

$f = f_1.f_2$ et $\bar{\mu}_j = \min \left\{ \frac{(x^*)_i}{\bar{a}_{ij}} \mid \bar{a}_{ij} > 0 \right\}, j \in N_l$.

sont les valeurs mises à jours.

À chaque itération de l'algorithme, les $m+2$ lignes sont modifiées à travers les opérations ordinaires de pivot, par contre la dernière ligne est modifiée selon la formule du vecteur de la direction de croissance de la fonction $\phi(x)$. Une fois les valeurs de la direction de croissance de la fonction $\phi(x)$ calculées pour les variables hors base, on teste la condition d'optimalité :

- Si $\gamma_j \leq 0, \forall j \in N$, alors la solution courante est une solution optimale pour le problème (QP_I) .
- Sinon, déterminer la variable $x_s, s \in N$, qui rentre en base et la variable $x_r, r \in B$ qui sort de la base selon les critères suivant :

Critère 1 : (Choix de la variable entrante)

On détermine la variable $x_s, s \in N$, en sélectionnant $\gamma_s = \max_{j \in N} \gamma_j$.

Critère 2 : (Choix de la variable sortante)

Choisir la variable $x_r, r \in B$, pour laquelle $\frac{x_{Br}}{\bar{a}_{rs}} = \min_{i=1, \dots, m} \left\{ \frac{(x^*)_i}{\bar{a}_{is}} \mid \bar{a}_{is} > 0 \right\}$

1.5 Conclusion

Dans ce chapitre, nous nous sommes intéressés aux problèmes de l'optimisation non linéaire à savoir l'optimisation quadratique, nous avons parcouru toute la richesse et la variété des travaux consacrés aux programmes quadratique. Ce petit tour d'horizon nous a incité à étudier et détailler d'avantage ce type de problèmes que nous utiliserons à la suite de cette thèse dans le cadre d'une étude et de résolution d'un problème de l'optimisation d'un critère quadratique sur l'ensemble efficient d'un problème multi-objectif linéaire discret.

Chapitre 2

Généralité sur l'Optimisation Multi-objectif

Dans ce chapitre, nous présentons l'optimisation mathématique multi-objectif qui sera le cadre de travail de cette thèse. Nous introduisons les concepts fondamentaux tels que la dominance et la surface de compromis. Nous décrivons aussi les principales approches de résolution pour ces problèmes.

2.1 Notions fondamentales

La plupart des problèmes d'optimisation réels sont décrits à l'aide de plusieurs objectifs ou critères souvent contradictoires ou conflictuels devant être optimisés simultanément. Il n'existe généralement pas de solution qui optimise tous les critères en même temps, le concept de solution optimale devient alors plus difficile à définir. En effet, en considérant deux critères contradictoires "a" et "b", améliorer "a" détériore forcément "b" et inversement, il faut donc trouver un compromis. Dans ce cas, la solution optimale cherchée n'est plus un point unique, mais un ensemble de compromis. Résoudre un problème comprenant plusieurs critères, consiste donc à calculer le meilleur ensemble de solutions compromis : *le front Pareto*.

Les premiers travaux dans ce domaine sont dus à Koopmans [50] en 1951 qui donna une condition nécessaire et suffisante d'efficacité d'une solution suivis par les travaux de Kuhn et Tucker [51] la même année qui formulèrent un problème de maximisation vectorielle. Depuis, ce domaine a connu un développement fulgurant, traitant aussi bien du domaine de la programmation multi-objectif linéaire [82], [73] et non-linéaire [55], [56], [33], [19] que des problèmes multi-objectifs booléen ou en nombres entiers [76].

2.1.1 Position du problème

On définit un problème multi-objectif comme un problème de décision qui consiste à optimiser (maximiser ou minimiser) simultanément r fonctions réelles notées f_k , $k = 1, \dots, r$, appelées critères souvent contradictoires, sur un ensemble de solutions S .

Ce problème peut être formulé mathématiquement comme :

$$(M_{OP}) \begin{cases} \text{"opt"}[f_1(x), f_2(x), \dots, f_r(x)] \\ x \in S \end{cases}$$

Le symbole " " signifie qu'il n'est généralement pas possible de trouver dans S une solution qui optimise simultanément les r critères. Il est remarquable que de cette façon un problème multi-objectif est correctement formulé par rapport à la réalité concernée par le problème de décision. Il faut donc déterminer une solution $x^* \in S$ telle que, en regard des solutions de S , le vecteur $[f_1(x), f_2(x), \dots, f_r(x)]$ est bon, acceptable selon les préférences du décideur (DM : Décision Maker), voir optimal, à condition de se doter d'un cadre décisionnel donnant une signification à cette notion.

Dans tout ce qui suit, on considère que toutes les fonctions objectifs sont à "Maximiser".

- Chaque solution est caractérisée par un vecteur $x = (x_1, x_2, \dots, x_n)$ représentant le vecteur de décision avec x_i , $i = 1, \dots, n$, les variables de décision et n le nombre de ces variables,
- $f(x) = (f_1(x), f_2(x), \dots, f_r(x))$ est le vecteur de r critères f_k , $k = 1, \dots, r$ et r le nombre de critères,
- L'ensemble S est un sous-ensemble de R^n décrit implicitement par des contraintes d'égalités et/ou d'inégalité,
- L'ensemble R^n qui contient S est dit *espace de décision*,
- L'ensemble R^r qui contient F est dit *espace des critères*,
- L'ensemble $F = f(S)$ qui est la projection de l'espace S sur l'espace des critères.

2.1.2 La relation de dominance

Lors de résolution d'un problème multi-objectif, toute solution admissible ($x \in S$), ne constitue pas évidemment un bon compromis à considérer ; pour qu'elle le soit, on impose généralement une propriété, basée sur une relation d'ordre partiel, notée " \succ " et appelée dominance :

Définition 2.1.1 (Dominance) Soient deux vecteurs critères $f(x), f(y) \in F$. On dit que $f(x)$ domine $f(y)$, et on note $f(x) \succ f(y)$, si et seulement si $f(x) \geq f(y)$ et

$$f(x) \neq f(y)$$

i.e.

$$f_i(x) \geq f_i(y) \quad \forall i = \{1, \dots, r\} \quad \text{et} \quad \exists i \in \{1, \dots, r\} \quad \text{tel que} \quad f_i(x) > f_i(y).$$

Si $f(x)$ domine $f(y)$, alors $f(x)$ est au moins aussi bon que $f(y)$ sur tous les critères et meilleur que lui sur au moins un critère.

Nous illustrons la relation de dominance par un exemple en dimension 2 (Voir figure 2.1).

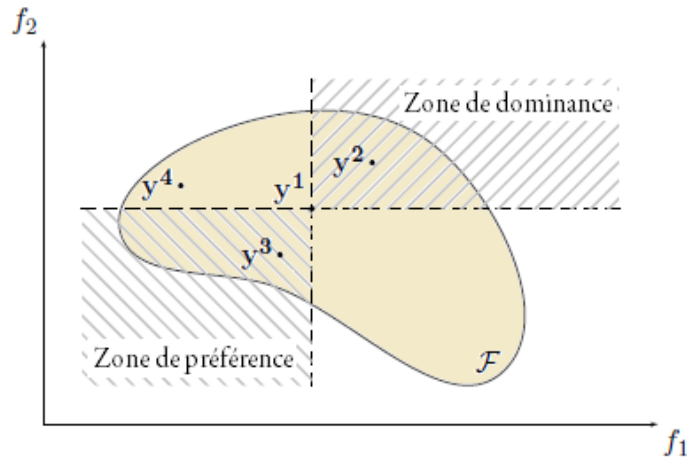


FIGURE 2.1 – Exemple de dominance

Sur cette figure, F (l'espace réalisable dans l'espace des critères) est l'image de S . Ainsi chaque point y^i est l'image de x^i par $f : y^i = f(x^i)$. Prenons le point y^1 comme point de référence, nous pouvons distinguer trois zones :

- La zone de préférence est la zone contenant les points dominés par y^1 .
- La zone de dominance est la zone contenant les points dominant y^1 .
- La zone d'indifférence contient les points incomparables avec y^1 .

Ainsi, il est clair que y^3 est dominée par y^1 , que y^2 domine y^1 et que y^4 est non dominée par y^1 (incomparable avec y^1).

2.1.3 Optimalité de Pareto

Définition 2.1.2 (Efficacité) Une solution $\bar{x} \in S$ est dite solution efficace (ou Pareto optimale) si et seulement s'il n'existe pas de solution $x \in S$ telle que $f(x)$ domine $f(\bar{x})$.

Un point est efficace si son image par f est un vecteur critère non dominé. Une définition équivalente de l'efficacité est :

Définition 2.1.3 Une solution $\bar{x} \in S$ est dite solution efficace si et seulement s'il n'existe pas de solution $x \in S$ telle que

$$f_i(x) \geq f_i(\bar{x}), \forall i \in \{1, \dots, r\} \text{ et } \exists j \in \{1, \dots, r\} \text{ avec } f_j(x) > f_j(\bar{x}).$$

A partir d'un point efficace, il est impossible d'augmenter la valeur d'un des critères sans diminuer la valeur d'au moins un autre critère.

Définition 2.1.4 Une solution $\bar{x} \in S$ est dite solution faiblement efficace si et seulement s'il n'existe pas de solution $x \in S$ telle que :

$$f_i(x) > f_i(\bar{x}), \forall i \in \{1, \dots, r\}.$$

Il est clair qu'une solution efficace est faiblement efficace, mais l'inverse est faux.

Nous pouvons définir l'optimalité locale et l'optimalité globale au sens de Pareto comme suit :

Définition 2.1.5 (Optimalité locale au sens de Pareto) Un vecteur $x \in S$ optimal localement au sens de Pareto s'il existe un réel $\delta > 0$ tel qu'il n'y ait pas de vecteur x' dont le vecteur critère domine celui de x avec $x' \in S \cap B(x, \delta)$, ou $B(x, \delta)$ représente une boule de centre x et de rayon δ .

D'une manière équivalente un vecteur x est optimal localement au sens de Pareto s'il est optimal au sens de Pareto sur une restriction de l'ensemble S .

Définition 2.1.6 (Optimalité globale au sens de Pareto) Un vecteur $x \in S$ optimal globalement au sens de Pareto (ou optimal au sens de Pareto) s'il n'existe pas de vecteur x' dont le vecteur critère domine celui de x .

Une version "graphique" de l'optimalité au sens de Pareto utilise le théorème du contact.

Définition 2.1.7 Soit $V \subset \mathbb{R}^r$, $V \neq \emptyset$. V est un cône si $\alpha.v \in V$ pour tous scalaire $\alpha \geq 0$ et tout $v \in V$.

Définition 2.1.8 (Cône non négatif) Un cône non négatif est défini dans \mathbb{R}^r par $C^+ = \{x | f(x) \in \mathbb{R}^r \text{ et } f(x) \geq 0\}$.

Définition 2.1.9 (Théorème du contact) Un vecteur x est optimal au sens de Pareto pour un problème d'optimisation multi-objectif donné si $(C^+ + x) \cap F = \{x\}$.

L'utilisation de ce théorème est illustré par la figure 2.2

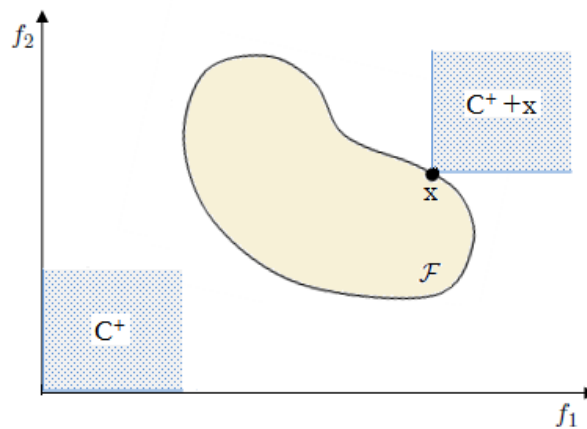


FIGURE 2.2 – Le théorème du contact

Définition 2.1.10 (Ensemble Pareto optimal) *L'ensemble Pareto optimal de S (ou solutions efficaces), est défini par l'ensemble Eff :*

$$Eff = \{x \in S \mid \nexists x' \in S, f(x') \text{ domine } f(x)\}.$$

Définition 2.1.11 (Frontière Pareto de F) *Soit F l'image dans l'espace des critères de l'ensemble réalisable S . La frontière Pareto SND de F est définie par :*

$$SND = \{y \in F \mid \nexists y' \in F, y' > y\}.$$

Elle est aussi définie comme l'image de l'ensemble Pareto optimal dans l'espace F .

Un exemple de surface de Pareto en dimension 2 est montré à la figure 2.3.

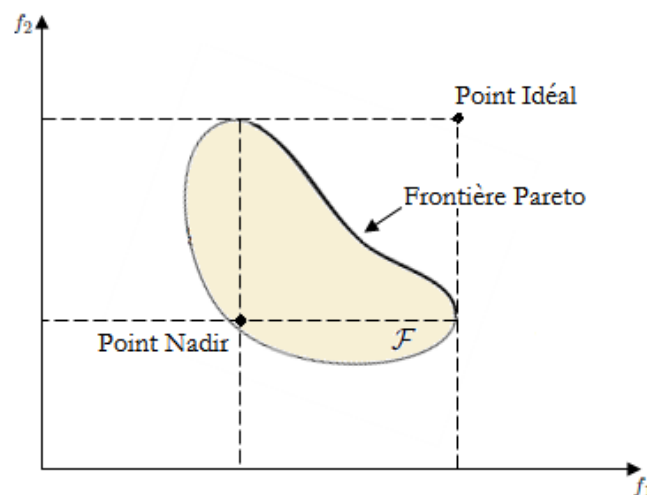


FIGURE 2.3 – La Frontière Pareto

Dans cet exemple, le problème considéré est un problème de maximisation avec deux critères.

Deux points particuliers apparaissent clairement : le *point idéal* et le *point Nadir*, ces deux points sont calculés à partir de la frontière Pareto. Le point idéal (resp. le point Nadir) domine (resp. est dominé par) tous les autres points de la surface de Pareto. Bien que ces points ne soient pas forcément compris dans la zone réalisable, ils servent comme *point de référence* permettant de discuter de l'intérêt des solutions trouvées. Les coordonnées de ces points sont définies comme suit :

Définition 2.1.12 (Point Idéal) *Les coordonnées du point idéal (Z^I) correspondent aux meilleurs valeurs de chaque objectif des points du front Pareto. Les coordonnées de ce point correspondent aussi aux valeurs obtenues en optimisant chaque fonction objectif séparément :*

$$Z_i^I = \max\{y_i \mid y \in \text{SND}(F)\}.$$

Définition 2.1.13 (Point Nadir) *Les coordonnées du point Nadir (Z^{nad}) correspondent aux pires valeurs de chaque objectif des points du front Pareto.*

$$Z_i^{\text{nad}} = \min\{y_i \mid y \in \text{SND}(F)\}.$$

La figure 2.3 nous indique aussi que la frontière Pareto peut avoir des propriétés particulières quant à sa forme. La principale caractéristique utilisée pour comparer les formes de ces courbes est la convexité.

La convexité est le premier indicateur de la difficulté du problème. En effet, certaines méthodes sont dans l'incapacité de résoudre des problèmes non convexes de manière optimale. Mais il existe d'autres indicateurs tout aussi importants, notamment la continuité, la nature des variables de décision (entières ou réelles),...

2.1.4 Dominance au sens de Geoffrion

Une forme de dominance importante dans le monde de l'optimisation multi-objectif est la dominance au sens de Geoffrion [39], [33] et [56]. Les solutions optimales obtenues par ce type de dominance sont appelées les *solutions Pareto optimales propres*.

Définition 2.1.14 (Dominance au sens de Geoffrion [39]) *Une solution $x^* \in S$ est appelée solution Pareto optimale propre si :*

- Elle est Pareto optimale ;

- Il existe un nombre $M > 0$ tel que $\forall i$ et $\forall x \in S$ vérifiant $f_i(x) > f_i(x^*)$, il existe un indice j tel que

$$f_j(x) > f_j(x^*) \text{ et } \frac{f_i(x^*) - f_i(x)}{f_j(x) - f_j(x^*)} \geq M$$

Un théorème relatif à la méthode de pondération des critères utilisant ce résultat est le suivant :

Théorème 2.1.1 *Soit la méthode d'agrégation des critères suivantes :*

$$f(x) = \sum_{i=1}^r \lambda_i \cdot f_i(x) \text{ avec } \forall i \in \{1, \dots, r\}, \lambda_i > 0 \text{ et } \sum_{i=1}^r \lambda_i = 1$$

Si x^ est une solution optimale obtenue en utilisant la méthode d'agrégation ci-dessus, alors cette solution est aussi Pareto optimale propre.*

D'autres types de relations de dominance sont définis dans (Collette et al.) [19].

2.1.5 Caractérisation des solutions efficaces

Nous présentons quelques caractérisations qui permettent de tester l'efficacité d'une solution réalisable d'un problème d'optimisation multi-objectif.

Théorème 2.1.2 (Wendell[78]) *Soit $x^* \in S$ un vecteur décision donné et \hat{x} un vecteur quelconque de S . Le vecteur x^* est Pareto optimal pour le problème multi-objectif (P_{MO}) si et seulement si x^* est une solution optimale du problème auxiliaire suivant :*

$$\begin{cases} \min \sum_{i=1}^r f_i(x) \\ \text{s.c. } x \in S \text{ avec } f_i(x) \leq f_i(\hat{x}), \forall i \in \{1, \dots, r\}. \end{cases} \quad (2.1)$$

Théorème 2.1.3 (Benson [9]) *Soit x^* une solution réalisable arbitraire donnée et soit le problème unicritère suivant :*

$$\begin{cases} \max \Theta \sum_{i=1}^r \varepsilon_i \\ \text{s.c. } x \in S \text{ avec } f_i(x) + \varepsilon_i = f_i(x^*), \text{ avec } \varepsilon_i \geq 0 \forall i \in \{1, \dots, r\}. \end{cases} \quad (2.2)$$

Le vecteur x^ est Pareto optimal pour (PMO) si et seulement si la valeur optimale de la fonction objectif Θ est nulle dans le problème 2.2.*

Si pour un point \hat{x} , la valeur de Θ est différente de zéro alors \hat{x} est Pareto optimal.

Définition 2.1.15 *Soit $S \in \mathbb{R}^n \rightarrow \mathbb{R}$ et $\bar{x} \in S$. Alors :*

$L_{\geq} f(\bar{x}) = \{x \in S : f(x) \geq f(\bar{x})\}$ est appelé ensemble niveau (level set) de \bar{x} pour f .

$L_{=} f(\bar{x}) = \{x \in S : f(x) = f(\bar{x})\}$ est appelé courbe niveau (level curve) de \bar{x} pour f .

Théorème 2.1.4 (Ehrgott [31]) *Soit $\bar{x} \in S$ est Pareto optimale pour le problème multi-objectif P_{MO} si et seulement si :*

$$\bigcap_{i=1}^r L_{\geq} f_i(\bar{x}) = \bigcap_{i=1}^r L_{=} f_i(\bar{x}) \quad (2.3)$$

2.2 Choix de la méthode d'aide à la décision

Dans la résolution d'un problème multi-objectif menant à la détermination d'un ensemble de solutions Pareto, il est nécessaire de faire intervenir l'humain à travers un décideur, pour le choix final de la solution à garder. Ainsi, avant de se lancer dans la résolution d'un problème multi-objectif, il faut se poser la question du type de méthode d'optimisation à utiliser. Elles peuvent être classées suivant trois catégories qui diffèrent selon les préférences du décideur pour la construction de sa fonction d'utilité.

