

N° d'ordre :08/2019-D/MT

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université des Sciences et de la Technologie Houari Boumediène USTHB  
Faculté des Mathématiques



## Thèse de Doctorat en Sciences

Présentée pour l'obtention du grade de Docteur

En : Mathématiques

Spécialité : Recherche Opérationnelle

par DRICI WASSILA

### THÈME

L'OPTIMISATION SUR L'ENSEMBLE EFFICIENT D'UN PROBLÈME  
VECTORIEL NON LINÉAIRE

Soutenue le 14 Septembre 2019, devant le jury composé de :

M. H. AIT HADDADÈNE	Professeur à l'USTHB/FMT	Président
M. M. MOULAI	Professeur à l'USTHB/FMT	Directeur de thèse
M. D. CHAABANE	Professeur à l'USTHB/FMT	Examineur
M. M. AIDENE	Professeur à l'U. Tizi Ouzou	Examineur
M. D. BENTERKI	Professeur à l'U. Sétif	Examineur
M. M. OUANES	Professeur à l'U. Tizi Ouzou	Examineur

À ma mère, mon père  
À mes adorables enfants  
À mes sœurs et à mon frère  
À ma grand-mère et mes deux tantes

# Remerciements

Mes vifs remerciements vont, en premier lieu, à mon directeur de thèse, mon guide, monsieur Mustapha Moulai, Professeur à l'USTHB, qui m'a donné la chance de réaliser ces travaux et de les mener à bien, pour la confiance qu'il m'a faite en acceptant de diriger mes recherches, 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 aide qu'il m'a apporté pour l'aboutissement de ce travail.

Mon profond respect va à Monsieur le Professeur H. AIT HADDADÈNE qui m'a honoré en acceptant de présider mon jury.

Que messieurs les membres de jury :

- Professeur M. AIDENE, enseignant chercheur à l'université de Tizi-Ouzou,
- Professeur D. BENTERKI, enseignant chercheur à la faculté des sciences de l'université de Sétif,
- Professeur D. CHAABANE, enseignant chercheur à la faculté des mathématiques de l'USTHB,
- Professeur M. OUANES, enseignant chercheur à l'université de Tizi-Ouzou,

trouvent ici l'expression de ma profonde reconnaissance pour avoir accepté d'examiner ce travail de recherche.

Je voudrais plus particulièrement exprimer ma reconnaissance envers Mme MEKHI-LEF Amel, doctorante à l'USTHB, avec qui une collaboration scientifique fructueuse s'est établie.

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.



# Table des matières

Introduction générale	9
<b>1 L'optimisation non linéaire</b>	<b>13</b>
1.1 Introduction . . . . .	13
1.2 Généralité sur les matrices . . . . .	15
1.3 La convexité en optimisation . . . . .	16
1.3.1 Fonctions convexes généralisées . . . . .	18
1.4 Conditions d'optimalité . . . . .	20
1.4.1 Conditions de Kuhn-Tucker . . . . .	21
1.5 La programmation fractionnaire linéaire . . . . .	22
1.5.1 Résultats théoriques . . . . .	23
1.5.2 Applications des problèmes fractionnaires . . . . .	23
1.5.3 Méthodes de résolution . . . . .	24
1.6 La programmation quadratique . . . . .	29
1.6.1 La programmation quadratique indéfinie . . . . .	30
1.6.2 Résultats théoriques . . . . .	31
1.6.3 Applications des problèmes quadratiques indéfinis . . . . .	31
1.6.4 Méthode de Résolution . . . . .	32
1.7 Conclusion . . . . .	35
<b>2 Généralité sur l'optimisation multiobjectif</b>	<b>37</b>
2.1 Introduction . . . . .	37
2.2 Notions fondamentales . . . . .	38
2.2.1 Position du problème . . . . .	38
2.2.2 La relation de dominance . . . . .	38
2.2.3 Optimalité de Pareto . . . . .	39
2.2.4 Détection graphique de l'efficacité . . . . .	41
2.2.5 Caractérisation des solutions efficaces . . . . .	42

2.3	Choix de la méthode d'aide à la décision . . . . .	42
2.4	Programmation linéaire multiobjectif en nombre entiers MOILP . . . . .	44
2.4.1	Formulation du problème . . . . .	44
2.4.2	Solutions supportées et non supportées . . . . .	45
2.4.3	Quelques méthodes de résolution d'un problème <i>MOILP</i> . . . . .	46
2.5	Conclusion . . . . .	54
<b>3</b>	<b>Une méthode de résolution des problèmes d'optimisation multiobjectif quadratique indéfinie en nombres entiers MOIIQP</b>	<b>55</b>
3.1	Introduction . . . . .	55
3.2	Notations et définitions . . . . .	56
3.2.1	La coupe efficace . . . . .	59
3.3	Développement de la méthode . . . . .	59
3.3.1	Description de la méthode . . . . .	59
3.3.2	Formulation de l'algorithme . . . . .	61
3.3.3	Résultats théoriques . . . . .	62
3.4	Exemple Illustratif . . . . .	63
3.5	Expérimentation et résultats . . . . .	75
3.5.1	Discussion des résultats . . . . .	76
3.6	Conclusion . . . . .	77
<b>4</b>	<b>Optimisation d'un critère sur l'ensemble efficient</b>	<b>79</b>
4.1	Introduction . . . . .	79
4.2	Formulation du problème . . . . .	80
4.3	Méthodes de résolution de $(P_E)$ dans le cas discret . . . . .	81
4.3.1	Le cas (Linéaire-Linéaire) . . . . .	81
4.3.2	Le cas (Fractionnaire-Linéaire) . . . . .	85
4.4	Conclusion . . . . .	88
<b>5</b>	<b>Méthodes d'optimisation d'un critère non linéaire sur l'ensemble efficient d'un problème multiobjectif linéaire discret</b>	<b>89</b>
5.1	Introduction . . . . .	89
5.2	Optimisation d'une fonction quadratique indéfinie sur l'ensemble efficient d'un problème multiobjectif linéaire discret . . . . .	91
5.2.1	Définitions et notations . . . . .	91
5.2.2	La coupe efficace . . . . .	93
5.2.3	Test d'efficacité . . . . .	94

5.2.4	Développement de la méthode . . . . .	94
5.2.5	Exemple illustratif . . . . .	98
5.2.6	Expérimentation et résultats . . . . .	104
5.3	Optimisation d'une fonction fractionnaire linéaire sur l'ensemble efficient d'un problème multiobjectif linéaire discret . . . . .	107
5.3.1	Définitions et notations . . . . .	107
5.3.2	Les coupes utilisées . . . . .	109
5.3.3	Test d'efficacité . . . . .	110
5.3.4	Développement de la méthode . . . . .	110
5.3.5	Exemple illustratif . . . . .	114
5.3.6	Expérimentation et résultats . . . . .	123
5.4	Conclusion . . . . .	125
	<b>Conclusion Générale</b>	<b>127</b>
	<b>Bibliographie</b>	<b>129</b>



# Introduction générale

Les ingénieurs se heurtent quotidiennement à des problèmes technologiques de complexité grandissante, qui surgissent dans des secteurs très divers, comme dans le traitement des images, la conception de systèmes mécaniques, la recherche opérationnelle et l'électronique. Le problème à résoudre peut fréquemment être exprimé sous la forme générale d'un problème d'optimisation, dans lequel on définit une fonction objectif, que l'on cherche à minimiser (ou à maximiser). La définition du problème d'optimisation est souvent complétée par la donnée de contraintes : tous les paramètres (ou variables de décision) de la solution proposée doivent respecter ces contraintes, faute de quoi la solution n'est pas réalisable.

Il existe de nombreuses méthodes d'optimisation "classiques" pour résoudre de tels problèmes, applicables lorsque certaines conditions mathématiques sont satisfaites : ainsi, la programmation linéaire traite efficacement le cas où la fonction objectif, ainsi que les contraintes, s'expriment linéairement en fonction des variables de décision. Malheureusement, les situations rencontrées en pratique comportent souvent une ou plusieurs complications, qui mettent en défaut ces méthodes : par exemple, la fonction objectif peut être non linéaire, ou encore, le problème peut exiger la considération simultanée de plusieurs objectifs contradictoires.

La difficulté principale de problèmes à plusieurs objectif est liée à la présence de conflits entre ces divers objectifs pouvant être linéaires ou non [37, 43, 54, 31, 63, 77]. 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 multiobjectif 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 inévi-

table, on parle alors d'optimisation discrète ou programmation en nombres entiers. De ce fait, la programmation mathématique multiobjectif en nombre entiers représente le cadre général de notre travail dans cette thèse.

Dans le cadre de la programmation mathématique multiobjectif, nous nous sommes particulièrement intéressés au cas où chaque fonction objectif est le produit de deux fonctions affines et que l'ensemble réalisable est non convexe. Le problème, noté *MOIIQP*, consiste à déterminer toutes les solutions efficaces d'un problème multiobjectif quadratique indéfinie à variables entières. Un algorithme basé sur la technique de Branch & Cut a été présentée pour résoudre ce type de problème. La méthode a été conçu à l'aide de la technique Branch & Bound pour la programmation en nombres entiers, de la coupe efficace [30] et de la technique du simplexe modifié pour la résolution des problèmes de maximisation quadratique indéfinie mono-objectif [93]. Ce travail a fait l'objet d'une première publication dans la revue "International Optimization Methods and Software" [40].

Une recherche bibliographique nous a permis de conclure que la résolution du problème *MOIIQP* n'avait pas reçu autant d'attention que les problèmes linéaires multiobjectif discret et jusqu'à ce jour, il n'existe pas d'article résolvant le problème (*MOIIQP*) mis à part l'article [9] où le problème est résolu en utilisant une approche de coupe plane.

L'ensemble efficient d'un problème multiobjectif 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 multiobjectif 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 efficientes comme outil pour mesurer les préférences des décideurs et distinguer parmi les nombreuses solutions efficaces.

Ce problème est en principe 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 de l'optimisation d'un critère linéaire sur l'ensemble des points efficaces d'un problème multiobjectif sont relativement peu nombreuses et quasi-inexistantes pour le cas d'un critère non linéaire. Cet état de choses, bien que source de difficultés potentielles, nous a d'avantage 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 efficient d'un problème

multiobjectif linéaire discret *MOILP* [74], [5], [56], [25].

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 efficient d'un problème *MOILP* [64], le cas (Linéaire - Fractionnaire) qui optimise un critère linéaire sur un ensemble efficient d'un problème multiobjectif fractionnaire linéaire discret *MOILFP* [99], le cas (Fractionnaire - Fractionnaire) optimisant un critère fractionnaire linéaire sur l'ensemble efficient *MOILFP* [100], et le cas (Linéaire - Quadratique) qui, quant à lui, optimise un critère linéaire sur l'ensemble efficient d'un problème multiobjectif quadratique convexe [14].

Notre deuxième contribution consiste à mettre au point une nouvelle méthode exacte qui maximise une fonction quadratique indéfinie sur l'ensemble efficient d'un problème multiobjectif linéaire en nombres entiers *MOILP*, évitant le passage systématique par toutes les solutions efficaces du problème multiobjectif *MOILP*, ce travail a fait l'objet d'une publication dans la revue internationale "Optimization" [71]. Une extension de cette méthode a été réalisée pour permettre de résoudre les problèmes d'optimisation d'un critère fractionnaire linéaire sur l'ensemble efficient d'un problème *MOILP*, objet d'une autre publication sur la revue "Annals of Operations Research" [41].

Les deux méthodes développées sont basées sur le principe de Branch & Cut. Le processus de stérilisation est renforcé par des coupes ainsi que par un test d'efficacité permettant d'éviter un grand nombre de solutions réalisables non efficaces. Un étude expérimentale des deux approches est rapportée pour évaluer les performances des méthodes proposées.

La thèse est organisée de la façon suivante :

- Le chapitre 1 intitulé "L'optimisation non linéaire", expose l'essentiel de l'optimisation non linéaire nécessaire à la compréhension du reste de cette thèse. On présente en particulier l'optimisation quadratique indéfinie et l'optimisation fractionnaire linéaire, nous présentons les principales approches de résolution proposées dans la littérature pour ces différents problèmes.
- Le chapitre 2 intitulé "L'optimisation multiobjectif" expose le cadre générale de notre travail, nous présentons pour cela l'essentiel des définitions et résultats reliés à l'optimisation multiobjectif et nous parlerons de quelques travaux récents existant en littérature traitant les problèmes de l'optimisation multiobjectif linéaire discrète.
- Le chapitre 3 expose notre contribution et présente une méthode de résolution

des programmes d'optimisation multiobjectif quadratiques indéfinies en nombres entiers *MOIIQP*. Une étude comparative est rapportée pour évaluer les performances de la méthode proposée avec celle proposée par Arora [9].

- Le chapitre 4 intitulé "Optimisation d'un critère sur l'ensemble efficient" aborde la problématique de l'optimisation d'un critère sur l'ensemble efficient d'un problème multiobjectif linéaire discret et nous détaillons quelques méthodes de résolution pour le cas (Linéaire-Linéaire) et une méthode de résolution pour le cas (Fractionnaire-Linéaire).
- Dans le chapitre 5, deux méthodes exactes sont proposées ; une pour résoudre le problème d'optimisation d'un critère quadratique indéfini sur l'ensemble efficient d'un problème multiobjectif linéaire discret *MOILP* et l'autre pour résoudre le problème d'optimisation d'un critère fractionnaire linéaire sur l'ensemble efficient d'un problème MOILP. Les deux méthodes ont été programmées à l'aide du logiciel Matlab afin d'évaluer leurs performances.
- Enfin, le mémoire s'achèvera par une conclusion générale et quelques perspectives de recherche.

# Chapitre 1

## L'optimisation non linéaire

### 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 : [62, 73, 78, 87, 94, 91].

Il existe des algorithmes polynômiaux efficaces pour résoudre un programme linéaire comme ceux dits de points intérieurs initiés par Karmarkar [57] en 1984 et l'algorithme du simplexe introduit en 1947 par G.B.Dantzig [34].

En termes de complexité des algorithmes, l'algorithme du simplexe est théoriquement exponentiel alors qu'il se comporte dans la pratique comme un algorithme polynômial. Il se 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.

Nous illustrons la difficulté de la recherche de la solution optimale dans le cas non linéaire par l'exemple suivant :

**Exemple :** 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.

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

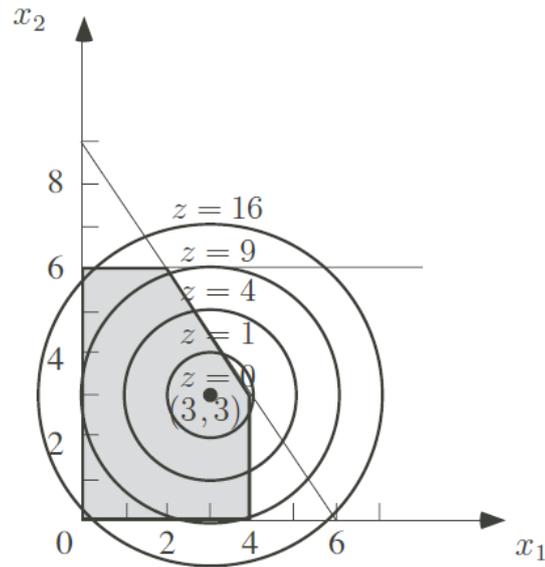


FIGURE 1.1 – Solution intérieure

Dans ce 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 sont la programmation fractionnaire et la programmation quadratique dont nous allons détailler dans ce qui suit, les différents outils pour résoudre ces types de programmes.

## 1.2 Généralité sur les matrices

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

- $A$  est dite définie positive, si  $x^t A x > 0$ ,  $\forall x \in \mathbb{R}^n$ ,  $x \neq 0$ .
- $A$  est dite définie négative, si  $x^t A x < 0$ ,  $\forall x \in \mathbb{R}^n$ ,  $x \neq 0$ .
- $A$  est dite semi-définie positive, si  $x^t A x \geq 0$ ,  $\forall x \in \mathbb{R}^n$  et  $\exists x \in \mathbb{R}^n$  tel que  $x^t A x = 0$ ,  $x \neq 0$ .
- $A$  est dite semi-définie négative, si  $x^t A x \leq 0$ ,  $\forall x \in \mathbb{R}^n$  et  $\exists x \in \mathbb{R}^n$  tel que  $x^t A x = 0$ ,  $x \neq 0$ .
- $A$  est dite indéfinie, si  $x^t A x$  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 :  $\nabla_1(A) = a_{11}$ ,  $\nabla_2(A) = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$ , ...,  $\nabla_n(A) = \det(A)$

**Définition 1.2.3** Soit  $\lambda \in \mathbb{R}$ ,  $\lambda$  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. i.e,  $\nabla_j(A) > 0 \forall j = \overline{1, n}$ .
- $A$  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.
- $A$  est semi-définie négative si et seulement si toutes les valeurs propres de  $A$  sont négatives ou nulles.

## 1.3 La convexité en optimisation

**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 existent des coefficients réels  $\lambda_i$  ;  $i \in \{1, \dots, p\}$ , 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 :

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

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  $\beta$  finie et positive telle que :

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

- 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 :

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

**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 convexe (resp. strictement concave)  $\Leftrightarrow (-f)$  est concave (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'intérieur 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  $\bar{x} \in \text{int}(S)$  s'il existe un vecteur gradient noté  $\nabla f(\bar{x})$  et une fonction  $\alpha : \mathbb{R}^n \rightarrow \mathbb{R}$  tels que :*

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

$$\text{où } \lim_{x \rightarrow \bar{x}} \alpha(\bar{x}; x - \bar{x}) = 0 \text{ et } \nabla f(\bar{x}) = \left( \frac{\delta f(\bar{x})}{\delta x_1}, \dots, \frac{\delta f(\bar{x})}{\delta x_n} \right)^t.$$

**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 :

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

**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 :

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

**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.

## 1.4 Conditions d'optimalité

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 1.4.1 (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 1.4.2 (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. 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 1.4.1 (Condition 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 (maximum local) alors il est un minimum global pour le problème  $(P_m)$ .

**Théorème 1.4.2** 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  $\nabla f(\bar{x}) = 0$ , alors  $\bar{x}$  est un minimum global de  $f$  sur  $S$ .

**Théorème 1.4.3** Soit  $f : S \rightarrow \mathbb{R}$  une fonction quasiconcave 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 1.4.4** 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 1.4.2, avec les fonctions convexes.

**Théorème 1.4.5** 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  $\nabla f(\bar{x}) = 0$ , alors  $\bar{x}$  est un minimum global de  $f$  sur  $S$ .

### 1.4.1 Conditions de Kuhn-Tucker

Considérons le problème de minimisation  $(P)$  :

$$(P) \begin{cases} \min f(x) \\ sc \\ h_i(x) = 0, \quad i = 1, \dots, m_1; \\ g_j(x) \leq 0, \quad j = 1, \dots, m_2; \\ x \geq 0. \end{cases}$$

On suppose que  $f$  et  $h_i, i = 1, \dots, m_1$  et  $g_j, j = 1, \dots, m_2$ , sont différentiables sur un domaine de  $\mathbb{R}^n$ .

**Définition 1.4.3** • Une contrainte  $g_j(x) \leq 0$  est dite active (ou saturée) en  $x^*$  si  $g_j(x^*) = 0$  et elle est dite inactive si  $g_j(x^*) < 0$ .

- Une contrainte  $h_i(x) = 0$  est dite active (ou saturée) en  $x^*$  si elle est vérifiée en  $x^*$ , c-à-d  $h_i(x^*) = 0$ .
- L'ensemble des indices des contraintes actives en  $x^*$  est noté  $I(x^*)$ .

**Définition 1.4.4** On appelle Lagrangien associé au problème (P) la fonction définie par :

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^{m_1} \lambda_i h_i(x) + \sum_{j=1}^{m_2} \mu_j g_j(x); \quad \lambda \in \mathbb{R}^{m_1}, \mu \in \mathbb{R}^{m_2}; \mu \geq 0$$

Les  $\lambda_i$  pour  $i = 1, \dots, m_1$  et  $\mu_j$  pour  $j = 1, \dots, m_2$  sont appelés multiplicateurs de Lagrange.

**Définition 1.4.5 (Contraintes qualifiées)** Les contraintes du problème (P) sont dites qualifiées au point  $x^*$  si  $h_i$  et  $g_j$  sont différentiables au point  $x^*$  et si les gradients des contraintes actives en  $x^*$  sont linéairement indépendants.

Le point  $x^*$  est dit dans ce cas point régulier.

**Théorème 1.4.6 (CN d'optimalité de Karush-Kuhn-Tucker)** Soit  $x^* \in S$  un point réalisable du problème (P). Supposons  $f$ ,  $g_j$  et  $h_i$  sont différentiables en  $x^*$  et les contraintes sont qualifiées au point  $x^*$ . Si  $x^*$  est un minimum local de  $f$  sur  $S$  alors il existe  $\lambda^* \in \mathbb{R}^{m_1}$  et  $\mu^* \in \mathbb{R}^{m_2}$  tels que :

$$\begin{cases} \nabla L(x^*, \lambda^*, \mu^*) = 0; \\ h_i(x^*) = 0, \quad i = 1, \dots, m_1; \\ \mu_j^* g_j(x^*) = 0, \quad j = 1, \dots, m_2; \\ g_j(x^*) \leq 0, \quad j = 1, \dots, m_2; \\ \mu_j^* \geq 0, \quad j = 1, \dots, m_2, \quad x^* \geq 0. \end{cases} \quad (\text{KT})$$

Si de plus le problème (P) est convexe, alors les conditions de KKT sont suffisantes pour que  $x^*$  soit un minimum global de  $f$ .

## 1.5 La programmation fractionnaire linéaire

Les programmes fractionnaires consistent à optimiser un objectif mis sous la forme d'un rapport de deux fonctions linéaires ou non, soumis à un ensemble de contraintes. Différentes versions de ce modèle, en variables entières, en mixtes, en continues ou en variables bivalentes ont plusieurs applications à savoir l'ingénierie, l'optimisation combinatoire, la programmation stochastique, l'informatique et l'économie (voir travaux de Stancu-Minasian [88, 89, 90] et Schaible [80, 81, 82, 83]).

Nous nous intéressons dans notre travail au problème de la programmation fractionnaire linéaire (*LFP*). La forme particulière de ce programme a fait que de nombreux auteurs ont élaboré des méthodes de résolution spécifiques qui se sont avérées plus efficaces que l'application directe de méthodes générales de la programmation non linéaire. Trois grandes stratégies de résolution émergent dans la littérature : la résolution directe, la résolution par paramétrisation et la résolution d'un problème équivalent à objectif simplifié que nous allons présenter brièvement dans ce chapitre.

Le problème de programmation fractionnaire linéaire, ou encore hyperbolique, peut se formuler comme suit :

$$(LFP) \begin{cases} \max f(x) = \frac{p^T x + \alpha}{q^T x + \beta} \\ sc. \\ x \in S. \end{cases} \quad (1.1)$$

où  $\alpha$  et  $\beta$  sont des réels,  $p$  et  $q$  sont des vecteurs de  $\mathbb{R}^n$ .  $S = \{x \in \mathbb{R}^n | Ax \leq b, x \geq 0\}$ , avec  $A$  une  $(m \times n)$ -matrice réelle et  $b$  un vecteur de  $\mathbb{R}^m$ .

On suppose que

- L'ensemble  $S$  est borné non vide.
- $q^T x + \beta > 0 \forall x \in S$ .
- $\exists x \in S : p^T x + \alpha > 0$ .

Si les variables sont astreintes à ne prendre que des valeurs entières, le programme est noté (*ILFP*) et il est appelé programme fractionnaire linéaire en nombres entiers ou encore programme hyperbolique discret.

**Lemme 1.5.1** [29] *La fonction  $f(x) = \frac{p^T x + \alpha}{q^T x + \beta}$  est à la fois pseudoconvexe et pseudoconcave sur  $S$ .*

### 1.5.1 Résultats théoriques

D'après ce lemme 1.5.1, les sous-section 1.3.1 et 1.4, on peut citer quelques conséquences pour le problème de programmation linéaire fractionnaire :

- Puisque  $f$  est à la fois pseudoconvexe et pseudoconcave sur  $S$  donc, d'après le théorème 1.4.2, elle est aussi quasiconvexe, strictement quasiconvexe, quasiconcave et strictement quasiconcave sur  $S$ .

- Puisque  $f$  est à la fois quasiconvexe et quasiconcave donc elle est quasilinéaire.
- Puisque la fonction objectif  $f$  est strictement quasiconvexe et strictement quasiconcave sur  $S$  alors, d'après le théorème 1.4.4, un minimum (maximum) local est aussi un minimum (maximum) global sur  $S$  respectivement.
- Puisque  $f$  est quasiconvexe et quasiconcave et si  $f$  est continue sur  $S$  alors, d'après le théorème 1.4.3, au moins une solution optimale du problème ( $PLF$ ) est atteinte en un point extrême de  $S$ .
- Puisque  $f$  est pseudoconvexe et pseudoconcave sur  $S$  donc, d'après le théorème 1.4.5, si  $\nabla f(x) = 0$ , alors  $x$  est un minimum (maximum) global de  $f$  sur  $S$ .

### 1.5.2 Applications des problèmes fractionnaires

Plusieurs applications des programmes mathématiques fractionnaires ont déjà été décrites dans la littérature [37, 82, 83]. Comme domaines d'applications des problèmes fractionnaires soumis à un ensemble de contraintes, nous pouvons citer par exemple :

- Le domaine des finances où il est souvent question d'optimiser un rapport de deux fonctions telles que [dette/capitaux propres], [rendement/employé], [coût effectif/coût standard], [bénéfice/coût], [inventaire/ventes], [risque des actifs/capital].
- Le domaine de l'économie offre un large éventail d'applications. En effet, la mesure de l'efficacité des systèmes étudiées s'exprime sous forme de rapports entre les critères techniques et/ou économiques. Par exemple, [rendement/risque], [rendement/investissement], [coût/temps], [dette/capitaux propres], [inventaire/ventes].
- Le domaine de la santé comme la planification dans un hôpital [coût/patient], [infirmière/patient], [docteur/patient]...etc
- Les processus de décision de Markov peuvent également mener à la maximisation du rapport de la moyenne et l'écart type.
- Dans la théorie de l'information, la capacité d'un canal de transmission peut être définie comme étant le taux maximal de transmission sur toutes les probabilités.
- Programmation stochastique : Dans un programme linéaire, en supposant que les composantes de l'objectif sont non constantes, indépendantes et varient suivant une loi de probabilité donnée. La programmation stochastique se propose de maximiser la probabilité pour que la valeur de l'objectif soit supérieure à une valeur donnée.

Lorsque les composantes des fonctions objectifs considérées comme des variables aléatoires, suivant une loi de Gauss. Berau [21] a montré que le programme stochastique est équivalent à un programme déterministe fractionnaire linéaire convexe en variables continues.

### 1.5.3 Méthodes de résolution

#### Résolution directe :

Cette section traite des méthodes qui résolvent le programme fractionnaire sous sa forme originale, c'est-à-dire sans modifier ni l'objectif ni l'ensemble des contraintes. Cette approche est utilisée pour résoudre les programmes hyperboliques tant qu'en variables continues qu'en variables entières (en particulier en 0–1).

#### Résolution d'un programme hyperbolique continu :

Plusieurs approches furent élaborées [37, 22, 50, 23] pour la recherche d'une solution optimale d'un programme hyperbolique en variables continues. Parmi les plus récentes, celle de A. Cambini et al. [23]. Les grandes lignes de cette méthode sont présentées ci-dessous :

**La méthode de Cambini et Martein [23] :** Rappelons le programme hyperbolique continu (*LFP*) :

$$(LFP) \begin{cases} \max f(x) = \frac{p^T x + \alpha}{q^T x + \beta} \\ sc. \\ x \in S = \{x \in \mathbb{R}^n | Ax \leq b; x \geq 0\}. \end{cases} \quad (1.2)$$

où  $\alpha$  et  $\beta$  sont des réels,  $p$  et  $x$  des vecteurs de  $\mathbb{R}^n$ ,  $q$  un vecteur non nul 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 et  $q^T x + \beta > 0$  sur  $S$ .

**Définition 1.5.1** *Un point réalisable  $\bar{x}$  est une solution niveau optimale pour le problème (*LFP*) si  $\bar{x}$  est une solution optimale pour le programme linéaire suivant :*

$$(LFP_{\bar{x}}) \begin{cases} \max(p^T x + \alpha) \\ sc. \\ x \in S \text{ et } qx = q\bar{x}. \end{cases} \quad (1.3)$$

*Si de plus  $\bar{x}$  est un point extrême de  $S$ ,  $\bar{x}$  est dit solution niveau optimale de base.*

L'algorithme de (Cambini et al.) [23], génère une séquence finie de solutions niveau optimales dont la première  $x^0$  est une solution optimale du programme linéaire :

$$\begin{cases} \min(q^T x + \beta) \\ sc. \\ x \in S. \end{cases} \quad (1.4)$$

- Si  $x^0$  est unique, alors elle est aussi une solution niveau optimale de base pour (LFP).
- Sinon, résoudre le programme linéaire (LFP $_{x^0}$ ) pour obtenir une solution niveau optimale de base.

**Théorème 1.5.1** [66] *Le point  $x^0$  de  $S$  est une solution optimale du problème linéaire fractionnaire (LFP) si et seulement si le vecteur gradient réduit  $\bar{\gamma} = \bar{\beta} \cdot \bar{p} - \bar{\alpha} \cdot \bar{q}$  est tel que  $\bar{\gamma}_j \leq 0$  pour tout indice hors base  $j \in N$ .*

---

**Algorithme 1 : CAMBINI ET MARTEIN**

---

**Étape 1 :** Trouver une solution niveau optimale  $x^1$ .

- Si une telle solution n'existe pas, alors  $\max_{x \in S} f(x) = +\infty$
- sinon, poser  $k = 1$  et aller à l'Étape 2.

