

N° d'ordre : 41/2008-M/MT

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



## Mémoire

Présenté pour l'obtention du diplôme de MAGISTER

En : Mathématiques

Spécialité : Recherche Opérationnelle

Option : Mathématique de gestion

Par : OUAÏL Fatma Zohra

## SUJET

# Optimisation Vectorielle Discrète

Soutenu le 01 juillet 2008, devant le jury composé de :

Mlle. BOUCHEMAKH Isma, USTHB	Professeur,	Président
Mr. MOULAÏ Mustapha, USTHB	Maître de conférences,	Directeur de mémoire
Mr. BOUROUBI Sadek, USTHB	Maître de conférences,	Examineur
Mr. CHAABANE Djamal, USTHB	Maître de conférences,	Examineur
Mr. CHERGUI Mohamed El Amine, USTHB	Chargé de Cours,	Invité



À mes

grands-parents,

À mes parents, À mes

frères, À mes oncles et tantes,

À tous ceux qui m'ont té-

moigné leur soutien, je

dédie le présent

mémoire.



# Remerciements

*Louange à Dieu, Clément et  
Miséricordieux, sans lui rien de tout  
cela n'aurait pu être*

J'adresse mes remerciements à Monsieur Moulaï Mustapha, mon directeur de mémoire, pour ses conseils, pour la confiance et la liberté qu'il m'a accordées durant la réalisation de ce mémoire ainsi que pour toute la documentation qu'il a mis à ma disposition.

Je tiens ensuite à exprimer ma très sincère reconnaissance envers Monsieur Chergui Mohammed El Amine, pour ses qualités scientifiques, pédagogiques mais surtout humaines. Merci d'avoir guidé mes premiers pas dans le chemin de la recherche.

Je ne manquerais pas non plus de dire un grand merci aux membres du jury qui ont accepté, sans réserve aucune, d'évaluer ce mémoire à sa juste valeur, et de me faire part de leur remarques sûrement pertinentes qui, contribueront, sans nul doute, au perfectionnement du présent travail.

Le travail de mémoire est un voyage intense de réflexions, d'apprentissages et de découvertes qui ne peuvent pas aboutir de façon isolée. J'ai eu pour ma part la satisfaction de pouvoir interagir avec Melle Aït Mehdi Meriem et à qui j'exprime ma gratitude.

Je ne terminerai pas sans adresser un immense merci à mes parents et grands-parents, pour tout ce qu'ils ont fait et pour le soutien qu'ils m'ont apporté durant toutes mes études, et à qui je dois tout.

## Table des matières

<b>Introduction générale</b>	<b>1</b>
<b>1 Introduction à l'optimisation multiobjectif discrète</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.2 Programmation linéaire en variables continues ([28], [31], [33]) . . . . .	3
1.2.1 Forme générale d'un programme linéaire . . . . .	4
1.2.2 Forme canonique et forme standard . . . . .	4
1.2.3 Bases, bases réalisables, solutions de base . . . . .	5
1.2.4 Caractérisation des bases optimales . . . . .	6
1.2.5 Algorithme du simplexe . . . . .	7
1.2.6 Dualité . . . . .	8
1.3 Programmation linéaire en nombres entiers ([28], [31], [36]) . . . . .	9
1.3.1 Formulation mathématique . . . . .	9
1.3.2 Coupe fractionnaire de Gomory [15] . . . . .	10
1.3.3 Méthode par séparation et évaluation . . . . .	13
1.4 Les problèmes d'optimisation non linéaires ([24],[25]) . . . . .	14
1.4.1 Convexité . . . . .	15
1.4.2 Conditions de Kuhn et Tucker . . . . .	16
1.5 Programmation multiobjectif discrète ([8], [34], [37]) . . . . .	17
1.5.1 Formulation Générale . . . . .	17

---

1.5.2	Concepts de base . . . . .	18
1.5.3	Cônes . . . . .	22
1.5.4	Détection graphique de l'efficacité . . . . .	23
1.5.5	Illustration des définitions . . . . .	25
1.6	Conclusion . . . . .	26
<b>2</b>	<b>Quelques méthodes appliquées en optimisation multiobjectif discrète</b>	<b>27</b>
2.1	Introduction . . . . .	27
2.2	Méthode de D. Klein & E. Hannan [21] . . . . .	27
2.3	Méthode de R. Gupta & R. Malhotra [18] (2 <sup>ème</sup> procédure) . . . . .	32
2.4	Méthode de A. Crema & J. Sylva [35] . . . . .	36
2.5	Méthode de R. Gupta & R. Malhotra [18] (1 <sup>ère</sup> procédure) . . . . .	40
2.6	Méthode de M. Abbas & M. Moulai [1] . . . . .	46
2.7	Méthode de M. Abbas & D. Chaabane [2] . . . . .	48
2.8	Une méthode de détermination de toutes les solutions alternatives . . . . .	50
2.9	Conclusion . . . . .	52
<b>3</b>	<b>Une méthode d'optimisation multiobjectif fractionnaire discrète</b>	<b>53</b>
3.1	Introduction . . . . .	53
3.2	Programmation fractionnaire linéaire mono objectif . . . . .	54
3.2.1	Formulation mathématique . . . . .	54
3.2.2	Résolution des programmes fractionnaires linéaires mono objectif . . . . .	56
3.2.3	Méthode de A. Cambini <i>et al.</i> [6] . . . . .	57
3.2.4	Méthode de D. Granot <i>et al.</i> [16] . . . . .	59
3.3	Une méthode d'optimisation multiobjectif fractionnaire discrète . . . . .	60
3.3.1	Définitions et notations . . . . .	60
3.3.2	Résultats théoriques . . . . .	63
3.3.3	Développement de la méthode . . . . .	66
3.3.4	Exemple illustratif . . . . .	69
3.4	Conclusion . . . . .	76

---

<b>4</b>	<b>Expérimentation et résultats</b>	<b>77</b>
4.1	Introduction . . . . .	77
4.2	Génération aléatoire des données . . . . .	77
4.3	Résultats pour les méthodes de programmation linéaire multiobjectif discrète . . . . .	79
4.3.1	Méthode de M. Abbas & M. Moulai [1] . . . . .	79
4.3.2	Méthode de M. Abbas & D. Chaabane [2] . . . . .	80
4.3.3	R. Gupta & R. Malhotra (2 <sup>ème</sup> procédure) [18] . . . . .	82
4.3.4	Méthode de A. Crema & J. Sylva [35] . . . . .	84
4.4	Résultats pour la méthode d'optimisation multiobjectif fractionnaire discrète . . . . .	87
4.5	Conclusion . . . . .	89
	<b>Conclusion générale</b>	<b>90</b>
	<b>Bibliographie</b>	<b>92</b>

## Introduction générale

La recherche opérationnelle s'est principalement constituée, jusqu'à assez récemment du moins, à partir des modèles qui postulent l'existence d'une fonction objectif (d'un critère) unique. On admettait ainsi implicitement que pour aider les entreprises à mieux décider, il y avait une règle générale, une fonction objectif qui s'imposait aux yeux de tous pour caractériser la bonne direction dans laquelle il convenait de faire évoluer le système auquel on s'intéressait (B. Roy [29]). Cependant, nombreuses sont les situations concrètes où cette modélisation ne reflète pas la réalité car, souvent il y a lieu de considérer plusieurs objectifs contradictoires.

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

Dans ce mémoire, la programmation mathématique multiobjectif en nombres entiers représente le cadre général de travail. L'intérêt de tels problèmes provient du fait qu'il existe un large éventail d'applications concrètes dont l'introduction des variables discrètes est obligatoire.

Notre premier objectif fut de réaliser une étude détaillée sur la programmation linéaire multiobjectif en nombres entiers en passant en revue l'important de la littérature existante ([1], [2], [35], [18], [21]).

Ensuite notre attention s'est focalisée sur la programmation multiobjectif fractionnaire linéaire discrète. En essayant de surmonter les difficultés liées à la non linéarité des fonctions objectifs, nous avons mis au point une méthode pour la détermination de toutes les solutions efficaces du problème de la programmation multiobjectif fractionnaire linéaire en nombres entiers (MOILFP).

L'organisation de ce mémoire est la suivante :

Le chapitre 1 expose le cadre général de notre travail. Dans un premier temps, les principaux résultats concernant la programmation linéaire (en variables continues et en variables discrètes) sont décrits, notamment les méthodes de résolution classiques (méthodes du simplexe, duale du simplexe, coupes de Gomory et branch & bound). Ensuite, l'essentiel des définitions et résultats liés à l'optimisation multiobjectif en nombres entiers est présenté.

Le chapitre 2 détaille quelques méthodes d'optimisation multiobjectif discrète existantes dans la littérature. Au cours de ce chapitre, un contre exemple est fourni montrant une faille dans la deuxième méthode proposée par R. Gupta & R. Malhotra [18] et à laquelle nous proposons des corrections. Finalement, une méthode de détermination de toutes les solutions alternatives d'un problème de programmation linéaire en nombres entiers est proposée.

Le chapitre 3 aborde la programmation fractionnaire linéaire mono objectif et multiobjectif. Une méthode exacte est proposée pour la détermination de toutes les solutions efficaces du problème MOILFP.

Le chapitre 4 finalise notre travail par une implémentation des différentes méthodes traitées dans ce mémoire en présentant une synthèse des caractéristiques principales de ces méthodes et de leurs difficultés de mise en œuvre.

## Introduction à l'optimisation multiobjectif discrète

### 1.1 Introduction

Les problèmes d'optimisation combinatoires issus de la réalité sont la plupart du temps de nature multiobjectif car, plusieurs critères souvent contradictoires sont à considérer simultanément. Contrairement à l'optimisation mono objectif, la solution d'un problème multiobjectif n'est pas unique, mais un ensemble de solutions, connu comme l'ensemble des solutions efficaces. Toute solution de cet ensemble est optimale dans le sens qu'aucune amélioration ne peut être faite sur un critère sans dégradation d'au moins un autre critère.

Ce chapitre a pour objectif principal de présenter le contexte de l'optimisation multiobjectif discrète. Nous rappelons en premier lieu les éléments essentiels de la théorie de la programmation linéaire (en variables continues et en variables entières). Nous introduisons ensuite, les notions fondamentales ainsi que les principaux résultats liés à la théorie de la programmation multiobjectif en nombres entiers. Enfin et pour fixer les idées, nous présentons un exemple illustratif des différentes définitions.

### 1.2 Programmation linéaire en variables continues ([28], [31], [33])

La programmation linéaire est l'une des plus importantes techniques d'optimisation utilisées en recherche opérationnelle. Ses développements théoriques ont été

suggérés et accélérés par un grand nombre d'applications pratiques, dans le domaine de l'économie, de la gestion et autres.

### 1.2.1 Forme générale d'un programme linéaire

Le problème général de programmation linéaire est la recherche de l'optimum (maximum ou minimum) d'une fonction linéaire de  $n$  variables  $x_j$  ( $j = \overline{1, n}$ ) liées par des équations et/ou inéquations appelées contraintes.

La formulation mathématique d'un tel problème étant la suivante :

$$(P) \left\{ \begin{array}{l} \text{Optimiser } z = \sum_{j=1}^n c_j x_j \\ \sum_{j=1}^n a_{ij} x_j \leq b_i \quad i \in I \subseteq \{1, \dots, m\} \\ \sum_{j=1}^n a_{kj} x_j \geq b_k \quad k \in K \subseteq \{1, \dots, m\} \\ \sum_{j=1}^n a_{rj} x_j = b_r \quad r \in R \subseteq \{1, \dots, m\} \\ x_j \geq 0 \quad i = \overline{1, l} \\ x_j \text{ quelconque} \quad i = \overline{l+1, n} \end{array} \right.$$

Où :

- Optimiser pouvant signifier Minimiser ou Maximiser selon le problème traité.
- $c = (c_i)_{i=\overline{1, n}} \in \mathbb{R}^{1 \times n}$ ,  $A = (a_{ij})_{\substack{i=\overline{1, m} \\ j=\overline{1, n}}} \in \mathbb{R}^{m \times n}$ ,  $b = (b_i)_{i=\overline{1, m}} \in \mathbb{R}^m$ .
- Les ensembles  $I$ ,  $K$ , et  $R$  sont disjoints et  $I \cup K \cup R = \{1, \dots, m\}$ .

### 1.2.2 Forme canonique et forme standard

Les deux programmes linéaires :

$$(PC) \left\{ \begin{array}{l} \text{Max } z = \sum_{j=1}^n c_j x_j \\ \sum_{j=1}^n a_{ij} x_j \leq b_i \quad i = \overline{1, m} \\ x_j \geq 0 \quad j = \overline{1, n} \end{array} \right. \quad (PS) \left\{ \begin{array}{l} \text{Max } z = \sum_{j=1}^n c_j x_j \\ \sum_{j=1}^n a_{ij} x_j = b_i \quad i = \overline{1, m} \\ x_j \geq 0 \quad j = \overline{1, n} \end{array} \right.$$

sont écrits sous *forme canonique* et sous *forme standard* respectivement.

**Remarque.** Tout programme linéaire peut être écrit sous forme canonique ou sous forme standard en utilisant les règles de transformation suivantes :

1. Minimisation  $\leftrightarrow$  Maximisation :  $\min f(x) = -\max(-f(x))$ .

Pour minimiser  $z = cx$ , il suffit de maximiser  $w = -cx$  et de multiplier la valeur optimale de  $w$  par  $-1$  pour obtenir celle de  $z$ .

2. Inéquation " $\geq$ "  $\leftrightarrow$  inéquation " $\leq$ " :

$$ax \geq b \Leftrightarrow -ax \leq -b$$

3. Équation  $\rightarrow$  Inéquation " $\leq$ " :

$$ax = b \Leftrightarrow \begin{cases} ax \leq b \\ ax \geq b \end{cases} \Leftrightarrow \begin{cases} ax \leq b \\ -ax \leq -b \end{cases}$$

4. Inéquation  $\rightarrow$  équation : On ajoute une variable d'écart

$$ax \leq b \Leftrightarrow ax + s = b, s \geq 0$$

$$ax \geq b \Leftrightarrow ax - s = b, s \geq 0$$

5. variable de signe quelconque  $\rightarrow$  variable non négative : une variable de signe quelconque  $x$  peut toujours être remplacée par deux variables non négatives  $x'$  et  $x''$ .

$$x \in \mathbb{R} \Rightarrow \begin{cases} x = x' - x'' \\ x', x'' \geq 0 \end{cases}$$

### 1.2.3 Bases, bases réalisables, solutions de base

On considère le programme linéaire :

$$(P) \begin{cases} \text{Max} & z = cx \\ & Ax = b \\ & x \geq 0 \end{cases}$$

Où :  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$  et  $c \in \mathbb{R}^{1 \times n}$  tel que le système  $Ax = b$  soit de plain rang, c'est-à-dire le nombre de colonnes de  $A$  linéairement indépendantes est égal à  $m$ .

On appelle **base** un sous ensemble  $B \subset \{1, \dots, n\}$  d'indices de colonnes de  $A$  tel que  $A^B$  soit carrée non singulière. Le complémentaire de  $B$  dans  $\{1, \dots, n\}$  est l'ensemble d'indices **hors base** associé noté  $N$ .

Le système de contraintes et la fonction objectif peuvent s'écrire de manière équivalente :

$Ax = b \Leftrightarrow A^B x_B + A^N x_N = b$  et  $cx = c_B x_B + c_N x_N$ . Ainsi toute solution de  $(P)$  vérifie :  $x_B + (A^B)^{-1} A^N x_N = (A^B)^{-1} b$  et on constate que si l'on fixe la valeur des variables hors-bases  $x_N$ , on détermine celle des  $x_B$ .

On appelle **solution de base** (associée à  $B$ ), la solution particulière obtenue en fixant  $x_N$  à  $0_{\mathbb{R}^{|N|}}$ , d'où  $x_B = (A^B)^{-1} b$ . Une telle solution est dite **réalisable** si  $x_B \geq 0$ . Une solution de base est dite **dégénérée** si  $x_B$  a des composantes nulles.

#### 1.2.4 Caractérisation des bases optimales

Étant donnée une base réalisable  $B$  du programme linéaire  $(P)$ , le programme linéaire :

$$(PC) \begin{cases} x_B + (A^B)^{-1} A^N x_N = (A^B)^{-1} b \\ \hat{c}_N x_N = z - \pi b \\ x_B, x_N \geq 0 \end{cases}$$

Où :

- $\pi = c_B (A^B)^{-1}$  est dit *vecteur multiplicateur relatif à la base  $B$* .
- $\hat{c} = c - \pi A$  est dit *vecteur coût relatif à la base  $B$* .

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

**Théorème 1.1.** *Si le vecteur coût  $\hat{c}$  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**.*

### 1.2.5 Algorithme du simplexe

L'algorithme du simplexe, introduit en 1947 par G.B. Dantzig [12], décrit un moyen intelligent de se déplacer d'une solution de base admissible à une autre améliorant la valeur de la fonction objectif, jusqu'à trouver une solution optimale en un nombre fini d'étapes. Malgré une complexité théorique dans le pire des cas exponentielle, il permet de résoudre la plupart des problèmes rapidement.

Soit à résoudre le programme linéaire :

$$(P) \begin{cases} \text{Max} & z = cx \\ Ax & = b \\ x & \geq 0 \end{cases}$$

On définit l'application linéaire  $col$  par :

$$\begin{aligned} col : \{1, \dots, m\} &\longrightarrow \{1, \dots, n\} \\ i &\longrightarrow col(i) = \text{indice de la variable de base associée à la ligne } i. \end{aligned}$$

1. **Écriture canonique.** Soit  $B$  une base réalisable de départ, alors l'écriture canonique de  $(P)$  par rapport à  $B$  donne :

1	0	...	0	$\hat{A}^N = (A^B)^{-1}A^N$	$\hat{b} = (A^B)^{-1}b$
0	...		$\vdots$		
$\vdots$		...	0		
0	...	0	1		
0	...	0	0	$\hat{c}_N = c_N - c_B(A^B)^{-1}A^N$	$z - c_B(A^B)^{-1}b$

TAB. 1.1 – Écriture canonique de  $(P)/B$

2. **Choix de la colonne pivot.** (variable à entrer en base)

- (a) Si  $\forall j \in N, \hat{c}_j \leq 0$ , alors **STOP** (la solution optimale est trouvée).
- (b) Sinon, choisir une colonne  $s$  telle que :  $\hat{c}_s = \max_{j \in N} \hat{c}_j$

3. **Choix de la ligne pivot.** (variable à sortir de la base)

- (a) Si  $\forall i = \overline{1, m}, \hat{A}_i^s < 0$ , alors **STOP** (la fonction objectif n'est pas bornée).

(b) Sinon, choisir une ligne  $r$ , telle que :

$$\frac{\widehat{b}_r}{\widehat{A}_r^s} = \min_{i=\overline{1,m}} \left\{ \frac{\widehat{b}_i}{\widehat{A}_i^s} \mid \widehat{A}_i^s > 0 \right\}$$

$$B = B \cup \{s\} \setminus \text{col}(r)$$

4. **Opération pivot.** (passage au tableau suivant) : Soient  $L_1, L_2, \dots, L_m$  les  $m$  premières lignes du tableau (TAB.1.1) correspondants aux contraintes du problème et  $L_{m+1}$  la  $(m+1)$  ème ligne correspondant à la fonction objectif, alors les lignes du nouveau tableau sont calculées ainsi :

$$(a) L_i \leftarrow L_i - \frac{L_r \widehat{A}_i^s}{\widehat{A}_r^s} \quad i = \overline{1,m}, i \neq r$$

$$(b) L_{m+1} \leftarrow L_{m+1} - \frac{L_r \widehat{c}_s}{\widehat{A}_r^s}$$

$$(c) L_r \leftarrow \frac{L_r}{\widehat{A}_r^s}.$$

**Retour à (1)**

### 1.2.6 Dualité

Les deux programmes linéaires :

$$(P) \begin{cases} \text{Primal} \\ \text{Max} & z = cx \\ Ax = b \\ x \geq 0 \end{cases} \qquad (D) \begin{cases} \text{Dual} \\ \text{Min} & w = yb \\ yA \geq c \\ y \in \mathbb{R}^m \end{cases}$$

Sont dits **duaux** l'un de l'autre.

Si on a une base  $B$  alors :  $y(A^B, A^N) \geq (c_B, c_N) \Rightarrow \begin{cases} yA^B \geq c_B \\ yA^N \geq c_N \end{cases}$

On peut choisir  $y$  tel que  $yA^B = c_B$  en prenant  $y = c_B(A^B)^{-1}$

De plus on a :

$$c_B(A^B)^{-1}A^N \geq c_N$$

C'est à dire  $\hat{c}_j \leq 0$  pour tout  $j$  indice de variable hors base. Ca signifie donc que  $B$  réalise l'optimum si  $B$  est dual réalisable.

Cette remarque donne lieu à *l'algorithme dual* du simplexe qui part d'une situation où  $\hat{c}_j \leq 0, \forall j \in N$  mais où la base n'est pas admissible (des  $b_i$  sont  $<0$ ). L'algorithme consiste à choisir un pivot qui rend les  $b_i \geq 0$  tout en maintenant les  $\hat{c}_j \leq 0$ .