Nous pouvons trouver les familles suivantes [27] :

- *Les méthodes d'optimisation à priori* : dans ce cas, le compromis que l'on désire faire entre les critères a été défini avant l'exécution de la méthode. Ainsi une seule exécution permettra d'obtenir la solution recherchée. Cette approche est donc rapide, mais il faut cependant prendre en compte le temps de modélisation du compromis et la possibilité pour le décideur de ne pas être satisfait de la solution trouvée et de relancer la recherche avec un autre compromis.
- *Les méthodes d'optimisation progressives* : ici, le décideur intervient dans le processus de recherche de solutions en répondant à différentes questions afin d'orienter la recherche. Cette approche permet donc de bien prendre en compte les préférences du décideur, mais nécessite sa présence tout au long du processus de recherche.
- *Les méthodes d'optimisation à posteriori* : Dans cette troisième famille de méthodes, on cherche à fournir au décideur un ensemble de bonnes solutions bien réparties. Il peut ensuite, au regard de l'ensemble des solutions, sélectionner celle qui lui semble la plus appropriée. Ainsi, il n'est plus nécessaire de modéliser les préférences du décideur (ce qui peut s'avérer être très difficile), mais il faut en contre-partie fournir un ensemble de solutions bien réparties, ce qui peut également être difficile et requérir un temps de calcul important (mais ne nécessite pas la présence du décideur).

On peut trouver des méthodes d'optimisation multi-objectif qui n'entrent pas exclusivement dans une famille. On peut utiliser, par exemple, une méthode a priori en lui fournissant des préférences choisies aléatoirement.

Nous nous placerons dans le cadre de cette troisième famille de méthodes où la modélisation des préférences n'est pas requise et où le procédé d'optimisation doit être

puissant afin de fournir l'ensemble de solutions Pareto optimales ou à défaut une très bonne approximation de la frontière Pareto.

Dans ce type de méthode, deux phases importantes sont à considérer : la phase de recherche de l'ensemble des solutions Pareto optimales, que nous appellerons de façon abusive, *résolution du problème d'optimisation* et la phase de choix parmi ces solutions, qui relève de l'aide à la décision. Cette deuxième phase ne sera pas traitée dans notre travail.

2.3 Approches de résolution d'un problème multi-objectif (MOP)

Dans la littérature, plusieurs approches de résolution du problème de la programmation multi-objectif sont considérées. Des ouvrages ou articles de synthèse ont été rédigés, des états de l'art plus complets peuvent être consultés notamment dans (Ulungu et al.) [77], Miettinen [56], Ehrgott [33], (Ehrgott et al.) [32], Deb [26], (Collette et al.) [19].

Deux approches de résolution d'un problème d'optimisation multi-objectif peuvent être distinguées dans la littérature voir Ehrgott [33], (Roy et al.) [65] et Roy [66].

- La première approche dite *approche mono-objectif* consiste à ramener le problème à un problème d'optimisation unicritère, au risque d'enlever toute signification au problème.
- La deuxième approche dite *approche multi-objectif* consiste à proposer des solutions en tenant compte de l'ensemble des critères.

2.3.1 Approche unicritère

Elle caractérise les méthodes "à priori" utilisés pour leur simplicité de mise en oeuvre. En effet, les critères du problème d'optimisation sont transformés en un seul critère. Dans ce cas, le décideur est supposé quantifier à priori l'importance de chaque critère afin de construire un critère unique. Le processus d'optimisation unicritère est ensuite lancé afin d'obtenir la solution "optimale". Plusieurs exécutions sont effectuées dans le but de trouver un ensemble de solutions qui approxime l'ensemble optimal de Pareto.

Nous donnons ci-dessous des méthodes a priori ou méthodes scalaires les plus connues, d'autres méthodes sont exposées dans Miettinen [56], Collette et al. [19].

Théorie de l'utilité multi-attributs

La théorie de l'utilité multi-attributs a fait l'objet de nombreux travaux de recherche dans les années 70 voir Fishburn [36], Keeney [48], Huber [42], Farquhar [35]. Elle repose sur l'axiome fondamental suivant : tout décideur essaie inconsciemment d'optimiser une fonction $U = U(f_1, \dots, f_r)$ qui agrège tous les points de vue à prendre en compte, comme cela se présente habituellement dans la théorie du consommateur en économie. En d'autres termes, si l'on interroge le décideur sur ses préférences, ses réponses seront en accord avec une certaine fonction U que l'on ne connaît pas. Le problème est donc d'essayer d'estimer cette fonction.

Définition 2.3.1 *Étant donné un ordre de préférence noté " \succ " sur l'ensemble F , une fonction U à valeurs réelles vérifiant $U(f_1) > U(f_2) \Leftrightarrow f_1 \succ f_2$; est appelée fonction de préférence ou d'utilité.*

Deux problèmes essentiels se posent dans le cadre de cette théorie :

- Quelles propriétés doivent posséder les préférences du décideur pour être représentable par une fonction U ayant une forme analytique donnée (additive, multiplicative, mixte,...).
- Comment construire ces fonctions et estimer les paramètres intervenant dans la forme analytique choisie.

Différentes formes de fonctions d'utilité conduisent à différentes solutions pour le problème multi-objectif. Les modèles les plus couramment utilisés pour la fonction d'utilité U sont le modèle additif et le modèle multiplicatif :

$$U(f(x)) = \sum_{i=1}^r U_i(f_i(x)) \text{ et } U(f(x)) = \prod_{i=1}^r U_i(f_i(x))$$

où les fonctions U_i sont strictement croissantes et à valeurs réelles. Elles servent uniquement à transformer les critères initiaux f_i de manière à ce qu'ils s'expriment tous suivant la même échelle.

Méthode des sommes pondérées

Cette méthode de résolution d'un problème d'optimisation multi-objectif est la plus évidente et, probablement, la plus largement utilisée en pratique. Elle consiste à ramener le problème multi-objectif au problème de l'optimisation d'une combinaison linéaire des objectifs initiaux. Ainsi, il s'agit d'associer à chaque critère un coefficient de pondération et à faire la somme des critères pondérées pour obtenir un nouveau et unique critère.

Le problème multi-objectif (*MOP*) se transforme alors de la manière suivante :

$$(MOP) \begin{cases} \max \sum_{i=1}^r \lambda_i f_i(x) \\ sc. \\ x \in S \text{ et } \lambda_i \in \Lambda = \{\lambda_i : \sum_{i=1}^r \lambda_i = 1 \text{ et } \lambda_i \geq 0, \forall i \in \{1, \dots, r\}\}. \end{cases}$$

- Les résultats obtenus avec de telles méthodes dépendent fortement des paramètres choisis pour le vecteur de poids. Les poids λ_i doivent également être choisis en fonction des préférences associées aux objectifs, ce qui est une tâche délicate. Une approche généralement utilisée consiste à répéter l'exécution de l'algorithme avec des vecteurs poids différents.
- Cette méthode est très efficace du point de vue algorithmique, mais son principal inconvénient est qu'elle ne permet pas de trouver les solutions enfermées dans des concavités (surface de compromis non-convexe).

Méthode Goal programming

Cette méthode, qui relève d'une théorie très avancée dans le domaine des problèmes multi-objectifs, a été initialement conçue par Charnes et Cooper [18] dans le cas linéaire ; elle a été prolongée par des travaux d'Ijiri [44] et d'Ignizio [43] dans le cas non linéaire. Cette théorie a fait l'objet d'un nombre important de travaux théoriques et pratiques voir Chankong [17], Martel [54], Spronk [72] et Steuer [73]. L'idée générale de la méthode est d'établir un but à atteindre pour chaque critère. Généralement, le point qui satisfait tous les buts n'est pas réalisable, la solution préférée serait donc celle qui se rapproche le plus possible de ces buts. Soit $f^* = (f_1^*, \dots, f_r^*)$ le vecteur de référence ou but fixé par le décideur relativement à tous les critères, le problème revient à considérer la relation suivante :

$$\begin{cases} \min(\sum_{i=1}^r |f_i(x) - f_i^*|^p)^{1/p} \\ sc. \\ x \in S. \end{cases}$$

où $p \geq 1$, et f_i^* est le vecteur de référence (but) ou le vecteur idéal.

La méthodologie du Goal programming repose sur les points suivants :

- Fixer les valeurs des cibles f_i^* que l'on désire atteindre sur chaque critère ;
- Définir des déviations positives d_i^+ et négatives d_i^- relativement à ces buts ;
- Minimiser la somme pondérée de ces déviations relativement à une norme $\|\cdot\|_p$ définie dans \mathbb{R}^n .

La formulation mathématique du Goal programming est la suivante :

$$\left\{ \begin{array}{l} \min_{x \in E} (\sum_{i=1}^r (d_i^- + d_i^+)^p)^{1/p} \\ sc. \\ f_i(x) + d_i^+ d_i^- = f_i^* \\ d_i^- \geq 0, d_i^+ \geq 0, i \in \{1, \dots, r\}. \end{array} \right.$$

d_i^+ et d_i^- sont respectivement appelés la sur-réalisation et la sous-réalisation du $i^{\text{ème}}$ critère.

Le Goal programming se ramène toujours à un problème de minimisation d'une seule fonction. La solution qui en découle est efficace ou faiblement efficace selon la norme utilisée. Son avantage est que la solution qui en résulte satisfasse le décideur de la façon la plus proche possible lorsque les buts et les priorités de chaque but à atteindre sont bien définis.

La méthode ϵ -contrainte

Dans cette méthode, on n'optimise qu'un seul objectif jugé important par le décideur, les autres sont transformés en contraintes d'inégalités par rapport à un vecteur seuil ϵ . Le principe de la méthode ϵ -contrainte ou méthode de compromis est intéressant lorsque l'on cherche à énumérer toutes les solutions d'un front Pareto. Le problème peut être reformulé de la manière suivante :

$$\left\{ \begin{array}{l} \max f_i(x) \\ sc. \\ f_1(x) \leq \epsilon_1 \\ \vdots \\ f_{i-1}(x) \leq \epsilon_{i-1} \\ f_{i+1}(x) \leq \epsilon_{i+1} \\ \vdots \\ f_r(x) \leq \epsilon_r \\ x \in S \end{array} \right.$$

L'approche par ϵ -contrainte doit aussi être appliquée plusieurs fois en faisant varier le vecteur ϵ pour trouver un ensemble de points Pareto optimaux. Cette approche a l'avantage par rapport à la précédente de ne pas être trompée par les problèmes non convexes.

L'inconvénient de cette approche réside dans le fait qu'il faille lancer un grand nombre de fois le processus de résolution. De plus, pour obtenir des points intéressants et bien

répartis sur la surface de compromis, le vecteur ϵ doit être choisi judicieusement. Il est clair qu'une bonne connaissance du problème a priori est requise.

2.3.2 Approche multi-objectif

A- Approche à posteriori

Dans ces approches l'expression des préférences et la pondération des objectifs se fait a posteriori, c'est à dire que le processus d'optimisation détermine un ensemble de solutions candidates et la sélection de la meilleure solution ne se fait qu'à la fin du processus.

Les méthodes hybrides

La méthode hybride la plus connue est la méthode de Corley [21] en 1980. Cette méthode utilise la méthode de pondération des fonctions objectifs et la méthode de ϵ -contraintes. Le problème peut être reformulé de la manière suivante :

$$\left\{ \begin{array}{l} \max \sum_{i=1}^r \lambda_i f_i(x) \\ sc. \\ f_j(x) \leq \epsilon_j \text{ pour } j \in \{1, \dots, r\} \\ x \in S \end{array} \right.$$

En faisant varier le vecteur ϵ dans ce problème paramétrique, on génère ainsi l'ensemble de toutes les solutions Pareto optimales.

- Cette méthode permet de combiner les avantages de la méthode des sommes pondérées et la méthode ϵ -contrainte. Elle est ainsi efficace sur différents types de problèmes, qu'ils soient convexes ou non convexes.
- Le nombre de paramètres à déterminer a été multiplié par deux. Il sera beaucoup plus difficile à l'utilisateur d'exprimer sa préférence en jouant sur l'ensemble des paramètres.

D'autres approches à posteriori sont détaillées dans Miettinen [56], Collette et al. [19].

B- Approche interactive

Le principe des méthodes de l'approche progressive est de guider l'exploration tout au long du processus d'optimisation. Le but est d'alterner entre le processus de recherche

et le processus de décision défini par un classement de solutions, ou bien un choix des poids de pondération des objectifs. A chaque étape, l'ensemble des solutions est analysé afin d'orienter les futures itérations vers les zones les plus intéressantes de l'espace de recherche. Plusieurs méthodes interactives sont traitées dans Miettinen [56], Collette et al. [19].

Méthode STEM

Dans cette méthode, les informations sur la préférence de l'utilisateur permettent de restreindre l'espace de recherche étape par étape en ajoutant des contraintes sur les valeurs des critères. A chaque étape interactive, un nouveau compromis est trouvé en optimisant une norme de type :

$$S(Z, \lambda) = \max_{1 \leq i \leq r} \{\lambda_i |Z_i - \bar{z}_i|\} + \rho \sum_{i=1}^r \lambda_i |Z_i - \bar{z}_i|, \quad \rho > 0.$$

sur une partie de $f(S)$ selon une direction correspondant à la relaxation d'un critère fixé par les décideurs.

Les différentes étapes de cette méthodes sont décrites dans [8].

- Le problème de cette méthode, commun à toutes les méthodes interactives, est que l'on ne peut trouver qu'un seul point solution, et non la totalité de la surface de compromis.
- Le mode d'interaction fait que, pour une bonne utilisation de la méthode, l'utilisateur doit avoir une bonne connaissance à priori du problème. Il doit être capable de trouver de bonnes bornes pour restreindre son domaine.

2.4 Conclusion

Nous avons présenté dans ce chapitre les principaux concepts de l'optimisation multi-objectif tels que la modélisation d'un problème multi-objectif, la notion de la dominance de Pareto et la structure de la surface de compromis. Une classification des méthodes de résolution a été introduite tout en essayant de présenter plusieurs approches utilisées pour aborder les problèmes de l'optimisation multi-objectif, ces méthodes sont passées en revue et les avantages et les inconvénients de chacune sont mentionnées.

Pour le reste de cette thèse, nous nous sommes astreints à cerner le cadre de notre étude à l'optimisation "à posteriori" (cherchant à générer l'ensemble du front Pareto) à l'aide de méthodes exactes.

Le chapitre suivant est dédié aux problèmes de la programmation multi-objectif en nombre entiers sous contraintes linéaires qui représente le cadre générale de notre travail.

Chapitre 3

Programmation Linéaire

Multi-objectif en Nombre Entiers

MOILP

LA programmation linéaire est l'une des plus importantes techniques d'optimisation utilisées en recherche opérationnelle. Ses développements théoriques ont été suggérés et accélérés par un grand nombre d'applications pratiques, dans le domaine de l'économie, de la gestion et autres.

En pratique, très souvent, la présence de variables discrètes (ou entières) est inévitable dans la modélisation en optimisation et ces variables modifient considérablement la structure mathématique du problème. Il en résulte que des méthodes spécifiques à cette situation sont nécessaires.

Ce chapitre a pour objectif principal de présenter le contexte de l'optimisation linéaire multi-objectif discrète et de présenter quelques méthodes exactes permettant de caractériser totalement ou partiellement l'ensemble de solutions efficaces du problème.

3.1 Formulation du problème

Un problème de la programmation linéaire multi-objectif en nombres entiers est défini comme suit :

$$(P) \begin{cases} \max Z_1(x) = c^1 x \\ \max Z_2(x) = c^2 x \\ \vdots \\ \max Z_r(x) = c^r x \\ sc \\ x \in S = \{x \in \mathbb{R}^n | Ax \leq b, x \geq 0\} \\ x \text{ vecteur entier} \end{cases}$$

où $r \geq 2$, $c^i \in \mathbb{R}^n$ pour tout $i = \{1, \dots, r\}$; A est une $m \times n$ -matrice et b un m -vecteur à coefficients entiers.

La solution du problème de programmation multi-objectif linéaire à variables entières (P) consiste à trouver toutes les solutions efficaces au sens de la définition suivante :

Définition 3.1.1 *Un point $x^0 \in D$ est efficace ou Pareto optimale pour (P) si et seulement si il n'existe pas un autre point $x^1 \in S$ tel que $Z_i(x^1) \geq Z_i(x^0)$ pour tout $i \in \{1, \dots, r\}$ et $Z_i(x^1) > Z_i(x^0)$ pour au moins un $i \in \{1, \dots, r\}$.*

3.2 Solutions supportées et non supportées

Considérons le problème (P_R) relaxation continue du problème (P) :

$$(P_R) \begin{cases} \max Z_1(x) = c^1 x \\ \max Z_2(x) = c^2 x \\ \vdots \\ \max Z_r(x) = c^r x \\ sc \\ x \in S \end{cases}$$

Le théorème suivant s'applique à un programme avec objectifs linéaires et domaine réalisable convexe.

Théorème 3.2.1 (Geoffrion[39]) *Soit le problème unicritère linéaire suivant :*

$$(P_\lambda) \begin{cases} \max \sum_{i=1}^r \lambda_i f_i(x) \\ sc \\ x \in S \text{ et } \lambda_i \in \Lambda, \forall i \in \{1, \dots, r\} \end{cases} \quad (3.1)$$

La solution x^* est optimale au sens de Pareto si et seulement si x^* est une solution optimale du problème paramétrique (P_λ) où $\Lambda = \{\lambda_i : \sum_{i=1}^r \lambda_i \geq 0, \forall i \in \{1, \dots, r\}\}$

D'après ce théorème, l'ensemble des solutions efficaces du problème (P_R) sans les contraintes d'intégrité est bien caractérisé par les solutions du problème paramétrique (P_λ) . Ces solutions se trouvent sur la frontière de S , elles sont appelées *solutions supportées*.

Différemment du cas continu, la difficulté principale rencontrée lorsqu'on traite les problèmes multi-objectif à variables discrètes est l'existence de solutions efficaces pour (P) qui ne sont pas optimales pour (P_λ) et ce en raison de la non-convexité du domaine réalisable, ces solutions efficaces sont dites solutions *non supportées* (le front Pareto de (P) est l'union de l'ensemble des solutions supportées et de l'ensemble des solutions non supportées de (P)).

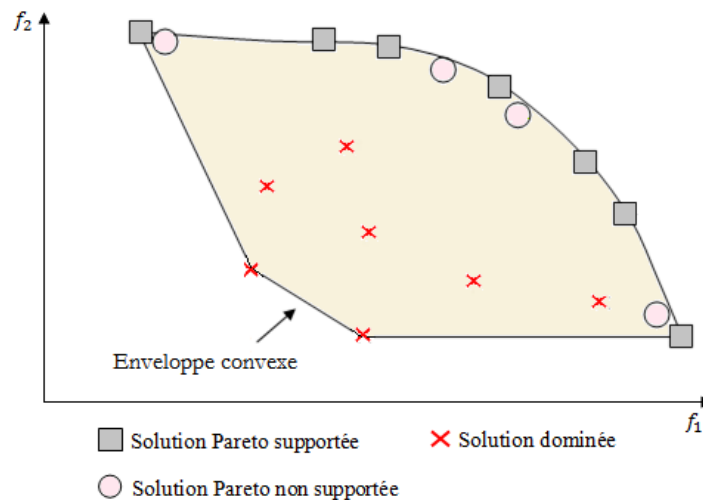


FIGURE 3.1 – Représentation des différents types de solutions en bicritère

La nature des problèmes de programmation linéaire en variables continues et les problèmes de programmation linéaire en variables discrètes est différente. Contrairement à la programmation linéaire continue où on s'intéresse seulement aux solutions sommets du polyèdre, les solutions optimales du problème discret peuvent se trouver à l'intérieur du polyèdre et par conséquent la recherche d'une solution optimale d'un problème de la programmation en nombres entiers est souvent NP-difficile et peut être même NP-complet [69].

Il faut noter cependant que, si un problème d'optimisation combinatoire est facile à résoudre, il n'est pas de même pour sa version multi-objectif.