**Étape 2 :** • Si  $J = \{j | \bar{q}_j > 0\} = \emptyset$ , terminer  $x^k$  est une solution optimale du problème (LFP) ;

- Sinon, soit  $s$  tel que  $\frac{\bar{p}}{\bar{q}} = \max_{j \in J} \left( \frac{\bar{p}_j}{\bar{q}_j} \right)$  ;
  - Si  $\gamma_s > 0$ , aller à l'Étape 3 ;
  - Sinon, terminer ;  $x^k$  est une solution optimale de (LFP).

**Étape 3 :** La variable hors base  $x_s$  entre dans la base en moyennant une opération pivot,

poser  $k = k + 1$  et aller à l'Étape 2.

Si une telle opération n'est pas possible,  $\max_{x \in S} f(x) = \frac{\bar{p}_s}{\bar{q}_s}$  ; terminer.

---

### Résolution d'un programme hyperbolique discrêt

De nombreux auteurs ont résolu le problème de la programmation fractionnaire linéaire en nombres entiers par la méthode directe en utilisant différents algorithmes tels par exemple les techniques de séparation et évaluation progressive [1, 7, 26, 2] et les coupes planes, comme celle de D. Granot et al. [49].

**Méthode Granot et al.[49]** L'algorithme de Granot et al. s'applique directement au programme (ILFP) suivant :

$$(ILFP) \begin{cases} \max f(x) = \frac{p^T x + \alpha}{q^T x + \beta} \\ sc. \\ x \in S = \{x \in \mathbb{R}^n | Ax \leq b, x \geq 0\}. \quad x^n \text{ entier} \end{cases} \quad (1.5)$$

où  $\alpha$  et  $\beta$  sont des réels,  $p$  et  $x$  des vecteurs de  $\mathbb{R}^n$ ,  $q$  un vecteur non nul 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 et  $q^T x + \beta > 0$  sur  $S$ .

Dans cette méthode, la structure originale des contraintes est maintenue et les itérations sont réalisées dans un tableau du simplexe augmenté qui comprend  $m$  et 3 lignes supplémentaires. Les  $m$  premières lignes correspondent aux contraintes d'origine, les lignes  $(m + 1)$ ,  $(m + 2)$  correspondent au numérateur et dénominateur de la fonction objectif fractionnaire du problème (ILFP) et la ligne  $(m + 3)$  correspond au vecteur du gradient réduit  $\bar{\gamma}_j$  de la fonction objectif où  $\bar{\gamma} = \bar{\beta} \cdot \bar{p}_j - \bar{\alpha} \cdot \bar{q}_j, \forall j \in \{1, \dots, m\}$ .

A 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 gradient réduit citée ci-dessus.

Une fois les valeurs du gradient réduit pour les variables hors base calculées, on teste :

- Si  $\bar{\gamma}_j \leq 0$ , pour tout indice hors base  $j$ , alors la solution obtenue est une solution optimale du problème (ILFP)
- Sinon, il existe un indice  $s, s \in N$  ( $N$  étant l'ensemble des indices hors base) pour lequel  $\bar{\gamma}_s > 0$ . Soit  $\mu_r$  tel que :

$$\mu_r = \min \left\{ \frac{\bar{b}_i}{\bar{A}_i^s} \mid \bar{A}_i^s > 0 \right\}$$

alors la ligne  $r$  est comme ligne pivot pour générer une coupe de la forme suivante :

$$v + \sum_{j \in N} \left[ \frac{\bar{A}_{rj}}{\bar{A}_r^s} \right] \cdot x_j = \left[ \frac{\bar{b}_i}{\bar{A}_r^s} \right], \quad v \geq 0$$

Cette coupe est rajoutée au tableau du simplexe pour servir comme ligne pivot, avec la colonne  $s$  comme colonne pivot. Sachant que la valeur du pivot dans ce cas est de 1, les nouveaux coefficients obtenus après l'exécution des opérations pivot usuelles sont tous des nombres entiers.

## Méthode par paramétrisation

A l'inverse de la résolution directe, on construit un problème à objectif simplifié, combinaison linéaire du numérateur et du dénominateur par l'intermédiaire d'un paramètre, tout en gardant inchangé l'ensemble des contraintes [72].

Une séquence de résolutions de ce type de problème fournit une solution du programme fractionnaire. Cette méthode a été utilisée pour les différents programmes fractionnaires linéaires et non linéaires, en variables continues comme en variables discrètes, sur des domaines bornés.

Afin de simplifier l'objectif du programme mathématique fractionnaire, un paramètre est introduit permettant par exemple de ramener un programme hyperbolique en un programme paramétré linéaire, tout en gardant l'ensemble des contraintes inchangé. Ainsi le programme obtenu peut être résolu paramétriquement, une séquence de résolutions de tels programmes à objectif simplifié engendre une suite de solutions convergeant vers une solution optimale du programme fractionnaire initial.

Appliquons l'algorithme de Dinkelbach [37] directement sur le programme hyperbolique. Le programme paramétré  $(Q_\lambda)$  associé consiste à simplifier l'objectif en combinant linéairement le numérateur et le dénominateur par l'intermédiaire d'un paramètre,  $\lambda \in \mathbb{R}$ . Dans le cas d'un programme hyperbolique,  $(Q_\lambda)$  est un programme linéaire dont l'objectif est fonction du paramètre  $\lambda$ , et il est défini comme suit :

$$(Q_\lambda) \begin{cases} \max v(\lambda) = (p - \lambda q)^T x + (\alpha - \lambda \beta) \\ sc. \\ x \in S = \{x \in \mathbb{R}^n | Ax \leq b, x \geq 0\}. \end{cases} \quad (1.6)$$

En notant,  $\lambda^*$  la valeur optimale de  $(LFP)$ , le résultat fondamental liant le programme fractionnaire au programme paramétré associé est donné par :

**Proposition 1.5.1** [37] *Toute solution optimale de  $(Q_{\lambda^*})$  est une solution optimale de  $(LFP)$ .*

En tant que fonction de la variable  $\lambda$ , la valeur optimale  $v(\lambda)$  du programme paramétré vérifie quelques propriétés [37] que nous résumons ci-après :

**Proposition 1.5.2** *La fonction  $\lambda \rightarrow v(\lambda)$  est continue, strictement décroissante, convexe.*

*$v(0) > 0$  et  $v(\lambda)$  tend vers  $-\infty$  quand  $\lambda$  tend vers  $+\infty$ .*

*Si de plus le programme est hyperbolique, alors  $v$  est linéaire par morceaux.*

En particulier, l'équation  $v(\lambda) = 0$  admet une solution unique  $\lambda^*$ , plus précisément :

**Proposition 1.5.3** [37]

- $v(\lambda) = 0 \Leftrightarrow \lambda = \lambda^*$
- $v(\lambda) > 0 \Leftrightarrow \lambda < \lambda^*$
- $v(\lambda) < 0 \Leftrightarrow \lambda > \lambda^*$

Ainsi la résolution du programme fractionnaire (*LFP*) revient à trouver la racine de l'équation non linéaire à une seule variable,  $v(\lambda) = 0$ .

Pour ce faire, un catalogue d'algorithmes de la littérature est présenté dans Nagih et al. [72]. Il inclut des algorithmes de résolution itératifs (Newton, interpolation linéaire) et dichotomiques (recherche dichotomique, recherche dichotomique modifiée et interpolation convexe) ainsi qu'un algorithme  $\epsilon$ -approchant.

## Résolution d'un programme équivalent à objectif non fractionnaire

Un changement de variables permet lui aussi de simplifier l'objectif, mais en transportant la difficulté sur l'ensemble des contraintes [72]. Par exemple, un programme fractionnaire concave-convexe est transformé en un programme concave, et un programme hyperbolique en un programme à objectif linéaire, avec des contraintes additionnelles. La transformation du programme linéaire fractionnaire en un programme équivalent à objectif non fractionnaire est obtenu par un changement de variables. A l'inverse de l'approche paramétrée, ce changement de variables induit l'ajout d'une contrainte et d'une variable. Plus précisément, cette transformation, proposée par Charnes et al. [27] s'applique au programme hyperbolique (*LFP*) en variables continues et s'effectue en introduisant deux nouvelles variables :

$$y = \frac{1}{qx + \beta}x \text{ et } t = \frac{1}{qx + \beta}$$

pour obtenir un programme linéaire équivalent :

$$\left\{ \begin{array}{l} \max = p^T y + \alpha t \\ sc. \\ Ay - \beta t \leq 0 \\ q^T y + \beta t = 1 \\ y, t \geq 0. \end{array} \right. \quad (1.7)$$

Cette notion d'équivalence est précisée par :

**Proposition 1.5.4** [27] *Si  $(y^*; t^*)$  est une solution optimale de (LFP), alors  $t^* > 0$  et  $x^* = \frac{y^*}{t^*}$  est une solution optimale de (LFP).*

## 1.6 La programmation quadratique

Les programmes quadratiques sont parmi les programmes non linéaires les plus proches de la programmation linéaire, ils représentent une transition naturelle de la théorie de la programmation linéaire vers la théorie de la programmation non linéaire, cependant, il existe 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 [98], la production efficace [39], la sélection de portefeuille [65] ...

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 :

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

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.

C'est le cas d'une fonction quadratique indéfinie que nous allons nous intéresser et exploiter dans notre thèse.

### 1.6.1 La 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 ( $IQP$ ) 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.

Ainsi, notre problème, noté ( $IQP$ ), peut être écrit sous la forme suivante :

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

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

**Théorème 1.6.1** [93] *La fonction  $f(x) = (p^t x + \alpha)(q^t x + \beta)$  est strictement quasiconcave sur  $\mathcal{X}$  convexe.*

### 1.6.2 Résultats théoriques

Les programmes quadratiques quasiconcaves 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 1.4.3, 1.4.4, des conséquences pour un problème de programmation quadratique  $IQP$  peuvent être cités :

- Puisque la fonction  $f$  est strictement quasiconcave sur  $\mathcal{X}$  alors d'après le théorème 1.4.4 maximum local du problème ( $IQP$ ) est un maximum global.
- Puisque  $f$  est quasiconcave et continue sur  $\mathcal{X}$  alors, d'après le théorème 1.4.3, une solution optimale du problème ( $IQP$ ) est atteinte en un point extrême de  $\mathcal{X}$ .

### 1.6.3 Applications des problèmes quadratiques indéfinis

Une application pour ce type de programme peut être trouvée dans la théorie microéconomique [32], [53]; 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 la consommation est limitée par les niveaux de production des produits et par conséquent il est utile d'inclure des contraintes linéaires supplémentaires relatives aux ressources et aux capacités.

### 1.6.4 Méthode de Résolution

De nombreuses méthodes exactes ont été proposées pour la résolution des problèmes modélisés sous forme de ( $IQP$ ), nous pouvons citer en particulier les méthodes proposées par K. Swarup[93], S.D. Sharma[84], C.R. Bector et M. Dahl [11], et P. Anand [8].

Dans un autre travail, Swarup [93] a montré que la fonction quadratique indéfinie est explicitement quasiconcave et que, par conséquent, un minimum local se situe en un point extrême de l'ensemble réalisable et a fourni une méthode similaire à la méthode du simplexe pour trouver un minimum local. Bector et Dahl [11] ont également présenté une technique similaire à la méthode du simplexe pour résoudre le problème  $IQP$ . Erenguc et Benson [46] ont proposé un algorithme pour trouver le minimum global d'une fonction quadratique indéfinie sur des entiers contenus dans un ensemble convexe compact en transformant le problème en un problème équivalent avec une fonction objectif séparable. Li et al.[61] ont proposé un algorithme basé sur la relaxation lagrangienne alors que Sun et al. [92] ont proposé une relaxation convexe et une décomposition lagrangienne pour résoudre les problèmes de programmation quadratique indéfini en nombres entiers .

On considère le problème ( $IQP$ ) suivant :

$$(IQP) \begin{cases} \max & f(x) = (p^t x + \alpha)(q^t x + \beta) = f_1(x).f_2(x) \\ & sc \\ & \mathcal{X} = \{x \in \mathbb{R}^n / Ax \leq b, x \geq 0\} \end{cases} \quad (1.8)$$

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

La fonction  $f$  étant explicite quasi-concave dans  $\mathcal{X}$ , le maximum de  $f$  sera pris à un sommet de  $\mathcal{X}$  et le maximum local atteint sera le maximum global [93]. Par conséquent, la technique du simplexe [12] peut être appliquée pour obtenir une solution optimale du problème 1.8.

La méthode Simplexe [35] fournit un algorithme systématique qui consiste à passer d'une solution réalisable de base correspondant à un sommet du polyèdre  $\mathcal{X}$  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.

La méthode proposée pour résoudre le problème (*IQP*) [93] 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  $m$  premières 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 (*IQP*) et la ligne  $m+3$  correspond au vecteur de la direction de croissance de la fonction  $f(x)$ , noté  $\gamma_j$ , calculés pour les variables hors base dont la formule sera donnée dans ce qui suit.

### La méthode variante du simplexe pour la résolution des programmes quadratique indéfinie

Soit  $x_B$  une solution réalisable de base pour le problème (1.8) avec une matrice de base correspondante  $B$  tel que :

$$x_B = B^{-1}b.$$

$I$  et  $N$  représentent, respectivement, l'ensemble des indices de variables de base et l'ensemble des indices de variables hors bases correspondant à  $x_B$  tel que :

$$\begin{aligned} I &= \{i | a_i \in B\}, \\ N &= \{j | a_j \notin B\}. \end{aligned}$$

Soit  $f(x_B) = f_1(x_B)f_2(x_B)$  tel que

$$\begin{aligned} f_1(x_B) &= p_B^T x_B + \alpha = p_B B^{-1} b + \alpha, \\ f_2(x_B) &= q_B^T x_B + \beta = q_B B^{-1} b + \beta. \end{aligned}$$

aussi, on suppose que pour cette solution de base réalisable

$$\begin{aligned} z_j^{(1)} &= p_B^T B^{-1} a_j \\ z_j^{(2)} &= q_B^T B^{-1} a_j, \end{aligned}$$

sont connus pour chaque colonne  $a_j$ ,  $j \in N$ .

Comme dans [51], si la solution réalisable de base est notée  $\hat{x}_B$  alors  $\hat{x}_B = \hat{B}^{-1}b$ , où

$$\hat{B} = (\hat{b}_1, \dots, \hat{b}_m)$$

est la nouvelle matrice inversible obtenue à partir de  $B$  en supprimant  $b_k$  et en le remplaçant par  $a_j$ .

Les colonnes de la nouvelle matrice  $\hat{B}$  sont données par :

$$\begin{aligned} \hat{b}_i &= b_i, \quad i \neq k \\ \hat{b}_k &= a_j, \quad j \in N \end{aligned}$$

et les valeurs des nouvelles variables de base en fonction des variables originaux et de  $y_{ij}$  sont,

$$\begin{aligned} \hat{x}_{B_i} &= x_{B_i} - x_{B_k} \frac{y_{ij}}{y_{kj}}, \quad i \neq k \\ \hat{x}_{B_k} &= \frac{x_{B_k}}{y_{kj}}, \end{aligned}$$

où

$$a_j = \sum_1^m y_{ij} b_i.$$

Par conséquent, la nouvelle valeur de la fonction objectif est  $f(\hat{x}_B) = f_1(\hat{x}_B)f_2(\hat{x}_B)$ ,

où

$$\begin{aligned}
 f_1(\hat{x}_B) &= p_B \hat{x}_B + \alpha \\
 &= \sum_{\substack{i=1 \\ i \neq k}}^m p_{B_i} \hat{x}_{B_i} + p_{B_k} \hat{x}_{B_k} + \alpha \\
 &= \sum_{\substack{i=1 \\ i \neq k}}^m p_{B_i} (x_{B_i} - x_{B_k} \frac{y_{ij}}{y_{kj}}) + \frac{x_{B_k}}{y_{kj}} p_{B_j} + \alpha \\
 &= \sum_{i=1}^m p_{B_i} (x_{B_i} - x_{B_k} \frac{y_{ij}}{y_{kj}}) + \frac{x_{B_k}}{y_{kj}} p_{B_j} + \alpha \\
 &= \sum_{i=1}^m p_{B_i} x_{B_i} + \alpha - \frac{x_{B_k}}{y_{kj}} \sum_{i=1}^m p_{B_i} y_{ij} + \frac{x_{B_k}}{y_{kj}} p_{B_j} \\
 &= p_B x_B + \alpha + \frac{x_{B_k}}{y_{kj}} p_{B_j} - \frac{x_{B_k}}{y_{kj}} z_j^{(1)} \\
 &= f_1(x_B) + \frac{x_{B_k}}{y_{kj}} (p_{B_j} - z_j^{(1)}) \\
 &= f_1(x_B) + \mu_j (p_j - z_j^{(1)}),
 \end{aligned} \tag{1.9}$$

avec  $\mu_j = \frac{x_{B_k}}{y_{kj}} = \min \left\{ \frac{x_{B_i}}{y_{ij}} \mid y_{ij} > 0, i \in I \right\}, j \in N$ .

D'une manière similaire, on peut montrer que

$$f_2(\hat{x}_B) = f_2(x_B) + \mu_j (q_j - z_j^{(2)}). \tag{1.10}$$

Par conséquent, en utilisant (1.9-1.10), on obtient

$$\begin{aligned}
 f(\hat{x}_B) &= (p_B \hat{x}_B + \alpha)(q_B \hat{x}_B + \beta) \\
 &= [f_1(x_B) + \mu_j (p_j - z_j^{(1)})][f_2(x_B) + \mu_j (q_j - z_j^{(2)})] \\
 &= f_1(x_B) f_2(x_B) + \mu_j [f_2(x_B)(p_j - z_j^{(1)}) + f_1(x_B)(q_j - z_j^{(2)}) + \mu_j (p_j - z_j^{(1)})(q_j - z_j^{(2)})] \\
 &= f(x_B) + \mu_j [f_2(x_B)(p_j - z_j^{(1)}) + f_1(x_B)(q_j - z_j^{(2)}) + \mu_j (p_j - z_j^{(1)})(q_j - z_j^{(2)})],
 \end{aligned}$$

avec  $\mu_j$  est strictement positif dans le cas de non dégénérescence.

Notons les composantes du vecteur croissance  $\gamma$  by

$$\gamma_j = [f_2(x_B)(p_j - z_j^{(1)}) + f_1(x_B)(q_j - z_j^{(2)}) + \mu_j (p_j - z_j^{(1)})(q_j - z_j^{(2)})].$$

Par conséquent,

$$f(\hat{x}_B) - f(x_B) = \mu_j \gamma_j. \tag{1.11}$$

Soit une solution de base réalisable  $x_B = B^{-1}b$  pour un problème quadratique indéfini, la valeur de la fonction objectif de cette solution étant  $(p_B x_B + \alpha)(q_B x_B + \beta)$ . Si pour une colonne  $a_j, j \in N$  pour laquelle  $\gamma_j > 0$ , et au moins un  $y_{ij} > 0, i = 1, 2, \dots, m$ , alors il est possible d'obtenir une nouvelle solution réalisable de base en remplaçant l'une des colonne de  $B$  par  $a_j$ , et la nouvelle valeur de la fonction objectif  $f(\hat{x}_B)$  satisfait  $f(\hat{x}_B) > f(x_B)$  si la dégénérescence n'est pas présente. Ainsi, nous pouvons passer d'une

base à l'autre tant qu'il existera des  $a_j$  hors base, avec la condition  $\gamma_j > 0$  et à chaque étape, la fonction objectif est améliorée.

Par conséquent, ce processus ne se terminera que lorsque  $\gamma_j \leq 0$  pour chaque colonne  $a_j$ ,  $j \in N$ .

Le théorème suivant [93] donne une condition nécessaire et suffisante d'optimalité pour le problème (1.8).

**Théorème 1.6.2** *Une solution de base réalisable  $x_B$  de  $\mathcal{X}_l$  est optimale pour le problème (1.8) si et seulement si  $\gamma_j \leq 0$  pour tout  $j \in N$ .*

## 1.7 Conclusion

Dans ce chapitre, nous nous sommes intéressés aux problèmes de l'optimisation non linéaire à savoir l'optimisation fractionnaire linéaire, et l'optimisation quadratique, nous avons parcouru toute la richesse et la variété des travaux consacrés aux programmes fractionnaire linéaire et quadratique. Ce petit tour d'horizon nous a incité à étudier et détailler d'avantage ces types 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 et d'un critère fractionnaire linéaire sur l'ensemble efficient d'un problème multiobjectif linéaire discret.

## Chapitre 2

# Généralité sur l'optimisation multiobjectif

Dans ce chapitre, nous présentons l'optimisation mathématique multiobjectif qui sera le cadre de travail de cette thèse. Nous rappelons les notions fondamentales en décision multiobjectif, les terminologies usuelles, les principaux résultats ainsi que les approches de base de la programmation mathématique multiobjectif.

### 2.1 Introduction

La principale difficulté que l'on rencontre en optimisation monoobjectif vient du fait que modéliser un problème sous la forme d'une équation unique peut être une tâche difficile. Avoir comme but de se ramener à une seule fonction objectif peut aussi biaiser la modélisation. L'optimisation multiobjectif autorise ces degrés de liberté qui manquaient en optimisation monoobjectif.

La plupart des problèmes d'optimisation réels sont alors décrits à l'aide de plusieurs objectifs ou critères qui sont souvent contradictoires ou conflictuels et qui doivent 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, 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 dûs à Koopmans [59] en 1951 qui donna une condition nécessaire et suffisante d'efficacité d'une solution suivis par les travaux

de Kuhn et Tucker [60] 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 multiobjectif linéaire [91] et non-linéaire [68], [69], [45], [28] que des problèmes multiobjectifs booléens ou en nombres entiers [95].

## 2.2 Notions fondamentales

### 2.2.1 Position du problème

On définit un problème multiobjectif 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 :

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

Le symbole " " signifie qu'il n'est généralement pas possible de trouver dans  $S$  une solution qui optimise simultanément les  $r$  critères.

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  $x_i$ ,  $i = 1, \dots, n$ , des 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  $\mathbb{R}^n$  décrit implicitement par des contraintes d'égalités et/ou d'inégalités.
- L'ensemble  $\mathbb{R}^n$  qui contient  $S$  est dit *espace de décision*.
- L'ensemble  $\mathbb{R}^r$  qui contient  $F = f(S)$  est dit *espace des critères*.

### 2.2.2 La relation de dominance

Lors de résolution d'un problème multiobjectif, 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 et appelée dominance.

**Définition 2.2.1 (Dominance)** Soient deux vecteurs critères  $f(x), f(y) \in F$ . On dit que  $f(x)$  domine  $f(y)$  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.

### 2.2.3 Optimalité de Pareto

**Définition 2.2.2 (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  $f_i(x)$  domine  $f_i(\bar{x}) \quad \forall i \in \{1, \dots, r\}$ .

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

**Définition 2.2.3** 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

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

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 2.2.4** 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 :

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

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

**Définition 2.2.5 (Ensemble Pareto optimal)** L'ensemble Pareto optimal de  $S$  (ou l'ensemble efficients), est défini par l'ensemble  $\mathcal{X}$  :

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

**Définition 2.2.6 (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 2.1.

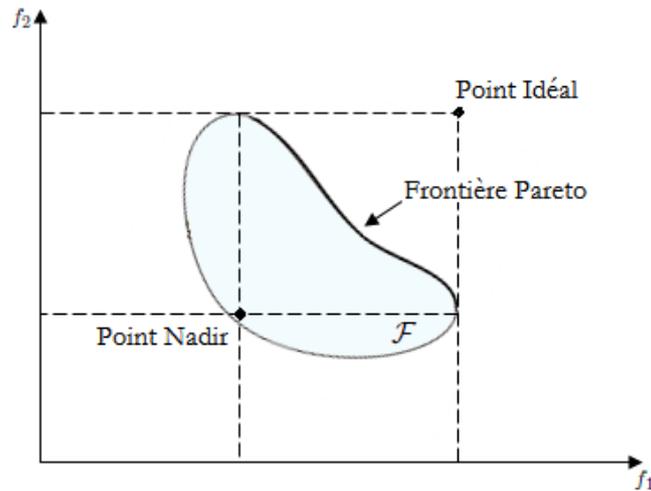


FIGURE 2.1 – La Frontière Pareto

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 2.2.7 (Point Idéal)** Les coordonnées du point idéal ( $Z^I$ ) correspondent aux meilleures 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 :

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

**Définition 2.2.8 (Point Nadir)** Les coordonnées du point Nadir ( $Z^{nad}$ ) correspondent aux pires valeurs de chaque objectif des points du front Pareto.

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

La figure 2.1 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),...

## 2.2.4 Détection graphique de l'efficacité

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

**Définition 2.2.9** 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 2.2.10 (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 2.2.11 (Théorème du contact)** Un vecteur  $x$  est optimal au sens de Pareto pour un problème d'optimisation multiobjectif donné si  $(C^+ + x) \cap F = \{x\}$ .

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

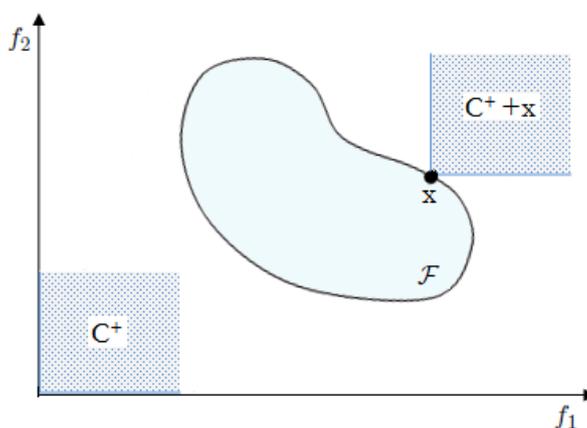


FIGURE 2.2 – Le théorème du contact

## 2.2.5 Caractérisation des solutions efficaces

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

**Théorème 2.2.1 (Wendell[96])** *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 multiobjectif (MOP) 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 2.2.2 (Benson [15])** *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 (mop) si et seulement si la valeur optimale de la fonction objectif  $\Theta$  est nulle dans le problème 2.2.*

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

**Définition 2.2.12** *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 2.2.3 (Ehrgott [44])** *Soit  $\bar{x} \in S$  est Pareto optimale pour le problème multiobjectif (MOP) si et seulement si :*

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

## 2.3 Choix de la méthode d'aide à la décision

Dans la résolution d'un problème multiobjectif 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 multiobjectif, il faut se poser la question du type de méthodes 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 [38] :

- *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 multiobjectif qui n'entrent pas exclusivement dans une famille. On peut utiliser, par exemple, une méthode à 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.

## 2.4 Programmation linéaire multiobjectif 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 multiobjectif discrète et de présenter quelques méthodes exactes permettant de caractériser totalement ou partiellement l'ensemble de solutions efficaces du problème.

### 2.4.1 Formulation du problème

Un problème de la programmation linéaire multiobjectif en nombres entiers est défini comme suit :

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

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 multiobjectif linéaire à variables entières ( $P$ ) consiste à trouver toutes les solutions efficaces au sens de la définition suivante :

**Définition 2.4.1** *Un point  $x^0 \in S$  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  $i \in \{1, \dots, r\}$ .*

## 2.4.2 Solutions supportées et non supportées

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

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

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

**Théorème 2.4.1 (Geoffrion[47])** *Soit le problème unicritère linéaire suivant :*

$$(P_\lambda) \left\{ \begin{array}{l} \max \sum_{i=1}^r \lambda_i Z_i(x) \\ sc \\ x \in S \text{ et } \lambda_i \in \Lambda, \forall i \in \{1, \dots, r\} \end{array} \right. \quad (2.4)$$

*La solution  $x^*$  est efficace pour le problème  $(P_R)$  si et seulement si  $x^*$  est une solution optimale du problème paramétrique  $(P_\lambda)$  où  $\Lambda = \{\lambda_i : \sum_{i=1}^r \lambda_i = 1 \text{ et } \lambda_i > 0, \forall i \in \{1, \dots, r\}\}$*

D'après ce théorème, l'ensemble efficient du problème  $(P_R)$  sans les contraintes d'intégrité est bien caractérisé par les solutions du problème paramétrique  $(P_\lambda)$ . 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 multiobjectif à variables discrètes est l'existence de solutions efficaces pour  $(P)$  qui ne sont pas optimales pour  $(P_\lambda)$  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  $(P)$ ).

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 [85].

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

### 2.4.3 Quelques méthodes de résolution d'un problème *MOILP*

Plusieurs chercheurs, citons en particulier, Steuer et Choo [91], Klein et Hannan [58], Crema et Sylva [33], Gupta et Malhotra [48], Abbas et Moulai [3], Abbas et Chaabane [4], motivés par de nombreuses stratégies, se sont intéressés à caractériser totalement ou partiellement l'ensemble des solutions efficaces du problème de programmation linéaire multiobjectif en nombres entiers. Dans cette partie, quelques méthodes de résolution des problèmes *MOILP* sont exposées.

