

Quelques fonctionnalités de  $\text{\LaTeX}$

N° d'ordre : 10/2016-M/MT

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

Faculté des Mathématiques



MEMOIRE

Présenté pour l'obtention du **diplôme** de **MAGISTER**

**En** : MATHEMATIQUES

**Spécialité** : Recherche Opérationnelle

**Par** : ZERFA Lamia

**Sujet**

**Programmation linéaire discrète à objectifs multiples de coefficients réels**

Soutenu publiquement le : 15/03/2016. Devant le jury composé de :

|                            |            |           |                      |
|----------------------------|------------|-----------|----------------------|
| M ABBAS Moncef             | Professeur | U.S.T.H.B | Président            |
| M CHERGUI Mohamed El-Amine | MCA        | U.S.T.H.B | Directeur de mémoire |
| M MOULAÏ Mustapha          | Professeur | U.S.T.H.B | Examinateur          |
| M MEZGHICHE Abdelhak       | MAA        | U.S.T.H.B | Invité               |

# Remerciements

C'est avec un grand plaisir qu'on réserve cette page en signe de gratitude et de profonde reconnaissance à tous ceux qui ont contribué de près ou de loin à la réalisation de ce modeste travail.

Nous remercions DIEU le tous puissant.

Je tiens à exprimer toute ma reconnaissance à mon Directeur de mémoire Monsieur CHERGUI Mohamed El-Amine. Je le remercie pour m'avoir proposé le sujet et encadré, pour ses orientations et ses conseils.

Mes vifs remerciements sont également adressés à Monsieur ABBAS Moncef pour avoir eu l'amabilité de bien vouloir présider le jury.

De même que j'adresse mes sincères remerciements à Monsieur MOULAÏ Mustapha et à Monsieur MEZGHICHE Abdelhak pour l'intérêt qu'ils portent à mon travail et aussi au fait d'avoir accepté de faire partie du jury.

*Je dédie ce travail*

*A ma très chère mère et adorable tante, pour leur amour inestimable, leur confiance, leur soutien, leurs sacrifices et toutes les valeurs qu'ils ont su m'inculquer.*

*A mon cher mari pour son soutien et son précieux encouragement.*

*A ma soeur Ines et son époux Madani.*

*A mon frère Mouhamed.*

*A mes oncles Belkacem et Sadek.*

*A Ayoub, Younes et Houda.*

*A ma chère amie Meriem.*

*A tous ceux qui me sont chers et que j'ai omis de citer.*

# Table des matières

|   |           |
|---|-----------|
| <b>Introduction</b>   | <b>1</b>  |
| <b>1 Sur la programmation linéaire mono-objectif</b>                            | <b>3</b>  |
| 1.1 Résultats fondamentaux de la programmation linéaire mono-objectif . . . . . | 3         |
| 1.2 Programmation mono-objectif discrète . . . . .                              | 8         |
| 1.2.1 Méthode "Branch and Bound" . . . . .                                      | 9         |
| 1.2.2 Coupes de Gomory [39] . . . . .   | 10        |
| 1.3 La programmation linéaire fractionnaire mono-objectif . . . . .             | 12        |
| 1.3.1 Notations et définitions . . . . .  | 12        |
| 1.3.2 Relation entre $PL$ et $PLF$ . . . . .                                    | 13        |
| 1.3.3 Résolution d'un programme linéaire fractionnaire . . . . .                | 15        |
| <b>2 Sur la programmation multi-objectif</b>                                    | <b>19</b> |
| 2.1 Sur la programmation multi-objectif discrète linéaire . . . . .             | 20        |
| 2.1.1 Concepts de base . . . . .  | 20        |
| 2.1.2 Caractérisation d'une solution efficace . . . . .                         | 22        |
| 2.1.3 Solutions supportées et non supportées . . . . .                          | 24        |
| 2.1.4 Exemple illustratif . . . . .   | 26        |
| 2.1.5 Approches de résolution . . . . .   | 27        |
| 2.1.6 Quelques méthodes de résolution . . . . .                                 | 28        |
| 2.2 Programmation multi-objectif discrète fractionnaire . . . . .               | 34        |
| 2.2.1 Méthode de Abbas and Moulai [2] . . . . .                                 | 35        |
| 2.2.2 Méthode de Chergui et Moulai [9] . . . . .                                | 37        |
| 2.2.3 Méthode de Ait Mehdi, Chergui et Abbas [3] . . . . .                      | 38        |

---

|  |               |
|--|---------------|
| <b>3 Contribution algorithmique en optimisation multi-objectif discrète</b>  | <b>41</b>     |
| 3.1 Génération des solutions non dominées pour le problème multi-objectif linéaire discret . . . . .               | 41            |
| 3.1.1 Définitions et notations . . . . .   | 41            |
| 3.1.2 Principe de la méthode . . . . .   | 42            |
| 3.2 Génération des solutions non dominées pour le problème multi-objectif linéaire fractionnaire discret . . . . . | 44            |
| 3.2.1 Formulation du problème . . . . .  | 44            |
| 3.2.2 Développement de la méthode . . . . .  | 45            |
| 3.3 Résultats théoriques . . . . .   | 47            |
| 3.4 Exemples illustratifs . . . . .  | 51            |
| <br><b>Conclusion et perspectives</b>  | <br><b>61</b> |
| <br><b>Bibliographie</b>   | <br><b>62</b> |

# Table des figures

|     |   |    |
|-----|---|----|
| 2.1 | Illustration du problème de Benson 2.1.4. . . . .   | 23 |
| 2.2 | Schéma des différentes solutions et points caractéristiques de l'espace des objectifs . . . . . | 25 |
| 2.3 | Espace des décisions. . . . .   | 26 |
| 2.4 | Espace des critères. . . . .  | 26 |
| 2.5 | Illustration de la méthode $\epsilon$ -contrainte. . . . .                                      | 29 |
| 2.6 | Illustration des différentes étapes de la méthode deux phases. . . . .                          | 32 |

# Introduction

Le thème général de cette thèse est l'optimisation combinatoire. Cela consiste à trouver parmi un ensemble discret de solutions respectant des contraintes une solution qui optimise une fonction objectif. Par optimiser, on entend trouver la plus petite valeur (problème de minimisation) ou la plus grande valeur (problème de maximisation) de la fonction.

De nombreux problèmes réels peuvent se modéliser comme des problèmes d'optimisation combinatoire : planification, contrôle de la production, distribution dans des réseaux, industrie manufacturière, énergie (pétrole, gaz, électricité, nucléaire), transports (aériens, routiers et ferroviaires), télécommunications, industrie forestière et finance.

Dans ce document, on s'intéresse particulièrement aux problèmes d'optimisation combinatoires qui comportent plusieurs objectifs. On parle alors d'optimisation combinatoire multi-objectif. Ce domaine possède ses sources dans les travaux de Edgeworth [16] et de Pareto [52] dans le cadre d'études d'économie au 19<sup>ème</sup> siècle. Cependant, l'optimisation multi-objectif connaît un intérêt croissant depuis le milieu des années 1980 [40] et le domaine connaît une expansion importante depuis le milieu des années 1990 avec l'apparition de méthodes évolutionnaires pour l'optimisation multi-objectif [6]. Actuellement, l'optimisation multi-objectif est appliquée dans de nombreux domaines académiques et industriels.

De manière formelle, un problème d'optimisation multi-objectif est un problème de la forme  $\max f(x) = (f_1(x), f_2(x), \dots, f_k(x))$  tel que  $x$  soit une solution réalisable. La solution d'un problème multi-objectif n'est pas une unique solution mais un ensemble de solutions appelé ensemble non-dominé. Les composantes du vecteur  $f$  sont les différentes fonctions à optimiser. Les solutions non dominées sont celles pour lesquelles l'amélioration d'un des objectifs entraîne systématiquement la détérioration de la qualité d'au moins un autre



objectif. La présence d'objectifs contradictoires apporte de nouveaux défis à relever pour les méthodes d'optimisation classique.

Le premier chapitre est dédié à la programmation mono-objectif. Dans un premier temps, des principaux résultats concernant la programmation linéaire (en variables continues et en variables discrètes) sont expliquées ainsi que les méthodes de résolution classiques (méthodes du simplexe, coupes de Gomory et "branch and bound"). Ensuite, l'essentiel des définitions et résultats liés à l'optimisation fractionnaire mono-objectif en nombres entiers est présenté et les trois grandes stratégies de résolution d'un programme fractionnaire.

Le chapitre deux pose le contexte du travail : l'optimisation multi-objectif linéaire discrète (*MOILP*) et multi-objectif fractionnaire linéaire discrète (*MOILFP*). Différentes notions, concepts et méthodes récentes dédiées à la programmation multi-objectif aussi bien linéaire discrète et que fractionnaire linéaire discrète sont développées.

Une méthode exacte  $Z\epsilon$  est proposée dans le chapitre trois, pour la détermination de toutes les solutions non dominées du problème (*MOILP*) dans le cas où les coefficients des variables dans les vecteurs  $f_i(x)$ ,  $i = 1, \dots, k$ , ne sont pas restreints aux valeurs entières. Une généralisation de la méthode  $Z\epsilon$  est aussi conçue pour le problème (*MOILP*).

# Chapitre 1

## Sur la programmation linéaire mono-objectif

Ce chapitre a pour objectif principal de présenter le contexte de l'optimisation linéaire. Nous rappelons en premier lieu des éléments essentiels de la théorie de la programmation linéaire et linéaire factionnaire, en variables aussi bien continues qu'entières. Nous introduisons ensuite, les notions fondamentales ainsi que les principaux résultats liés à la théorie de la programmation mono-objectif linéaire et factionnaire en nombres entiers.

### 1.1 Résultats fondamentaux de la programmation linéaire mono-objectif

La programmation linéaire PL peut se définir comme une technique mathématique permettant de résoudre des problèmes de gestion et particulièrement ceux où le gestionnaire doit déterminer, face à différentes possibilités, l'utilisation optimale des ressources de l'entreprise pour atteindre un objectif spécifique comme la maximisation des bénéfices ou la minimisation des coûts. Dans la plupart des cas, les problèmes de l'entreprise pouvant être traités par la programmation linéaire comportent un certain nombre de ressources. On peut mentionner, par exemple, la main-d'œuvre, les matières premières, les capitaux, ... qui sont disponibles en quantité limitée et qu'on veut répartir d'une façon optimale entre un certain nombre de processus de fabrication.

La programmation linéaire est un domaine central de l'optimisation, car les problèmes

de PL sont les problèmes d'optimisation les plus faciles, la fonction objectif et les contraintes sont toutes linéaires. Beaucoup de problèmes réels de recherche opérationnelle peuvent être exprimés comme un problème de PL. Pour cette raison un grand nombre d'algorithmes pour la résolution d'autres problèmes d'optimisation sont fondés sur la résolution de problèmes linéaires.

**Définition 1.1.** On appelle problème d'optimisation linéaire sous forme générale tout programme linéaire de la forme :

$$\left\{ \begin{array}{l} \text{maximiser } F(x) \\ G_1(x) \leq 0 \\ G_2(x) \leq 0 \\ \vdots \\ G_m(x) \leq 0 \\ x \in \mathbb{R}^n. \end{array} \right. \quad (1.1.1)$$

où  $n, m \in \mathbb{N}^*$ ,  $F : \mathbb{R}^n \rightarrow \mathbb{R}$  est une fonction linéaire et  $G_1(x), G_2(x), \dots, G_m(x)$  sont des applications affines définies sur  $\mathbb{R}^n$  et à valeurs réelles .

**Définition 1.2.** On dit que  $\bar{x} \in \mathbb{R}^n$  est une solution réalisable du problème (1.1.1) si  $x$  satisfait aux contraintes, autrement dit si  $G_1(\bar{x}) \leq 0, G_2(\bar{x}) \leq 0, \dots, G_m(\bar{x}) \leq 0$ .

L'ensemble  $S$  de toutes les solutions réalisables d'un problème d'optimisation est appelé son ensemble réalisable.

**Définition 1.3.** On dit que  $x^* \in \mathbb{R}^n$  est une solution optimale du problème (1.1.1) si  $x^*$  est une solution réalisable du problème (1.1.1), et si de plus, quelle que soit la solution réalisable  $y \in \mathbb{R}^n$  du problème (1.1.1) on a nécessairement  $F(y) \leq F(x^*)$ . Autrement dit, une solution réalisable est optimale si elle maximise la fonction objectif sur l'ensemble des solutions réalisables.

**Remarque 1.1.** L'ensemble  $S$  est un polyèdre dans  $\mathbb{R}^n$ , c'est-à-dire une intersection finie de demi-espaces fermés de  $\mathbb{R}^n$ . Si  $S$  est de plus borné (cela n'est pas nécessairement le cas), et puisque la fonction objectif est une fonction continue, l'existence d'au moins une

solution optimale est alors garantie par le théorème des bornes atteintes. Dans tous les cas, nous verrons que si la fonction est majorée sur  $S$ , alors le problème de maximisation a toujours au moins une solution réalisable.

Un même problème d'optimisation linéaire peut être écrit de diverses manières équivalentes, les unes aux autres. Parmi ces versions, nous distinguerons les formes canoniques et les formes standards.

**Définition 1.4.** On appelle problème d'optimisation linéaire sous forme canonique un programme linéaire de la forme

$$\begin{cases} \text{maximiser} & \sum_{j=1}^n c_j x_j \\ \sum_{j=1}^n a_{ij} x_j \leq b_i & (i = 1, \dots, m) \\ x_j \in \mathbb{R} & (j = 1, \dots, n). \end{cases} \quad (1.1.2)$$

où les  $b_i$ ,  $i = 1, \dots, m$ ; les  $c_j$ ,  $j = 1, \dots, n$ ; et les  $a_{ij}$ ,  $i = 1, \dots, m$ ;  $j = 1, \dots, n$  sont des constantes réelles.

En écriture matricielle, un problème sous forme canonique s'écrit donc :

$$\begin{cases} \text{maximiser} & c^t x \\ Ax \leq b \\ x \in \mathbb{R}^n \end{cases} \quad (1.1.3)$$

où  $c = (c_j)_{j=1, \dots, n}$  et  $x = (x_j)_{j=1, \dots, n}$  sont des vecteurs réels,  $A = (a_{ij})_{i=1, \dots, m; j=1, \dots, n}$  est une  $m \times n$ -matrice réelle et  $b = (b_i)_{i=1, \dots, m}$  est un vecteur réel.

Il est immédiat que tout problème d'optimisation linéaire sous forme canonique est un problème d'optimisation linéaire sous forme générale.

**Définition 1.5.** On appelle problème d'optimisation linéaire sous forme standard, tout programme linéaire de la forme

$$\begin{cases} \text{maximiser} & \sum_{j=1}^n c_j x_j \\ \sum_{j=1}^n a_{ij} x_j = b_j & (i = 1, \dots, m) \\ x_j \in \mathbb{R} & (j = 1, \dots, n) \end{cases} \quad (1.1.4)$$

où  $c = (c_j)_{j=1, \dots, n}$  et  $x = (x_j)_{j=1, \dots, n}$  sont des vecteurs réels,  $A = (a_{ij})_{i=1, \dots, m; j=1, \dots, n}$  est une  $m \times n$ -matrice réelle et  $b = (b_i)_{i=1, \dots, m}$  est un vecteur réel.

En écriture matricielle, un problème sous forme standard s'écrit donc

$$\begin{cases} \text{maximiser} & c^t x \\ Ax = b \\ x \in \mathbb{R}^n \end{cases} \quad (1.1.5)$$

où  $c = (c_j)_{j=1, \dots, n}$  et  $x = (x_j)_{j=1, \dots, n}$  sont des vecteurs réels,  $A = (a_{ij})_{i=1, \dots, m; j=1, \dots, n}$  est une  $m \times n$ -matrice réelle et  $b = (b_i)_{i=1, \dots, m}$  est un vecteur réel.

A tout problème d'optimisation linéaire sous forme canonique, on peut associer un problème d'optimisation linéaire sous forme standard.

**Remarque 1.2.** Sans perte de généralité, on peut supposer que dans un problème sous forme standard, les lignes de  $A$  sont linéairement indépendantes (si ce n'est pas le cas soit certaines contraintes sont redondantes, soit l'ensemble des contraintes est vide). Dans la suite, lorsque nous parlerons de problème sous forme standard, nous supposerons implicitement que les lignes de  $A$  sont linéairement indépendantes, autrement dit que  $\text{rang}(A) = m$  ce qui implique également que  $n \geq m$ .

**Définition 1.6.** Considérons le problème d'optimisation linéaire sous forme standard

$$(P) \begin{cases} \text{maximiser} & c^t x \\ Ax = b \\ x \in \mathbb{R}^n \end{cases} \quad (1.1.6)$$

**Définition 1.7.** Soit  $B \in \{1, 2, \dots, n\}$  un ensemble d'indices avec  $\text{card}(B) = n$  tel que les colonnes  $A^j, j \in B$ , de  $A$  sont linéairement indépendantes. On dit que l'ensemble  $B$  des

indices est une base.

- les variables  $x_B = (x_j, j \in B)$  sont appelées variables de base.
- les variables  $x_N = (x_j, j \notin B)$  sont appelées variables hors base.