### 1.3 Programmation linéaire en nombres entiers ([28], [31], [36])

#### 1.3.1 Formulation mathématique

Considérons le programme linéaire :

$$(P) \begin{cases} \text{Max} & z = cx \\ & x \in S \end{cases} \quad (1.1)$$

Où :

- $S = \{x \in \mathbb{R}^n | Ax = b, x \geq 0\}$ .
- $A \in \mathbb{Z}^{m \times n}, b \in \mathbb{Z}^m, c \in \mathbb{Z}^{1 \times n}$ .
- $S$  est supposé borné et non vide.

Une solution optimale de  $(P)$  comportera généralement des composantes fractionnaires. Pour certains problèmes une telle solution n'est pas admissible. On devra dans ce cas imposer aux variables des contraintes supplémentaires (dites contraintes d'intégrité) :  $x_j$  entier pour  $j = \overline{1, n}$ .

Le problème deviendra donc :

$$(Q) \begin{cases} \text{Max} & z = cx \\ & x \in D \end{cases} \quad (1.2)$$

Où :  $D = \{x \in \mathbb{Z}^n | Ax = b, x \geq 0\}$  avec  $\mathbb{Z}$  l'ensemble des nombres entiers relatifs. Un tel problème est appelé *programme linéaire en nombres entiers*.

Les deux principales familles de méthodes actuellement connues pour résoudre les

problèmes de programmation linéaire en nombres entiers sont les *méthodes de coupes* et les *méthodes arborescentes*.

### 1.3.2 Coupe fractionnaire de Gomory [15]

Soit à résoudre le programme linéaire en nombres entiers suivant :

$$(Q) \begin{cases} \text{Max} & z = cx \\ & x \in D \end{cases}$$

Dont le problème relaxé est :  $(P) \text{Max } \{z = cx | x \in S\}$ .

Dans cette méthode, le programme linéaire  $(P)$  est résolu en première étape en utilisant la méthode du simplexe. A l'optimum on a les conditions : (1)  $\hat{c} \leq 0$  et (2)  $\hat{b} \geq 0$  qui sont vérifiées. Si de plus : (3)  $\hat{b}$  est entier, alors la solution optimale de  $(P)$  est aussi optimale pour  $(Q)$ . Sinon, une ou plus des variables de base soumises à l'intégrité sont à valeurs fractionnaires.

L'idée principale de l'algorithme de Gomory [15] est de maintenir les conditions (1) et (2) satisfaites et de rajouter des contraintes dites *coupes de Gomory* [15] une par une jusqu'à ce que la troisième condition soit vérifiée.

**Définition 1.1.** Etant donné un programme linéaire en nombres entiers  $(Q)$ , On dit que l'inéquation

$$\sum_{j=1}^n \alpha_j x_j \leq \beta$$

est *valide* si elle est satisfaite par tout point de  $D$ . Une coupe est une inéquation valide qui n'est pas satisfaite pour tout point de  $S$ .

**Définition 1.2.** Soit  $\alpha$  un scalaire quelconque, on désigne par :

- $[\alpha]$  le plus grand entier inférieur ou égal à  $\alpha$ .
- $\lceil \alpha \rceil$  le plus petit entier supérieur ou égal à  $\alpha$ .
- $\langle \alpha \rangle = \alpha - [\alpha]$ .

$\langle \alpha \rangle$  est appelée la *partie fractionnaire* de  $\alpha$  et  $[\alpha]$  sa *partie entière*.

**Génération de la coupe de Gomory [15]**

Soit le tableau optimal issu de la résolution de  $(P)$  par la méthode du simplexe où la solution optimale est supposée non entière (voir Tab.1.2). Sans perte de généralités, on suppose qu'à l'optimum, il ya un total de  $(m + n)$  variables où  $m$  représente le nombre de variables de base notées  $x_i$ , ( $i = \overline{1, m}$ ) et  $n$  représente le nombre de variables hors base notées  $y_j$ , ( $j = \overline{1, n}$ ).

$vb$	$x_1$	$x_2$	$\cdots$	$x_i$	$\cdots$	$x_m$	$y_1$	$y_2$	$\cdots$	$y_j$	$\cdots$	$y_n$	$\widehat{b}_i$
$x_1$	1	0	$\cdots$	0	$\cdots$	0	$\widehat{a}_{11}$	$\widehat{a}_{12}$	$\cdots$	$\widehat{a}_{1j}$	$\cdots$	$\widehat{a}_{1n}$	$\widehat{b}_1$
$x_2$	0	1	$\cdots$	0	$\cdots$	0	$\widehat{a}_{21}$	$\widehat{a}_{22}$	$\cdots$	$\widehat{a}_{2j}$	$\cdots$	$\widehat{a}_{2n}$	$\widehat{b}_2$
$\vdots$													
$x_i$	0	0	$\cdots$	1	$\cdots$	0	$\widehat{a}_{i1}$	$\widehat{a}_{i2}$	$\cdots$	$\widehat{a}_{ij}$	$\cdots$	$\widehat{a}_{in}$	$\widehat{b}_i$
$\vdots$													
$x_m$	0	0	$\cdots$	0	$\cdots$	1	$\widehat{a}_{m1}$	$\widehat{a}_{m2}$	$\cdots$	$\widehat{a}_{mj}$	$\cdots$	$\widehat{a}_{mn}$	$\widehat{b}_m$
$-z$	0	0	$\cdots$	0	$\cdots$	0	$\widehat{c}_1$	$\widehat{c}_2$	$\cdots$	$\widehat{c}_j$	$\cdots$	$\widehat{c}_n$	$z^*$

TAB. 1.2 – Tableau optimal

A partir du tableau optimal, choisir une variable de base dont la valeur est fractionnaire, soit  $x_i$ . La  $i^{\text{ème}}$  équation du tableau est donnée par :

$$x_i + \sum_{j=1}^n \widehat{a}_{ij} y_j = \widehat{b}_i \quad (1.3)$$

En décomposant chaque coefficient en la somme de sa partie entière et de sa partie fractionnaire, on obtient :

$$x_i + \sum_{j=1}^n \langle \widehat{a}_{ij} \rangle y_j + \sum_{j=1}^n [\widehat{a}_{ij}] y_j = \langle \widehat{b}_i \rangle + [\widehat{b}_i]$$

Puisque  $0 \leq \langle \widehat{a}_{ij} \rangle < 1$ , ceci implique :

$$x_i + \sum_{j=1}^n [\widehat{a}_{ij}] y_j \leq \langle \widehat{b}_i \rangle + [\widehat{b}_i]$$

Or le membre gauche est entier et  $0 < \langle \widehat{b}_i \rangle < 1$ . Par conséquent cette dernière équation implique :

$$x_i + \sum_{j=1}^n \lfloor \widehat{a}_{ij} \rfloor y_j \leq \lfloor \widehat{b}_i \rfloor$$

et en soustrayant cette dernière inéquation de (1.3), on obtient :

$$x_i + \sum_{j=1}^n \langle \widehat{a}_{ij} \rangle y_j \geq \langle \widehat{b}_i \rangle \tag{1.4}$$

qui est bien valide. De plus (1.4) n'est pas satisfaite par la solution optimale de (P), donc (1.4) est bien une coupe.

En ajoutant une variable d'écart  $s_i$  à (1.4), et en posant  $\alpha_{ij} = \langle \widehat{a}_{ij} \rangle$  et  $\beta_i = \widehat{b}_i$ , la coupe de Gomory [15] devient :

$$s_i - \sum_{j=1}^n \alpha_{ij} y_j = -\beta_i \tag{1.5}$$

Une fois la coupe est générée, les coefficients de (1.5) sont insérés dans une nouvelle ligne du tableau (Tab.1.2). Le nouveau tableau est donné par :

$vb$	$x_1$	$x_2$	$\dots$	$x_i$	$\dots$	$x_m$	$s_i$	$y_1$	$y_2$	$\dots$	$y_j$	$\dots$	$y_n$	$\widehat{b}_i$
$x_1$	1	0	$\dots$	0	$\dots$	0	0	$\widehat{a}_{11}$	$\widehat{a}_{12}$	$\dots$	$\widehat{a}_{1j}$	$\dots$	$\widehat{a}_{1n}$	$\widehat{b}_1$
$x_2$	0	1	$\dots$	0	$\dots$	0	0	$\widehat{a}_{21}$	$\widehat{a}_{22}$	$\dots$	$\widehat{a}_{2j}$	$\dots$	$\widehat{a}_{2n}$	$\widehat{b}_2$
$\vdots$														
$x_i$	0	0	$\dots$	1	$\dots$	0	0	$\widehat{a}_{i1}$	$\widehat{a}_{i2}$	$\dots$	$\widehat{a}_{ij}$	$\dots$	$\widehat{a}_{in}$	$\widehat{b}_i$
$\vdots$														
$x_m$	0	0	$\dots$	0	$\dots$	1	0	$\widehat{a}_{m1}$	$\widehat{a}_{m2}$	$\dots$	$\widehat{a}_{mj}$	$\dots$	$\widehat{a}_{mn}$	$\widehat{b}_m$
$s_i$	0	0	$\dots$	0	$\dots$	0	1	$-\alpha_{i1}$	$-\alpha_{i2}$	$\dots$	$-\alpha_{ij}$	$\dots$	$-\alpha_{in}$	$-\beta_i$
$-z$	0	0	$\dots$	0	$\dots$	0	0	$\widehat{c}_1$	$\widehat{c}_2$	$\dots$	$\widehat{c}_j$	$\dots$	$\widehat{c}_n$	$z^*$

TAB. 1.3 – Tableau augmenté

Pour ce nouveau programme, la condition (1) est toujours satisfaite mais pas la condition (2). On effectue donc une ou plusieurs itérations de l'algorithme dual du simplexe, jusqu'à ce que la condition (2) soit satisfaite. Si  $\widehat{b}$  est entier, la solution courante est

optimale pour  $(Q)$ , sinon on recommence le processus.

**Remarque.** Cette méthode a un sérieux inconvénient. C'est associé aux erreurs d'arrondis qui surgissent pendant les calculs numériques. En raison de ces erreurs d'arrondis, nous pouvons finalement obtenir de faux résultats concernant le test d'intégrité sur les variables. D'autre part, le nombre de coupes de Gomory [15] à générer peut être très grand. Ce qui fait augmenter la taille du problème sans limite, puisqu'une variable d'écart et une contrainte sont ajoutés à chaque fois que  $\hat{b}$  n'est pas entier.

### 1.3.3 Méthode par séparation et évaluation

La méthode par séparation et évaluation (*Branch & Bound*) est très efficace pour la résolution des problèmes de programmation linéaire en nombres entiers. Elle a été à l'origine développée par Land et Doig [22] pour résoudre un problème de programmation linéaire en nombres entiers et a été modifiée plus tard par Dakin [11].

Une approche naïve pour résoudre un problème de programmation linéaire en nombres entiers est d'énumérer tous les points entiers réalisables du problème, d'évaluer la fonction objectif en chaque point et d'identifier celui qui a la meilleure valeur de fonction objectif. Bien qu'une recherche si approfondie dans l'espace des solutions réalisables soit simple de mettre en œuvre, elle sera très coûteuse en terme de temps de calcul même pour des problèmes de taille réduite.

La méthode par séparation et évaluation peut être considérée comme une méthode d'énumération raffinée dans laquelle plusieurs points entiers réalisables sont écartés sans les évaluer. Son principe repose sur trois notions distinctes : séparation du problème principal, relaxation des sous problèmes et stérilisation de l'arbre de recherche.

1. **Séparation** : Considérons le problème de programmation linéaire en nombres entiers  $(Q)$  dont l'espace des solutions réalisables est noté  $D(Q)$ .

L'ensemble de sous problèmes  $(Q_1), (Q_2), \dots, (Q_n)$  est une séparation de  $(Q)$  si :

$$\bigcup_i D(Q_i) = D(Q) \text{ et } \forall i, j \text{ avec } i \neq j, D(Q_i) \cap D(Q_j) = \emptyset.$$

2. **Relaxation** : Le domaine  $S$  des solutions réalisables de la relaxation  $(P)$  du problème  $(Q)$  contient celui de  $(Q)$  ce qui implique que :

- (a) Si  $(P)$  n'est pas réalisable, alors  $(Q)$  l'est aussi ;
- (b) La solution optimale de  $(P)$  est une borne supérieure de la solution optimale de  $(Q)$  ;
- (c) Une solution optimale de  $(P)$  réalisable pour  $(Q)$  est une solution optimale de  $(Q)$ .

3. **Stérilisation** : Soit  $(Q_k)$  un sous problème de  $(Q)$  susceptible de mener à la solution optimale de  $(Q)$ .

Trois critères peuvent autoriser à stériliser  $(Q_k)$  :

- (a) La relaxation de  $(Q_k)$  n'est pas réalisable ;
- (b) L'évaluation de la solution optimale de la relaxation de  $(Q_k)$  est inférieure à celle de la meilleure solution réalisable trouvée antérieurement (soit  $U$  son évaluation) ;
- (c) la solution optimale de la relaxation de  $(Q_k)$  est réalisable, donc optimale pour  $(Q_k)$ . Si en plus son évaluation est supérieure à  $U$ , alors cette dernière peut être actualisée.

Un problème de programmation linéaire en nombres entiers  $(Q)$  peut avoir plusieurs solutions optimales, on parle dans ce cas de solutions alternatives dont la définition est la suivante :

**Définition 1.3.** Soit  $x^*$  une solution optimale du problème  $(Q)$ . Une solution réalisable  $x' \in D$  est dite alternative à  $x^*$  si  $cx' = cx^*$ .

## 1.4 Les problèmes d'optimisation non linéaires ([24],[25])

On parle de problème d'optimisation non linéaire lorsque l'on doit maximiser ou minimiser une fonction sous contraintes et que la fonction objectif, ou au moins une

contrainte est non linéaire. Dans cette partie, nous allons donc nous attacher à la résolution du problème suivant :

$$\begin{cases} \text{Min} & f(x) \\ & g_j(x) \leq 0, j = \overline{1, m} \end{cases}$$

où  $f(x), g_j(x), j = \overline{1, m}$  sont des fonctions continûment différentiables définies sur  $\mathbb{R}^n$  à valeurs dans  $\mathbb{R}$ . Il s'agit donc de la généralisation au cas non linéaire du problème classique de la programmation linéaire :

$$\begin{cases} \text{Min} & cx \\ & Ax \leq 0 \\ & x \geq 0 \end{cases}$$

Les problèmes non linéaires sont beaucoup plus difficiles à résoudre en général que les problèmes linéaires. En effet, contrairement à la programmation linéaire, on peut avoir une solution optimale qui n'est pas située en un point extrême de la région réalisable. Aussi, on peut avoir des optimums locaux<sup>1</sup> qui ne sont pas globaux<sup>2</sup>.

La détermination de tous les optimums locaux est un problème très délicat en général. En effet, lorsqu'un optimum local est atteint, dans quelle direction faut-il continuer la recherche ? Cependant, il existe une classe de problèmes pour lesquels ce problème ne se produit pas dans le sens que tous les optimums locaux sont des optimums globaux. Il s'agit des problèmes convexes. Un problème d'optimisation est dit convexe s'il s'agit de l'optimisation d'une fonction convexe sur une région réalisable convexe.

### 1.4.1 Convexité

**Définition 1.4** (Ensemble convexe). L'ensemble  $S \in \mathbb{R}^n$  est convexe, si et seulement si quels que soient les deux points  $p_1$  et  $p_2$  pris dans l'ensemble  $S$ , tout le segment  $[p_1, p_2]$  est contenu dans l'ensemble  $S$ .

**Définition 1.5** (Fonction convexe).  $f$  est convexe si et seulement si, pour tout  $(x, y) \in$

---

<sup>1</sup> $x^*$  est un minimum local :  $\exists \epsilon > 0$  tel que  $f(x) - f(x^*) \geq 0, \forall x \neq x^*$  avec  $\|x - x^*\| < \epsilon$ .

<sup>2</sup> $x^*$  est un minimum global :  $f(x) - f(x^*) \geq 0 \forall x \neq x^*$ .

$S^2$  et tout  $\lambda \in [0, 1]$ ,  $f((1 - \lambda)x + \lambda y) \leq (1 - \lambda)f(x) + \lambda f(y)$ .

### 1.4.2 Conditions de Kuhn et Tucker

Nous considérons le problème de minimisation sous contraintes suivant :

$$\begin{cases} \text{Min} & f(x) \\ & g_j(x) \leq 0, j = \overline{1, m} \end{cases}$$

où  $f(x)$ ,  $g_j(x)$ ,  $j = \overline{1, m}$  sont des fonctions de  $\mathbb{R}^n$  à valeurs dans  $\mathbb{R}$ .

Pour écrire les conditions d'optimalité, on a besoin de calculer les dérivées partielles.

Rappelons cette notion pour une fonction de  $\mathbb{R}^n$  dans  $\mathbb{R}$ .

**Définition 1.6.** La dérivée partielle de  $f$  par rapport à  $x_i$ , notée  $\frac{\partial f}{\partial x_i}$ , est obtenue en dérivant  $f$  en considérant les autres variables comme des constantes.

**Définition 1.7.** La fonction de Lagrange est obtenue en multipliant le membre de gauche de chaque contrainte d'inégalité par son multiplicateur  $\mu_j$  et en additionnant le tout à la fonction objectif :

$$L(x) = f(x) + \sum_{j=1}^m \mu_j g_j(x)$$

On peut alors écrire les très importantes conditions nécessaires suivantes :

**Théorème 1.2** (Conditions de Kuhn et Tucker). *Soit  $x^*$  un minimum local pour le problème :*

$$\begin{cases} \text{Min} & f(x) \\ & g(x) \leq 0 \end{cases}$$

*et supposons  $x^*$  régulier pour les contraintes. Alors il existe des multiplicateurs  $\mu \geq 0$  tels que :*

$$\begin{cases} \frac{\partial L(x^*)}{\partial x_i} = 0 & \forall i = \overline{1, n} \\ \mu_j g_j(x^*) = 0 & \forall j = \overline{1, m} \\ \mu_j \geq 0 & \forall j = \overline{1, m} \end{cases}$$

**Proposition 1.1.** *Si le problème est un problème convexe, les conditions nécessaires sont aussi suffisantes pour montrer que l'on est à l'optimum.*

## 1.5 Programmation multiobjectif discrète ([8], [34], [37])

Dans les sections précédentes, nous n'avons considéré que les cas où le problème à traiter possédait un objectif unique à optimiser. Pour de nombreuses applications pratiques, ce n'est en fait pas le cas et il faut optimiser en même temps plusieurs objectifs pour un même problème. La fonction objectif n'est alors plus scalaire mais vectorielle, on parle dans ce cas de problème multiobjectif.

### 1.5.1 Formulation Générale

Le problème de programmation multiobjectif en nombres entiers est défini comme un problème de décision, qui consiste à optimiser (maximiser ou minimiser) simultanément  $r$  fonctions à valeurs réelles notées  $f^i$ ,  $i = \overline{1, r}$  sur un domaine défini par le système de contraintes  $g_j(x) \leq 0$ ,  $j = \overline{1, m}$  où  $g_j$  est une fonction réelle définie sur  $R$ ,  $j = \overline{1, m}$ .

Ce problème peut être formulé comme suit :

$$(MOIP) \begin{cases} \text{"Optimiser"} & f^i(x) & i = \overline{1, r} \\ g_j(x) \leq 0 & j = \overline{1, m} \\ x \in \mathbb{Z}^n \end{cases}$$

Le symbole " " signifie qu'il n'est généralement pas possible de trouver un vecteur  $x \in \mathbb{Z}^n$  vérifiant les différentes contraintes et qui optimise simultanément les  $r$  critères.

**Remarque.** Sans perte de généralités, nous supposons dans la suite que toutes les fonctions objectif sont à maximiser.

Si les  $r$  critères sont linéaires et les contraintes sont linéaires en  $x$ , on parle de problème

de programmation linéaire multiobjectif en nombres entiers (MOILP) :

$$(MOILP) \begin{cases} \text{''Max''} & z^i = c^i x \quad i = \overline{1, r} \\ x \in D \end{cases}$$

Où :

- $r$  le nombre d'objectifs ( $r \geq 2$ ).
- $D = \{x \in \mathbb{Z}^n | Ax = b, x \geq 0\}$ .

Avec :  $A \in \mathbb{Z}^{m \times n}$ ,  $b \in \mathbb{Z}^m$ ,  $C = (c^1, c^2, \dots, c^r) \in \mathbb{Z}^{r \times n}$ .

Si par contre, les  $r$  critères sont fractionnaires linéaires et les contraintes sont linéaires en  $x$ , on obtient un problème de programmation fractionnaire linéaire multiobjectif en nombres entiers (MOILFP) :

$$(MOILFP) \begin{cases} \text{''Max''} & z^i = \frac{c^i x + \alpha^i}{d^i x + \beta^i} \quad i = \overline{1, r} \\ x \in D \end{cases}$$

Avec :  $A \in \mathbb{Z}^{m \times n}$ ,  $b \in \mathbb{Z}^m$ ,  $c^i$  et  $d^i \in \mathbb{Z}^{1 \times n}$ ,  $\alpha^i$  et  $\beta^i \in \mathbb{Z}$ ,  $i = \overline{1, r}$  et  $d^i x + \beta^i > 0$  sur  $S = \{x \in \mathbb{R}^n | Ax = b, x \geq 0\}$  pour tout  $i = \overline{1, r}$ .

### 1.5.2 Concepts de base

Soit le problème de programmation linéaire en nombres entiers à objectifs multiples suivant :

$$(P) \begin{cases} \text{''Max''} & z^i = c^i x \quad i = \overline{1, r} \\ x \in D \end{cases}$$

Où :  $D = \{x \in \mathbb{Z}^n | Ax = b, x \geq 0\}$  est supposé borné et non vide.