#### Méthode de D.Klein & E.Hannan :

La technique proposée par D.Klein & E.Hannan [58] 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 :

$$(P_1) : \max\{Z_i = c^i x, x \in D = \{x \in \mathbb{R}^n \mid Ax \leq b, x \geq 0, x \text{ entier}\}\}$$

(L'indice  $i$  est pris arbitrairement dans  $1, \dots, r$ )

**Si** la solution optimale de  $(P_1)$ , notée  $x_1$ , 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.

### 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 multiobjectif en nombres entiers. Elle peut être vue comme une alternative à celle de Gupta & Malhotra [48] (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 :

$$(P_1) : \max\{Z_1 = c^1 x, x \in D = \{x \in \mathbb{R}^n \mid Ax \leq b, x \geq 0, x \text{ entier}\}\}$$

dont le problème relaxé est :

$$(P_1) : \max\{Z_1 = c^1 x, x \in S = \{x \in \mathbb{R}^n \mid Ax \leq b, x \geq 0\}\}$$

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

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

- $S_k = \{x \in \mathbb{R}^{n_k} \mid A_k x \leq b_k, A_k \in \mathbb{R}^{m_k \times n_k}, b_k \in \mathbb{R}^{m_k}, x \geq 0\}$  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 hors base de  $x_k$ .
- $\widehat{A}_k = (A_{B_k})^{-1} A_k$ .
- $\widehat{C}_k = (C_k) - \pi_k A_k$  avec  $\pi_k = (C_{B_k})(A_{B_k})^{-1}$ .
- $\Gamma_k = \{j \in N_k \mid (\widehat{c}_k^1)_j < 0 \text{ et } (\widehat{c}_k^i)_j > 0 \text{ pour au moins un critère } i, i = \overline{2, r}\}$ .
- $\Omega_k = \{j \in N_k \mid (\widehat{c}_k^1)_j = 0\}$ .
- $\Psi_k = \{j \in N_k \mid (\widehat{c}_k^1)_j < 0 \text{ et } (\widehat{c}_k^i)_j < 0 \text{ pour au moins un critère } i, i = \overline{2, r}\}$ .

**Définition 2.4.2** Une arête  $E^{j_k}$ ,  $j_k \in N_k$  incidente à  $x^k$  est définie comme étant l'ensemble :

$$\left\{ X \in \mathbb{R}^{n_k} \mid \begin{cases} x_i = x_i^k - \theta_{j_k} \widehat{a}_{lig(i)_k}^{j_k} & i \in B_k \\ x_{j_k} = \theta_{j_k} \\ x_l = 0 \quad \forall l \in N_k \setminus j_k \end{cases} \right\} \quad (2.5)$$

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

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

**Remarque 2.4.1** 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 2** : ABBAS & MOULAI

---

**Étape 1** : Résoudre le problème  $(P_1)$  défini par :

$$(P_1) : \max\{Z_1 = c^1, x \in D = \{x \in \mathbb{R}^n \mid Ax \leq b, x \geq 0, x \text{ entier}\}\}$$

et trouver la solution optimale entière  $x^1$  sur  $S_1$ . Construire l'ensemble  $\Omega_1$ .

**Étape 2** : Tester l'ensemble  $\Omega_1$ .

**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  $\theta$  de l'opération pivot.

si  $\theta \geq 1$ , alors déterminer toutes les solutions entières 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 choisi 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'arête  $E^{j_1}$  par la coupe :

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

L'algorithme dual du simplexe et des coupes successives 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 duale 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.

---

### Méthode de M.Abbas et D.Chaabane

M. Abbas et D. Chaabane ont proposé dans [4] 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 3** : 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  $\Omega_1$ .

**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  $\theta$ .

**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 entières 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  $\Psi_k$ .

**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  $\theta$ .

**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 duales 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 2.4.2** *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.*

### Méthode A. Sylva et J. Crema

La méthode développée par Crema et Sylva [33] 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 4 : 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 :

$$(P_1) : \max \left\{ \sum_{i=1}^r \lambda_i c^i x, x \in D = \{x \in \mathbb{R}^n \mid Ax \leq b, x \geq 0, x \text{ entier}\} \right\}$$

**Si**  $(P_1)$ , n'admet pas de 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 de 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 :

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

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 :

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

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

---

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.

## 2.5 Conclusion

Nous avons présenté dans ce chapitre les principaux concepts de l'optimisation multiobjectif tels que la modélisation d'un problème multiobjectif, la notion de la dominance de Pareto et la structure de la surface de compromis. Une classification des méthodes de résolution a été introduite.

Nous avons, par la suite, présenté les problèmes de la programmation multiobjectif en nombre entiers sous contraintes linéaires qui représente le cadre général de notre travail. 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 mentionnés.

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.

## Chapitre 3

# Une méthode de résolution des problèmes d'optimisation multiobjectif quadratique indéfinie en nombres entiers MOIIQP

Dans ce chapitre on présente une méthode de résolution des problèmes d'optimisation multiobjectif quadratique indéfinie en nombres entiers. L'algorithme proposé pour ce type de problèmes multiobjectif non linéaire permet d'obtenir l'ensemble de toutes les solutions efficaces du problème en utilisant le principe de Branch & Cut. À chaque nœud de l'arborescence, un problème quadratique indéfinie mono-objectif, obtenu en considérant une seule fonction objectif et en éliminant la contrainte d'intégrité, est résolu en utilisant la méthode du simplexe modifié. Des règles de saturation et des tests permettent soit de saturer un nœud de l'arborescence, soit de brancher le nœud en sous-problèmes ou d'ajouter une coupe. Après une présentation de l'algorithme, un exemple et une expérimentation numériques seront présentés.

### 3.1 Introduction

La programmation multiobjectif est un domaine de recherche bien connu en optimisation et en recherche opérationnelle. Dans de nombreuses situations, les modèles de la vie réelle ne sont pas représentés par des fonctions linéaires et les chercheurs ont commencé à se concentrer sur les problèmes de la programmation multiobjectif non linéaire [69, 70, 13].

De plus, un travail considérable a été fait sur les problèmes de la programmation en nombres entiers. Ces problèmes sont de grande importance dans le monde des affaires et de l'industrie, où des solutions fractionnaires sont souvent irréalistes car les unités ne sont pas divisibles.

La principale difficulté pour résoudre le problème de programmation quadratique indéfinie *IQP* réside dans la non convexité de la fonction objectif. Le produit de deux fonctions affines n'est pas convexe en général, alors qu'il est quasi-concave [93], donc, si le problème *IQP* a une solution optimale, alors il a une solution maximale extrême.

Un algorithme de type simplexe est donné dans [12, 52, 84, 93] et une technique présentée dans [71] résout le problème d'optimisation quadratique indéfinie dans le cas d'objectif unique. Malgré ses applications potentielles dans la théorie microéconomique [53] et le modèle de contrôle prédictif [10], une recherche bibliographique nous a permis de conclure que la résolution du cas à objectifs multiples, le *MOIIQP*, n'a pas reçu l'attention qu'il mérite, et que la littérature sur ce sujet est rare ([9]), alors qu'il existe un besoin croissant de nouveaux algorithmes capables de calculer rapidement des solutions efficaces.

Ce chapitre est consacré au problème de la programmation multiobjectif quadratique indéfinie en nombres entiers *MOIIQP*. Nous considérons le problème de la maximisation, sous des contraintes linéaires, d'un nombre  $r$  de fonctions objectives, chacune d'elles représentant le type spécial de fonction quadratique, appelée indéfinie, qui peut être écrit comme un produit de deux fonctions affines strictement positives, les variables de décision prenant des valeurs entières.

La méthodologie proposée dans ce chapitre utilise un processus de Branch & Bound renforcé par une coupe efficace, de sorte que toutes les solutions efficaces du problème *MOIIQP* soient générées sans calculer l'ensemble des solutions réalisables entières.

## 3.2 Notations et définitions

Mathématiquement, le problème *MOIIQP* peut être formulé comme suit :

$$(MOIIQP) \quad \left\{ \begin{array}{l} \max \quad f^1(x) = (p^1x + \alpha^1)(q^1x + \beta^1) \\ \max \quad f^2(x) = (p^2x + \alpha^2)(q^2x + \beta^2) \\ \quad \quad \quad \vdots \\ \max \quad f^r(x) = (p^rx + \alpha^r)(q^rx + \beta^r) \\ s.c. \\ \quad \quad \quad x \in \mathcal{D} = \mathcal{X} \cap \mathbb{Z}^n, \end{array} \right. \quad (3.1)$$

avec  $\mathbb{Z}$  est l'ensemble des entiers,  $r \geq 2$ ,  $A \in \mathbb{Z}^{m \times n}$ ,  $b \in \mathbb{Z}^m$ ,  $p^i = (p_1^i, p_2^i, \dots, p_n^i) \in \mathbb{Z}^{1 \times n}$ ,  $q^i = (q_1^i, q_2^i, \dots, q_n^i) \in \mathbb{Z}^{1 \times n}$ ,  $\alpha^i, \beta^i \in \mathbb{R}$ ;  $i \in \{1, 2, \dots, r\}$  et  $\mathcal{X} = \{x \in \mathbb{R}^n / Ax \leq b, x \geq 0\}$ .

Dans ce qui suit, on suppose que  $\mathcal{X}$  est un polyèdre convexe fermé, borné et non vide,  $p^i(x) + \alpha^i$  et  $q^i(x) + \beta^i$  sont positifs pour tout  $i \in \{1, \dots, r\}$ , et finalement on suppose que l'ensemble  $\mathcal{D}$  est non vide.

**Définition 3.2.1** Une solution réalisable  $x \in \mathcal{D}$  est dite solution efficace pour le problème *MOIIQP* si et seulement s'il n'existe pas de solution  $y \in \mathcal{D}$  telle que  $f^i(y) \geq f^i(x)$  pour tout  $i \in \{1, \dots, r\}$  et  $f^k(y) > f^k(x)$  pour au moins un indice  $k \in \{1, \dots, r\}$ . Sinon,  $x$  est dite non efficace et son vecteur critère  $f(x)$  est dominé par le vecteur critère  $f(y)$ , où  $f(x) = (f^i(x))_{i=1, \dots, r}$ .

Pour générer toutes les solutions efficaces du problème principal 3.1, la méthode proposée est basée sur la résolution, à chaque étape, d'un problème de programmation quadratique indéfinie  $IQP_l$  ( $l \in \mathbb{N}$ ) défini par :

$$(IQP_l) \quad \begin{cases} \max & f^1(x) = (p^1x + \alpha^1)(q^1x + \beta^1) \\ s.c. & \\ & x \in \mathcal{X}_l, \end{cases} \quad (3.2)$$

où  $p^1, q^1 \in \mathbb{Z}^{1 \times n}$ ,  $\alpha^1, \beta^1 \in \mathbb{R}$ ,  $\mathcal{X}_0 = \mathcal{X}$  et  $\mathcal{X}_l$  est un sous ensemble de l'ensemble original  $\mathcal{X}$  à explorer au nœud  $l$  de l'arborescence. Notons que à la place de  $f^1$ , et d'une façon similaire, on peut considérer le problème  $(IQP_l)$  avec une autre fonction  $f^i$  pour n'importe quel indice  $i$ ,  $i \in \{2, \dots, r\}$ .

**Problème de la programmation quadratique indéfinie** La recherche d'une solution optimale du problème quadratique indéfinie nécessite l'introduction des notations suivantes :

- $x_B$  est une solution réalisable de base du problème (3.2) correspondante à la matrice de base  $B$  tel que  $x_B = B^{-1}b$ .
- $I$  est l'ensemble des indices des variables de base correspondante à  $x_B$ .
- $N$  est l'ensemble des indices des variables hors-base correspondante à  $x_B$ .

De plus,

- $f(x_B) = f_1(x_B)f_2(x_B)$  tel que,

$$\begin{aligned} f_1(x_B) &= p_B^T x_B + \alpha = p_B^T B^{-1}b + \alpha \\ f_2(x_B) &= q_B^T x_B + \beta = q_B^T B^{-1}b + \beta. \end{aligned}$$

- Aussi, on assume que, à cette solution réalisable de base, on a

$$\begin{aligned} z_j^{(1)} &= p_B^T B^{-1}a_j \\ z_j^{(2)} &= q_B^T B^{-1}a_j, \end{aligned}$$

définies pour chaque colonne  $a_j = \sum_{i=1}^m y_{ij}b_i$ ,  $j \in N$ .

Si la nouvelle solution réalisable de base est notée  $\hat{x}_B$ , alors  $\hat{x}_B = \hat{B}^{-1}b$ , la nouvelle valeur de la fonction objectif est

$$f(\hat{x}_B) - f(x_B) = \mu_j \gamma_j. \quad (3.3)$$

avec

$$\begin{aligned} \mu_j &= \frac{(x_B)_k}{y_{kj}} = \min \left\{ \frac{(x_B)_i}{y_{ij}} \mid y_{ij} > 0, i \in I \right\}, j \in N \\ \gamma_j &= f_2(x_B)(p_j - z_j^{(1)}) + f_1(x_B)(q_j - z_j^{(2)}) + \mu_j(p_j - z_j^{(1)})(q_j - z_j^{(2)}), \end{aligned}$$

$\mu_j$  est strictement positif dans le cas non dégénéré.

Le théorème suivant [71] nous donne la condition d'optimalité du problème (3.2).

**Théorème 3.2.1** [71] *Étant donnée une solution réalisable  $x_B = B^{-1}b$  du problème (3.2), tel que  $\gamma_j \leq 0$  pour chaque colonne  $a_j$  dans  $A$  non dans  $B$ , alors  $x_B$  est une solution réalisable de base optimale.*

Afin de décrire notre procédure pour générer des solutions efficaces du programme (*MOIIQP*), les notations suivantes sont utilisées tout au long de ce chapitre :

- $x^{*(l)}$  est la première solution entière obtenue en résolvant le problème 3.2 en utilisant, éventuellement, le processus de branchement bien connu dans les techniques de Branch&Bound.
- $I_l$  désigne l'ensemble d'indices des variables de base correspondante à  $x^{*(l)}$ .
- $N_l$  désigne l'ensemble d'indices des variables hors base correspondante à  $x^{*(l)}$ .
- $\gamma_j^i$  est la  $j^{\text{ième}}$  composante du vecteur de la direction de croissance  $\gamma^i$  pour chaque fonction objectif  $f^i$ ,  $i \in \{1, 2, \dots, r\}$  défini dans le dernier tableau du simplexe.

### 3.2.1 La coupe efficace

Notons que notre approche est basée sur la méthode de Branch&Cut, de sorte que, à chaque étape  $l$  de l'algorithme une coupe efficace [30] est ajouté au domaine initial pour supprimer des solutions non efficaces pour le problème multiobjectif (*MOIIQP*).

Avant de déterminer l'expression mathématique de cette coupe efficace, il faut d'abord définir les ensembles suivants à la solution entière  $x^{*(l)}$  :

$$H_l = \{j \in N_l | \exists i \in \{2, \dots, r\}, \bar{\gamma}_j^i > 0\} \cup \{j \in N_l | \bar{\gamma}_j^i = 0, \forall i \in \{1, \dots, r\}\} \quad (3.4)$$

et la coupe

$$\sum_{j \in H_l} x_j \geq 1 \quad (3.5)$$

est efficace. Alors, l'ensemble

$$\mathcal{X}_{l+1} = \left\{ x \in \mathcal{X}_l \mid \sum_{j \in H_l} x_j \geq 1 \right\}. \quad (3.6)$$

## 3.3 Développement de la méthode

Dans cette section, nous présentons notre méthode pour générer toute les solutions efficaces d'un problème d'optimisation multiobjectif quadratique indéfinie en nombres entiers (3.1). Pour ce, on donne une description détaillée de la méthode, puis la formulation de l'algorithme et enfin des résultats théoriques permettant de prouver l'efficacité et la finitude de l'algorithme.

### 3.3.1 Description de la méthode

Soit  $\mathcal{X}_E$  l'ensemble de toutes les solutions efficaces du problème (3.1) initialisé à l'ensemble vide.

La méthode consiste à résoudre le problème 3.2 à chaque étape de l'algorithme. Chaque programme (3.2) correspond à un nœud  $l$  dans une arborescence structurée.

Un nœud  $l$  est sondé si :

- le programme correspondant  $IQP_l$  est irréalisable.
- l'ensemble  $H_l$  est vide.

On résout le problème  $IQP_l$  à l'étape  $l$  de l'algorithme. Initialement, pour résoudre le programme  $IQP_0$  on utilise la méthode décrite dans [93] qui est une variante de la méthode du simplexe, sinon pour les autres problèmes  $IQP_l$ ,  $l \neq 0$  on utilise la méthode dual du simplexe.  $r$  lignes sont ajoutés au tableau du simplexe pour mettre à jour les vecteurs de la direction de croissance, notés  $\gamma^i$ ,  $i \in \{1, \dots, r\}$ , des  $r$  fonctions objectif  $f^i$  en respectant la base correspondante.

Si une solution entière  $x^{*(l)}$  est retrouvée au nœud  $l$ , elle est comparée aux solutions entières de l'ensemble  $\mathcal{X}_E$  afin de le mettre à jour. On établie la coupe (5.6) et on l'ajoute au tableau du simplexe courant, ce qui permet d'éliminer la solution  $x^{*(l)}$  ainsi que des solutions non efficaces du problème (3.1) et de déterminer une nouvelle solution entière. Toutefois, si aucune amélioration des critères ne peut être apportée sur le domaine restant ( $H_l = \emptyset$ ), le nœud  $l$  sera stérilisé.

Dans le cas où la solution optimale du problème 3.2 est non entière, on note  $x_j$  la composante non entière de  $x^{*(l)}$  telle que  $x_j = e$  où  $e$  est un nombre fractionnaire. Le nœud  $l$  sera alors séparé en deux autres nœuds qui lui sont imposées par les contraintes additionnelles  $x_j \leq [e]$  et  $x_j \geq [e]$  où  $[e]$  indique la partie entière du nombre réel  $e$ .

Dans chaque nœud, le programme quadratique obtenu doit être résolu jusqu'à obtention d'une solution entière si elle existe.

La méthode se termine quand tous les nœuds créés sont sondés. L'ensemble des solutions entières efficaces du problème (3.1) est alors  $\mathcal{X}_E$ .

L'algorithme générant l'ensemble efficient du problème (3.1) est présenté dans ce qui suit.

### 3.3.2 Formulation de l'algorithme

---

**Algorithm 1** Résolution du problème multiobjectif quadratique indéfini en nombres entiers

---

Étape 1 : *Initialisation*

$$l = 0, \mathcal{X}_0 = \mathcal{X}, \mathcal{X}_E = \emptyset$$

Étape 2 : *Étape générale*

Tant qu'il existe un nœud non encore sondé dans l'arborescence, choisir un nœud non sondé et résoudre le programme quadratique indéfini correspondant (3.2) en utilisant la technique détaillé dans [93].

- Si  $IQP_l$  n'est pas réalisable, le nœud  $l$  correspondant est sondé ;
- Sinon, soit  $x^{*(l)}$  une solution optimale du problème  $IQP_l$  ;
  1. Si  $x^{*(l)}$  est entière, aller à l'étape 2a.
  2. Sinon, aller à l'étape 2b.

Étape 2a : *Mise à jour de l'ensemble  $\mathcal{X}_E$ .*

- Si le vecteur  $f(x^{*(l)})$  est non dominé par le vecteur  $f(x)$  pour tout  $x \in \mathcal{X}_E$  alors  $\mathcal{X}_E = \mathcal{X}_E \cup \{x^{*(l)}\}$ .
- Si il existe une solution  $x \in \mathcal{X}_E$  telle que  $f(x^{*(l)})$  domine  $f(x)$  alors  $\mathcal{X}_E = \mathcal{X}_E \setminus \{x\} \cup \{x^{*(l)}\}$ .

Déterminer les ensembles  $N_l, H_l$  ;

1. Si  $H_l = \emptyset$  alors le nœud  $l$  est sondé ; aller à l'étape 2 ;
2. Sinon, ajouter la coupe  $\sum_{j \in H_l} x_j \geq 1$  au problème  $IQP_l$  et aller à l'étape 2, un nouveau nœud est créé,  $l = l + 1$ .

Étape 2b : *Processus de Branchement*

Choisir une composante non entière de la solution  $x^{*(l)}$ , noté  $(x^{*(l)})_j$  et diviser l'ensemble réalisable  $\mathcal{X}_l$  en deux sous-ensembles ( $\mathcal{X}_{l_1}$  et  $\mathcal{X}_{l_2}$ ) ( $l_1 > l+1, l_2 > l+1, l_1 \neq l_2$ ) en ajoutant à  $\mathcal{X}_l$  les contraintes de branchement  $x_j \leq \lfloor (x^{*(l)})_j \rfloor$  et  $x_j \geq \lceil (x^{*(l)})_j \rceil$  respectivement et aller à l'étape 2.

---

### 3.3.3 Résultats théoriques

Dans ce qui suit, nous donnons les résultats théoriques justifiant la finitude de notre algorithme.

**Théorème 3.3.1** *Supposons que  $H_l \neq \emptyset$  à la solution entière courante  $x^{*(l)}$ . Si  $x$  est une solution entière efficace sur le domaine  $\mathcal{X}_l \setminus \{x^{*(l)}\}$ , alors  $x \in \mathcal{X}_{l+1}$ .*

**Preuve 3.3.1** *Soit  $x$  une solution entière sur l'ensemble  $\mathcal{X}_l \setminus \{x^{*(l)}\}$  tel que  $x \notin \mathcal{X}_{l+1}$ , alors  $x \notin \{x \in \mathcal{X}_l \mid \sum_{j \in H_l} x_j \geq 1\}$  ce qui implique  $x \in \{x \in \mathcal{X}_l \mid \sum_{j \in N_l \setminus H_l} x_j \geq 1\}$ .*

*Par conséquent, nous obtiendrons les inégalités suivantes :*

$$\begin{aligned} \sum_{j \in H_l} x_j &< 1 \\ \sum_{j \in N_l \setminus H_l} x_j &\geq 1. \end{aligned}$$

*Il en résulte que  $x_j = 0$  pour tout  $j \in H_l$  et  $x_j \geq 1$  pour au moins un indice  $j \in N_l \setminus H_l$ .*

*En utilisant le tableau de simplexe à la solution  $x^{*(l)}$ , la valeur mise à jour de chaque objectifs  $f^i$ ,  $i \in \{1, \dots, r\}$  est écrite par rapport aux indices de bases  $j \in I_l$  et des indices hors base  $j \in N_l$ . Ainsi, on aura les égalités suivantes pour tout les critères :*

$$\begin{aligned} f^i(x) - f^i(x_l^*) &= \widehat{f}_1^i \widehat{f}_2^i - f_1^i f_2^i \\ &= [f_1^i + \mu_j(p_j^i - z_j^{i(1)})][f_2^i + \mu_j(q_j^i - z_j^{i(2)})] - f_1^i f_2^i \\ &= \mu_j[f_2^i(p_j^i - z_j^{i(1)}) + f_1^i(q_j^i - z_j^{i(2)}) + \mu_j(p_j^i - z_j^{i(1)})(q_j^i - z_j^{i(2)})] \\ &= \mu_j \gamma_j^i. \end{aligned}$$

*Comme  $\mu_j > 0$  et les composantes du vecteur de direction de croissance  $\gamma_j^1 \leq 0$  pour tout  $j \in N_l \setminus H_l$  et est strictement inférieur à zéro pour au moins un indice  $j$ , alors  $f^i(x) \leq f^i(x^{*(l)})$  pour tout critère  $f^i$   $i \in \{1, \dots, r\}$ , avec  $f^i(x) < f^i(x^{*(l)})$  pour au moins un critère  $i \in \{1, \dots, r\}$ .*

*Par conséquent, le vecteur de critère  $(f^1(x), \dots, f^r(x))$  est dominé par  $(f^1(x^{*(l)}), \dots, f^r(x^{*(l)}))$  et  $x$  n'est pas efficace.*

**corrolaire 3.3.1** *La contrainte  $\sum_{j \in H_l} x_j \geq 1$  définie une coupe efficace.*

**Preuve 3.3.2** *Il est clair que  $\sum_{j \in H_l} x_j \geq 1$  est une contrainte efficace valide par le théorème ci-dessus, puisque toutes les solutions efficaces entières dans le domaine actuel  $\mathcal{X}_l$  vérifient cette contrainte. De plus, la solution entière courante  $x^{*(l)}$  ne satisfait pas cette contrainte, puisque  $x_j = 0$  pour tout  $j \in H_l$ . Alors, on peut dire que la contrainte  $\sum_{j \in H_l} x_j \geq 1$  est une coupe efficace.*

**Proposition 3.3.1** *Si  $H_l = \emptyset$  à la solution entière courante  $x^{*(l)}$ , alors  $\mathcal{X}_l \setminus \{x^{*(l)}\}$  est un domaine exploré.*

**Preuve 3.3.3**  *$H_l = \emptyset$  signifie que  $x^{*(l)}$  est une solution optimale entière pour tous les critères, par conséquent  $x^{*(l)}$  est un point idéal dans le domaine  $\mathcal{X}_l$  et  $\mathcal{X}_l \setminus \{x^{*(l)}\}$  ne contient pas de solutions efficaces.*

**Théorème 3.3.2** *L'algorithme proposé génère toute les solutions efficaces entières du programme (*MOIIQP*) en un nombre fini d'itérations, si de telles solutions existent.*

**Preuve 3.3.4** *Comme  $D$ , l'ensemble des solutions entières réalisables du problème *MOILP*, est un ensemble fini borné contenu dans  $\mathcal{X}$ , la cardinalité de l'ensemble efficace  $\mathcal{X}_E$  est aussi un nombre fini . À chaque fois qu'une solution optimale entière  $x^{*(l)}$  est trouvée, la coupe efficace est ajoutée. Ainsi, selon le théorème et corollaire ci-dessus, au moins la solution  $x^{*(l)}$  est éliminée lorsque l'on étudie un sous-problème (*IQP<sub>k</sub>*),  $k > l$ , mais aucune solution efficace entière n'est omise. Dans le cas où il n'y a pas de solution entière optimale dans l'ensemble de décision  $\mathcal{X}$ , le sous-arbre enraciné au nœud  $l$  est exploré et tous les nœuds sont explorés après un nombre fini d'étapes.*

## 3.4 Exemple Illustratif

Pour montrer l'utilisation de l'algorithme, on considère l'exemple suivant :

$$MOIIQP \left\{ \begin{array}{l} \max f^1(x) = (x_1 + 2x_2 + 1)(-x_2 + 10) \\ \max f^2(x) = (3x_1 + x_2 + 2)(2x_1 + x_2 + 2) \\ \max f^3(x) = (x_1 + 3x_2 + 2)(2x_1 + 3x_2 + 2) \\ s.c. \\ \quad 3x_1 + 4x_2 \leq 24 \\ \quad 2x_1 + x_2 \leq 10 \\ \quad x_1 \geq 0, x_2 \geq 0 \text{ et entiers.} \end{array} \right.$$

**Initialization** : Soit  $\mathcal{X}_E = \emptyset$ ,  $l = 0$ . Le programme *IQP<sub>0</sub>* est résolu. La première solution optimale est  $x^{*(0)} = (16/5, 18/5)$ , la solution n'est pas entière et les résultats sont résumés dans la Table 3.1.

TABLE 3.1 – La première solution optimale

$B_1$	$x_3$	$x_4$	RHS
$x_1$	-1/5	4/5	16/5
$x_2$	2/5	-3/5	18/5
$\mu_j$	9	4	
$p_j^1 - z_j^{1(1)}$	-3/5	2/5	$f_1^1=57/5$
$q_j^1 - z_j^{1(2)}$	2/5	-3/5	$f_2^1=32/5$
$\gamma_j^1$	-36/25	-131/25	
$p_j^2 - z_j^{2(1)}$	1/5	-9/5	$f_1^2=76/5$
$q_j^2 - z_j^{2(2)}$	0	-1	$f_2^2=12$
$\gamma_j^2$	12/5	-148/5	
$p_j^3 - z_j^{3(1)}$	-1	1	$f_1^3=16$
$q_j^3 - z_j^{3(2)}$	-4/5	1/5	$f_2^3=96/9$
$\gamma_j^3$	-124/5	116/5	

Comme la solution n'est pas entière, le processus de branchement est déclenché et deux nœuds  $N_1, N_2$ , correspondants aux deux contraintes suivantes, seront créer :

$$N_1 : x_1 \leq 3.$$

$$N_2 : x_1 \geq 4.$$

$N_1$  : la contrainte  $x_1 \leq 3$  est ajoutée au problème  $IQP_0$  ; on obtient le problème  $IQP_1$  avec une solution optimale  $x^{*(1)} = (3, \frac{15}{4})$  donnée par la Table 3.2.

TABLE 3.2 – Solution optimale continue au nœud  $N_1$ .

$B_2$	$x_3$	$x_5$	RHS
$x_1$	0	1	3
$x_2$	1/4	-3/4	15/4
$x_4$	-1/4	-5/4	1/4
$\mu_j$	15	3	
$p_j^1 - z_j^{1(1)}$	-1/2	1/2	$f_1^1 = 23/2$
$q_j^1 - z_j^{1(2)}$	1/4	-3/4	$f_2^1 = 25/4$
$\gamma_j^1$	-17/8	-91/16	
$p_j^2 - z_j^{2(1)}$	-1/4	-9/4	$f_1^2 = 59/4$
$q_j^2 - z_j^{2(2)}$	-1/4	-5/4	$f_2^2 = 47/4$
$\gamma_j^2$	-91/16	-583/16	
$p_j^3 - z_j^{3(1)}$	-3/4	5/4	$f_1^3 = 65/4$
$q_j^3 - z_j^{3(2)}$	-3/4	1/4	$f_2^3 = 77/4$
$\gamma_j^3$	-291/16	465/16	

le processus de branchement est encore déclenché avec comme contraintes :

$$N_3 : x_2 \leq 3.$$

$$N_4 : x_2 \geq 4.$$

$N_3$  : la contrainte  $x_2 \leq 3$  est ajoutées au problème  $IQP_1$ , on obtient le problème  $IQP_2$  avec une solution optimale entière  $x^{*(2)} = (3, 3)$ , donnée dans la Table 3.3,  $f(x^{*(2)}) = (70, 154, 238)$ .  $\mathcal{X}_E = \{(3, 3)\}$ ,  $H_2 = \emptyset$ . Le nœud correspondant est sondé.

TABLE 3.3 – Solution entière efficace au nœud  $N_3$ .