**Définition 1.8.** On dit que  $x = (x_B, x_N)$  est solution de base associée à la base  $B$  si  $x_N = 0$ .

$x = (x_B, x_N)$  est une solution de base réalisable si  $x_N = 0$  et  $x_B = A_B^{-1}b \geq 0$ . Dans ce cas, on dit que la base  $B$  est réalisable.

**Définition 1.9.** Etant donnée une base réalisable  $B$  du programme linéaire  $(P)$ , le programme linéaire équivalent à  $(P)$  :

$$\begin{cases} \hat{c}^N x_N = z(\max) - \pi b \\ x_B + A_B^{-1} A^N x_N = A_B^{-1} b \quad x \geq 0 \end{cases}$$

est dit forme canonique de  $(P)$  par rapport à la base  $B$  où :

- $N = \{1, 2, \dots, n\} \setminus B$
- $\pi = c^B A_B^{-1}$  est dit vecteur multiplicateur relatif à la base  $B$
- $\hat{c} = c - \pi A$  est dit vecteur coût réduit relatif à la base  $B$

est dit forme canonique de  $(P)$  par rapport à la base  $B$ .

**Théorème 1.1.** Si le vecteur coût réduit  $\hat{c}$  relatif à une base réalisable  $B$  est négatif ou nul, la solution de base correspondante est solution optimale de  $(P)$ . La base  $B$  est alors dite base optimale.

Un problème linéaire continu peut être résolu en temps polynomial (Khachiyan 1979) [29]. Il existe des algorithmes polynomiaux efficaces pour résoudre un programme linéaire comme ceux dits de points intérieurs initiés par Karmarkar (1984) [36]. Néanmoins l'algorithme du simplexe (Dantzig 1947) [10] est le plus célèbre (et le plus efficace dans le cas général) des algorithmes de résolution, bien qu'il ne soit pas polynomial!

L'algorithme du simplexe permet de résoudre les problèmes de  $PL$  en construisant tout d'abord une solution réalisable qui est un sommet du polytope puis en se déplaçant selon les arêtes du polytope pour atteindre des sommets pour lesquels la valeur de l'objectif est de plus en plus grande, jusqu'à atteindre l'optimum.

**Algorithme du simplexe**

Soit à résoudre le programme linéaire :

$$\begin{cases} \max & c^t x \\ Ax = & b \\ x \in & \mathbb{R}^n \end{cases} \quad (1.1.7)$$

où  $c = (c_j)_{j=1,\dots,n}$  et  $x = (x_j)_{j=1,\dots,n}$  sont des vecteurs réels,  $A = (a_{ij})_{i=1,\dots,m;j=1,\dots,n}$  est une  $m \times n$ -matrice réelle et  $b = (b_i)_{i=1,\dots,m}$  est un vecteur réel.  $B$  une base réalisable de départ.

## 1. Test d'optimalité

- Exprimer la fonction objectif à partir des variables hors base.
- Si tous les coûts sont négatifs ou nuls Alors STOP, la solution obtenue est la solution optimale.

## 2. Choix de la variable entrante

- Choisir la variable hors base dont le coût est positif et le plus grand possible. Soit  $x_r$  la variable entrante.

## 3. Choix de la variable sortante

- La variable sortante est la première à s'annuler : c'est celle pour laquelle le  $\frac{b_i}{a_{ir}}$  est le plus petit avec  $a_{ir} > 0$ . Soit  $x_s$  la variable sortante.

## 4. Déterminer la nouvelle solution de base : opération du pivot

- Remplacer les  $a_{ij}$  par  $a_{ij} - a_{jr} \frac{a_{si}}{a_{sr}}$ .
- Remplacer les  $b_j$  par  $b_j - a_{jr} \frac{b_s}{a_{sr}}$ .

Retour à (1)

**1.2 Programmation mono-objectif discrète**

Un problème de programmation linéaire en nombres entiers PLNE n'est pas un programme linéaire dans le sens où son domaine de réalisabilité n'est pas un polyèdre, mais

un ensemble discret de points. Pourtant, on peut le décrire comme un  $PL$  auquel on ajoute la contrainte supplémentaire que certaines variables ne peuvent prendre que des valeurs entières. On obtient dans ce cas, le programme linéaire en nombres entiers suivant :

$$\begin{cases} \max & z(x) \\ & x \in S, x \text{ vecteur entier} \end{cases} \quad (1.2.1)$$

où  $x$  est un vecteur de  $n$  variables entières positives ou nulles,  $z$  est une fonction scalaire et  $S$  l'ensemble des contraintes défini comme suit :  $S = \{x \in \mathbb{R}^n / Ax \leq b, x \geq 0\}$ , tel que  $A \in \mathbb{Z}^{m,n}, b \in \mathbb{Z}^m$ .

Le PLNE est un problème NP-complet, car de nombreux problèmes NP-complets peuvent être exprimés comme des PLNE (par exemple trouver un stable dans un graphe  $G = (V, E)$  revient à trouver un vecteur  $x \in \{0, 1\}^E$  satisfaisant  $x_u + x_v \leq 1$  pour toute arête  $uv \in E$ ).

Les deux principales familles de méthodes actuellement connues pour résoudre les problèmes de programmation linéaire en nombres entiers sont les méthode des coupes et les méthodes arborescentes. Nous présentons ci-dessous la méthode "Branch and Bound", proposée par Land et Doig en 1960 [25] et une méthode de coupe, proposée en 1958 par R.Gomory [39].

### 1.2.1 Méthode "Branch and Bound"

L'algorithme de séparation et évaluation ("Branch and Bound"), repose sur une méthode arborescente de recherche d'une solution optimale par séparations et évaluations, en représentant les états solutions par un arbre d'états, avec des noeuds, et des feuilles. Le "Branch and Bound" est basé sur trois axes principaux :

#### – L'évaluation

L'évaluation permet de réduire l'espace de recherche en éliminant quelques sous ensembles qui ne contiennent pas la solution optimale. L'objectif est d'essayer d'évaluer l'intérêt de l'exploration d'un sous-ensemble de l'arborescence. Le "Branch and Bound" utilise une élimination de branches dans l'arborescence de recherche de la



manière suivante : la recherche d'une solution de coût maximal, consiste à mémoriser la solution de plus haut coût rencontré pendant l'exploration, et à comparer le coût de chaque noeud parcouru à celui de la meilleure solution. Si le coût du noeud considéré est inférieur au meilleur coût, on arrête l'exploration de la branche et toutes les solutions de cette branche seront nécessairement de coût plus bas que la meilleure solution déjà trouvée.

– **La séparation**

La séparation consiste à diviser le problème en sous-problèmes. Ainsi, en résolvant tous les sous-problèmes et en gardant la meilleure solution trouvée, on est assuré d'avoir résolu le problème initial. Cela revient à construire un arbre permettant d'énumérer toutes les solutions. L'ensemble de noeuds de l'arbre qu'il reste encore à parcourir comme étant susceptibles de contenir une solution optimale, c'est-à-dire encore à diviser, est appelé ensemble des noeuds actifs.

– **La stratégie de parcours**

**La largeur d'abord :** Cette stratégie favorise les sommets les plus proches de la racine en faisant moins de séparations du problème initial. Elle est moins efficace que les deux autres stratégies présentées.

**La profondeur d'abord :** Cette stratégie avantage les sommets les plus éloignés de la racine (de profondeur la plus élevée) en appliquant plus de séparations au problème initial. Cette voie mène rapidement à une solution optimale en économisant la mémoire.

**Le meilleur d'abord :** Cette stratégie consiste à explorer des sous-problèmes possédant la meilleure borne. Elle permet aussi d'éviter l'exploration de tous les sous-problèmes qui possèdent une mauvaise évaluation par rapport à la valeur optimale.

### 1.2.2 Coupes de Gomory [39]

L'algorithme utilisant les coupes de Gomory procède comme suit :

Soit  $\bar{x}$  une solution optimale du problème relaxé ( $PL$ ) trouvée par la méthode du simplexe. Si  $\bar{x}$  est une solution entière, elle est optimale pour ( $PLNE$ ). Sinon, choisir une

variable  $x_j$  telle que la valeur  $\bar{x}_j$  est fractionnaire et considérer la ligne correspondante du tableau du simplexe, par exemple la ligne  $i$  :

$$x_j + \sum_{k \in N} \hat{a}_{ik} x_k = \bar{x}_j$$

où  $N$  est l'ensemble des indices des variables hors-base.

La contrainte

$$\sum_{k \in N} f(\hat{a}_{ik}) x_k \geq f(\bar{x}_j) \quad (1.2.2)$$

est alors est alors déduite de l'expression précédente, où  $f(r) = r - [r]$  désigne la partie fractionnaire du nombre réel  $r$ .

Cette coupe, appelée coupe fractionnaire de Gomory, ou coupe fondamentale, peut être rajouter au tableau courant du simplexe. Dans la pratique, les coupes fractionnaires de Gomory convergent très lentement. D'autres coupes ont été introduites dont l'intention de les rendre plus performantes. En particulier, Gomory [39] lui même a donné les coupes suivantes :

**Proposition 1.** *Pour tout entier naturel  $t$ , l'inéquation*

$$\sum_{k \in N} f(t\hat{a}_{ik}) x_k \geq f(t\bar{x}_j) \quad (1.2.3)$$

est une coupe. De plus, si  $f(\hat{a}_{ik}) < \frac{1}{2}$  et  $\frac{1}{2} \leq tf(\hat{a}_{ik}) < 1$ , alors la coupe (1.2.3) est plus profonde que la coupe (1.2.2).

**Proposition 2.** *L'inéquation*

$$\sum_{k \in N} \min\left\{f(\hat{a}_{ik}), f(\bar{x}_j) \frac{1 - f(\hat{a}_{ik})}{1 - f(\bar{x}_j)}\right\} x_k \geq f(\bar{x}_j) \quad (1.2.4)$$

*est une coupe. De plus, elle est plus profonde que la coupe (1.2.2)*

*De même, pour tout entier naturel  $t$ , l'inéquation*

$$\sum_{k \in N} \min\left\{f(t\hat{a}_{ik}), f(t\bar{x}_j) \frac{1 - f(t\hat{a}_{ik})}{1 - f(t\bar{x}_j)}\right\} x_k \geq f(t\bar{x}_j) \quad (1.2.5)$$

est une coupe plus profonde que (1.2.3)

### 1.3 La programmation linéaire fractionnaire mono-objectif

Les programmes fractionnaires [33] 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, linéaires ou non linéaires, en nombres entiers ou en continu, ont une multitude d'applications que ce soit en optimisation combinatoire, en programmation stochastique, en bases de données ou en économie [43], [44], [45], [46], [47], [48], [49]. La programmation fractionnaire linéaire (PLF) se réfère au même type de problèmes d'optimisation mais où le numérateur et le dénominateur sont des fonctions affines, et le domaine des solutions réalisables est un polyèdre convexe.

#### 1.3.1 Notations et définitions

Étant donnée  $f, h$  et  $g_i, i = 1, \dots, m$ , des fonctions réelles définies sur  $\mathbb{R}^n$ , avec  $h$  ne s'annulant pas sur un sous-ensemble  $X$  de  $\mathbb{R}^n$ , le problème de programmation fractionnaire consiste à déterminer un élément  $x^*$  de  $X$  optimisant le rapport  $f(x)/g(x)$  sur un domaine défini par le système de contraintes  $g(x) \leq 0$  avec  $x$  dans l'ensemble  $X$ . Il a donc la forme suivante :

$$\begin{cases} \text{maximiser} & \frac{f(x)}{h(x)} \\ g(x) \leq 0 \\ x \in X \end{cases} \quad (1.3.1)$$

et doit vérifier les hypothèses classiques suivantes :

- L'ensemble des solutions réalisables de  $(PF)$  est non vide,
- les fonctions  $f, h$  et  $g$  sont continues sur  $\mathbb{R}^n$ ,
- $\forall x \in X : h(x) > 0$ .

Le programme 1.3.1 est dit fractionnaire linéaire, ou encore hyperbolique, lorsque  $f, h$  et  $g$  sont des fonctions linéaires ou affines de la variable  $x$ . Il se modélise alors comme suit :

$$(PF) \begin{cases} \text{maximiser} & \frac{cx + \alpha}{dx + \beta} \\ Ax \leq b \\ x \in X \end{cases} \quad (1.3.2)$$

où  $\alpha$  et  $\beta$  sont des réels,  $c$  et  $d$  sont des vecteurs de  $\mathbb{R}^n$  ;  $A$  est une matrice réelle de format  $m \times n$  et  $b$  est un vecteur de  $\mathbb{R}^m$ .

Selon Steuer [40], les programmes fractionnaires linéaires présentent l'intérêt particulier d'avoir pour leurs fonctions objectifs des courbes de niveaux linéaires. Pour illustrer cette propriété, considérons une  $\bar{z}$ -courbe niveau quelconque de la fonction objectif :

$$\bar{z} = \frac{cx + \alpha}{dx + \beta} \Leftrightarrow (c - \bar{z}d)x = \bar{z}\beta - \alpha \quad (1.3.3)$$

$\bar{z}$  étant quelconque, il est donc clair que chaque courbe de niveau est une expression linéaire sur l'ensemble des solutions réalisables, respectant la contrainte de positivité stricte du dénominateur.

Si un programme fractionnaire linéaire possède une solution optimale, il existera donc au moins un point extrême optimal de  $S$ .

**Théorème 1.2.** [30] *La solution optimale d'un programme fractionnaire linéaire est représentée par l'un des sommets extrêmes du domaine admissible.*

Malgré la linéarité des courbes de niveaux de la fonction objectif, lorsque  $c \neq 0$ ,  $d \neq 0$  et  $c \neq wd$  pour tout  $w \in \mathbb{R}$ , les courbes de niveaux ne sont pas parallèles comme dans le cas de la programmation linéaire.

On appelle ensemble de rotation, l'ensemble de tous les points d'intersection entre la 0-courbe niveau du numérateur et la 0-courbe niveau du dénominateur. L'ensemble de rotation est appelé point de rotation dans  $\mathbb{R}^2$  et axe de rotation dans  $\mathbb{R}^3$ .

### 1.3.2 Relation entre PL et PLF

Le problème général de programmation linéaire PL peut être écrit de la forme :

$$(P) \begin{cases} \max z = cx \\ Ax \leq b \\ x \in X \end{cases}$$

et la formulation d'un *PLF* est comme suit :

$$(PF) \begin{cases} \max \frac{cx + \alpha}{dx + \beta} \\ Ax \leq b \\ x \in X \end{cases}$$

où  $\alpha$  et  $\beta$  sont des réels,  $c$  et  $d$  sont des vecteurs de  $\mathbb{R}^n$ ,  $A$  est une matrice réelle de format  $m \times n$ ,  $b$  est un vecteur de  $\mathbb{R}^m$  et  $X \subset \mathbb{R}^n$

Il est clair que si  $d$  est un vecteur nul avec  $\beta = 1$  alors,  $(PF)$  devient un problème de *PL*. Dans ce cas,  $PF$  est une généralisation d'un *PL*.

Il y a des cas où *PLF* peut être remplacé par *LP* approprié. Ces cas sont discutés comme suit :

**Cas 1 :**

Si  $d = 0$  et  $\beta \neq 0$  alors  $z$  devient  $z = \frac{c}{\beta}x + \frac{\alpha}{\beta} = \frac{z'}{\beta}$  où  $z' = cx + \alpha$  est une fonction linéaire. Dans ce cas  $z$  peut être remplacé par  $\frac{z'}{\beta}$  correspondant au même ensemble de solution réalisable  $S$ , alors  $(PF)$  devient un  $(PL)$ .

**Cas 2 :**

Si  $c = 0$  alors  $z$  devient  $z = \frac{\alpha}{dx + \beta} = \frac{\alpha}{z'}$ , où  $z' = dx + \beta$  est une fonction linéaire. Donc  $(PF)$  devient un  $(PL)$ .

Dans ce cas,  $z$  devient linéaire sur le même ensemble des solutions réalisables  $S$ .

**Cas 3 :**

Si  $c$  et  $d$  sont linéairement dépendant, alors, il existe  $\mu \neq 0$  tel que  $c = \mu d$ , donc

$$z = \frac{\mu dx + \alpha}{dx + \beta} = \mu + \frac{\alpha - \mu\beta}{dx + \beta}$$

- Si  $\alpha - \mu\beta = 0$ , alors  $z = \mu$  est une constante.
- Si  $\alpha - \mu\beta \neq 0$ , alors  $z$  est une fonction linéaire et  $(PF)$  devient un  $(PL)$ .

Si  $c \neq 0$  et  $d \neq 0$ , alors il faut trouver une nouvelle façon de convertir le problème  $(PF)$  à  $(PL)$  pour le résoudre.

### 1.3.3 Résolution d'un programme linéaire fractionnaire

Dans la littérature émergent trois grandes stratégies de résolution d'un programme fractionnaire :

#### Résolution directe

Dans cette stratégie, le programme fractionnaire est traité 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 et variables entières ([19], [51], [11], [31]).

Une des premières méthodes [7] de résolution concernant le programme hyperbolique est une version améliorée de la méthode de Martos [30].

Rappelons le programme linéaire fractionnaire continu  $(PF)$  :

$$(PF) \begin{cases} \max & \frac{cx + \alpha}{dx + \beta} \\ \text{s.c} & Ax \leq b \\ & x \geq 0 \end{cases}$$

où  $\alpha$  et  $\beta$  sont des réels,  $c$  et  $x$  des vecteurs de  $\mathbb{R}^n$ ,  $d$  un vecteur non nul de  $\mathbb{R}^n$ ,  $A$  une matrice réelle d'ordre  $m \times n$  et  $b$  un vecteur de  $\mathbb{R}^m$ .