Dans le paragraphe suivant, nous donnons un bref aperçu sur des méthodes générales pour la résolution d'un *MOILP*. Rappelons que la résolution d'un problème multi-objectif consiste à déterminer soit l'ensemble des solutions efficaces dans l'espace des décisions noté *Eff*, soit l'ensemble des solutions non dominées dans l'espace des critères noté *SND*. Dans la littérature, l'accent est mis sur la recherche de l'ensemble des solutions non dominées compte tenu de son cardinal qui est généralement moins important que celui des solutions efficaces ; plusieurs solutions efficaces pouvant donner lieu à un même vecteur non dominé. Il arrive que deux solutions efficaces différentes dans l'espace de décisions ont exactement les mêmes valeurs pour tous les objectifs. Si on garde les deux solutions dans *Eff*, on parle alors de l'ensemble complet, sinon c'est juste un ensemble minimal.

3.3 Quelques méthodes de résolution d'un problème *MOILP*

Plusieurs chercheurs, citons en particulier, Steuer et Choo [73], Klein et Hannan [49], Crema et Sylva [22], Gupta et Malhotra [40], Abbas et Moulai [3], Abbas et Chaabane [2], motivés par de nombreuses stratégies, se sont intéressés à caractériser totalement ou partiellement l'ensemble des solutions efficaces du problème de programmation linéaire multi-objectif en nombres entiers. Dans cette partie, quelques méthodes de résolution des problèmes *MOILP* sont exposées.

3.3.1 Méthode de D.Klein & E.Hannan :

La technique proposée par D.Klein & E.Hannan [49] peut être utilisée aussi bien pour identifier l'ensemble de toutes les solutions efficaces que pour en caractériser une partie seulement. Elle consiste à résoudre progressivement une séquence de programmes linéaires unicritère en nombres entiers avec des contraintes ajoutées à chaque étape. Les contraintes supplémentaires éliminent les solutions efficaces déjà trouvées, et font en sorte que les nouvelles solutions générées soient efficaces.

Algorithme 1 : KLEIN & HANNAN

Étape1 : Résoudre le problème (P_1) défini comme suit (L'indice i est pris arbitrairement dans $1, \dots, r$) :

$$(P_1) : \max\{Z_i = c^i, x \in D\}$$

Si la solution optimale de (P_1) , soit x_1 , est unique, **alors** elle est efficace pour (P) .

Sinon , déterminer toutes les solutions alternatives et à x_1 et par comparaison deux à deux des vecteurs critères associés, garder uniquement celles qui sont efficaces pour construire l'ensemble $Eff(P_1)$ des solutions efficaces générées à l'étape 1.

Étape générale j : A l'étape j , on résout le problème (P_j) qui est défini comme suit :

$$(P_j) \begin{cases} \max Z_i = c^i x \\ sc \\ x \in D \\ \bigcap_{l=1}^q \left(\bigcup_{\substack{i=1 \\ i \neq s}}^r c^i x \geq c^i y^l + f_i \right) \end{cases}$$

Avec $f_i \geq 1$ entier, y^l ($l = 1, \dots, q$) les points efficaces obtenus jusqu'à l'étape $j - 1$. Si $Eff(P_j)$ est l'ensemble des solutions efficaces obtenues à l'étape j et Y^j l'ensemble des points efficaces accumulés à la fin de l'étape j , alors $Y^j = Y^{j-1} \cup Eff(Y^j)$ pour $j \geq 2$ avec $Y^1 = Eff(P_1)$.

Étape finale n : La procédure s'arrête lorsque le problème (P_n) est irréalisable.

Si à chaque étape j , $f_i = 1$; $\forall i = \{1, \dots, r\}$; $i \neq s$, la procédure trouve l'ensemble de toutes les solutions efficaces du problème (P) . Cependant, si pour certaines valeurs de i , $f_i > 1$ seulement un sous ensemble de solutions efficaces sera généré.

3.3.2 Méthode de Gupta et Malhotra

Les auteurs R. Gupta et R. Malhorta [40] ont proposé deux procédures pour résoudre le problème *MOILP*. La première est basée sur une méthode de coupe Mais il s'est avéré qu'une erreur au niveau du second test d'arrêt empêche dans certains cas l'algorithme de donner tous les points efficaces du problème étudié. Un contre exemple a été présenté par M. Abbas et M. Moulai [3] ainsi que par D. Chaabane. La deuxième est une variante de la méthode de Klein et Hannan [49], elle est décrite pour générer l'ensemble de toutes les solutions efficaces en réduisant, à chaque étape, le nombre de contraintes additionnelles de Klein et Hannan. Malheureusement, ce n'est pas toujours le cas; la méthode peut s'arrêter avant terme, sans trouver toutes les solutions efficaces. Un contre exemple est

donné par [3] et dans [2].

Algorithme 2 : GUPTA & MALHOTRA

Étape1 : Résoudre le problème (P_1) définit comme suit :

$$(P_1) : \max\{Z_1 = c^1, x \in D\}$$

Déterminer toutes les solutions optimales de (P_1) , éliminer celles dont les vecteurs critères associés sont dominés pour construire l'ensemble $E(P_1) = \{y_1^1, y_1^2, \dots, y_1^r\}$ des solutions efficaces générées à l'étape 1.

Étape générale $j + 1$: Résoudre le problème (P_{j+1}) définit par :

$$(P_{j+1}) \begin{cases} \max Z_1 = c^1 x \\ sc \\ x \in D \\ c_1 x \leq \max\{c_1 y - 1 | y \in E(P_j)\} = c_1 y^s - 1 \\ c_i x \leq c_i y^s + 1 \text{ pour au moins un } i = \overline{2, r} \end{cases}$$

Où $E(P_j)$ représente l'ensemble de toutes les solution potentiellement efficaces obtenues à l'étape j .

S'il y a plus d'un y^s donnant le maximum de $c_1 y - 1$, $y \in E(P_j)$ alors, sélectionner y^s arbitrairement.

Enregistrer toutes les solutions potentiellement efficaces obtenues à l'étape $j + 1$ dans $E(P_{j+1})$. La valeur de z_1 a diminué de celle enregistrée à l'étape j d'au moins une unité et la valeur d'au moins un critère i , $i = \overline{2, r}$, s'est améliorée grâce aux contraintes supplémentaires.

Étape finale n : La procédure prend fin dans l'un des cas suivants :

- Toutes les solutions de (P_n) ne sont pas efficaces pour (P) .
 - (P_n) n'est pas réalisable.
-

Remarque 3.3.1 Selon le premier test d'arrêt proposé par GUPTA & MALHOTRA [40], si toutes les solutions de (P_n) ne sont pas efficaces pour le problème (P) alors, toutes les étapes ultérieures donnent des solutions non efficaces. Un contre exemple donné dans [60] montre que ceci n'est pas vrai et qu'il est possible de continuer à chercher d'autres solutions efficaces.

3.3.3 Méthode de M.Abbas et M.Moulaï

Cette méthode a été proposée par M. Abbas & M. Moulai [3] pour la détermination de toutes les solutions efficaces du problème de programmation linéaire multi-objectif en nombres entiers. Elle peut être vue comme une alternative à celle de Gupta & Malhotra [40] (première procédure), où les auteurs ont proposé un autre test d'arrêt permettant à l'algorithme de fournir toutes les solutions efficaces.

On considère le problème :

$$(P_1) : \max\{Z_1 = c^1, x \in D\}$$

dont le problème relaxé est :

$$(P_1) : \max\{Z_1 = c^1, x \in S\}$$

avec $S = \{x \in \mathbb{R}^n | Ax \leq b, x \geq 0\}$.

Pour les besoins de la description de la méthode, on définit les paramètres suivants pour $k \leq 1$:

- $S_k = \{x \in \mathbb{R}^{n_k} | A_k x \leq b_k, A_k \in \mathbb{R}^{m_k \times n_k}, b_k \in \mathbb{R}^{m_k}, x \geq 0\}$ comme étant la région courante tronquée de S obtenue par application de la coupe :

$$\sum_{j \in N_{k-1} \setminus \{j_{k-1}\}} x_j \geq 1.$$

et éventuellement des coupes successives de Gomory, avec j_{k-1} un indice hors base quelconque.

- x^k : la $k^{\text{ième}}$ solution optimale entière du problème (P_1) obtenue sur S_k .
- B_k : une base de S_k .
- N_k : ensemble des indices des variables hors base de x_k .
- $\hat{A}_k = (A_{B_k})^{-1} A_k$.
- $\hat{C}_k = (C_k) - \pi_k A_k$ avec $\pi_k = (C_{B_k})(A_{B_k})^{-1}$.
- $\Gamma_k = \{j \in N_k | (\hat{c}_k^1)_j < 0 \text{ et } (\tilde{c}_k^i)_j > 0 \text{ pour au moins un critère } i, i = \overline{2, r}\}$.
- $\Omega_k = \{j \in N_k | (\hat{c}_k^1)_j = 0\}$.
- $\Psi_k = \{j \in N_k | (\hat{c}_k^1)_j < 0 \text{ et } (\tilde{c}_k^i)_j < 0 \text{ pour au moins un critère } i, i = \overline{2, r}\}$.

Définition 3.3.1 Une arête E^{j_k} , $j_k \in N_k$ incidente à x^k est définie comme étant l'ensemble :

$$\left\{ X \in \mathbb{R}^{n_k} \mid \begin{cases} x_i = x_i^k - \theta_{j_k} \widehat{a}_{lig(i)_k}^{j_k} & i \in B_k \\ x_{j_k} = \theta_{j_k} \\ x_l = 0 \quad \forall l \in N_k \setminus j_k \end{cases} \right\} \quad (3.2)$$

$$\text{où } 0 \leq \theta_{j_k} \leq \theta = \min_{i \in B_k} \left\{ \frac{x_i^k}{\widehat{a}_{lig(i)_k}^{j_k}} \mid \widehat{a}_{lig(i)_k}^{j_k} > 0 \right\}.$$

Les points entiers se trouvant sur l'arête E_{j_k} sont identifiés de telle sorte que θ_{j_k} soit entier et $\theta_{j_k} \times \widehat{a}_{lig(i)_k}^{j_k}$ entier $\forall i \in B_k$.

Remarque 3.3.2 Si x^0 , solution optimale du problème (P_1) , n'est pas unique alors il existe une autre solution $x^1 \neq x^0$ tel que $Z_1(x^1) = Z_1(x^0)$. On dit alors que x^1 est une solution optimale alternative de x^0 .

La relation précédente permet de déterminer ces solutions alternatives.

On note par nb_{j_k} le nombre de solutions entières se trouvant sur l'arête E_{j_k} y compris x_k .

Notons par $SND(P)$ l'ensemble des solutions potentiellement non dominées générées jusqu'à l'étape k , $k \geq 1$.

Algorithme 3 : ABBAS & MOULAI

Étape 1 : Résoudre le problème (P_1) définit par :

$$(P_1) : \max\{Z_1 = c^1, x \in D\}$$

et trouver la solution optimale entière x^1 sur S_1 . Construire l'ensemble Ω_1 .

Étape 2 : Tester l'ensemble Ω_1 .

Cas 1 : $\Omega = \emptyset$, alors x^1 est l'unique solution optimale sur S_1 .

Soit $(z_1^1, z_1^2, \dots, z_1^r)$ le vecteur critère correspondant, il est enregistré dans $SND(P)$ comme étant le premier r-uplet non dominé.

Tronquez le point x^1 par la coupe de Dantzig :

$$\sum_{j \in N_1} x_j \geq 1.$$

et par application de la méthode dual de simplexe et des coupes successives de Gomory si nécessaire, on obtient une solution entière, soit x^2 dans la région tronquée S_2 . Mettre à jour $SND(P)$.

Cas 2 : Si $\Omega \neq \emptyset$, alors choisir un indice quelconque $j_1 \in \Omega_1$ et calculer le nombre θ de l'opération pivot.

si $\theta \geq 1$, alors déterminer toute les solutions entière alternatives à x^1 , y_1^q , $q = \overline{2, nb_{j_1}}$, le long de l'arête E^{j_1} et mettre à jour $SND(P)$.

Comme les solutions alternatives ont la même valeur de z^1 que celle de x^1 , le premier point potentiellement non dominé est choisit comme le r-uplet ayant la plus grande valeur de z^2 , sinon choisir celui qui a la plus grande valeur de z^3 et ainsi de suite jusqu'à l'obtention du premier r-uplet potentiellement non dominé.

Tronquer l'aête E^{j_1} par la coupe :

$$\sum_{j \in N_1 \setminus \{j_1\}} x_j \geq 1.$$

L'algorithme dual du simplexe et des coupes successive de Gomory éventuelles, permettant d'obtenir une solution entière x^2 dans la région tronquée S_2 . Mettre à jour $SND(P)$.

Si pour tous $j_1 \in \Omega_1$, on a $\theta < 1$, alors choisir un indice quelconque $j_1 \in \Omega_1$ et appliquer la coupe : $\sum_{j \in N_1 \setminus \{j_1\}} x_j \geq 1$.

De la même manière (appliquer la méthode dual du simplexe et des coupes de Gomory éventuelles), on obtient une solution entière x^2 dans la région tronquée S_2 . Mettre à jour $SND(P)$.

Étape k : ($k \geq 3$) Choisir un indice $j_{k-1} \in \Gamma_{k-1}$, déterminer toutes les solutions entières y_{k-1}^q , $q = \overline{2, nb_{j_{k-1}}}$ alternatives à x^{k-1} se trouvant sur l'arête $E^{j_{k-1}}$, lorsqu'elles existent et mettre à jour l'ensemble $SND(P)$.

Tronquer l'arête $E^{j_{k-1}}$ par la coupe :

$$\sum_{j \in N_{k-1} \setminus \{j_{k-1}\}} x_j \geq 1.$$

et chercher de nouveau une solution entière dans la région tronquée S_k , soit x^k . Mettre à jour l'ensemble $SND(P)$.

Après application de la méthode dual du simplexe et éventuellement, des coupes successives de Gomory, la solution optimale entière obtenue sur la région S_k sera x^k . Ceci marque le début de l'étape $k + 1$.

Étape finale : Le processus se termine quand l'impossibilité de l'opération pivot de la méthode dual du simplexe apparaît, indiquant que la région courante ne contient

aucun point entier et que l'ensemble des solutions efficaces est complètement déterminé.

3.3.4 Méthode de M.Abbas et D.Chaabane

M. Abbas et D. Chaabane ont proposé dans [2] la méthode de détermination des solutions efficaces dans l'espace des variables discrètes, cette méthode est une forme améliorée de la méthode de Gupta et Malhotra où le test d'arrêt est corrigé pour déterminer toutes les solutions efficaces du problème (P) sans n'en manquer aucune. Cette méthode utilise les mêmes paramètres que ceux définis dans la méthode précédente.

Notons par $SND(P)$ l'ensemble des solutions potentiellement non dominées générées jusqu'à l'étape k ; $k \geq 1$.

Algorithme 4 : ABBAS & CHAABANE

Étape 1 : Résoudre le problème (P_1).

Soit x^1 la solution optimale entière x^1 et $(z_1^1, z_1^2, \dots, z_1^r)$ le vecteur critère correspondant. Construire l'ensemble Ω_1 .

Si $\Omega_1 = \emptyset$, **alors** la solution optimale est unique $SND(P) := \{(z_1^1, z_1^2, \dots, z_1^r)\}$.

Aller à l'étape 2.

Sinon, la solution optimale peut ne pas être unique. Pour chaque $j \in \Omega_1$, calculer θ .

Si $\forall j \in \Omega_1$, on a $\theta < 1$, **alors** il n'y a pas de solution alternative à x^1 le long de l'arête E^j , $j \in \Omega_1$.

$SND(P) := \{(z_1^1, z_1^2, \dots, z_1^r)\}$. Aller à l'étape 2.

Sinon,

tant que il existe au moins un $j \in \Omega_1$, tel que $\theta \geq 1$, **faire** :

- Explorer l'arête E^j ,
- Évaluer en chacune des solutions entière trouvées les r critères,
- Mettre à jour $SND(P)$.

Choisir arbitrairement un $j \in \Omega_1$, initialiser k à 1 et aller à l'étape 2.2 .

Étape 2 : ($k = 1$)

Étape 2.1 : construire l'ensemble Ψ_k .

Si $\Psi_k = \emptyset$, **alors** aller à l'étape 2.2 (la coupe devient une coupe de Dantzig : $\sum_{j \in N} x_j \geq 1$).

Sinon , poser $\psi = \Psi_k$. Aller à l'étape (a).

Étape (a) : Choisir un indice $j_k \in \psi$ et calculer le nombre θ .

Si $\theta < 1$, **alors** il n'y a aucune solution entière sur l'arête E^{j_k} .

$\psi := \psi \setminus \{j_k\}$.

Si $\psi = \emptyset$, **alors** choisir un $j_k \in \Psi_k$ et aller à l'étape 2.2.

Sinon , aller à l'étape (a).

Sinon , déterminer les solutions entières sur E^{j_k} et évaluer en chacune d'elles les r critères.

Mettre à jours l'ensemble $SND(P)$. Aller à l'étape 2.2.

Étape 2.2 : Utiliser la coupe $\sum_{j \in N_k \setminus \{j_k\}} x_j \geq 1$. pour réduire le domaine de recherche et par application des méthodes dual du simplexe et les coupe de Gomory si nécessaire, on obtient x^{k+1} comme étant la solution optimale du problème augmenté.

Mettre à jour $SND(P)$, $k := k + 1$ et aller à l'étape 2.1.

Test d'arrêt : La procédure prend fin quand l'opération pivot est impossible, le problème est devenu non réalisable dans la nouvelle région tronquée.

Remarque 3.3.3 *Aussi bien dans l'algorithme ABBAS & MOULAI que dans l'algorithme ABBAS & CHAABANE, on visent à énumérer l'ensemble de toutes les solutions réalisables du problème étudié. De plus, pour la recherche des solutions entières, ces deux algorithmes utilisent les coupes fractionnaires de Gomory qui convergent très lentement.*

3.3.5 Méthode A.Sylva et J.Crema

La méthode développée par Crema et Sylva [22] est une variante de celle de Klein et Hannan étudiée précédemment. Son principe repose sur la résolution d'une succession de programmes linéaires en nombres entiers optimisant à chaque étape une combinaison positive des critères. Un ensemble de contraintes est rajouté à chaque fois assurant la détection d'une nouvelle solution efficace. A la fin, la méthode fournit l'ensemble de toutes les solutions non dominées du problème de programmation linéaire discrète à objectifs multiples.

Les auteurs considèrent que les données du problème (P) sont entières, aussi bien la matrice des contraintes A , le vecteur second membre b que les coefficients c_j^i de tous les critères.

Algorithme 5 : SYLVA & CREMA

Étape1 : Après avoir fixé le vecteur poids $(\lambda_1, \lambda_2, \dots, \lambda_r)$ à des valeurs strictement positives, la première étape consiste à résoudre le problème (P_1) défini comme suit :

$$(P_1) : \max \left\{ \sum_{i=1}^r \lambda_i c^i x, x \in D \right\}$$

Si (P_1), n'admet pas solution, **alors** (P) l'est aussi.

Sinon , Une solution x_1 est trouvée et elle est efficace.

Ensuite, une suite de programmes linéaires en nombres entiers augmentés par certaines contraintes sont résolus progressivement.

Après k étapes du processus :

Si (P_k), n'admet pas solution, **alors** l'algorithme s'arrête.

Sinon , une nouvelle solution efficace x_k est trouvée et le nouveau problème (P_{k+1}) est défini à partir de (P_k) en éliminant toutes les solutions vérifiant $c^i x \leq c^i x^k, \forall i = 1, \dots, r$ ceci peut être traduit par le rajout de contraintes suivantes :

$$\begin{cases} C^i x \geq (C^i x^k + f_i) y_i^k - M_i (1 - y_i^k) & i = 1, \dots, r \\ \sum_{i=1}^r y_i^k = 1. & y_i^k \geq 0 \quad i = 1, \dots, r. \end{cases}$$

où $-M_i$ est un minorant pour toute valeur réalisable de la $i^{\text{ème}}$ fonction objectif et $f_i \geq 1$; entier, représente la plus petite augmentation possible du $i^{\text{ème}}$ critère.