$B_3$	$x_5$	$x_6$	RHS
$x_1$	0	1	3
$x_2$	1	0	3
$x_4$	-1	-2	3
$x_3$	-4	-3	1
$\mu_j$	3	3	
$p_j^1 - z_j^{1(1)}$	-1	-2	$f_1^1 = 10$
$q_j^1 - z_j^{1(2)}$	0	1	$f_2^1 = 7$
$\gamma_j^1$	-10	-7	
$p_j^2 - z_j^{2(1)}$	-3	-1	$f_1^2 = 14$
$q_j^2 - z_j^{2(2)}$	-2	-1	$f_2^2 = 11$
$\gamma_j^2$	-22	-43	
$p_j^3 - z_j^{3(1)}$	-1	-3	$f_1^3 = 14$
$q_j^3 - z_j^{3(2)}$	-2	-3	$f_2^3 = 17$
$\gamma_j^3$	-66	-39	

$N_4$  : la contrainte  $x_2 \geq 4$  est ajouté aux contraintes du problème  $IQP_1$ , on obtient un nouveau problème  $IQP_3$  et le tableau suivant 3.4 indiquant que la solution  $x^{*(3)} = (\frac{8}{3}, 4)$  est optimale.

TABLE 3.4 – Solution optimale continue au nœud  $N_4$ .

$B_4$	$x_3$	$x_6$	RHS
$x_1$	1/3	4/3	8/3
$x_2$	0	-1	4
$x_4$	-2/3	-5/3	2/3
$x_5$	-1/3	-4/3	1/3
$\mu$	8	2	
$p_j^1 - z_j^{1(1)}$	-1/3	-2/3	$f_1^1 = 35/3$
$q_j^1 - z_j^{1(2)}$	0	-1	$f_2^1 = 6$
$\gamma_j^1$	-2	-9	
$p_j^2 - z_j^{2(1)}$	-1	-3	$f_1^2 = 14$
$q_j^2 - z_j^{2(2)}$	-2/3	-5/3	$f_2^2 = 34/3$
$\gamma_j^2$	-46/3	-142/3	
$p_j^3 - z_j^{3(1)}$	-1/3	5/3	$f_1^3 = 50/3$
$q_j^3 - z_j^{3(2)}$	-2/3	1/3	$f_2^3 = 58/3$
$\gamma_j^3$	-142/9	350/9	

Le processus de branchement est appliqué.

$$N_5 : x_1 \leq 2.$$

$$N_6 : x_1 \geq 3.$$

$N_5$  : l'inéquation  $x_1 \leq 2$  est ajouté à  $IQP_3$ , on obtient un nouveau problème  $IQP_4$ , une solution optimale entière est alors obtenue  $x^{*(4)} = (2, 4)$ , comme le montre le tableau 3.5, avec  $f(x^{*(4)}) = (66, 120, 288)$ . L'ensemble  $\mathcal{X}_E$  est donc mis à jour,  $\mathcal{X}_E = \{(3, 3), (2, 4)\}$ ,  $H_4 = \{6\}$ .

TABLE 3.5 – Solution entière efficace au nœud  $N_5$ .

$B_5$	$x_7$	$x_6$	RHS
$x_1$	1	0	2
$x_2$	0	-1	4
$x_4$	-2	1	2
$x_5$	-1	0	1
$x_3$	-3	4	2
$\mu_j$	2	1/2	
$p_j^1 - z_j^{1(1)}$	2	-1	$f_1^1 = 11$
$q_j^1 - z_j^{1(2)}$	-1	0	$f_2^1 = 6$
$\gamma_j^1$	-6	0	
$p_j^2 - z_j^{2(1)}$	1	-3	$f_1^2 = 12$
$q_j^2 - z_j^{2(2)}$	1	-2	$f_2^2 = 10$
$\gamma_j^2$	-42	-45/2	
$p_j^3 - z_j^{3(1)}$	3	-1	$f_1^3 = 16$
$q_j^3 - z_j^{3(2)}$	3	-2	$f_2^3 = 18$
$\gamma_j^3$	-46	213/2	

On réduit l'ensemble réalisable du problème  $IQP_4$  en utilisant la coupe efficace  $x_6 \geq 1$ , on obtient un nouveau problème quadratique  $IQP_5$ ; la Table 3.6 donne sa solution optimale  $x^{*(5)} = (\frac{4}{3}, 5)$  qui est non entière.

TABLE 3.6 – Solution continue au nœud  $N_6$ .

$B_6$	$x_3$	$x_8$	RHS
$x_1$	1/3	4/3	4/3
$x_2$	0	-1	5
$x_4$	-2/3	-5/3	7/3
$x_5$	-1/3	-4/3	5/3
$x_7$	-1/3	-4/3	2/3
$x_6$	0	-1	1
$\mu_j$	4	1	
$p_j^1 - z_j^{1(1)}$	-1/3	2/3	$f_1^1 = 37/3$
$q_j^1 - z_j^{1(2)}$	0	-1	$f_2^1 = 5$
$\gamma_j^1$	-5/3	-29/3	
$p_j^2 - z_j^{2(1)}$	-1	-3	$f_1^2 = 11$
$q_j^2 - z_j^{2(2)}$	-2/3	-5/3	$f_2^2 = 29/3$
$\gamma_j^2$	-43/3	-127/3	
$p_j^3 - z_j^{3(1)}$	-1/3	5/3	$f_1^3 = 55/3$
$q_j^3 - z_j^{3(2)}$	-2/3	1/3	$f_1^3 = 59/3$
$\gamma_j^3$	-161/9	355/9	

Le processus de branchement est appliqué.

$N_7 : x_1 \leq 1$ .

$N_8 : x_1 \geq 2$ .

$N_7$  : La contrainte  $x_1 \leq 1$  ajoutée au programme  $IQP_5$  donne un nouveau problème  $IQP_6$  avec une solution optimale entière  $x^{*(6)} = (1, 5)$  comme le montre la Table 3.7 avec  $f(x^{*(6)}) = (60, 90, 342)$ . Mise à jour :  $\mathcal{X}_E = \{(3, 3), (2, 4), (1, 5)\}$ ,  $H_6 = \{8\}$ .

TABLE 3.7 – Solution entière efficace au nœud  $N_7$ .

$B_7$	$x_9$	$x_8$	RHS
$x_1$	1	0	1
$x_2$	0	-1	5
$x_4$	-2	1	3
$x_5$	-1	0	2
$x_7$	-1	0	1
$x_6$	0	-1	1
$x_3$	-3	4	1
$\mu_j$	4	1	
$p_j^1 - z_j^{1(1)}$	2	-1	$f_1^1 = 12$
$q_j^1 - z_j^{1(2)}$	-1	0	$f_2^1 = 5$
$\gamma_j^1$	-5	-5/2	
$p_j^2 - z_j^{2(1)}$	1	-3	$f_1^2 = 10$
$q_j^2 - z_j^{2(2)}$	1	-2	$f_2^2 = 9$
$\gamma_j^2$	-41	77/4	
$p_j^3 - z_j^{3(1)}$	3	-1	$f_1^3 = 18$
$q_j^3 - z_j^{3(2)}$	3	-2	$f_2^3 = 19$
$\gamma_j^3$	-53	453/4	

On applique la coupe  $x_8 \geq 1$  et on utilise la méthode duale du simplexe pour résoudre le problème  $IQP_7$ .

TABLE 3.8 – Solution entière efficace au nœud  $N_8$ .

$B_8$	$x_3$	$x_{10}$	RHS
$x_1$	1/3	4/3	0
$x_2$	0	-1	6
$x_4$	-2/3	-5/3	4
$x_5$	-1/3	-4/3	3
$x_7$	-1/3	-4/3	2
$x_6$	0	-1	2
$x_9$	-1/3	-4/3	1
$x_8$	0	-1	1
$\mu_j$	0	0	
$p_j^1 - z_j^{1(1)}$	-1/3	2/3	$f_1^1 = 13$
$q_j^1 - z_j^{1(2)}$	0	-1	$f_2^1 = 4$
$\gamma_j^1$	-4/3	-31/3	
$p_j^2 - z_j^{2(1)}$	-1	-3	$f_1^2 = 8$
$q_j^2 - z_j^{2(2)}$	-2/3	-5/3	$f_2^2 = 8$
$\gamma_j^2$	-40/3	-112/3	
$p_j^3 - z_j^{3(1)}$	-1/3	5/3	$f_1^3 = 20$
$q_j^3 - z_j^{3(2)}$	-2/3	1/3	$f_2^3 = 20$
$\gamma_j^3$	-20	40	

La solution optimale est entière  $x^{*(7)} = (0, 6)$ ,  $f(x^{*(7)}) = (52, 16, 400)$  et est donnée par la Table 3.8. Comme la solution est efficace, on mis à jour  $\mathcal{X}_E = \{(3, 3), (2, 4), (1, 5), (0, 6)\}$ ,  $H_7 = \{10\}$ . La coupe efficace  $x_{10} \geq 1$  ajoutée au problème  $IQP_7$  donne le problème  $IQP_8$  qui est irréalisable, le nœud est alors sondé.

$N_8$  : la contrainte  $x_1 \geq 2$  ajoutée à  $IQP_5$  permet d'obtenir le problème infaisable  $IQP_9$ , le nœud est sondé.

$N_6$  : la contrainte  $x_1 \geq 3$  ajoutée à  $IQP_3$  permet d'obtenir aussi un problème  $IQP_{10}$  infaisable, le nœud est alors sondé.

$N_2$  : on ajoute la contrainte  $x_1 \geq 4$  au problème  $IQP_0$ , on obtient le problème  $IQP_{11}$ , sa solution optimale  $x^{*(11)} = (4, 2)$ , donnée dans la Table 3.9, est entière.  $f(x^{*(11)}) = (72, 192, 192)$ .

TABLE 3.9 – Solution entière efficace au nœud  $N_9$ .

$B_9$	$x_5$	$x_4$	RHS
$x_1$	-1	0	4
$x_2$	2	1	2
$x_3$	-5	-4	4
$\mu_j$	1	2	
$p_j^1 - z_j^{1(1)}$	-2	-3	$f_1^1 = 9$
$q_j^1 - z_j^{1(2)}$	1	2	$f_2^1 = 8$
$\gamma_j^1$	-12	-11	
$p_j^2 - z_j^{2(1)}$	-1	1	$f_1^2 = 16$
$q_j^2 - z_j^{2(2)}$	-1	0	$f_2^2 = 12$
$\gamma_j^2$	12	-26	
$p_j^3 - z_j^{3(1)}$	-3	5	$f_1^3 = 12$
$q_j^3 - z_j^{3(2)}$	-3	-4	$f_2^3 = 16$
$\gamma_j^3$	-108	-66	

Mise à jour de l'ensemble  $\mathcal{X}_E = \{(3, 3), (2, 4), (1, 5), (0, 6), (4, 2)\}$ ,  $H_{11} = \{5\}$ .

En ajoutant la coupe efficace  $x_5 \geq 1$  au problème  $IQP_{11}$ , on obtient aura la Table 5.3.6. La solution optimale du problème obtenu ( $IQP_{12}$ ) est  $x^{*(12)} = (5, 0)$ , elle est entière,  $f(x^{*(12)}) = (60, 204, 84)$ . Mise à jour :  $\mathcal{X}_E = \{(3, 3), (2, 4), (1, 5), (0, 6), (4, 2), (5, 0)\}$  et  $H_{12} = \{6\}$ ,

TABLE 3.10 – Solution entière efficace au nœud  $N_{12}$ .

$B_{10}$	$x_6$	$x_4$	RHS
$x_1$	-1	0	5
$x_2$	2	1	0
$x_3$	-5	-4	9
$x_5$	-1	0	1
$\mu_j$	0	0	
$p_j^1 - z_j^{1(1)}$	-2	-3	$f_1^1 = 6$
$q_j^1 - z_j^{1(2)}$	1	2	$f_2^1 = 10$
$\gamma_j^1$	-18	-14	
$p_j^2 - z_j^{2(1)}$	-1	1	$f_1^2 = 17$
$q_j^2 - z_j^{2(2)}$	-1	0	$f_2^2 = 12$
$\gamma_j^2$	12	-29	
$p_j^3 - z_j^{3(1)}$	-3	-5	$f_1^3 = 7$
$q_j^3 - z_j^{3(2)}$	-3	-4	$f_2^3 = 12$
$\gamma_j^3$	-88	-57	

la coupe  $x_6 \geq 1$  ajoutée à  $IQP_{12}$  rend le problème  $IQP_{13}$  infaisable, le nœud est alors sondé.

L'algorithme s'arrête puisque tous les nœuds créés sont sondés et l'ensemble de toutes les solutions entières efficaces du problème *MOIIQP* est

$$\mathcal{X}_E = \{(3, 3), (2, 4), (1, 5), (0, 6), (4, 2), (5, 0)\}$$

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

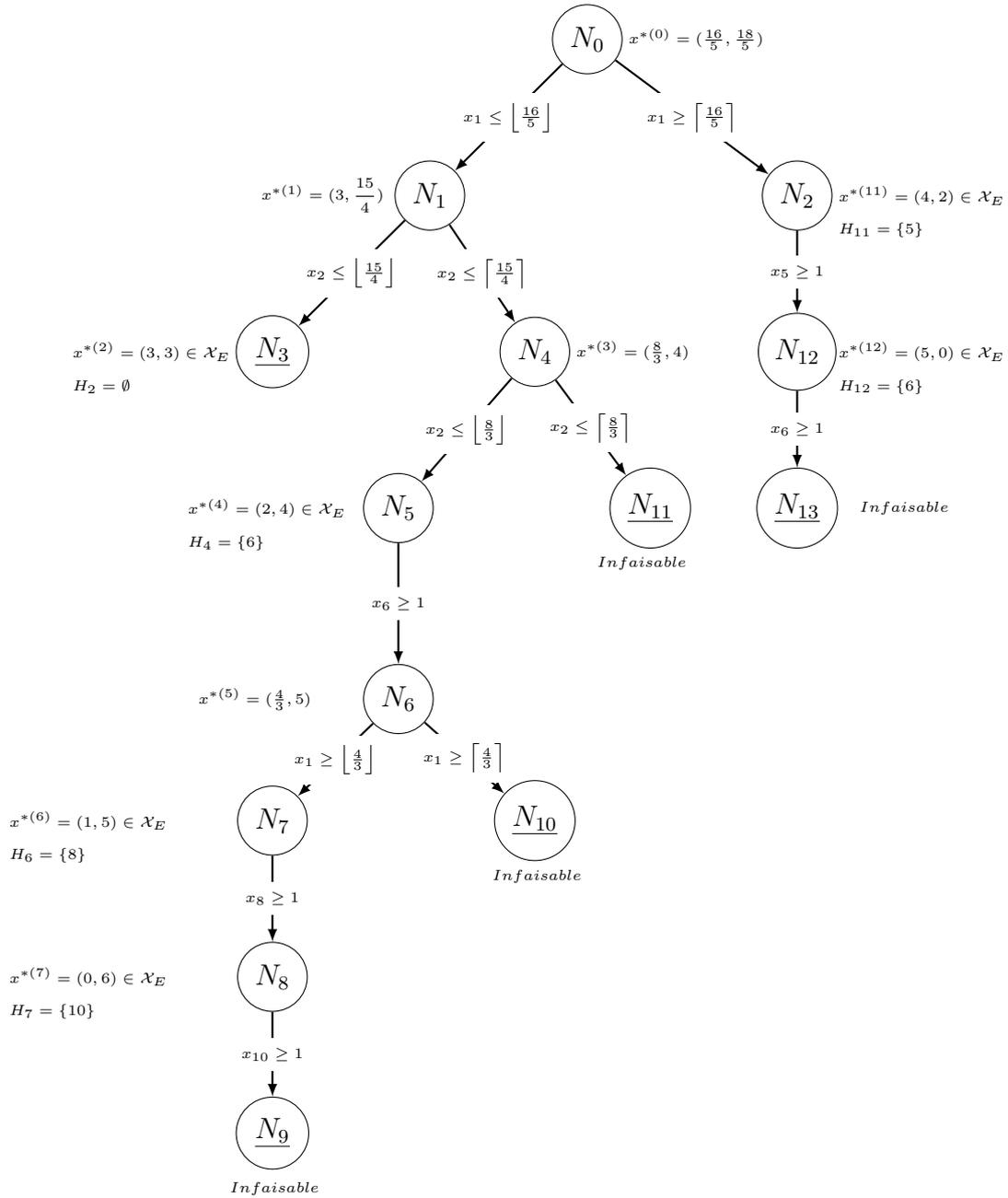


FIGURE 3.1 – Arborescence représentant les états des nœuds en utilisant l'algorithme Branch & Cut.

### 3.5 Expérimentation et résultats

Dans cette section, nous donnons quelques résultat relatifs à notre méthode qui a été mis en œuvre sous le langage de programmation Matlab et testé sur des instances générées aléatoirement suivant une loi uniforme à l'aide de la fonction `predefine` 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  $[v_{min}, v_{max}]$ .

$[v_{min}, v_{max}]$  est fixé à  $[1, 100]$  pour les coefficients de la matrice  $A$ , il varie dans l'intervalle  $[1, \sum_{i \in \{1, \dots, n\}} A_i^m]$  pour second membre  $b$ , si  $\sum_{i \in \{1, \dots, n\}} A_i^m \leq 100$ , et dans l'intervalle  $[100, \sum_{i \in \{1, \dots, n\}} A_i^m]$  si  $\sum_{i \in \{1, \dots, n\}} A_i^m > 100$ . Pour les vecteurs  $p^i, q^i$  et les scalaires  $\alpha^i, \beta^i$  sont générés dans  $[1, 100]$  afin d'assurer la condition de stricte positivité des facteurs  $(p^i x + \alpha^i)$  et  $(q^i x + \beta^i)$  pour  $i \in \{1, \dots, r\}$ . Pour chaque instance  $(n, m, r)$  ( $n$  est le nombre de variables,  $m$  le nombre de contraintes, et  $r$  le nombre d'objectifs), une série de 10 problèmes a été résolues.

L'application à été déployée sur une machine équipé d'un processeur IntelCore i3 CPU, 2.53GHZ et 2Go de mémoire.

Les performances obtenues de l'algorithme sont résumées dans la Table 3.11 où la moyenne du temps de calcul CPU (en secondes) et le nombre moyen des solutions efficaces trouvés sont rapportés. Aussi les valeurs minimale et maximale de chaque mesure sont indiquées entre parenthèses.

TABLE 3.11 – Résultats de calculs

$n$	$m$	$r$	CPU (secondes)	Nbre de solutions efficaces
			Moyenne [Min ;Max]	Moyenne [Min ;Max]
10	5	5	13,03 [1,54 ; 38,05]	17,65 [4 ; 46]
10	10	5	10,94 [2,43 ; 9,15]	16,25 [4 ; 34]
10	10	10	7,14 [3,18 ; 7,23]	19,24 [3 ; 46]
15	10	5	13,63 [10,83 ; 27,2]	24,80 [8 ; 54]
15	10	10	31,93 [9,7 ; 189,83]	48,2 [24 ; 128]
15	15	5	35,15 [5,61 ; 78,47]	53 [20 ; 157]
15	15	10	23,64 [23 ; 24,43]	37,5 [26 ; 75]
15	15	15	39,55 [7,73 ; 130,94]	72,7 [4 ; 205]
20	10	10	48,32 [6,25 ; 165,23]	78,23 [3 ; 190]
20	20	20	59,32 [3,88 ; 146,22]	98,2 [15 ; 208]
30	30	30	62,45 [5,13 ; 187,33]	201,3 [4 ; 243]
80	60	5	278,28 [42,04 ; 532,05]	68,32 [19 ; 212]
80	60	10	527,64 [24,73 ; 1421,31]	73,25 [14 ; 217]
80	60	20	1252,64 [254,12 ; 2154,25]	97,08 [26 ; 165]
100	80	5	1164,55 [168,88 ; 2371,25]	74,60 [5 ; 120]
100	80	10	1737,49 [307,7 ; 3011,83]	82,32 [27 ; 149]
100	80	20	2845,15 [594,39 ; 4178,32]	101 [24 ; 228]
140	100	5	2476,69 [721 ; 4632,48]	98,76 [13 ; 205]
140	100	10	3268,91 [1724,07 ; 5282,46]	89,13 [20 ; 190]
140	100	20	3712,32 [1746,25 ; 6215,83]	120,34 [29 ; 250]

### 3.5.1 Discussion des résultats

À travers ce tableau, on peut constater que l'algorithme proposé est efficace, en termes du temps d'exécution CPU, pour les petites et moyennes dimensions.

Cependant, pour les problèmes à grande dimensions, il devient irréaliste de générer l'ensemble efficace dans un temps raisonnable. Cela est dû à divers facteurs tels que :

- l'exactitude de la méthode,
- la non-linéarité des fonctions objectifs,
- le nombre important de critères et

- l'environnement discret du domaine de recherche.

La méthode proposée peut être utilisée pour des variables bornées et elle a été appliquée pour résoudre l'exemple de R. Arora et S. R. Arora dans [9].

Les résultats obtenus par notre algorithme et celui d'Arora sont rapportés dans le tableau 3.12.

TABLE 3.12 – Comparaison avec la méthode d'Arora [9]

Solutions Efficaces	Vecteurs non dominés $(f^1, f^2, f^3)$	
	Notre méthode	Méthode d'Arora
(3,0,0,0)	(180, 165, 108)	(180, 165, 108)
(2,0,1,0)	(165, 169, 110)	—
(2,0,2,0)	(156, 180, 117)	—
(2,0,1,1)	(144, 165, 143)	—
(2,0,2,1)	(130, 170, 150)	(130, 170, 150)
(3,1,1,0)	(170, 180, 104)	—
(3,1,2,0)	(165, 196, 105)	(165, 196, 105)
(3,1,0,1)	(160, 168, 130)	—
(3,1,1,1)	(154, 182, 135)	—
(4,2,1,0)	(171, 187, 90)	(171, 187, 90)
(4,2,0,1)	(162, 176, 120)	—

Il est montré que 7 solutions efficaces sur 11 ont été ignorées dans [9]. Notre algorithme a donc trouvé plus de solutions efficaces que l'algorithme d'Arora.

## 3.6 Conclusion

Dans ce chapitre, nous avons présenté une nouvelle méthode exacte qui résout les problèmes d'optimisation multiobjectif quadratique indéfini en nombres entiers 3.1. Le problème quadratique 3.2 est résolu en utilisant une variante de la méthode du simplexe et la solution entière est générée en utilisant le branchement bien connu dans la méthode de Branch & Bound. La coupe efficace proposée exploite tous les critères de la table simplexe, et seules les parties du domaine des solutions réalisables contenant des solutions efficaces sont explorées. L'utilisation de la coupe efficace 5.6 élimine un grand nombre de solutions réalisables et non efficaces. L'ensemble de toutes les solutions efficaces du problème *MOIIQP* est obtenu en un nombre fini d'itérations, si de telles solutions

existent. Cependant, afin de rendre l'algorithme plus puissant, nous suggérons, pour les travaux de recherche futurs, d'utiliser l'optimisation parallèle pour les problèmes à grande échelle.

# Chapitre 4

## Optimisation d'un critère sur l'ensemble efficient

### 4.1 Introduction

Dans certaines situations pratiques, l'énumération de tout l'ensemble efficient d'un problème multiobjectif 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 [76] et depuis plusieurs chercheurs, citons en particulier : Benson [16], [17], [18], [19], [20], Isermann [55], Yamamoto [97], Ecker et Song [42], Sayin [79], motivés par de nombreuses applications, se sont intéressés à l'optimisation d'une fonction sur l'ensemble efficient d'un problème multiobjectif 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 [74], 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. [5], 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 [56] 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. [25] proposèrent 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. Tous ces travaux traitent le cas (Linéaire-Linéaire) c'est à dire optimiser un critère linéaire sur l'ensemble efficient d'un problème multiobjectif linéaire discret *MOILP*. Très peu d'articles récent ont traité des cas plus généraux, à savoir le cas (Fractionnaire-Linéaire) qui consiste à optimiser un critère fractionnaire sur l'ensemble efficient d'un problème *MOILP* [6], le cas (Linéaire-Fractionnaire) qui optimise un critère linéaire sur un ensemble efficient d'un problème *MOILFP* [99], le cas (Fractionnaire-Fractionnaire) optimisant un critère fractionnaire linéaire sur l'ensemble efficient d'un problème *MOILFP* [100], le cas (Linéaire- Quadratique) qui, quand à lui, optimise un critère linéaire sur l'ensemble efficient d'un problème multiobjectif quadratique convexe [14] et plus récemment encore, le cas (Fractionnaire-Linéaire) qui optimise une fonction fractionnaire linéaire sur l'ensemble efficient d'un problème *MOILP* [64].

## 4.2 Formulation du problème

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

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

où  $\phi$  est une fonction continue de  $\mathbb{R}^n$  dans  $\mathbb{R}$ , et  $\mathcal{X}$  représente l'ensemble efficient du problème multiobjectif suivant :

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

avec  $x \in \mathbb{R}^n$ ,  $b \in \mathbb{R}^m$ ,  $A \in \mathbb{R}^{m \times n}$ ,  $f(x) = (f^1(x), f^2(x), \dots, f^r(x))$ ,  $r \geq 2$  est le nombre de critères du problème multiobjectif. L'ensemble  $S$  est supposé borné non vide.

Le reste de ce chapitre sera entièrement consacré à la description de quelques méthodes existantes pour l'optimisation d'une fonction sur l'ensemble efficient d'un problème multiobjectif dans le cas discret qui est le cadre de nos travaux.

## 4.3 Méthodes de résolution de $(P_E)$ dans le cas discret

### 4.3.1 Le cas (Linéaire-Linéaire)

Comme nous l'avons signalé précédemment, le cas discret n'a pas été aussi bien développé que le cas continu. La plupart de ces articles traitent le cas (Linéaire-Linéaire) c'est à dire optimiser un critère linéaire sur un ensemble efficace d'un problème multiobjectif linéaire discret (*MOILP*), qui se formule par :

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

où  $d \in \mathbb{R}^n$  et  $\mathcal{X}$  représente l'ensemble efficient du problème multiobjectif linéaire à variables entières suivant :

$$(MOILP) \begin{cases} \max Z_i(x) = c^i x, \quad i \in \{1, \dots, r\} \\ sc. \\ x \in \mathcal{D} \end{cases}$$

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

On définit  $(P_R)$  le problème relaxé suivant :

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

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

**Théorème 4.3.1 (Caractérisation d'une solution efficace [43])** *Soit  $x^*$  une solution quelconque de  $\mathcal{D}$ .  $x^*$  est efficace pour le problème (MOILP) si et seulement si la*

valeur optimale de la fonction objectif  $\theta$  est nulle dans le programme de programmation linéaire mixte suivant :

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

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

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

### La méthode de Jorge

Jorge a proposé dans [56] 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 problème *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 1 : 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  $\hat{x}^l \in \mathcal{X}$  dont le vecteur critère domine  $Cx^l$  et soit  $\tilde{x}^l$  une solution optimale du problème  $(T_l)$  suivant :

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

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 :

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

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**  $dx^{l+1} \leq \phi_{inf}$ , **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** [56] *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  $x^{l+1} \in \nabla$  alors  $x^{l+1}$  est une solution optimale de  $(P_E)$ .*

**Proposition 4.3.2** [56] *Soient  $\tilde{x}^1, \dots, \tilde{x}^l \in \mathcal{X}$ , si  $(R_l)$  est irréalisable, alors l'ensemble de toutes les solutions non dominées est  $SND(MOP) = C\tilde{x}^1, \dots, C\tilde{x}^l$ .*

### La méthode de Chaabane et al.

Les auteurs dans [25] 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  $\phi$  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 2** : 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  $k = 1$ ,  $H^0 = \mathcal{D}$ ,  $\mathcal{D} = \emptyset$ . Résoudre le problème  $(P_\lambda^0)$  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  $\mathcal{X} = \mathcal{X} \cup x_0$ . Aller à l'étape 3.

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

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

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

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

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  $\sum_{i=1}^r y_i^k \geq 1$  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  $\phi$  si elle existe.

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

**Sinon**, soit  $x_k$  la solution optimale de  $(P_\lambda^k)$ .

**Si**  $x^k$  est efficace, **alors**  $\mathcal{X} = \mathcal{X} \cup x_k$ ,  $x_{opt} = x_k$ ,  $\phi_{opt} = dx_{opt}$  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  $J_k = \{j \in N_k \mid Z_{\lambda,j} - c_j = 0\}$ .

**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 2.4.2 formule 2.5).

**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  $\mathcal{X} = \mathcal{X} \cup \{x'_k\}$ ,  $x_{opt} = x'_k$  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  $(P_\lambda^k)$  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  $\mathcal{X} = \mathcal{X} \cup \{x_k\}$ ,  $x_{opt} = x_k$  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  $\phi_{opt}$ . Le sous ensemble efficient obtenues qui améliorent la fonction  $\phi$  est  $\mathcal{X}$ .

---

### 4.3.2 Le cas (Fractionnaire-Linéaire)

Les deux méthodes présentées ci-dessus optimisent une fonction linéaire sur l'ensemble efficace d'un problème multiobjectif linéaire discret. Dans ce qui suit on présentera une méthode d'un cas plus général, proposée par Mahdi et Chaabane [64], et qui s'agit de l'optimisation d'une fonction fractionnaire linéaire sur l'ensemble efficace d'un problème multiobjectif linéaire discret, qui est l'objet d'une de nos contributions qui sera présentée dans le chapitre suivant.