**Définition 1.10.** Un point réalisable  $\bar{x}$  est une solution niveau optimale pour le problème  $(PF)$ , si  $\bar{x}$  est une solution optimale pour le programme linéaire suivant :

$$(P_{\bar{x}}) \begin{cases} \max & cx + \alpha \\ \text{s.c} & Ax \leq b \\ & dx = d\bar{x} \\ & x \geq 0 \end{cases}$$

Si de plus,  $\bar{x}$  est un point extrême de  $S = \{Ax \leq b, x \geq 0\}$ , Alors  $\bar{x}$  est dite solution niveau optimale de base.

L'algorithme de Cambini et Martein [7] génère une séquence finie de solutions niveau optimales, dont la première solution  $x_0$  est une solution optimale du programme linéaire  $(P_{x_0})$ . Si cette solution est unique, alors elle est solution niveau optimale pour le  $(PF)$  sinon on résout  $(P_{x_0})$  pour obtenir une solution niveau optimale de base.

### Résolution 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 de contraintes. Une séquence de résolutions de ce type de problème fournit une solution optimale du programme fractionnaire. Plusieurs algorithmes ont été proposés, initialement en 1956 par Isebell et Marlow [27] pour des programmes hyperboliques, ce n'est qu'à partir de 1967 que cette approche a connue un grand élan avec Dinkelbach [53], qui l'a généralisé aux programmes fractionnaires non linéaires.

Soit le programme fractionnaire suivant :

$$(P) \begin{cases} \max & \frac{f(x)}{h(x)} \\ \text{s.c} & g(x) \leq 0 \\ & x \in X \end{cases}$$

Le programme paramétré associé consiste à simplifier l'objectif en combinant linéairement le numérateur et le dénominateur par l'intermédiaire d'un paramètre  $\lambda \in \mathbb{R}$ . Il est donc défini comme suit

$$(P(\lambda)) \begin{cases} \max & f(x) - \lambda h(x) \\ \text{s.c} & g(x) \leq 0 \\ & x \in X \end{cases}$$

pour tout  $\lambda \in \mathbb{R}$ .

Dans le cas d'un programme hyperbolique

$$(PF) \begin{cases} \max & \frac{cx + \alpha}{dx + \beta} \\ \text{s.c} & g(x) \leq 0 \\ & x \in X \end{cases}$$

le programme paramétré est un programme linéaire dont l'objectif est fonction de  $\lambda \in \mathbb{R}$  :

$$\begin{cases} \max & (\alpha - \lambda\beta) + (c - \lambda d)x \\ \text{s.c} & g(x) \leq 0 \\ & x \in X \end{cases}$$

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

**Proposition 3.** (Dinkelbach [53]) *Toute solution optimale de  $P(\lambda^*)$  est une solution optimale de  $(P)$ .*

Soit  $v(\lambda)$  la valeur optimale du programme paramétré.

**Proposition 4.** [53]  $v(\lambda) = 0$  si et seulement si  $\lambda = \lambda^*$

### Résolution d'un programme équivalent

Un changement de variables permet de simplifier aussi l'objectif mais en faisant augmenter le nombre de variables et de contraintes. Charnes and Cooper [8] sont les premiers à avoir linéarisés un problème hyperbolique en un problème linéaire équivalent.

$$(PF) \begin{cases} \max & \frac{cx + \alpha}{dx + \beta} \\ \text{s.c} & Ax \leq b \\ & x \geq 0 \end{cases}$$

s'effectue en introduisant deux nouvelles variables

$$y = \left(\frac{1}{dx + \beta}\right)x \quad \text{et} \quad t = \frac{1}{dx + \beta}$$



Pour aboutir à un programme linéaire équivalent

$$(P) \begin{cases} \max & cy + \alpha t \\ \text{s.c} & Ay - bt \leq 0 \\ & dy + \beta t = 1 \\ & y, t \geq 0 \end{cases}$$

**Proposition 5.** [8]

Si  $(y^*, t^*)$  est une solution optimale de  $(P)$ , alors  $t^* > 0$  et  $x^* = \frac{y^*}{t^*}$  est une solution optimale de  $(PF)$ .

Cette transformation en un programme linéaire équivalent a pour but d'appliquer les algorithmes standards tels que la méthode du simplexe (Arsham et Khan [24] et Charnes and Cooper [8]).

Pour les programmes fractionnaires en variables entières, Granot et Granot [11] proposent une méthode de génération de coupes (de type Gomory) appliquée au programme linéaire  $(P)$ .

## Chapitre 2

# Sur la programmation multi-objectif

Les problèmes d'optimisation combinatoire issus des problématiques réelles sont la plupart du temps de nature multi-objectif, car plusieurs critères d'évaluation souvent contradictoires sont à considérer simultanément. Optimiser un tel problème relève donc de l'optimisation combinatoire multi-objectif.

L'optimisation multi-objectif possède ses racines dans les travaux en économie de Edgeworth [17] et Pareto [52]. Elle a ainsi été initialement utilisée en économie et dans les sciences du management, puis graduellement dans les sciences pour l'ingénieur.

Pourtant, malgré l'intérêt indéniable de la modélisation et de la résolution multi-objectif des problèmes rencontrés dans industrie, dans les télécommunications et et d'autres domaines de la vie quotidienne, peu de travaux ont été réalisés en optimisation combinatoire multi-objectif avant les années 80-90. Mais depuis, un fort intérêt a été montré pour l'aide à la décision multi-objectif qui consiste pour un problème comportant plusieurs objectifs, à déterminer, parmi les solutions de meilleurs compromis entre les objectifs, la solution la plus intéressante pour le problème en question. Ainsi, une phase importante concerne l'optimisation multi-objectif qui recherche les solutions de compromis.

L'objet de ce chapitre est de présenter quelques notions fondamentales concernant les problèmes de programmation linéaire en nombres entiers dans le cas où plusieurs objectifs à optimiser sont à considérer.

## 2.1 Sur la programmation multi-objectif discrète linéaire

Un problème d'optimisation mono-objectif peut être formulé comme suit :

$$(PL) \begin{cases} \text{optimiser} & z(x) \\ \text{sous} & x \in S \end{cases} \quad (2.1.1)$$

où  $x$  est un vecteur de  $n$  variables,  $z$  est une fonction scalaire et  $S$  l'ensemble des contraintes défini comme suit :  $S = \{x \in \mathbb{R}^n / Ax \leq b\}$ ,  $A$  est une  $m \times n$ -matrice réelle et  $b$  un  $m$ -vecteur réel.

Un problème d'optimisation multi-objectif est un problème d'optimisation pour lequel  $k$  ( $k > 1$ ) fonctions objectifs à optimiser (minimiser ou maximiser). Il se définit de la façon suivante :

$$(P) \begin{cases} \text{optimiser} & z(x) = (z_1(x), z_2(x), \dots, z_k(x)) \\ \text{sous} & x \in S \end{cases} \quad (2.1.2)$$

où  $S = \{x \in \mathbb{R}^n / Ax \leq b\}$  est l'ensemble des solutions réalisables de  $(P)$  (appelée aussi espace de décision) définies ci-dessus. A chaque solution réalisable  $x$  dans  $S$ , on associe son image  $z(x) = (z_1(x), z_2(x), \dots, z_k(x))$  dans  $\mathbb{R}^k$  (espace des critères) et on construit donc, l'ensemble  $C = \{y \in \mathbb{R}^k : y = z(x), x \in S\} = z(S)$ .

Sans perte de généralité nous supposons par la suite que nous considérons des problèmes de maximisation.

### 2.1.1 Concepts de base

Une différence fondamentale entre l'optimisation mono-objectif et l'optimisation multi-objectif repose sur le fait que pour la plupart des problèmes multi-objectif, les critères étant antagonistes, il n'existe pas de solution réalisable qui maximise tous les objectifs simultanément. Par conséquent, il n'existe plus de relation d'ordre total entre solutions. Celles-ci doivent être comparées par la relation de dominance de Pareto qui repose sur la définition suivante :

**Définition 2.1.** On dit qu'un vecteur  $z = (z_1, z_2, \dots, z_k)$  domine un vecteur  $y = (y_1, y_2, \dots, y_k)$ , si  $\forall i = 1 \dots k, z_i \geq y_i$  et  $\exists i \in \{1 \dots k\}$  tel que  $z_i > y_i$ .

**Définition 2.2.** Une solution réalisable  $x^* \in S$  est efficiente (ou efficace) si'il n'existe pas une autre solution  $x \in S$  telle que  $z(x)$  domine  $z(x^*)$ .

Résoudre le problème  $(P)$  revient à trouver, soit l'ensemble des solutions efficaces dans l'espace des décisions, soit l'ensemble des solutions non dominées dans l'espace des critères.

**Définition 2.3.** Soit  $\bar{x} \in S$  et  $C^{\geq}$  le cône semi-polaire positif généré par les gradients des  $k$  fonctions objectifs où  $C^{\geq} = \{x \in \mathbb{R}^n \mid Cx \geq 0, Cx \neq 0\} \cup \{0 \in \mathbb{R}^n\}$ . L'ensemble de dominance sur  $\bar{x}$  est donné par :  $D_{\bar{x}} = \{\bar{x}\} \cup C^{\geq} = \{\bar{x} + x \mid x \in C^{\geq}\}$ .

**Définition 2.4.** Soit  $D_{\bar{x}}$  l'ensemble dominant en  $\bar{x} \in S$ . Alors,  $\bar{x}$  est efficace si et seulement si  $D_{\bar{x}} \cap S = \{\bar{x}\}$ .

**Corollaire 1.** Si  $C^{\geq} = \{0_{\mathbb{R}^n}\}$ , alors  $\forall x \in S, x$  est solution efficace.

**Définition 2.5.** Le vecteur idéal  $z^* = (z_1^*, z_2^*, \dots, z_n^*)$  est le vecteur qui optimise chacune des fonctions objectifs  $z_i$ , i.e :  $z_i^* = \max\{c_i x \mid x \in S\}$

La matrice des gains associée au problème  $(P)$  peut être représentée par une matrice carrée  $G$  de dimension  $k(k > 1)$  comme suit :

$$\begin{pmatrix} z_1^* & z_{12} & \dots & z_{1k} \\ z_{21} & z_2^* & \dots & z_{2k} \\ \vdots & \vdots & \vdots & \vdots \\ z_{31} & z_{32} & \dots & z_k^* \end{pmatrix} \quad (2.1.3)$$

avec  $z_i^* = \max_{x \in D} z_i = c_i x_i^*, \forall i = 1, \dots, k$ , et  $z_{ij} = c_i x_j^*, \forall i = 1, \dots, k, \forall j = 1, \dots, k$ , avec  $i \neq j$ .

Il faut noter que cette matrice dépend de la solution optimale  $x_i^*$  et peut donc, ne pas être unique.

Le point nadir  $N \in \mathbb{R}^k : n_i = \min\{c_i x \mid x \text{ efficace pour } (P)\}$ .

Si la matrice des gains n'est pas unique alors, le point  $N$  aussi n'est pas unique.

### 2.1.2 Caractérisation d'une solution efficace

**Test de Benson :** Le théorème de Benson [4] permet de tester l'efficacité d'une solution réalisable d'un problème d'optimisation multi-objectif.

L'idée est de choisir une solution réalisable initiale  $x^* \in S$  et, si la solution elle-même n'est pas efficace, produire une solution réalisable  $x$  telle que  $z(x)$  domine  $z(x^*)$ .

Soient  $v_i = z_i(x) - z_i(x^*)$ ,  $i = 1, \dots, k$  des variables non négatives dont on doit maximiser leur somme.

Considérons le problème (PE) suivant :

$$(PE) \begin{cases} \max \varphi = \sum_{i=1}^k v_i \\ z_i(x) - z_i(x^*) - v_i = 0, i = 1, \dots, k; \\ x \in S \\ v_i \geq 0, i = 1, \dots, k. \end{cases} \quad (2.1.4)$$

L'exemple illustratif 2.1 dans l'espace des critères, démontre l'idée.

La solution initiale  $z(x^*)$  a des valeurs inférieures à  $z(\hat{x})$ , maximisant la déviation totale de  $\hat{v}_1 + \hat{v}_2$  où l'intention est de trouver une solution dominante et qui est efficace.

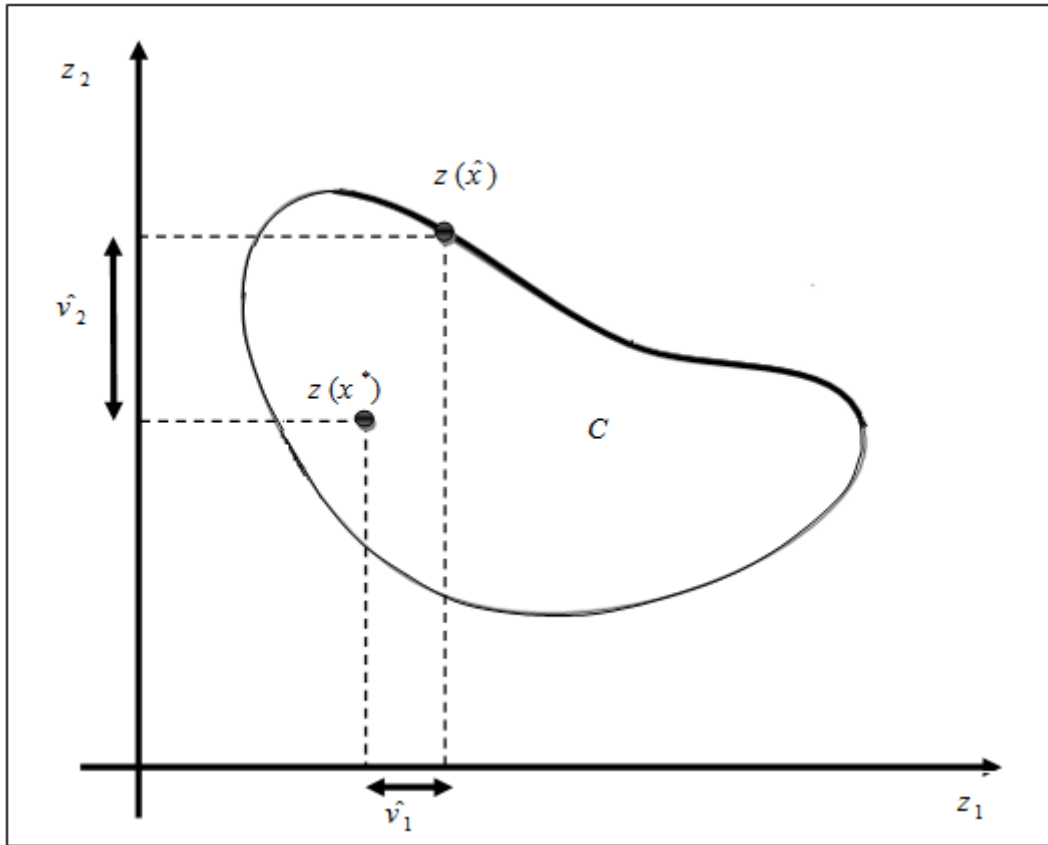


FIGURE 2.1 – Illustration du problème de Benson 2.1.4.

**Théorème 2.1.** [4]. Soit  $x^*$  une solution réalisable pour le problème (P).  $x^*$  est une solution efficace pour le problème (P) si et seulement si la valeur optimale de la fonction objectif  $\varphi$  est nulle dans le programme linéaire 2.1.4.

**Preuve 1.** Soit  $(x, v)$  une solution réalisable de problème 2.1.4. Comme  $v_i \geq 0, i = 1, \dots, k$  et la définition de  $v_i$  est  $z_i(x) - z_i(x^*)$  nous avons

$$\begin{aligned} \sum_{i=1}^k v_i = 0 &\iff v_i = 0 \quad i = 1, \dots, k \\ &\iff z_i(x) - z_i(x^*) = 0 \quad i = 1, \dots, k \\ &\iff z_i(x) = z_i(x^*) \quad i = 1, \dots, k \end{aligned}$$

Dans ce cas la,  $x^*$  est efficace. Si, d'autre part,  $x^*$  est efficace, l'ensemble des solution réalisable du problème 2.1.4 se compose de  $(x, v)$  pour chaque  $x \in S$  et  $z(x) = z(x^*)$  ainsi  $v = 0$ .

**Proposition 6.** Si le problème 2.1.4 a une solution optimal  $(\hat{x}, \hat{v})$  (et la valeur objectif

optimal est finie) alors  $\hat{x}$  est efficace.