On considère l'application linéaire  $\varphi$  qui associe à chaque vecteur  $x \in D$  son image  $\varphi(x) = (c^1 x, c^2 x, \dots, c^r x)$  dans l'espace des critères.

**Définition 1.8.** L'espace  $\mathbb{Z}^n$  dans lequel se situe l'ensemble des actions  $D$  est appelé *espace des décisions*. D'autre part, et dans le cadre multiobjectif, le décideur raisonne plutôt en terme d'évaluation d'une solution pour chaque objectif et se place naturel-

lement dans l'espace :

$$\varphi(D) = \{z = Cx \in \mathbb{R}^r \mid x \in D\}$$

appelé *espace des critères*.

La résolution d'un problème d'optimisation multiobjectif conduit à l'obtention d'une multitude de solutions dites efficaces. En effet, les solutions n'admettent pas une relation d'ordre total mais une relation d'ordre partiel notée  $>$  car une solution pouvant être meilleure qu'une autre sur certains objectifs et moins bonne sur d'autres. Dans ce cas, la notion de dominance est introduite et est définie comme suit :

### **Dominance**

Soient  $z, z'$  deux vecteurs de  $\mathbb{R}^r$ . On dit que  $z$  domine  $z'$  ssi :  
 $z \geq z'$  et  $z \neq z'$  (i.e.  $z^i \geq z'^i \forall i = \overline{1, r}$  et  $z^i > z'^i$  pour au moins un  $i$ ).

Autrement dit,  $z$  domine  $z'$  ssi :

- $z$  est au moins aussi bon que  $z'$  pour tous les critères et,
- $z$  est strictement meilleur que  $z'$  pour au moins un critère.

### **Efficacité**

Une solution  $x^*$  est dite *efficace* si et seulement s'il n'existe pas de  $x \in D$  telle que :

$$Cx \geq Cx^*$$

avec au moins une inégalité stricte.

Autrement dit, une solution  $x^*$  est efficace si et seulement si le vecteur critère qui lui est associé n'est dominé par aucun autre vecteur.

On note par  $ESE(P)$  l'ensemble des solutions efficaces d'un problème de programmation linéaire multiobjectif en nombres entiers et par  $SND(P)$  celui des solutions non dominées.

### Efficacité faible

Une solution  $x^*$  est dite *faiblement efficace* si et seulement s'il n'existe pas de  $x \in D$  telle que :  $Cx > Cx^*$ .

Parmi les points particuliers qui apparaissent dans l'espace des critères, on trouve : le point *idéal*, le point *nadir*, etc.

### Point idéal

Les coordonnées de ce point sont obtenues en optimisant chaque fonction objectif séparément :  $\bar{z} = \left( \underset{x \in D}{\text{Max}} c^1 x, \underset{x \in D}{\text{Max}} c^2 x, \dots, \underset{x \in D}{\text{Max}} c^r x \right) \in \mathbb{R}^r$ . Généralement, parce que les objectifs sont contradictoires, le point idéal ne correspond pas à une solution réalisable.

### Point nadir

Soit  $\bar{x}_i$  la solution optimale obtenue en optimisant le critère  $z^i$  sur  $D$ . La matrice carrée de dimension  $r$  suivante :

$$\begin{pmatrix} \bar{z}_1 & z_{12} & \cdots & z_{1r} \\ z_{21} & \bar{z}_2 & \cdots & z_{2r} \\ \vdots & \vdots & \vdots & \vdots \\ z_{r1} & z_{r2} & \cdots & \bar{z}_r \end{pmatrix}$$

est appelée *matrice des gains*.

Où :

- $\bar{z}_i = \underset{x \in D}{\text{Max}} c^i x = c^i \bar{x}_i, \forall i = \overline{1, r}$ .
- $z_{ij} = c^i \bar{x}_j \forall i = \overline{1, r}, \forall j = \overline{1, r}$  avec  $i \neq j$ .

De la matrice des gains peut être défini le point nadir noté  $\eta \in \mathbb{R}^r$  :

$$\eta_i = \underset{i=\overline{1, r}}{\text{Min}} z_{ij} \forall j = \overline{1, r}$$

**Remarque.** Si pour un critère  $j$ , il existe plusieurs solutions optimales, la matrice

des gains n'est pas unique. En effet, la colonne  $j$  de la matrice des gains dépendra de la solution  $\bar{x}_j$  choisie. Dans ce cas, les coordonnées du point idéal ne changent pas d'une solution à une autre.

### Solutions supportées et non supportées

L'ensemble des solutions efficaces pour les problèmes de programmation linéaire multiobjectif en variables continues :

$$(MOLP) \begin{cases} \text{"Max"} & z^i = c^i x \quad i = \overline{1, r} \\ x \in S \end{cases}$$

où  $S = \{x \in \mathbb{R}^n | Ax = b, x \geq 0\}$  est bien caractérisé. En effet, pour toute solution efficace  $\bar{x}$  du problème  $(MOLP)$ ,  $\exists \bar{\lambda} > 0$  tel que  $\bar{x}$  soit optimale pour le problème paramétrique :

$$(P_\lambda) \begin{cases} \text{Max} & \sum_{i=1}^r \lambda_i c^i x \\ x \in S \end{cases}$$

$\lambda = (\lambda_1, \lambda_2, \dots, \lambda_r) \in \mathbb{R}^r$  avec,  $\lambda_i \geq 0 \forall i = \overline{1, r}$  et  $\sum_{i=1}^r \lambda_i = 1$ , et réciproquement (c'est à dire : si pour une certaine valeur de  $\lambda$ ,  $\bar{x}$  est une solution optimale de  $(P_\lambda)$ , alors  $\bar{x}$  est une solution efficace du problème  $(MOLP)$ ).

**Théorème 1.3** (Geoffrion [14]). *Étant donné le problème  $(P_\lambda)$ , alors  $\bar{x}$  est une solution efficace pour  $(MOLP)$  si et seulement si  $\bar{x}$  est une solution optimale du problème paramétrique  $(P_\lambda)$ .*

Dans le cas des problèmes  $(MOILP)$ , la difficulté principale rencontrée différemment des problème  $(MOLP)$ , est l'existence de solutions qui ne sont pas optimales pour  $(P_\lambda)$  (en remplaçant  $S$  par  $D$ ), et ce en raison de la non convexité du domaine des solutions réalisables  $D$ , ces solutions sont dites *solutions efficaces non supportées* notées **SENS** (contre exemple présenté par Bowman [5]). Celles qui sont optimales pour  $(P_\lambda)$ , sont appelées *solutions efficaces supportées* notées **SES**.

### 1.5.3 Cônes

**Définition 1.9 (Cône).** Soit  $u \in U \subset \mathbb{R}^n$ ,  $U \neq \emptyset$ . Alors,  $U$  est un *cône* si et seulement si  $\alpha u \in U$  pour tout scalaire  $\alpha \geq 0$ . L'origine  $0 \in \mathbb{R}^n$  est contenu dans chaque cône.

Excepté de l'ensemble singleton qui contient uniquement l'origine, tous les cônes sont non bornés. Comme exemple, le demi espace fermé  $\{x \in \mathbb{R}^n | c^t x \leq 0\}$  est un cône mais le demi espace ouvert  $\{x \in \mathbb{R}^n | c^t x > 0\}$  n'est pas un cône car il ne contient pas l'origine.

**Définition 1.10 (Générateurs).** Considérons  $\{u^1, u^2, \dots, u^k\}$ , un ensemble de  $k$  vecteurs de  $\mathbb{R}^n$  et l'ensemble  $U$  tel que :

$$U = \left\{ u \in \mathbb{R}^n \mid u = \sum_{i=1}^k \alpha_i u^i, \alpha_i > 0 \right\}$$

$U$  est l'ensemble de toutes les combinaisons linéaires à coefficients non négatifs des  $u^i$ ,  $i = \overline{1, k}$  et est le cône convexe engendré par l'ensemble  $\{u^1, u^2, \dots, u^k\}$ . Les vecteurs  $u^i$ ,  $i = \overline{1, k}$  sont appelés les *générateurs* de  $U$ .

Le seul cône pour lequel l'ensemble des générateurs est unique est  $\{0 \in \mathbb{R}^n\}$ .

Soit  $\{u^1, u^2, \dots, u^k\}$  un ensemble de générateurs pour le cône convexe  $U$  et soit  $u^l \in \{u^1, u^2, \dots, u^k\}$ . Alors,  $u^l$  est *générateur non essentiel* si  $U$  peut être généré par  $\{u^1, u^2, \dots, u^k\} \setminus \{u^l\}$ . Un générateur non essentiel est celui qui peut être exprimé comme combinaison linéaire d'autres générateurs, il est dit *essentiel* sinon.

**Définition 1.11 (Dimension d'un cône).** La *dimension* d'un cône  $U \subset \mathbb{R}^n$  est donné par le nombre de vecteurs linéairement indépendants de  $U$ .

Par exemple, la dimension du cône singleton  $\{0 \in \mathbb{R}^n\}$  est 0, et la dimension d'un cône convexe généré par un ensemble de  $k$  vecteurs linéairement indépendants est  $k$ . On peut déterminer la dimension d'un cône en calculant le rang de la matrice dont les lignes (ou les colonnes) sont les générateurs de ce cône.

**Définition 1.12 (Cône polaire).** Soit  $U \subset \mathbb{R}^n$  un cône. Alors, le *cône polaire* non négatif de  $U$  (noté  $U^\geq$ ) est le cône convexe :

$$U^\geq = \{y \in \mathbb{R}^n \mid y^t u \geq 0 \text{ pour tout } u \in U\}$$

C'est-à-dire, tous les vecteurs de  $U^\geq$  font un angle inférieur ou égal à  $90^\circ$  avec chaque vecteur de  $U$ .

**Définition 1.13 (Cône polaire semi positif).** Soit  $U \subset \mathbb{R}^n$  un cône généré par l'ensemble  $\{u^1, u^2, \dots, u^k\}$ . Alors, le *cône polaire semi positif* de  $U$  noté  $U^>$  est le cône convexe :

$$U^> = \{y \in \mathbb{R}^n \mid y^t u^i \geq 0 \text{ pour tout } i \text{ et } y^t u^i > 0 \text{ pour au moins un } i\} \cup \{0 \in \mathbb{R}^n\}$$

Notons qu'un vecteur  $y \in U^>$  doit avoir un produit vectorielle positif avec au moins l'un des  $u^i$ ,  $i = \overline{1, k}$ . L'origine  $\{0 \in \mathbb{R}^n\}$  est incluse car sinon  $U^>$  ne serait pas un cône.

#### 1.5.4 Détection graphique de l'efficacité

Soit le problème de programmation linéaire multiobjectif en nombres entiers suivant :

$$(P) \begin{cases} \text{"Max"} & z^i = c^i x \quad i = \overline{1, r} \\ x \in D \end{cases}$$

Où :  $D = \{x \in \mathbb{Z}^n \mid Ax = b, x \geq 0\}$  avec :  $A \in \mathbb{Z}^{m \times n}$ ,  $b \in \mathbb{Z}^m$  et  $C = (c^i)_{i=\overline{1, r}} \in \mathbb{Z}^{r \times n}$ .

Pour tester l'efficacité en un point  $x^* \in D$ , Ralph E. Steuer [34] a introduit le concept d'*ensemble dominant* qui est principalement basé sur la notion du cône étudiée précédemment.

**Définition 1.14 (Ensemble dominant).** Soit  $x^* \in D$  et  $C^>$  le cône polaire semi positif du cône  $C$  généré par les gradients des  $r$  fonctions objectifs i.e. :

$$C^> = \{y \in \mathbb{R}^n \mid y^t (c^i)^t \geq 0 \text{ pour tout } i \text{ et } y^t (c^i)^t > 0 \text{ pour au moins un } i\} \cup \{0 \in \mathbb{R}^n\}$$

On définit l'ensemble dominant noté  $ED_{x^*}$ , comme étant la somme des ensembles  $\{x^*\}$  et  $C^>$  :

$$ED_{x^*} = x^* \oplus C^>$$

C'est à dire :

$$ED_{x^*} = \{x \in \mathbb{R}^n \mid x = x^* + y, Cy \geq 0, Cy \neq 0\}$$

L'ensemble dominant  $ED_{x^*}$  contient tous les points dont les vecteurs critères dominant le vecteur critère de  $x^* \in D$ . Notons que la somme des ensembles  $\{x^*\}$  et  $C^>$  effectue une translation du cône polaire semi positif de l'origine vers le point en question.

**Théorème 1.4** ([34]). *Soit  $ED_{x^*}$  l'ensemble dominant en  $x^* \in D$ . Alors  $x^*$  est efficace si et seulement si :  $ED_{x^*} \cap D = \{x^*\}$ .*

*Démonstration.*  $\Rightarrow$  / Supposons que  $ED_{x^*} \cap D \neq \{x^*\}$ . Alors, il existe  $\bar{x} \in ED_{x^*} \cap D$ ,  $\bar{x} \neq x^*$ . Puisque  $\bar{x} \in ED_{x^*}$ , alors  $\bar{x} = x^* + y$  où  $y \in C^>$ . Comme  $Cy \geq 0$ ,  $Cy \neq 0$  alors  $C\bar{x} \geq Cx^*$ ,  $C\bar{x} \neq Cx^*$ . Ceci contredit le fait que  $x^*$  est efficace. Alors si  $x^*$  est efficace,  $ED_{x^*} \cap D = \{x^*\}$ .

$\Leftarrow$  / Si  $ED_{x^*} \cap D = \{x^*\}$ , ceci implique que si le vecteur critère de  $\bar{x}$  domine le vecteur critère de  $x^*$ ,  $\bar{x} \notin D$  alors le vecteur critère de  $x^*$  est non dominé, et par conséquent,  $x^*$  est efficace.  $\square$

Le théorème (1.2) fournit un test permettant de détecter les points efficaces qui peuvent être visualisés géométriquement :

- Si l'intersection de l'ensemble dominant avec la région réalisable contient seulement  $x^*$ , alors  $x^*$  est efficace.
- S'il existe d'autres points appartenant à l'intersection de ces deux ensembles, alors  $x^*$  n'est pas efficace.

### 1.5.5 Illustration des définitions

Considérons le programme linéaire biobjectif en nombres entiers suivant :

$$(P) \begin{cases} \text{"Max"} & (x_1, -x_1 + x_2) \\ x_1 + 2x_2 \leq 10 \\ x_1 + x_2 \leq 6 & x_1, x_2 \in \mathbb{Z}_+ \\ x_1 \leq 4 \end{cases}$$

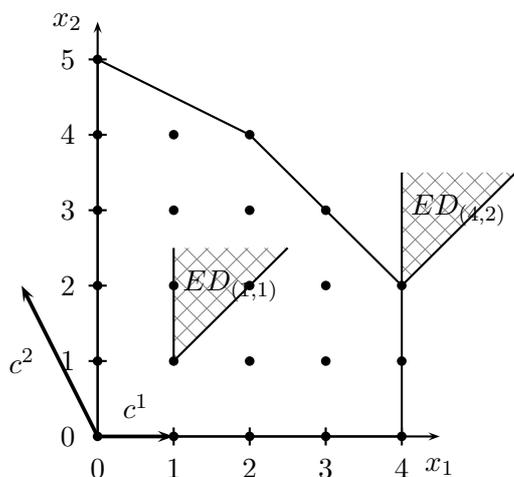


FIG. 1.1 – Espace des décisions

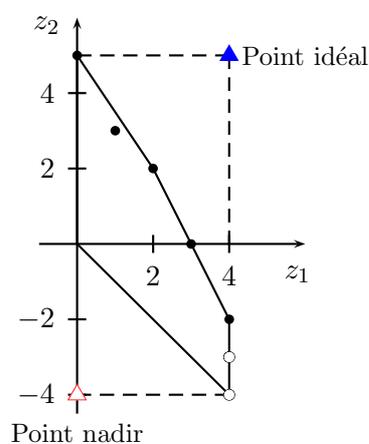


FIG. 1.2 – Espace des critères

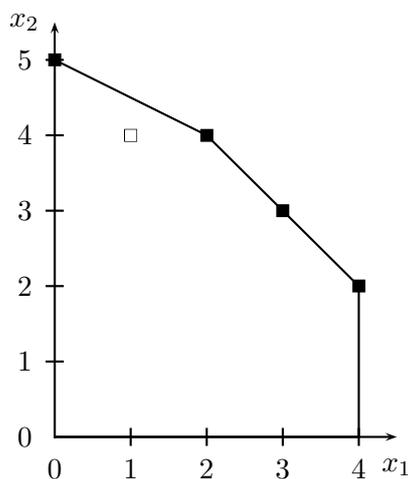


FIG. 1.3 – Solutions supportées et non supportées

- 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  $ED_{x'} \cap D = \{x'\}$ , tandis que le

point  $x^* = (1, 1)$  n'est pas efficace car  $ED_{x^*} \cap D \neq \{x^*\}$ .

- Deux solutions faiblement non dominées sont détectées, c'est les points  $(4, -4)$  et  $(4, -3)$ .
- Les solutions efficaces supportées sont :  $\{(4, 2), (3, 3), (2, 4), (0, 5)\}$ . Une solution efficace non supportée est détectée, c'est le point  $(1, 4)$ .
- Le point idéal est le point :  $(4, 5)$ .
- Il existe trois matrices des gains :  $\begin{pmatrix} 4 & 0 \\ -4 & 5 \end{pmatrix}$ ,  $\begin{pmatrix} 4 & 0 \\ -3 & 5 \end{pmatrix}$ ,  $\begin{pmatrix} 4 & 0 \\ -2 & 5 \end{pmatrix}$ .
- Le point nadir est le point de coordonnées :  $\eta = (0, -4)$

## 1.6 Conclusion

Dans ce chapitre, nous avons défini quelques concepts importants de l'optimisation multiobjectif discrète et utiles à la compréhension des méthodes exactes exposées dans le suivant chapitre.

## Quelques méthodes appliquées en optimisation multiobjectif discrète

### 2.1 Introduction

Nombreuses stratégies ont été proposées, consistant à caractériser totalement ou partiellement l'ensemble des solutions efficaces du problème de programmation linéaire multiobjectif en nombres entiers. Parmi ces méthodes, nous citons : méthode de D. Klein & E. Hannan [21], méthode de A. Crema & J. Sylva [35], méthode de R. Gupta & R. Malhotra [18] (première et deuxième procédure), méthode de M. Abbas & M. Moulai [1] et finalement méthode de M. Abbas & D. Chaabane [2] lesquelles nous allons présenter en détail dans le présent chapitre.

Considérons le problème de programmation linéaire multiobjectif discrète suivant :

$$(P) \text{ "Max" } \{z^i = c^i x, i = \overline{1, r} | x \in D\}$$

Où :

- $D = \{x \in \mathbb{Z}^n | Ax = b, x \geq 0\}$ , il est supposé borné et non vide.
- $A \in \mathbb{Z}^{m \times n}$ ,  $b \in \mathbb{Z}^m$  et  $C = (c^i)_{i=\overline{1, r}} \in \mathbb{Z}^{r \times n}$ .

### 2.2 Méthode de D. Klein & E. Hannan [21]

La technique proposée par les auteurs peut être utilisée aussi bien pour identifier l'ensemble de toutes les solutions efficaces que pour en caractériser une partie seulement. Elle consiste à résoudre progressivement une séquence de programmes linéaires

mono objectif en nombres entiers avec des contraintes ajoutées à chaque étape. Les contraintes supplémentaires éliminent les solutions efficaces déjà trouvées, et font en sorte que les nouvelles solutions générées soient efficaces.

### Algorithme

**Étape 1.** Résoudre le problème  $(P_1)$  défini comme suit :

$$(P_1) \text{ Max } \{z^s = c^s x | x \in D\}$$

**Cas 1.** Si la solution optimale de  $(P_1)$ , soit  $x^1$ , est unique alors elle est efficace pour  $(P)$ .

**Cas 2.** Sinon, déterminer toutes les solutions alternatives et à  $x^1$  et par comparaison deux à deux des vecteurs critère associés, garder uniquement celles qui sont efficaces pour construire l'ensemble  $ESE(P_1)$  des solutions efficaces générées à l'étape 1.

**Étape générale j.** A l'étape  $j$ , c'est le problème  $(P_j)$  qui est résolu, il est défini comme suit :

$$(P_j) \left\{ \begin{array}{l} \text{Max } z^s = c^s x \\ x \in D \\ \bigcap_{i=1}^q \left( \bigcup_{\substack{k=1 \\ k \neq s}}^r c^k x \geq c^k y^i + f_k \right) \end{array} \right. \quad (*)$$

Avec :

- L'indice  $s$  est pris arbitrairement dans  $\{1, \dots, r\}$ .
- $f_k \geq 1$  et il est entier pour  $k = \overline{1, r}$ ,  $k \neq s$ .
- $y^i$  ( $i = \overline{1, q}$ ) les points efficaces obtenus aux étapes  $1, \dots, j - 1$ .