Le problème de maximisation d'une fonction fractionnaire linéaire sur l'ensemble efficient d'un problème *MOILP* est défini comme suit :

$$(ILFP)_E \begin{cases} \max f(x) = \frac{p^T x + \alpha}{q^T x + \beta} \\ sc. \\ x \in \mathcal{X}_E. \end{cases} \quad (4.1)$$

où  $p, q$  sont des vecteurs de  $\mathbb{R}^n$ ,  $\alpha, \beta \in \mathbb{R}$ . On suppose que le facteur  $q^T x + \beta$  est positif pour tout  $x \in \mathcal{X}$ .  $\mathcal{X}_E$  représente l'ensembles des solutions efficaces du problème (*MOILP*) suivant : .

$$(MOILP) \begin{cases} \max Z^i(x) = Cx \\ sc. \\ x \in \mathcal{D} = \mathcal{X} \cap \mathbb{Z}^n. \end{cases} \quad (4.2)$$

$\mathbb{Z}$  est l'ensemble des entiers,  $r \geq 2$ ,  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ ,  $C = (c^1, c^2, \dots, c^r)^T$ , où  $c^i \in \mathbb{R}^n$ .  $\mathcal{X} = \{x \in \mathbb{R}^n | Ax \leq b, x \geq 0\}$ .  $\mathcal{D}$  un ensemble non vide.

### La méthode de Mahdi et Chaabane

Les auteurs dans [64] proposent un algorithme exact d'optimisation d'une fonction fractionnaire linéaire sur l'ensemble efficace d'un problème de programmation linéaire à objectifs multiples (*MOILP*) sans avoir à énumérer toutes les solutions efficaces. Dans cette méthode, des contraintes sont ajoutées d'une manière itérative pour éliminer les points non intéressants et réduire progressivement la région admissible. À chaque itération, une solution entière est trouvée pour laquelle une nouvelle direction qui améliore la fonction objectif est définie. L'algorithme est présenté comme suit :

---

#### Algorithme 3 : MAHDI ET CHAABANE

---

- Initialisation :**
- Trouver le minimum de chaque objectif  $Z^i(x)$ ,  $\forall i = 1, \dots, r$ , en résolvant le problème  $-M_i = \min\{c^i x | x \in \mathcal{D}\}$ , si  $c_j^i \geq 0$ ,  $j = 1, \dots, n$ , sinon  $M_i = 0$ ;
  - On pose  $f_{opt} = -\infty$  avec  $f_{opt}$  représente la valeur optimale du critère principale  $f(x)$ ,  $k = 0$ , Fin=Faux.

**Tant que Fin=Faux faire**

Résoudre le problème relaxé

$$(ILFP)_R^k = (ILFP)_R = \max\{f(x) = \frac{px + \alpha}{qx + \beta} | x \in \mathcal{D}\};$$

**Si**  $(ILFP)_R$  est irréalisable, **alors**  $(ILFP)_E$  n'admet pas de solution, Fin=Vrai;

**Sinon**, soit  $x^k$  une solution optimale du problème  $(ILFP)_R^k$ .

**Test d'efficacité** : Résoudre le problème

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

où  $I$  est une matrice identité d'ordre  $r$  et  $\psi = (\psi_i)_{i=1, \dots, r}$ .

**Si**  $x^k \in \mathcal{X}_E$ , **alors**  $x_{opt} = x^k$  et  $f_{opt} = f(x^k)$ . Fin=vrai.

**Sinon**, soit  $\hat{x}^k$  une solution optimale du problème  $T_{eff}(x^k)$ .

Résoudre le problème  $(FT_k) = \max\{f(x) = \frac{px + \alpha}{qx + \beta} | Cx = C\hat{x}^k, x \in \mathcal{D}\}$ .

Soit  $\bar{x}^k$  une solution optimale du problème  $(FT_k)$ , **Si**  $f(\bar{x}^k) > f_{opt}$ , **alors**  $x_{opt} = \bar{x}^k$  et  $f_{opt} = f(\bar{x}^k)$ ;  $k=k+1$ .

Résoudre le problème  $(FP_k) = \max\{f(x) | x \in \mathcal{D}_k = \mathcal{D} \setminus \bigcup_{s=0}^{k-1} \mathcal{D}_s\}$ .  $\mathcal{D}_s = \{x \in \mathbb{Z}^n | Cx \leq C\bar{x}^s\}$ .

**Si**  $(FP_k)$  est infaisable **alors** Fin=vrai;

**Sinon** Soit  $x^k$  une solution optimale pour le problème  $(FP_k)$ .

**Si**  $x^k \in \mathcal{X}$ , **alors**  $x_{opt} = \bar{x}^k$  et  $f_{opt} = f(\bar{x}^k)$ . Fin=vrai.

**Sinon** Soit  $\hat{x}^k$  soit une solution optimale du problème  $(T_{eff}(x^k))$ .

Soit  $J_k = \{j \in N_k | \gamma_j = 0\}$ ;

**Tant que**  $J \neq \emptyset$  **faire**

Soit  $\hat{x}^k$  une solution optimale du problème  $(FP_k)$ .

Sélectionner  $j_k \in J_k$ ;  $\theta_{j_k}^* = \text{int} \left( \min \left\{ \frac{x^{k_i}}{y_{k, i j_k}} \right\}; y_{k, i j_k} > 0 \right)$ ;

Explorer l'arête  $E_{j_k}$  : recherche des solutions alternatives entières correspondantes à  $\theta_{j_k}$ ,  $\theta_{j_k} \in \{1, \dots, \theta_{j_k}^*\}$  commençant de  $\theta_{j_k}^*$  à 1.

**Si** une telle solution existe, **alors** on la note  $\tilde{x}$ ,

$x_{opt} = \tilde{x}$  et  $f_{opt} = f(\tilde{x})$ ; Fin=vrai.

**Fin Tant que.**

**Fin Tant que.**

---

## 4.4 Conclusion

Dans ce chapitre, nous avons présenté quelques méthodes de résolution du problème d'optimisation d'un critère donné sur l'ensemble des solutions efficaces d'un problème multiobjectif où les variables de décision sont entières. Nous avons pris en considération le cas où la fonction principale est linéaire ou fractionnaire linéaire et vu la pauvreté de la littérature relatant le problème dans le cas non linéaire, ceci, nous a d'avantage motivé pour explorer cette classe de problème en mettant au point une nouvelle méthode exacte de résolution d'un problème d'optimisation d'un critère fractionnaire linéaire sur un ensemble efficace d'un problème multiobjectif linéaire en nombres entiers que nous allons présentée dans le chapitre suivant.

## Chapitre 5

# Méthodes d'optimisation d'un critère non linéaire sur l'ensemble efficient d'un problème multiobjectif linéaire discrêt

Dans ce chapitre, deux méthodes exactes sont proposées ; une pour résoudre le problème d'optimisation d'un critère quadratique indéfini sur l'ensemble efficient d'un problème multiobjectif linéaire discrêt *MOILP* et l'autre pour résoudre le problème d'optimisation d'un critère fractionnaire linéaire sur l'ensemble efficient d'un problème *MOILP*. En effet, nous avons développé les deux méthodes en utilisant le principe de Branch & Cut et en se basant sur la résolution d'un programme non linéaire à variables continues pour permettre de retrouver une solution optimale entière aux problèmes sans avoir à énumérer toutes les solutions efficaces du problème multiobjectif *MOILP*.

### 5.1 Introduction

Le problème sous considération est un problème de maximisation d'un critère non linéaire notée  $f(x)$  sur l'ensemble efficient  $\mathcal{X}_E$  dit aussi ensemble Pareto optimal d'un problème de maximisation multiobjectif linéaire à variables entières *MOILP*. Ce problème est en principe très difficile à résoudre [75], 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 ce, nous avons pris en considération deux fonctions différentes non linéaires :

- La fonction quadratique indéfinie écrite sous forme d'un produit de deux fonctions linéaires positives qui est une fonction quasi-concave, le critère principale  $f(x)$  s'écrit alors sous la forme  $f(x) = (p^T x + \alpha)(q^T x + \beta)$ .
- La fonction fractionnaire linéaire qui s'écrit sous la forme d'un rapport de deux fonctions linéaires et qui est une fonction quasi-linéaire, le critère  $f(x)$  s'écrit sous la forme  $f(x) = \frac{p^T x + \alpha}{q^T x + \beta}$ .

A notre connaissance, et comparant au cas linéaire, résoudre les problèmes sous considération n'a pas reçu beaucoup d'attention malgré leur importance et leurs applications dans la vie réelle.

La première méthode pour l'optimisation d'un critère linéaire sur l'ensemble efficace discret fut proposée en 1992 par Nguyen [74], 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. [5] 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 [56] 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. Très récemment, en 2010, Chaabane et Pirlot [25] proposèrent une méthode de résolution dans l'espace des critères dans laquelle la valeur de la fonction objectif principale est améliorée en optimisant une somme pondérée des critères à chaque itération.

Des travaux récents traitant des cas plus généraux ont été proposés. Zerdani et Moulai [99] ont optimisé une fonction linéaire sur l'ensemble efficient d'un problème multiobjectif fractionnaire discret en utilisant le test d'efficacité de Ehrgott [44]. En 2004, Belkeziz et Metrane [14] optimisent un critère linéaire sur l'ensemble efficient d'un problème multiobjectif quadratique convexe ; les auteurs ramènent ainsi le problème de la minimisation d'une fonction linéaire sur l'ensemble des points efficients à la résolution d'un problème de programmation fractionnaire. Récemment Mahdi et Chaabane [64] ont optimisé une fonction fractionnaire linéaire sur l'ensemble efficient d'un problème *MOILP* en utilisant la technique des coupes planes semblable à la méthode donnée par Chaabane et Pirlot dans [25].

La première section de ce chapitre, traite le cas où le critère principale à optimiser est une fonction quadratique indéfinie et la deuxième sera consacrée au cas d'une fonction fractionnaire linéaire.

## 5.2 Optimisation d'une fonction quadratique indéfinie sur l'ensemble efficient d'un problème multiobjectif linéaire discrêt

### 5.2.1 Définitions et notations

Le problème de programmation multiobjectif linéaire discrêt (*MOILP*) est formulé comme suit :

$$(MOILP) \begin{cases} \max Z^i(x) = c^i x & i \in \{1, \dots, r\} \\ sc. \\ x \in \mathcal{D} = \mathcal{X} \cap \mathbb{Z}^n. \end{cases} \quad (5.1)$$

$\mathbb{Z}$  est l'ensemble des entiers,  $r \geq 2$ ,  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ ,  $C = (c^1, c^2, \dots, c^r)^T$ , où  $c^i \in \mathbb{R}^n$  est un vecteur ligne.  $\mathcal{X} = \{x \in \mathbb{R}^n | Ax \leq b, x \geq 0\}$ . Dans tout ce qui suit, on suppose que  $\mathcal{X}$  est un polyèdre convexe, borné, fermé et non vide.  $\mathcal{D}$  un ensemble non vide.

Une solution efficace du problème (*MOILP*) qui maximise une fonction d'utilité quadratique indéfinie  $f$  du décideur, qui n'est pas nécessairement une combinaison des fonctions objectives du problème (*MOILP*), peut être obtenu en résolvant le programme suivant :

$$(IQP)_E \begin{cases} \max f(x) = (p^T x + \alpha)(q^T x + \beta) \\ sc. \\ x \in \mathcal{X}_E \subset \mathcal{D}. \end{cases} \quad (5.2)$$

où  $p, q$  sont des vecteur de  $\mathbb{R}^n$ ,  $\alpha, \beta \in \mathbb{R}$  et  $\mathcal{X}_E$  représente l'ensembles des solutions efficaces du problème (*MOILP*). On suppose que les facteurs  $p^T x + \alpha$  et  $q^T x + \beta$  sont positifs pour tout  $x \in \mathcal{X}$ .

Une application de cette classe de fonctions quadratiques peut être trouvé dans la théorie microéconomique [53] et les modèles prédictifs de contrôle [10].

Malgré son importance dans les applications réelles, une recherche bibliographique nous a permis de conclure que la résolution du problème  $(IQP)_E$  n'a pas encore été étudiée et que le cas non linéaire [75, 14] n'a pas reçu beaucoup d'attention comme il se doit, comparé au cas linéaire [16, 97].

La stratégie adoptée pour générer une solution optimale entière du problème principal  $(QP)_E$  est basée sur la résolution d'une séquence de problèmes quadratique indéfini

continus  $(QP)_l$ , à chaque étape  $l$ ,  $l \geq 0$ , définie par :

$$(IQP)_l \begin{cases} \max f(x) = (p^T x + \alpha)(q^T x + \beta) \\ sc. \\ x \in \mathcal{X}_l. \end{cases} \quad (5.3)$$

$\mathcal{X}_0 = \mathcal{X}$  et  $\mathcal{X}_l$  est un sous ensemble de l'ensemble original  $\mathcal{X}$  à explorer dans l'étape  $l$ .

La recherche d'une solution optimale du problème  $(IQP)_l$  nécessite l'introduction des notations suivantes :

- $x_B$  est une solution réalisable de base du problème (5.3) correspondante à la matrice de base  $B$  tel que  $x_B = B^{-1}b$ .
- $I$  est l'ensemble des indices des variables de base correspondante à  $x_B$ .
- $N$  est l'ensemble des indices des variables hors-base correspondante à  $x_B$ .

De plus,

- $f(x_B) = f_1(x_B)f_2(x_B)$  tel que,

$$\begin{aligned} f_1(x_B) &= p_B^T x_B + \alpha = p_B^T B^{-1}b + \alpha \\ f_2(x_B) &= q_B^T x_B + \beta = q_B^T B^{-1}b + \beta. \end{aligned}$$

- Aussi, on assume que, à cette solution réalisable de base, on a

$$\begin{aligned} z_j^{(1)} &= p_B^T B^{-1}a_j \\ z_j^{(2)} &= q_B^T B^{-1}a_j, \end{aligned}$$

définies pour chaque colonne  $a_j = \sum_{i=1}^m y_{ij}b_i$ ,  $j \in N$ .

Si la nouvelle solution réalisable de base est notée  $\hat{x}_B$ , alors  $\hat{x}_B = \hat{B}^{-1}b$ , la nouvelle valeur de la fonction objectif est

$$f(\hat{x}_B) - f(x_B) = \boldsymbol{\mu}_j \boldsymbol{\gamma}_j. \quad (5.4)$$

avec

$$\boldsymbol{\mu}_j = \frac{(x_B)_k}{y_{kj}} = \min \left\{ \frac{(x_B)_i}{y_{ij}} \mid y_{ij} > 0, i \in I \right\}, j \in N$$

$$\boldsymbol{\gamma}_j = f_2(x_B)(p_j - z_j^{(1)}) + f_1(x_B)(q_j - z_j^{(2)}) + \boldsymbol{\mu}_j(p_j - z_j^{(1)})(q_j - z_j^{(2)}),$$

Le théorème suivant nous donne la condition d'optimalité du problème (3.2).

**Théorème 5.2.1** *Étant donnée une solution réalisable  $x_B = B^{-1}b$  du problème (3.2), tel que  $\gamma_j \leq 0$  pour chaque colonne  $a_j$  dans  $A$  non dans  $B$ , alors  $x_B$  est une solution réalisable de base optimale.*

## 5.2.2 La coupe efficace

Afin de décrire notre procédure pour générer des solutions efficaces du programme  $(IQP)_E$ , on utilise les notations suivantes :

- $x^{*(l)}$  est la première solution entière obtenue en résolvant le problème 3.2 en utilisant, éventuellement, le processus de branchement bien connu dans les techniques de Branch & Bound.
- $I_l$  désigne l'ensemble d'indices des variables de base correspondante à  $x^{*(l)}$ .
- $N_l$  désigne l'ensemble d'indices des variables hors base correspondante à  $x^{*(l)}$ .
- $\bar{c}_j^i$  est la  $j^{\text{ième}}$  composante du vecteur gradient  $c^i$  pour chaque fonction objectif  $Z^i$ ,  $i \in \{1, 2, \dots, r\}$  défini dans le dernier tableau du simplexe.

Notons que notre approche est basée sur la méthode de Branch&Cut, de sorte que, à chaque étape  $l$  de l'algorithme une coupe efficace [30] est ajouté au domaine initial pour supprimer des solutions non efficaces pour le problème multiobjectif (*MOILP*).

Avant de déterminer l'expression mathématique de cette coupe efficace, il faut d'abord définir l'ensemble suivant à la solution entière  $x^{*(l)}$  :

$$H_l = \left\{ j \in N_l \mid \exists i \in \{1, \dots, r\}, \bar{c}_j^i > 0 \right\} \cup \left\{ j \in N_l \mid \bar{c}_j^i = 0, \forall i \in \{1, \dots, r\} \right\} \quad (5.5)$$

et la coupe

$$\sum_{j \in H_l} x_j \geq 1 \quad (5.6)$$

est efficace. Therefore, l'ensemble

$$\mathcal{X}_{l+1} = \left\{ x \in \mathcal{X}_l \mid \sum_{j \in H_l} x_j \geq 1 \right\}. \quad (5.7)$$

### 5.2.3 Test d'efficacité

Étant donnée  $x^{*(l)}$  une solution optimale entière du problème  $(IQP)_l$ , On peut tester son efficacité en résolvant le programme linéaire mixte suivant ([43]) :

$$EK(x^{*(l)}) \left\{ \begin{array}{l} \max \theta = \sum_{i=1}^r \psi_i \\ sc. \\ Cx - I\psi = Cx^{*(l)}. \\ x \in \mathcal{D}; \\ \psi_i \in \mathbb{R}^+, \forall i \in \{1, \dots, r\}, \end{array} \right. \quad (5.8)$$

avec  $C = (c^1, c^2, \dots, c^r)^T$  est une matrice d'ordre  $(r \times n)$ ,  $I$  est une matrice d'identité d'ordre  $r$  et  $\psi = (\psi_i)_{i \in \{1, \dots, r\}}$ .

Un point  $x^{*(l)}$  est efficace si et seulement si la valeur optimale de la fonction objectif  $\theta$  en  $EK(x^{*(l)})$  est nulle (voir [54]).

Si la solution  $x^{*(l)}$  n'est pas efficace, la solution optimale  $\hat{x}^{(l)}$  obtenue en résolvant le programme  $EK(x^{*(l)})$  est efficace.

**Théorème 5.2.2** [43] *Si la valeur optimale de la fonction objectif  $\theta$  du programme  $EK(x^{*(l)})$  est non nulle, atteinte à une solution entière  $\hat{x}^{(l)}$ , alors  $\hat{x}^{(l)}$  est efficace.*

### 5.2.4 Développement de la méthode

#### Description de la méthode

Soit  $f_{opt}$  la valeur optimale de la fonction objectif du problème  $(IQP)_E$  initialisé à  $-\infty$  auquel aucune solution entière efficace ne lui correspond encore ( $x_{opt}$  inconnu au début). Initialement, pour résoudre le programme  $(IQP)_0$ , seule la méthode de Swarup [93] est requise. Nous commençons par la résolution d'un problème  $(IQP)_l$  défini par programme 5.3 en utilisant la version variante du simplexe méthode à l'étape  $l$  de l'algorithme (éventuellement la méthode dual du simplexe). Ensuite, pour mettre à jour le vecteur gradient réduit  $c^i$ ,  $i \in \{1, \dots, r\}$ ,  $r$  lignes sont ajoutées au tableau du simple et les coûts réduits sont calculés par rapport à la base correspondante.

Le nœud  $l$  est saturé si le programme 5.3 correspondant est irréalisable.

Dès qu'une solution entière  $x^{*(l)}$  est trouvée au nœud  $l$ , son efficacité est testée en résolvant le programme linéaire mixte 5.8. Ici, la valeur de  $f_{opt}$  peut éventuellement être améliorée et mise à jour.

Si  $x^{*(l)}$ , avec la valeur critère correspondante  $f(x^{*(l)})$ , est efficace, alors le nœud  $l$  est sondé. De plus, si  $f_{opt} < f(x^{*(l)})$ , alors nous mettons à jour  $x_{opt} = x^{*(l)}$  et  $f_{opt} = f(x^{*(l)})$ .

Si la solution entière  $x^{*(l)}$  n'est pas efficace, une solution optimale  $\hat{x}^{(l)}$  obtenue par la résolution du programme  $EK(x^{*(l)})$  est efficace. Donc  $x_{opt} = \hat{x}^{(l)}$ , avec une valeur de la fonction objectif correspondante  $f_{opt} = f(\hat{x}^{(l)})$  si cette solution est meilleure que la solution entière efficace précédente déjà trouvée. Donc, une nouvelle coupe [30] et ajoutée à la table du simplexe courante, ce qui nous permet d'éviter des solutions non efficace du problème 5.2 et de déterminer une nouvelle solution entière. Cependant, si aucune amélioration des critères ne peut être faits le long du domaine restant ( $H_l = \emptyset$ ), le nœud  $l$  est saturé.

Si la solution optimale  $x^{*(l)}$  du programme  $(IQP_l)$  n'est pas entière, soit  $x_j^{*(l)}$  une composante non entière de  $x^{*(l)}$ . Le nœud  $l$  de l'arborescence est alors séparé en deux nœuds qui lui sont imposées par les contraintes additionnelles  $x_j \leq \lfloor x_j^{*(l)} \rfloor$  et  $x_j \leq \lceil x_j^{*(l)} \rceil$  où  $\lfloor x_j^{*(l)} \rfloor$  indique la partie entière inférieure du nombre réel  $x_j^{*(l)}$  et  $\lceil x_j^{*(l)} \rceil$  indique la partie entière supérieure du nombre réel  $x_j^{*(l)}$ . Dans chaque nœud, le programme quadratique indéfini obtenu doit être résolu jusqu'à obtention d'une solution entière si elle existe.

La méthode se termine quand tous les nœuds créés seront sondés. La solution entière optimale du problème  $(IQP)_E$  est alors  $x_{opt}$  qui correspond à  $f_{opt}$ .

## Formulation de l'algorithme

L'algorithme générant une solution optimale pour le problème 5.2 est présenté comme suit :

---

### Algorithme 1 : L'optimisation quadratique indéfinie sur l'ensemble efficace

---

**Étape 1 : Initialisation**  $l = 0$ ,  $\mathcal{X}_0 = \mathcal{X}$ ,  $f_{opt} = -\infty$

**Étape 2 : Étape Générale** Tant qu'il existe un nœud non encore sondé dans l'arborescence, choisir le nœud  $l$  non encore sondé et résoudre le programme quadratique correspondant  $(IQP_l)$  en utilisant la version modifiée de la méthode de simplexe [93], ou la méthode dual du simplexe.

- Si  $(IQP_l)$  est infaisable, alors le nœud correspondant  $l$  est sondé ;
- Sinon, soit  $x^{*(l)}$  une solution optimale du problème  $(IQP_l)$  ;
  1. Si  $f_{opt} \geq f(x^{*(l)})$ , alors le nœud  $l$  est sondé et aller à l'**étape 2** ;
  2. Sinon, tester l'intégrité de la solution  $x^{*(l)}$ .

- (a) Si  $x^{*(l)}$  est entière, aller à l'**étape 2a**.
- (b) Sinon, aller à l'**étape 2b**.

**Étape 2a :** Résoudre le problème (5.8) au point entier  $x^{*(l)}$ .

- Si  $x^{*(l)}$  est efficace, le nœud  $l$  est sondé. Mettre à jour  $f_{opt}$ , si nécessaire, et aller à l'**étape 2**;
- Sinon, soit  $\hat{x}^{(l)}$  une solution optimale du problème  $EK(x^{*(l)})$ . Mettre à jour  $f_{opt}$ , si nécessaire, et déterminer les ensembles  $N_l, H_l$ ;
  1. Si  $H_l = \emptyset$ , alors le nœud  $l$  est sondé et aller à l'**étape 2**;
  2. Sinon, ajouter la coupe efficace (5.6) au problème  $(IQP_l)$ ,  $l = l + 1$  et aller à l'**étape 2**.

**Étape 2b : Processus de Branchement** Puisque parmi les valeurs  $x^{*(l)}$  il existe au moins une composante dont la valeur est non entière, choisir une d'entre elles, notée  $x_j^{*(l)}$ , et partitionner l'ensemble réalisable  $\mathcal{X}_l$  en deux sous ensembles ( $\mathcal{X}_{l_1}$  et  $\mathcal{X}_{l_2}$ ) ( $l_1 > l + 1$ ,  $l_2 > l + 1$ ,  $l_1 \neq l_2$ ) en ajoutant à  $\mathcal{X}_l$  les nouvelles contraintes de branchement  $x_j \leq \lfloor x_j^{*(l)} \rfloor$  et  $x_j \geq \lceil x_j^{*(l)} \rceil$  respectivement et passer à l'Étape 2.

---

## Résultats fondamentaux

Les résultats théoriques suivants sont établis afin de justifier les différentes étapes de l'algorithme proposé :

**Théorème 5.2.3** *Supposons que  $H_l = \emptyset$  à la solution entière non efficace  $x^{*(l)}$  obtenue au nœud  $l$ . Si  $x$  est une solution entière efficace dans le domaine  $\mathcal{X}_l \setminus \{x^{*(l)}\}$ , alors  $x \in \mathcal{X}_{l+1}$ .*

**Preuve 5.2.1** *Soit  $x$  une solution entière dans l'ensemble  $\mathcal{X}_l \setminus \{x^{*(l)}\}$  tel que  $x \notin \mathcal{X}_{l+1}$ , alors  $x \notin \left\{ x \in \mathcal{X}_l \mid \sum_{j \in H_l} x_j \geq 1 \right\}$ , ce qui implique  $x \in \left\{ x \in \mathcal{X}_l \mid \sum_{j \in N_l \setminus H_l} x_j \geq 1 \right\}$ . Par conséquent, les inégalités suivantes sont satisfaites :*

$$\begin{aligned} \sum_{j \in H_l} x_j &< 1, \\ \sum_{j \in N_l \setminus H_l} x_j &\geq 1. \end{aligned}$$

Il en résulte que  $x_j = 0$  pour tout  $j \in H_l$  et  $x_j \geq 1$  pour au moins un indice  $j \in N_l \setminus H_l$ .

Dans la table du simplexe correspondante à la solution entière optimale  $x^{*(l)}$ , la valeur mise à jour de chaque fonction objectif  $Z^i$  est écrite en fonction des indices de base  $j \in I_l$  et des indices hors base  $j \in N_l$ . Ainsi, l'égalité suivante est valable pour tous les critères  $i \in \{1, \dots, r\}$  :

$$Z^i(x) = c^i x = \sum_{j \in I_l} \bar{c}_j^i x_j + \sum_{j \in N_l} \bar{c}_j^i x_j, \quad \text{où} \quad \sum_{j \in I_l} \bar{c}_j^i x_j = Z^i(x^{*(l)}).$$

Alors, on peut écrire :

$$\begin{aligned} Z^i(x) - Z^i(x^{*(l)}) &= \sum_{j \in N_l} \bar{c}_j^i x_j \\ &= \sum_{j \in H_l} \bar{c}_j^i x_j + \sum_{j \in N_l \setminus H_l} \bar{c}_j^i x_j \\ &= \sum_{j \in N_l \setminus H_l} \bar{c}_j^i x_j. \end{aligned}$$

Donc,  $Z^i(x) \leq Z^i(x^{*(l)})$  pour tout  $i \in \{1, \dots, r\}$ , avec  $Z^i(x) < Z^i(x^{*(l)})$  pour au moins un critère  $i \in \{1, \dots, r\}$  since  $\bar{c}_j^i \leq 0$  pour tout  $j \in N_l \setminus H_l$  et pour au moins un indice  $j$ , il est strictement inférieure à zéro. Par conséquent, le vecteur critère  $(Z^1(x), \dots, Z^r(x))$  est dominé par le vecteur critère  $((Z^1(x^{*(l)}), \dots, Z^r(x^{*(l)})))$  et  $x$  n'est pas efficace.

**corrolaire 5.2.1** La contrainte  $\sum_{j \in H_l} x_j \geq 1$  définie une coupe efficace.

**Preuve 5.2.2** Il est clair que  $\sum_{j \in H_l} x_j \geq 1$  est une contrainte efficace valide par le théorème ci-dessus, puisque toutes les solutions efficaces entières dans le domaine actuel  $\mathcal{X}_l$  vérifient cette contrainte. De plus, la solution entière actuelle  $x^{*(l)}$  ne satisfait pas cette contrainte, puisque  $x_j = 0$  pour tout  $j \in H_l$ . En conclusion, on peut dire que la contrainte  $\sum_{j \in H_l} x_j \geq 1$  est une coupe efficace.

**Proposition 5.2.1** Si  $H_l = \emptyset$  à la solution entière actuelle  $x^{*(l)}$ , alors  $\mathcal{X}_l \setminus \{x^{*(l)}\}$  est un domaine exploré.

**Preuve 5.2.3**  $H_l = \emptyset$  cela veut dire que  $x^{*(l)}$  est une solution optimale entière pour tous les critères, hence  $x^{*(l)}$  est un point idéal dans le domaine  $\mathcal{X}_l$  et  $\mathcal{X}_l \setminus \{x^{*(l)}\}$  ne contient pas de solutions efficaces.

**Théorème 5.2.4** L'algorithme converge vers une solution optimale pour le problème  $QP_E$ , si une telle solution existe, en un nombre fini d'itérations.

**Preuve 5.2.4** Comme  $D$ , l'ensemble des solutions entières réalisables pour le problème MOILP, est un ensemble fini borné contenu dans  $\mathcal{X}$ , la cardinalité de l'ensemble efficace  $\mathcal{X}_E$  est aussi un nombre fini. À chaque fois qu'une solution optimale entière  $x^{*(l)}$  est trouvée, la coupe efficace est ajoutée si  $x^{*(l)} \notin \mathcal{X}_E$  ou si aucune amélioration de la valeur  $f_{opt}$  n'est possible. Ainsi, selon le théorème et le corollaire ci-dessus, au moins la solution  $x^{*(l)}$  est éliminée lorsque on étudie un sous-problème  $(P_k)$ ,  $k > l$ , mais aucune solution efficace entière n'est omise.