Étape générale k : A l'étape k , on résout le problème (P_k) qui est défini comme suit :

$$(P_k) \begin{cases} \max \sum_{i=1}^r \lambda_i c^i x \\ sc \\ x \in D \\ C^i x \geq (C^i x^j + f_i) y_i^j - M_i (1 - y_i^j), \quad i = \overline{1, r}; \quad j = \overline{1, k-1} \\ \sum_{i=1}^r y_i^j = 1. \quad i = \overline{1, r}, \quad j = \overline{1, k-1} \\ y_i^j \in \{0, 1\} \quad i = \overline{1, r}, \quad j = \overline{1, k-1}. \end{cases}$$

Étape finale n : La procédure s'arrête lorsque le problème (P_n) est irréalisable.

Pour $f_i = 1, \forall i = 1, \dots, r$, la méthode génère toutes les solutions non dominées. Lorsque $f_i > 1$; elle prend une valeur permettant d'atteindre la valeur minimale souhaitait par le décideur pour le $i^{\text{ème}}$ critère. Dans ce cas, seul un sous ensemble de solutions non dominées est trouvé.

3.4 Conclusion

Ce chapitre est consacré à l'introduction de la problématique multi-objectif en nombres entiers. Les définitions et les concepts de base sont présentés ainsi que les principaux résultats nécessaires à la résolution d'un problème MOILP. Nous avons décrit quelques méthodes exactes de résolution de MOILP. Ces méthodes sont passées en revue et les avantages et les inconvénients de chacune sont mentionnées.

Chapitre 4

Optimisation d'un critère sur un Ensemble Efficient

4.1 Introduction

Dans certaines situations pratiques, l'énumération de tout l'ensemble des solutions efficaces d'un problème multi-objectif n'est pas toujours recommandée car il peut s'avérer que cet ensemble efficient soit très grand et il devient impossible pour le décideur de choisir le meilleur compromis en termes de ses préférences. L'optimisation d'un critère, qui exprime les préférences du décideur, sur l'ensemble efficient constitue, dès lors, un sujet de recherche essentiel dans ce domaine.

Ce problème est en principe difficile à résoudre, ceci est dû principalement à la non-convexité de son ensemble réalisable.

Ce type de problème a été étudié la première fois dans le cas continu en 1972 par Philip [62] et depuis plusieurs chercheurs, citons en particulier : Benson [10], [11], [12], [13], [14], Isermann [45], Yamamoto [81], Ecker et Song [30], Sayin [68], motivés par de nombreuses applications, se sont intéressés à l'optimisation d'une fonction sur l'ensemble efficient d'un problème multi-objectif linéaire.

Contrairement au cas continu qui a été largement étudié par de nombreux auteurs, le cas discret n'a pas vu autant de développement semblable et on ne trouve que très peu d'articles dédiés au problème à variables entières.

La première méthode pour l'optimisation d'un critère linéaire sur l'ensemble efficace discret fut proposée en 1992 par Nguyen [59], cette méthode consiste à calculer une borne supérieure de la valeur optimale de la fonction objectif. En 2006, un algorithme évitant l'énumération explicite de tous les points efficaces dans l'espace des variables de

décision, est proposé par Abbas et al. [4], où différents types de coupes sont imposées de telle manière que l'amélioration de la valeur optimale de la fonction objectif soit garantie à chaque itération. En 2008, Jorge [46] développe un algorithme basé sur l'analyse d'un ordre approprié de problèmes linéaires en nombres entiers pour éliminer successivement les solutions moins bonnes sur le critère principal. Récemment, en 2010, Chaabane et al. [15] proposent une méthode de résolution dans l'espace des critères dans laquelle la valeur de la fonction objectif principal est améliorée en optimisant une somme pondérée des critères à chaque itération.

4.2 Formulation du problème

Le problème d'optimisation d'une fonction, supposée connue, sur l'ensemble efficient d'un problèmes multi-objectifs linéaire peut être formulé d'une manière générale par :

$$(P_E) \begin{cases} \max \phi(x) \\ sc. \\ x \in Eff \end{cases}$$

où ϕ est une fonction continue de \mathbb{R}^n dans \mathbb{R} , et Eff représente l'ensemble efficient du problème multi-objectif linéaire suivant :

$$(P_{MO}) \begin{cases} \max Cx \\ sc. \\ x \in S = \{x \in \mathbb{R}^n \mid Ax \leq b, x \geq 0\} \end{cases}$$

avec $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$ et $C \in \mathbb{R}^{p \times n}$. L'ensemble S est supposé borné non vide.

4.3 Méthode de résolution de (P_E) dans le cas discret

Pour le cas discret, on y trouve que peu d'article dédiés au problèmes (P_E) à variables entières : [59], [4], [46], [15]. Tous ces articles traitent le cas (*Linéaire-Linéaire*) c'est à dire optimiser un critère linéaire sur l'ensemble efficient d'un problème multi-objectif linéaire discret *MOILP*, qui peut être formulé d'une manière par :

$$(P_E) \begin{cases} \max \phi(x) = dx \\ sc. \\ x \in Eff \end{cases}$$

où $d \in \mathbb{R}^n$ et Eff représente l'ensemble efficient du problème multi-objectif linéaire à variables entières suivant :

$$(P_{MO}) \begin{cases} \max Z_i(x) = c^i x, & i \in \{1, \dots, p\} \\ sc. \\ x \in D \end{cases}$$

où $D = S \cap \mathbb{Z}^n$, $S = \{x \in \mathbb{R}^n \mid Ax \leq b, x \geq 0\}$. $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c^i \in \mathbb{R}^n$. D supposé non vide et S un polyèdre borné.

On définit (P_R) le problème relaxé suivant :

$$(P_R) \begin{cases} \max \phi(x) = dx \\ sc. \\ x \in D. \end{cases}$$

avec $D = S \cap \mathbb{Z}^n$, $S = \{x \in \mathbb{R}^n \mid Ax \leq b, x \geq 0\}$.

Théorème 4.3.1 (Caractérisation d'un solution efficace [29]) *Soit x^* une solution quelconque de D . x^* est efficace pour le problème (P_{MO}) si et seulement si la valeur optimale de la fonction objectif θ est nulle dans le programme de programmation linéaire mixte suivant :*

$$EK(x^*) \begin{cases} \max \theta = \sum_{i=1}^{i=p} \psi_i \\ sc. \\ Cx - I\psi = Cx^*. \\ x \in D; \psi_i \in \mathbb{R}^+, \forall i \in \{1, \dots, p\} \end{cases}$$

où C est la matrice d'ordre $p \times n$ dont la $i^{\text{ème}}$ ligne correspond à c^i , $i \in \{1, \dots, p\}$; I est la matrice identité d'ordre p et $\psi = (\psi_i)_{i \in \{1, \dots, p\}}$.

Cette caractérisation des solutions efficaces servira comme test d'efficacité dans les algorithmes qui suivent.

4.3.1 Méthode de Jorge

Jorge a proposé dans [46] un algorithme qui fournit une solution optimale du problème (P_E) en un nombre fini d'itérations, sans avoir à déterminer toutes les solutions efficaces du MOILP.

La procédure commence à résoudre le problème relaxé (P_R) . Évidemment, seulement dans un nombre réduit de cas spéciaux la solution optimale de (P_R) fournit une solution

optimale de (P_E) . Donc, si ce n'était pas le cas, une nouvelle solution efficace qui domine la précédente est alors obtenue. Ensuite, dans chaque itération, le critère principal est optimisé sur le domaine restreint par des contraintes en nombres entiers qui sont incluses progressivement pour éliminer les solutions dominées par la solution efficace courante, afin de fournir une solution non dominée par les solutions détectées antérieurement jusqu'à ce qu'une solution optimale soit finalement trouvée.

Algorithme 6 : JORGE

Étape 0 : (Initialisation) Poser $\phi_{inf} = -\infty$, $\phi_{sup} = +\infty$, $l = 1$ et résoudre le problème relaxé (P_R) .

Si (P_R) est irréalisable, **Stop**. (P_E) est aussi irréalisable.

Sinon, soit x^l une solution optimale de (P_R) .

Étape 1 : Si x^l est efficace (test d'efficacité 4.3.1), **Stop**. $x_{opt} = x^l$ est une solution optimale de (P_E) et $\phi_{opt} = dx^l$.

Sinon, poser $\phi_{sup} = dx^l$ et aller à l'étape 2.

Étape 2 : Trouver $\hat{x}^l \in \text{Eff}(P)$ dont le vecteur critère domine Cx^l et soit \tilde{x}^l une solution optimale du problème (T_l) suivant :

$$\max\{dx \mid Cx = C\hat{x}^l, x \in D\}.$$

Dans l'espace des critères, plusieurs solutions efficaces peuvent avoir le même vecteur critères, pour cela le problème (T_l) est résolu pour optimiser le critère principal sur toutes les solutions équivalentes à \hat{x}^l .

Si $d\tilde{x}^l > \phi_{inf}$, poser $\phi_{inf} = d\tilde{x}^l$ et $x_{opt} = \tilde{x}^l$.

Si $\phi_{inf} = \phi_{sup}$, **Stop**. x_{opt} est une solution optimale de (P_E) .

Étape 3 : Résoudre le problème (R_l) suivant :

$$\max\{dx/x \in D - \cup_{s=1}^l D_s\}, \text{ où } D_s\{x \in \mathbb{Z}^n / C\tilde{x}^s \geq Cx\}.$$

avec $\tilde{x}^1, \tilde{x}^2, \dots, \tilde{x}^l$ sont les solution optimales des problèmes $(T_1), (T_2), \dots, (T_l)$ respectivement

Si (R_l) est irréalisable, **Stop**. x_{opt} est une solution optimale du problème (P_E) .

Sinon, soit x^{l+1} une solution optimale de (R_l) .

Si $dx^{l+1} \leq \phi_{inf}$, **Stop**. x_{opt} est une solution optimale de (P_E) .

Sinon, poser $l = l + 1$ et aller à l'étape 1.

Proposition 4.3.1 [46] Soit x^{l+1} une solution optimale du problème (R_l) telle que $\phi(x^{l+1}) > \max_{s \in \{1, \dots, l\}} \{\phi(\tilde{x}^s)\}$. Si $x^{l+1} \in Eff$ alors x^{l+1} est une solution optimale de (P_E) .

Proposition 4.3.2 [46] Soient $\tilde{x}^1, \dots, \tilde{x}^l \in Eff$, si (R_l) est irréalisable, alors l'ensemble de toutes les solutions non dominées est $SND(P_{MO}) = C\tilde{x}^1, \dots, C\tilde{x}^l$.

4.3.2 Méthode de Chaabane et al.

Les auteurs dans [15] introduisent une nouvelle méthode exacte pour la résolution du problème (P_E) dans l'espace des critères en un nombre fini d'itérations. Dans cette méthode, la valeur de la fonction objectif ϕ est améliorée à chaque itération en optimisant une somme pondérée des critères.

Dans cette méthode tous les coefficients de A , b , C et d sont supposés entiers.

Algorithme 7 : CHAABANE ET AL.

Étape 1 : Résoudre le problème relaxé (P_R) , soit x^* la solution obtenue.

Si x^* est efficace, **Stop**. x^* est une solution optimale pour (P_E) .

Sinon, aller à l'étape 2.

Étape 2 : Poser $k = 1$, $H^0 = D$, $Eff = \emptyset$. Résoudre le problème (P_λ^0) suivant :

$$(P_\lambda^0) \begin{cases} \max Z_\lambda(x) = \sum_{i=1}^p \lambda_i Z_i(x) \\ sc. \\ x \in H^0. \end{cases}$$

Sa solution x_0 étant efficace. On pose $x_{opt} = x_0$, $\phi_{opt} = dx_0$ et $Eff = Eff \cup x_0$.

Aller à l'étape 3.

Étape 3 : Poser $k = k + 1$ et résoudre le problème suivant :

$$(P_\lambda^k) \begin{cases} \max Z_\lambda(x) = \sum_{i=1}^p \lambda_i Z_i(x) \\ sc. \\ x \in D^k. \end{cases}$$

où $D^k = H^k \cap \{x \in D / dx \geq dx_{opt} + 1\}$ et

$$H^k = H^{k-1} \cap \left\{ \begin{array}{l} x \in D \\ Z_i(x) \geq (Z_i(x_{opt}) + 1)y_i^k - M_i(1 - y_i^k); \quad i \in \{1, \dots, p\} \\ \text{avec } \sum_{i=1}^p y_i^k = 1; \quad y_i^k \in \{0, 1\}; \quad i \in \{1, \dots, p\} \end{array} \right\}$$

où $-M_i$ est une borne inférieure de la $i^{\text{ème}}$ fonction objectif dans S .

y_i^k est une variable binaire associée à chaque critère et définie par :

$$y_i^k = \begin{cases} 1 & \text{si } Z_i \text{ est strictement amélioré par rapport à } Z_i(x_{opt}) \\ 0 & \text{sinon} \end{cases}$$

La contrainte $\sum_{i=1}^p y_i^k \geq 1$ signifie qu'au moins un des critère est amélioré.

La contrainte supplémentaire $dx \geq dx_{opt} + 1$ permet de se déplacer vers une autre solution meilleure sur ϕ si elle existe.

Si $D^k = \emptyset$, aller à l'étape 6.

Sinon , soit x_k la solution optimale de (P_λ^k) .

Si x^k est efficace, **alors** $Eff = Eff \cup x_k$, $x_{opt} = x_k$, $\phi_{opt} = dx_{opt}$ et aller à l'étape 3.

Sinon aller à l'étape 4.

Étape 4 : On explore toutes les arêtes E_{j_k} incidentes à x_k . Soit $J_k = \{j \in N_k / Z_\lambda, j - c_j = 0\}$.

Si $J_k \neq \emptyset$, **alors** poser $\gamma = J_k$ et aller à l'étape 4.1.

Sinon Aller à l'étape 5.

Étape 4.1 **Si** $\gamma = \emptyset$ **alors**, soit $j_k \in J_k$. aller à l'étape 5.

Sinon , prendre $j_k \in J_k$ et calculer $\theta_{j_k}^0$ (voir définition 3.3.1 formule 3.2).

Si $\theta_{j_k}^0 = 0$, poser $\gamma = \gamma \setminus j_k$ et aller à l'étape 4.1.

Sinon $\theta_{j_k}^0 \geq 1$, aller à l'étape 4.2.

Étape 4.2 **Si** il existe une solution efficace entière x'_k sur l'arête E_{j_k} telle que $dx'_k > \phi_{opt}$, poser $Eff = Eff \cup \{x'_k\}$, $x_{opt} = x'_k$ et $\phi_{opt} = dx_{opt}$. Aller à l'étape 3.

Sinon poser $\gamma = \gamma \setminus \{j_k\}$. Aller à l'étape 4.1.

Étape 5 : Soit $k = k + 1$, rajouter la coupe de Dantzig à (P_λ^k) et appliquer la méthode dual du simplexe pour obtenir une nouvelle solution optimale x_k .

Si x_k est efficace et $dx_k > \phi_{opt}$, on pose $Eff = Eff \cup \{x_k\}$, $x_{opt} = x_k$ et $\phi_{opt} = dx_{opt}$. Aller à l'étape 3.

Sinon , aller à l'étape 4.

Étape 6 (étape finale) : La solution optimale de (P_E) est donc x_{opt} , sa valeur objectif correspondante est ϕ_{opt} . Le sous ensemble de $E(P)$ des solutions efficaces obtenues qui améliorent la fonction ϕ est Eff .

4.4 Conclusion

Dans ce chapitre, nous avons présenté deux méthodes de résolution du problème (P_E) où les variables de décision sont entières.

La plupart des méthodes développées dans la littérature traitent le cas (*Linéaire-Linéaire*) c'est à dire optimiser un critère linéaire sur un ensemble efficient d'un problème multi-objectif linéaire.

Très peu d'articles récents ont traité des cas plus général à savoir le cas (*Linéaire-Fractionnaire*) [83], qui optimise un critère linéaire sur un ensemble efficient d'un problème multi-objectif fractionnaire linéaire, le cas (*Fractionnaire-Fractionnaire*) [84], optimisant une fonction fractionnaire linéaire sur un ensemble efficient d'un problème multi-objectif fractionnaire linéaire.

Vu la pauvreté de la littérature relatant le problème (P_E) dans le cas discret, ceci, nous a davantage motivé pour explorer cette classe de problèmes en mettant au point une méthode exacte pour la du cas (*Quadratique-Linéaire*), c'est à dire l'optimisation d'un critère quadratique sur un ensemble efficient d'un problème multi-objectif linéaire en nombres entiers.

Chapitre 5

Optimisation d'un Critère Quadratique sur l'Ensemble Efficient d'un Problème Multi-objectif Linéaire Discret

5.1 Introduction

Le problème sous considération (P) est un problème de maximisation d'un critère quadratique notée $\phi(x)$ sur l'ensemble efficient Eff dit aussi ensemble Pareto optimal d'un problème de maximisation multi-objectif linéaire à variables entières $MOILP$.

Ce problème est en principe très difficile à résoudre, ceci est dû principalement à la non convexité de son ensemble réalisable, ajouter à cela la forme implicite du critère principale qui est non linéaire.

Pour résoudre ce problème, nous avons pris en considération deux formes différentes d'une fonction quadratique :

- La fonction quadratique indéfinie écrite sous forme d'un produit de deux fonctions linéaires positives qui est une fonction quasi-concave, le critère principale $\phi(x)$ s'écrit alors sous la forme $\phi(x) = (p^t x + \alpha)(q^t x + \beta)$.
- La fonction quadratique semi-définie écrite sous sa forme générale $\frac{1}{2}x^t G x + g^t x$ avec G une matrice semi-définie négative, ainsi le critère principale $\phi(x) = \frac{1}{2}x^t G x + g^t x$ est concave.

les notations, définitions et les résultats de base utilisés dans notre travail sont présentés dans la deuxième section de ce chapitre.

La troisième section est consacrée à la fonction quadratique indéfinie ; le problème considéré (P) consistera alors à maximiser un critère quadratique sur l'ensemble efficient d'un problème (*MOILP*), avec $\phi(x)$ représentant le critère principale du problème, est le produit de deux fonctions linéaires positives.

Nous proposons pour cela deux méthodes exactes différentes ; la première est une adaptation de de la méthode de Jorge[46] détaillé dans le chapitre 4 précédent , et la deuxième est une nouvelle méthode basé sur le principe de séparation et évaluation "Branch & Bound" et l'utilisation d'une coupe efficace à chaque itération de l'algorithme.

La quatrième section sera attribuée à l'optimisation d'un critère quadratique, écrit sous la forme $\phi(x) = \frac{1}{2}x^t Gx + g^t x$ avec G une matrice semi-définie négative, sur l'ensemble efficient d'un problème *MOILP* ; nous proposons pour cela un algorithme exacte qui est une adaptation de l'algorithme de Jorge[46] au cas d'un critère quadratique.

La cinquième section va conclure le chapitre.

5.2 Optimisation d'un critère quadratique indéfini sur l'ensemble efficient d'un problème *MOILP*

Le problème principale que nous allons étudier est formulé par :

$$(P) \begin{cases} \max \phi(x) = (p^t x + \alpha)(q^t x + \beta) \\ sc. \\ x \in Eff \end{cases}$$

où p, q et x des vecteurs de \mathbb{R}^n et $\alpha, \beta \in \mathbb{R}$.

$\phi(x)$ une fonction d'utilité quadratique représentant les préférences du décideur et Eff représente l'ensemble efficient du problème multi-objectif linéaire discret (*MOILP*) définit au chapitre 3 de la thèse par :

$$(P_{MO}) \begin{cases} \max Z_i(x) = c^i x, \quad i \in \{1, \dots, r\} \\ sc. \\ x \in D = S \cap \mathbb{Z}^n \end{cases}$$

où $r \geq 2$ est un nombre entier, $S = \{x \in \mathbb{R}^n / Ax \leq b, x \geq 0\}$; $c^i \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, D étant supposé l'ensemble non vide des points entiers de S et S un polyèdre convexe borné de \mathbb{R}^n .

On suppose que les facteurs $(p^t x + \alpha)$ et $(q^t x + \beta) > 0 \forall x \in S$.