**Preuve 2.** Supposons que la solution  $\hat{x}$  n'est pas efficace. Alors il exist une solution  $x' \in S$  tel que  $z_i(\hat{x}) \leq z_i(x'), \forall i = 1, \dots, k$  et  $\exists j \in \{1, \dots, k\}, z_j(\hat{x}) < z_j(x')$ . Soit  $v' = z(x') - z(x^*)$ . Alors  $(x', v')$  est une solution réalisable de 2.1.4 car pour  $i$  fixé :

$$v'_i = z_i(x') - z_i(x^*) \geq z_i(\hat{x}) - z_i(x^*) = \hat{v}_i.$$

En outre,  $\sum_{i=1}^k v'_i > \sum_{i=1}^k \hat{v}_i$  comme  $v'_j > \hat{v}_j$ , alors  $(\hat{x}, \hat{v})$  n'est pas solution optimale pour 2.1.4

**Test de Ehrgott et al. [12] :**

**Définition 2.6.** Soit  $f : D \subset \mathbb{R}^n \rightarrow \mathbb{R}$  et  $\bar{x} \in D$ . Alors

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

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

**Théorème 2.2.** [12] La solution  $\bar{x} \in D$  est Pareto optimale pour le problème multi-objectif si et seulement si  $\bigcap_{i=1}^{i=k} L_{\geq} z_i(\bar{x}) = \bigcap_{i=1}^{i=k} L_{=} z_i(\bar{x})$ .

**Preuve 3.**  $\bar{x}$  est solution Pareto optimale pour

$\iff \nexists x \in D$  tel que  $[z_i(x) \geq z_i(\bar{x}) \forall i \in \{1, \dots, k\}$  et  $z_j(x) > z_j(\bar{x})$  pour au moins un  $j]$ .

$\iff \nexists x \in D$  tel que  $[x \in \bigcap_{i=1}^{i=k} L_{\geq} z_i(\bar{x})$  et  $\exists j \in \{1, \dots, k\} : x \in L_{>} z_j(\bar{x})]$ .

$\iff \nexists x \in D$  tel que  $[x \in \bigcap_{i=1}^{i=k} L_{\geq} z_i(\bar{x})$  et  $\exists j \in \{1, \dots, k\} : x \in L_{\geq} z_j(\bar{x}) \setminus L_{=} z_j(\bar{x})]$ .

$\iff \bigcap_{i=1}^{i=k} L_{\geq} z_i(\bar{x}) = \bigcap_{i=1}^{i=k} L_{=} z_i(\bar{x})$ .

### 2.1.3 Solutions supportées et non supportées

Sur le front Pareto, deux types de solutions peuvent être différenciées : les solutions supportées et les solutions non supportées. Les premières sont celles situées sur l'enveloppe convexe de l'ensemble des solutions (Figure 2.2) et peuvent donc être trouvées à l'aide d'une agrégation linéaire des objectifs [18]. Elles sont donc plus simples à obtenir que les

solutions non supportées. D'ailleurs, les premiers travaux en optimisation combinatoire multi-objectif se sont pour la plupart focalisés sur la recherche de ces solutions supportées en optimisant des combinaisons linéaires des objectifs utilisant différents vecteurs de poids.

**Théorème 2.3.** [18]

$$(P_\lambda) : \text{Max}\{\lambda^t Cx \mid x \in S\}$$

avec

$$\lambda \in \Lambda = \{\lambda \in \mathbb{R}^k \mid \sum_{i=1}^k \lambda_i = 1, \lambda_i > 0, i = 1, \dots, k\},$$

alors  $x$  est une solution efficace pour  $(P)$  si et seulement s'il existe  $\lambda \in \Lambda$  tel que  $x$  est une solution optimale du problème paramétrique  $(P_\lambda)$ .

Alors pourquoi ne pas se satisfaire des solutions supportées ? Tout d'abord parce que ces solutions peuvent ne représenter qu'un petit sous ensemble des solutions efficaces. De plus, ces solutions supportées ne sont pas forcément bien réparties le long du front et ne représentent pas toujours un bon compromis.

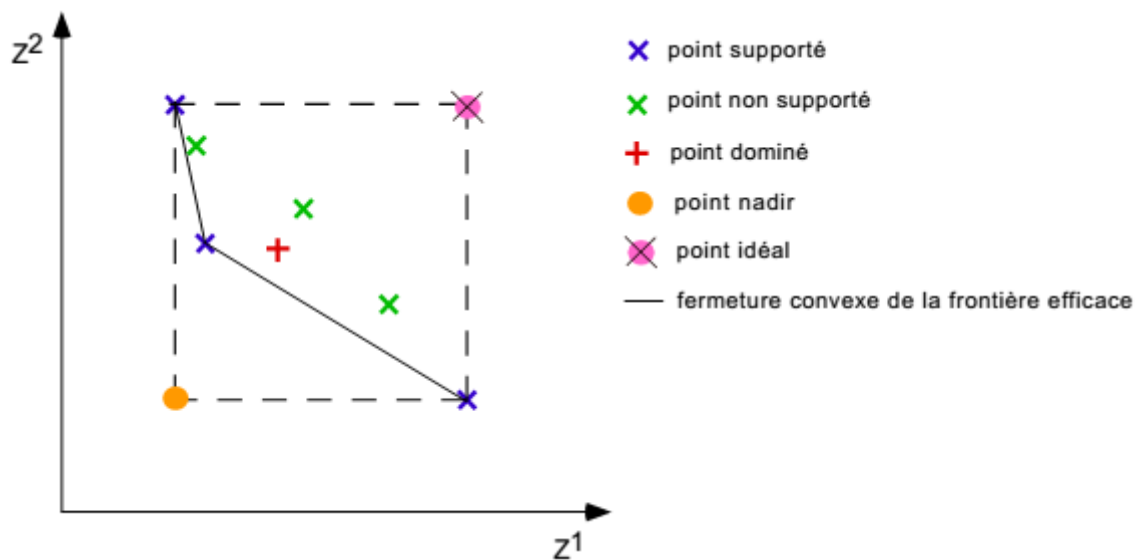


FIGURE 2.2 – Schéma des différentes solutions et points caractéristiques de l'espace des objectifs



### 2.1.4 Exemple illustratif

Considérons le programme linéaire bi-objectif en nombres entiers suivant :

$$(P) \begin{cases} \max z(x) = (x_1, -x_1 + x_2) \\ x \in D \end{cases} \quad (2.1.5)$$

où  $D = \{x = (x_1, x_2) \in \mathbb{N}^2 / x_1 + 2x_2 \leq 10; x_1 + x_2 \leq 6; x_1 \leq 4\}$  .

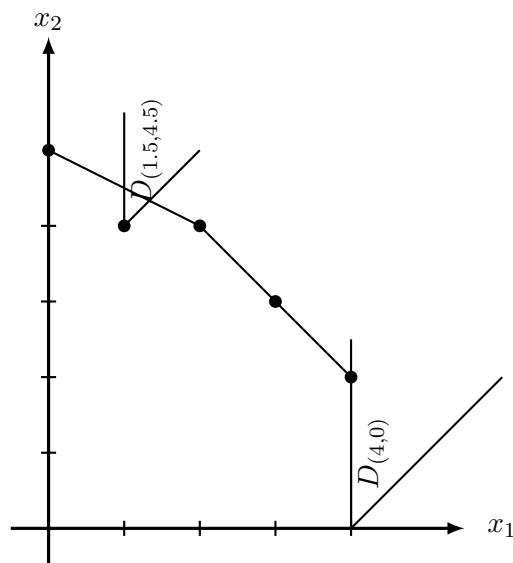


FIGURE 2.3 – Espace des décisions.

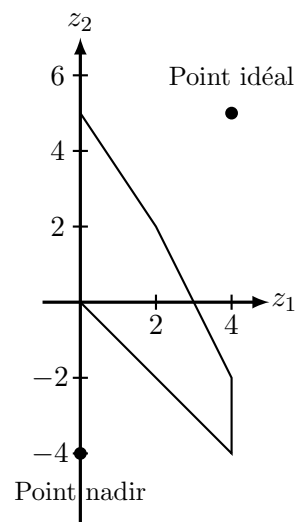


FIGURE 2.4 – Espace des critères.

– Les solutions efficaces du programme  $(P)$  sont :  $\{(4, 2), (3, 3), (2, 4), (1, 4), (0, 5)\}$ . Par

exemple le point  $x = (4, 2)$  est efficace car  $D_x \cap D = \{x\}$ , tandis que le point  $x = (4, 0)$

n'est pas efficace car  $D_x \cap D = \{(4, 2), (4, 0)\}$

- Deux solutions faiblement non dominées sont détectées :  $(4, -4), (4, -3)$ .
- Les solutions efficaces supportées sont :  $\{(4, 2), (3, 3), (2, 4), (0, 5)\}$  et une solution efficace non supportée :  $(1, 4)$ .
- Le point idéal est :  $(4, 5)$  et le point nadir est le point des coordonnées :  $(0, -4)$ .
- La matrice des gains, dans ce cas, est unique : 
$$\begin{pmatrix} 4 & 0 \\ -4 & 5 \end{pmatrix}$$

### 2.1.5 Approches de résolution

La solution d'un problème multi-objectif est un ensemble de solutions. Cependant, pour un problème réel, une seule solution pourra être déployée. Un choix par un décideur doit donc être effectué; le décideur peut intervenir en amont de la résolution, après celle-ci, ou de manière interactive. On distingue classiquement trois grandes familles de méthodes selon la manière dont sont combinés ces processus :

- Préférence à priori : le décideur définit ses préférences entre les différents objectifs avant d'utiliser la méthode d'optimisation.
- Préférence progressive : le décideur affine son choix de compromis au fur et à mesure du déroulement de la méthode d'optimisation.
- Préférence à posteriori : le décideur choisit la solution de son choix parmi l'ensemble des solutions fournies par la méthode d'optimisation.

Il est à noter que certaines méthodes n'entrent pas forcément dans une seule catégorie. En effet, la méthode qui consiste à agréger les différents objectifs à l'aide de poids est une méthode à préférence à priori; le décideur affectant les poids de manière à favoriser tel ou tel objectif. Cependant, si les poids sont affectés aléatoirement et si la méthode est itérée en changeant les poids à chaque exécution, il s'agit alors d'une méthode à préférence à posteriori. Nous nous intéressons ici à l'optimisation à posteriori de problèmes multi-objectif, c'est-à-dire l'étude de méthodes pour générer l'ensemble des solutions efficaces ou non dominées. Le choix d'une solution parmi cet ensemble n'est pas étudié dans ce document.

### 2.1.6 Quelques méthodes de résolution

Un grand nombre d'approches existent pour résoudre les problèmes multi-objectif. Certaines utilisent des connaissances du problème pour fixer des préférences sur les critères et ainsi contourner l'aspect multi-critère du problème. D'autres mettent tous les critères au même niveau d'importance, mais là aussi il existe plusieurs façons de réaliser une telle opération.

Nous nous placerons dans le cadre 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 des solutions de bons compromis par des méthodes exactes.

#### Méthode $\epsilon$ -contrainte. [22]

La méthode a été proposée par Haïmes et al. 1971. Elle consiste à transformer les  $(k - 1)$  objectifs en contraintes. L'objectif restant, qui peut être choisi arbitrairement, est la fonction d'objectif du problème d'optimisation mono-objectif qui en résulte, en d'autres termes dans cette approche, le problème consiste à optimiser une fonction  $f_i$  soumise à des contraintes sur les autres fonctions. En effet, en utilisant cette méthode itérativement, en repartant à chaque fois de la solution trouvée pour définir la borne suivante [22], il est possible en utilisant une méthode exacte mono-objectif de générer l'ensemble des solutions Pareto.

Soit le problème à résoudre :

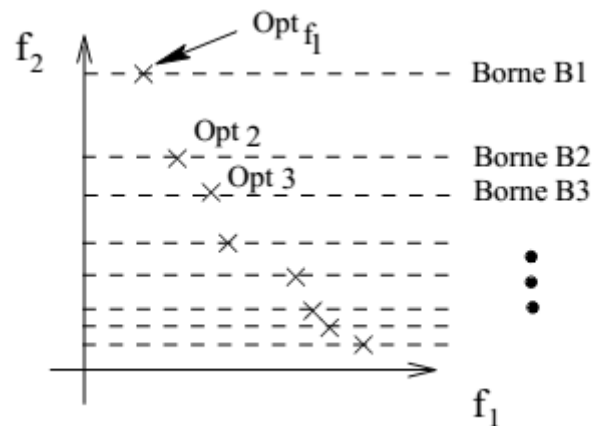
$$(P) \begin{cases} \min & f(x) = (f_1(x), f_2(x), \dots, f_k(x)) \\ \text{sous} & x \in S \end{cases} \quad (2.1.6)$$

où  $f : \mathbb{R}^n \rightarrow \mathbb{R}^k$  telle que  $f(x) = (f_i(x))_{i=1, \dots, k}$

Pour un objectif  $i \in \{1, \dots, k\}$  du problème (P), résoudre le problème suivant :

$$(P_i) \begin{cases} \min & f_i(x) \\ x \in S \\ f_j(x) \leq \epsilon_j \quad \forall j \in \{1, \dots, k\} \setminus \{i\} \end{cases} \quad (2.1.7)$$

où  $\epsilon_j$  est une borne supérieure pour le  $j^{\text{ème}}$  objectif.

FIGURE 2.5 – Illustration de la méthode  $\epsilon$ -contrainte.

La figure 2.5 illustre un exemple pour lequel, la solution efficace optimale pour l'objectif  $f_1$  est d'abord recherchée (solution  $\text{Opt}_{f_1}$ ). Cette solution détermine la borne  $B_1$  sur l'objectif  $f_2$  en dessous de laquelle l'objectif  $f_2$  va devoir être optimisé. Cela nous donne la solution  $\text{Opt}_{f_2}$  qui elle même détermine la borne  $B_2$ , etc...

La méthode Epsilon-Contrainte est facile à utiliser. Elle permet de trouver les solutions du front même les fronts non-convexe. L'inconvénient principal de cette méthode est de trouver les bornes de variation pour les fonctions objectifs qui seront transformées en contraintes.

### Méthode de Özlen et Azizoğlu. [37]

La méthode de Özlen et Azizoğlu est une amélioration de la méthode  $\epsilon$ -contrainte. Dans la méthode [22], les valeurs  $\epsilon_j$  sont diminuées d'une unité dans le cas de minimisation multi-objectif, la méthode décrite dans [37] diminue ces valeurs en tenant compte de la solution non dominée actuelle. En outre, la fonction objectif n'est plus l'un des critères, mais une combinaison pondérée et appropriée des critères de  $(P)$  qui assure l'efficacité des solutions obtenues.

Dans cette méthode, on montre que le problème multi-objectif  $(P)$  est équivalent (au sens des solutions efficaces) au problème suivant, obtenu en transformant le  $k^{i\text{me}}$  critère en contrainte :

$$\left\{ \begin{array}{l} \min \quad f_1(x) + \epsilon_k f_k(x) \\ \vdots \\ \min \quad f_{k-1}(x) + \epsilon_k f_k(x) \\ f_k(x) \leq l_r \\ x \in S \end{array} \right. \quad (2.1.8)$$

où  $l_r$  est une borne inférieure de  $f_k$ . Ce problème est à son tour équivalent au problème obtenu en transformant le critère  $k - 1$  en contrainte et ainsi de suite jusqu'à arriver à la résolution du programme linéaire en nombre entiers mono-objectif suivant :

$$\left\{ \begin{array}{l} \min \quad f_1(x) + \sum_{i=2}^k \omega_i f_i(x) \\ f_i(x) \leq l_i \quad i = 2, \dots, k \\ x \in S \end{array} \right. \quad (2.1.9)$$

où les poids  $\omega_i$  sont calculés en fonction des  $\overline{f_i}$  et  $\underline{f_i}$ , bornes supérieure et inférieure de  $f_i$  respectivement, de sorte que la solution optimale de (2.19) soit efficace pour le problème (P) et induit donc, une solution non dominée.

### Méthode de deux phases [37]

La méthode deux-phases a initialement été proposée par Ulungu et Teghem pour la résolution d'un problème d'affectation bi-objectif. Comme son nom l'indique, cette méthode est décomposée en deux étapes :

- Trouver toutes les solutions supportées du front Pareto.
- Chercher entre ces solutions les solutions Pareto non supportées.

L'objectif de la première phase est d'obtenir l'ensemble des solutions Pareto supportées. Comme nous l'avons vu précédemment, ces solutions ont l'avantage d'être relativement faciles à trouver puisqu'elles optimisent une certaine combinaison linéaire des objectifs. Ainsi, durant la première phase de la méthode, les deux solutions extrêmes (solutions optimisant chacun un des deux objectifs) sont recherchées (voir figure 2.6.a). Puis, de façon récursive, dès que deux solutions supportées  $r$  et  $s$  sont trouvées, la méthode recherche

d'éventuelles autres solutions supportées entre  $r$  et  $s$ , à l'aide de combinaisons linéaires bien choisies des objectifs (voir figure 2.6.b et 2.6.c). A la fin de la première phase l'ensemble des solutions supportées est donc trouvé (voir figure 2.6.d). Cette première phase rappelle la méthode dichotomique, mais ici seules les solutions supportées sont recherchées. Pour cela, lors de l'exploration entre deux solutions, on s'autorise à retrouver l'une de ces deux solutions, lorsqu'il n'existe pas d'autres solutions supportées dans l'intervalle.

La deuxième phase consiste alors en la recherche des solutions non supportées appartenant au front Pareto. Ces solutions ne peuvent être obtenues par combinaisons d'objectifs. Ulungu et Teghem proposent alors d'utiliser les solutions supportées trouvées pour réduire l'espace de recherche en argumentant que les solutions Pareto non supportées restantes sont forcément dans les triangles rectangles basés sur deux solutions supportées consécutives (voir figure 2.6.e). Ainsi, une recherche de type deuxième phase est exécutée entre chaque couple de solutions supportées adjacentes (voir figure 2.6.f et 2.6.g). La méthode de recherche au sein de ces triangles dépend du problème étudié. A la fin de la deuxième phase, toutes les solutions Pareto sont trouvées. Notons, qu'il aura été nécessaire au préalable de préciser si l'on recherche le front minimal ou maximal complet.