Dans le cas où, l'ensemble de décision  $\mathcal{X}_l$  ne contient aucune solution entière optimale, le sous-arbre enraciné au nœud  $l$  est exploré et tous les autres nœuds sont explorés après un nombre fini d'étapes.

### 5.2.5 Exemple illustratif

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

$$(QP_E) \quad \begin{cases} \max & \phi(x) = (2x_1 + x_2 + 1)(x_1 + x_2 + 2) \\ \text{s.c.} & \\ & x \in \mathcal{X}_E, \end{cases}$$

où  $\mathcal{X}_E$  représente l'ensemble des solutions entières efficaces du problème MOILP définie par

$$(MOILP) \quad \begin{cases} \max & Z^1 = 3x_1 + x_2 \\ \max & Z^2 = -2x_1 - x_2 \\ \text{s.c.} & \\ & 2x_1 + x_2 \leq 6 \\ & x_1 - x_2 \leq 2 \\ & x_2 \leq 4 \\ & x_1 \geq 0, x_2 \geq 0 \\ & x_1, x_2 \text{ entiers.} \end{cases}$$

**Initialization** : Soit  $\phi_{opt} = -\infty$ ,  $l = 0$ . Le programme  $(P_0)$  est résolu. La première solution optimale est  $x^{*(0)} = (1, 4)$ , qui est entière, est donnée par la Table 5.1 et on a  $f(x^{*(0)}) = 49$ .

TABLE 5.1 – La première solution optimale entière

$B_0$	$x_3$	$x_5$	RHS
$x_1$	1/2	-1/2	1
$x_2$	0	1	4
$x_4$	-1/2	3/2	5
$p_j - z_j^{(1)}$	-1	0	$f_1=7$
$q_j - z_j^{(2)}$	-1/2	-1/2	$f_2=7$
$\mu_j$	2	10/3	
$\gamma_j$	-19/2	-7/2	
$\bar{c}^1$	-3/2	1/2	
$\bar{c}^2$	1	0	

En testant l'efficacité de la solution, on obtient une solution optimale du problème  $EK(x^{*(0)})$ ,  $\hat{x}^0 = (2, 1)$ , qui est, selon le théorème (4.3.1), une solution entière efficace du problème  $MOILP$ . Mise à jour :  $x_{opt} = \hat{x}^0 = (2, 1)$ ,  $f_{opt} = f(\hat{x}^0) = 30$ , et  $H_0 = \{3, 5\}$ .

On applique la coupe efficace  $x_3 + x_5 \geq 1$  et en utilisant la méthode dual du simplexe on obtient la solution non entière  $x^{*(1)} = (\frac{3}{2}, 3)$ ,  $f(x^{*(1)}) = 91/2$  donnée par la Table 5.2 :

TABLE 5.2 –

$B_1$	$x_3$	$x_6$	RHS
$x_1$	1	-1/2	3/2
$x_2$	-1	1	3
$x_4$	-2	3/2	7/2
$x_5$	1	-1	1
$p_j - z_j^{(1)}$	-1	0	$f_1=7$
$q_j - z_j^{(2)}$	0	-1/2	$f_2=13/2$
$\mu_j$	1	7/3	
$\gamma_j$	-13/2	-7/2	
$\bar{c}^1$	-2	1/2	
$\bar{c}^2$	1	0	

Comme la solution n'est pas entière, le processus de branchement génère deux nœuds  $N_2, N_3$  correspondant aux contraintes suivantes :

$$N_2 : x_1 \leq 1.$$

$$N_3 : x_1 \geq 2.$$

$N_2$  : la contrainte  $x_1 \leq 1$  est ajoutée à la Table 5.2 pour avoir la Table 5.3 avec une solution optimale  $x^{*(2)} = (1, \frac{7}{2})$  et  $f(x^{*(2)}) = 169/4$ .

TABLE 5.3 –

$B_2$	$x_6$	$x_7$	RHS
$x_1$	0	1	1
$x_2$	1/2	-1	7/2
$x_4$	1/2	-2	9/2
$x_5$	-1/2	1	1/2
$x_3$	-1/2	-1	1/2
$p_j - z_j^{(1)}$	-1/2	-1	$f_1 = 13/2$
$q_j - z_j^{(2)}$	-1/2	0	$f_2 = 13/2$
$\mu_j$	7	1/2	
$\gamma_j$	-19/4	-13/2	
$\bar{c}^1$	-1/2	-2	
$\bar{c}^2$	1/2	1	

le processus de branchement est encore déclenché et nous avons les contraintes :

$$N_4 : x_2 \leq 3.$$

$$N_5 : x_2 \geq 4.$$

$N_4$  : la contrainte  $x_2 \leq 3$  est ajoutée à la Table 5.3 pour obtenir la Table 5.4 avec une solution entière optimale  $x^{*(4)} = (1, 3)$ ,  $f(x^{*(4)}) = 36$  mais  $x^{*(4)}$  n'est pas efficace. Donc, une solution efficace entière est générée par le test d'efficacité est  $\hat{x}^{(4)} = (2, 1)$  avec  $f(\hat{x}^{(4)}) = 30$ ,  $H_4 = \{6, 7\}$ .

TABLE 5.4 –

$B_4$	$x_7$	$x_8$	RHS
$x_1$	1	0	1
$x_2$	0	1	3
$x_4$	-1	1	4
$x_5$	0	-1	1
$x_3$	-2	-1	1
$x_6$	-2	-2	1
$p_j - z_j^{(1)}$	-2	-1	$f_1=6$
$q_j - z_j^{(2)}$	-1	-1	$f_2=6$
$\mu_j$	1	3	
$\gamma_j$	-16	-9	
$\bar{c}^1$	-3	-1	
$\bar{c}^2$	2	1	

La Table 5.2.5 est obtenu en ajoutant la coupe efficace  $x_7 + x_8 \geq 1$  à la Table 5.4. On obtient la solution optimale entière  $x^{*(6)} = (1, 2)$ , qui est efficace et  $f(x^{*(6)}) = 25 < f_{opt}$ , le nœud est sondé. Une solution entière efficace  $\hat{x}^{(6)} = (2, 0)$  est générée par le test d'efficacité avec  $f(\hat{x}^{(6)}) = 20$ .

$B_6$	$x_7$	$x_9$	RHS
$x_1$	1	0	1
$x_2$	-1	1	2
$x_4$	-2	1	3
$x_5$	1	-1	2
$x_3$	-1	-1	2
$x_6$	0	-2	3
$x_8$	1	-1	1
$p_j - z_j^{(1)}$	-1	-1	$f_1=5$
$q_j - z_j^{(2)}$	0	-1	$f_2=5$
$\mu_j$	1	2	
$\gamma_j$	-5	-8	
$\bar{c}^1$	-2	-1	
$\bar{c}^2$	1	1	

$N_5$  : la contrainte  $x_2 \geq 4$  est ajoutée à la Table 5.3, on obtient la Table 5.5 suivante indiquant que  $x^{*(5)} = (\frac{1}{2}, 4)$  est une solution optimale avec  $f(x^{*(5)}) = 39$  :

TABLE 5.5 –

$B_5$	$x_6$	$x_8$	RHS
$x_1$	1/2	1	1/2
$x_2$	0	-1	4
$x_4$	-1/2	-2	11/2
$x_5$	0	1	0
$x_3$	-1	-1	1
$x_7$	-1/2	-1	1/2
$p_j - z_j^{(1)}$	-1	-1	$f_1 = 6$
$q_j - z_j^{(2)}$	-1/2	0	$f_2 = 13/2$
$\mu_j$	1	0	
$\gamma_j$	-9	-13/2	
$\bar{c}^1$	-3/2	-2	
$\bar{c}^2$	1	1	

Le processus de branchement est appliqué.

$$N_7 : x_1 \leq 0.$$

$$N_8 : x_1 \geq 1.$$

$N_7$  : la contrainte  $x_1 \leq 0$  est ajoutée à la Table 5.5. On obtient une solution entière optimale  $x^{*(7)} = (0, 4)$ , avec  $f(x^{*(7)}) = 30$ . Comme cette solution est non efficace et  $f(x^{*(7)}) = f_{opt}$ , e nœud est sondé.

TABLE 5.6 –

$B_7$	$x_5$	$x_9$	RHS
$x_1$	0	1	0
$x_2$	1	0	4
$x_4$	1	-1	6
$x_6$	-2	-2	1
$x_3$	-1	-2	2
$x_7$	0	-1	1
$x_8$	1	0	0
$p_j - z_j^{(1)}$	-1	-2	$f_1=5$
$q_j - z_j^{(2)}$	-1	-1	$f_2=6$
$\mu_j$	0	0	
$\gamma_j$	-11	-17	
$\bar{c}^1$	-1	-3	
$\bar{c}^2$	1	2	

$N_8$  : la contrainte  $x_1 \geq 1$  est ajoutée à la Table 5.5, le nœud est sondé avec un dual non réalisable.

$N_3$  : la contrainte  $x_1 \geq 2$  ajoutée à la Table 5.1 permet d'obtenir la Table 5.7 avec une solution entière efficace  $x^{*(3)} = (2, 2)$  et  $f(x^{*(3)}) = 42 > f_{opt}$ . Une mise à jour est alors faite  $f_{opt} = 42$ ,  $x_{opt} = x^{*(3)} = (2, 2)$  et le nœud courant est donc sondé.

TABLE 5.7 – Une solution efficace entière du problème  $QP_E$

$B_3$	$x_3$	$x_7$	RHS
$x_1$	0	-1	2
$x_2$	1	2	2
$x_4$	1	3	2
$x_5$	-1	-2	2
$x_6$	-2	-2	1
$p_j - z_j^{(1)}$	-1	0	$f_1=7$
$q_j - z_j^{(2)}$	-1	-1	$f_2=6$
$\mu_j$	2	2/3	
$\gamma_j$	-11	-7	
$\bar{c}^1$	-1	1	
$\bar{c}^2$	1	0	

L'algorithme s'arrête, puisque tous les nœuds créés sont sondés. La solution optimale du problème ( $IQPE$ ) est  $x_{opt} = (2, 2)$  est la valeur optimale de l'objectif quadratique indéfini est  $f_{opt} = 42$ .

Pour cet exemple, l'ensemble de toutes les solutions efficaces  $\mathcal{X}_E$  du problème multiobjectif (5.1) est :  $\mathcal{X}_E = \{(2, 2), (2, 1), (2, 0), (1, 1), (1, 0), (0, 1), (0, 0)\}$ . Cependant, l'algorithme proposée maximise la fonction quadratique indéfinie  $f$  sur l'ensemble  $\mathcal{X}_E$  sans avoir à énumérer toutes ces solutions mais seulement la solution efficace  $x_{opt} = (2, 2)$ .

Pour résumer l'approche proposée à travers cet exemple, nous présentons l'arborescence suivante qui représente les états des nœud durant le processus.

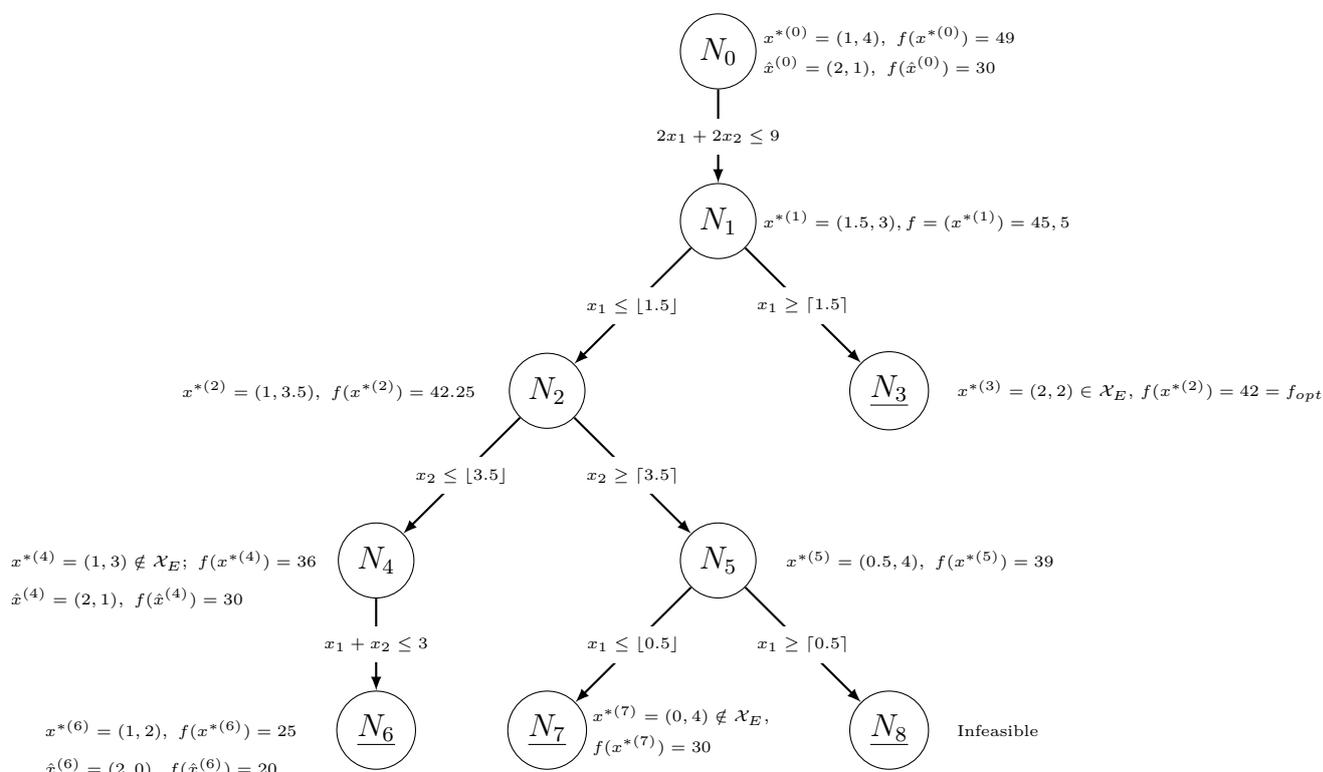


FIGURE 5.1 – Arborescence représentant les états des nœuds en utilisant l'algorithme Branch & Cut.

## 5.2.6 Expérimentation et résultats

La méthode décrite dans la section 5.2.4 a été mise en œuvre dans l'environnement MATLAB et testée sur des problèmes  $QP_E$  and  $MOILP$  générés aléatoirement.

Les données sont générées aléatoirement par une distribution uniforme discrète dans l'intervalle  $[-20, 20]$  pour les coefficients de la matrice des contraintes  $A$ ,  $[0, 20]$  pour

le coefficients du second membre  $b$  et pour les coefficients de la matrice des fonctions objectifs  $C$  du problème  $MOILP$ . Les vecteurs  $p$ ,  $q$  et les scalaires  $\alpha$ ,  $\beta$  sont générés de la même manière que  $C$  pour assurer la condition de positivité requise pour  $(p^T x + \alpha)$  et  $(q^T x + \beta)$ . Pour chaque instance  $(n, m, r)$  ( $n$  est le nombre de variables,  $m$  le nombre de contraintes et  $r$  le nombre de critères), une série de 10 problèmes a été résolue. Des expérimentations informatiques ont été effectuées sur un ordinateur 2,53 GHZ DELL, un processeur Intel Core (TM) i3 et 2 Go de mémoire. Les performances obtenues de l'algorithme sont résumées dans le tableau 5.8, où la moyenne, le nombre maximal de temps de calcul (en secondes) et le nombre d'itérations de la méthode du simplexe nécessaires sont donnés.

TABLE 5.8 – Résultats de l'expérimentation

n	m	r	CPU (secondes)		Itérations	
			Moyenne	Max	Moyenne	Max
10	5	5	0,746	1,611	333,25	718
10	10	5	1,149	3,950	468,625	2000
10	10	10	1,122	2,264	383,5	755
15	10	5	13,632	25,625	2042	4099
15	10	10	33,496	214,786	4899,375	27198
15	15	5	13,184	35,879	1843,625	4101
15	15	10	11,056	20,928	2110,25	4255
15	15	15	9,6621	18,6405	1945	3976
20	10	10	27,823	110,254	3639	12124
20	15	10	15,779	40,889	8729,875	17965
20	20	10	14,391	36,0528	7692,125	15980
20	20	15	42,536	109,345	11302,25	20687
20	20	20	13,066	30,554	7725,375	14869
30	30	30	81,449	170,634	38384,625	78183

### Discussion des résultats

Comme on peut le constater, l'algorithme proposé est efficace pour les petites et moyennes dimensions en termes de temps d'exécution (cpu (seconde)).

Pour les problèmes à grande échelle, le nombre de solutions efficaces peut être très élevé, de sorte qu'il devient irréaliste d'utiliser la méthode naïve qui consiste à énumérer explicitement toutes les solutions efficaces en nombres entiers du problème  $MOILP$  et de choisir celle qui maximise la fonction quadratique indéfinie  $f$  (voir Table 5.9). La

dernière colonne du tableau  $\rho$  indique le rapport entre le nombre de solutions efficaces répertoriées et le nombre de solutions efficaces du problème (*MOILP*).

TABLE 5.9 – Comparaison de notre méthode avec la méthode naive (*MOILP*) pour l'instance (15,10,5).

	CPU(s)		$\rho$
	Notre méthode	Méthode naive ( <i>MOILP</i> )	
	2,400	49,715	0,0046
	49,542	63,035	0,0044
	31,372	75,638	0,0030
	1,627	65,844	0,0052
	11,137	78,355	0,0050
	8,780	68,917	0,0026
	6,831	53,966	0,0029
	7,333	77,932	0,0067
	2,386	43,059	0,0044
	2,995	61,936	0,0021
Moyenne	12,340	63,8399	0,0041
Max	49,542	78,355	0,0068
Min	1,627	43,059	0,0021

Cependant, la méthode étant exacte, on s'attendait à ce que le problème devienne difficile en raison de :

- la non linéarité de la fonction de préférence,
- du nombre très grand des itérations de la méthode du simplexe,
- du nombre très grand de critères et
- de l'environnement discret du domaine de recherche.

Dans la section suivante, nous présentons une nouvelle méthode qui maximise une fonction fractionnaire linéaire sur l'ensemble efficient d'un problème *MOILP*. La méthode proposée est une adaptation de la méthode présentée précédemment avec l'ajout d'une nouvelle coupe, en plus de la coupe efficace [30], qui nous permettra d'éliminer des solutions qui ne sont pas optimaux pour notre problème.

## 5.3 Optimisation d'une fonction fractionnaire linéaire sur l'ensemble efficient d'un problème multiobjectif linéaire discrêt

### 5.3.1 Définitions et notations

Le problème de programmation multiobjectif linéaire discrêt (*MOILP*) est formulé comme suit :

$$(MOILP) \begin{cases} \max Z^i(x) = c^i x & i \in \{1, \dots, r\} \\ sc. \\ x \in \mathcal{D} = \mathcal{X} \cap \mathbb{Z}^n. \end{cases} \quad (5.9)$$

$\mathbb{Z}$  est l'ensemble des entiers,  $r \geq 2$ ,  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ ,  $C = (c^1, c^2, \dots, c^r)^T$ , où  $c^i \in \mathbb{R}^n$  est un vecteur ligne.  $\mathcal{X} = \{x \in \mathbb{R}^n | Ax \leq b, x \geq 0\}$ . Dans tout ce qui suit, on suppose que  $\mathcal{X}$  est un polyèdre convexe, borné, fermé et non vide.  $\mathcal{D}$  un ensemble non vide.

L'ensemble efficient du programme (*MOILP*) peut être de cardinalité très grande, il devient alors difficile pour le décideur (DM) de choisir la meilleure solution efficace qui correspond à ses préférences.

Une solution efficace du problème (*MOILP*) qui maximise une fonction d'utilité fractionnaire linéaire  $f$  du décideur, qui n'est pas nécessairement une combinaison des fonctions objectives du problème (*MOILP*), peut être obtenu en résolvant le programme suivant :

$$(ILFP)_E \begin{cases} \max f(x) = \frac{p^T x + \alpha}{q^T x + \beta} \\ sc. \\ x \in \mathcal{X}_E \subset \mathcal{D}. \end{cases} \quad (5.10)$$

où  $p, q$  sont des vecteur de  $\mathbb{R}^n$ ,  $\alpha, \beta \in \mathbb{R}$  et  $\mathcal{X}_E$  représente l'ensembles des solutions efficaces du problème (*MOILP*). On suppose que le facteur  $q^T x + \beta$  est positif pour tout  $x \in \mathcal{X}$ .

Comme application à notre problème, nous considérons le problème de flux de coûts minimal multiobjectif linéaire sur un réseau donné [31, 63, 77]. Soit  $G = (V, E)$  un graphe connexe orienté,  $V = \{v_1, v_2, \dots, v_m\}$  l'ensemble de ses sommets et  $E$  l'ensembles de ses arcs. Pour chaque sommet  $v_i$ ,  $i \in \{1, 2, \dots, m\}$  est attribué un réel  $b_i$  tel que  $\sum_{i=1}^m b_i = 0$ . De plus, chaque arc  $(i, j)$  a capacité  $u_{ij} \geq 0$  et  $r$  coûts unitaires de transport

$c_{ij}^k$ , ( $k = 1, 2, \dots, r$ ).

Le problème du flux de coûts minimum multiobjectif linéaire est formulé comme suit :

$$\left\{ \begin{array}{l} \min z^k = \sum_{(i,j) \in E} c_{ij}^k x_{ij} \quad k = 1, \dots, r \\ s.c. \\ \sum_{(i,j) \in E} x_{ij} - \sum_{(j,i) \in E} x_{ji} = b_i, \quad i = 1, \dots, m \\ 0 \leq x_{ij} \leq u_{ij} \end{array} \right. \quad (5.11)$$

Comme la rentabilité est un indice d'efficacité de l'opération exprimé en fraction, il peut être écrit comme suit :

$$\psi(x) = \frac{\sum_{(i,j) \in E} p_{ij} x_{ij} + \alpha}{\sum_{(i,j) \in E} q_{ij} x_{ij} + \beta}$$

où pour chaque arc  $(i, j) \in E$ ;  $p_{ij}$  représente le gain unitaire et  $q_{ij}$  le coût unitaire,  $\alpha$  représente un gain fixe et  $\beta$  un coût fixe. Le décideur peut considérer  $\psi$  comme fonction utilitaire à optimiser sur l'ensemble des efficace du problème (5.11).

La stratégie adoptée pour générer une solution optimale entière du problème principal  $(ILFP)_E$  est basée sur la résolution d'une séquence de problèmes fractionnaires linéaires continus  $(LFP)_l$ , à chaque étape  $l$ ,  $l \geq 0$ , définie par :

$$(LFP)_l \left\{ \begin{array}{l} \max f(x) = \frac{p^T x + \alpha}{q^T x + \beta} \\ s.c. \\ x \in \mathcal{X}_l. \end{array} \right. \quad (5.12)$$

$\mathcal{X}_0 = \mathcal{X}$  et  $\mathcal{X}_{l+1}$  est un sous ensemble de l'ensemble original  $\mathcal{X}$  à explorer dans l'étape  $l$ .

Au long de ce manuscrit, nous utiliserons les notations suivantes :

- On note  $x_{opt}$  la meilleure solution efficace de problème 5.10 trouvée jusqu'à l'étape  $l$  et  $f_{opt}$  la valeur du critère  $f$  correspondante.
- Soit  $x^{*(l)}$  la solution optimale du problème 5.12 correspondante à la base  $B_l$ , obtenue au nœud  $l$  de l'arborescence.
- Les sous ensembles  $I_l$  et  $N_l$  représentent l'ensemble des indices des variables de base et les variables hors base, respectivement, de  $x^{*(l)}$  tel que :

$$I_l = \{i | a_i \in B_l\},$$

$$N_l = \{i | a_i \notin B_l\},$$

où  $a_i$  est la  $i^{\text{ème}}$  colonne de la matrice  $A$ .

- Notons  $\bar{\gamma}_j$  la  $j^{\text{ème}}$  composante du vecteur de la direction de croissance  $\bar{\gamma}$  de la fonction principale  $f$  défini, à chaque étape  $l$ , par la relation suivante :

$$\bar{\gamma}_j = f^2(p_j - \bar{z}_j^1) - f^1(p_j - \bar{z}_j^2) \quad (5.13)$$

où :

- $\bar{z}_j^1 = p_{B_l} B_l^{-1} a_j$ ;  $\bar{z}_j^2 = q_{B_l} B_l^{-1} a_j$  et  $a_j$  est la  $j^{\text{ème}}$  colonne de la matrice  $A$ .
- $f^1 = p^T x^{*(l)} + \alpha$  et  $f^2 = q^T x^{*(l)} + \beta$ .
- $f = f^1 / f^2$ .
- Le bar signifie que les valeurs sont mises à jour.

Le théorème suivant [67], donne une condition nécessaire et suffisante d'optimalité d'une solution pour le problème  $(LFP)_l$  :

**Théorème 5.3.1** [67] *Une solution réalisable de base  $x^*$  est optimale pour le problème 5.12, si et seulement si le vecteur  $\bar{\gamma}$  est tel que  $\bar{\gamma}_j \geq 0$  pour tout  $j \in N_l$*

### 5.3.2 Les coupes utilisées

Soit  $x^{*(l)}$  une solution optimale entière du problème  $(LFP)_l$ , obtenue en utilisant éventuellement le processus de branchement Branch & Bound et soit  $\bar{c}_j^i$  la  $j^{\text{ème}}$  composante du vecteur gradient réduit  $\bar{c}^i = c^i - Z^i$ ,  $i \in \{1, 2, \dots, r\}$ , au niveau du dernier tableau de simplexe, de la fonction  $Z^i$  du problème  $(MOILP)$ .

Notre méthode, basée sur le principe de Branch & Cut, utilise ces informations pour construire une coupe efficace en mesure de supprimer des solutions entières qui ne sont pas efficaces pour le problème multiobjectif  $(MOILP)$ . Pour ce, nous définissons l'ensemble suivant en  $x^{*(l)}$  :

$$H_l = \{j \in N_l | \exists i \in \{1, \dots, r\}; \bar{c}_j^i > 0\} \cup \{j \in N_l | \bar{c}_j^i = 0, \forall i \in \{1, \dots, r\}\} \quad (5.14)$$

et la coupe efficace [30] est :

$$\sum_{j \in H_l} x_j \geq 1. \quad (5.15)$$

Une deuxième coupe est construite pour éliminer des solutions moins bonnes que celles déjà trouvées, elle est donné par l'inégalité suivante :

$$f(x) \geq f_{opt}. \quad (5.16)$$

On peut définir alors les sous-ensembles suivant au nœud  $l$  :

$$\mathcal{X}_{l+1}^1 = \{x \in \mathcal{X}_l \mid \sum_{j \in H_l} x_j \geq 1\},$$

$$\mathcal{X}_{l+1}^2 = \{x \in \mathcal{X}_l \mid f(x) \geq f_{opt}\}$$

et

$$\mathcal{X}_{l+1} = \mathcal{X}_{l+1}^1 \cup \mathcal{X}_{l+1}^2$$

### 5.3.3 Test d'efficacité

Étant donnée  $x^{*(l)}$  une solution optimale entière du problème  $(LFP)_l$ , On peut testé son efficacité en résolvant le programme linéaire mixte suivant ([43]) :

$$EK(x^{*(l)}) \left\{ \begin{array}{l} \max \theta = \sum_{i=1}^r \psi_i \\ st. \\ Cx - I\psi = Cx^{*(l)}. \\ x \in \mathcal{D}; \\ \psi_i \in \mathbb{R}^+, \forall i \in \{1, \dots, r\}, \end{array} \right. \quad (5.17)$$

avec  $C = (c^1, c^2, \dots, c^r)^T$  est une matrice d'ordre  $(r \times n)$ ,  $I$  est une matrice d'identité d'ordre  $r$  et  $\psi = (\psi_i)_{i \in \{1, \dots, r\}}$ .

Un point  $x^{*(l)}$  est efficace si et seulement si la valeur optimale de la fonction objectif  $\theta$  en  $EK(x^{*(l)})$  est nulle (voir [54]). Sinon, le programme  $EK(x^{*(l)})$  peut être utilisé pour générer une solution efficace dans le cas où  $x^{*(l)}$  ne l'ai pas.

### 5.3.4 Développement de la méthode

Dans cette section, nous développons un algorithme basé sur le principe de Branch and Cut pour générer une solution optimale entière du problème  $(ILFP)_E$  sans avoir à énumérer explicitement toutes les solutions entières efficaces du problème  $(MOILP)$ . Le processus de Branch and Bound, renforcés par des coupes et des tests d'efficacité, nous permettent de sonder des nœuds de l'arborescence, ainsi, un grand nombre de solutions réalisables et non efficaces peuvent être évitées.

### Description de la méthode

Chaque programme  $(LFP)_l$  correspond à un nœud dans une arborescence structurée. Un nœud  $l$  de l'arborescence est sondé si :

- la solution  $x^{*(l)}$  du problème 5.12 est une solution entière et efficace ( $x^{*(l)} \in \mathcal{X}$ ),
- $x^{*(l)}$  n'est pas efficace et l'ensemble  $H_l = \emptyset$ , ce qui veut dire qu'on ne pourra plus avoir de solutions efficaces dans le domaine courent,
- la valeur de  $f_{opt}$  obtenue est supérieure ou égal à la valeur de la fonction  $f$  en ce nœud,
- le programme 5.12 est infaisable.

Si la solution optimale  $x^{*(l)}$  du programme 5.12 est non entière, nous imposons alors des restrictions d'intégrité sur les variables du programme jusqu'à obtention d'une solution entière. Pour le faire, on suppose  $x_j^{*(l)}$  une composante fractionnaire de  $x^{*(l)}$ , le nœud  $l$  de l'arborescence est alors séparé en deux nœuds qui lui sont imposées par les contraintes additionnelles  $x_j \leq \lfloor x_j^{*(l)} \rfloor$  et  $x_j \geq \lceil x_j^{*(l)} \rceil$ , où  $x_j \leq \lfloor x_j^{*(l)} \rfloor$  indique la partie entière inférieure du nombre réel  $x_j^{*(l)}$  et  $x_j \geq \lceil x_j^{*(l)} \rceil$  sa partie entière supérieure.