Dans cette section, nous proposons deux algorithmes exacts pour résoudre le problème (P) , les deux algorithmes fournissent une solution optimale de (P) sans avoir à énumérer l'ensemble de toutes les solutions efficaces du problème multi-objectif (P_{MO}) . La première méthode est une adaptation de de la méthode de Jorge[46] au cas d'une fonction d'utilité quadratique, et la deuxième méthode est une nouvelle méthode basé sur le principe de séparation et évaluation "Branch & Bound" et l'utilisation d'une coupe efficace à chaque itération de l'algorithme.

5.2.1 Première approche

La première méthode que nous proposons pour générer la solution optimale du problème centrale (P) est inespérée de la méthode de Jorge [46], cité dans le chapitre 4. Cette méthode fournit une solution optimale globale du problème (P) sans passer par toutes les solutions efficaces du (P_{MO}) .

L'approche adoptée dans ce travail est basé sur la résolution d'un programme noté R_l à chaque itération l de l'algorithme, ce programme optimise un critère quadratique sur un domaine restreint par des contraintes linéaire supplémentaires imposées au programme R_{l-1} résolu précédemment. ces nouvelles contraintes, connues dans la littérature sous le nom "Corner Constraints" D.Klein & E.Hannan [49], sont incluses progressivement pour éliminer les solutions efficaces déjà trouvées, et faire en sorte que les nouvelles solutions générées soient efficaces.

Description de la méthode

On initialise l à 1 représentant le nombre d'itération.

L'algorithme commence par la résolution du programme relaxé (P_R) suivant :

$$(P_R) \begin{cases} \max \phi(x) = (p^t x + \alpha)(q^t x + \beta) = f_1(x).f_2(x) \\ sc. \\ x \in D. \end{cases}$$

avec $D = S \cap \mathbb{Z}^n$, $S = \{x \in \mathbb{R}^n \mid Ax \leq b, x \geq 0\}$.

On fait appel à une des méthodes existante en littérature pour résoudre le problème (P_R) .

À chaque itération l de l'algorithme, l'étape l consiste à tester l'efficacité de la solution x^l précédemment trouvée en appliquant un des théorèmes cités précédemment 4.3.1,

2.1.4, 2.1.3, 2.1.2 ces tests permettent de vérifier l'efficacité de la solution x^l et nous permettent d'avoir une autre solution \hat{x}^l qui est efficace si x^l ne l'est pas.

Si la solution x^l est efficace, l'algorithme s'arrête, sinon, on passe à l'étape 2.

À l'étape 2, on résout le problème (T_l) suivant :

$$\max\{\phi(x) = (p^t x + \alpha)(q^t x + \beta) \mid Cx = C\hat{x}^l, x \in D\}.$$

Ce problème permet d'optimiser le critère principale $\phi(x)$ sur toutes les solutions alternatives à \hat{x}^l . Soit \tilde{x}^l la solution optimale du problème.

Notons que pour ce problème, au moins une solution réalisable initiale est disponible, à savoir \hat{x}^l , qui peut être optimale.

L'étape 3 de l'algorithme vise à fournir une nouvelle solution entière non générée précédemment et qui n'est pas dominée par aucune solution efficace déjà trouvée. Cette tâche est assurée par la résolution du problème (R_l) suivant :

$$\max\{\phi(x) = (p^t x + \alpha)(q^t x + \beta) \mid x \in D - \cup_{s=1}^l D_s\}$$

où $D_s = \{x \in \mathbb{Z}^n \mid C\tilde{x}^s \geq Cx\}$, avec $\tilde{x}^1, \tilde{x}^2, \dots, \tilde{x}^l$ sont les solutions optimales des problèmes $(T_1), (T_2), \dots, (T_l)$ respectivement.

Il est possible de fournir une solution optimale au problème (R_l) , en résolvant le problème \hat{R}_l qui lui est équivalent :

$$(\hat{R}_l) \begin{cases} \max \phi(x) = (p^t x + \alpha)(q^t x + \beta) \\ x \in D \\ c^i x > (c^i \tilde{x}^l + 1)y_i^l - M_i(1 - y_i^l), \quad i = 1, \dots, r \\ \sum_{i=1}^r y_i^l = 1 \\ y_i^l \geq 0, \quad i = 1, \dots, r. \end{cases}$$

où $-M_i$ est un minorant pour toute valeur réalisable du $i^{\text{ème}}$ critère.

Notons que lorsque $y_i^l = 0$ la contrainte associée est redondante et lorsque $y_i^l = 1$, une stricte amélioration du $i^{\text{ème}}$ critère est imposée.

On note x^{l+1} la solution obtenue en résolvant (R_l) , on pose $l = l + 1$ et on passe à l'étape l .

Développement de l'algorithme

Algorithme 8

Étape 0 : (Initialisation) Poser $\phi_{inf} = -\infty$, $\phi_{sup} = +\infty$, $l = 1$ et résoudre le problème relaxé (P_R) en utilisant une des méthodes de résolution d'un programme quadratique indéfini existantes en littérature.

Si (P_R) est irréalisable, **Stop**. (P) est aussi irréalisable.

Sinon, soit x^l une solution optimale de (P_R).

Étape 1 : Si x^l est efficace, **Stop**. $x_{opt} = x^l$ est une solution optimale de (P_E) et $\phi_{opt} = \phi(x^l)$.

Sinon, poser $\phi_{sup} = \phi(x^l)$ et aller à l'étape 2.

Étape 2 : Soit $\hat{x}^l \in E_{ff}$ efficace fournie par le test d'efficacité et soit \tilde{x}^l une solution optimale du problème (T_l) suivant :

$$\max\{\phi(x) = (p^t x + \alpha)(q^t x + \beta) \mid Cx = C\hat{x}^l, x \in D\}.$$

Résoudre le problème (T_l) pour optimiser le critère principal sur toutes les solutions équivalentes à \hat{x}^l .

Si $\phi(\tilde{x}^l) > \phi_{inf}$, poser $\phi_{inf} = \phi(\tilde{x}^l)$ et $x_{opt} = \tilde{x}^l$.

Si $\phi_{inf} = \phi_{sup}$, **Stop**. x_{opt} est une solution optimale de (P_E).

Étape 3 : Résoudre le problème (R_l) suivant :

$$\max\{\phi(x) = (p^t x + \alpha)(q^t x + \beta) \mid x \in D - \cup_{s=1}^l D_s\}$$

où $D_s = \{x \in \mathbb{Z}^n \mid C\tilde{x}^s \geq Cx\}$, avec $\tilde{x}^1, \tilde{x}^2, \dots, \tilde{x}^l$ sont les solution optimales des problèmes (T_1), (T_2), ..., (T_l) respectivement

Si (R_l) est irréalisable, **Stop**. x_{opt} est une solution optimale du problème (P_E).

Sinon, soit x^{l+1} une solution optimale de (R_l).

Si $\phi(x^{l+1}) \leq \phi_{inf}$, **Stop**. x_{opt} est une solution optimale de (P_E).

Sinon, poser $l = l + 1$ et aller à l'étape l .

Théorème 5.2.1 *L'algorithme de recherche d'une solution optimale du problème (P) converge en un nombre fini d'itérations*

Preuve 5.2.1 *Par l'hypothèse que la région d'admissibilité S est bornée et D supposé non vide, il existe un nombre limité de solutions entières, ainsi l'ensemble E_{ff} contient un nombre fini de solutions entières efficaces.*

Chaque fois qu'une solution optimale entière x^l du problème (R_l) est trouvée, le test d'efficacité nous permet soit de vérifier l'efficacité de cette solution et donc obtenir une solution optimale à notre problème (P) , soit d'obtenir une autre solution retournées qui est efficace. Donc à chaque itération de l'algorithme, une solution entière efficace est générée, et avec les propositions 4.3.14.3.2, chaque itération permet une amélioration du critère principale et une réduction du domaine de recherche progressivement jusqu'à ce qu'il devient vide et donc la convergence de l'algorithme vers une solution optimale pour le problème (P) est assurée.

Exemple numérique

Pour illustrer l'utilisation de l'algorithme, on considère le problème suivant :

$$(P) \begin{cases} \max \phi(x) = (2x_1 + 4x_2 + 1)(x_1 + x_2 + 2) \\ sc. \\ x \in Eff \end{cases}$$

où Eff est l'ensemble de solutions efficaces du problème linéaire multi-objectif en nombres entiers P_{MO} suivant :

$$(P_{MO}) \begin{cases} \max Z_1 = -3x_1 + x_2 \\ \max Z_2 = x_1 - x_2 \\ sc. \\ 4x_1 + 3x_2 \leq 20 \\ x_1 - x_2 \leq 3 \\ x_2 \leq 4 \\ x_1 \geq 0, x_2 \geq 0 \\ x \text{ entier.} \end{cases}$$

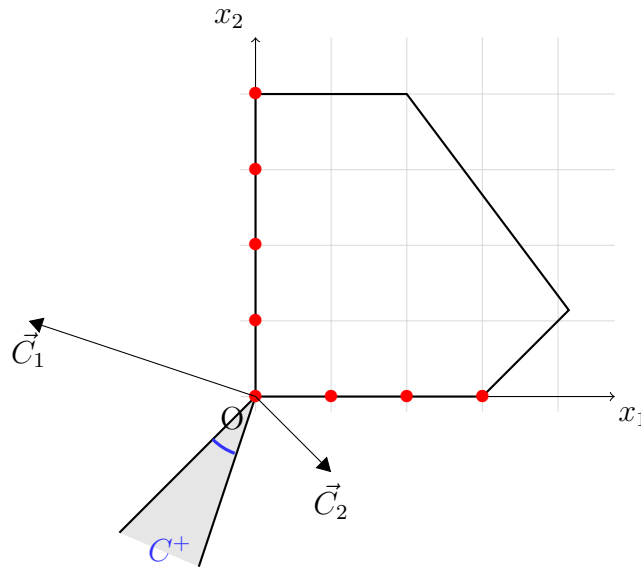


FIGURE 5.1 – L'ensemble des solutions efficients

L'ensemble de toute les solutions efficaces du problème (P_{MO}) est $Eff = \{(0; 0), (0; 1), (0; 2), (0; 3), (0; 4), (1; 0), (2; 0), (3; 0)\}$.

x	(0;0)	(0;1)	(0;2)	(0;3)	(0;4)	(1;0)	(2;0)	(3;0)
$Z(x)$	(0;0)	(1;-1)	(2;-2)	(3;-3)	(4;-4)	(-3;1)	(-6;2)	(-9;3)
$\phi(x)$	2	15	36	65	102	9	20	35

On prend les bornes inférieures des deux fonctions objectifs $-M1 = -11, -M2 = -4$.

Étape 0 : Initialisation soit $\phi_{opt} = -\infty$.

On résout le problème relaxé suivant :

$$(P_R) \begin{cases} \max \phi(x) = (2x_1 + 4x_2 + 1)(x_1 + x_2 + 2) \\ sc. \\ (x_1, x_2) \in D \end{cases}$$

La résolution du problème (P_R) a abouti à la solution entière suivante $x^1 = (2, 4)$ et $\phi(x^1) = 168$.

Étape 1 : On teste l'efficacité de x^1 , on trouve x^1 pas efficace.

On pose $\phi_{sup} = \phi(x^1) = 168$, aller à l'étape 2.

Étape 2 :

Soit $\hat{x}^1 = (0, 2)$ la solution efficace générée par le test d'efficacité. Le vecteur critère correspondant à cette solution est $Z(\hat{x}^1) = (2, -2)$.

Pour trouver les solutions efficaces qui ont le même vecteur critère on résout le problème (T_1) suivant :

$$(T_1) \begin{cases} \max \phi(x) = (2x_1 + 4x_2 + 1)(x_1 + x_2 + 2) \\ sc. \\ (x_1, x_2) \in D \\ -3x_1 + x_2 = 2 \\ x_1 - x_2 = -2 \end{cases}$$

La solution optimale du problème (T_1) est $\tilde{x}^1 = \hat{x}^1 = (0, 2)$ et on a $\phi(\tilde{x}^1) = 36 > \phi_{inf}$, Alors $\phi_{inf} = 36$ et $x_{opt} = (0, 2)$.

On a $\phi_{inf} \neq \phi_{sup}$, aller à l'étape 3.

Étape 3 :

Résoudre le problème (\hat{R}_1) suivant :

$$(\hat{R}_1) \begin{cases} \max \phi(x) = (2x_1 + 4x_2 + 1)(x_1 + x_2 + 2) \\ sc. \\ (x_1, x_2) \in D \\ -3x_1 + x_2 \geq 3y_1^1 - 11(1 - y_1^1) \\ x_1 - x_2 \geq -y_2^1 - 4(1 - y_2^1) \\ y_1^1 + y_2^1 \geq 1, y_1^1, y_2^1 \in \{0, 1\} \end{cases}$$

La solution optimale du problème (\hat{R}_1) est $x^2 = (2, 3)$, $y = (0, 1)$ et $\phi(x^2) = 119$. Comme $\phi(x^2) = 119 > \phi_{inf}$, on pose $l=l+1$.

Étape 1 :

La solution x^2 n'est pas efficace. une autre solution efficace est donnée $\hat{x}^2 = (0, 1)$. Poser $\phi_{sup} = 119$ et aller à l'étape 2.

Étape 2 :

Soit $\hat{x}^1 = (0, 1)$ la solution efficace trouvée. Le vecteur critère correspondant à cette solution est $Z(\hat{x}^1) = (1, -1)$.

On résout le problème (T_2) suivant :

$$(T_2) \begin{cases} \max \phi(x) = (2x_1 + 4x_2 + 1)(x_1 + x_2 + 2) \\ sc. \\ (x_1, x_2) \in D \\ -3x_1 + x_2 = 1 \\ x_1 - x_2 = -1 \end{cases}$$

La solution optimale du problème (T_2) est $\tilde{x}^2 = \hat{x}^2 = (0, 1)$ et on a $\phi(\tilde{x}^2) = 15 < \phi_{inf}$.
 $\phi_{inf} \neq \phi_{sup}$, aller à l'étape 3.

Étape 3 :

Résoudre le problème (R_2) équivalent au problème (\dot{R}_2) :

$$(\dot{R}_2) \begin{cases} \max \phi(x) = (2x_1 + 4x_2 + 1)(x_1 + x_2 + 2) \\ sc. \\ (x_1, x_2) \in D \\ -3x_1 + x_2 \geq 3y_1^1 - 11(1 - y_1^1) \\ x_1 - x_2 \geq -y_2^1 - 4(1 - y_2^1) \\ -3x_1 + x_2 \geq 2y_1^2 - 11(1 - y_1^2) \\ x_1 - x_2 \geq -4(1 - y_2^2) \\ y_1^1 + y_2^1 \geq 1, y_1^1, y_2^1 \in \{0, 1\} \\ y_1^2 + y_2^2 \geq 1, y_1^2, y_2^2 \in \{0, 1\} \end{cases}$$

La solution du problème (\dot{R}_2) est $x^3 = (3, 2)$, $y = (0, 1, 0, 1)$ et $\phi(x^3) = 105$. Comme
 $\phi(x^3) = 105 > \phi_{inf}$, on pose $l=l+1$. Aller à l'étape 1.

Étape 1 :

La solution x^3 est testée pour son efficacité. La solution x^3 n'est pas efficace et la
solution efficace engendrée est $\hat{x}^3 = (1, 0)$.

On pose $\phi_{sup} = \phi(x^3) = 105$. Aller à l'étape 2.

Étape 2 :

Soit $\hat{x}^3 = (1, 0)$ une solution efficace. Le vecteur critère correspondant à cette solution
est $Z(\hat{x}^3) = (-3, 1)$.

On résout le programme (T_3) suivant :

$$(T_3) \begin{cases} \max \phi(x) = (2x_1 + 4x_2 + 1)(x_1 + x_2 + 2) \\ sc. \\ (x_1, x_2) \in D \\ -3x_1 + x_2 = -3 \\ x_1 - x_2 = 1 \end{cases}$$

La solution optimale du problème (T_3) est $\tilde{x}^3 = \hat{x}^3 = (1, 0)$ et on a $\phi(\tilde{x}^3) = 9 < \phi_{inf}$.
 $\phi_{inf} \neq \phi_{sup}$, aller à l'étape 3.

Étape 3 :

Résoudre le problème (R_3) équivalent au problème (\dot{R}_2) suivant :

$$(\dot{R}_3) \begin{cases} \max \phi(x) = (2x_1 + 4x_2 + 1)(x_1 + x_2 + 2) \\ sc. \\ (x_1, x_2) \in D \\ -3x_1 + x_2 \geq 3y_1^1 - 11(1 - y_1^1) \\ x_1 - x_2 \geq -y_2^1 - 4(1 - y_2^1) \\ -3x_1 + x_2 \geq 2y_1^2 - 11(1 - y_1^2) \\ x_1 - x_2 \geq -4(1 - y_2^2) \\ -3x_1 + x_2 \geq -2y_1^3 - 11(1 - y_1^3) \\ x_1 - x_2 \geq 2y_2^3 - 4(1 - y_2^3) \\ y_1^1 + y_2^1 \geq 1, y_1^1, y_2^1 \in \{0, 1\} \\ y_1^2 + y_2^2 \geq 1, y_1^2, y_2^2 \in \{0, 1\} \\ y_1^3 + y_2^3 \geq 1, y_1^3, y_2^3 \in \{0, 1\} \end{cases}$$

La solution du problème (\dot{R}_3) est $x^4 = (0, 4)$, $y = (1, 0, 1, 0, 1, 0)$ et $\phi(x^4) = 102$. Comme
 $\phi(x^4) = 102 > \phi_{inf}$, on pose $l=l+1$. Aller à l'étape 1.

Étape 1 :

La solution $x^4 = (0, 4)$ est efficace (test d'efficacité).

Terminer ; la solution optimale du problème (P) est $x_{opt} = (0, 4)$ avec une valeur optimale $\phi_{opt} = 102$.

5.2.2 Deuxième approche

Le deuxième algorithme que nous proposons fournit une solution optimale du problème principal (P) sans avoir à générer l'ensemble de toutes les solutions efficaces du problème multi-objectif linéaire (P_{MO}) . Notre méthode est principalement basée sur le

processus de séparation et évaluation de "Branch & Bound" et l'utilisation d'une coupe efficace à chaque itération de l'algorithme en mesure de supprimer des solutions entières qui ne sont pas efficaces pour le problème multi-objectif (P_{MO}).

L'approche adoptée pour ce travail pour générer la solution optimale du problème centrale (P) est basée sur la résolution du problème unicritère quadratique (P_l) à chaque étape l ($l \geq 0$) de l'algorithme. Ce problème (P_l) est formulé par :

$$(P_l) \begin{cases} \max \phi(x) = (p^t x + \alpha)(q^t x + \beta) = f_1(x).f_2(x) \\ sc. \\ x \in S_l \end{cases}$$

où les facteurs $(p^t x + \alpha)$ et $(q^t x + \beta) > 0 \forall x \in S$, $p, q \in \mathbb{R}^n$ et $\alpha, \beta \in \mathbb{R}$.
et $S_0 = S$ et sans contrainte d'intégrité des variables.,

Définition 5.2.1 Une coupe est dite efficace pour le problème (P_{MO}) si son adjonction au domaine S ne supprime pas de solutions réalisables entières efficaces de S .

- Soit x_l^* la première solution entière obtenue après résolution du problème (P_l) (pour $l = 0$) en utilisant une des méthodes de résolution du problème quadratique indéfinie.
- $B_l(N_l)$ est défini comme l'ensemble des indices des variables de base (hors base) respectivement de x_l^* .
- Soit $\bar{\gamma}_j$ la $j^{\text{ème}}$ composante du vecteur de la direction de croissance de la fonction principale γ défini par la relation suivante à chaque étape l de l'algorithme :

$$\bar{\gamma}_j = f_1 \bar{\gamma}_j^2 + f_2 \bar{\gamma}_j^1 + \bar{\mu}_j \bar{\gamma}_j^1 \bar{\gamma}_j^2. \quad (5.1)$$

Où

$$\bar{\gamma}_j^1 = \bar{p}_j - \bar{p}_{B_l} x_j, j \in N_l; \bar{\gamma}_j^2 = \bar{q}_j - \bar{q}_{B_l} x_j, j \in N_l; f_1 = p^t x_l^* + \alpha, i \in B_l; f_2 = q^t x_l^* + \beta, i \in B_l; \phi = f_1 f_2 \text{ and } \bar{\mu}_j = \min \left\{ \frac{(x_l^*)_i}{\bar{a}_{ij}} \mid \bar{a}_{ij} > 0 \right\}, j \in N_l$$

sont les valeurs mise à jour.