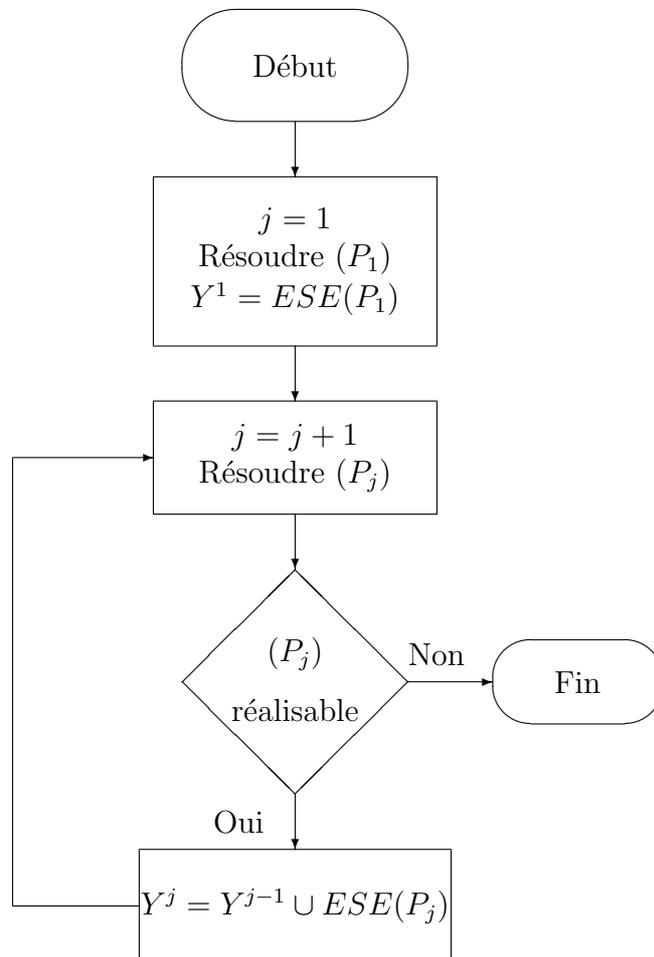
Si  $ESE(P_j)$  est l'ensemble des solutions efficaces obtenues à l'étape  $j$  et  $Y^j$  l'ensemble des points efficaces accumulés à la fin de l'étape  $j$ , alors  $Y^j = Y^{j-1} \cup ESE(P_j)$  pour  $j \geq 2$  avec  $Y^1 = ESE(P_1)$ .

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

L'ensemble de contraintes (\*) assure que les solutions de  $(P_j)$  seront meilleurs que toutes les solutions dans  $Y^{j-1}$ , pour au moins un critère  $k \neq s$ . Il est clair que la procédure est finie étant donné qu'on élimine au moins une solution admissible à chaque étape et qu'il en existe un nombre finie ( $D$  est supposé borné).

Lorsqu'à chaque étape  $j$ ,  $f_k = 1, \forall k = \overline{1, r}, k \neq s$ , la procédure fournit l'ensemble de toutes les solutions efficaces du problème. Cependant, si  $f_k > 1$  pour certains  $k, k = \overline{1, r}, k \neq s$ , seulement un sous ensemble de points efficaces est généré.

### Organigramme de la méthode



**Théorème 2.1** ([21]). *Si le problème  $(P_j)$  possède un ensemble de solutions optimales  $X_j^*$ , alors le sous ensemble de solutions efficaces dans  $X_j^*$ , l'ensemble  $ESE(P_j)$ , est efficace pour  $(P)$ .*

**Corrolaire 2.1** ([21]). *Si la solution optimale de  $(P_j)$  est unique, alors elle est efficace pour  $(P)$ .*

**Remarque.** Le nombre de sous problèmes créés par la méthode à chaque étape  $j$  croit exponentiellement en fonction de  $|Y^{j-1}|$ , il est égal à  $(r-1)^{|Y^{j-1}|}$ . C'est à dire plus la cardinalité de  $Y^{j-1}$  est grande plus ce nombre explose.

Nous présentons par la suite un exemple numérique permettant d'illustrer le fonctionnement de la méthode.

### Exemple illustratif

Considérons le problème à trois objectifs suivant :

$$(P) \begin{cases} \text{"Max"} (x_1 + x_2, 3x_1 - 2x_2, -x_1 + 2x_2) \\ (x_1, x_2) \in D \end{cases}$$

Où :

$$D = \left\{ (x_1, x_2) \in \mathbb{Z}_+^2 \left| \begin{array}{l} x_1 + x_2 \leq 4 \\ x_1 \leq 3 \\ x_2 \leq 3 \end{array} \right. \right\}$$

L'étape d'initialisation de la procédure requiert la (ou les) solution(s) de  $(P_1)$ . Prenons  $s = 1$  et résoudre :

$$(P_1) \begin{cases} \text{Max } x_1 + x_2 \\ (x_1, x_2) \in D \end{cases}$$

La solution optimale de  $(P_1)$  est  $x^1 = (3, 1)$  qui n'est pas unique. Les solutions qui lui sont alternatives sont :  $x^2 = (2, 2)$  et  $x^3 = (1, 3)$ .

$$Y^1 = ESE(P_1) = \{x^1, x^2, x^3\}.$$

Dans l'étape 2 de la procédure, on résout le problème  $(P_2)$  avec  $f_k = 1, \forall k \in \{2, 3\}$  :

$$(P_2) \left\{ \begin{array}{l} \text{Max } x_1 + x_2 \\ (x_1, x_2) \in D \\ \text{et } (3x_1 - 2x_2 \geq 8 \text{ ou } -x_1 + 2x_2 \geq 0) \\ \text{et } (3x_1 - 2x_2 \geq 3 \text{ ou } -x_1 + 2x_2 \geq 3) \\ \text{et } (3x_1 - 2x_2 \geq -2 \text{ ou } -x_1 + 2x_2 \geq 6) \end{array} \right.$$

Le problème  $(P_2)$  contient  $(2)^3$  sous problèmes à traiter séparément, les résultats obtenues sont représentés dans le tableau suivant :

$(P_{21}) \left\{ \begin{array}{l} \text{Max } x_1 + x_2 \\ (x_1, x_2) \in D \\ 3x_1 - 2x_2 \geq 8 \\ 3x_1 - 2x_2 \geq 3 \\ 3x_1 - 2x_2 \geq -2 \end{array} \right.$	$(P_{22}) \left\{ \begin{array}{l} \text{Max } x_1 + x_2 \\ (x_1, x_2) \in D \\ 3x_1 - 2x_2 \geq 8 \\ 3x_1 - 2x_2 \geq 3 \\ -x_1 + 2x_2 \geq 6 \end{array} \right.$	$(P_{23}) \left\{ \begin{array}{l} \text{Max } x_1 + x_2 \\ (x_1, x_2) \in D \\ 3x_1 - 2x_2 \geq 8 \\ -x_1 + 2x_2 \geq 3 \\ 3x_1 - 2x_2 \geq -2 \end{array} \right.$
$x^4 = (3, 0)$ ( <i>unique</i> )	<i>irréalizable</i>	<i>irréalizable</i>
$(P_{24}) \left\{ \begin{array}{l} \text{Max } x_1 + x_2 \\ (x_1, x_2) \in D \\ 3x_1 - 2x_2 \geq 8 \\ -x_1 + 2x_2 \geq 3 \\ -x_1 + 2x_2 \geq 6 \end{array} \right.$	$(P_{25}) \left\{ \begin{array}{l} \text{Max } x_1 + x_2 \\ (x_1, x_2) \in D \\ -x_1 + 2x_2 \geq 0 \\ 3x_1 - 2x_2 \geq 3 \\ 3x_1 - 2x_2 \geq -2 \end{array} \right.$	$(P_{26}) \left\{ \begin{array}{l} \text{Max } x_1 + x_2 \\ (x_1, x_2) \in D \\ -x_1 + 2x_2 \geq 0 \\ 3x_1 - 2x_2 \geq 3 \\ -x_1 + 2x_2 \geq 6 \end{array} \right.$
<i>irréalizable</i>	$x^5 = (2, 1)$ ( <i>unique</i> )	<i>irréalizable</i>
$(P_{27}) \left\{ \begin{array}{l} \text{Max } x_1 + x_2 \\ (x_1, x_2) \in D \\ -x_1 + 2x_2 \geq 0 \\ -x_1 + 2x_2 \geq 3 \\ 3x_1 - 2x_2 \geq -2 \end{array} \right.$	$(P_{28}) \left\{ \begin{array}{l} \text{Max } x_1 + x_2 \\ (x_1, x_2) \in D \\ -x_1 + 2x_2 \geq 0 \\ -x_1 + 2x_2 \geq 3 \\ -x_1 + 2x_2 \geq 6 \end{array} \right.$	
$x^6 = (1, 2)$ ( <i>unique</i> )	$x^7 = (0, 3)$ ( <i>unique</i> )	

TAB. 2.1 – Résultats

D'après le tableau (Tab. 2.1),  $ESE(P_2) = \{x^4, x^5, x^6, x^7\}$  et  $Y^2 = Y^1 \cup ESE(P_2) =$

$\{x^1, x^2, x^3, x^4, x^5, x^6, x^7\}$ .

L'étape 3 de la procédure requiert la résolution du problème  $(P_3)$  défini comme suit :

$$(P_3) \begin{cases} \text{Max} & x_1 + x_2 \\ (x_1, x_2) & \in D \\ \bigcap_{i=1}^{q=7} & \left( \bigcup_{\substack{k=1 \\ k \neq 1}}^{r=3} c^k x \geq c^k y^i + 1 \right) \end{cases}$$

Où :  $y^i \in Y^2$ ,  $i = \overline{1, 7}$ .

La résolution de  $(P_3)$  entraîne la résolution de  $(2)^7$  sous problèmes qui sont dans cette étape tous irréalisables. Donc la procédure s'arrête et l'ensemble des solutions efficaces du problème  $(P)$  est  $Y^3 = Y^2 = \{(3, 1), (2, 2), (1, 3), (3, 0), (2, 1), (1, 2), (0, 3)\}$ .

### 2.3 Méthode de R. Gupta & R. Malhotra [18] (2<sup>ème</sup> procédure)

Cette méthode est une variante de celle proposée par Klein & Hannan [21]. Elle consiste à déterminer l'ensemble de toutes les solutions efficaces où les auteurs ont pu réduire le nombre de contraintes supplémentaires rajoutées à chaque étape de la procédure.

En réalité, la démarche adoptée par les auteurs, ne réussit pas toujours à donner toutes les solutions efficaces du problème de programmation linéaire discrète à objectifs multiples, car on a pu trouver un contre exemple qui montre que l'algorithme s'arrête avant que toutes les solutions efficaces ne soient produites.

#### Développement de la méthode

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

$$(P_1) \text{Max} \{z^1 = c^1 x \mid x \in D\}$$

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

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

$$(P_{j+1}) \begin{cases} \text{Max} & z^1 = c^1 x \\ & x \in D \\ & c^1 x \leq \max \{c^1 y - 1 \mid y \in ESE(P_j)\} = c^1 y^s - 1 \\ & c^i x \geq c^i y^s + 1 \text{ pour au moins un } i, \quad i = \overline{2, r} \end{cases}$$

Où  $ESE(P_j)$  représente l'ensemble de toutes les solutions potentiellement efficaces obtenues à l'étape  $j$ . S'il y a plus d'un  $y^s$  donnant le maximum de  $c^1 y - 1$ ,  $y \in ESE(P_j)$  alors, sélectionner  $y^s$  arbitrairement. Enregistrer toutes les solutions potentiellement efficaces obtenues à l'étape  $j + 1$  dans  $ESE(P_{j+1})$ . La valeur de  $z^1$  a diminué de celle enregistrée à l'étape  $j$  d'au moins une unité et la valeur d'au moins un critère  $i$ ,  $i = \overline{2, r}$  s'est améliorée grâce aux contraintes supplémentaires.

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

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

**Remarque.** Selon le premier test d'arrêt proposé par Gupta & Malhotra [18], si toutes les solutions de  $(P_n)$  ne sont pas efficaces pour le problème  $(P)$  alors, toutes les étapes ultérieures donnent des solutions non efficaces. En fait, nous montrons à travers un contre exemple que ceci n'est pas vrai et qu'il est possible de continuer à chercher d'autres solutions efficaces.

### Contre exemple et analyse critique

Soit à résoudre le problème à trois objectifs suivant :

$$(P) \begin{cases} \text{"Max"} & (6x_1 - 2x_2 + 6x_3, 8x_1 + 9x_2 + 5x_3, -5x_2 + 9x_3) \\ & (x_1, x_2, x_3) \in D \end{cases}$$

Où :

$$D = \left\{ (x_1, x_2, x_3) \in \mathbb{Z}_+^3 \left| \begin{array}{l} 9x_1 + 7x_2 + 10x_3 \leq 52 \\ 4x_1 + 2x_2 + 3x_3 \leq 18 \\ x_1 + 6x_2 + 5x_3 \leq 24 \end{array} \right. \right\}$$

$$\boxed{j = 1}$$

$$(P_1) \begin{cases} \text{Max} & 6x_1 - 2x_2 + 6x_3 \\ & (x_1, x_2, x_3) \in D \end{cases}$$

La résolution de  $(P_1)$  donne les solutions optimales :  $x^1 = (1, 0, 4)$ ,  $x^2 = (2, 0, 3)$  et  $x^3 = (3, 0, 2)$ . Les premiers triplets non dominés sont  $(z_1^1, z_1^2, z_1^3) = (30, 28, 36)$ ,  $(z_2^1, z_2^2, z_2^3) = (30, 31, 27)$  et  $(z_3^1, z_3^2, z_3^3) = (30, 34, 18)$ .

$$ESE(P_1) = \{(1, 0, 4), (2, 0, 3), (3, 0, 2)\}.$$

$$\boxed{j = 2}$$

$$\max \{6x_1 - 2x_2 + 6x_3 - 1 \mid (x_1, x_2, x_3) \in ESE(P_1)\} = 29.$$

Comme  $z_1^1 = z_2^1 = z_3^1$  alors, choisir n'importe quelle solution dans  $ESE(P_1)$ , soit  $x^1$ .

$$(P_2) \begin{cases} \text{Max} & 6x_1 - 2x_2 + 6x_3 \\ & (x_1, x_2, x_3) \in D \\ & 6x_1 - 2x_2 + 6x_3 \leq 29 \\ & \text{et } (8x_1 + 9x_2 + 5x_3 \geq 29 \text{ ou } -5x_2 + 9x_3 \geq 37) \end{cases}$$

Les solutions de  $(P_2)$  sont :  $x^4 = (3, 0, 1)$  correspondant à  $(z_4^1, z_4^2, z_4^3) = (24, 29, 9)$  et  $x^5 = (4, 0, 0)$  correspondant à  $(z_5^1, z_5^2, z_5^3) = (24, 32, 0)$ .

$ESE(P_2) = \{(3, 0, 1), (4, 0, 0)\}$  et toutes les solutions dans  $ESE(P_2)$  ne sont pas efficaces. Donc, d'après Gupta & Malhotra [18], toutes les prochaines étapes donnent

des solutions non efficaces, ainsi la procédure s'arrête avec :

$$ESE(P) = \{(1, 0, 4), (2, 0, 3), (3, 0, 2)\}$$

Or les points  $(1, 1, 3)$ ,  $(2, 1, 2)$ ,  $(2, 2, 2)$ ,  $(4, 1, 0)$ ,  $(3, 3, 0)$  et  $(3, 1, 1)$  qui sont efficaces ne sont pas détectés. Ceci indique que si à une étape donnée  $l$ , toutes les solutions dans  $ESE(P_l)$  ne sont pas efficaces, les étapes qui suivent l'étape  $l$  peuvent produire des solutions efficaces. Pour notre exemple, l'étape 3 donne :

$\boxed{j = 3}$   $\max \{6x_1 - 2x_2 + 6x_3 - 1 | (x_1, x_2, x_3) \in ESE(P_2)\} = 23$  et puisque  $z_4^1 = z_5^1$  alors, choisir arbitrairement une solution dans  $ESE(P_2)$ , soit  $x^4$ .

$$(P_3) \begin{cases} \text{Max } 6x_1 - 2x_2 + 6x_3 \\ (x_1, x_2, x_3) \in D \\ 6x_1 - 2x_2 + 6x_3 \leq 23 \\ \text{et } (8x_1 + 9x_2 + 5x_3 \geq 30 \text{ ou } -5x_2 + 9x_3 \geq 10) \end{cases}$$

Les solutions de  $(P_3)$  sont  $x^6 = (1, 1, 3)$ ,  $x^7 = (2, 1, 2)$ ,  $x^8 = (3, 1, 1)$  et  $x^9 = (4, 1, 0)$  relatives aux vecteurs critères  $(z_6^1, z_6^2, z_6^3) = (22, 32, 22)$ ,  $(z_7^1, z_7^2, z_7^3) = (22, 35, 13)$ ,  $(z_8^1, z_8^2, z_8^3) = (22, 38, 4)$  et  $(z_9^1, z_9^2, z_9^3) = (22, 41, -5)$  respectivement.

$ESE(P_3) = \{(1, 1, 3), (2, 1, 2), (3, 1, 1), (4, 1, 0)\}$ . Les solutions dans  $ESE(P_3)$  sont toutes efficaces ce qui contredit le fait que toutes les étapes après l'étape 2 donnent des solutions inefficaces.

De la même manière que précédemment, nous poursuivons la recherche jusqu'à une étape  $n$  où le problème  $(P_n)$  soit irréalisable.

$$ESE(P) = \{(1, 0, 4), (2, 0, 3), (3, 0, 2), (1, 1, 3), (2, 1, 2), (3, 1, 1), (4, 1, 0), (2, 2, 2), (3, 3, 0)\}$$

Donc l'étape finale de la méthode devient :

**Étape finale  $n$ .** La procédure prend fin dans le cas où  $(P_n)$  n'est pas réalisable.

## 2.4 Méthode de A. Crema & J. Sylva [35]

La méthode développée par Sylva & Crema [35] est aussi une variante de celle de Klein & Hannan [21] étudiée précédemment. Son principe repose sur la résolution d'une succession de programmes linéaires en nombres entiers optimisant à chaque étape une combinaison positive des critères. Un ensemble de contraintes est rajouté à chaque fois assurant la détection d'une nouvelle solution efficace. A la fin, la méthode fournit l'ensemble de toutes les solutions non dominées du problème de programmation linéaire discrète à objectifs multiples.

### Algorithme

**Étape 1.** Après avoir fixé le vecteur poids  $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_r)$  à des valeurs strictement positives, la première étape de l'algorithme consiste en la résolution du problème :  $(P_1) \text{ Max } \left\{ \sum_{i=1}^r \lambda_i c^i x \mid x \in D \right\}$ .

Deux cas se présentent :

**Cas 1.** Si  $(P_1)$  n'admet pas de solutions, alors  $(P)$  l'est aussi.

**Cas 2.** Sinon, une solution  $x^1$  est trouvée et elle est efficace.

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

Après  $k$  étapes du processus :

**Cas 1.** Si  $(P_k)$  est irréalisable, alors l'algorithme prend fin.

**Cas 2.** Sinon, une nouvelle solution efficace, soit  $x^k$ , est trouvée et le nouveau problème  $(P_{k+1})$  est défini à partir de  $(P_k)$  en lui éliminant toutes les solutions vérifiant  $c^i x \leq c^i x^k, \forall i = \overline{1, r}$ . Ceci peut être traduit par le rajout de contraintes suivantes :

$$\begin{cases} c^i x \geq (c^i x^k + 1)y_i^k - M_i(1 - y_i^k) & i = \overline{1, r} \\ \sum_{i=1}^r y_i^k \geq 1 & y_i^k \in \{0, 1\} & i = \overline{1, r} \end{cases}$$

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

**Étape générale (k+1).** Résoudre le problème  $(P_{k+1})$  :

$$(P_{k+1}) \left\{ \begin{array}{l} \text{Max} \quad \sum_{i=1}^r \lambda_i c^i x \\ x \in D \\ c^i x \geq (c^i x^j + 1)y_i^j - M_i(1 - y_i^j) \quad i = \overline{1, r} \quad j = \overline{1, k} \\ \sum_{i=1}^r y_i^j \geq 1 \quad y_i^j \in \{0, 1\} \quad i = \overline{1, r} \quad j = \overline{1, k} \end{array} \right.$$

**Étape finale n.** La procédure prend fin dans le cas où  $(P_n)$  est irréalisable.

**Proposition 2.1** ([35]). Soient  $x^1, x^2, \dots, x^k$  des solutions efficaces du problème  $(P)$ .

Posons  $\Delta_j = \{x \in \mathbb{Z}^n \mid c^i x \leq c^i x^j \quad \forall i = \overline{1, r}\}$  ( $j = \overline{1, k}$ ). Si  $x^*$  est efficace pour le problème :

$$(Q) \left\{ \begin{array}{l} \text{''Max''} \quad c^i x \quad i = \overline{1, r} \\ x \in \left( D - \bigcup_{j=1}^k \Delta_j \right) \end{array} \right.$$

alors  $x^*$  est une solution efficace pour le problème  $(P)$ . De plus, si le problème  $(Q)$  est irréalisable, alors  $\{(c^i x^j)_{i=\overline{1, r}} \quad j = \overline{1, k-1}\}$  est l'ensemble de toutes les solutions non dominées du problème  $(P)$ .