En présence d'une solution entière, la solution est testée pour son efficacité en résolvant le programme 5.17, dans le but d'améliorer et de mettre à jour la valeur de la fonction objectif  $f_{opt}$ , si cette solution est meilleure que les solutions entières efficaces déjà trouvées.

Si la solution n'est pas efficace, la coupe 5.15 est alors rajoutée et aussi la coupe 5.16 si  $f_{opt}$  a été mis à jour, et le nouveau programme obtenu sera résolu.

La méthode se termine lorsque tous les nœuds créés sont sondés. La solution optimale du problème 5.10 est  $x_{opt}$  correspondante à la meilleure valeur  $f_{opt}$  trouvée.

### Formulation de l'algorithme

Les étapes de l'algorithme qui génère une solution optimale pour le problème 5.10 est décrit dans ce qui suit :

---

**Algorithme 2** : L'OPTIMISATION FRACTIONNAIRE LINÉAIRE SUR L'ENSEMBLE EFFICACE DISCRÊT

---

**Étape 1 (Initialisation)** : On initialise la valeur optimale de  $f$ ,  $f_{opt} = -\infty$ ; la solution  $x_{opt}$  inconnue;  $l = 0$  et  $\mathcal{X}_0 = \mathcal{X}$ .

**Étape 2 (Étape générale) :** Tant qu'il existe un nœud non encore sondé dans l'arborescence, choisir un nœud  $l$  non sondé et résoudre le programme fractionnaire linéaire correspondant  $(LFP)_l$  en utilisant la méthode dual du simplexe et une méthode de résolution des problème fractionnaire linéaire ([24, 67]).

- Si  $(LFP)_l$  n'est pas réalisable, alors le nœud correspondant est sondé ;
- Sinon, soit  $x^{*(l)}$  sa solution optimale obtenue ;
  1. Si  $f_{opt} \geq f(x^{*(l)})$ , alors le nœud  $l$  est sondé. Aller à l'**étape 2** ;
  2. Sinon, tester l'intégrité de la solution  $x^{*(l)}$ .
    - (a) Si  $x^{*(l)}$  est entière, aller à l'**étape 2a**.
    - (b) Sinon, aller à l'**étape 2b**.

**Étape 2a (Test d'efficacité) :** Résoudre le programme 5.17 au point  $x^{*(l)}$ .

- Si  $x^{*(l)}$  est efficace, le nœud  $l$  est sondé. Mettre à jour la valeur de  $f_{opt}$  si nécessaire, et aller à l'**étape 2**.
- Sinon, soit  $\bar{x}^{(l)}$  une solution optimale du programme  $EK(x^{*(l)})$ . Mettre à jour la valeur de  $f_{opt}$  si nécessaire, et aller à l'**étape 2c**.

**Étape 2b (Processus de branchement) :** Parmi les composantes de la solution  $x^{*(l)}$ , il y'a au moins une valeur non entière, choisir une d'entre elle  $x_j^{*(l)}$  et diviser l'ensemble  $\mathcal{X}_l$  en deux sous ensembles  $\mathcal{X}_{l_1}$  et  $\mathcal{X}_{l_2}$  en ajoutant les contraintes  $x_j \leq \lfloor x_j^{*(l)} \rfloor$  et la coupe 5.16, si c'est nécessaire, pour obtenir le programme  $(LFP_{l_1})$  et  $x_j \geq \lceil x_j^{*(l)} \rceil$  et la coupe 5.16, si c'est nécessaire, pour obtenir le programme  $(LFP_{l_2})$  tel que  $l_1 = l + 1$  et  $l_2 = l_1 + 1$ . Aller à l'**étape 2**.

**Étape 2c (Coupe efficace) :** Déterminer les ensembles  $N_l$  et  $H_l$  ;

1. Si  $H_l = \emptyset$  alors le nœud  $l$  est sondé. Aller à l'**étape 2** ;
2. Sinon, ajouter la coupe 5.15 au problème  $(LFP)_l$  et si nécessaire, ajouter la coupe 5.16 ;  $l = l + 1$ . Aller à l'**étape 2**.

---

## Résultats fondamentaux

Les résultats théoriques suivants justifient les différentes étapes de l'algorithme proposé.

**Théorème 5.3.2** *Supposons que  $H_l \neq \emptyset$  au point entier non efficace courant  $x^{*(l)}$ . Si  $x \neq x^{*(l)}$  est une solution entière efficace dans le domaine  $\mathcal{X}_l \setminus \{x^{*(l)}\}$ , alors  $x \in \mathcal{X}_{l+1}$ .*

**Preuve 5.3.1** *Soit  $x$  une solution entière efficace dans le domaine  $\mathcal{X}_l \setminus \{x^{*(l)}\}$  tel que  $x \notin \mathcal{X}_{l+1}$ , alors  $x \notin \mathcal{X}_{l+1}^1$  et  $x \notin \mathcal{X}_{l+1}^2$ .*

- Si  $x \notin \mathcal{X}_{l+1}^1$ , alors  $x \notin \{x \in \mathcal{X}_l / \sum_{j \in H_l} x_j \geq 1\}$  ce qui implique  $x \in \{x \in \mathcal{X}_l / \sum_{j \in N_l \setminus H_l} x_j \geq 1\}$ .

Par conséquent, on aura les inégalités suivantes :  $\sum_{j \in H_l} x_j < 1$  et  $\sum_{j \in N_l \setminus H_l} x_j \geq 1$ . Il s'ensuit que  $x_j = 0$  pour tout indice  $j \in H_l$ , et  $x_j \geq 1$  pour au moins un indice  $j \in N_l \setminus H_l$ .

En utilisant le tableau de simplexe en  $x^{*(l)}$ , l'égalité suivante est vérifiée pour tout critère  $i \in \{1, \dots, r\}$  :

$$\begin{aligned} Z^i(x) &= c^i x = c^i x^{*(l)} + \sum_{j \in N_l} \bar{c}_j^i x_j \\ &= c^i x^{*(l)} + \sum_{j \in H_l} \bar{c}_j^i x_j + \sum_{j \in N_l \setminus H_l} \bar{c}_j^i x_j \\ &= c^i x^{*(l)} + \sum_{j \in N_l \setminus H_l} \bar{c}_j^i x_j \end{aligned}$$

Donc,  $Z^i(x) \leq Z^i(x^{*(l)})$  pour tout critère  $i \in \{1, \dots, r\}$ , avec  $Z^i(x) < Z^i(x^{*(l)})$  pour au moins un critère puisque  $\bar{c}_j^i \leq 0$  pour tout  $j \in N_l \setminus H_l$ .

Par conséquent,  $Z^i(x)$  est dominé par  $Z^i(x^{*(l)})$  et  $x$  est non efficace.

- Si  $x \notin \mathcal{X}_{l+1}^2$ ,  $f(x) < f_{opt}$ . Ainsi,  $x$  n'est pas optimale.

**Corollaire 5.3.1** *La contrainte  $\sum_{j \in H_l} x_j \geq 1$  définit une coupe efficace.*

**Preuve 5.3.2** *Il est clair que  $\sum_{j \in H_l} x_j \geq 1$  est une contrainte valide efficace par le théorème theorem 5.3.2, puisque toute solution entière efficace dans le domaine courant  $\mathcal{X}_l$  vérifie cette contrainte. Par ailleurs, la solution entière courante  $x^{*(l)}$  ne satisfait pas cette contrainte puisque  $x_j = 0$  pour tout  $j \in H_l$ . Donc, on peut dire que la contrainte  $\sum_{j \in H_l} x_j \geq 1$  est une coupe efficace.*

**Proposition 5.3.1** *Si  $H_l = \emptyset$  à la solution entière courante  $x^{*(l)}$ , alors  $\mathcal{X}_l \setminus \{x^{*(l)}\}$  est un domaine exploré.*

**Preuve 5.3.3**  $H_l = \emptyset$  signifie que  $x^{*(l)}$  est une solution optimale entière pour chaque critère. Par conséquent  $x^{*(l)}$  est un point idéal dans le domaine  $\mathcal{X}_l$  et  $\mathcal{X}_l \setminus \{x^{*(l)}\}$  ne contient aucune solution efficace.

**Théorème 5.3.3** L'algorithme converge vers une solution optimale du programme  $(ILFP)_E$ , si une telle solution existe, en un nombre fini d'itérations.

**Preuve 5.3.4** Comme  $\mathcal{D}$ , l'ensemble des solutions réalisables entières du problème  $(MOILP)$ , est un ensemble fini et borné contenu dans  $\mathcal{X}$ , la cardinalité de l'ensemble efficient  $\mathcal{X}_E$  est aussi fini. À chaque fois qu'une solution optimale entière  $x^{*(l)}$  est atteinte, la coupe efficace est ajoutée si  $x^{*(l)} \notin \mathcal{X}_E$ . D'autre part, si  $H_l = \emptyset$ , la solution correspondante  $x^{*(l)}$  constitue un point idéal local et le nœud actuel peut être sondé car aucun critère ne peut être amélioré et aucune amélioration de la valeur  $f_{opt}$  ne sera possible. Ainsi, selon le théorème et le corollaire ci-dessus, au moins la solution  $x^{*(l)}$  est éliminée lorsqu'on étudie un sous-problème  $(LFP)_k$ ,  $k > l$ , mais aucune solution efficace entière n'est omise.

### 5.3.5 Exemple illustratif

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

$$(ILFP)_E \left\{ \begin{array}{l} \max \quad f(x) = \frac{-x_1 - x_2 + 2x_3 + 3x_4 - 15}{x_2 + x_3 + x_4 + 1} \\ s.c \\ x \in \mathcal{X}_E \end{array} \right.$$

où  $\mathcal{X}_E$  représente l'ensemble efficace entier du problème  $(MOILP)$  définie par :

$$(MOILP) \left\{ \begin{array}{l} \max \quad Z^1 = x_1 + x_2 - x_3 - 2x_4 \\ \max \quad Z^2 = x_1 - 3x_2 - x_3 + x_4 \\ \max \quad Z^3 = 2x_2 + 2x_3 + 2x_4 \\ \max \quad Z^4 = -x_2 + 2x_3 + 2x_4 \\ s.t. \\ 3x_1 - 2x_2 + x_3 + x_4 \leq 3 \\ x_2 + 3x_3 + 4x_4 \leq 4 \\ x_1 + x_2 + x_3 + x_4 \leq 5 \\ 2x_2 + x_3 \leq 6 \\ x_i \geq 0, \text{ entiers } \forall i \in \{1, \dots, 4\} \end{array} \right.$$

**Initialisation** : Soit  $f_{opt} = -\infty$ ,  $l = 0$ .

Le programme  $(LFP)_0$  est résolu. La première solution optimale obtenue est  $x^{*(0)} = (0, \frac{14}{5}, \frac{2}{5}, 0)$ . La solution n'est pas entière et  $f(x^{*(0)}) = -\frac{85}{21}$ . Les résultats de la résolution sont résumés dans le tableau 5.3.5 :

$B_1$	$x_1$	$x_4$	$x_6$	$x_8$	RHS
$x_7$	1	1/5	-1/5	-2/5	9/5
$x_3$	0	8/5	2/5	-1/5	2/5
$x_2$	0	-4/5	-1/5	3/5	14/5
$x_5$	3	-11/5	-4/5	7/5	41/5
$p_j - \bar{z}_j^1$	-1	-1	-1	1	$f_1 = -17$
$q_j - \bar{z}_j^2$	0	1/5	-1/5	-2/5	$f_2 = 21/5$
$\bar{\gamma}_j$	-21/5	-4/5	-38/5	-13/5	
$\bar{c}^1$	1	2/5	3/5	-4/5	
$\bar{c}^2$	1	1/5	-1/5	8/5	
$\bar{c}^3$	0	2/5	-2/5	-4/5	
$\bar{c}^4$	0	-2	-1	1	

TABLE 5.10 – La première solution optimale

Comme cette solution n'est pas entière, le processus de branchement est déclenché et deux nœuds sont créés,  $N_1$  et  $N_2$ , qui correspondent aux contraintes suivantes :

$$N_1 : x_2 \leq \lfloor \frac{14}{5} \rfloor.$$

$$N_2 : x_2 \geq \lceil \frac{14}{5} \rceil.$$

$N_1$  : la contrainte  $x_2 \leq \lfloor \frac{14}{5} \rfloor$  est ajoutée au tableau 5.3.5 pour obtenir le tableau 5.3.5 avec une solution optimale  $x^{*(1)} = (0, 2, \frac{2}{3}, 0)$  et  $f(x^{*(1)}) = -\frac{47}{11}$ .

$B_2$	$x_1$	$x_4$	$x_6$	$x_9$	RHS
$x_7$	1	-1/3	-1/3	-2/3	7/3
$x_3$	0	4/3	1/3	-1/3	2/3
$x_2$	0	0	0	1	2
$x_5$	3	-1/3	-1/3	7/3	19/3
$x_8$	0	-4/3	-1/3	-5/3	4/3
$p_j - \bar{z}_j^1$	-1	1/3	-2/3	5/3	$f_1 = -47/3$
$q_j - \bar{z}_j^2$	0	-1/3	-1/3	-5/3	$f_2 = 11/3$
$\bar{\gamma}_j$	-11/3	-4	-23/3	-13/3	
$\bar{c}^1$	1	-2/3	1/3	-4/3	
$\bar{c}^2$	1	7/3	1/3	8/3	
$\bar{c}^3$	0	-2/3	-2/3	-4/3	
$\bar{c}^4$	0	-2/3	-2/3	5/3	

TABLE 5.11 –

$N_2$  : la contrainte  $x_2 \geq \lceil \frac{14}{5} \rceil$  est ajoutée au tableau 5.3.5 pour avoir le tableau 5.3.5 avec une solution optimale  $x^{*(2)} = (0, 3, 0, \frac{1}{4})$  et  $f(x^{*(2)}) = -\frac{69}{17}$ .

$B_3$	$x_1$	$x_6$	$x_8$	$x_9$	RHS
$x_7$	1	-1/4	-1/4	1/4	7/4
$x_3$	0	0	1	2	0
$x_2$	0	0	0	-1	3
$x_5$	3	-1/4	-1/4	-11/4	35/4
$x_4$	0	1/4	-3/4	-5/4	1/4
$p_j - \bar{z}_j^1$	-1	-3/4	1/4	-5/4	$f_1 = -69/4$
$q_j - \bar{z}_j^2$	0	-1/4	-1/4	1/4	$f_2 = 17/4$
$\bar{\gamma}_j$	-17/4	-15/2	-13/4	-1	
$\bar{c}^1$	1	1/2	-1/2	1/2	
$\bar{c}^2$	1	-1/4	7/4	1/4	
$\bar{c}^3$	0	-1/2	-1/2	1/2	
$\bar{c}^4$	0	-1/2	-1/2	-5/2	

TABLE 5.12 –

Comme  $f(x^{*(2)}) > f(x^{*(1)})$ , nous explorons le nœud  $N_2$  en premier. La solution  $x^{*(2)}$

n'est pas entière, le processus de branchement est encore déclenché avec les contraintes suivantes :

$$N_3 : x_4 \leq \lfloor \frac{1}{4} \rfloor.$$

$$N_4 : x_4 \geq \lceil \frac{1}{4} \rceil.$$

$N_3$  : la contrainte  $x_4 \leq \lfloor \frac{1}{4} \rfloor$  est ajoutée au tableau 5.3.5 et on obtient le tableau 5.3.5 avec une solution optimale  $x^{*(3)} = (0, 3, 0, 0)$  et  $f(x^{*(3)}) = -\frac{9}{2}$ .

$B_4$	$x_1$	$x_8$	$x_9$	$x_{10}$	RHS
$x_7$	1	-1	-1	-1	2
$x_3$	0	1	2	0	0
$x_2$	0	0	-1	0	3
$x_5$	3	-1	-4	-1	9
$x_4$	0	0	0	1	0
$x_6$	0	-3	-5	-4	1
$p_j - \bar{z}_j^1$	-1	-2	-5	-3	$f_1 = -18$
$q_j - \bar{z}_j^2$	0	-1	-1	-1	$f_2 = 4$
$\bar{\gamma}_j$	-4	-26	-38	-30	
$\bar{c}^1$	1	1	3	2	
$\bar{c}^2$	1	1	-1	-1	
$\bar{c}^3$	0	-2	-2	-2	
$\bar{c}^4$	0	-2	-5	-2	

TABLE 5.13 –

$N_4$  : la contrainte  $x_4 \geq \lceil \frac{1}{4} \rceil$  est ajoutée au Tableau 5.3.5 et le programme obtenu devient irréalisable, le nœud est alors sondé.

Comme la solution optimale  $x^{*(3)} = (0, 3, 0, 0)$  est entière, On teste son efficacité. La solution n'est pas efficace. Une solution entière efficace est générée par le test d'efficacité est  $\hat{x}^{(3)} = (2, 3, 0, 0)$  avec  $f(\hat{x}^{(3)}) = -5$ . Mettre à jour  $f_{opt} = -5$ ,  $x_{opt} = (2, 3, 0, 0)$ .  $H_3 = \{1, 8, 9, 10\}$ .

Appliquer la coupe efficace  $x_1 + x_8 + x_9 + x_{10} \geq 1$  et la coupe  $f(x) \geq -5$  pour obtenir la solution optimale  $x^{*(5)} = (1, 3, 0, 0)$ ,  $f(x^{*(5)}) = -\frac{19}{4}$ . Les résultats sont donnés dans le tableau suivant 5.3.5 :

$B_5$	$x_8$	$x_9$	$x_{10}$	$x_{11}$	RHS
$x_7$	-2	-2	-2	1	1
$x_3$	1	2	0	0	0
$x_2$	0	-1	0	0	3
$x_5$	-4	-7	-4	3	6
$x_4$	0	0	1	0	0
$x_6$	-3	-5	-4	0	1
$x_1$	1	1	1	-1	1
$x_{12}$	6	9	7	1	1
$p_j - \bar{z}_j^1$	-1	-4	-2	-1	$f_1 = -19$
$q_j - \bar{z}_j^2$	-1	-1	-1	0	$f_2 = 4$
$\bar{\gamma}_j$	-23	-35	-27	-4	
$\bar{c}^1$	0	2	1	1	
$\bar{c}^2$	0	-2	-2	1	
$\bar{c}^3$	-2	-2	-2	0	
$\bar{c}^4$	-2	-5	-2	0	

TABLE 5.14 –

La solution  $x^{*(5)}$  est entière mais non efficace. Une solution efficace entière est générée par le test d'efficacité est  $\hat{x}^{(5)} = (2, 3, 0, 0)$  avec  $f(\hat{x}^{(5)}) = -5 = f_{opt}$  et  $H_5 = \{9, 10, 11\}$ .

Appliquer la coupe  $x_9 + x_{10} + x_{11} \geq 1$  et utiliser le dual du simplexe pour obtenir la solution entière efficace  $x^{*(6)} = (2, 3, 0, 0)$ ,  $f(x^{*(6)}) = -5$  donnée par le tableau 5.3.5. Le nœud courant est sondé.

$B_6$	$x_8$	$x_9$	$x_{10}$	$x_{13}$	RHS
$x_7$	-2	-3	-3	1	0
$x_3$	1	2	0	0	0
$x_2$	0	-1	0	0	3
$x_5$	-4	-10	-7	3	3
$x_4$	0	0	1	0	0
$x_6$	-3	-5	-4	0	1
$x_1$	1	2	2	-1	2
$x_{12}$	6	8	6	1	0
$x_{11}$	0	1	1	-1	1
$x_{14}$	6	8	6	1	0
$p_j - \bar{z}_j^1$	-1	-3	-1	-1	$f_1 = -20$
$q_j - \bar{z}_j^2$	-1	-1	-1	0	$f_2 = 4$
$\bar{\gamma}_j$	-24	-32	-24	-4	
$\bar{c}^1$	0	1	0	1	
$\bar{c}^2$	0	-3	-3	1	
$\bar{c}^3$	-2	-2	-2	0	
$\bar{c}^4$	-2	-5	-2	0	

TABLE 5.15 –

$N_1$  : La solution  $x^{*(1)} = (0, 2, \frac{2}{3}, 0)$  est non entière, le processus de branchement génère deux nœuds  $N_7, N_8$  :

$$N_7 : x_3 \leq \lfloor \frac{2}{3} \rfloor, f(x) \geq -5.$$

$$N_8 : x_3 \geq \lceil \frac{2}{3} \rceil, f(x) \geq -5 .$$

$N_7$  : La contrainte  $x_3 \leq \lfloor \frac{2}{3} \rfloor$  est ajoutée au tableau 5.3.5, on obtient le tableau 5.3.5 avec une solution optimale  $x^{*(7)} = (0, 2, 0, \frac{1}{2})$  et  $f(x^{*(7)}) = -\frac{31}{7}$ .

$B_7$	$x_1$	$x_6$	$x_9$	$x_{10}$	RHS
$x_7$	1	-1/4	-3/4	-1/4	5/2
$x_3$	0	0	0	1	0
$x_2$	0	0	1	0	2
$x_5$	3	-1/4	9/4	-1/4	13/2
$x_8$	0	0	-2	-1	2
$x_4$	0	1/4	-1/4	-3/4	1/2
$x_{11}$	1	2	2	1	2
$p_j - \bar{z}_j^1$	-1	-3/4	7/4	1/4	$f_1 = -31/2$
$q_j - \bar{z}_j^2$	0	-1/4	-3/4	-1/4	$f_2 = 7/2$
$\bar{\gamma}_j$	-7/2	-13/2	-11/2	-3	
$\bar{c}^1$	1	1/2	-3/2	-1/2	
$\bar{c}^2$	1	-1/4	13/4	7/4	
$\bar{c}^3$	0	-1/2	-3/2	-1/2	
$\bar{c}^4$	0	-1/2	-3/2	-1/2	

TABLE 5.16 –

$N_8$  : La contrainte  $x_3 \geq \lceil \frac{2}{3} \rceil$  est ajoutée à la table 5.3.5 et on obtient après résolution le tableau 5.3.5 avec une solution optimale entière  $x^{*(8)} = (0, 1, 1, 0)$  et  $f(x^{*(8)}) = -\frac{14}{3}$ .

$B_8$	$x_1$	$x_6$	$x_9$	$x_{10}$	RHS
$x_7$	1	-3	-1	-2	3
$x_3$	0	0	0	-1	1
$x_2$	0	4	1	3	1
$x_5$	3	9	2	7	4
$x_8$	0	-8	-2	-5	3
$x_9$	0	-4	-1	-3	1
$x_{11}$	1	8	4	3	1
$p_j - \bar{z}_j^1$	-1	-7	1	5	$f_1 = -14$
$q_j - \bar{z}_j^2$	0	-3	-1	-2	$f_2 = 3$
$\bar{\gamma}_j$	-3	-21	-11	-13	
$\bar{c}^1$	1	-6	-1	-4	
$\bar{c}^2$	1	13	3	8	
$\bar{c}^3$	0	-6	-2	-4	
$\bar{c}^4$	0	6	1	5	

TABLE 5.17 –

Comme  $f(x^{*(7)}) > f(x^{*(8)})$ , on explore le nœud  $N_7$ . La solution  $x^{*(7)}$  est non entière, le processus de branchement est déclenché :

$$N_9 : x_4 \leq \lfloor \frac{1}{2} \rfloor.$$

$$N_{10} : x_4 \geq \lceil \frac{1}{2} \rceil.$$

$N_9$  : La contrainte  $x_4 \leq \lfloor \frac{1}{2} \rfloor$  ajoutée à la table 5.3.5 rend le problème irréalisable, le nœud est alors sondé.

$N_{10}$  : La contrainte  $x_4 \geq \lceil \frac{1}{2} \rceil$  ajoutée à la table 5.3.5 rend le problème irréalisable aussi, le nœud est sondé.

$N_8$  : La solution  $x^{*(8)} = (0, 1, 1, 0)$  est entière mais pas efficace, la solution efficace générée par le test d'efficacité est  $\hat{x}^{(8)} = (1, 1, 1, 0)$  avec  $f(\hat{x}^{(8)}) = -5 = f_{opt}$ . Une autre solution optimale du problème ( $LFP_E$ ) est donc  $x_{opt} = (1, 1, 1, 0)$  et  $f_{opt} = -5$ .  $H_8 = \{1, 4, 6, 10\}$ .

Appliquer la coupe efficace  $x_1 + x_4 + x_6 + x_{10} \geq 1$  pour obtenir une solution entière optimale  $x^{*(11)} = (1, 1, 1, 0)$ ,  $f(x^{*(11)}) = -5$  (table 5.3.5) qui est efficace, le nœud est alors sondé.

$B_9$	$x_4$	$x_6$	$x_{10}$	$x_{12}$	RHS
$x_7$	-4	-2	-3	1	2
$x_3$	0	0	-1	0	1
$x_2$	4	1	3	0	1
$x_5$	6	-1	4	3	1
$x_8$	-8	-2	-5	0	3
$x_9$	-4	-1	-3	0	1
$x_{11}$	7	3	4	1	0
$x_1$	1	1	1	-1	1
$x_{13}$	7	3	4	1	0
$p_j - \bar{z}_j^1$	8	2	6	-1	$f_1 = -15$
$q_j - \bar{z}_j^2$	-3	-1	-2	0	$f_2 = 3$
$\bar{\gamma}_j$	-21	-9	-12	-3	
$\bar{c}^1$	-7	-2	-5	1	
$\bar{c}^2$	12	2	7	1	
$\bar{c}^3$	-6	-2	-4	0	
$\bar{c}^4$	6	1	5	0	

TABLE 5.18 –

L'algorithme s'arrête puisque tous les nœuds créés sont sondés et les solutions optimales du problème  $(ILFP)_E$  sont donc  $x_{opt_1} = (2, 3, 0, 0)$  et  $x_{opt_2} = (1, 1, 1, 0)$  avec une valeur optimale de la fonction fractionnaire linéaire  $f_{opt} = -5$ .

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

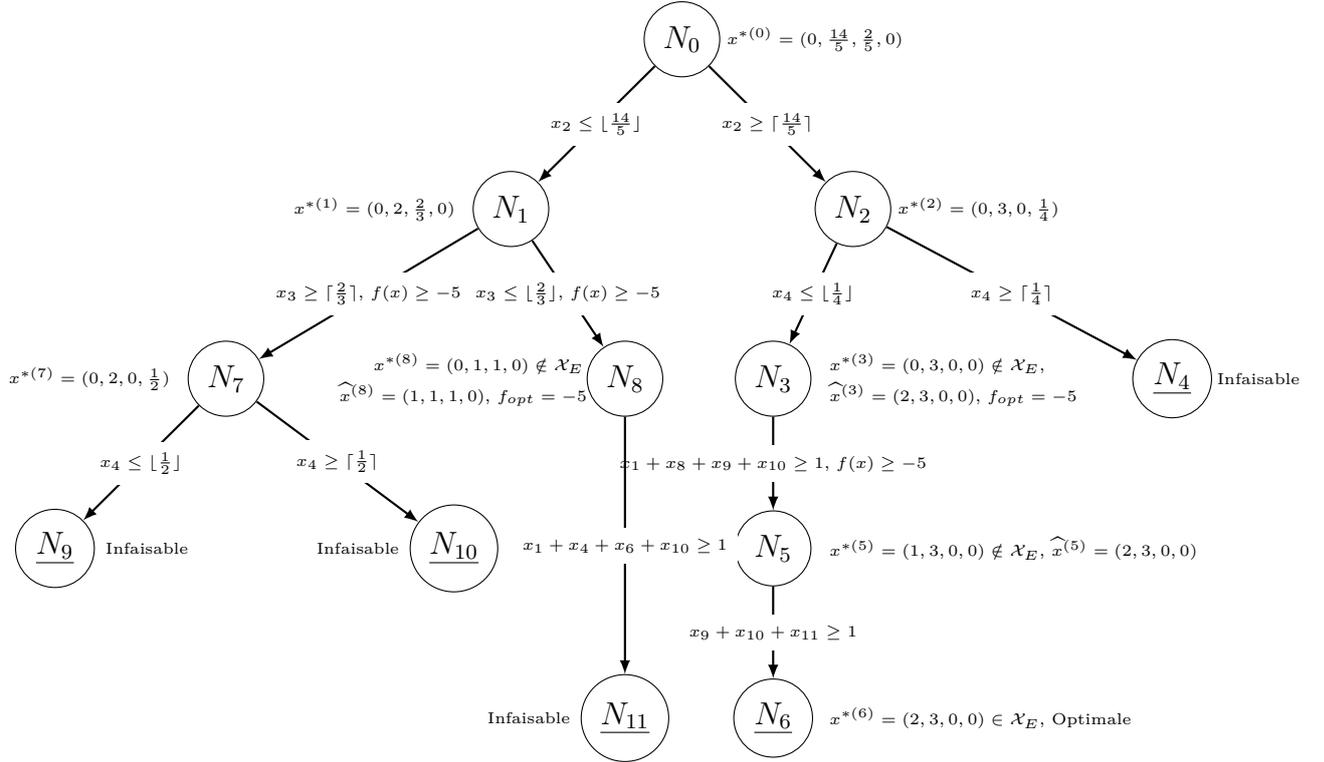


FIGURE 5.2 – Arborescence représentant les états des nœuds en utilisant l’algorithme Branch & Cut.