La méthode deux-phases présente un schéma de résolution exacte très intéressant car très général et qui ne dépend pas du problème. Son intérêt réside dans une décomposition de l'espace de recherche et l'utilisation de méthodes mono-objectif pour les différentes résolutions successives (recherche des points extrêmes, résolution par agrégation...). Appliquer la méthode deux-phases pour la résolution d'un problème bi-objectif nécessite donc d'avoir une méthode mono-objectif efficace (si possible polynomiale). C'est le cas pour les problèmes traités par Ulungu et Teghem, et c'est ce qui rend leur méthode performante pour ces problèmes là. Cependant, lorsque la restriction du problème à un seul objectif génère un problème déjà NP-difficile, la non existence de méthode exacte efficace pouvant optimiser chaque objectif séparément peut compromettre l'intérêt de la méthode et notamment rendre la première phase très coûteuse.

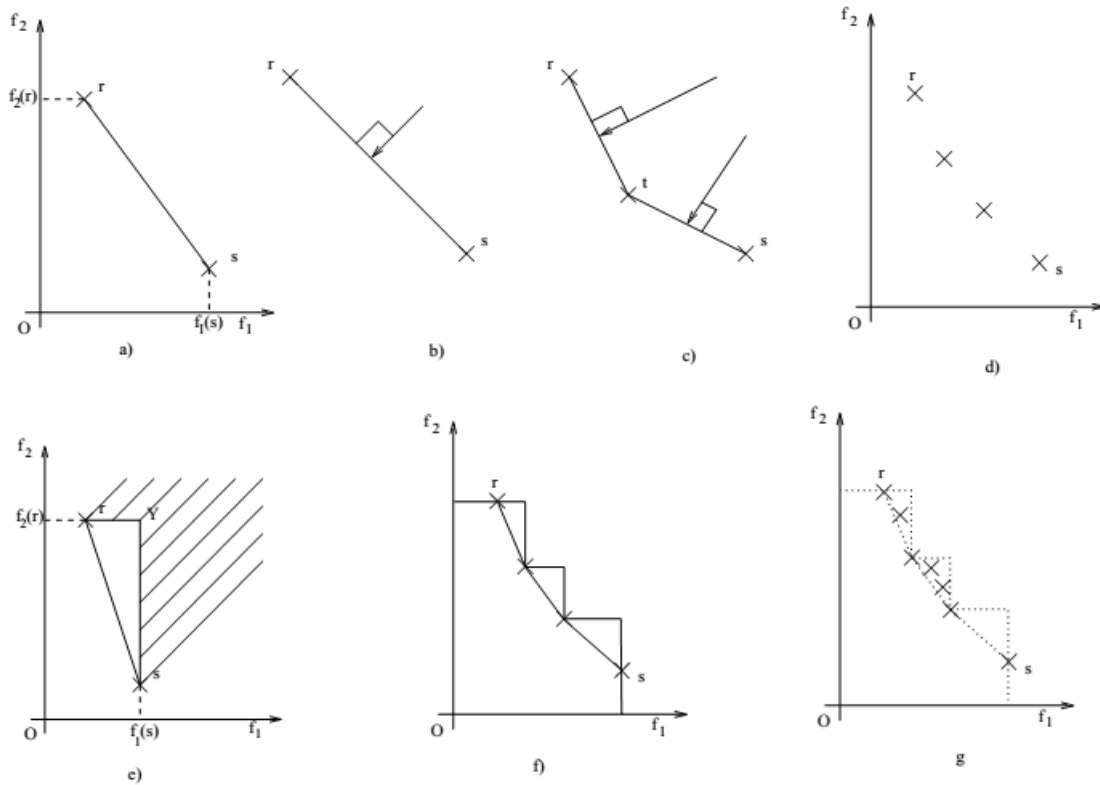


FIGURE 2.6 – Illustration des différentes étapes de la méthode deux phases.

### Méthode de Abbas, Chergui et Ait Mehdi [1]

L'approche [1] génère toutes les solutions entières non dominées sans passer en revue toutes les solutions possibles de  $Y$ . C'est une méthode branch and bound qui fait appel aux coupes efficaces pour passer d'un vecteur entier à un autre dans  $Y$ . Elle fait appel à la méthode du simplexe pour résoudre le programme linéaire  $(P_l)$  à l'étape  $l$  de la méthode. Les vecteurs critères sont alors adjoints au tableau du simplexe et évoluent de façon dynamique dans ce dernier de la même façon que le critère  $z$  de  $(P_l)$ . Si la solution optimale obtenue n'est pas entière, ce qui correspond à un noeud de type 1 de l'arbre, seul un processus de branchement est effectué pour détecter une solution entière (contrairement aux autres méthodes existantes, la méthode n'a pas besoin de rechercher une solution entière optimale, mais seulement une solution entière voisine à la solution optimale courante qui n'est pas entière). Quand une solution entière est obtenue, le noeud est de type 2, le vecteur critère correspondant est comparé à ceux déjà trouvés et l'ensemble

des solutions potentiellement non dominées est mis à jour. En vue de rechercher une autre solution entière, les directions d'amélioration des critères sont utilisées pour construire des coupes efficaces permettant d'éviter l'exploration de domaines ne contenant que des solutions entières dominées.

### Notations

- $x^l$  : Une solution du problème  $(P_l)$ .
- $B^l$  : Ensemble des indices des variables de base de  $x^l$ .
- $N^l$  : Ensemble des indices des variables hors base de  $x^l$ .
- $\tilde{c}^i$  : Vecteur critère réduit relatif au critère  $i, i = \{1, \dots, k\}$ .
- $\tilde{c}_j^i$  : La  $j^{ime}$  composante du coût réduit du vecteur critère  $i$ .
- $H_l^i$  : L'ensemble définit comme suit :

$$H_l^i = \{j \in N^l \mid \tilde{c}_j^i > 0\}$$

- $K_l$  :

$$K_l = \{i \in \{1, \dots, k\} \mid H_l^i \neq \emptyset\}$$

est l'ensemble des critères pouvant être améliorés à partir de  $x^l$ .

- Pour chaque  $i \in K_l$ , on définit la coupe efficace suivante :

$$\tilde{c}^i x \geq \tilde{c}^i x^l + \sum_{j \in N^l \setminus H_l^i} [\hat{c}_{j_0}^i] x_j + \max\{1, [\tilde{c}_j^i]\}. \quad (2.1.10)$$

où :  $\hat{c}_{j_0}^i = \min_{j \in H_l^i} \{\tilde{c}_j^i\}$  pour  $i \in K_l$ .

**Algorithme 1.**  $SND = \emptyset, i = 0, pg = \{(P_0)\}$ .

1. Tantque  $pg \neq \emptyset$ 
  - Choisir un programme  $(P_l)$  de  $pg$  et  $pg = pg/(P_l)$ .
  - Résoudre  $(P_l)$ .
  - Si  $(P_l)$  est irréalisable,  $i := i + 1$ .
  - Sinon, soit  $x^l$  la solution trouvée, aller à 2

Fin tantque
2. – Si  $x^l$  est entière.



- Si  $Cx^l$  n'est pas dominée par toute solution de  $SND$  alors  $SND := SND \cup \{Cx^l\}$ .
- Si  $Cx^l$  domine  $Cy$ , une solution de  $SND$ , alors  $SND := SND \setminus \{Cy\} \cup \{Cx^l\}$ .
- Déterminer :  $N^l$ ,  $H_l^j$  pour chaque vecteur critère  $j$  et  $K_l$ .
- Si  $|K_l| = \emptyset$ , aller à 1
- Sinon,  $|K_l|$  nouveaux programmes ajoutée au pg :
 
$$\left\{ \begin{array}{l} (P_l) \\ \text{la contrainte 2.1.10} \end{array} \right.$$
 aller à 1
- Sinon, deux nouveaux programmes ajoutée au pg :
 
$$\left\{ \begin{array}{l} (P_l) \\ x \leq \lfloor x_j \rfloor \end{array} \right. \quad \left\{ \begin{array}{l} (P_l) \\ x \geq \lceil x_j \rceil \end{array} \right.$$
 aller à 1

## 2.2 Programmation multi-objectif discrète fractionnaire

Le problème de la programmation multi-objectif discrète fractionnaire (*MOILFP*) peut être formulé comme suit :

$$(P) \left\{ \begin{array}{l} \max Z_1 = \frac{c^1 x + \alpha^1}{d^1 x + \beta^1} \\ \max Z_2 = \frac{c^2 x + \alpha^2}{d^2 x + \beta^2} \\ \vdots \\ \max Z_k = \frac{c^k x + \alpha^k}{d^k x + \beta^k} \\ Ax \leq b \\ x \in \mathbb{N}^n \end{array} \right. \quad (2.2.1)$$

où  $k \geq 2$ ;  $c^i, d^i$  sont des  $1 \times n$ -vecteurs;  $\alpha^i, \beta^i$  sont des scalaires pour tout  $i \in \{1, 2, \dots, k\}$ ;

$A$  est une  $m \times n$ -matrice réelle et  $b \in \mathbb{R}^m$ .

### 2.2.1 Méthode de Abbas and Moulaï [2]

La méthode adoptée par Abbas and Moulaï [2] pour résoudre le problème ( $P$ ) est une généralisation du cas fractionnaire de l'algorithme de résolution d'un programme linéaire en nombres entiers à objectifs multiples détaillé dans [2].

Afin de résoudre le problème ( $P$ ), une approche consistant à résoudre une séquence finie de problèmes de programmation fractionnaire linéaire discrète est présentée.

Considérons le problème de programmation fractionnaire linéaire en nombres entiers mono-objectif suivant :

$$(P_1) \begin{cases} \max Z_1 = \frac{c^1 x + \alpha^1}{d^1 x + \beta^1} \\ Ax \leq b \\ x \in \mathbb{N}^n \end{cases} \quad (2.2.2)$$

La recherche des solutions réalisables de ( $P_1$ ) nécessite l'introduction des notations suivantes :

- $S_k = \{x \in \mathbb{R}^n, A_k x \leq b_k, x \geq 0\}$  est la région tronquée courante de  $S$  obtenue par application de la coupe  $\sum_{j \in N_{k-1} \setminus \{j_{k-1}\}} x_j \geq 1$  où :  $j_{k-1} \in \Gamma_{k-1}$  et par la coupe de Gomory successives.
- $x_k^1$  est la  $k^{me}$  solution optimale entière du problème ( $P_1$ ) obtenue sur  $S_k$  à l'étape  $k$ .
- $B_k^1$  est une base de  $S_k$ .
- $a_{k,j}^1$  sont les vecteurs activités de appropriées à la région tronquée courante  $S_k$ .
- $y_{k,j}^1 = (B_k^1)^{-1} a_{k,j}^1, I_k = \{j/a_{k,j}^1 \in B_k^1\}$  et  $N_k = \{j/a_{k,j}^1 \notin B_k^1\}$ .
- $\bar{\gamma}_{k,j}^1$  est le coût réduit relatif à la  $j^{me}$  composante du vecteur gradient réduit  $\bar{\gamma}_k^1$ .
- $\Gamma_k = \{j \in N_k / \bar{\gamma}_{k,j}^1 = 0\}$

**Théorème 2.4.** [2] *Le point  $x_k^1$  de  $S$  est une solution optimale du problème fractionnaire ( $P_1$ ) si et seulement si, le vecteur gradient réduit  $\bar{\gamma}_k^1$  est tel que  $\bar{\gamma}_{k,j}^1 \leq 0$  pour tout indice  $j_k \in N_k$ .*

**Définition 2.7.** Soit  $x_k^1$  une solution optimale de ( $P_1$ ). Supposons que  $j_k \in N_k$ , une arête  $E_{jk}$  incidente à la solution  $x_k^1$  est définie par l'ensemble :

$$E_{jk} = \{x = (x_i) \in S_k / x_i = x_{k,i}^1 - \theta_{jk} y_{k,i}^1 \forall i \in I_k, x_{jk} = \theta_{jk}, x_v = 0 \forall v \in N_k \setminus \{j_k\}\} \text{ où}$$

$$0 \leq \theta_{jk} \leq \min_{i \in I^k} \left\{ \frac{x_k^1}{y_{k,j_k}^1} / y_{k,j_k}^1 > 0 \right\}$$

**Algorithme 2.**

Pour résoudre le problème fractionnaire linéaire en nombres entiers à objectifs multiples, une procédure basée sur une technique de coupes planes est présentée dans les étapes suivantes :

**Etape 1 :**

Résoudre le problème  $(P_1)$  par n'importe quelle méthode directe de la programmation fractionnaire linéaire en nombres entiers.

Soit  $x_1^1$  la solution optimale entière sur  $S_1$ , construire l'ensemble  $\Gamma_1$  .

**Etape 2 :**

– Si  $\bar{\gamma}_{1,j}^1 < 0$  pour  $j \in N_1$  est l'unique solution optimale sur  $\Gamma_1$ , enregistrer le premier vecteur non dominé pour construire l'ensemble des vecteur non dominés  $Eff_0$ .

Tronquer le point  $x_1^1$  par la coupe de Dantzig  $\sum_{j \in N_1 \setminus \{j_1\}} x_j \geq 1$ . Par l'application de la méthode duale fractionnaire relative à la programmation fractionnaire, on obtient une solution réalisable entière dans la région tronquer  $S_2$ . Rajouter le vecteur correspondant à  $Eff_0$  s'il n'est pas dominé par l'un des précédents vecteurs critères non dominés. Enregistrer l'ensemble efficace  $Eff_1$ .

– Sinon, il existe un indice  $j_1 \in N_1$  pour lequel  $\bar{\gamma}_{1,j}^1 = 0$ . Déterminer dans ce cas toutes les solutions qui lui sont alternatives, éliminer celles qui ne sont pas efficaces et mettre à jour l'ensemble  $Eff_0$  . Appliquer la coupe  $\sum_{j \in N_1 \setminus \{j_1\}} x_j \geq 1$  pour tronquer l'arête  $E_{j_1}$

**Etape  $k > 2$  :**

Choisir un indice  $j_{k-1} \in \Gamma_{k-1}$  et explorer l'arête  $E_{j_{k-1}}$  pour déterminer d'éventuelles solutions entières réalisables alternatives à la solution courante  $x_{k-1}^1$ . Augmenter l'ensemble  $Eff_{k-2}$  par les vecteurs critères non dominés correspondants pour construire  $Eff_{k-1}$ .

Tronquer l'arête  $E_{j_{k-1}}$  par la coupe  $\sum_{j \in N_{k-1} \setminus \{j_{k-1}\}} x_j \geq 1$  pour tronquer l'arête  $E_{j_1}$

**Etape finale :** La procédure s'arrête lorsque la méthode duale du simplexe est irréalisable, indiquant ainsi que la région tronquée courante ne contient aucun point réalisable entier et que l'ensemble des points efficaces est complètement déterminé.

### 2.2.2 Méthode de Chergui et Moulaï [9]

L'approche dans cette méthode pour générer toutes les solutions efficaces du problème  $(P)$  est basée sur la résolution du programme fractionnaire linéaire  $(P_l)$  suivant à chaque étape  $l$  de l'algorithme :

$$(P_l) \begin{cases} \max Z_1 = \frac{c_1^t x + \alpha^1}{d_1^t x + \beta^1} \\ Ax \leq b \\ x \in \mathbb{N}^n \end{cases}$$

Soit  $x_l^*$  la solution entière obtenue après la résolution du problème  $(P_l)$ , on note  $B_l$  l'ensemble des indices des variables de base et  $N_l$  l'ensemble des indices de variables hors base de  $x_l^*$ . Soit  $\bar{\gamma}_j^i$  la  $j^{\text{me}}$  composante du vecteur gradient réduit  $\bar{\gamma}^i$  défini pour chaque critère  $i, i = 1, \dots, K$  par :

$$\bar{\gamma}^i = \bar{d}_1^0 \bar{c}_1 - \bar{c}_1^0 \bar{d}_1$$

où  $\bar{c}_1, \bar{d}_1, \bar{c}_1^0$  et  $\bar{d}_1^0$  sont les mises à jour de  $c_1, d_1, c_1^0$  et  $d_1^0$  respectivement.

On définit l'ensemble  $H_l$  en  $x_l^*$  par :

$$H_l = \{j \in N_l / \exists i \in \{1, \dots, K\} : \bar{\gamma}_j^i > 0\} \cup \{j \in N_l : \bar{\gamma}_j^i = 0, \forall i \in \{1, \dots, K\}\}$$

Ainsi, l'ensemble  $S_{l+1}$  est défini par :  $S_{l+1} = \{x \in S_l \mid \sum_{j \in H_l} x_j \geq 1\}$

L'algorithme de génération de toutes les solutions entières efficaces du programme  $(P)$  est présenté dans les étapes suivantes :

#### Algorithme 3.

**Etape 0 :** Initialisation

$l = 0$  ;

Créer le premier nœud avec le programme  $(P_0)$  ;

$Eff = \emptyset$  (ensemble des solutions entières efficaces du programme  $(P)$ )

**Etape 1 :** Etape générale

Tant qu'il existe un nœud non encore sondé, faire :

Choisir le nœud de plus grand numéro  $l$  non encore sondé ;

Résoudre le programme fractionnaire linéaire correspondant  $(P_l)$  par la méthode duale du simplexe et la méthode de Cambini and Martein [7]; (Initialement, pour la résolution du programme  $(P_0)$ , seule la méthode de Cambini and Martein [7] est utilisée);

Si le programme  $(P_l)$  n'a pas de solutions réalisables, alors le nœud correspondant est sondé.