**Corrolaire 2.2** ([35]). Soient  $x^1, x^2, \dots, x^k$  des solutions efficaces du problème  $(P)$  et

$\Delta_j = \{x \in \mathbb{Z}^n \mid c^i x \leq c^i x^j \quad \forall i = \overline{1, r}\}$  ( $j = \overline{1, k}$ ). Si  $x^*$  est une solution optimale pour

le problème mono objectif :  $\text{Max} \left\{ \sum_{i=1}^r \lambda_i c^i x \mid x \in \left( D - \bigcup_{j=1}^k \Delta_j \right) \right\}$  pour certaines va-

leurs du vecteur  $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_r)$ ,  $\lambda > 0$ , alors  $x^*$  est une solution efficace pour le problème  $(P)$ .

**Remarque.** Pour les problèmes de grande taille, la détermination de l'ensemble de toutes les solutions non dominées devient très coûteuse en terme de temps de calcul. Pour cela, une étape d'interaction avec le décideur peut être intégrée à la procédure. Cette étape a pour objectif d'éliminer des solutions efficaces que le décideur juge

insatisfaisantes. Dans ce cas le problème  $(P_{k+1})$  devient :

$$(P_{k+1}) \left\{ \begin{array}{l} \text{Max} \quad \sum_{i=1}^r \lambda_i c^i x \\ x \in D \\ c^i x \geq (c^i x^j + f_i) y_i^j - M_i (1 - y_i^j) \quad i = \overline{1, r}, \quad j = \overline{1, k} \\ \sum_{i=1}^r y_i^j \geq 1 \quad y_i^j \in \{0, 1\} \quad i = \overline{1, r} \quad j = \overline{1, k} \end{array} \right.$$

Où  $f_i$  représente l'amélioration minimale dans la  $i^{\text{ème}}$  fonction objectif fixée par le décideur,  $f_i > 1$  (entier).

**Remarque.** Le nombre de problèmes de programmation linéaire discrète devant être résolu est donné par le nombre de solutions non dominées plus un dernier problème irréalisable. En réalité, cette méthode est fortement liée au nombre de critères et de solutions non dominées du problème étudié. Car, généralement, plus le nombre de critères est grand, plus le cône engendré est restreint, ce qui fait augmenter le nombre de solutions non dominées du problème entraînant le traitement d'un nombre considérable de problèmes de programmation linéaire discrète. Cette remarque a été validée par une expérimentation numérique de la méthode au chapitre 4.

**Remarque.** Afin de s'assurer du bon fonctionnement de la méthode (i.e. trouver toutes les solutions non dominées du problème), le paramètre  $(-M_i)$ ,  $i = \overline{1, r}$  doit être choisi convenablement. Par exemple, en cas d'existence d'un indice  $i = \overline{1, r}$  tel que  $c^i$  est à composantes positives, alors prendre  $M_i = 0$ . Sinon, attribuer à chaque critère  $i$  ayant des composantes négatives un minorant dont la valeur est relativement petite. Cependant, si  $-M_i$  est plus grand que la borne inférieure du critère  $i$  sur le domaine en question, une partie seulement de l'ensemble des solutions non dominées est trouvée.

Pour les méthodes qui vont suivre, nous utilisons les notations suivantes :

On considère le problème :

$$(P_1) \text{ Max } \{c^1 x | x \in D\}$$

dont le problème relaxé est :

$$(R) \text{ Max } \{c^1 x | x \in S\}$$

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

Notons qu'à la place du problème  $(P_1)$ , on pourra considérer d'une manière analogue le problème  $(P_i)$ ,  $i = \overline{2, r}$  qui maximise  $c^i x$  sur  $D$ .

On définit pour  $k \geq 1$  :

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

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

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

- $x^k$  : la  $k^{\text{ème}}$  solution optimale entière du problème  $(P_1)$  obtenue sur  $S_k$ .
- $B_k$  : est une base de  $S_k$ .
- $N_k$  : ensemble des indices des variables hors base de  $x^k$ .
- $\widehat{A}_k = (A_k^{B_k})^{-1} A_k = (\widehat{a}_{lig(i)k}^j)_{\substack{i=\overline{1, m_k} \\ j=\overline{1, n_k}}}$ , où "lig" est une application linéaire définie par :

$$lig : B_k \longrightarrow \{1, \dots, m_k\}$$

$$i \longrightarrow lig(i) = \text{position ligne de la variable de base } x_i^k \text{ dans le}$$

tableau du simplexe à l'étape  $k$ .

- $\widehat{C}_k = C_k - \pi_k A_k$ , avec  $\pi_k = (C_k^{B_k})(A_k^{B_k})^{-1}$ .
- $\Gamma_k = \{j \in N_k | (\widehat{c}_k^1)_j < 0 \text{ et } (\widehat{c}_k^i)_j > 0 \text{ pour au moins un } i, i = \overline{2, r}\}$ .
- $\Omega_k = \{j \in N_k | (\widehat{c}_k^1)_j = 0\}$ .

- $\Psi_k = \{j \in N_k \mid (\widehat{c}_k^1)^j < 0 \text{ et } (\widehat{c}_k^i)_j < 0 \text{ pour au moins un } i, i = \overline{2, r}\}$ .

**Définition 2.1.** Une arête  $E^{j_k}$ ,  $j_k \in N_k$  incidente à  $x^k$  est définie comme étant l'ensemble :

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

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

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

On note par  $nb_{j_k}$ , le nombre de solutions entières sur l'arête  $E^{j_k}$  y compris  $x^k$ .

## 2.5 Méthode de R. Gupta & R. Malhotra [18] (1<sup>ère</sup> procédure)

Cette méthode a été proposée par R. Gupta & R. Malhotra [18], ayant pour but de trouver toutes les solutions efficaces du problème de programmation linéaire discrète à objectifs multiples. Mais il s'est avéré qu'une erreur au niveau du second test d'arrêt empêche dans certains cas l'algorithme de donner tous les points efficaces du problème étudié. Un contre exemple a été présenté par M. Moulai [26] ainsi que par D. Chaabane [7].

Nous rapportons dans ce qui suit, quelques résultats théoriques sur la méthode :

**Théorème 2.2** ([18]). *Toutes les solutions entières réalisables du problème  $(P_1)$  alternatives à  $x^1$  sur l'arête  $E^{j_1}$ ,  $j_1 \in \Gamma_1$  de la région  $S$  (ou la région tronquée de  $S$ ) et émanant de  $x^1$ , appartiennent au demi espace ouvert :*

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

**Théorème 2.3** ([18]). *Une solution entière réalisable du problème  $(P_1)$  qui n'est pas sur l'arête  $E^{j_1}$ ,  $j_1 \in \Gamma_1$  à travers  $x^1$ , appartient au demi espace fermé :*

$$\sum_{j \in N_1 \setminus \{j_1\}} x_j \geq 1 \quad (2.1)$$

**Corrolaire 2.3** ([18]). *Une solution entière réalisable du problème  $(P_1)$  qui n'est pas sur l'arête  $E^{j_k}$ ,  $j_k \in \Gamma_k$ ,  $k \geq 2$  à travers  $x^k$  dans la région tronquée de  $S$ , appartient au demi espace fermé :*

$$\sum_{j \in N_k \setminus \{j_k\}} x_j \geq 1 \quad (2.2)$$

**Remarque.** Les coupes (2.1) et (2.2) sont une généralisation de la coupe de Dantzig [12]. Dans le cas où  $\Gamma_1$  ou  $\Gamma_k$  sont vides, alors la coupe correspondante se réduit à une coupe de Dantzig. L'avantage d'utiliser de telles coupes, c'est de tronquer toute une arête au lieu d'un seul point.

### Algorithme

Notons par  $SND_0(P)$  l'ensemble des solutions non dominées du problème  $(P)$  générées à l'étape 1, et par  $SND(P)$  l'ensemble des solutions potentiellement non dominées générées aux étapes  $k$ ,  $k \geq 2$ .

**Étape 1.** Résoudre le problème  $(P_1)$ .

**Cas 1.** Si la solution optimale de  $(P_1)$  est unique, soit  $x^1$ , alors le vecteur critère  $(z_1^1, z_1^2, \dots, z_1^r)$  qui lui est associé est enregistré dans  $SND_0(P)$  comme étant le premier  $r$ -uplet non dominé.

**Cas 2.** Si la solution trouvée n'est pas unique, alors déterminer toutes les solutions qui lui sont alternatives, et par comparaison deux à deux des vecteurs critères associés, éliminer ceux qui sont dominés pour construire l'ensemble  $SND_0(P)$  des premiers vecteurs non dominés.

$(z_1^1, z_1^2, \dots, z_1^r)$  est choisit comme étant le premier  $r$ -uplet non dominé pris dans l'ordre lexicographique (i.e. le vecteur qui a la plus grande valeur de

$z^2$ , en cas d'égalité choisir celui qui a la plus grande valeur de  $z^3$ , et ainsi de suite), soit  $x^1$  la solution correspondante.

**Étape 2.** Construire l'ensemble  $\Gamma_1$ , choisir un indice quelconque  $j_1 \in \Gamma_1$  et trouver le rapport minimum  $\theta$  de l'opération pivot.

**Cas 1.** Si  $\theta < 1$ , alors choisir un autre  $j_1 \in \Gamma_1$ .

**Cas 2.** Si  $\theta \geq 1$ , déterminer toutes les solutions entières se trouvant sur l'arête  $E^{j_1}$ , évaluer les  $r$  critères en chacune d'elles et éliminer ceux qui sont dominés pour construire l'ensemble  $SND(P)$  des solutions potentiellement non dominées générées à l'étape 2.

Appliquer la coupe : 
$$\sum_{j \in N_1 \setminus \{j_1\}} x_j \geq 1.$$

Si pour tout  $j_1 \in \Gamma_1$  on a,  $\theta < 1$ , alors choisir arbitrairement un  $j_1 \in \Gamma_1$  et appliquer la coupe 
$$\sum_{j \in N_1 \setminus \{j_1\}} x_j \geq 1.$$

Utiliser la méthode duale du simplexe et des coupes successives de Gomory si nécessaire pour obtenir une solution entière  $x^2$  dans la région tronquée  $S_2$ . Mettre à jour  $SND(P)$ .

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

Tronquer l'arête  $E^{j_{k-1}}$  par la coupe : 
$$\sum_{j \in N_{k-1} \setminus \{j_{k-1}\}} x_j \geq 1$$
 et chercher de nouveau une solution entière dans la région tronquée  $S_k$ , soit  $x^k$ . Mettre à jour l'ensemble  $SND(P)$ .

**Étape finale n+1.** L'algorithme prend fin dans l'un des cas suivants :

1.  $\Gamma_n = \emptyset$  et  $(\hat{c}_n^1)_j < 0 \forall j \in N_n$ .
2.  $\Gamma_n \neq \emptyset$  mais pour tout  $j_n \in N_n$ , les solutions entières se trouvant sur l'arête  $E^{j_n}$  ne sont pas efficaces.

L'ensemble courant des solutions non dominées  $SND(P)$  union  $SND_0(P)$  enregistré à l'étape 1, fournit l'ensemble de toutes les solutions non dominées du problème.

Nous donnons ci-après un autre contre exemple permettant de localiser la défaillance de la méthode.

**Contre exemple**

Considérons le problème bi objectif suivant :

$$(P) \text{ "Max" } \{(x_1, x_2) | 2x_1 + x_2 \leq 12, x_1 + 2x_2 \leq 9, x_1 + 3x_2 \leq 12, (x_1, x_2) \in \mathbb{Z}_+^2\}$$

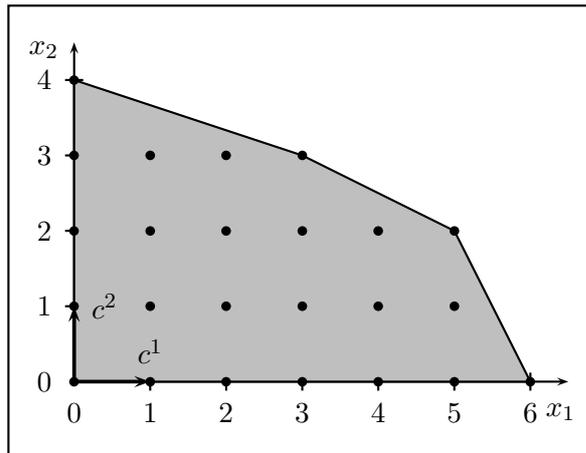


FIG. 2.1 – Domaine d’admissibilité

$k=1$

La résolution de  $(P_1)$ , donne l’unique solution  $x^1 = (6, 0)$ . Les résultats sont représentés sur le tableau qui suit :

<i>base</i>	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_B$
$x_1$	1	1/2	1/2	0	0	6
$x_4$	0	3/2	-1/2	1	0	3
$x_5$	0	5/2	-1/2	0	1	6
$-z_1$	0	-1/2	-1/2	0	0	6
$-z_2$	0	1	0	0	0	0

TAB. 2.2 – Tableau 1

Le premier couple non dominé est  $(z_1^1, z_1^2) = (6, 0)$ . Par suite,  $SND_0(P) = \{(6, 0)\}$ .  $N_1 = \{2, 3\}$ .

$$\boxed{k=2}$$

$\Gamma_1 = \{2\}$ . Prendre  $j_1 = 2$  et l'arête  $E^2$  est explorée pour d'éventuelles solutions entières.

$\theta = \min \left\{ \frac{12}{1}, \frac{6}{3}, \frac{12}{5} \right\} = 2$ , donc la seule solution entière sur l'arête  $E^2$  est  $(5, 2)$  et  $SND(P) = \{(5, 2)\}$ .

L'arête  $E^2$  est tronquée par la coupe :  $x_3 \geq 1$ , (i.e.  $x_3 - x_6 = 1$  ou bien  $-x_3 + x_6 = -1$ ).

Les coefficients de  $-x_3 + x_6 = -1$  sont insérés dans une nouvelle ligne du tableau (Tab. 2.2) et par application de la méthode duale du simplexe et des coupes de Gomory, on obtient le tableau suivant : ( $x_7$  provient d'une coupe de Gomory)

base	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_B$
$x_1$	1	0	0	0	0	0	1	5
$x_4$	0	0	0	1	0	-2	3	2
$x_5$	0	0	0	0	1	-3	5	4
$x_3$	0	0	1	0	0	-1	0	1
$x_2$	0	1	0	0	0	1	-2	1
$-z_1$	0	0	0	0	0	0	-1	5
$-z_2$	0	0	0	0	0	-1	2	1

TAB. 2.3 – Tableau 2

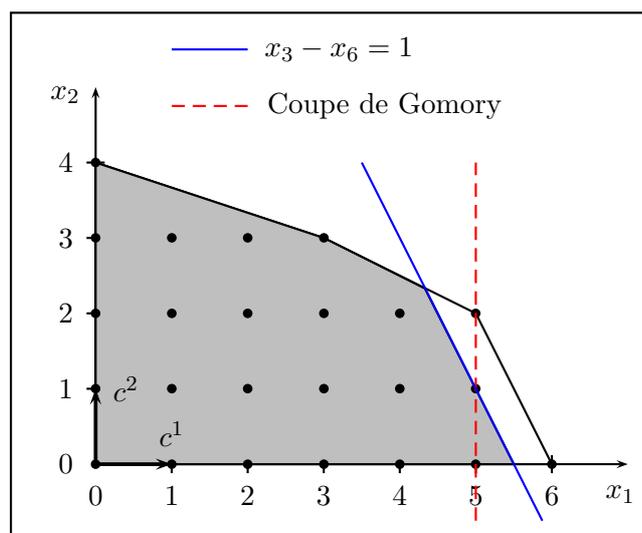


FIG. 2.2 – Domaine réduit (1)

D'après le tableau (Tab. 2.3), la solution entière optimale est  $x^2 = (5, 1)$  dont le couple  $(z_2^1, z_2^2) = (5, 1)$  est le vecteur critère correspondant.  $(5, 1)$  est dominé, donc  $SND(P) = \{(5, 2)\}$ .  $N_2 = \{6, 7\}$ .

$\boxed{k=3}$

$\Gamma_2 = \{7\}$ . Prendre  $j_2 = 7$  et l'arrête  $E^7$  est scannée pour des solutions entières possibles.

$\theta = \min \left\{ \frac{5}{1}, \frac{2}{3}, \frac{4}{5} \right\} = \frac{2}{3}$ . Puisque  $\theta < 1$ , il n'y a aucune solution entière sur l'arête  $E^7$ . Tronquer  $E^7$  par la coupe  $x_6 \geq 1$ , (i.e.  $x_6 - x_8 = 1$  ou bien  $-x_6 + x_8 = -1$ ).

La contrainte  $-x_6 + x_8 = -1$  est rajoutée au tableau (Tab. 2.3), l'algorithme dual du simplexe nous fournit le tableau suivant :

<i>base</i>	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_B$
$x_1$	1	0	0	0	0	0	1	0	5
$x_4$	0	0	0	1	0	0	3	-2	4
$x_5$	0	0	0	0	1	0	5	-3	7
$x_3$	0	0	1	0	0	0	0	-1	2
$x_2$	0	1	0	0	0	0	-2	1	0
$x_6$	0	0	0	0	0	1	0	-1	1
$-z_1$	0	0	0	0	0	0	-1	0	5
$-z_2$	0	0	0	0	0	0	2	-1	0

TAB. 2.4 – Tableau 3

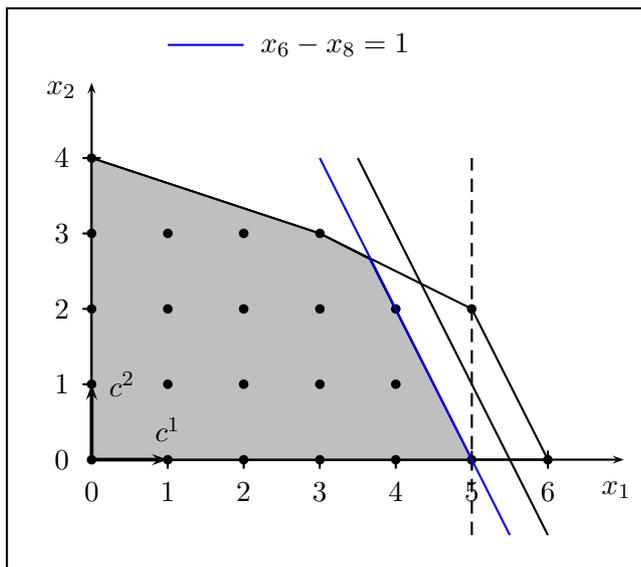


FIG. 2.3 – Domaine réduit (2)

D'après le tableau (Tab. 2.4), la solution optimale est  $(5, 0)$ , ce qui donne  $(z_3^1, z_3^2) = (5, 0)$  qui est dominé, donc  $SND(P)$  reste le même que précédemment.  $N_3 = \{7, 8\}$ .

$k=4$

$\Gamma_3 = \{7\}$ . Prendre  $j_3 = 7$ ,  $\theta = \min \left\{ 5, \frac{4}{3}, \frac{7}{5} \right\} = \frac{4}{3}$ . La solution entière  $(4, 2)$  étant la seule sur  $E^7$  et elle est relative au vecteur critère  $(4, 2)$  qui est dominé. Donc pour tout  $j_3$  dans  $\Gamma_3$ , les solutions entières sur  $E^{j_3}$  ne sont pas efficaces. Selon le second test d'arrêt proposé par Gupta & Malhotra [18] la procédure s'arrête et  $SND(P) = \{(6, 0), (5, 2)\}$ , c'est à dire l'ensemble des solutions efficaces est  $ESE = \{(6, 0), (5, 2)\}$ . Or, les points  $(0, 4)$  et  $(3, 3)$  qui sont efficaces ne sont pas détectés. Donc même si  $\Gamma_n \neq \emptyset$  et que toutes les solutions entières sur  $E^{j_n}$ ,  $j_n \in \Gamma_n$  produisent des  $r$ -uplets dominés, l'exploration du domaine doit poursuivre pour d'éventuelles solutions efficaces en tronquant une arête quelconque  $E^{j_n}$ ,  $j_n \in \Gamma_n$ .

## 2.6 Méthode de M. Abbas & M. Moulai [1]

Cette méthode, dénommée "MODILIM", a été proposée par M. Abbas & M. Moulai [1] pour la détermination de toutes les solutions efficaces du problème de pro-

grammation linéaire multiobjectif en nombres entiers. Elle peut être vue comme une alternative à celle de Gupta & Malhotra [18] (première procédure), où les auteurs ont proposé un autre test d'arrêt permettant à l'algorithme de fournir toutes les solutions efficaces.

### Développement de la méthode

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

**Étape 1.** Résoudre le problème  $(P_1)$  et trouver la solution optimale entière  $x^1$  sur  $S_1$ . Construire l'ensemble  $\Omega_1$ .

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

**Cas 1.** Si  $\Omega_1 = \emptyset$ ,  $x^1$  est l'unique solution optimale sur  $S_1$ . Soit  $(z_1^1, z_1^2, \dots, z_1^r)$  le vecteur critère correspondant, il est enregistré dans  $SND(P)$  comme étant le premier  $r$ -uplet non dominé.

Tronquer le point  $x^1$  par la coupe :

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

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

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

(a) Si  $\theta \geq 1$ , déterminer toutes les solutions entières alternatives à  $x^1$ , soient  $y_1^q$ ,  $q = \overline{2, nb_{j_1}}$  le long de l'arête  $E^{j_1}$  et mettre à jour  $SND(P)$ . Comme les solutions alternatives ont la même valeur de  $z^1$  que celle de  $x^1$ , le premier point potentiellement non dominé est choisit comme le  $r$ -uplet ayant la plus grande valeur de  $z^2$ , sinon choisir celui qui a la plus grande valeur de  $z^3$  et ainsi de suite jusqu'à l'obtention du premier

$r$ -uplet potentiellement non dominé.