### 5.3.6 Expérimentation et résultats

La méthode décrite dans la section 5.3.4 a été implémentée dans l’environnement MATLAB et générée aléatoirement pour les problèmes  $(ILFP)_E$  et  $(MOILP)$ . Les données sont générées de manière aléatoire à partir d’une distribution uniforme discrète dans l’intervalle  $[1, 30]$  pour les coefficients des contraintes  $A$ ,  $[50, 100]$  pour le second membre  $b$  et  $[-10, 10]$  pour les coefficients de la matrice  $C$  des fonctions objectives du problème  $(MOILP)$ . Les vecteurs  $p$  et le scalaire  $\alpha$  sont générés de la même manière que la matrice  $C$ , néanmoins le scalaire  $\beta$  et le vecteur  $q$  sont générés tels que  $(q^t x + \beta) > 0$  pour toute solution possible  $x \in \mathcal{X}$ . Pour chaque instance  $(n, m, r)$  ( $n$  est le nombre de variables,  $m$  le nombre de contraintes et  $r$  le nombre d’objectifs), une série de 10 problèmes a été résolue.

Les expériences de calcul ont été effectuées sur un MacBook Pro, une mémoire Intel Core i7 de 2,7 GHz et une mémoire de 4 Go.

La performance obtenue de l’algorithme est résumée dans le tableau 5.3.6 où le temps

moyen CPU (en secondes), le nombre moyen de nœuds requis, le nombre moyen de coupes efficaces sont rapportés. De plus, le minimum et le maximum de chaque mesure sont indiqués entre parenthèses. La dernière colonne  $\rho$  du tableau indique le rapport entre le nombre moyen de solutions efficaces répertoriées et le nombre moyen de toutes solutions efficaces du problème (*MOILP*).

TABLE 5.19 – Résultats de l'expérimentation

r	$n \times m$	CPU (secondes)	$N^{bre}$ de Nœux	$N^{bre}$ de coupes efficaces	$\rho$
3	$25 \times 20$	8.105 [2.574; 19.853]	571.4 [91; 1167]	57.2 [8; 126]	0.190 [0.035; 1]
	$25 \times 25$	6.266 [1.611; 19.222]	334.9 [5; 1195]	29.5 [2; 84]	0.077 [0; 0.181]
	$30 \times 20$	5.865[1.8; 15.276]	382.8 [4; 847]	36.8 [1; 102]	0.113 [0.022; 0.666]
	$30 \times 25$	8.320 [3.821; 17.585]	630[165; 1295]	48.8 [5; 90]	0,091 [0, 0312; 0, 187]
	$35 \times 20$	12.898 [3.213; 23.368]	835.1 [147; 1705]	59.3 [4; 109]	0.0594 [0; 0.1]
	$35 \times 25$	8.902 [4.235; 16.101]	712.1 [315; 1369]	47.7 [14; 106]	0.046 [0.013; 0.08]
	$40 \times 25$	16,989 [1, 847; 45, 343]	1038,7 [1; 2921]	63,4 [0; 189]	0,066 [0, 028; 0, 173]
5	$25 \times 20$	2.689 [1.406; 4.213]	116.3 [1; 359]	10.9 [0; 35]	0.030 [0.005; 0.125]
	$25 \times 25$	2.673 [1.536; 5.279]	93.5 [17; 254]	6.2 [0; 21]	0.022 [0.006; 0.060]
	$30 \times 20$	3.744 [1.581; 10.822]	263.1 [3; 862]	10.5 [0; 30]	0.015 [0.003; 0.056]
	$30 \times 25$	4.852[1.505; 15.010]	187.204 [3; 586]	21.4[0; 83]	0.017 [0.004; 0.053]
	$35 \times 20$	8.756 [4.301; 18.541]	1943.3 [278; 14556]	45.9 [13; 176]	0.020 [0.008; 0.032]
	$35 \times 25$	3.450 [1.616; 6.336]	118.2 [4; 362]	9.2 [0; 49]	0.006 [0; 0.013]
	$40 \times 25$	13,912 [1, 718; 67, 265]	727,2 [5; 3224]	47,4 [0; 227]	0,011 [0, 004; 0, 04]
7	$25 \times 20$	2.848 [1.478; 8.137]	75.9 [3; 293]	5.7 [0; 30]	0.009 [0.002; 0.017]
	$25 \times 25$	1.926 [1.525; 3.175]	45.4 [3; 179]	2.2 [0; 10]	0.008 [0; 0.029]
	$30 \times 20$	3.320 [1.443; 9.298]	174.9 [1; 613]	10.5 [0; 52]	0.009 [0.001; 0.031]
	$30 \times 25$	2,522 [1, 505; 5, 636]	91,8 [3; 400]	3,4[0; 17]	0,007 [0, 001; 0, 014]
	$35 \times 20$	4,804 [1, 607; 14, 229]	285,5 [7; 680]	17,1 [0; 69]	0,006 [0, 001; 0, 015]
	$35 \times 25$	3,567 [1, 664; 10, 267]	166,9 [3; 674]	6,3 [0; 31]	0,006 [0, 001; 0, 022]
	$40 \times 25$	4,777[1, 815; 11, 892]	209,3 [9; 667]	13,3 [0; 53]	0,005 [0, 001; 0, 014]

### Discussion des résultats

Notons que pour un grand nombre d'objectifs, nous appliquons moins de coupes efficaces en conséquence ; nous avons moins de nœuds.

On peut également voir que les coupes efficaces éliminent un grand nombre de solu-

tions efficaces car la moyenne du rapport entre le nombre de solutions efficaces répertoriées sur le nombre de toutes les solutions efficaces du problème (*MOILP*) ne dépasse pas 0,9% pour 8 objectifs, 3% pour 5 objectifs et 19% pour 3 objectifs (voir tableau 5.3.6).

Les coupes 5.15 et 5.16 nous évitent de visiter un grand nombre de solutions efficaces du problème (*MOILP*), ainsi la solution optimale de notre problème principal (*ILFP*)<sub>E</sub> est atteinte en un temps très réduit.

Dans le tableau 5.3.6, nous présentons une comparaison avec la méthode de Mahdi & Chaabane [64]. Nous remarquons que dans les problèmes à grande taille, notre méthode donne des résultats nettement meilleurs du point de vue temps de calcul.

## 5.4 Conclusion

Dans ce chapitre, nous avons développé un nouvel algorithme pour résoudre le problème de l'optimisation d'une fonction fractionnaire linéaire sur l'ensemble efficace d'un problème de programmation linéaire multiobjectif en évitant le passage systématique par toutes les solutions efficaces. En effet, le processus de Branch & Bound renforcé par des coupes et des tests nous ont permis de saturer un nombre considérable de nœuds dans l'arborescence. Ainsi, plusieurs solutions réalisables entières non efficace ou efficaces non optimales ont été évitées.

r	$n \times m$	CPU(secondes)	
		Méthode Mahdi & Chaabane	Notre méthode
3	$5 \times 5$	0,21 [0, 01; 0, 16]	0.51 [0, 42; 1, 15]
	$10 \times 5$	0,44 [0, 02; 1, 6]	0,55 [0, 31; 0, 86]
	$15 \times 5$	0,7 [0, 08; 1, 72]	1,16[0, 35; 2, 57]
	$20 \times 5$	1,23[0, 2; 5, 23]	1,48 [0, 79; 3, 22]
	$20 \times 10$	1,45 [0, 24; 5, 12]	1,82 [0, 38; 4, 23]
	$30 \times 10$	8,98 [0, 87; 19, 31]	8,54 [1, 57; 31, 43]
	$35 \times 15$	7,45 [1, 78; 17, 34]	5,58 [1, 84; 22, 88]
	$40 \times 15$	10,82 [0, 22; 38, 27]	3,64 [1.50; 6, 81]
	$50 \times 15$	45,48 [2, 02; 210]	19,89 [1.52; 118, 10]
	$60 \times 20$	78,37 [5, 98; 272]	10,97 [1.68; 43, 96]
	$70 \times 20$	76,82 [19, 6; 286]	35,70[3, 69; 104, 92]
	$80 \times 20$	89 [16, 14; 350, 45]	55,41 [2, 33; 195, 54]
	$20 \times 70$	24,65 [0, 44; 134, 29]	4,65 [1, 58; 20, 24]
	$20 \times 80$	23,59 [0, 46; 105, 61]	4,67 [1, 57; 11, 04]
5	$5 \times 5$	0,31 [0, 03; 3, 18]	1,91 [1, 37; 5, 45]
	$10 \times 5$	1,24 [0, 12; 4, 16]	1,51[1, 37; 1, 63]
	$15 \times 5$	3,17 [0, 18; 5, 36]	2,68 [1, 43; 11, 01]
	$20 \times 5$	3,92 [0, 2; 6, 23]	4,46 [1, 43; 12, 34]
	$20 \times 10$	4,14 [1, 24; 5, 82]	1,87 [1, 43; 2, 51]
	$30 \times 10$	7,38[1, 17; 21, 31]	2,63[1, 62; 4, 31]
	$40 \times 15$	10,45 [3, 78; 17, 34]	1,94[1, 39; 2, 52]
	$50 \times 15$	23,2 [4, 22; 58, 27]	2,32 [1, 51; 5, 06]
	$60 \times 20$	89,17 [5, 45; 232]	5,68 [1, 64; 22, 47]
	$70 \times 20$	95, 25[20, 6; 256]	24,74 [1, 78; 160, 09]
	$80 \times 20$	98,43 [17, 14; 350, 45]	17,59 [1, 83; 87, 9]
	$20 \times 70$	20,85 [0, 41; 145, 77]	2,47 [1, 65; 5, 16]
	$20 \times 80$	20,91 [0, 57; 150, 12]	2,44 [1, 82; 3, 27]
	8	$5 \times 5$	0,57[0, 13; 11, 54]
$10 \times 5$		1,25 [0, 11; 25]	1,44 [1, 29; 1, 69]
$10 \times 5$		1,24 [0, 12; 4, 16]	1,51[1, 37; 1, 63]
$15 \times 5$		5,56 [1, 18; 45, 32]	2,18 [1, 4; 5, 5]
$20 \times 5$		5,02[0, 2; 5, 23]	1,76 [1, 4; 2, 84]
$20 \times 10$		10,5 [1, 35; 81]	1,63 [1, 48; 1, 97]
$30 \times 10$		14,5[3, 18; 89, 2]	2,27 [1, 53; 4, 31]
$40 \times 15$		20,69 [2, 57; 93, 27]	1,92 [1, 49; 2, 51]
$50 \times 15$		37,32 [5, 18; 143, 21]	2,17 [1, 54; 3, 62]
$60 \times 20$		71,25 [7, 33; 228, 04]	2,03[1, 68; 3]
$70 \times 20$		96,82 [19, 64; 286]	2,16[1, 53; 3, 44]
$80 \times 20$		101,02 [16, 14; 364, 15]	7,37 [1, 349; 31, 49]
$20 \times 70$		13,48 [0, 53; 101, 13]	3,21 [1, 57; 9, 64]
$20 \times 80$		10,03 [0, 63; 59, 75]	2,41 [1, 67; 4, 10]

TABLE 5.20 – Comparaison avec la méthode Mahdi &amp; Chaabane [64]

# Conclusion générale

Dans ce présent travail, nous avons passé en revue dans les premiers chapitres, les concepts fondamentaux de la programmation non linéaire et la programmation multiobjectif en nombres entiers qui constitue le sujet central de cette thèse. C'est ainsi que nous nous sommes focalisé sur la mise en oeuvre de trois méthodes exactes.

Nous avons déployées, dans une première étape, une méthode pour résoudre les problèmes de la programmation multiobjectif quadratique indéfinie en nombres entiers *MOIQP*. Une expérience est établie mettant en exergue les résultats obtenus sur des instances générées aléatoirement et un exemple illustratif est développé montrant le déroulement de l'algorithme. Une étude comparative est rapportée pour évaluer les performances de notre méthode avec celle proposée par Arora [9]. Notre algorithme a trouvé plus de solutions efficaces que l'algorithme d'Arora.

Nous avons aussi abordés le problème de l'optimisation d'un critère sur l'ensemble efficient d'un problème multiobjectif qui s'inscrit dans l'optique de l'optimisation globale, elle concerne la recherche de la meilleure solution parmi les solutions efficaces d'un problème d'optimisation multiobjectif linéaire en nombres entiers *MOILP*. Nous avons pu maximiser une fonction quadratique indéfinie sur l'ensemble efficient d'un problème *MOILP* (le cas (Quadratique indéfinie-Linéaire)), Ceci a été possible grâce au principe Branch & Bound renforcé par une coupe efficace et des tests permettant ainsi de réduire l'espace de recherche. De la même manière, et en ajoutant une deuxième coupe, nous avons réussi à mettre au point une deuxième méthode qui permet de maximiser une fonction fractionnaire linéaire sur l'ensemble efficient d'un problème *MOILP* (le cas (Fractionnaire-Linéaire)). L'étude expérimentale a montré que notre méthode pour le cas (Fractionnaire-Linéaire) est meilleure que celle proposée par Mahdi et Chaabane [64] en considérant les mêmes conditions de programmation. Dans le cas (Quadratique indéfinie-Linéaire), une comparaison avec la méthode naïve a été faite en l'absence de telle méthode dans la littérature.

Parmi les travaux qui peuvent présenter des perspectives et qu'on souhaite aborder pour l'avenir on y trouve :

- La généralisation de la première méthode proposée pour la résolution d'un problème d'optimisation multiobjectif quadratique semi-définie positive en nombre entiers *MOIQP*.
- La généralisation de la méthode proposée pour la résolution du problème de l'optimisation d'un critère sur l'ensemble efficient d'un *MOILP* au cas de l'optimisation d'un critère quadratique semi-défini sur l'ensemble efficient d'un problème multiobjectif.
- L'utilisation de l'optimisation parallèle pour la résolution des problèmes traités dans cette thèse à grande échelle.

# Bibliographie

- [1] Abbas M, Moulai M. (1997) Penalties Method for Integer Linear Fractional Programs, *Belgian Journal of Operations Research, Statistics and computer sciences*, Vol. 37/4, 41-51.
- [2] Abbas M, Moulai M. (1999) An algorithm for mixed integer linear fractional programming problem, *Belgian Journal of Operations Research, Statistics and computer sciences*, Vol. 39/1, 21-30.
- [3] Abbas M, Moulai M. (1999) Solving multiple objective integer linear programming, *Journal of the Italian Operations Research Society (Ricerca Operativa)*29,pp. 15-38, 1999.
- [4] Abbas M, Chaabane D. (2002) An algorithm for solving multiple objective integer linear programming problem, *RAIRO Operations Research* 36,pp. 351-364, 2002.
- [5] Abbas M, Chaabane D. (2006) Optimizing a linear function over an integer efficient set, *European Journal of Operational Research*, 174, No2; 1140-1161.
- [6] Abbaci L, Moulai M. (2010) L'optimisation sur l'ensemble efficient d'un problème stochastique multi-objectif discret, *thèse de Magister en Recherche Opérationnelle*, USTHB, Algérie.
- [7] Agrawal S. C, M. Chand M. (1981) A note on integer solutions to linear fractional interval programming problems by Branch and Bound technique, *Naval Res. Logist. Quart.*, 28, No4, pp. 671-677.
- [8] Anand P. (1972) Decomposition principle for indefinite quadratic programme, *Trabajos de Estadística de Investigación Operativa* 23, 61-71, 1972.
- [9] Arora R, Arora S.R. (2015) A cutting plane approach for multi-objective integer indefinite quadratic programming problem. *Opsearch*,52/2 , pp.367-381.

- 
- [10] Axehill D. (2008) Applications of integer quadratic programming in control and communication [Dissertation No. 1158]. *Linköping (Sweden)*, Linköping University.
- [11] Bector C.R, Dahl M. (1974) Simplex type finite iteration technique and duality for a special type of pseudo-concave quadratic program, *Cahiers du Centre d'Etudes de Recherche Operationnelle 16*, 207-222, 1974.
- [12] Bector C.R. (1972) Indefinite quadratic fractional functional programming. *Metrika*; 18(1) :21-30.
- [13] Bector C.R., Chandra S., Singh C. (1990) Duality in Multiobjective Fractional Programming. *Lecture Notes in Economics and Mathematical Systems*, vol 345. Springer, Berlin, Heidelberg
- [14] Belkeziz K, Metrane A. (2004) 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.
- [15] Benson H.P. (1978) Existence of Efficient Solutions for Vector Maximization Problems, *Journal of Optimization Theory and Applications 26*, No4, 569-580.
- [16] Benson H.P. (1984) Optimization over the efficient set. *J Math Ana App*, 98 :562-580.
- [17] Benson H.P. (1991) An All-Linear Programming Relaxation Algorithm for Optimizing over the Efficient Set, *Journal of Global Optimization*, Vol.1, 83-104.
- [18] Benson H.P. (1992) A finite Nonadjacent Extreme point Search Algorithm over the efficient Set, *Journal of Optimization Theory and Applications*, Vol.73, 47-64.
- [19] Benson H.P. (1993) A bisection-extreme point search algorithm for optimizing over the efficient set in the linear dependence case, *Journal of Global Optimization 3*, 95-111.
- [20] Benson H.P., Sayin S. (1994) Optimizing over the Efficient Set : Four Special Cases, *Journal of optimization Theory and Applications*, Vol.80, No1 ; 3-17.
- [21] Bereanu B. (1965) Decision regions and minimum risk solutions in linear programming, in : A. Prekopa, Ed., *Colloquium on applications of mathematics to economics*, Budapest, 1963, Publ. house of the Hungarian academy of sciences, Budapest, 37-42.

- 
- [22] Bitran G.R., Novaes G. (1976) Linear Programming with a fractional objective function, *Operations Research*, 24, 675-699.
- [23] Cambini A., Martein L. (1986) A modified version of Martos's algorithm for the linear fractional problem, *Methods of Oper. Res.* 53, pp. 33-44.
- [24] Cambini A., Martein L. (1992) Equivalence in Linear Fractional Programming, *Optimization* 23, 41-51
- [25] Chaabane D., Pirlot M. (2010) A method for optimizing over the integer efficient set, *J. Ind. Man. Opt.*, 6/4 : 811–823. DOI :10.3934/jimo.2010.6.811.
- [26] Chandra S., Chandramohan M. (1980) A Branch and Bound Method for Integer Nonlinear Fractional Programs, *Z. A. M. M.*, 60, pp. 735-737.
- [27] Charnes A., Cooper W.W. (1962) Programming with linear fractional functionals, *Naval Res. Logist. Quart.* 9, 181-186.
- [28] Collette Y., Siarry P. (2002) Optimisation multiobjectif, *Editions Eyrolles*, 2002.
- [29] Choo E.U. (1984) Proper efficiency and the linear fractional vector maximum problem, *Operational Research* 32/1, pp. 216-220.
- [30] Chergui MEA., Moulaï M.(2008) An exact method for a discrete multiobjective linear fractional optimization. *J Appl Math Deci sci*, vol.2008 : Article ID 760191, 12 pages. <http://dx.doi.org/10.1155/2008/760191>
- [31] Chinchuluun A., Pardalos P. M. (2007) A survey of recent developments in multiobjective optimization. *Annals of Operations Research*, 154(1), 29-50. DOI : 10.1007/s10479-007-0186-0
- [32] Cottle R.W. (1975) Three remarks about two papers on quadratic forms, *Zeitschrift für Operations Research* 19, 123-124, 1975.
- [33] Crema A., Sylva J. (2003) A method for finding the set of nondominated vectors for multiple objective integer linear programs, *European Journal of Operational Research*, in press.
- [34] Dantzig G.B. (1963) Linear programming and extension, *Princeton University Press*.

- 
- [35] Dantzig G.B. (1951) 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.
- [36] Datta N., Bhatia D.(1984) Algorithm to determine an initial efficient basic solution for a linear fractional multiple objective transportation problem. *Cahiers Centre Etudes Rech. Oper*, Kaiser-slauternol, 26/ 1-2 : 127-136.
- [37] Dinkelbach W. (1967) On nonlinear fractional programming, *Management Science*, 13 :492-498.
- [38] Dhaenens-Flipo C. (2005) Optimisation Combinatoire Mult-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.
- [39] Dorfman R. (1951) Application of linear programming to the theory of the firm, *University of California Press*.
- [40] Drici W., Moulaï M. (2019) : An exact method for solving multi-objective integer indefinite quadratic programs, *Optimization Methods and Software*, <https://doi.org/10.1080/10556788.2018.1560443>.
- [41] Drici W. Ouail, F.Z. & Moulaï M. (2018) : Optimizing a linear fractional function over the integer efficient set, *Ann Oper Re*, 267 : 135. <https://doi.org/10.1007/s10479-017-2691-0>
- [42] Ecker J.G., Song J.H. (1994) Optimizing a linear function over an efficient set, *Journal of Optimization Theory and Applications* Vol.83, No3; 541-563 .
- [43] Ecker J, Kouada I. (1975) Finding efficient points for linear multiple objective programs. *Math Prog*, 8 :375-377.
- [44] Ehrgott M, Hamacher HW., Klamroth K., Nickel S., Schobel A., Wiecek M.M. (1997) A note on the equivalence of balance points and Pareto solutions in multiple-objective programming. *Journal of Optimization Theory and Applications*, 92/1 :209–212.
- [45] Ehrgott M. (2000) Multicriteria Optimization, *Lecture Notes in Economics and Mathematical Systems*, No 491, Springer Verlag, Berlin, Heidelberg.
- [46] Erenguc SS, Benson H. (1991) An alirthm dor indefinite integer quadratic programming. *Comput Math Applic*,21,6/7 : 99 – –106.

- 
- [47] Geoffrion A.M. (1968) Proper Efficiency and the Theory of Vector Maximization, *Journal of Mathematical Analysis and Applications* 22, No-3, 618-630, 1968.
- [48] Gupta R., Malhotra R.(1992) Multi-criteria integer linear programming problem, *Cahiers de CERO* 34, (1992).
- [49] Granot D., Granot F. (1977) On integer and mixed integer fractional programming problems, *Ann. Discrete Math*, 1, 221-231.
- [50] Grunspan M. (1971) Fractional programming : A survey, *Technical Report 50*, Department of Industrial and Systems Engineering, University of Florida.
- [51] Hadley M. (1962) Linear Programming. Reading Mass : Addison Wesley.
- [52] Hasan M.B. (2012) A technique for solving special type quadratic programming problems, *Dhaka university j. sci.*, Volume 60(2), 209-215.
- [53] Henderson J.M., Quandt R.E. (1971) Microeconomic theory. *New York (NY) : McGraw-Hill*.
- [54] Isermann H. (1974) Proper efficiency and the linear vector maximization problem. *Op Res.* ; 22 :189–191.
- [55] Isermann H., Steuer R.E. (1987) Computational Experience Concerning Payoff Tables and Minimum Criterion Values over the Efficient Set. *European Journal of Operational Research*, Vol.33, 91-97.
- [56] Jorge J.M. (2009) An algorithm for optimizing a linear function over an integer efficient set. *Eur J Oper Res.*, 195 :98–103.
- [57] Karmarkar N.K. (1984) A new polynomial-time algorithm for linear programming, *Combinatorica* 373-395.
- [58] Kein D, Hannan E. (1992) An algorithm for multiple objective integer linear programming problem, *European Journal of Operational Research* 9, pp. 378-385.
- [59] Koopmans T.C. (1971) Analysis and Production as an Efficient Combination of Activities, *Activity Analysis of Production and Allocation*, Yale University Press, New Haven, London, 33-97 (originally published in 1951).
- [60] Kuhn H.W, Tucker A.W. (1951) Nonlinear Programming, *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, Berkeley, Los Angeles, 481-492.

- 
- [61] Li J.L., Wang P., Jian J. (2015) A New Algorithm Based on Lagrangian Relaxation for Solving Indefinite Quadratic Integer Programming, *J. COMPUT. ANAL. APPL.*, 19/6 pp. 1038–1048.
- [62] Luenberger D.G, Ye Y. (2008) Linear and Nonlinear Programming, Third Edition, *Springer*.
- [63] Lokman B., Köksalan M., Korhonen P.J., Wallenius J. (2016) An interactive algorithm to find the most preferred solution of multi-objective integer programs. *Annals of operations research*, 245(1-2), 67-95.
- [64] Mahdi S., Chaabane D.(2015) A linear fractional optimization over an integer efficient set. *RAIRO-Oper. Res.*, 49 : 265-278. DOI : 10.1051/ro/2014036.
- [65] Markowitz H. (1959) Portfolio selection : Efficient diversification of investment, *John Wiley and Sons*, New York.
- [66] Martos B. (1964) Hyperbolic programming, *Naval Res. Logist. Quart.*, 11, pp. 135-155.
- [67] Martos B. (1975) Nonlinear programming, *Theory and Methods*. North-Holland, Amsterdam.
- [68] Miettinen K. (1984) On the Methodology of Multiobjective Optimization Problems, *Journal of Optimization Theory and Applications* 42, No4, 499-524.
- [69] Miettinen K. (1999 ) Nonlinear Multiobjective Optimization, *Kluwer Academic Publishers*, Boston.
- [70] Miettinen K. (2001) Some Methods for Nonlinear Multi-objective Optimization, *In : Zitzler E., Thiele L., Deb K., Coello Coello C.A., Corne D. (eds) Evolutionary Multi-Criterion Optimization. EMO 2001*. Lecture Notes in Computer Science, vol 1993. Springer, Berlin, Heidelberg
- [71] Moulaï M., Drici W. (2018) An indefinite quadratic optimization over an integer efficient set, *Optimization*, 67/8, pp. 1143-1156. Available at <https://doi.org/10.1080/02331934.2018.1456539>.
- [72] Nagih A., Plateau G. (1999) Problèmes fractionnaires : Tour d’horizon sur les applications et méthodes de résolution, *RAIRO Oper. Res.* 33, 383-419.

- 
- [73] Nemhauser G.L., Wolsey L.A. (1988) Integer and combinatorial optimization, *Wiley*, Chichester.
- [74] Nguyen N.C. (1992) An Algorithm for Optimizing a Linear Function over the Integer Efficient Set, *Konrad-Zuse-zentrum fur Informationstechnik Berlin*.
- [75] Ozlen M., Azizoglu M., Burton BA. (2013) Optimising a nonlinear utility function in multi-objective integer programming. *J Global Optim.* ;56(1) :93-102.
- [76] Philip J. (1972) Algorithms for the vector maximization problem, *Mathematical programming* 2,207-229.
- [77] Padberg M. (2005) Classical cuts for mixed-integer programming and branch-and-cut. *Annals of Operations Research*, 139(1), 321-352.
- [78] Sakarovitch M. (1984) Optimisation combinatoire, Graphes et programmation linéaire, Hermann.
- [79] Sayin S. (2000) Optimizing over the efficient set using a topdown search of faces, *Operations Research* 48, 65-72.
- [80] Schaible S.(1976) Duality in fractional programming : a unified approach, *Operations Research*, 24, No-3 :452-461.
- [81] Schaible S.(1981) Fractional programming : applications and algorithms, *European Journal of Operational Research* 7, 111-120.
- [82] Schaible S.(1981) Fractional programming, *Zeitschrift für, Operations Research*, 27 :39-54.
- [83] Schaible S.(1981) Fractional programming, in R. Horst and P.M. Pardalos (eds.), *Handbook of Global Optimization*, Kluwer Academic Publishers, 495- 608.
- [84] Sharma S.D. (1994) Non-linear and dynamic Programming, *Kedar Nath Ram Nath & CO*, Meerut, India.
- [85] Schrijver A. (1986) Theory of Linear and Integer Programming, *John Wiley and Sons*, New-York.
- [86] Shigeno M, Takahashi I, Yamamoto Y. (2003) Minimum maximal flow problem An optimization over the efficient set. *Journal of Global Optimization*, 22 : 425-443.
- [87] Simonard M. (1972) Programmation linéaire, 2<sup>ème</sup> édition, tomes 1 et 2, Dunod.

- 
- [88] Stancu-Minasian I.M. (1997) Fractional Programming : Theory, Methods and Applications, *Kluwer Academic Publishers*, 45, 1-4 :343-367.
- [89] Stancu-Minasian I.M. (1999) A fifth bibliography of fractional programming, *Optimization* 45, 1-4 :343-367.
- [90] Stancu-Minasian I.M. (2006) A sixth bibliography of fractional programming, *Optimization* 55, 4, 405–428.
- [91] Steuer R.E. (1985) Multiple Criteria Optimization : Theory, Computation and Applications, *John Wiley and Sons*, New-York.
- [92] Sun X.L., Li J.L., Luo H.Z. (2010) Convex Relaxation and Lagrangian Decomposition for Indefinite Quadratic Integer Programming, *Optimization*, 59 ,pp. 627–641.
- [93] Swarup K. (1966) Quadratic programming. *Cahier du Centre d'Etudes et de Recherche Opérationnelle*, 8(4) : 223 – 234.
- [94] Teghem J. (1998) Programmation linéaire, *Ellipses*.
- [95] Teghem J., Kunsh P.L. (1986) A Survey Of Techniques For Finding Efficient Solutions To Multi-Objective Integer Linear Programming, *Asia-Pacific Journal of Operational Research* 3 95-108.
- [96] Wendell R.E, Lee D.N. (1977) Efficiency in Multiple Objective Optimization Problems, *Mathematical Programming* 12, No-3, 406-414.
- [97] Yamamoto Y. (2002) Optimization over the efficient set : Overview. *J Glo Opt.*, 22/1 – 4 : 285 – –317.
- [98] Zellner A. (1961) Linear regression with inequality constraints on the coefficients : An application of quadratic programming and linear decision rules. *Econ Inst Netherlands School of Economics*, 6109.
- [99] Zerdani O, Moulaï M. (2011) Optimization over an integer efficient set of a multiple objective linear fractional problem. *App Mat Sci.*, 5/50 : 2451 – –2466.
- [100] Zerdani O. (2013) L'Optimisation non Linéaire Multiobjectif, *thèse de doctorat en mathématiques*, option Recherche Opérationnelle, UMMTO, Algérie.