Théorème 5.2.2 Une solution $x^* \in S$ est une solution optimale du problème quadratique indéfini (P_l) si et seulement si le vecteur de croissance de la fonction principale $\phi(x)$, $\bar{\gamma}$ est tel que $\bar{\gamma}_j \leq 0$ pour tout indice $j \in N_l$.

La direction de croissance de chaque critère Z_i , $i \in \{1, \dots, r\}$ du problème (P_{MO}) est déterminée à l'aide de leur gradient et celle du critère principal ϕ du problème (P)

est déterminée par γ . La méthode utilise ces informations pour construire une coupe en mesure de supprimer des solutions entières qui ne sont pas efficaces pour le problème multi-objectif (P_{MO}).

Avant de déterminer l'expression mathématique de cette coupe efficace, il faut d'abord définir les ensembles suivants en x_l^* .

$$H_l = H_l^1 \cup H_l^2 \text{ où} \quad (5.2)$$

$$H_l^1 = \left\{ j \in N_l \mid \exists i \in \{1, \dots, r\}; \bar{\gamma}_j^i > 0 \text{ et } \bar{\gamma}_j \geq 0 \right\}. \quad (5.3)$$

$$H_l^2 = \left\{ j \in N_l \mid \bar{\gamma}_j^i = 0, \forall i \in \{1, \dots, r\} \text{ et } \bar{\gamma}_j \geq 0 \right\}. \quad (5.4)$$

$$S_{l+1} = \left\{ x \in S_l \mid \sum_{j \in H_l} x_j \geq 1 \right\}. \quad (5.5)$$

La coupe efficace $\sum_{j \in H_l} x_j \geq 1$ élimine les solutions entières non efficaces.

Description de la méthode

On pose $\phi_{opt} = -\infty$. La méthode consiste à résoudre le programme (P_l) à chaque étape de l'algorithme. Chaque programme (P_l) correspond au nœud l dans une arborescence structurée.

Un nœud l de l'arborescence est sondé si le programme correspondant (P_l) n'est pas réalisable ou que l'ensemble H_l est vide (c'est à dire qu'aucun critère ne peut être amélioré et donc il n'aura plus de solutions efficaces).

Si la solution optimale x_l^* du programme (P_l) n'est pas entière, soit x_j une composante de x_l^* telle que $x_j = \alpha_j$ où α_j est un nombre fractionnaire. Le nœud l de l'arborescence est alors séparé en deux nœuds qui lui sont imposées par les contraintes additionnelles $x_j \leq \lfloor \alpha_j \rfloor$ et $x_j \geq \lfloor \alpha_j \rfloor + 1$ où $\lfloor \alpha_j \rfloor$ indique la partie entière du nombre réel α_j . Dans chaque nœud, le programme quadratique obtenu doit être résolu jusqu'à obtention d'une solution entière si elle existe.

En présence d'une solution entière, la solution est testée pour son efficacité; si elle est efficace, ϕ_{opt} est mis à jour est le nœud correspondant est sondé, sinon la coupe $\sum_{j \in H_l} x_j \geq 1$ est rajoutée au programme et le nouveau programme est résolue.

La méthode se termine quand tous les nœuds créés sont sondés. La solution entière optimale du problème (P) est alors x_{opt} qui correspond à ϕ_{opt} .

Formulation de l'algorithme

Algorithme 9

Étape 1 : Initialiser $l = 0$ et $\phi_{opt} = -\infty$.

Étape 2 : Tant qu'il existe un nœud non encore sondé, choisir le nœud non sondé le plus récemment créé et résoudre le programme quadratique correspondant (P_l) en utilisant la méthode dual du simplexe. (On utilise une des méthodes de résolution du problème quadratique indéfini, existantes en littérature, pour résoudre le problème initial (P_l), pour $l = 0$).

- Si (P_l) n'est pas réalisable, alors le nœud correspondant est sondé;
- Sinon, soit x_l^* sa solution optimale obtenue;
 - ◊ Si $\phi_{opt} \geq \phi(x_l^*)$ alors le nœud est sondé. Aller à l'étape 2;
 - ◊ Sinon
 - ★ Si x_l^* est entière, aller à l'étape (2a);
 - ★ Sinon, aller à l'étape (2b).

Étape 2a : Tester l'efficacité de x_l^* .

- Si x_l^* est efficace, mettre à jour $\phi_{opt} = \phi(x_l^*)$ et $x_{opt} = x_l^*$; le nœud l est sondé. Aller à l'étape 2;
- Sinon, Déterminer les ensembles N_l , H_l^1 , H_l^2 et H_l , construire la coupe efficace;
 - ◊ Si $H_l = \emptyset$, le nœud l est sondé puisque il n'y aura plus d'autre solutions efficaces. Aller à l'étape 2;
 - ◊ Sinon Ajouter la coupe efficace :
 $S_{l+1} = \{x \in S_l / \sum_{j \in H_l} x_j \geq 1\}$ et aller à l'étape 2.

Étape 2b : Appliquer le processus de branchement "Branch & Bound" et séparer le nœud actuel l en deux nouveaux nœud l_1 , $l_1 > l + 1$ et l_2 , $l_2 > l + 1$. Aller à l'étape 2.

Théorème 5.2.3 *Supposons que $H_l \neq \emptyset$ à la solution entière courante x_l^* . $x \neq x_l^*$ est une solution optimale du problème (P) dans le domaine S_l alors $x \in S_{l+1}$.*

Corollaire 5.2.1 *Supposons que $H_l \neq \emptyset$ à la solution entière courante x_l^* . Alors, la contrainte $\sum_{j \in H_l} x_j \geq 1$ est une coupe efficace*

Théorème 5.2.4 *L'algorithme proposé génère une solution optimale du problème (P) et converge en un nombre fini d'itérations.*

Preuve 5.2.2 *L'ensemble S des solutions réalisables de (P_{MO}) , étant borné, il contient un nombre fini de solutions entières, ainsi l'ensemble Eff contient aussi un nombre fini de solutions entières efficaces. Chaque fois qu'une solution entière optimale x_l^* est trouvée, la coupe efficace $\sum_{j \in H_l} x_j \geq 1$ est rajoutée. Donc compte tenu du théorème 5.2.3 et du corollaire 5.2.1, au moins la solution x_l^* est éliminée mais aucune solution efficace n'est supprimée quand tout problème (P_k) ; $k > l$ est résolu.*

Exemple numérique

Pour illustrer l'utilisation de l'algorithme, on considère le même exemple traité dans l'exemple 5.2.1 :

$$(P) \begin{cases} \max \phi(x) = (2x_1 + 4x_2 + 1)(x_1 + x_2 + 2) \\ sc. \\ x \in Eff \end{cases}$$

où Eff est l'ensemble de solutions efficaces du problème linéaire multi-objectif en nombres entiers P_{MO} suivant :

$$(P_{MO}) \begin{cases} \max Z_1 = -3x_1 + x_2 \\ \max Z_2 = x_1 - x_2 \\ sc. \\ 4x_1 + 3x_2 \leq 20 \\ x_1 - x_2 \leq 3 \\ x_2 \leq 4 \\ x_1 \geq 0, x_2 \geq 0 \\ x \text{ entier.} \end{cases}$$

Initialisation : soit $\phi_{opt} = -\infty$, $l = 0$. On résout le problème relaxé :

$$(P_0) \begin{cases} \max \phi(x) = (2x_1 + 4x_2 + 1)(x_1 + x_2 + 2) \\ sc. \\ 4x_1 + 3x_2 \leq 20 \\ x_1 - x_2 \leq 3 \\ x_2 \leq 4 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}$$

Les résultats de résolution du problème (P_0) , sont résumés dans le Tableau 1.

Tab 1	x_3	x_5	b
x_2	0	1	4
x_4	-1/4	7/4	5
x_1	1/4	-3/4	2
γ	-33/4	-657/28	
z_1	3/4	-13/4	
z_2	-1/4	7/4	

La solution optimale obtenue est $x_0 = (2, 4)$ qui n'est pas efficace et $\phi(x_0) = 168$. $H_0 = \{3, 5\}$.

La contrainte $x_3 + x_5 \geq 1$ est rajoutée au Tableau 1 et on utilisant les opérations de pivot, on obtient la solution optimale $x_1 = (7/4, 4)$ qui n'est pas entière et $\phi(x_1) = 1271/8$ (voir tableau 2) .

Tab 2	x_5	x_6	b
x_2	1	0	4
x_4	2	-1/4	21/4
x_1	-1	1/4	7/4
x_3	1	-1	1
γ	-65/8	-31/2	
z_1	3/4	-4	
z_2	-1/4	2	

On utilise le processus de Branch & Bound. Deux nœuds sont créés :

$$N_1 : x_1 \leq 1.$$

$$N_2 : x_1 \geq 2.$$

N_1 : La contrainte $x_1 \leq 1$ est rajoutée au Tableau 2 et on obtient le Tableau 3 avec une solution optimale entière $x_2 = (1, 4)$ qui n'est pas efficace et $\phi(x_2) = 133$, $H_1 = \{5, 7\}$.

Tab 3	x_5	x_7	b
x_2	1	0	4
x_4	1	-1	6
x_1	0	1	1
x_3	-3	-4	4
x_6	-4	-4	3
γ	-31	-31	
z_1	3	-1	
z_2	-1	1	

La coupe $x_5 + x_7 \geq 1$ est rajoutée au Tableau 3 et le Tableau 4 est obtenu. La solution optimale est $x_3 = (0, 4)$ qui est efficace et $\phi(x_3) = 102$. Alors, $\phi_{opt} = 102$ et $x_{opt} = (0, 4)$. Le nœud est sondé.

Tab 4	x_5	x_8	b
x_2	1	0	4
x_4	2	-1	7
x_1	-1	1	0
x_3	1	-4	8
x_6	0	-4	7
x_7	1	-1	1
γ	-29	-12	
z_1	3	-4	
z_2	-1	2	

N_2 : La contrainte $x_1 \geq 2$ est rajoutée au Tableau 2. Le Tableau 5 est obtenu :

Tab 5	x_6	x_7	b
x_2	1/4	1	15/4
x_4	1/4	2	19/4
x_1	0	1	2
x_3	-3/4	1	3/4
x_5	-1/4	-1	1/4
γ	-9	-31/2	
z_1	-1/4	-4	
z_2	1/4	2	

La solution optimale $x_4 = (2, 15/4)$ n'est pas entière, le processus de Branch & Bound est encore lancé. Deux autres nœuds sont créés :

$$(N_3) : x_2 \leq 3.$$

$$(N_4) : x_2 \geq 4.$$

En procédant toujours de cette manière, les nœuds de l'arborescence seront tous sondés, le processus se termine et la solution du problème (P) sera $x_{opt} = (0, 4)$ et $\phi_{opt} = 102$.

Pour résumer l'approche proposé à travers cet exemple, nous présentons une arborescence qui représente les états des nœuds pendant le processus.

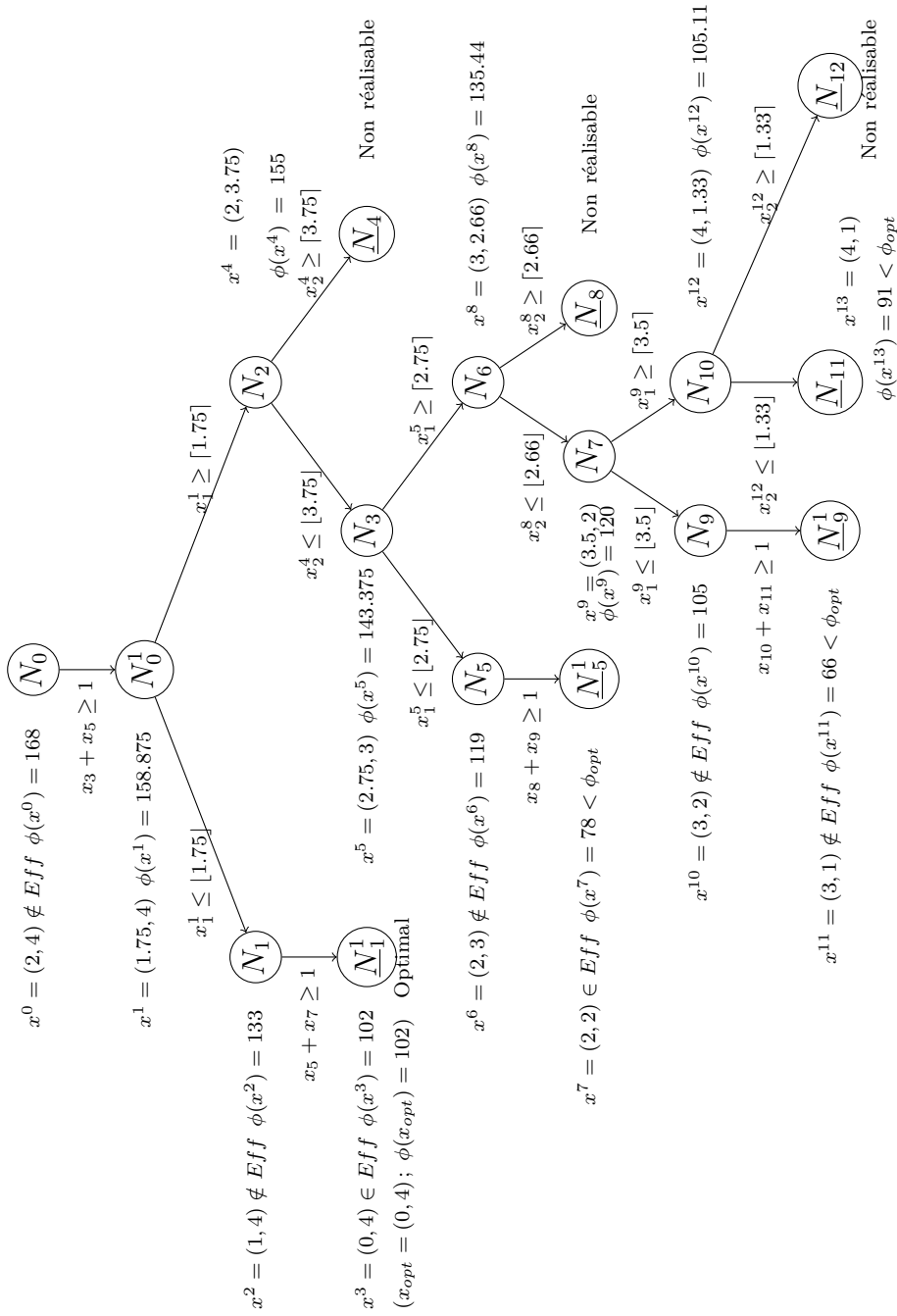


FIGURE 5.2 – Arbre représentant les états des nœuds relatif à la résolution de l'exemple

5.2.3 Expérience numérique

Les deux algorithmes ont été mis en œuvre sous le langage de programmation Matlab R2012a, qui est un langage de programmation interactive, simple, efficace et optimisé pour le traitement des matrices. L'application a été déployée sur une machine équipée d'un processeur IntelCore i3 CPU, 2.53GHZ et 2Go de mémoire.

Nous avons testé les algorithmes sur des instances générées aléatoirement suivant une loi uniforme à l'aide de la fonction redéfinie en Matlab $Randi([v_{min}, v_{max}], n, m)$ qui renvoie une $n \times m$ matrice à coefficient entiers indépendamment uniformément distribués dans l'intervalle $[v_{min}, v_{max}]$.

$[v_{min}, v_{max}]$ est fixé à $[-20, 20]$ pour la matrice A et pour les coefficients des critères du problème (P_{MO}) et il varie dans l'intervalle $[0, 20]$ pour les vecteurs p, q et les réels α et β afin d'assurer la condition de stricte positivité des facteurs $(p^t x + \alpha)$ et $(q^t x + \beta)$. Pour chaque contrainte j , la valeur du second membre b_j est fixée à la partie entière de la somme des coefficients de cette contrainte divisé par 3.

Pour chaque instance (n, m, k) une séquence de 10 problèmes a été résolues par les deux algorithmes proposées, les résultats de l'expérience sont présentés dans le tableau suivant où *temps* représentant le temps de calcul estimé en secondes et *iter* représentant le nombre d'itérations de la méthode du simplexe sont en moyenne.

n	m	k	1 ^{ère} méthode				2 ^{ème} méthode			
			temps (s)		iter		temps (s)		iter	
			moy	max	moy	max	moy	max	moy	max
10	5	5	0,915	1,482	295,5	584	0,746	1,611	333,25	718
10	10	5	1,444	5,090	527,63	2000	1,149	3,950	468,625	2000
10	10	10	1,204	2,033	355,254	753	1,122	2,264	383,5	755
15	10	5	15,983	75,446	2294,125	8291	13,632	25,625	2042	4099
15	10	10	5,131	18,976	1815,875	2937	33,496	214,786	4899,375	27198
15	15	5	11,143	35,881	1519,75	3782	13,184	35,879	1843,625	4101
15	15	10	18,909	45,108	2899,625	5253	11,056	20,928	2110,25	4255
15	15	15	12,7213	20,8198	1945	2315	9,6621	18,6405	1945	3976
20	10	10	13,050	26,066	2250,25	483	27,823	110,254	3639	12124
20	15	10	23,770	74,026	12970,625	37793	15,779	40,889	8729,875	17965
20	20	10	16,612	39,896	8602,875	17599	14,391	36,0528	7692,125	15980
20	20	15	56,472	220,967	12101,125	24602	42,536	109,345	11302,25	20687
20	20	20	13,583	21,938	7475,5	13667	13,066	30,554	7725,375	14869
30	30	30	84,199	180,774	34406,75	78183	81,449	170,634	38384,625	78183

TABLE 5.1 – Résultats de l'étude comparative

A travers les résultats résumés dans le tableau 5.1, nous pouvons affirmer que les deux méthodes proposées pour résoudre le problème (P) sont efficaces.

L'expérimentation des deux méthodes nous a permis de conclure les résultats suivants :

- Le temps de calcul et le nombre d'itérations des deux méthodes augmentent rapidement avec la taille des données, en particulier avec l'augmentation du nombre de variables, en revanche l'effet d'augmenter le nombre de contraintes pour un nombre fixe de variables est moins perceptible, puisque le nombre de contraintes est généralement inversement proportionnel à la taille de l'ensemble réalisable.
- Le nombre de critères ne fait pas augmenté le temps de calcul de façon significatif.
- L'étude comparative des deux méthodes sur les instances proposées montre que la deuxième méthode développée est moins coûteuse en terme de temps d'exécution et de nombre d'itérations nécessaire et donc plus intéressante que la deuxième méthode.
- La complexité algorithmique de la méthode dépend fortement du nombre de va-

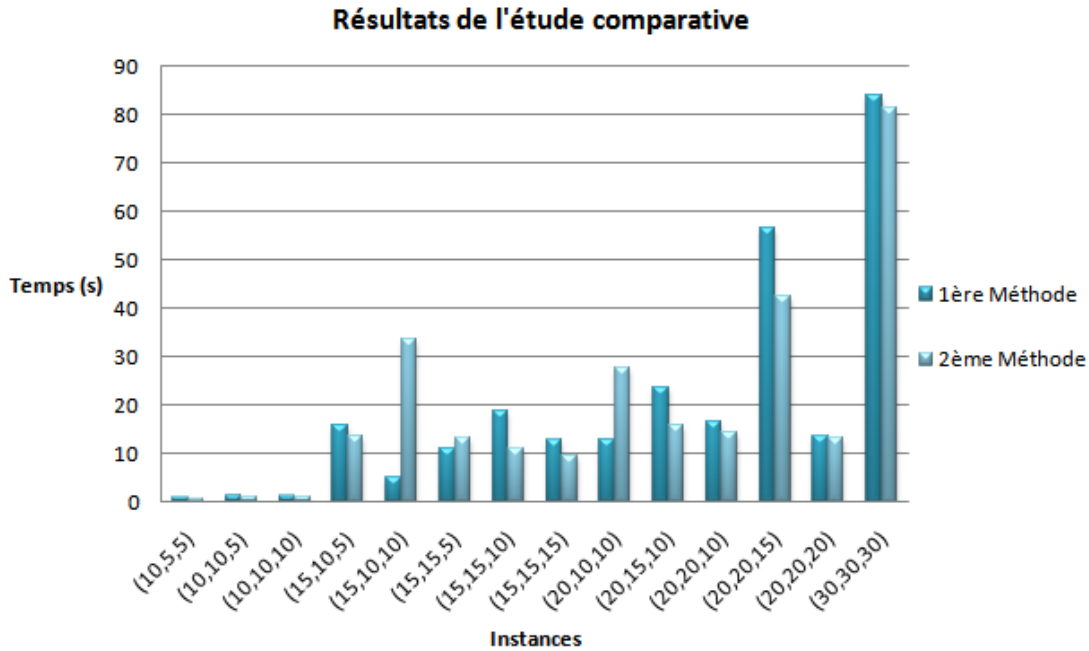


FIGURE 5.3 – Histogramme de l'étude comparative des deux méthodes

riables n et de contraintes m utilisées.