Tronquer l'arête  $E^{j_1}$  par la coupe : 
$$\sum_{j \in N_1 \setminus \{j_1\}} x_j \geq 1.$$

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

(b) Si pour tout  $j_1 \in \Omega_1$ , on a  $\theta < 1$ , alors choisir un indice quelconque

$j_1 \in \Omega_1$  et appliquer la coupe : 
$$\sum_{j \in N_1 \setminus \{j_1\}} x_j \geq 1.$$

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

**Étape k.** ( $k \geq 3$ ) Choisir un indice  $j_{k-1} \in \Omega_{k-1}$  et explorer l'arête correspondante pour des possibles solutions entières  $y_{k-1}^q$ ,  $q = \overline{2, nb_{j_{k-1}}}$  alternatives à  $x^{k-1}$ . Mettre à jour l'ensemble  $SND(P)$ .

L'arête  $E^{j_{k-1}}$  est tronquée par la coupe : 
$$\sum_{j \in N_{k-1} \setminus \{j_{k-1}\}} x_j \geq 1.$$

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

**Étape finale** Le processus se termine quand l'impossibilité de l'opération pivot de la méthode duale du simplexe apparaît, indiquant que la région courante ne contient aucun point entier et que l'ensemble des solutions efficaces est complètement déterminé.

## 2.7 Méthode de M. Abbas & D. Chaabane [2]

Cette méthode dénommée "SEEVD", est une forme modifiée de la méthode de Gupta & Malhotra [18] (première procédure) où le test d'arrêt a été modifié afin de produire toutes les solutions efficaces du problème de programmation linéaire multiojectif en nombres entiers.

La technique étant présentée en détail dans l'algorithme suivant :

### Algorithme

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

**Étape 1.** Résoudre le problème  $(P_1)$ . Soit  $x^1$  la solution optimale et  $(z_1^1, z_1^2, \dots, z_1^r)$  le vecteur critère correspondant.

**Cas 1.** Si  $\Omega_1 = \emptyset$ , alors la solution optimale est unique. Enregistrer le premier  $r$ -uplet non dominé  $(z_1^1, z_1^2, \dots, z_1^r)$  dans  $SND(P)$  et aller à l'étape 2.

**Cas 2.** Si  $\Omega_1 \neq \emptyset$ , alors la solution optimale peut ne pas être unique. Pour chaque  $j \in \Omega_1$  calculer  $\theta$ .

(a) Si pour tout  $j \in \Omega_1$ , on a  $\theta < 1$ , alors il n'y a pas de solutions alternatives à  $x^1$  le long des arêtes  $E^j$ ,  $j \in \Omega_1$ .  $SND(P) = \{(z_1^1, z_1^2, \dots, z_1^r)\}$  et aller à l'étape 2.

(b) Sinon, tant qu'il existe au moins un  $j \in \Omega_1$  tel que  $\theta \geq 1$  faire :

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

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

**Étape 2.**  $k = 1$ ;

**2.1** Construire l'ensemble  $\Psi_k$ .

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

**Cas 2.** Sinon, mettre  $\psi = \Psi_k$  et aller à (a).

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

- Si  $\theta < 1$ , il n'y a aucune solution entière sur l'arête  $E^{j_k}$ , faire  $\psi = \psi \setminus \{j_k\}$ . Si  $\psi = \emptyset$ , choisir un  $j_k \in \Psi_k$  et aller à l'étape (2.2), sinon aller à (a).
- Sinon, déterminer les solutions entières sur  $E^{j_k}$ , évaluer en chacune d'elles les  $r$  critères et mettre à jour l'ensemble  $SND(P)$ . Aller à l'étape (2.2).

**2.2** Utiliser la coupe  $\sum_{j \in N_k \setminus \{j_k\}} x_j \geq 1$  pour réduire le domaine de recherche et par application des méthodes duale du simplexe et les coupes de Gomory si nécessaire, on obtient  $x^{k+1}$  comme étant la solution optimale du problème augmenté. Mettre à jour  $SND(P)$ ,  $k = k + 1$  et aller à l'étape (2.1).

**Étape 3.** La procédure prend fin quand l'opération pivot est impossible, le problème est devenu irréalisable dans la nouvelle région tronquée et la liste finale  $SND(P)$  représente l'ensemble de toutes les solutions non dominées.

## 2.8 Une méthode de détermination de toutes les solutions alternatives

Soit le programme linéaire en nombres entiers suivant :

$$(P) \begin{cases} Max & z = cx \\ & x \in D \end{cases}$$

dont

$$(R) \begin{cases} Max & z = cx \\ & x \in S \end{cases}$$

le problème relaxé associé.

Où :

- $D = \{x \in \mathbb{Z}^n | Ax = b, x \geq 0\}$ .
- $S = \{x \in \mathbb{R}^n | Ax = b, x \geq 0\}$ .
- $A \in \mathbb{Z}^{m \times n}$ ,  $b \in \mathbb{Z}^m$  et  $c \in \mathbb{Z}^{1 \times n}$ .

avec  $S$  est supposé borné et non vide.

La résolution de  $(P)$  nécessite une première phase qui consiste en la résolution du programme  $(R)$ . Si la solution optimale de  $(R)$  réalise  $(P)$ , alors elle est optimale pour  $(P)$ . Sinon une deuxième phase est obligatoire pour trouver une solution optimale entière en appliquant une ou plusieurs coupes de Gomory [15].

Soit  $x^*$  la solution optimale de  $(P)$  obtenue sur  $S_g$ , région tronquée de  $S$  par applica-

tion des coupes de Gomory éventuelles :

$$S_g = \{x \in \mathbb{R}^{n_g} \mid A_g x = b_g; A_g \in \mathbb{R}^{m_g \times n_g}, b_g \in \mathbb{R}^{m_g}, x \geq 0\}$$

Notons par  $B_g$  l'ensemble des indices des variables de base de  $x^*$  et par  $N_g$  celui des indices des variables hors base.

Cette section fournit une troisième phase qui doit être annexée aux deux premières, elle est applicable en cas d'existence de solutions alternatives à  $x^*$ .

Après résolution de  $(P)$ , nous définissons les notations suivantes :

- $L = \left\{ x \in S_g \mid cx \geq cx^* \text{ et } \sum_{j \in N_g} x_j \geq 1 \right\}$ . Cet ensemble désigne la région tronquée de  $S_g$  par adjonction de la contrainte  $cx \geq cx^*$  et la coupe  $\sum_{j \in N_g} x_j \geq 1$ , et éventuellement, par des coupes de Gomory successives.
- Pour  $k \geq 1$ ;
  - $L_k = \left\{ x \in L_{k-1} \mid \sum_{j \in N_{k-1} \setminus \{j_{k-1}\}} x_j \geq 1 \right\}$  avec  $L_0 = L$ , c'est la région courante tronquée de  $L_{k-1}$  par application de la coupe  $\sum_{j \in N_{k-1} \setminus \{j_{k-1}\}} x_j \geq 1$  et par des coupes successives de Gomory éventuelles.
  - $x^k$  est la  $k^{\text{ème}}$  solution optimale de  $(P)$  obtenue sur  $L_k$ .
  - $B_k$  est l'ensemble des indices des variables de base de  $x^k$  et  $N_k$  étant l'ensemble des indices des variables hors base associé.

### Algorithme

Soit  $Alt(P)$  l'ensemble de toutes les solutions optimales du problème  $(P)$ .

$Alt(P) = \emptyset$  au départ.

**Étape 0.** Résoudre  $(P)$ , soit  $x^*$  la solution optimale obtenue sur  $S_g$ ,  $Alt(P) = \{x^*\}$  et construire l'ensemble  $\Omega_g = \{j \in N_g \mid \widehat{c}_j = 0\}$ .

**Cas 1.** Si  $\Omega_g = \emptyset$  alors,  $x^*$  est unique, terminer.

**Cas 2.** Sinon,  $x^*$  ne peut pas être unique. Dans ce cas, réduire le domaine

de recherche par ajout de la contrainte  $cx \geq cx^*$  et la coupe  $\sum_{j \in N_g} x_j \geq 1$  éliminant le point  $x^*$ . Utiliser la méthode duale du simplexe et des coupes de Gomory si nécessaire, pour obtenir une solution optimale entière de  $(P)$ , soit  $x^0$ , sur  $L$  et mettre à jour  $Alt(P)$ .

**Étape  $k$ ,  $k \geq 1$ .** Choisir un indice  $j_{k-1} \in N_{k-1}$ , explorer l'arête correspondante pour d'éventuelles solutions entières et mettre à jour l'ensemble  $Alt(P)$ .

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

$$\sum_{j \in N_{k-1} \setminus \{j_{k-1}\}} x_j \geq 1 \quad (2.3)$$

et chercher de nouveau une solution optimale entière de  $(P)$ , soit  $x^k$ , dans la région tronquée  $L_k$ .

**Étape finale** l'algorithme s'arrête lorsque le problème devient irréalisable. Ainsi, toutes les solutions optimales du problème  $(P)$  sont déterminées.

**Théorème 2.4.** *L'algorithme décrit ci-dessus est fini et donne toutes les solutions alternatives à  $x^*$ .*

*Démonstration.* A chaque étape  $k$  de l'algorithme, au moins la solution  $x^k$  est supprimée, éventuellement toutes les solutions se trouvant sur l'arête incidente à  $x^k$ . Le domaine  $L$  étant borné contenant donc, un nombre fini de solutions, alors l'algorithme est fini. De plus, l'algorithme consiste en un balayage systématique de  $L$ , à l'aide de la coupe (2.3) et éventuellement, celles de Gomory, aucune solution ne sera ratée.  $\square$

## 2.9 Conclusion

Ce chapitre avait pour objectif de réaliser une étude détaillée de quelques méthodes d'optimisation multiobjectif en nombres entiers. Le suivant chapitre contient une généralisation au cas fractionnaire de la méthode proposée par M. Abbas & D. Chaabane [2] décrite dans ce chapitre avec quelques modifications majeures.

## Une méthode d'optimisation multiobjectif fractionnaire discrète

### 3.1 Introduction

Le terme *programmation fractionnaire* est utilisé pour désigner un type de problèmes d'optimisation où la fonction objectif est un quotient  $f(x)/g(x)$ , sous réserve de certaines contraintes. La programmation fractionnaire linéaire 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. Ces problèmes ont suscité beaucoup d'intérêt, car ils ont une multitude d'applications dans plusieurs domaines tels que : la planification de la production, la gestion de bilans de banques, les ressources en eau, les opérations militaires, la planification hospitalière, etc.

Notons que la programmation fractionnaire linéaire peut être aussi étendue aux problèmes de la programmation fractionnaire linéaire multiobjectif. Une recherche bibliographique nous a permis de constater qu'il y a peu de travaux traitant ce problème en variables continues et moins encore pour le cas où les variables sont entières. Nous citons pour ce dernier problème, la méthode décrite dans [27] par M. Abbas & M. Moulai, la méthode de M. Saad Omar and B. Hughes John [30] pour un problème biobjectif et la méthode de R. Gupta & R. Malhotra [19]. Cette dernière méthode généralise celle décrite dans [18] (première procédure). Nous avons là aussi, noté une défaillance dans le critère d'arrêt de la méthode. En effet, la méthode peut s'arrêter

sans avoir générer toutes les solutions efficaces du problème.

Le but de ce chapitre est de présenter une généralisation au cas fractionnaire de l'algorithme de résolution d'un programme linéaire multiobjectif en nombres entiers (MOILP) proposé par M. Abbas & D. Chaabane [2]. Avec quelques modifications de cette dernière et en utilisant des techniques de programmation fractionnaire linéaire en nombres entiers, nous avons pu élaboré une procédure pour la détermination de toutes les solutions efficaces d'un problème de programmation fractionnaire linéaire multiobjectif en nombres entiers.

## 3.2 Programmation fractionnaire linéaire mono objectif

### 3.2.1 Formulation mathématique

Le problème de programmation fractionnaire linéaire mono objectif a la forma suivante :

$$(P) \begin{cases} \text{Max} & z = \frac{cx + \alpha}{dx + \beta} \\ x \in S \end{cases}$$

Où :

- $S = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$ , avec  $\alpha$  et  $\beta$  sont des réels,  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ ,  $c$  et  $d$  des vecteurs de  $\mathbb{R}^{1 \times n}$ .
- $S$  borné et non vide.
- $dx + \beta \neq 0, \forall x \in S$ .

Si  $x \in \mathbb{Z}^n$ , on parle dans ce cas de problème de programmation fractionnaire linéaire en nombre entiers.

Selon Steuer [34], les programmes fractionnaires linéaires présentent l'intérêt particulier d'avoir des courbes de niveaux linéaires de leurs fonctions objectif.

Pour illustrer cette propriété, considérons une  $\bar{z}$ -courbe niveau quelconque de la fonction objectif :

$$\bar{z} = \frac{cx + \alpha}{dx + \beta}$$

Après simplification, nous obtenons :  $cx + \alpha = \bar{z}(dx + \beta)$ .

Ce qui donne :  $(c - \bar{z}d)x = \bar{z}\beta - \alpha$ , qui est une expression linéaire de la  $\bar{z}$ -courbe niveau de la fonction objectif.

Puisque  $\bar{z}$  est quelconque, on constate que chaque courbe niveau du critère fractionnaire linéaire est linéaire sur  $S$  pourvu que le dénominateur ne soit pas nul sur  $S$ . Donc, si un programme fractionnaire linéaire mono objectif possède une solution optimale, au moins un point extrême de  $S$  est optimal.

Malgré la linéarité de la courbe niveau de la fonction objectif, les courbes niveaux ne sont pas parallèles (lorsque  $c \neq 0$ ,  $d \neq 0$  et  $c \neq \omega d$  pour tout  $\omega \in \mathbb{R}$ ) comme ils le sont en programmation linéaire.

On appelle *ensemble rotation*, l'ensemble de tous les points d'intersection entre la 0-courbe niveau du numérateur et la 0-courbe niveau du dénominateur. Dans  $\mathbb{R}^2$ , l'ensemble rotation est appelé *point de rotation* et est appelé *axe de rotation* dans  $\mathbb{R}^3$ . Les éléments de cet ensemble sont déterminés par la résolution du système :

$$\begin{cases} cx = -\alpha \\ dx = -\beta \end{cases}$$

**Exemple illustratif [34]**

On Considère le programme fractionnaire linéaire suivant :

$$\begin{cases} \text{Max } z = \frac{x_1 + x_2 - 1}{5x_1 + x_2 - 1} \\ 3x_1 + 2x_2 \geq 6 \\ x_1 \leq 3 \\ x_2 \leq 3 \\ x_1, x_2 \geq 0 \end{cases}$$

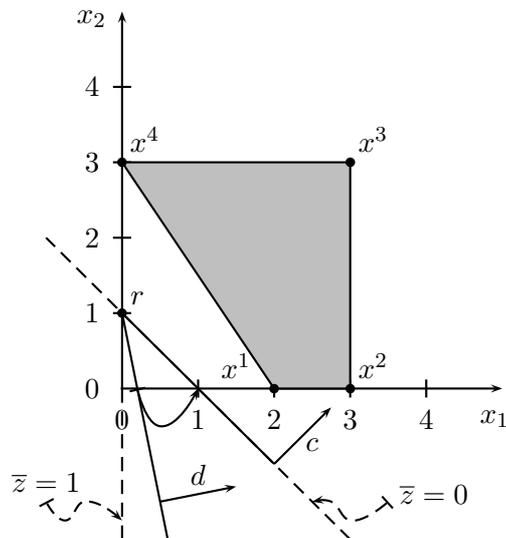


FIG. 3.1 – Graphe de l'exemple

La courbe de niveau  $\bar{z}$  est l'ensemble  $CN$  défini par :

$$CN = \{(x_1, x_2) \mid (1 - 5\bar{z})x_1 + (1 - \bar{z})x_2 = 1 - \bar{z}\}$$

. Donc pour :

- $\bar{z} = 0$ , on a  $x_1 + x_2 = 1$  : courbe de niveau 0.
- $\bar{z} = 1$ , on a  $x_1 = 0$  : courbe de niveau 1.

Les lignes discontinues représentent les courbes de niveau 0 du numérateur et du dénominateur dont l'intersection est le point rotation  $r = (0, 1)$ . La flèche circulaire représente le gradient de la fonction fractionnaire linéaire, elle indique le sens et l'angle avec lequel se déplacent les courbes de niveaux. Tandis que  $c$  et  $d$  représentent respectivement les gradient des courbes de niveau 0 du numérateur et du dénominateur. Le point extrême  $x^4 = (0, 3)$  de valeur optimale  $z^* = 1$  est l'intersection du domaine  $S$  avec la courbe de niveau 1 en faisant déplacer la courbe de niveau 0 autour du point  $r$  suivant le sens de rotation trigonométrique.

### 3.2.2 Résolution des programmes fractionnaires linéaires mono objectif

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

#### La 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 qu'en variables entières ([4], [6], [16], [17]).

#### La 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 réso-

lutions de ce type de problème fournit une solution optimale du programme fractionnaire. Autour de cette approche, plusieurs algorithmes ont été proposés, nous citons entre autres : algorithme de Isbell et Marlow [20] pour les programmes hyperboliques, algorithme de Dinkelbach [13] et algorithme de Seshan et Tikekar [32].

### La 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 [9] sont les premiers à avoir linéarisé un problème hyperbolique en un problème linéaire équivalent :

$$(P) \begin{cases} \text{Max} & \frac{cx + \alpha}{dx + \beta} \\ Ax = b \\ x \geq 0 \end{cases} \Leftrightarrow (PE) \begin{cases} \text{Max} & cy + \alpha z \\ Ay - bz \leq 0 \\ dy + \beta z = 1 \\ y \geq 0, z \geq 0 \end{cases}$$

en posant  $z = \frac{1}{dx + \beta}$  et  $y = zx$ .

**Proposition 3.1** (Charnes & Cooper [9]). *Si  $(y^*, z^*)$  est une solution optimale de (PE), alors  $z^* > 0$  et  $x^* = \frac{y^*}{z^*}$  est une solution optimale de (P).*

#### 3.2.3 Méthode de A. Cambini *et al.* [6]

On considère le programme fractionnaire linéaire continu (P) :

$$(P) \begin{cases} \text{Max} & z = \frac{cx + \alpha}{dx + \beta} \\ x \in S \end{cases}$$

Où  $S = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$  avec  $\alpha$  et  $\beta$  sont des réels,  $c$  et  $d$  sont des vecteurs de  $\mathbb{R}^{1 \times n}$ ,  $A$  est une matrice réelle de format  $m \times n$  et  $b$  un vecteur de  $\mathbb{R}^m$ .

**Définition 3.1.** On dira que  $x^*$  est une *solution optimale niveau* pour le problème  $(P)$  si et seulement si  $x^*$  est solution optimale du problème  $P(\theta)$  pour certaines valeurs de  $\theta$  :

$$(P(\theta)) \begin{cases} \text{Max} & cx + \alpha \\ & x \in S \\ & dx + \beta = \theta \end{cases}$$

L'algorithme de Cambini *et al.* [6] génère une séquence finie  $x^k$ ,  $k = \overline{1, l}$  de solutions optimales niveau dont la première est trouvée de la façon suivante :

Résoudre le programme linéaire  $(P_0) \{ \text{Min } dx + \beta \mid x \in S \}$ , soit  $x^0$  la solution optimale (car sa fonction objectif est bornée). Si  $x^0$  est unique, alors elle est une solution niveau, sinon résoudre le programme linéaire  $(P_1) \text{Max } \{ cx + \alpha \mid dx = dx^0; x \in S \}$ . Si  $(P_1)$  n'admet pas de solutions, alors la valeur de la fonction objectif est infinie; sinon une solution optimale  $x^1$  de  $(P_1)$  est aussi une solution optimale niveau.

**Théorème 3.1** ([6]). *Si  $J = \{ j \in N \mid \widehat{d}_j > 0 \} = \emptyset$  ou  $\widehat{\gamma} = \widehat{\beta}\widehat{c} - \widehat{\alpha}\widehat{d}_N$  est tel que  $\widehat{\gamma}_j \leq 0$  pour tout indice hors base  $j \in N$ , alors  $x^k$  est une solution optimale du problème  $(P)$ .*

### Algorithme

**Étape 1.** Trouver la solution optimale niveau  $x^1$ .

Si une telle solution n'existe pas, alors  $\sup \left\{ \frac{cx + \alpha}{dx + \beta} \mid x \in S \right\} = +\infty$ , terminer.

Sinon, poser  $k = 1$  et aller à l'étape 2.

**Étape 2.** Si  $J = \{ j \in N \mid \widehat{d}_j > 0 \} = \emptyset$ , terminer,  $x^k$  est une solution optimale du problème  $(P)$ . Sinon, soit  $s$  tel que  $\frac{\widehat{c}_s}{\widehat{d}_s} = \max_{j \in J} \left( \frac{\widehat{c}_j}{\widehat{d}_j} \right)$ .

Si  $\widehat{\gamma}_s > 0$ , aller à l'étape 3. Sinon, terminer,  $x^k$  est une solution optimale de  $(P)$ .

**Étape 3.** La variable hors base  $x_s$  entre dans la base au moyen d'une opération pivot, poser  $k = k + 1$  et aller à l'étape 2. Si une telle opération n'est pas possible, terminer :  $\sup \left\{ \frac{cx + \alpha}{dx + \beta} \mid x \in S \right\} = \frac{\widehat{c}_s}{\widehat{d}_s}$ .