Sinon, soit  $\tilde{x}_l$  la solution optimale obtenue.

Si  $\tilde{x}_l$  n'est pas entière, aller à l'étape 1.a, sinon aller à l'étape 1.b.

**Etape 1.a : Séparation**

Choisir une coordonnée non entière  $\tilde{x}_{lj}$  de  $\tilde{x}_l$  et séparer le nœud  $l$  actuel en deux nœuds  $k, k \geq l+1$  et  $h, h \geq l+1, h \neq k$ , dans le tableau courant du simplexe, la contrainte  $x_j \leq \lfloor \tilde{x}_{lj} \rfloor$  est rajoutée et un nouveau domaine est considéré au nœud  $k$  et de façon similaire, la contrainte  $x_j \geq \lfloor \tilde{x}_{lj} \rfloor + 1$  est rajoutée pour obtenir un autre domaine au nœud  $h$ . Aller à l'étape 1.

**Etape 1.b : Mise à jour de l'ensemble  $Eff$**

Si le vecteur  $Z(\tilde{x}_l)$  n'est pas dominé par le vecteur  $Z(x)$  pour toute solution alors  $x \in Eff$ , alors  $Eff := Eff \cup \{\tilde{x}_l\}$ . S'il existe  $x \in Eff$  tel que  $Z(\tilde{x}_l)$  domine  $Z(x)$ , alors  $Eff := Eff \setminus \{x\} \cup \{\tilde{x}_l\}$ . Déterminer les ensembles  $N_l$  et  $H_l$ ; Si  $H_l = \emptyset$  alors le nœud  $l$  correspondant est sondé. Aller à l'étape 1; Sinon, rajouter la coupe efficace  $\sum_{j \in H_l} x_j \geq 1$  dans le programme  $(P_l)$  Aller à l'étape 1.

### 2.2.3 Méthode de Ait Mehdi, Chergui et Abbas [3]

Cette méthode a été proposée par Ait Mehdi, Chergui et Abbas pour la détermination de toutes les solutions efficaces du problème de programmation multi-objectif fractionnaire linéaire en nombres entiers. Cette méthode est une amélioration à celle de Chergui et Moulaï [9] citée précédemment, où les auteurs ont proposés un autre test pour réduire le nombre de branches. Cette méthode utilise les mêmes paramètres que ceux définis dans la méthode précédente.

Notons par  $Id$  et  $Nd$  : le point idéal et nadir du problème  $MOILFP$ , où :

$$Id = (Id_1, Id_2, \dots, Id_k); Id_i = \max_{x \in D} Z_i(x), \text{ où } D = \{x \in \mathbb{Z}^n \mid Ax \leq b, x \geq 0\}$$

$$Nd = (Nd_1, Nd_2, \dots, Nd_k); Nd_i = \min_{x \in Eff} Z_i(x)$$

**Algorithme 4.**

**Etape 0 :** Initialisation

$$Eff = \emptyset, L = \{(P_0)\}.$$

Determiner le point nadir  $Nd$  pour MOILFP

**Etape 1 :**

- Si  $L$  est vide, alors l'ensemble  $Eff$  est efficace, et terminer.
- Sinon, sectionner le programme linéaire avec le plus grand indice  $l$  dans  $L$ . Résoudre le problème  $(P_l)$  en utilisant la méthode du simplex ou le dual du simplex,  $L = L/\{(P_l)\}$  et aller à l'étape 2.

**Etape 2 :**

- Si  $(P_l)$  n'est pas réalisable, aller à l'étape 1.
- Sinon, soit  $x_l$  la solution optimale,  $B_l$  (resp.,  $N_l$ ) l'ensemble des indices des variables de base (resp., variables hors base) de  $x_l$  et  $Z_l$  le vecteur critère correspondant.
- Si  $Z_l$  est non dominé, aller à l'étape 3.
- Sinon, calculer le point idéal  $Id^l$  de  $(MOILFP_l)$  comme suit. L'optimisation individuelle de chaque objectif fractionnaire  $i$  sous  $S_l$  est fait en ajoutant trois lignes au tableau optimal de  $(P_l)$ ; la première et la deuxième lignes correspondent, respectivement, le numérateur et dénominateur de la fonction fractionnaire  $i$  écrite dans la base  $B_l$ . La dernière ligne correspond le vecteur gradient réduit  $\bar{\gamma}_j^i$  de la  $i^{ime}$  objectif. Le pivot du simplex est alors utilisé dans les nouvelles lignes de table, excepter la dernière qui est modifié en utilisant  $\bar{\gamma}^i = \bar{d}_1^0 \bar{c}_1 - \bar{c}_1^0 \bar{d}_1$ , jusqu'à ce que la condition optimale  $\bar{\gamma}^i \leq 0; \forall j \in N_l$  est atteint.

Si  $Id^l$  est dominé par le vecteur critère avec au moins une solution de  $Eff$  ou s'il existe  $i \in \{1..k\}$  tel que  $Id_i^l < Nd_i^l$ , alors sondé le nœd correspondant et aller à l'étape 1.

sinon, aller a l'étape 3.

**Etape 3 :**

- Si  $x^l$  est entière, aller à l'étape 4.
- sinon sélectionner une variable fractionnaire  $x_j^l = \alpha_j$ . Créer et ajouter à  $L$  les deux

problèmes identique à  $(Pl)$  avec la contrainte  $x_j \geq \lfloor \alpha_j \rfloor + 1$  dans le premier,  $x_j \leq \lceil \alpha_j \rceil$  dans le deuxième et aller à l'étape 1.

**Etape 4 :** (Cas général) Mettre à jour  $Eff$  :

S'il n'existe pas un  $x \in Eff$  tel que le vecteur critère correspondant domine  $Z^l$ , alors  $Eff = Eff \cup \{x^l\}$  et supprimer toute les solution pour lesquelles le vecteur critère est dominé par  $Z^l$ .

Déterminer l'ensemble  $H_l$ .

- Si  $H_l$  est vide, aller à l'étape 1.
- Si  $H_l = N_l$  générer les arêtes incidentes  $E_p$ , déterminer toutes les solutions entières se trouvant sur l'arête et mettre à jour  $Eff$ , l'ensemble  $H_l = N_l \setminus \{p\}$ . Ajouter à  $L$  le problème obtenu par l'extension du  $(Pl)$  avec la coupe efficace  $\sum_{j \in H_l} x_j \geq 1$  et aller à l'étape 1.

## Chapitre 3

# Contribution algorithmique en optimisation multi-objectif discrète

Dans notre travail, nous nous sommes penchés tout d'abord sur la programmation linéaire multi-objectif discrète. Une méthode exacte, que nous avons nommé  $Z\epsilon$ , est proposée pour la recherche des solutions non dominées dans le cas où les vecteurs coûts ne sont pas forcément des entiers naturels, contrairement à toutes les méthodes exactes existantes. Puis, une généralisation de la méthode  $Z\epsilon$  est faite pour le problème multi-objectif linéaire fractionnaire.

### 3.1 Génération des solutions non dominées pour le problème multi-objectif linéaire discret

La résolution d'un problème multi-objectif consiste à déterminer soit l'ensemble des solutions efficaces dans l'espace des décisions noté  $Eff$ , soit l'ensemble des solutions non dominées dans l'espace des critères noté  $SND$ .

#### 3.1.1 Définitions et notations

Soit le programme suivant :



$$(P) \begin{cases} \max & z(x) = (c_1x, c_2x, \dots, c_kx) \\ \text{sous} & x \in S \\ & x \text{ vecteur entier} \end{cases} \quad (3.1.1)$$

où  $c_j$  est un vecteur ligne réel pour  $j = 1, \dots, k$ ,  $S = \{x \in \mathbb{R}^n / Ax \leq b, x \geq 0\}$  est le domaine des solutions réalisables du problème relaxé de  $(P)$ ,  $A \in \mathbb{Z}^{n+m}$  et  $b \in \mathbb{Z}^m$ . On suppose que le domaine  $S$  est non vide et compact. On note  $z_i(x) = c_jx$ ,  $j = 1, \dots, k$ .

Soit  $x^*$  une solution réalisable pour le problème  $(P)$ . On définit le programme linéaire mono-objectif suivant :

$$(PE) \begin{cases} \max \varphi = \sum_{i=1}^k v_i \\ c_i x - v_i = c_i x^*, i = 1, \dots, k \\ x \in S, x \text{ vecteur entier} \\ v_i \geq 0, i = 1, \dots, k. \end{cases} \quad (3.1.2)$$

Rappelons les résultats suivants que nous utiliserons dans ce chapitre :

$x^*$  est une solution efficace pour le problème  $(P)$  si et seulement si la valeur optimale de la fonction objectif  $\varphi$  est nulle dans le programme linéaire  $(PE)$  [4].

Si le problème  $(PE)$  a une solution optimale  $(\hat{x}, \hat{v})$  (et la valeur optimale de la fonction objectif est finie), alors  $\hat{x}$  est efficace [4].

### 3.1.2 Principe de la méthode

L'approche proposée  $Z\epsilon$  génère toutes les solutions entières non dominées, qui consiste à optimiser le problème sur la somme des objectifs du programme  $(P)$ . Si la solution optimale obtenue n'est pas entière, seul un processus de branchement est effectué pour détecter une solution entière. Quand une solution entière  $x_0$  est obtenue, le vecteur critère correspondant est comparé à ceux déjà trouvés, si cette solution est incomparable, un test d'efficacité est alors, effectué sur cette solution. Si la solution  $x_0$  est efficace, on rajoute des coupes pour supprimer cette solution et les solutions entières dominées par  $z(x_0)$  contenues dans l'ensemble des solutions réalisables courant. Pour ce faire, des sous-problèmes

$(P_1), (P_2), \dots, (P_k)$  sont créés à partir de  $(P)$  après avoir rajouter les contraintes sur les valeurs des objectifs en  $x_0$  de la forme  $z_j(x) \geq z_j(x_0) + \epsilon_j, \forall j = 1, \dots, k$ ,  $\epsilon_j$  représente une valeur positive plus petite que l'erreur qu'on accepte sur l'objectif. Si  $x_0$  n'est pas efficace, le test d'efficacité de Benson [4] donne une solution efficace  $y$ . Dans ce cas, les problèmes  $(P_j), j = 1, \dots, k$  sont créés par adjonction des contraintes  $z_j(x) \geq z_j(y) + \epsilon_j, \forall j = 1, \dots, k$ .

**Algorithme 5.**

$l := 0, Eff := \emptyset, SND := \emptyset$

Tant que l'arborescence contient, des nœuds non sondé, faire :

**Etape l :** Résoudre le programme linéaire  $(P_l)$

1. si  $(P_l)$  est non réalisable, alors le nœud  $l$  est sondé.
2. sinon, soit  $x^l$  la solution optimale de  $(P_l)$ .
3. si  $x^l$  n'est pas entière, aller à l'étape l1.
4. sinon, aller à l'étape l2.

**Etape l1 :** séparer par rapport à une variable  $x_j^l$  non entière, deux nouveaux nœuds sont créés  $x_j \leq \lfloor x_j^l \rfloor$  et  $x_j \geq \lceil x_j^l \rceil, l := l + 2$ .

**Etape l2 :**

1. si  $z(x^l)$  est dominé par  $z(y)$  pour une solution  $y \in Eff$ , aller à Etape l3.
2. sinon résoudre le programme linéaire  $(PE)$  :

(a) si  $\varphi = 0$ , alors  $x^l$  est efficace,  $Eff := Eff \cup \{x^l\}$ ,  $SND := SND \cup \{z(x^l)\}$  et poser  $y = x^l$ , aller à Etape l3.

(b) sinon, une solution efficace  $y$  trouvée par le programme  $(PE)$  qui est déjà résolu,  $Eff := Eff \cup \{y\}$ ,  $SND := SND \cup \{z(y)\}$ , aller à l'Etape l3.

**Etape l3 :**  $k$  nouveaux nœuds sont créés en ajoutant la contrainte  $z_j(x) \geq z_j(y) + \epsilon_j$ ,  $l := l + k$ .

**Remarque 3.1.** Pour assurer du bon fonctionnement de la méthode (i.e trouver toute les solutions non dominées du problème), la valeur de  $\epsilon$  doit être très petite, sinon une partie seulement de l'ensemble des solutions non dominées est trouvée.

## 3.2 Génération des solutions non dominées pour le problème multi-objectif linéaire fractionnaire discret

Comme pour le problème de programmation multi-objectif linéaire, la résolution du problème multi-objectif linéaire fractionnaire à variables entières *MOILFP* consiste à déterminer soit l'ensemble des solutions efficaces, soit l'ensemble des solutions non dominées.

### 3.2.1 Formulation du problème

Dans cette partie, nous nous intéressons à la résolution du programme multi-objectif fractionnaire à variables discrètes *MOILFP* :

$$(P) \left\{ \begin{array}{l} \max z_1 = \frac{c^1 x + \alpha^1}{d^1 x + \beta^1} \\ \max z_2 = \frac{c^2 x + \alpha^2}{d^2 x + \beta^2} \\ \vdots \\ \max z_k = \frac{c^k x + \alpha^k}{d^k x + \beta^k} \\ x \in S \\ x \in \mathbb{N}^n \end{array} \right. \quad (3.2.1)$$

où  $k \geq 2$ ;  $c^i, d^i$  sont des  $1 \times n$ -vecteurs;  $\alpha^i, \beta^i$  sont des scalaires pour tout  $i \in \{1, 2, \dots, k\}$ ;  $A$  est une  $m \times n$ -matrice entière et  $b$  est un  $m$ -vecteur entier.  $S = \{x \in \mathbb{R}^n \mid Ax \leq b, x \geq 0\}$

Dans toute la suite de cette étude, nous supposons que l'ensemble des solutions réalisables du problème relaxé de  $(P)$  est non vide, compact dans  $\mathbb{R}^n$  et  $d^i x + \beta^i > 0$  sur  $S = \{x \in \mathbb{R}^n \mid Ax \leq b, x \geq 0\}$  pour tout  $i \in \{1, \dots, k\}$ .

#### Test d'efficacité

Soit  $x^*$  une solution réalisable donnée, et soit le programme linéaire mono-objectif suivant :

$$(PEF) \begin{cases} \max \psi = \sum_{i=1}^k \frac{w_i}{d^i x + \beta^i} \\ x \in S, \quad x \text{ veceur entier} \\ (c^i - f_i^* d^i)x - w_i = f_i^* \beta^i - \alpha^i \quad i \in \{1, \dots, k\} \\ w_i \geq 0 \quad i \in \{1, \dots, k\} \end{cases} \quad (3.2.2)$$

$$\text{tel que : } f_i^* = \frac{c^i x^* + \alpha^i}{d^i x^* + \beta^i}, \quad i \in \{1, \dots, k\}$$

Le vecteur  $x^*$  est Pareto optimal pour *MOILFP* si et seulement si la valeur optimale de la fonction objectif  $\psi$  est nulle dans le problème linéaire (*PEF*).

Si le problème (*PEF*) a une solution optimale  $(\hat{x}, \hat{w})$  (et la valeur objectif optimale est finie) alors  $\hat{x}$  est efficace. (voir la démonstration dans le paragraphe 3.3)

Soit  $x^*$  une solution efficace du problème *MOILFP*, On définit la coupe suivante :

$$\frac{c^i x + \alpha^i}{d^i x + \beta^i} \geq \frac{c^i x^* + \alpha^i}{d^i x^* + \beta^i} + \epsilon_i, \quad i \in \{1, \dots, k\} \quad (3.2.3)$$

où  $\epsilon_i$  est une valeur positive plus petite que l'erreur qu'on accepte sur la valeur de  $z_i(x)$ , pour  $i \in \{1, \dots, k\}$ .

$$\text{posons : } l^i = \frac{c^i x^* + \alpha^i}{d^i x^* + \beta^i} + \epsilon_i, \quad i \in \{1, \dots, k\}.$$

$$(3.2.3) \iff \frac{c^i x + \alpha^i}{d^i x + \beta^i} \geq l^i, \quad i \in \{1, \dots, k\} \iff c^i x + \alpha^i \geq l^i (d^i x + \beta^i), \quad i \in \{1, \dots, k\}$$

Alors, la coupe (3.2.3) devient :

$$(c^i - l^i d^i)x \geq l^i \beta^i - \alpha^i, \quad \forall i \in \{1, \dots, k\} \quad (3.2.4)$$

### 3.2.2 Développement de la méthode

La méthode est décrite pour résoudre le problème multi-objectif linéaire fractionnaire dans l'espace des critères. Elle repose sur le principe de la méthode multi-objectif linéaire  $Z_\epsilon$  citée précédemment. L'algorithme proposé consiste à réaliser les étapes suivantes :

**Algorithme 6.****Initialisation**

$Eff = \emptyset, SND = \emptyset, l = 0.$

Soit le programme linéaire mono-objectif suivant :

$$(P_l) \begin{cases} \max c^1 x \\ x \in S \end{cases}$$

**Étape l**

Tant qu'il existe un nœud non sondé dans l'arborescence faire :

1. Résoudre le programme  $(P_l)$ .
2. Si  $(P_l)$  est irréalisable, alors le nœud  $l$  est sondé.
3. Sinon, soit  $x_l^*$  la solution optimale obtenue.
  - (a) Si  $x_l^*$  n'est pas entière aller à l'étape l1.
  - (b) Sinon, aller à l'étape l2.