- Les résultats obtenus par l'implémentation sont satisfaisants car l'utilisation de la coupe efficace évite l'exploration de tout le domaine réalisable.

5.3 Optimisation d'un critère quadratique semi-défini sur l'ensemble efficient d'un problème (MOILP)

Le problème principale que nous voulons étudier est formulé par :

$$(P) \begin{cases} \max \phi(x) = \frac{1}{2}x^t G x + g^t x \\ sc. \\ x \in Eff \end{cases}$$

où G est une matrice réelle semi-définie négative d'ordre $n \times n$, x et g deux vecteurs de \mathbb{R}^n .

$\phi(x)$ une fonction d'utilité quadratique représentant les préférences du décideur et Eff représente l'ensemble efficient du problème multi-objectif linéaire discret (MOILP)

définit par :

$$(P_{MO}) \begin{cases} \max Z_i(x) = c^i x, & i \in \{1, \dots, r\} \\ sc. \\ x \in D = S \cap \mathbb{Z}^n \end{cases}$$

où $r \geq 2$ est un nombre entier, $S = \{x \in \mathbb{R}^n / Ax \leq b, x \geq 0\}$; $c^i \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, D étant supposé l'ensemble non vide des points entiers de S et S un polyèdre convexe borné de \mathbb{R}^n .

5.3.1 Description de la méthode

La méthode que nous proposons pour générer la solution optimale du problème centrale (P) est inspirée de la méthode de Jorge [46]. Cette méthode fournit une solution optimale globale du problème (P) sans passer par toutes les solutions efficaces du (P_{MO}).

L'approche adoptée dans ce travail est basé sur la résolution d'un programme noté R_l à chaque itération l de l'algorithme, ce programme optimise un critère quadratique sur un domaine restreint par des contraintes linéaire supplémentaires imposées au programme R_{l-1} résolu précédemment. Ces nouvelles contraintes sont incluses progressivement pour éliminer les solutions efficaces déjà trouvées, et faire en sorte que les nouvelles solutions générées soient efficaces.

On initialise l à 1 représentant le nombre d'itération.

L'algorithme commence par la résolution du programme relaxé (P_R) suivant en utilisant une des méthodes de résolution d'un programme quadratique concave semi-défini citées dans le premier chapitre :

$$(P_R) \begin{cases} \max \phi(x) = \frac{1}{2} x^t G x + g^t x \\ sc. \\ x \in D. \end{cases}$$

avec $D = S \cap \mathbb{Z}^n$, $S = \{x \in \mathbb{R}^n / Ax \leq b, x \geq 0\}$.

Si la solution optimale retrouvée est non entière, alors le processus de séparation et évaluation de Branch & Bound sera déclenché afin de pouvoir fournir une solution optimale entière au problème (P_R).

À chaque itération l de l'algorithme, l'étape 1 consiste à tester l'efficacité de la solution x^l précédemment trouvée. Le test permet de vérifier l'efficacité de la solution x^l et me permet d'avoir une autre solution \hat{x}^l qui est efficace si x^l ne l'est pas.

Si la solution x^l est efficace, l'algorithme s'arrête, sinon, on passe à l'étape 2.

À l'étape 2, on résout le problème (T_l) suivant :

$$\max \{ \phi(x) = \frac{1}{2} x^t G x + g^t x \mid Cx = C\hat{x}^l, x \in D \}.$$

Ce problème permet d'optimiser le critère principale $\phi(x)$ sur toutes les solutions alternatives à \hat{x}^l . Soit \tilde{x}^l la solution optimale du problème.

Notons que pour ce problème, au moins une solution réalisable initiale est disponible, à savoir \hat{x}^l , qui peut être optimale.

L'étape 3 de l'algorithme vise à fournir une nouvelle solution entière non générée précédemment et qui n'est pas dominée par aucune solution efficace déjà trouvée. Cette tâche est assurée par la résolution du problème (R_l) suivant :

$$\max\{\phi(x) = \frac{1}{2}x^t Gx + g^t x / x \in D - \cup_{s=1}^l D_s\}$$

où $D_s = \{x \in \mathbb{Z}^n / C\tilde{x}^s \geq Cx\}$, avec $\tilde{x}^1, \tilde{x}^2, \dots, \tilde{x}^l$ sont les solutions optimales des problèmes $(T_1), (T_2), \dots, (T_l)$ respectivement.

Il est possible de fournir une solution optimale au problème (R_l) , en résolvant le problème \hat{R}_l qui lui est équivalent :

$$(\hat{R}_l) \begin{cases} \max \phi(x) = \frac{1}{2}x^t Gx + g^t x \\ x \in D \\ c^i x > (c^i \tilde{x}^l + 1)y_i^l - M_i(1 - y_i^l), \quad i = 1, \dots, r \\ \sum_{i=1}^r y_i^l = 1 \\ y_i^l \geq 0, \quad i = 1, \dots, r. \end{cases}$$

où $-M_i$ est un minorant pour toute valeur réalisable de la $i^{\text{ème}}$ critère.

Notons que lorsque $y_i^l = 0$ la contrainte associée est redondante et lorsque $y_i^l = 1$, une stricte amélioration du $i^{\text{ème}}$ critère est imposée.

On note x^{l+1} la solution obtenue en résolvant (\hat{R}_l) , on pose $l=l+1$ et on passe à l'étape 1.

Développement de l'algorithme

Algorithme 10

Étape 0 : (Initialisation) Poser $\phi_{inf} = -\infty$, $\phi_{sup} = +\infty$, $l = 1$ et résoudre le problème relaxé (P_R) par n'importe quelle méthode directe de la programmation quadratique discrète.

Si (P_R) est irréalisable, **Stop**. (P) est aussi irréalisable.

Sinon , soit x_l une solution optimale de (P_R) .

Étape 1 : Si x_l est efficace, **Stop**. $x_{opt} = x_l$ est une solution optimale de (P_E) et $\phi_{opt} = \phi(x_l)$.

Sinon , poser $\phi_{sup} = \phi(x_l)$ et aller à l'étape 2.

Étape 2 : Soit $\tilde{x}^l \in Eff$ efficace fournie par le test d'efficacité et soit \tilde{x}^l une solution optimale du problème (T_l) suivant :

$$\max\{\phi(x) = \frac{1}{2}x^t G x + g^t x \mid Cx = C\tilde{x}^l, x \in D\}.$$

Résoudre le problème (T_l) pour optimiser le critère principal sur toutes les solutions équivalentes à \tilde{x}^l .

Si $\phi(\tilde{x}^l) > \phi_{inf}$, poser $\phi_{inf} = \phi(\tilde{x}^l)$ et $x_{opt} = \tilde{x}^l$.

Si $\phi_{inf} = \phi_{sup}$, **Stop**. x_{opt} est une solution optimale de (P_E) .

Étape 3 : Résoudre le problème (R_l) suivant :

$$\max\{\phi(x) = \frac{1}{2}x^t G x + g^t x \mid x \in D - \cup_{s=1}^l D_s\}, \text{ où } D_s = \{x \in \mathbb{Z}^n \mid C\tilde{x}^s \geq Cx\}.$$

avec $\tilde{x}^1, \tilde{x}^2, \dots, \tilde{x}^l$ sont les solution optimales des problèmes $(T_1), (T_2), \dots, (T_l)$ respectivement

Si (R_l) est irréalisable, **Stop**. x_{opt} est une solution optimale du problème (P_E) .

Sinon , soit x^{l+1} une solution optimale de (R_l) .

Si $\phi(x^{l+1}) \leq \phi_{inf}$, **Stop**. x_{opt} est une solution optimale de (P_E) .

Sinon , poser $l = l + 1$ et aller à l'étape 1.

Théorème 5.3.1 *L'algorithme de recherche d'une solution optimale du problème (P) converge en un nombre fini d'itérations*

Preuve 5.3.1 *Par l'hypothèse que la région d'admissibilité S est bornée et D supposé non vide, ainsi l'ensemble Eff contient un nombre fini de solutions entières efficaces.*

Chaque fois qu'une solution optimale entière x^l du problème (R_l) est trouvées, le test d'efficacité nous permet de vérifier l'efficacité de cette solution et donc obtenir une solution optimale à notre problème (P) , soit d'obtenir une autre solution retournées qui est efficace. Donc à chaque itération de l'algorithme, une solution entière efficace est générée, et avec les propositions 4.3.14.3.2, chaque itération permet une amélioration du critère principale et une réduction du domaine de recherche progressivement jusqu'à ce qu'il devient vide et donc la convergence de l'algorithme vers une solution optimale pour le problème (P) est assurée.

5.3.2 Exemple numérique

Considérons le problème suivant :

$$(P) \begin{cases} \max \phi(x) = -2x_1^2 - 8x_2^2 + 8x_1 + 16x_2 \\ sc. \\ x \in Eff \end{cases}$$

où Eff est l'ensemble efficient du problème linéaire multi-objectif en nombres entiers P_{MO} suivant :

$$(P_{MO}) \begin{cases} \max Z_1 = 3x_1 - x_2 \\ \max Z_2 = -x_1 + x_2 \\ sc. \\ 4x_1 + 3x_2 \leq 20 \\ x_1 - x_2 \leq 3 \\ x_2 \leq 4 \\ x_1 \geq 0, x_2 \geq 0 \\ x \text{ entier.} \end{cases}$$

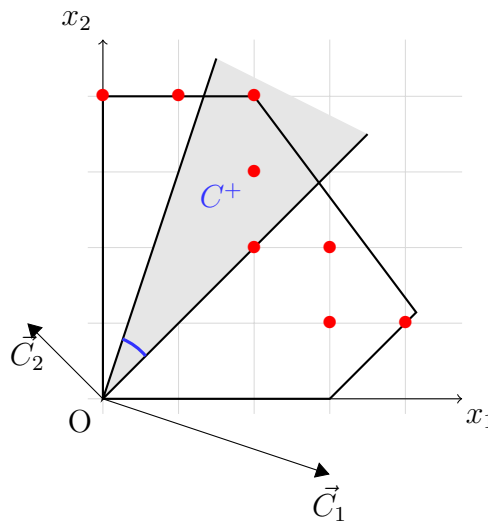


FIGURE 5.4 – L'ensemble des solutions efficaces

L'ensemble de toute les solutions efficaces du problème (P_{MO}) est :

$$Eff = \{(3; 1), (4; 1), (2; 2), (3; 2), (2; 3), (2; 4), (1; 4), (0; 4)\}.$$

x	(3;1)	(4;1)	(2;2)	(3;2)	(2;3)	(2;4)	(1;4)	(0;4)
$Z(x)$	(8;-2)	(11;-3)	(4;0)	(7,-1)	(3,1)	(2;2)	(-1;3)	(-4;4)
$\phi(x)$	14	8	8	6	-16	-56	-58	-64

On prend les bornes inférieures des deux fonctions objectifs $-M1 = -4$, $-M2 = -3$.

Étape 0 : Initialisation ; soit $\phi_{opt} = -\infty$.

On résout le problème relaxé par n'importe quelle méthode directe de la programmation quadratique discrète :

$$(P_R) \begin{cases} \max \phi(x) = -2x_1^2 - 8x_2^2 + 8x_1 + 16x_2 \\ sc. \\ (x_1, x_2) \in D \end{cases}$$

La résolution du problème (P_R) a abouti à la solution entière suivante $x^1 = (2, 1)$ et $\phi(x^1) = 16$.

Étape 1 :

On teste l'efficacité de x^1 . La solution x^1 n'est pas efficace, une autre solution efficace engendrée par le test d'efficacité est $\hat{x}^1 = (3, 2)$.

On pose $\phi_{sup} = \phi(x^1) = 16$. Aller à l'étape 2.

Étape 2 :

Soit $\hat{x}^1 = (3, 2) \in Eff$. Le vecteur critère correspondant à cette solution est $Z(\hat{x}^1) = (7, -1)$.

Pour trouver les solutions efficaces qui ont le même vecteur critère on résout le problème (T_1) suivant :

$$(T_1) \begin{cases} \max \phi(x) = -2x_1^2 - 8x_2^2 + 8x_1 + 16x_2 \\ sc. \\ (x_1, x_2) \in D \\ 3x_1 - x_2 = 7 \\ -x_1 + x_2 = -1 \end{cases}$$

La solution optimale du problème (T_1) est $\tilde{x}^1 = \hat{x}^1 = (3, 2)$ et on a $\phi(\tilde{x}^1) = 6 > \phi_{inf}$, Alors $\phi_{inf} = 6$ et $x_{opt} = (3, 2)$.

$\phi(\tilde{x}^1) \neq \phi_{sup}$, aller à l'étape 3.

Étape 3 :

Résoudre le problème (R_1) équivalent au programme (\hat{R}_1) suivant :

$$(\hat{R}_1) \left\{ \begin{array}{l} \max \phi(x) = -2x_1^2 - 8x_2^2 + 8x_1 + 16x_2 \\ sc. \\ (x_1, x_2) \in D \\ 3x_1 - x_2 \geq 8y_1^1 - 4(1 - y_1^1) \\ -x_1 + x_2 \geq -3(1 - y_2^1) \\ y_1^1 + y_2^1 \geq 1, y_1^1, y_2^1 \in \{0, 1\} \end{array} \right.$$

La solution optimale du problème \hat{R}_1 est $x^2 = (1, 1)$, $y = (0, 1)$ et $\phi(x^2) = 14$. Comme $\phi(x^2) = 14 > \phi_{inf}$, on pose $l=l+1$.

Étape 1 :

La solution x_2 est testée pour son efficacité. La solution $x^2 = (1, 1)$ n'est pas efficace et la solution efficace engendrée est $\hat{x}^2 = (2, 2)$

Étape 2 :

Soit $\hat{x}^2 = (2, 2)$ une solution efficace. Le vecteur critère correspondant à cette solution est $Z(\hat{x}^1) = (4, 0)$.

Pour trouver les solutions efficace qui ont le même vecteur critère on résout le problème (T_2) suivant :

$$(T_2) \left\{ \begin{array}{l} \max \phi(x) = -2x_1^2 - 8x_2^2 + 8x_1 + 16x_2 \\ sc. \\ (x_1, x_2) \in D \\ 3x_1 - x_2 = 4 \\ -x_1 + x_2 = 0 \end{array} \right.$$

La solution optimale du problème (T_2) est $\tilde{x}^2 = \hat{x}^2 = (2, 2)$ et on a $\phi(\tilde{x}^2) = 8 > \phi_{inf}$, Alors $\phi_{inf} = 8$ et $x_{opt} = (2, 2)$.

$\phi(\tilde{x}^2) \neq \phi_{sup}$, aller à l'étape 3.

Étape 3 :

Résoudre le problème (R_2) équivalent au programme (\hat{R}_2) suivant :

$$(\hat{R}_2) \left\{ \begin{array}{l} \max \phi(x) = -2x_1^2 - 8x_2^2 + 8x_1 + 16x_2 \\ sc. \\ (x_1, x_2) \in D \\ 3x_1 - x_2 \geq 8y_1^1 - 4(1 - y_1^1) \\ -x_1 + x_2 \geq -3(1 - y_2^1) \\ 3x_1 - x_2 \geq 5y_1^2 - 4(1 - y_1^2) \\ -x_1 + x_2 \geq y_2^2 - 3(1 - y_2^2) \\ y_1^1 + y_2^1 \geq 1; y_1^1, y_2^1 \in \{0, 1\} \\ y_1^2 + y_2^2 \geq 1; y_1^2, y_2^2 \in \{0, 1\} \end{array} \right.$$

La solution du problème (\hat{R}_2) est $x^3 = (3, 1)$, $y = (1, 0, 1, 0)$ et $\phi(x_3) = 14$. Comme $\phi(x^3) = 14 > \phi_{inf}$, on pose $l=l+1$. Aller à l'étape 1.

Étape 1 :

La solution x^3 est efficace (utilisation d'un test d'efficacité).

Terminer ; la solution optimale du problème (P) est $x_{opt} = (3, 1)$ et la valeur optimale $\phi_{opt} = 14$.

5.3.3 Expérience numérique

La méthode décrite dans la section précédente a également été programmée pour la recherche de la solution optimale du problème (P) . L'algorithme a été mis en œuvre sous le langage de programmation Matlab R2012a en utilisant une machine équipée d'un processeur IntelCore i3 CPU, 2;53GHZ et 2Go de mémoire. Il a été testé sur des instances générées aléatoirement suivant une loi uniforme à l'aide de la fonction redéfinie en Matlab $Randi([v_{min}, v_{max}], n, m)$ qui renvoie une $n \times m$ matrice à coefficient entiers indépendamment uniformément distribuées dans l'intervalle $[v_{min}, v_{max}]$.

L'intervalle $[v_{min}, v_{max}]$ est fixé à $[-20, 20]$ pour la matrice A , les coefficients des critères du problème (P_{MO}) et pour le vecteur g qui représente la partie linéaire du critère quadratique principale. Pour la partie quadratique du critère principale, la matrice G est prise comme le produit $Q^t \times Q$, avec Q une $(n \times n)$ -matrice non dégénérée, tel que l'intervalle $[v_{min}, v_{max}]$ pour Q est fixé à $[-2, 2]$.

Pour chaque contrainte j , la valeur du second membre b_j est fixée à la partie entière de la somme des coefficients de cette contrainte divisé par 3.

Pour chaque instance (n, m, k) une séquence de 10 problèmes a été résolues par l'algorithme proposé. Nous donnons les résultats obtenus dans le tableau 5.2.