### 3.2.4 Méthode de D. Granot *et al.* [16]

Dans cette section, un algorithme totalement en nombres entiers pour la résolution des problèmes de programmation fractionnaire linéaire discrète (*ILFP*) est présenté. Il est appliqué directement au problème (*ILFP*) dans un format initialement suggéré par Martos [23] pour le cas continu et constitue un prolongement direct de l'algorithme totalement en nombres entiers pour les problèmes de programmation linéaire discrète (*ILP*) aux problèmes (*ILFP*).

Considérons le problème de programmation fractionnaire linéaire en nombres entiers suivant (*P*) :

$$(P) \begin{cases} \text{Max} & z = \frac{cx + \alpha}{dx + \beta} \\ x \in D \end{cases}$$

Où  $D = \{x \in \mathbb{Z}^n \mid Ax = b, x \geq 0\}$ .

On suppose que toutes les données du problème sont entières et que l'on dispose initialement d'une solution réalisable pour (*P*). Ainsi, (*P*) peut être écrit d'une manière équivalente comme :

$$\begin{cases} \text{Max} & \frac{\sum_{j \in N} \widehat{c}_j x_j + \widehat{\alpha}}{\sum_{j \in N} \widehat{d}_j x_j + \widehat{\beta}} \\ x_B + \widehat{A}^N x_N = \widehat{b} \\ x_B, x_N \geq 0, \text{ entiers} \end{cases}$$

Où :  $B$  est l'ensemble des indices de base et  $N$  est l'ensemble des indices hors base. Dans cette méthode, la structure originale des contraintes est maintenue et les itérations sont réalisées dans un tableau du simplexe augmenté par trois lignes ( $m + 1$ ), ( $m + 2$ ) et ( $m + 3$ ) correspondant respectivement aux vecteurs numérateur, dénominateur et au vecteur gradient réduit :

$$\widehat{\gamma}_j = \widehat{\beta} \widehat{c}_j - \widehat{\alpha} \widehat{d}_j \quad \forall j \in N \tag{3.1}$$

A chaque itération de l'algorithme, les  $(m+2)$  premières lignes sont modifiées à travers les opérations ordinaires de pivot, tandis que la dernière ligne est modifiée selon (3.1).

Maintenant vient le test :

- Si  $\widehat{\gamma}_j \leq 0; \forall j \in N$ , alors la solution courante est une solution optimale pour le problème  $(P)$ .
- Sinon, il existe un indice  $s, s \in N$  pour lequel  $\widehat{\gamma}_s > 0$ . Soit  $\theta_r$  tel que :

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

alors la ligne  $r$  sert comme une ligne pivot pour générer une coupe de Gomory de la forme :

$$v + \sum_{j \in N} \left[ \frac{\widehat{A}_{rj}}{\widehat{A}_r^s} \right] x_j = \left[ \frac{\widehat{b}_i}{\widehat{A}_r^s} \right], \quad v \geq 0 \quad (3.2)$$

Afin de résoudre  $(P)$ , la coupe (3.2) est rajoutée au tableau du simplexe pour servir comme ligne pivot, avec la colonne  $s$  comme colonne pivot. Sachant que la valeur du pivot dans ce cas est de 1, les nouveaux coefficients obtenus après l'exécution des opérations de pivot sont tous des entiers.

### 3.3 Une méthode d'optimisation multiobjectif fractionnaire discrète

#### 3.3.1 Définitions et notations

Considérons le problème de programmation fractionnaire linéaire en nombres entiers à objectifs multiples (MOILFP) suivant :

$$(P) \begin{cases} \text{Max} & z^i = \frac{c^i x + \alpha^i}{d^i x + \beta^i} \quad i = \overline{1, r} \\ x \in D \end{cases}$$

Où :

- $r$  le nombre d'objectifs ( $r \geq 2$ ).
- $D = \{x \in \mathbb{Z}^n \mid Ax = b, x \geq 0\}$ .

Avec :  $A \in \mathbb{Z}^{m \times n}$ ,  $b \in \mathbb{Z}^m$ ,  $c^i$  et  $d^i \in \mathbb{Z}^{1 \times n}$ ,  $\alpha^i$  et  $\beta^i \in \mathbb{Z}$   $i = \overline{1, r}$ .

Nous supposons que le domaine des solutions réalisables du problème relaxé  $S = \{x \in \mathbb{R}^n | Ax = b, x \geq 0\}$  est compact et non vide et que pour tout  $i = \overline{1, r}$ , on a  $d^i x + \beta^i > 0 \forall x \in S$ .

**Remarque.** Comme pour les problèmes de programmation linéaire discrète à objectifs multiples (MOILP), le but de traiter les problèmes de programmation fractionnaire linéaire discrète à objectifs multiples (MOILFP) est de déterminer toutes les solutions qui sont efficaces au sens de la définition (cf. §1.4.2).

Nous adoptons pour la "résolution" de  $(P)$  une approche consistant à résoudre une séquence finie de problèmes de programmation fractionnaire linéaire discrète. Considérons donc, le problème  $(ILFP)$  suivant :

$$(P_1) \begin{cases} \text{Max} & z^1 = \frac{c^1 x + \alpha^1}{d^1 x + \beta^1} \\ x \in D \end{cases}$$

Dont le problème relaxé est :

$$(R) \begin{cases} \text{Max} & z^1 = \frac{c^1 x + \alpha^1}{d^1 x + \beta^1} \\ x \in S \end{cases}$$

Notons qu'au lieu de  $(P_1)$ , on peut considérer un des problèmes  $(P_i)_{i=\overline{2, r}}$  qui maximise  $z^i$  sur  $D$ .

Pour  $k \geq 1$  :

- $S_k = \{x \in \mathbb{R}^{n_k} | A_k x = b_k; A_k \in \mathbb{R}^{m_k \times n_k}; b_k \in \mathbb{R}^{m_k}\}$  avec  $S_0 = S$ , obtenu à partir de  $S_{k-1}$  en appliquant la coupe  $\sum_{j \in N_{k-1} \setminus j_{k-1}} x_j \geq 1$ ,  $j_{k-1} \in N_{k-1}$  et éventuellement des coupes successives de Gomory.
- $x^k$  est la  $k^{\text{ème}}$  solution optimale de  $(P_1)$  obtenue sur  $S_k$ .
- $B_k$  est l'ensemble des indices des variables de base de  $x^k$  et  $N_k$  est celui des indices des variables hors base correspondant.
- $\hat{A}_k = (A_k^{B_k})^{-1} A_k = (\hat{a}_{lig(i)k}^j)_{\substack{i=\overline{1, m_k} \\ j=\overline{1, n_k}}}$ , où "lig" est une application linéaire définie par :

$lig : B_k \longrightarrow \{1, \dots, m_k\}$   
 $i \longrightarrow lig(i) = \text{position-ligne de la variable de base } x_i^k \text{ dans le}$   
 tableau du simplexe à l'étape  $k$ .

- $\widehat{c}_k^i = c_k^i - \pi_k^{1i} A_k$  et  $\widehat{d}_k^i = d_k^i - \pi_k^{2i} A_k$ .  
 Avec :  $\pi_k^{1i} = (c_k^i)_{B_k} (A_k^{B_k})^{-1}$  et  $\pi_k^{2i} = (d_k^i)_{B_k} (A_k^{B_k})^{-1}$
- $(\widehat{\gamma}_k^i)_j$  est la  $j^{\text{ème}}$  composante du vecteur gradient réduit  $\widehat{\gamma}_k^i$  à l'étape  $k$  pour tout  $i = \overline{1, r}$  définit par :

$$\widehat{\gamma}_k^i = \widehat{\beta}_k^i \widehat{c}_k^i - \widehat{\alpha}_k^i \widehat{d}_k^i$$

avec  $\widehat{\alpha}_k^i = c^i x^k + \alpha^i$  et  $\widehat{\beta}_k^i = d^i x^k + \beta^i$ .

- On définit l'ensemble  $T_k$  par :

$$T_k = \{j \in N_k \mid (\widehat{\gamma}_k^1)_j < 0 \text{ et } (\widehat{\gamma}_k^i)_j \leq 0 \text{ pour tout } i = \overline{2, r}\}$$

- On définit l'ensemble  $\Omega$  par :

$$\Omega = \{j \in N_1 \mid (\widehat{\gamma}_1^1)_j = 0\}$$

- Comme au chapitre 2, une arête incidente à une solution  $x^k$  est définit par l'ensemble :

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

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

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

On note par  $ESE(P)$  l'ensemble des solutions efficaces du problème  $(P)$ .

### 3.3.2 Résultats théoriques

En  $x^k$  solution optimale entière de  $(P_1)$  sur  $S_k$ , soit :

$$T_k = \{j \in N_k \mid (\widehat{\gamma}_k^1)_j < 0 \text{ et } (\widehat{\gamma}_k^i)_j \leq 0 \text{ pour tout } i = \overline{2, r}\}$$

**Proposition 3.2.** *Soit  $j_k \in T_k$ ,  $E^{j_k}$  étant l'arête incidente à  $x^k$  alors, tous les points entiers se trouvant sur l'arête  $E^{j_k}$  ne sont pas efficaces.*

*Démonstration.* Soit  $x \in S_k$ ,  $x$  entier et  $x \neq x^k$ , alors :

$$z^i(x) = \frac{\sum_{j \in N_k} (\widehat{c}_k^i)_j x_j + \widehat{\alpha}_k^i}{\sum_{j \in N_k} (\widehat{d}_k^i)_j x_j + \widehat{\beta}_k^i} \quad \forall i = \overline{1, r}$$

En particulier, si  $x \in E^{j_k}$  alors, on aura :

$$z^i(x) = \frac{\widehat{\alpha}_k^i + (\widehat{c}_k^i)_{j_k} \theta_{j_k}}{\widehat{\beta}_k^i + (\widehat{d}_k^i)_{j_k} \theta_{j_k}} \quad \forall i = \overline{1, r}$$

En  $x^k$ , on a :

$$z^i(x^k) = \frac{\widehat{\alpha}_k^i}{\widehat{\beta}_k^i} \quad \forall i = \overline{1, r}$$

Calculons maintenant la différence  $z^i(x) - z^i(x^k)$  :

$$z^i(x) - z^i(x^k) = \frac{\widehat{\alpha}_k^i + (\widehat{c}_k^i)_{j_k} \theta_{j_k}}{\widehat{\beta}_k^i + (\widehat{d}_k^i)_{j_k} \theta_{j_k}} - \frac{\widehat{\alpha}_k^i}{\widehat{\beta}_k^i} \quad \forall i = \overline{1, r}$$

On obtient après simplification :

$$z^i(x) - z^i(x^k) = \frac{\theta_{j_k} (\widehat{\gamma}_k^i)_{j_k}}{\widehat{\beta}_k^i (\widehat{\beta}_k^i + (\widehat{d}_k^i)_{j_k} \theta_{j_k})} \quad \forall i = \overline{1, r} \quad (3.3)$$

Pour le critère  $z^1$ , on a  $(\widehat{\gamma}_k^1)_{j_k} < 0$  d'après la définition de  $T_k$ .

Par conséquent :

$$z^1(x) - z^1(x^k) = \frac{\theta_{j_k} (\widehat{\gamma}_k^1)_{j_k}}{\widehat{\beta}_k^1 (\widehat{\beta}_k^1 + (\widehat{d}_k^1)_{j_k} \theta_{j_k})} < 0 \quad (3.4)$$

De plus, pour tout  $i = \overline{2, r}$ , on a  $(\widehat{\gamma}_k^i)_{j_k} \leq 0$ , toujours d'après la définition de  $T_k$ .

Donc :

$$z^i(x) - z^i(x^k) = \frac{\theta_{j_k}(\widehat{\gamma}_k^i)_{j_k}}{\widehat{\beta}_k^i(\widehat{\beta}_k^i + (\widehat{d}_k^i)_{j_k} \theta_{j_k})} \leq 0 \quad \forall i = \overline{2, r} \quad (3.5)$$

D'après (3.4) et (3.5), on en conclut que  $z^i(x^k)$  domine  $z^i(x)$ . Ainsi, toutes les solutions se trouvant sur  $E^{j_k}$  ne sont pas efficaces.  $\square$

**Théorème 3.2** ([1]). *Toutes les solutions réalisables du problème  $(P_1)$  le long de l'arête  $E^{j_k}$ ,  $j_k \in T_k$  (ou  $j_k \in N_k$ ) de la région  $S$  (ou la région tronquée  $S_k$ ) déduites à partir de  $x^k$ , existent dans le demi espace ouvert :*

$$\sum_{j \in N_k \setminus \{j_k\}} x_j < 1$$

*Démonstration.* Puisque  $x^k$  est une solution réalisable de  $(P_1)$  alors,  $A(x_j^k)_{j=\overline{1, n}} = b$ .

À l'étape  $k$ , on a :

$$A_k(x_j^k)_{j=\overline{1, n_k}} = \sum_{i \in B_k} A_k^i x_i^k = b_k$$

Pour un certain  $j_k \in T_k$  :

$$\sum_{i \in B_k} A_k^i x_i^k + \theta_{j_k} A_k^{j_k} - \theta_{j_k} A_k^{j_k} = b_k \quad (3.6)$$

Comme :

$$A_k^{j_k} = \sum_{i \in B_k} A_k^i \widehat{a}_{lig(i)k}^{j_k} \quad (3.7)$$

Alors :

$$(3.6) \Leftrightarrow \sum_{i \in B_k} A_k^i \left( x_i^k - \theta_{j_k} \widehat{a}_{lig(i)k}^{j_k} \right) + \theta_{j_k} A_k^{j_k} = b_k \quad (3.8)$$

Donc pour  $0 \leq \theta_{j_k} \leq \theta$  avec  $\theta = \min_{i \in B_k} \left\{ \frac{x_i^k}{\widehat{a}_{lig(i)k}^{j_k}} \mid \widehat{a}_{lig(i)k}^{j_k} > 0 \right\}$ ,  $\widehat{x}$  est définie comme :

$$\widehat{x} = \begin{cases} \widehat{x}_i = x_i^k - \theta_{j_k} \widehat{a}_{lig(i)k}^{j_k} & i \in B_k \\ \widehat{x}_{j_k} = \theta_{j_k} \\ \widehat{x}_l = 0 & \forall l \in N_k \setminus j_k \end{cases}$$

qui est une nouvelle solution réalisable du problème  $(P_1)$  à condition que  $\theta_{j_k}$  soit entier

positif et  $\theta_{j_k} \widehat{a}_{lig(i)k}^{j_k}$  sont entiers pour tout  $i \in B_k$ .

Et en utilisant la démonstration de la proposition (3.2),  $\widehat{x}$  est une solution réalisable entière du problème  $(P_1)$  située sur l'arête  $E^{j_k}$  de la région  $S$  (ou la région tronquée  $S_k$ ) déduite à partir de  $x^k$  et qui n'est pas efficace. Comme  $\widehat{x}_j = 0$  pour tout  $j \in N_k \setminus \{j_k\}$ , on aura :

$$\sum_{j \in N_k \setminus \{j_k\}} \widehat{x}_j < 1$$

Donc, le point  $\widehat{x}$  appartient au demi espace ouvert :

$$\sum_{j \in N_k \setminus \{j_k\}} x_j < 1$$

□

**Théorème 3.3** ([1]). *Une solution réalisable du problème  $(P_1)$  ne se trouvant pas sur l'arête  $E^{j_k}$ ,  $j_k \in T_k$  (ou  $j_k \in N_k$ ) de la région  $S$  (ou la région la région tronquée  $S_k$ ) à travers un point réalisable  $x^k$  appartient au demi espace fermé :*

$$\sum_{j \in N_k \setminus \{j_k\}} x_j \geq 1$$

*Démonstration.* Soit  $\widehat{x}$  une solution réalisable du problème  $(P_1)$  qui n'est pas sur l'arête  $E^{j_k}$  mais qui vérifie  $\sum_{j \in N_k \setminus \{j_k\}} \widehat{x}_j < 1$ . Alors  $\widehat{x}_j = 0 \forall j \in N_k \setminus \{j_k\}$ . Pour  $j = j_k$  on a : soit  $\widehat{x}_{j_k} = 0$ , soit  $\widehat{x}_{j_k} > 0$ .

**Cas 1.** Si  $\widehat{x}_{j_k} = 0$ , alors  $\widehat{x}_{j_k} = 0$  est une variable hors base ce qui fait que l'ensemble des indices des variables hors base correspondant à  $\widehat{x}$  est  $N_k$ . Ceci implique que l'ensemble des indices des variables de base est  $B_k$ . Donc  $x^k = \widehat{x}$ , qui est une contradiction avec le fait que  $\widehat{x}$  n'est pas sur l'arête  $E^{j_k}$ .

**Cas 2.** Si  $\widehat{x}_{j_k} > 0$  alors, on a deux cas :

-  $\widehat{x}_{j_k}$  est tel que :

$$\widehat{x}_{j_k} > \min_{i \in B_k} \left\{ \frac{x_i^k}{\widehat{a}_{lig(i)k}^{j_k}} \mid \widehat{a}_{lig(s)k}^{j_k} > 0 \right\} = \frac{x_s^k}{\widehat{a}_{lig(s)k}^{j_k}}$$

alors la composante  $\hat{x}_s = x_s^k - \hat{x}_{j_k} \hat{a}_{lig(s)k}^s$  implique que  $\hat{x}$  n'est pas réalisable, ce qui est faux.

-  $\hat{x}_{j_k}$  est un entier positif tel que :

$$\hat{x}_{j_k} \leq \min_{i \in B_k} \left\{ \frac{x_i^k}{\hat{a}_{lig(i)k}^j} \mid \hat{a}_{lig(s)k}^{j_k} > 0 \right\} = \frac{x_s^k}{\hat{a}_{lig(s)k}^{j_k}}$$

ce qui implique que  $\hat{x}$  est sur l'arête  $E^{j_k}$ . D'où une contradiction avec l'hypothèse et par conséquent  $\hat{x}_j > 0$  pour au moins un  $j \in N_k \setminus \{j_k\}$ .

Ainsi, dans les deux cas, on a prouvé que pour au moins un indice  $j \in N_k \setminus \{j_k\}$ ,  $\hat{x}_j > 0$  et par suite  $\hat{x}_j \geq 1$  impliquant que  $\hat{x}$  appartient bien au demi espace fermé :

$$\sum_{j \in N_k \setminus \{j_k\}} x_j \geq 1$$

□

### 3.3.3 Développement de la méthode

Afin de caractériser l'ensemble de toutes les solutions efficaces du problème  $(P)$ , l'algorithme proposé consiste à réaliser les étapes suivantes :

**Étape 1.(initialisation)**  $ESE(P) = \emptyset$ ,  $k = 1$ .

Résoudre par la méthode de Granot *et al.* [16] étudiée précédemment le problème  $(P_1)$ . Construire l'ensemble  $\Omega$ .

**Cas 1.** Si  $\Omega = \emptyset$ , alors la solution optimale de  $(P_1)$ , soit  $x^1$  est unique et elle est efficace. Mettre à jour l'ensemble  $ESE(P)$  et aller à l'étape 2.

**Cas 2.** Sinon, la solution optimale de  $(P_1)$  peut ne pas être unique. Déterminer dans ce cas toutes les solutions qui lui sont alternatives (en cas d'existence), éliminer celles qui ne sont pas efficaces et mettre à jour l'ensemble  $ESE(P)$ . Appliquer la coupe suivante pour éliminer toutes les solutions alternatives

(en cas d'existence) trouvées à cette étape :

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

et utiliser la méthode duale du simplexe et des coupes de Gomory si nécessaire pour obtenir une solution optimale entière du problème augmenté, soit  $x^1$ . Mettre à jour  $ESE(P)$ .

**Étape  $k$  ;  $k \geq 2$ .** Construire l'ensemble  $T_{k-1}$ .

**Cas 1.** Si  $T_{k-1} = \emptyset$ , alors choisir arbitrairement un indice  $j_{k-1}$  dans  $N_{k-1}$ , explorer l'arête  $E^{j_{k-1}}$  pour d'éventuelles solutions entières et mettre à jour l'ensemble  $ESE(P)$ .

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

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

et par application de l'algorithme dual du simplexe et des coupes successives de Gomory éventuelles, on obtient une solution entière  $x^k$  dans la région tronquée  $S_k$ . Actualiser  $ESE(P)$ .

**Cas 2.** Si  $T_{k-1} \neq \emptyset$ , alors choisir un indice quelconque  $j_{k-1}$  dans  $T_{k-1}$  et procéder à la troncature de l'arête  $E^{j_{k-1}}$ . La méthode duale du simplexe et des coupes de Gomory successives permettent de trouver une solution optimale entière du problème  $(P_1)$ , soit  $x^k$  sur  $S_k$ . Mettre à jour l'ensemble  $ESE(P)$ .

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

**Théorème 3.4.** Appelons  $x^1$  solution optimale de  $(P_1)$  et soit  $\Omega = \{j \in N_1 | \widehat{\gamma}_{1j}^1 = 0\}$ .