**Étape l1**

Soit  $x_j^*$  une coordonnée fractionnaire de  $x_l^*$ . Séparer le nœud  $l$  en deux nouveaux nœuds :  
 $x_j \leq \lfloor x_j^* \rfloor$  et  $x_j \leq \lceil x_j^* \rceil$ .  $l := l + 2$ , aller à l'étape l

**Étape l2**

1. Si  $z(x_l^*) = z(y)$  pour une solution  $y \in Eff$ , alors  $x_l^*$  est une solution efficace et aller à l'étape l3
2. Si  $z(x_l^*) < z(y)$  pour une solution  $y \in Eff$ , alors  $z(y)$  domine  $z(x_l^*)$ , poser  $x_l^* = y$  et aller à l'étape l3
3. Si pour toute solution  $y \in Eff$ ,  $z(x_l^*)$  n'est pas dominé par  $z(y)$ , alors résoudre le programme linéaire mono-objectif (PFE)
  - (a) Si  $\psi = 0$ , alors  $x_l^*$  est efficace,  $Eff = Eff \cup \{x_l^*\}$ ,  $SND = SND \cup \{z(x_l^*)\}$ , aller à l'étape l3.
  - (b) Sinon  $x_l^*$  est non efficace, la résolution du programme (PFE) donne une solution  $y$  efficace,  $x_l^* = y$ , aller à l'étape l3.

**Étape l3**

Des problèmes  $(P_1^{l+1}), (P_2^{l+1}), \dots, (P_k^{l+1})$  sont créés à partir de  $(P_l)$  en lui ajoutant les contraintes (3.2.4).

$$l := l + k.$$

**3.3 Résultats théoriques**

Dans ce paragraphe, la justification des étapes des deux algorithmes ci-dessus est présentée.

Soit  $(P)$  un programme linéaire multi-objectif ou un programme multi-objectif linéaire fractionnaire.

Associons à  $x^l$ , solution optimale du problème  $(P_l)$ , les ensembles suivants :

$$S^l = \{x \in \mathbb{R}^n / A^l x \leq b^l\}.$$

C'est l'ensemble des solutions réalisables à l'étape  $l$  de la méthode.

$$E^l = \bigcap_{i=1}^k \{x \in S^l \mid z_i(x) \geq z_i(x^l) + \epsilon_i\} \subset S^l.$$

Cet ensemble est contenu dans le cône de sommet  $x^l$  et engendré par les gradients des  $k$  critères, obtenu par translation de ce dernier.

$$F^l = \bigcup_{i=1}^k \{x \in S^l \mid z_i(x) \geq z_i(x^l) + \epsilon_i\} \subset S^l.$$

où  $\epsilon_i$  est une valeur positive plus petite que l'erreur qu'on accepte sur la valeur de  $z_i(x)$ , pour  $i = 1 \dots k$ .

**Proposition 3.1.** *Soit  $x^l$  une solution efficace de  $(P)$ . Alors  $E^l$  ne contient pas de solutions efficaces.*

**Preuve 4.**

Soit  $w \in E^l$ , alors  $z_i(w) \geq z_i(x^l) + \epsilon_i, \forall i = 1, \dots, k$ , tel que la valeur de  $\epsilon_i$  est aussi petite qu'on veut, alors on a  $z_i(w) > z_i(x^l), \forall i = 1, \dots, k$ . Ainsi, pour tout  $w \in E^l$ , donc dans  $S^l$ ,  $z(w)$  domine  $z(x^l)$ . Ce qui signifie que  $x^l$  n'est pas une solution efficace.

Notons que cette proposition montre que dans l'algorithme 5, *Étape l3*, les  $k$  nouveaux sous problèmes créés sont disjoints et les ensembles des solutions réalisables correspondants constituent une partition de l'ensemble des solutions réalisables du problème à partir duquel ils sont générés.

**Proposition 3.2.** *Toutes les solutions non dominées de (P) se trouvant dans  $z(S^l)$ , appartiennent à l'ensemble  $z(F^l)$ .*

où pour tout  $X \subset S^l$ ,  $z(X)$  est l'image de  $X$  par  $z = (z_1, z_2, \dots, z_k)$ .

**Preuve 5.**

Soit  $y \notin z(F^l)$ , notons  $\bar{F}^l$  le complémentaire de  $F^l$ , alors :

$$\begin{aligned} \exists \bar{x} \in \bar{F}^l \text{ tel que } z(\bar{x}) = y &\implies \bar{x} \in \bigcap_{i=1}^k \{z_i(x) < z_i(x^l) + \epsilon_i\} \\ &\implies \bar{x} \in \{x \in S^l / z_i(x) < z_i(x^l) + \epsilon_i, \forall i\} \\ &\implies \bar{x} \in \{x \in S^l / z_i(x) \leq z_i(x^l)\}, \forall i, \text{ car } \epsilon_i > 0 \\ &\implies \bar{x} \in S^l \text{ avec } z_i(\bar{x}) \leq z_i(x^l), \forall i. \end{aligned}$$

Ainsi, pour toute solution  $\bar{x} \in S^l$  qui n'est pas alternative à la solution  $x^l$ , on a :  $z_i(\bar{x}) \leq z_i(x^l)$ ,  $\forall i$  avec  $z_j(\bar{x}) < z_j(x^l)$ , pour au moins un  $j \in \{1, \dots, k\}$ .

Ce qui signifie que  $z(\bar{x}) = y$  est dominée.

**Proposition 3.3.** *Les algorithmes 5 (§ 3.1.2) et 6 (§ 3.2.2) décrits plus haut convergent en un nombre fini d'étapes.*

**Preuve 6.** *L'ensemble des solutions réalisables  $S$  étant par hypothèse fermé et borné, il contient un nombre fini de solutions entières. De plus, à chaque étape de la méthode, on détermine une solution efficace  $x^l$  de (P) qui permet d'éliminer toutes les solutions  $y$ , telle que  $z(x^l)$  domine  $z(y)$  par adjonction des contraintes  $z_i(x) \geq z_i(x^l) + \epsilon_i$ ,  $\forall i \in \{1, \dots, k\}$ . D'autre part, la proposition 3.2 nous assure qu'aucune solution non dominée n'est ratée lorsqu'on rajoute ces dernières contraintes.*

Notons que cette proposition permet de justifier qu'aux étapes 13 des algorithmes 5 et 6, on ne rate aucune solution non dominée.

**Théorème 3.4.** *Soit  $x^*$  une solution réalisable donnée, et soit le programme fractionnaire linéaire mono-objectif suivant :*

$$(PEF) \begin{cases} \max \psi = \sum_{i=1}^k \frac{w_i}{d^i x + \beta^i} \\ x \in S, \quad x \text{ vecteur entier} \\ (c^i - f_i^* d^i)x - w_i = f_i^* \beta^i - \alpha^i \quad i \in \{1, \dots, k\} \\ w_i \geq 0 \quad i \in \{1, \dots, k\} \end{cases} \quad (3.3.1)$$

$$\text{tel que : } f_i^* = \frac{c^i x^* + \alpha^i}{d^i x^* + \beta^i}.$$

Le vecteur  $x^*$  est Pareto optimal pour MOILFP si et seulement si la valeur optimale de la fonction objectif  $\psi$  est nulle dans le problème linéaire (PEF)

**Preuve 7.** Soit  $(x, w)$  une solution réalisable de problème (PEF). Comme  $w_i \geq 0, i = 1, \dots, k$  et la définition de  $w_i$  est  $(d^i x + \beta^i)(z_i(x) - z_i(x^*))$  nous avons

$$\begin{aligned} \sum_{i=1}^k \frac{w_i}{d^i x + \beta^i} = 0 &\iff \frac{w_i}{d^i x + \beta^i} = 0 \quad i = 1, \dots, k \quad (w_i \geq 0, \frac{1}{d^i x + \beta^i} > 0) \\ &\iff w_i = 0 \quad i \in \{1, \dots, k\} \\ &\iff z_i(x) = z_i(x^*) \quad \{1, \dots, k\} \end{aligned}$$

Ainsi  $z(x^*)$  est non dominée et donc  $x^*$  est efficace.

Si, d'autre part,  $x^*$  est efficace, l'ensemble des solutions réalisables du programme (PEF) se compose de  $(x, w)$  pour chaque  $x \in S$  et  $z_i(x) \geq z_i(x^*), \forall i \in \{1, \dots, k\}$ . Comme  $x^*$  est efficace, alors  $\forall x \in S : z(x) \leq z(x^*)$  et  $z(x) \neq z(x^*)$ . Ainsi  $z(x) = z(x^*)$  et par suite  $w = 0$ , ce qui signifie  $\psi = 0$ .

Le théorème 3.4 permet de justifier l'étape l2 :3.a de l'algorithme 6.

**Proposition 3.5.** Si le problème (PEF) a une solution optimale  $(\hat{x}, \hat{w})$  (et la valeur objectif optimale est finie) alors  $\hat{x}$  est efficace.

**Preuve 8.** Supposons que la solution  $\hat{x}$  n'est pas efficace. Alors il existe une solution  $x' \in S$  tel que  $z_i(\hat{x}) \leq z_i(x'), \forall i = 1, \dots, k$  et  $\exists j \in \{1, \dots, k\}, z_j(\hat{x}) < z_j(x')$ .

Soit  $w' = (d^i x' + \beta^i)(z(x') - z(x^*))$ . Alors  $(x', w')$  est une solution réalisable de (PEF) :

$$\frac{w'_i}{d^i x' + \beta^i} = z_i(x') - z_i(x^*) \geq z_i(\hat{x}) - z_i(x^*) = \frac{\hat{w}_i}{d^i \hat{x} + \beta^i}.$$



En outre,  $\sum_{i=1}^k \frac{w'_i}{d^i x' + \beta^i} > \sum_{i=1}^k \frac{\hat{w}_i}{d^i \hat{x} + \beta^i}$  comme  $\frac{w'_j}{d^j x' + \beta^j} > \frac{\hat{w}_j}{d^j \hat{x} + \beta^j}$  .  
Donc,  $(\hat{x}, \hat{w})$  n'est pas solution optimale de (PEF).

La proposition 3.4 permet de justifier l'étape l2 :3.b de l'algorithme 6.

### 3.4 Exemples illustratifs

#### Exemple 1 :

Considérons le programme linéaire suivant :

$$(P) \begin{cases} \text{Max}(x_1 - 2/3x_2; -1/2x_1 + x_2) \\ x_1 + x_2 \leq 6 \\ 2x_1 \leq 9 \\ 2x_2 \leq 5 \\ x_1, x_2 \in \mathbb{N} \end{cases} \quad (3.4.1)$$

La résolution de  $(P_0)$  donne le tableau optimal suivant :

| $B$   | $b$   | 3    | 4     |
|-------|-------|------|-------|
| 2     | 3/2   | 1    | -1/2  |
| 1     | 9/2   | 0    | 1/2   |
| 5     | 2     | -2   | 1     |
| $z$   | -11/4 | -1/3 | -1/12 |
| $z_1$ | -7/2  | 2/3  | -1    |
| $z_2$ | 3/4   | -5/6 | 3/4   |

TABLE 3.1

La solution optimale  $(9/2, 3/2)$  est non entière, la séparation se fait donc par rapport à la variable  $x_1$  et on obtient les deux sous problèmes :

$$(P_1) \begin{cases} (P_0) \\ x_1 \leq 4 \end{cases} \quad (P_2) \begin{cases} (P_0) \\ x_1 \geq 5 \end{cases}$$

Le problème  $(P_2)$  étant non réalisable, le nœud correspondant est sondé. La résolution de  $(P_1)$  donne le tableau optimale suivant :

| $B$   | $b$    | 3      | 6      |
|-------|--------|--------|--------|
| 2     | 2      | 1      | -1     |
| 1     | 4      | 0      | 1      |
| 5     | 1      | -2     | 2      |
| 4     | 1      | -2     | -2     |
| $z$   | $-8/3$ | $-1/3$ | $-1/6$ |
| $z_1$ | $-8/3$ | $2/3$  | $-1$   |
| $z_2$ | 0      | $-5/3$ | $3/2$  |

TABLE 3.2

La solution optimale  $x = (4, 2)$  de  $(P_1)$  est entière.  $x$  est efficace car la valeur optimale de la fonction objectif de  $(PE)$  est nulle :

$$(PE) \left\{ \begin{array}{l} Max \varphi = v_1 + v_2 \\ x_1 + x_2 \leq 6 \\ 2x_1 \leq 9 \\ 2x_2 \leq 5 \\ x_1 - 2/3x_2 - v_1 = 8/3 \\ -1/2x_1 + x_2 - v_2 = 0 \\ x_1, x_2 \in \mathbb{N} \\ v_1 \geq 0, v_2 \geq 0 \end{array} \right.$$

$$Eff = \{(4, 2)\}, SND = \{(8/3, 0)\}.$$

Deux nouveaux problèmes  $(P_3)$  et  $(P_4)$  sont créés à partir de  $(P_1)$  :

$$(P_3) \left\{ \begin{array}{l} (P_1) \\ x_1 - 2/3x_2 \geq 8/3 + \epsilon \end{array} \right. \qquad (P_4) \left\{ \begin{array}{l} (P_1) \\ -1/2x_1 + x_2 \geq \epsilon \end{array} \right.$$

On pose :  $\epsilon = 10^{-3}$ .

La résolution de  $(P_3)$  donne :

| $B$   | $b$       | 3    | 6    |
|-------|-----------|------|------|
| 2     | 1333/667  | 3/2  | 3/2  |
| 1     | 4         | 1    | 0    |
| 5     | 334/33    | -3   | -3   |
| 4     | 1         | -2   | 0    |
| 3     | 1/667     | -5/2 | -3/2 |
| $z$   | -1781/668 | -1   | -1/2 |
| $z_1$ | -891/334  | 0    | -1   |
| $z_2$ | 1/667     | 1    | -3/2 |

TABLE 3.3

La solution optimale du tableau 3.3 est non entière. Au problème ( $P_3$ ), la contrainte  $x_2 \geq 2$  est rajoutée, le nœud correspondant est sondé car le problème ( $P_6$ ) est irréalisable et l'ajoute de la contrainte  $x_2 \leq 1$  au problème ( $P_3$ ), donne une solution optimale entière :

| $B$   | $b$     | 6   | 8    |
|-------|---------|-----|------|
| 2     | 1       | 0   | 1    |
| 1     | 4       | 1   | 0    |
| 5     | 3       | 0   | -2   |
| 4     | 1       | -2  | 0    |
| 3     | 1       | -1  | -1   |
| 7     | 221/332 | 1   | -2/3 |
| $z$   | -1/2    | -1  | -1/3 |
| $z_1$ | -10/3   | -1  | 1/2  |
| $z_2$ | 1       | 2/3 | -1   |

TABLE 3.4

La solution (4, 1) est efficace .  $Eff = \{(4, 2), (4, 1)\}$ ,  $SND = \{(8/3, 0), (10/3, -1)\}$ .

Comme la solution obtenue est efficace, deux nouveaux problèmes ( $P_7$ ) et ( $P_8$ ) sont créés :

$$(P_7) \begin{cases} (P_5) \\ x_1 - 2/3x_2 \geq 10/3 + \epsilon \end{cases} \qquad (P_8) \begin{cases} (P_5) \\ -1/2x_1 + x_2 \geq -1 + \epsilon \end{cases}$$

On sélectionne le problème  $(P_7)$ . La résolution de ce dernier donne le tableau optimal suivant :

| $B$   | $b$       | 6    | 9    |
|-------|-----------|------|------|
| 2     | 666/667   | 3/2  | 1    |
| 1     | 4         | 1    | 0    |
| 5     | 1000/333  | -3   | -2   |
| 4     | 1         | -2   | 0    |
| 3     | 668/667   | -5/2 | -1   |
| 7     | 1001/1000 | 0    | -1   |
| 8     | 1/667     | -3/2 | -3/2 |
| $z$   | 1556/667  | -1   | -1/2 |
| $z_1$ | -1107/332 | 0    | -1   |
| $z_2$ | 668/667   | 1    | -3/2 |

TABLE 3.5

La solution optimale de  $(P_7)$  est non entière. On génère deux sous-problèmes  $(P_9)$  et  $(P_{10})$  en ajoutant respectivement au problème  $(P_7)$ , les contraintes suivants :

$$x_2 \leq 0$$

$$x_2 \geq 1$$

Le nœud 10 est stérilisé car le problème  $(P_{10})$  est non réalisable.

la solution optimale de  $(P_9)$  :  $(4, 0)$  est entière, le vecteur critère correspondant  $(4, -2)$  est non-dominé :

| $B$   | $b$     | 6   | 10   |
|-------|---------|-----|------|
| 2     | 0       | 0   | 1    |
| 1     | 4       | 1   | 0    |
| 5     | 5       | 0   | -2   |
| 4     | 1       | -2  | 0    |
| 3     | 2       | -1  | -1   |
| 7     | 5/3     | 1   | -2/3 |
| 8     | 1       | 0   | -1   |
| 9     | 221/332 | 1   | -2/3 |
| $z$   | -2      | -1  | -1/2 |
| $z_1$ | -4      | -1  | 1/2  |
| $z_2$ | 2       | 2/3 | -1   |

TABLE 3.6

$$Eff = \{(4, 2), (4, 1), (4, 0)\}, SND = \{(8/3, 0), (10/3, -1), (4, -2)\}.$$

Deux nouveaux problèmes  $(P_{11})$  et  $(P_{12})$  sont créés à partir de  $(P_9)$  :

$$(P_{11}) \begin{cases} (P_9) \\ x_1 - 2/3x_2 \geq 4 + \epsilon \end{cases} \quad (P_{12}) \begin{cases} (P_9) \\ -1/2x_1 + x_2 \geq -2 + \epsilon \end{cases}$$

Le problème  $(P_{11})$  est irréalisable.