L'expérimentation de méthode nous a permis de conclure les résultats suivants :

n	m	k	temps (s)		iter	
			moy	max	moy	max
10	5	5	8,1353	14,7711	1326,25	2180
10	10	5	8,3669	12,1589	1558,75	2185
10	10	10	7,592	15,4988	1181,625	2938
15	5	5	14,6452	31,0516	6249,375	12617
15	10	5	22,6506	62,2693	8419,375	14115
15	10	10	28,2542	50,1647	5474,75	11539
15	15	10	23,9586	55,7623	8324,375	16952
15	15	15	61,4856	223,2984	10223,5	23147
20	10	10	52,8034	103,814	15633,375	23829
20	15	10	78,6641	234,349	25231,5	73328
20	20	10	166,1626	422,6611	34458	51863
20	20	15	134,7444	232,9062	40527,875	73238
20	20	20	349,8743	859,0814	26810,5	76376

TABLE 5.2 – Résultats de l'expérimentation

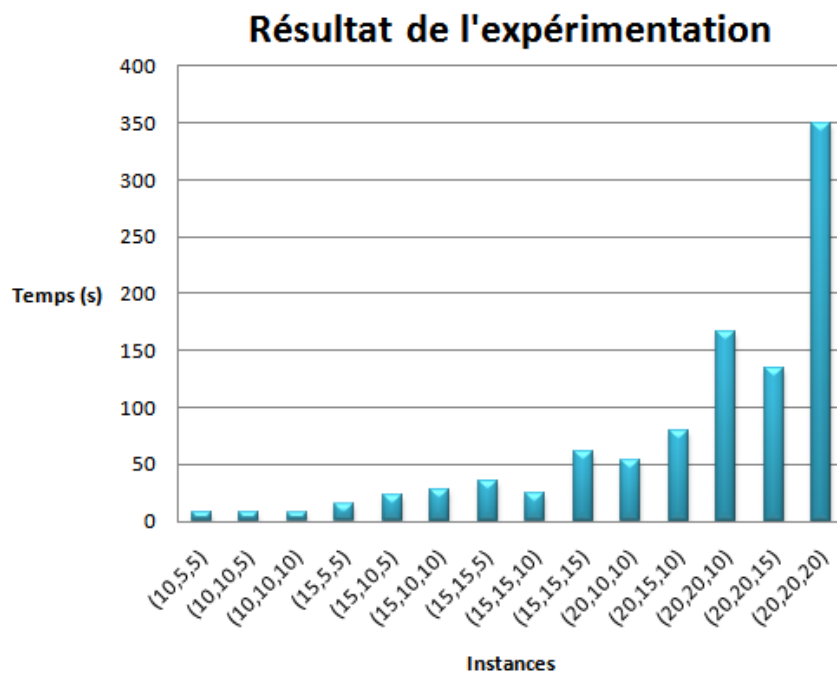


FIGURE 5.5 – Histogramme de l'expérimentation de la méthode

- Le temps de calcul et le nombre d'itérations de la méthode augmente rapidement avec la taille des données, en particulier avec l'augmentation du nombre de variables, en revanche l'effet d'augmenter le nombre de contraintes pour un nombre fixe de variables est moins perceptible.
- Le nombre de critères ne fait pas augmenter le temps de calcul de façon significative.
- La complexité algorithmique de la méthode dépend fortement du nombre de variables n et de contraintes m utilisées.

5.4 Conclusion

Dans ce travail nous avons présenté un algorithme pour optimiser une fonction quadratique (semi-définie et indéfinie) sur l'ensemble efficient d'un problème MOILP, cet algorithme est inspirée de la méthode de Jorge [46] et permet d'obtenir une solution optimale globale du problème (P) sans passer par toutes les solutions efficaces du (P_{MO}).

L'algorithme proposé est basé sur la résolution d'une séquence de programme quadratique (R_l) sur un domaine restreint par des contraintes linéaires supplémentaires imposées au programme (R_{l-1}) résolu précédemment.

Une autre méthode exacte combinant le principe de séparation "branch & bound" avec une coupe efficace est présentée pour résoudre un problème d'optimisation d'un critère quadratique indéfini sur l'ensemble efficient d'un problème multi-objectif linéaire en nombres entiers ($MOILP$). Cette approche permet de résoudre le problème (P) en évitant le passage systématique par toutes les solutions efficaces et la coupe efficace utilisée exploite tous les différents objectifs dans un même tableau du simplexe et seules les parties du domaine réalisable contenant des solutions efficaces sont explorées.

Un exemple numérique est traité pour chaque programme et une étude comparative des deux approches proposées pour la résolution d'un problème quadratique indéfini sur l'ensemble efficient d'un MOILP est faite.

Conclusion Générale

Les problèmes d'optimisation multi-objectif sont de plus en plus étudiés car ils peuvent donner de l'aide indispensable aux processus de décision. L'optimisation multi-objectif est sans doute un axe de recherche primordial pour les scientifiques et les ingénieurs, non seulement à cause de la nature multi-objectif de la plupart des problèmes réels, mais aussi parce que de nombreuses questions restent ouvertes dans ce domaine.

Dans cette étude, nous nous sommes intéressés aux problèmes de programmation multi-objectif en nombres entiers. L'intérêt de tels problèmes résulte du fait que dans de nombreuses situations réelles modélisables par la programmation mathématique, les variables de décision ne peuvent prendre que des valeurs entières.

Le décideur doit réaliser une analyse de l'ensemble des solutions efficaces du problème pour sélectionner les solutions efficaces préférées, et cela s'avère impossible lorsque cet ensemble est large. Dans le cas où les préférences du décideur sont modélisées, explicitement, par une fonction, alors le problème du choix des solutions efficient peut être énoncé sous forme d'un problème d'optimisation de cette fonction sur l'ensemble des solutions efficaces. Ce type de problèmes est très difficile à résoudre vu que son ensemble d'admissibilité n'est pas convexe.

Une complication additionnelle est superposée au problème lorsque les préférences du décideur sont modélisées sous forme d'une fonction quadratique, cela est dû principalement à la difficulté qu'on trouve lors de la résolution d'un problème non linéaire et qui réside dans le fait qu'un optimum local n'est pas forcément un optimum global.

Notre travail s'articule autour de deux volets :

- Le premier volet consiste à optimiser un critère quadratique indéfinie qui exprime les préférences du décideur sur l'ensemble efficient d'un problème multi-objectif linéaire discret (Quadratique-Linéaire), la fonction prise en considération peut être écrite sous forme d'un produit de deux fonctions affines.
- Le deuxième volet consiste à optimiser un critère quadratique semi-défini sur l'ensemble efficient d'un problème multi-objectif linéaire discret (Quadratique-

Linéaire).

Une extension du cas (Linéaire-Linéaire) de l'algorithme présenté dans [46], au cas (Quadratique-Linéaire) nous a permis de mettre au point une méthode exacte pour la résolution d'un problème d'optimisation d'un critère Quadratique définie et indéfinie sur l'ensemble efficient d'un problème multi-objectif linéaire discret *MOILP*. Une deuxième méthode basée principalement sur le processus de séparation de la méthode de "Branch & Bound" couplé à une coupe dite efficace est mise au point pour résoudre le cas (Quadratique-Linéaire) lorsque le critère représentant les préférences du décideur est une fonction quadratique indéfinie. Les deux méthodes proposées évitent le passage systématique par toutes les solutions efficaces.

Parmi les travaux qui peuvent présenter des perspectives et que nous souhaitons aborder pour l'avenir, nous y trouvons par exemple :

- Résolution d'un problème d'optimisation d'un critère quadratique sur l'ensemble efficace d'un problème multi-objectif fractionnaire linéaire discret.
- Résolution d'un problème d'optimisation d'un critère quadratique sur l'ensemble efficace d'un problème multi-objectif quadratique discret.

Bibliographie

- [1] L. Abbaci et M. Moulaï, *L'optimisation sur l'ensemble efficient d'un problème stochastique multi-objectif discret*, thèse de Magister en Recherche Opérationnelle, USTHB, Algérie, 2010.
- [2] M. Abbas and D. Chaabane, *An algorithm for solving multiple objective integer linear programming problem*, RAIRO Operations Research 36, pp. 351-364, 2002.
- [3] M. Abbas and M. Moulaï, *Solving multiple objective integer linear programming*, Journal of the Italian Operations Research Society (Ricerca Operativa) 29, pp. 15-38, 1999.
- [4] M. Abbas and D. Chaabane, *Optimizing a linear function over an integer efficient set*, European Journal of Operational Research 174, No2; 1140-1161, 2006.
- [5] P. Anand, *Decomposition principle for indefinite quadratic programme*, Trabajos de Estadística de Investigación Operativa 23, 61-71, 1972.
- [6] K. Belkeziz et A. Metrane, *Optimisation d'une fonction linéaire sur l'ensemble des solutions efficaces d'un problème multicritère quadratique convexe*, Laboratoire de mathématiques de l'université Blaise-Pascal, France, V11, n°1, p.19-33, 2004.
- [7] C.R. Bector and M. Dahl, *Simplex type finite iteration technique and duality for a special type of pseudo-concave quadratic program*, Cahiers du Centre d'Etudes de Recherche Opérationnelle 16, 207-222, 1974.
- [8] R. Benayoun, J. De Montgolfier, J. Tergny and O. Laritchev, *Linear Programming with Multiple Objective Functions : Step Method (STEM)*, Mathematical Programming 1, No-3, 366-375, 1971.
- [9] H.P. Benson, *Existence of Efficient Solutions for Vector Maximization Problems*, Journal of Optimization Theory and Applications 26, No-4, 569-580, 1978.
- [10] H.P. Benson, *Optimization over the efficient set*, Journal of Mathematical Analysis and Applications 98, 562-580, 1984.
- [11] H.P. Benson, *An All-Linear Programming Relaxation Algorithm for Optimizing over the Efficient Set*, Journal of Global Optimization, Vol.1, 83-104, 1991.

-
- [12] H.P. Benson, *A finite Nonadjacent Extreme point Search Algorithm over the efficient Set*, Journal of Optimization Theory and Applications, Vol.73, 47-64, 1992.
- [13] H.P. Benson, *A bisection-extreme point search algorithm for optimizing over the efficient set in the linear dependence case*, Journal of Global Optimization 3, 95-111, 1993.
- [14] H.P. Benson and S. Sayin, *14 Optimizing over the Efficient Set : Four Special Cases*, Journal of optimization Theory and Applications, Vol.80, No1 ; 3-17 :, 1994. Journal 1, 1, 9-46, 1999.
- [15] D. Chaabane and M. Pirlot, *A method for optimizing over the integer efficient set*, Journal of industrial and management optimization, Vol.6, No4 ; 811-823 ; 2010.
- [16] D. Chaabane and F.Mebrek, *Optimization of a linear function over the set of stochastic efficient solutions*, Springer Verlag, Berlin Heidelberg, 2012.
- [17] V. Chankong and Y.Y. Hamies, *Multiobjective Decision Making : Theory and Methodology*, New York : Elsevier-North-Holland, 1983.
- [18] A. Charnes and W.W. Cooper, *Management models and industrial applications of linear programming*, John Wiley, New York, 1961.
- [19] Y. Collette et P. Siarry, *Optimisation multiobjectif*, Editions Eyrolles, 2002.
- [20] R.W. Cottle, *Three remarks about two papers on quadratic forms*, Zeitschrift für Operations Research 19, 123-124, 1975.
- [21] H.W. Corley, *A New Scalar Equivalence for Pareto Optimization*, IEEE Transactions on Automatic Control 25, No-4, 829-830, 1980.
- [22] A. Crema and J. Sylva, *A method for finding the set of nondominated vectors for multiple objective integer linear programs*, European Journal of Operational Research, in press, 2003.
- [23] G.B. Dantzig, *Maximization of linear function of variables subject to linear inequalities : in t.c.*, Koopmans ed. Activity Analysis of. John Wiley Sons, New York, 339-347, 1951.
- [24] G.B Dantzig and P.Wolfe, *Decomposition principle for linear programs*, Operations Research 8, 101-111, 1960.
- [25] G.B. Dantzig *Linear programming and extention*, Princeton University Press, 1963
- [26] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, New-York : John Wiley, 2001.

-
- [27] C. Dhaenens-Flipo, *Optimisation Combinatoire Multi-Objectif : Apport des Méthodes Coopératives et Contribution à l'Extraction de Connaissances*, thèse d'Habilitation à diriger des Recherches de l'U.S.T.L, Lille, 2005.
- [28] R. Dorfman, *Application of linear programming to the theory of the firm*. University of California Press, 1951.
- [29] J.G. Ecker and I.A. Kouada, *Finding Efficient Points for Linear Multiple Objective Programs*, Mathematical Programming 8, No-8, 375-377, 1975.
- [30] J.G. Ecker and J.H. Song *Optimizing a linear function over an efficient set*, Journal of Optimization Theory and Applications : Vol.83, No3 ; 541-563 ; 1994.
- [31] M. Ehrgott, H.W Hamacher, K. Klamroth K,S. Nickel, A. Schobel and M.M. Wiecek, *A note on the equivalence of balance points and Pareto solutions in multiple-objective programming*, Journal of Optimization Theory and Applications : Vol.92, n°1 ; 209-212 ; 1997.
- [32] M. Ehrgott and X. Gandibleux, *A survey and annotated bibliography of multiobjective combinatorial optimization*, OR Spektrum, 22 :425-460, 2000.
- [33] M. Ehrgott, *Multicriteria Optimization*, Lecture Notes in Economics and Mathematical Systems, numéro 491, Springer Verlag, Berlin, Heidelberg, 2000.
- [34] J.E. Falk and S.W. Paloscy, *Optimising the sum of linear fractional functions*, Advances in Global Optimisation, Kluwer Academic Publishers, 221-258, 1992.
- [35] P.H. Farquhar, *Utility assesment methods*, Management Science 30, 1283-1300, 1984.
- [36] P.C. Fishburn, *Utility theory*, Managment science, vol. 14, no. 4, 355-378, 1968.
- [37] R.W. Freund and F. Jarre, *Solving the sum-of-ratios problem by an interior point method*, Journal of Global Optimization, Vol. 19, 83-102, 2001.
- [38] M. Frank and P. Wolfe, *An algorithm for quadratic programming*, Naval Research Logistics Quarterly, 3, 95-110,1956.
- [39] A.M. Geoffrion, *Proper Efficiency and the Theory of Vector Maximization*, Journal of Mathematical Analysis and Applications 22, No-3, 618-630, 1968.
- [40] R. Gupta and R. Malhotra, *Multi-criteria integer linear programming problem*, Cahiers de CERO 34, (1992).
- [41] J.M. Henderson and R.E. Quandt, *Microeconomic Theor*, McGraw-Hill, New York, 1971.
- [42] G.P. Huber, *Multiattribute utility models, A review of field and fieldlike studies*, Managment Science 20, No-10, 1393-1402, 1974.

-
- [43] J.P. Ignizio, *Goal Programming and Its Extensions*, D.C. Heath, Lexington, MA, 1976.
- [44] J. Ijiri, *Management Goals and Accounting for Control*, American Elsevier, New York, 1965.
- [45] H. Isermann R.E. Steuer, *Computational Experience Concerning Payoff Tables and Minimum Criterion Values over the Efficient Set*, European Journal of Operational Research. Vol.33, 91-97, 1987.
- [46] J.M. Jorge, *An algorithm for optimizing a linear function over an integer efficient set*, European Journal of Operational Research 195, 98-103, 2009.
- [47] N.K. Karmarkar, *A new polynomial-time algorithm for linear programming*, Combinatorica 373-395, 1984.
- [48] R.L. Keeney and H. Raiffa, *Decisions with multiple objectives : Preferences and value tradeoff*, éditions Cambridge University Press, 1993.
- [49] D. Klein and E. Hannan, *An algorithm for multiple objective integer linear programming problem*, European Journal of Operational Research 9, pp. 378-385, 1982.
- [50] T.C. Koopmans, *Analysis and Production as an Efficient Combination of Activities*, Activity Analysis of Production and Allocation, Yale University Press, New Haven, London, 33-97, 1971 (originally published in 1951).
- [51] H.W. Kuhn and A.W. Tucker, *Nonlinear Programming*, Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, Los Angeles, 481-492, 1951.
- [52] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, Third Edition, Springer, 2008.
- [53] H. Markowitz, *Portfolio selection : Efficient diversification of investment*, John Wiley and Sons, New York 1959.
- [54] J.M. Martel and B. Aouni, *Diverse Imprecise Goal Programming Model Formulations*, Journal of Global Optimization, 12, 1998, 127-138.
- [55] K. Miettinen, *On the Methodology of Multiobjective Optimization Problems*, Journal of Optimization Theory and Applications 42, No-4, 499-524, 1984.
- [56] K. Miettinen, *Nonlinear Multiobjective Optimization*, Kluwer Academic Publishers, Boston, 1999.
- [57] G.L. Nemhauser, L.A. Wolsey, *Integer and combinatorial optimization*, Wiley, Chichester, 1988.

-
- [58] Y. Nesterov and A. Nemirovski, *Interior-Point Polynomial Algorithms in Convex Programming*, Philadelphia, PA : SIAM, 1994.
- [59] N.C. Nguyen, *An Algorithm for Optimizing a Linear Function over the Integer Efficient Set*, Konrad-Zuse-zentrum fur Informationstechnik Berlin, 1992.
- [60] F.Z. Ouail et M.Moulaï , *Optimisation Vectorielle Discrète*, thèse de Magister en mathématiques, option Mathématique de gestion, USTHB, Algérie, 2008.
- [61] I. Othmani, *Optimisation multicritère*, Fondements et Concepts. PhD thesis, Université de Grenoble, 1998.
- [62] J. Philip, *Algorithms for the vector maximization problem*, Mathematical programming 2,207-229, 1972.
- [63] T. Radzik, *Fractional combinatorial optimization*, Handbook of Combinatorial Optimization, Edited by Z.-Z. Du and P. Pardalos, Kluwer Academic Publishers, 429-478, 1998.
- [64] R.T Rockafellar R.T *Lagrange multipliers and optimality*, SIAM Rev., 35, p. 183-238, 1993.
- [65] B. Roy et D. Bouyssou, *Aide multicritère à la décision : méthodes et cas*, éditions Economica, 1993.
- [66] B. Roy, *Paradigms and challenges*, Multiple Criteria Decision Analysis-State of the Art Surveys, Springer, 3-24, 2005.
- [67] M. Sakarovitch, *Optimisation combinatoire*, Graphes et programmation linéaire, Hermann, 1984.
- [68] S. Sayin, *Optimizing over the efficient set using a topdown search of faces*, Operations Research 48, 65-72, 2000.
- [69] A. Schrijver, *Theory of Linear and Integer Programming*, John Wiley and Sons, New-York, 1986.
- [70] S.D. Sharma, *Non-linear and dynamic Programming*, Kedar Nath Ram Nath & CO, Meerut, India, 1994.
- [71] M. Simonard, *Programmation linéaire*, 2^{ème} édition, tomes 1 et 2, Dunod, 1972.
- [72] J. Spronk, *Interactive Multiple Goal Programming : Applications to Financial Planning*, Martinus Nijhoff Publishing, Boston, 1981.
- [73] R.E. Steuer, *Multiple Criteria Optimization : Theory, Computation and Applications*, John Wiley and Sons, New-York, 1985.
- [74] K. Swarup, *Quadratic Programming*, CCERO, Belgium, 8(2), 132-136, 1966.

-
- [75] J. Teghem, *Programmation lineaire*, Ellipses, 1998.
- [76] J. Teghem and P.L. Kunsh, *A Survey Of Techniques For Finding Efficient Solutions To Multi-Objective Integer Linear Programming*, Asia-Pacific Journal of Operational Research 3 95-108, 1986.
- [77] E.L. Ulungu and J. Teghem, *Multi-objective combinatorial optimization : a survey*, Journal of Multi-Criteria Decision Analysis, 3,83-104, 1994.
- [78] R.E. Wendell and D.N. Lee *Efficiency in Multiple Objective Optimization Problems*, Mathematical Programming 12, No-3, 406-414, 1977.
- [79] P.Wolfe, *A duality theorem for nonlinear programming*, Quarterly of Applied Mathematics 19,239-244, 1961.
- [80] P.Wolfe, *Some simplex-like nonlinear programming procedures*, Operations Research 10, 438-447, 1962.
- [81] Y. Yamamoto, *Optimization over the Efficient Set : Overview*, Journal of Global Optimization.Vol.22, No1-4 ; 285-317, 2002.
- [82] M. Zeleny, *Multiple Criteria Decision Making*, McGraw-Hill, New York, 1982.
- [83] O. Zerdani and M. Moulaï, *Optimization over an Integer Efficient Set of a Multiple Objective Linear Fractional Problem*, Applied Mathematical Sciences, Vol. 5, No 50 ; 2451 - 2466, 2011.
- [84] O. Zerdani, *L'Optimisation non Linéaire Multiobjectif*, thèse de doctorat en mathématiques, option Recherche Opérationnelle, USTHB, Algérie, 2013.
- [85] A. Zellner, *Linear regression with inequality constraints on the coefficients : An application of quadratic programming and linear decision rules*. Econ Inst Netherlands School of Economics, 6109, 1961.