Dans le tableau 3.7 la solution optimale de  $(P_{12})$  est non entière :

| $B$   | $b$        | 10   | 11 |
|-------|------------|------|----|
| 2     | 0          | 1    | 0  |
| 1     | 1999/500   | 2    | 2  |
| 5     | 5          | -2   | 0  |
| 4     | 251/250    | -4   | -4 |
| 3     | 1001/500   | -3   | -2 |
| 7     | 999/500    | 4/3  | 2  |
| 8     | 1          | -1   | 0  |
| 9     | 499/500    | 4/3  | 2  |
| 6     | 1/500      | -2   | -2 |
| $z$   | -1999/1000 | -4/3 | -1 |
| $z_1$ | -1999/500  | -4/3 | 0  |
| $z_2$ | 1999/1000  | -2   | 1  |

TABLE 3.7

La séparation est déclenchée avec la création de deux nœuds. Le nœud 14 est sondé car en rajoutant la contrainte  $x_1 \geq 4$ ; le problème est non réalisable. La contrainte  $x_1 \leq 3$  est rajoutée au tableau 3.7 et la solution entière optimale est obtenue dans le tableau 3.8 suivant :

| $B$   | $b$     | 10   | 12   |
|-------|---------|------|------|
| 2     | 0       | 1    | 0    |
| 1     | 3       | 0    | 1    |
| 5     | 5       | -2   | 0    |
| 4     | 3       | 0    | -2   |
| 3     | 3       | -1   | -1   |
| 7     | 1       | -2/3 | 1    |
| 8     | 1       | -1   | 0    |
| 9     | 0       | -2/3 | 1    |
| 6     | 1       | 0    | -1   |
| 11    | 250/501 | 1    | -1/2 |
| $z$   | -3/2    | -1/3 | -1/2 |
| $z_1$ | -3      | 2/3  | -1   |
| $z_2$ | -3/2    | -1   | 1/2  |

TABLE 3.8

la solution optimale  $(3, 0)$ , ce qui donne  $(z_1, z_2) = (-3, 3/2)$  qui est dominé par  $(z_1, z_2) = (4, 1)$  déjà trouvée, donc l'ensemble  $SND$  reste le même que précédemment. dans ce cas là, deux coupes ajoutées au problème  $(P_{13})$

$$(P_{15}) \begin{cases} (P_{13}) \\ x_1 - 2/3x_2 \geq 4 + \epsilon \end{cases} \quad (P_{16}) \begin{cases} (P_{13}) \\ -1/2x_1 + x_2 \geq 1 + \epsilon \end{cases}$$

Les nœuds 15 et 16 de l'arborescence sont sondés.

Retirer les étapes de l'algorithme jusqu'à tous les nœuds de l'arborescence sont sondés.

l'ensemble des solutions non dominées du problème  $(P)$  est :

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

L'arborescence qui va suivre illustre l'exemple précédent :



page vide

**Exemple 2 :**

Dans le but d'illustrer l'utilisation l'algorithme 6, considérons le programme fractionnaire linéaire en nombres entiers à objectifs multiples suivant :

$$(P) \left\{ \begin{array}{l} \max z_1 = \frac{-5x_1 + 7x_2 + 4}{8x_1 + 10x_2 + 4} \\ \max z_2 = \frac{5x_1 + 8x_2 + 9}{8x_1 + 10x_2 + 4} \\ \max z_3 = \frac{7x_1 + 2x_2 + 6}{8x_1 + 10x_2 + 7} \\ 25x_1 + 23x_2 \leq 53 \\ 22/3x_1 + 25/3x_2 \leq 23 \\ x_1, x_2 \text{ entiers.} \end{array} \right. \quad (3.4.2)$$

On pose :  $\epsilon = 10^{-3}$

Le programme  $(P_0)$  est résolu avec l'objectif  $-5x_1 + 7x_2$ . La solution optimale obtenue  $x = (0, 2)$  dont le vecteur critère est :  $(3/4, 25/24, 10/27)$  est dominée par la solution  $(1, 9/4, 6/7)$ . L'ensemble  $SND$  est à jour  $SND = \{(1, 9/4, 6/7)\}$ . La séparation est déclenchée avec la création de trois nœuds :

- Le nœud 1 : est sondé car le domaine associé n'est pas réalisable en rajoutant la contrainte :  $-13.0080x_1 - 3.0100x_2 \leq 0.0040$  à  $(P_0)$ .
- Le nœud 2 : le problème  $(P_2)$  est créée, en rajoutant la contrainte :  $-13.0080x_1 - 14.5100x_2 \leq 0.0040$ .  $(P_2)$  est non réalisable et par conséquent, le nœud correspondant est sondé.
- Le nœud 3 : la coupe :  $-0.1349x_1 - 6.5814x_2 \leq 0.0070$  est rajoutée à  $(P_0)$  pour obtenir  $(P_3)$ . La solution optimale de  $(P_3)$  est  $(1, 0)$ , ce qui donne  $(z_1, z_2, z_3) = (-1/12, 7/6, 13/15)$  qui est non dominée, alors, l'ensemble  $SND$  est à jour  $SND = \{(1, 9/4, 6/7); (-1/12, 7/6, 13/15)\}$ .

Trois nouveaux problèmes  $(P_4)$ ,  $(P_5)$  et  $(P_6)$  sont créés à partir de  $(P_3)$  :

$$(P_4) \left\{ \begin{array}{l} (P_3) \\ 4.3413x_1 - 7.8233x_2 \geq 4.3293 \end{array} \right. \quad (P_5) \left\{ \begin{array}{l} (P_3) \\ 4.3413x_1 + 3.6767x_2 \geq 4.3293 \end{array} \right.$$

$$(P_6) \left\{ \begin{array}{l} (P_3) \\ -0.0587x_1 + 6.6767x_2 \geq -0.0737 \end{array} \right.$$

les nœud 4 et 5 sont sondés. La solution optimale de  $(P_6)$  :  $(2, 0)$  est comparée avec l'ensemble des solutions efficaces, elle n'est pas dominée, alors le teste d'efficacité est effectué sur cette solution. La solution  $(-3/10, 19/20, 20/23)$  est non dominée alors  $SND = \{(1, 9/4, 6/7); (-1/12, 7/6, 13/15); (-3/10, 19/20, 20/23)\}$ . L'algorithme 5 s'arrête puisque tous les nœuds sont sondés et l'ensemble de toutes les solutions entières efficaces du programme  $(P)$  est  $SND = \{(1, 9/4, 6/7); (-1/12, 7/6, 13/15); (-3/10, 19/20, 20/23)\}$ .

# Conclusion et perspectives

Résoudre un problème d'optimisation multi-objectif, c'est déterminer complètement ou partiellement l'ensemble des solutions non dominées.

Dans ce travail, nous nous sommes intéressés aux problèmes de programmation linéaire multi-objectif et fractionnaire en nombres entiers pour lesquels relativement peu de travaux ont été réalisés. On a présenté quelques définitions et concept de base de la programmation linéaire et fractionnaire mono-objectif, puis, les concepts de l'optimisation multi-objectif et multi-objectif fractionnaire qui ont fait l'objet du noyau central de cette thèse. Une étude bibliographique des méthodes de résolution de problème multi-objectif est menée. Une méthode exacte,  $Z\epsilon$ , de résolution du problème linéaire à objectifs multiples et coefficients réels en variables entière est mise au point dont l'algorithme converge en un nombre fini d'étapes. On a aussi abordé l'étude du problème de la programmation fractionnaire linéaire en nombres entiers MOILFP, et on a réussi à généraliser la méthode  $Z\epsilon$  donnant l'ensemble non dominé au problème MOILFP.

Ces deux méthodes sont exposées, illustrées sur des exemples montrant le déroulement des algorithmes et programmées en utilisant le logiciel Matlab.

On préconise d'aborder les points suivants dans nos travaux futurs :

- Etude comparative de méthodes dédiées au problème MOILFP,
- Amélioration des coupes proposées pour assurer une plus grande performance des méthodes décrites,
- Généralisation de la méthode  $Z\epsilon$  au problème multi-objectif quadratique et fractionnaire quadratique en variables entières.

# Bibliographie

- [1] Abbas M., Chergui M.E-A. and Aït Mehdi M.(2010) Efficient Cuts for Generating the Non dominated Vectors for Multiple Objective Integer Linear Programming, Special Issue on Polyhedra Combinatorial Optimization,IJMOR.
- [2] Abbas M., Moulaï M., Integer Linear Fractional Programming with Multiple Objective, *Ricerca Operativa Journal of the Italian Operations Research Society* 103,104, 15-38, 2002.
- [3] Ait Mehdi M., Chergui M.E-A. and Abbas M.(2014) An Improved Method for Solving Multiobjective Integer Linear Fractional Programming Problem.
- [4] Benson H. P.(1978) Existence of efficient solutions for vector maximization problems, *Journal of Optimization Theory and Appl*,26,569-580.
- [5] Bhatia H.L. Malhotra R. and Puri M.C. Bicriteria assignment problems,. *OP-SEARCH*, 19(2) :84-96, 1982.
- [6] C. A. Coello Coello and D. A. Van Veldhuizen. Evolutionary algorithms for solving multiobjective problems. Springer, New York, NY, USA, 2nd ed. edition, 2007.
- [7] Cambini A.,Martein L., A modified version of Martos' Algorithm, *Methods of operations research*, 53, 33-44, 1986.
- [8] Charnes, A, Cooper, W. W. (1973). An explicit general solution in linear fractional programming, *Naval Research Logistics Quarterly*, 20(3) :449-467.
- [9] Chergui M.E., Moulaï M., An exact method for a discrete multiobjective linear fractional optimization.10, 1155, 2008.
- [10] Dantzig G.B.(1947) Linear programming and extension, Princeton University Press.

- 
- [11] D. Granot and F. Granot, On integer and mixed integer fractional programming problems, *Ann. Discrete Math* 1, pp. 221-231,(1977).
- [12] Ehrgott M., Hamacher H.W., Klamroth k., Nickels., Schobel A. and Weicek M.M. A note on the equivalence of balance points and Pareto solutions in multiple-objective programming, *Journal of Optimization theory and Applications* : Vol.92, nol,209-212,1997.
- [13] E.L.Ulungu and J.Teghem. The two phases méthode : An efficient procedure to solve bi-objective combinatorial optimization problems. *Foundations of computing and Decision Sciences*, 20 :149-165,1995.
- [14] E.L.Ulungu. Optimisation combinatoire multi-critère : Détermination de l'ensemble des solutions efficaces et méthodes interactives. PhD thesis, Université de Mons-Hainault, Faculté des sciences, 1993.
- [15] Fischer R. and Richter K. Solving a multi-objective traveling salesman problem by dynamic programming,. *Math.Operetionsforsch.Statist., Ser.Optimization*,vol.13(2), 1982
- [16] F. Y. Edgeworth. *Mathematical physics*. P. Keagan, London, 1881.
- [17] F.Y. Edgeworth. *Mathematical Physics*. P. Keagan, London, 1881.
- [18] Geoffrion A.M. (1968) Proper efficiency and the theory of vector maximization,*Journal of Mathematical Analysis and Applications*, 22, pp. 618-630.
- [19] G.R. Bitran and G. Novaes, *Linear Programming with a fractional objective function*, *Operations Research* 24, pp. 675-699, (1976).
- [20] Green G.I Lee S.M. and Kim C.S. Multiple criteria model for the location-allocation problem. *Computers and Operations Research*, 8 :1-8, 1981.
- [21] Gupta A. and Warburton A. Approximation methods for multiple criteria travelling salesman problems,. *Lecteurs Notes in Economy and Mathematical Systems* 285, Springer Verlag.
- [22] Haimes Y., Lasdon L. and Wismer D.(1971) On a bicriterion formulation of the problems of integrated system identification and system optimization, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 1, pp.296-297.

- 
- [23] Harold W.Kuhn. The hungarian method for the assignment problem. *Naval Research logistics quarterly*, 2 :83-97, 1955.
- [24] H. Arsham et A. B. khan, A complete algorithm for linear fractional programs, *Comput. Math. Appl.*, 1990, 20, n 7, P. 11-23.
- [25] H. Land and A.G.Doig.(1960) An automatic method for solving discrete programming problems.*Econometrica*,28,497-520.
- [26] J.Munkres. Algorithms for the assignment and transportation problems. *Journal of the society of industrial and applied mathematics*, 5(1) :32-38, March 1957.
- [27] J.R. Isbell and W.H. Marlow, Attribution games, *Naval Res. Logist. Quart.* 3, pp. 71-94, (1956).
- [28] Keller C.P.and Goodchild M.F. The Multiobjective vending problem : a generalization of the traveling salesman problem., *Planning and Design*, 15,1988.
- [29] L. Khachiyan.(1979) A Polynomial Algorithm in Linear Programming,*Soviet Mathematics Doklady*,20,191-194.
- [30] Martos B., *Nonlinear programming, Theory and Methods*, North-Holland, Amsterdam, 1975.
- [31] M. Grunspan, *Fractional programming : A survey*, Technical Report 50, Department of Industrial and Systems Engineering, University of Florida,(1971).
- [32] M. Pirlot M.Visée, J. Teghem and E.L Ulungu. Two-phases method and branch and bound procedures to solve the bi-objective knapsack problem. *Journal of Global Optimization*, 12 :139-155, 1998.
- [33] Nagih A. and Plateau G. Problèmes fractionnaires : Tour d’horizon sur les applications : Tour d’horizon sur les applications et methods de resolution,*RAIRO Oper. Res.* 33, 383-419, 1999 .
- [34] N. Jozefowicz, F. Semet, and E. Talbi. Target aiming Pareto search and its application to the vehicle routing problem with route balancing. *Journal of heuristics*, 13 :455-469, 2007.
- [35] N. Jozefowicz, F. Semet, and E. Talbi. An evolutionary algorithm for the vehicle routing problem with route balancing. *European Journal of Operational Research*,195 :761-769,2009.

- 
- [36] N. Karmarkar.(1984) An new polynomial-time algorithm for linear programming ,*Combinatorica*,4,373-395.
- [37] Özlen et Azizoğlu M.(2009) Multi-objective integer programming : A general approach for generating all non-dominated solutions, *European Journal of Operational Research*, Vol. 199, pp.25-35.
- [38] Pelegrin P. and Fernandez F.R. Determination of efficient points in multiobjective location problems. *Naval Research and Logistics Quarterly*, 35 :695-705, 1988.
- [39] R.E.Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society* 64, pages 275-278, 1958.
- [40] R. E. Steuer. *Multiple criteria optimization : Theory, computation and application.* John Wiley, New York, 1986.
- [41] Rita Malhotra. Bi criteria assignment problem. *OPSEARCH*,19(2), 1982.
- [42] Ross T. and Soland R. A multicriteria approach to the location of public facilities,. *European Journal of Operational Research*, 4 :307-321, 1980.
- [43] Schaible S. Duality in fractional programming : a unified approach, *Operations Research*, 24, No-3, 452-461, 1976.
- [44] Schaible S. Fractional programming : applications and algorithms, *European Journal of Operational Research* 7, 111-120, 1981.
- [45] Schaible S. Fractional programming : *Zeitschrift fur Operational Research*, 27, 39-54, 1983.
- [46] Schaible S. Fractional programming, in R. Horst and P.M. Pardalos (eds.), *Handbook of Global Optimization*, Kluwer Academic Publishers, 495-608, 1995.
- [47] Stancu-Minasian I.M. A fifth bibliography of fractional programming, *Optimization* 45,1-4, 343-367, 1999.
- [48] Stancu-Minasian I.M. A sixth bibliography of fractional programming, *Optimization* 55, 4, 405-428, 2006.
- [49] Stancu-Minasian I.M. *Fractional Programming : Theory, Methods and Applications*, Kluwer.



- 
- [50] V. Barichard and J.K. Hao. Genetic tabu search for the multi-objective knapsack problem. *Tsinghua science and Technology*, 8 :8-13,2003.
- [51] V. Chankong and Y.Y. Haimes Multiobjective decision making : theory and methodology, North Holland Series in System Science and Engineering 8, p.406,(1983).
- [52] V.Pareto.(1896) *Cours d'economie politique*.Rouge, Lausanne.
- [53] W. Dinkelbach, On nonlinear fractional programming, *Management Science* 13,pp. 492-498, (1967).