Soit  $x \in S_1$ ,  $x$  entier. On a alors :

$$z^1(x) < z^1(x^1) \text{ si et seulement si } \sum_{j \in N_1 \setminus \Omega} x_j \geq 1$$

*Démonstration.* Soit  $x \in S_1$ ,  $x$  entier.

$$\begin{aligned} z^1(x) - z^1(x^1) &= \frac{\sum_{j \in N_1} (\widehat{c}_1^1)_j x_j + \widehat{\alpha}_1^1}{\sum_{j \in N_1} (\widehat{d}_1^1)_j x_j + \widehat{\beta}_1^1} - \frac{\widehat{\alpha}_1^1}{\widehat{\beta}_1^1} \\ z^1(x) - z^1(x^1) &= \frac{\sum_{j \in N_1} (\widehat{\gamma}_1^1)_j x_j}{\widehat{\beta}_1^1 (\sum_{j \in N_1} (\widehat{d}_1^1)_j x_j + \widehat{\beta}_1^1)} \end{aligned}$$

$$\begin{aligned} z^1(x) - z^1(x^1) < 0 &\Leftrightarrow \sum_{j \in N_1} (\widehat{\gamma}_1^1)_j x_j < 0 \\ &\Leftrightarrow \sum_{j \in N_1 \setminus \Omega} (\widehat{\gamma}_1^1)_j x_j + \sum_{j \in \Omega} (\widehat{\gamma}_1^1)_j x_j < 0 \\ &\Leftrightarrow \sum_{j \in N_1 \setminus \Omega} (\widehat{\gamma}_1^1)_j x_j < 0 \text{ car } \sum_{j \in \Omega} (\widehat{\gamma}_1^1)_j x_j = 0 \\ &\Leftrightarrow \exists j \in N_1 \setminus \Omega \text{ tel que } x_j > 0 \text{ car } (\widehat{\gamma}_1^1)_j < 0 \forall j \in N_1 \setminus \Omega \\ &\Leftrightarrow \exists j \in N_1 \setminus \Omega \text{ tel que } x_j \geq 1 \text{ car } x \text{ entier} \\ &\Leftrightarrow \sum_{j \in N_1 \setminus \Omega} x_j \geq 1 \end{aligned}$$

□

**Théorème 3.5.** *L'algorithme décrit ci-dessus converge en un nombre fini d'étapes et donne l'ensemble de toutes les solutions efficaces.*

*Démonstration.* L'ensemble  $D$  des solutions réalisables de MOILFP étant fermé et borné, il contient un nombre fini de solutions entières. Initialement, seul le domaine contenant les solutions alternatives à  $x^1$  solution optimale de  $(P_1)$  est supprimé d'après le théorème (3.4). A chaque étape  $k$  de la procédure,  $k \geq 2$  au moins la solution optimale entière  $x^k$  générée est éliminée, éventuellement toutes les solutions entières se trouvant sur une arête  $E^{j_k}$  incidente à  $x^k$  d'après le théorème (3.2). Ces dernières ne sont pas efficaces si  $j_k \in T_k$  d'après la proposition (3.2) ou sont testées par rapport aux solutions potentiellement efficaces déjà trouvées si  $T_k = \emptyset$ . De plus, aucune solution

réalisable du problème initial  $(P)$  n'est ratée en rajoutant la coupe

$$\sum_{j \in N_k \setminus \{j_k\}} x_j \geq 1$$

d'après le théorème (3.3). □

**Remarque.** La méthode proposée permet de déterminer l'ensemble de toutes les solutions efficaces du problème MOILFP en évitant de parcourir tous les points réalisables du problème grâce à la proposition (3.2). Notons que cette méthode peut être adaptée au problème de la programmation linéaire multiobjectif en nombres entiers. En effet, ce dernier est un cas particulier de celui de la programmation fractionnaire linéaire discrète à objectifs multiples en fixant  $d^i$  à  $0_{\mathbb{R}^n} \forall i = \overline{1, r}$ . Donc, une nouvelle méthode peut joindre celles décrites dans ([2], [1], [35], [18], [21]) pour la détermination de toutes les solutions efficaces du problème MOILP.

### 3.3.4 Exemple illustratif

Pour illustrer le fonctionnement de notre méthode, nous considérons le problème de programmation fractionnaire linéaire en nombres entiers à trois objectifs suivant :

$$(P) \left\{ \begin{array}{l} \text{Max} \quad z^1 = \frac{6x_2 - x_3 - 6}{6x_1 + 7x_2 + 7x_3 + 10} \\ \text{Max} \quad z^2 = \frac{6x_1 + 5x_2 + 8x_3 + 10}{6x_1 + 7x_2 + 8x_3 + 5} \\ \text{Max} \quad z^3 = \frac{6x_1 - 7x_2 - 8x_3 - 6}{2x_1 + 2x_2 + 5x_3 + 8} \\ 23x_1 + 12x_2 + 17x_3 \leq 52 \\ 28x_1 + 10x_2 + 24x_3 \leq 62 \\ 23x_1 + 16x_2 + 24x_3 \leq 63 \\ x_1, x_2, x_3 \in \mathbb{Z}_+ \end{array} \right.$$

$k=1$

Résoudre par la méthode de Granot *et al.* [16], le problème :

$$(P_1) \begin{cases} \text{Max} & z^1 = \frac{6x_2 - x_3 - 6}{6x_1 + 7x_2 + 7x_3 + 10} \\ & 23x_1 + 12x_2 + 17x_3 \leq 52 \\ & 28x_1 + 10x_2 + 24x_3 \leq 62 \\ & 23x_1 + 16x_2 + 24x_3 \leq 63 \\ & x_1, x_2, x_3 \in \mathbb{Z}_+ \end{cases}$$

Les résultats obtenus sont donnés par le tableau ci-après :

<i>base</i>	$x_1$	$x_7$	$x_3$	$b$
$x_4$	11	-12	5	16
$x_5$	18	-10	14	32
$x_6$	7	-16	8	15
$x_2$	1	1	1	3
$-c^1$	-6	-6	-7	-12
$-d^1$	-1	-7	0	-31
$\gamma^1$	-174	-102	-217	
$-c^2$	1	-5	3	-25
$-d^2$	-1	-7	1	-26
$\gamma^2$	51	45	53	
$-c^3$	13	7	-1	27
$-d^3$	0	-2	3	-14
$\gamma^3$	182	44	67	

TAB. 3.1 – Tableau 1

La solution optimale de  $(P_1)$  est  $x^1 = (0, 3, 0)$ .  $\Omega = \emptyset$ , donc  $x^1$  est unique.

$$z(x^1) = \left( \frac{12}{31}, \frac{25}{26}, \frac{-27}{14} \right).$$

$$ESE(P) = \{(0, 3, 0)\}. N_1 = \{1, 7, 3\}.$$

$k=2$

$T_1 = \emptyset$ , donc choisir n'importe quel indice  $j_1 \in N_1$  et explorer l'arête correspondante. Prenons  $j_1 = 1$ , la seule solution entière se trouvant sur  $E^{j_1}$  est le point  $y^1 = (1, 2, 0)$  avec  $z(y^1) = \left(\frac{1}{5}, \frac{26}{25}, -1\right)$ . Mettre à jour  $ESE(P)$ ,  $ESE(P) = \{(0, 3, 0), (1, 2, 0)\}$ . Tronquer l'arête  $E^{j_1}$  par ajout de la coupe :  $x_3 + x_7 \geq 1$ , i.e.  $-x_3 - x_7 + x_8 = -1$ . Les coefficients de cette dernière équation sont insérés dans une nouvelle ligne du tableau (Tab. 3.1) et par application de la méthode duale du simplexe, on obtient :

<i>base</i>	$x_1$	$x_3$	$x_8$	$b$
$x_4$	11	17	-12	28
$x_5$	18	24	-10	42
$x_6$	7	24	-16	31
$x_2$	1	0	1	2
$x_7$	0	1	-1	1
$-c^1$	-6	-1	-6	-6
$-d^1$	-1	7	-7	-24
$\gamma^1$	-138	-66	-102	
$-c^2$	1	8	-5	-20
$-d^2$	-1	8	-7	-19
$\gamma^2$	39	-8	45	
$-c^3$	13	-8	7	20
$-d^3$	0	5	-2	-12
$\gamma^3$	156	4	44	

TAB. 3.2 – Tableau 2

D'après le tableau (Tab. 3.2), la solution optimale est  $x^2 = (0, 2, 0)$  de vecteur critère  $\left(\frac{1}{4}, \frac{20}{19}, \frac{-5}{3}\right)$  qui est non dominé en tenant compte des solutions efficaces trouvées précédemment. Donc,  $ESE(P) = \{(0, 3, 0), (1, 2, 0), (0, 2, 0)\}$ .  $N_2 = \{1, 3, 8\}$ .

$k=3$

$T_2 = \emptyset$ . Prendre  $j_2 = 1$ , explorer l'arête correspondante, qui contient les solutions entières :  $y^2 = (1, 1, 0)$  et  $y^3 = (2, 0, 0)$  relatives aux vecteurs critères  $\left(0, \frac{7}{6}, \frac{-7}{12}\right)$

et  $\left(\frac{-3}{11}, \frac{22}{17}, \frac{1}{2}\right)$  respectivement.

$ESE(P) = \{(0, 3, 0), (1, 2, 0), (0, 2, 0), (1, 1, 0), (2, 0, 0)\}$ . Supprimer l'arête  $E^{j_2}$  par adjonction de la coupe  $x_3 + x_8 \geq 1$  i.e.  $-x_3 - x_8 + x_9 = -1$ . L'application de la méthode duale du simplexe nous fournit :

<i>base</i>	$x_1$	$x_9$	$x_8$	$b$
$x_4$	11	17	-29	11
$x_5$	18	24	-34	18
$x_6$	7	24	-40	7
$x_2$	1	0	1	2
$x_7$	0	1	-2	0
$x_3$	0	-1	1	1
$-c^1$	-6	-1	-5	-5
$-d^1$	-1	7	-14	-31
$\gamma^1$	-181	-66	-85	
$-c^2$	1	8	-13	-28
$-d^2$	-1	8	-15	-27
$\gamma^2$	55	-8	69	
$-c^3$	13	-8	15	28
$-d^3$	0	5	-7	-17
$\gamma^3$	221	4	59	

TAB. 3.3 – Tableau 3

La solution optimale est  $x^3 = (0, 2, 1)$ ,  $z(x^3) = \left(\frac{5}{31}, \frac{28}{27}, \frac{-28}{17}\right)$ .  $ESE(P)$  ne change pas, car  $z(x^3)$  est dominé.  $N_3 = \{1, 9, 8\}$

$k=4$

$T_3 = \emptyset$ . Sélectionner arbitrairement un indice  $j_3 \in N_3$ , soit  $j_3 = 1$ . La seule solution

se trouvant sur l'arête  $E^{j_3}$  est  $y^4 = (1, 1, 1)$ ,  $z(y^4) = (\frac{-1}{30}, \frac{29}{26}, \frac{-15}{17})$  et il est dominé.  $ESE(P)$  ne change pas. Utiliser la coupe  $x_8 + x_9 \geq 1$  pour réduire le domaine de recherche (i.e.  $-x_8 - x_9 + x_{10} = -1$ ). La méthode duale du simplexe nous fournit le tableau suivant :

<i>base</i>	$x_1$	$x_{11}$	$x_{10}$	$b$
$x_4$	11	-46	17	40
$x_5$	18	-58	24	52
$x_6$	7	-64	24	47
$x_2$	1	1	0	1
$x_7$	0	-3	1	2
$x_3$	0	2	-1	0
$x_9$	0	1	-1	0
$x_8$	0	-1	0	1
$-c^1$	-6	-4	-1	0
$-d^1$	-1	-21	7	-17
$\gamma^1$	-102	-68	-17	
$-c^2$	1	-21	8	-15
$-d^2$	-1	-23	8	-27
$\gamma^2$	27	93	-24	
$-c^3$	13	23	-8	13
$-d^3$	0	-12	5	-10
$\gamma^3$	130	74	-15	

TAB. 3.4 – Tableau 5

La solution optimale est  $x^4 = (0, 1, 0)$ ,  $z(x^4) = (0, \frac{5}{4}, \frac{-13}{10})$ .

$ESE(P) = \{(0, 3, 0), (1, 2, 0), (0, 2, 0), (1, 1, 0), (2, 0, 0), (0, 1, 0)\}$ .

$N_4 = \{1, 11, 10\}$ .

k=5

$T_4 = \{10\}$ ,  $T_4 \neq \emptyset$ .  $j_4 = 10$  et l'arête  $E^{j_4}$  est tronquée en ajoutant la coupe  $x_1 + x_{11} \geq 1$

i.e.  $-x_1 - x_{11} + x_{12} = -1$ . L'application de la méthode duale du simplexe donne comme résultat, le tableau qui suit :

<i>base</i>	$x_{10}$	$x_{12}$	$x_{11}$	$b$
$x_4$	17	11	-57	29
$x_5$	24	18	-76	34
$x_6$	24	7	-71	40
$x_2$	0	1	0	0
$x_7$	1	0	-3	2
$x_3$	-1	0	2	0
$x_9$	-1	0	1	0
$x_8$	0	0	-1	1
$x_{13}$	-1	0	1	0
$x_1$	0	-1	1	1
$-c^1$	-7	-6	8	6
$-d^1$	17	-1	30	-16
$\gamma^1$	-10	-102	-52	
$-c^2$	18	1	-32	-16
$-d^2$	13	-1	-27	-11
$\gamma^2$	-10	27	80	
$-c^3$	-14	13	16	0
$-d^3$	13	0	-20	-10
$\gamma^3$	-140	130	160	

TAB. 3.5 – Tableau 5

La solution optimale est  $x^5 = (1, 0, 0)$ ,  $z(x^5) = \left(\frac{-3}{8}, \frac{16}{11}, 0\right)$ .

$ESE(P) = \{(0, 3, 0), (1, 2, 0), (0, 2, 0), (1, 1, 0), (2, 0, 0), (0, 1, 0), (1, 0, 0)\}$ .

$N_4 = \{10, 12, 11\}$ .

$\boxed{k=6}$

$T_5 = \{10\}$ ,  $T_5 \neq \emptyset$ .  $j_5 = 10$  et l'arête  $E^{j_5}$  est tronquée en ajoutant la coupe  $x_{11} + x_{12} \geq$

1 i.e.  $-x_{12} - x_{11} + x_{14} = -1$ . L'application de la méthode duale du simplexe nous donne :

<i>base</i>	$x_{12}$	$x_{15}$	$x_1$	$b$
$x_4$	22	-17	-11	18
$x_5$	38	-24	-20	14
$x_6$	32	-24	-25	15
$x_{13}$	-3	1	3	3
$x_7$	1	-1	-1	1
$x_2$	1	0	0	0
$x_9$	-3	1	3	3
$x_8$	-1	0	1	2
$x_{11}$	-1	0	1	1
$x_{14}$	-2	0	1	0
$x_{10}$	-4	1	4	4
$x_3$	-2	1	2	2
$-c^1$	-8	1	2	8
$-d^1$	7	-7	-8	-24
$\gamma^1$	-136	-32	-16	
$-c^2$	11	-8	-10	-26
$-d^2$	9	-8	-10	-21
$\gamma^2$	-3	40	50	
$-c^3$	-9	8	22	22
$-d^3$	8	-5	-8	-18
$\gamma^3$	14	34	220	

TAB. 3.6 – Tableau 6

La solution optimale est  $x^6 = (0, 0, 2)$ ,  $z(x^6) = \left(\frac{-1}{3}, \frac{26}{21}, \frac{11}{9}\right)$ .  
 $ESE(P)$  reste le même, car  $z(x^6)$  est dominé.  $N_5 = \{12, 15, 1\}$ .

k=7

$T_6 = \emptyset$ . Prendre  $j_6 = 15$  et l'arête  $E^{j_6}$  est explorée pour d'éventuelles solutions

entières. Les solutions se trouvant sur  $E^{j_6}$  sont :  $y^5 = (0, 0, 1)$  et  $y^6 = (0, 0, 0)$  correspondants aux vecteurs critère  $z(y^5) = \left(\frac{-7}{17}, \frac{18}{13}, \frac{-14}{13}\right)$  et  $z(y^6) = \left(\frac{-3}{5}, 2, \frac{-3}{4}\right)$  respectivement.

$ESE(P) = \{(0, 3, 0), (1, 2, 0), (0, 2, 0), (1, 1, 0), (2, 0, 0), (0, 1, 0), (1, 0, 0), (0, 0, 0)\}$ . Appliquer la coupe :  $x_1 + x_{12} \geq 1$  i.e.  $-x_1 - x_{12} + x_{16} = -1$ . Lors de l'application de la méthode duale du simplexe, l'opération de pivot devient impossible indiquant que le domaine tronqué restant ne contient aucun point réalisable et que toutes les solutions efficaces du problème étudié ont été obtenues.

### 3.4 Conclusion

Dans ce chapitre nous nous sommes intéressés au problème de l'optimisation multiobjectif fractionnaire linéaire en nombres entiers (MOILFP). On a proposé une méthode pour la détermination de toutes les solutions efficaces du problème (MOILFP) tout en évitant l'exploration de tout le domaine de recherche et ce par élimination des solutions non efficaces sans que nous soyons obligés de les évaluer.

Les résultats obtenus lors de l'implémentation de cette méthode ainsi que les différentes méthodes étudiées dans le chapitre précédent seront donnés au chapitre suivant.

## 4.1 Introduction

Dans ce chapitre, sont résumés les résultats expérimentaux obtenus lors de l'implémentation des différentes méthodes étudiées dans ce mémoire. Nous avons réalisé tous les programmes à l'aide du langage **Matlab** ([10], [3]) qui est un langage de programmation interactive, simple, efficace et optimisé pour le traitement des matrices d'où son nom **Matrix Laboratory**. L'application a été déployée sur une machine Pentium IV à 2.4 GHZ et avec 512 MO de mémoire vive.

## 4.2 Génération aléatoire des données

Afin de tester les différents algorithmes, nous avons constitué un jeu de tests formé d'ensembles de données générées en utilisant la loi uniforme. Les problèmes considérés ont la forme suivante :

$$(P) \begin{cases} \text{Max } f^i(x) & i = \overline{1, r} \\ \sum_{j=1}^n a_{ij}x_j \leq b_i & i = \overline{1, m} \\ x_j \in \mathbb{Z} & j \overline{1, n} \end{cases}$$

Où  $n$  représente le nombre de variables et  $m$  le nombre de contraintes.

Les éléments des matrices  $A$  et  $f^i$  ( $i = \overline{1, r}$ ) sont générés à l'aide de la fonction prédéfinie en matlab  $\text{randint}(k, l, [vmin, vmax])$  qui renvoie une  $(k \times l)$  matrice à coefficients

entiers indépendamment et uniformément distribués dans l'intervalle  $[vmin, vmax]$ .

Pour chaque instance,  $[vmin, vmax]$  est fixé à  $[1, 30]$  pour la matrice de contraintes  $A$ . Concernant les coefficients des critères, ils varient dans l'intervalle  $[-10, 10]$  dans le cas des critères linéaires relativement au problème MOILP. Pour le problème MOILFP, l'intervalle  $[1, 10]$  est retenu pour les coefficients des dénominateurs afin d'assurer la condition de stricte positivité de ces derniers, et l'intervalle  $[-10, 10]$  pour les coefficients des numérateurs. Les seconds membres (vecteur  $b$ ) sont réglées à  $p$  fois la somme des poids de la contrainte correspondante, avec  $p = randint(1, 1, [1, 5])$ . Notons que pour les méthodes décrites dans ([2], [1], [35], [18]), une seule instance est générée pour chaque triplet  $(r, m, n)$ . Tandis que pour la méthode d'optimisation fractionnaire linéaire discrète multiobjectif développée dans le chapitre 3, dix (10) instances sont générées pour chaque triplet  $(r, m, n)$ .

Sans prétendre que ces expériences sont exhaustives, elles nous permettent quand même de déceler les avantages et les inconvénients des algorithmes de recherche étudiés dans ce mémoire à la lumière d'un certain nombre de paramètres que nous avons relevés.

### 4.3 Résultats pour les méthodes de programmation linéaire multiobjectif discrète

#### 4.3.1 Méthode de M. Abbas & M. Moulai [1]

$r$	$m$	$n$	$temps(s)$	$ ESE $	$iter$	$Cut1$	$Dcut$	$edge$	$seg$	$ D $
2	2	2	0.32	7	49	13	3	7	16	23
		3	44.88	29	1274	96	8	41	104	145
		4	20.76	22	1119	83	4	27	87	114
		5	203.56	11	2727	93	2	29	95	124
	3	2	1.33	38	222	29	9	5	38	43
		3	136.41	32	1568	198	1	429	199	628
		4	16.65	6	893	122	2	156	124	280
		5	488.57	5	3646	69	5	12	74	86
3	2	2	2.10	92	230	54	13	25	67	92
		3	130.81	102	1534	173	4	301	177	478
		4	1019.24	25	3044	238	3	203	241	444
		5	58.38	70	1787	154	1	34	155	189
	3	2	1.09	40	165	30	4	15	34	49
		3	7.23	54	555	131	0	368	131	499
		4	12.34	20	1090	39	0	10	39	49
		5	337.47	27	3327	102	1	30	103	133

TAB. 4.1 – Résultats (1)

Où :

- $ESE$  : est l'ensemble des solutions efficaces.
- $iter$  : le nombre d'itérations du simplexe effectuées.
- $Cut1$  : le nombre de coupes de type :  $\sum_{j \in N_k \setminus j_k} x_j \geq 1, j_k \in \Omega_k$  appliquées (voir chapitre 2, section 2.6).
- $Dcut$  : le nombre de coupes de Dantzig appliquées.

- *edge* : le nombre de solutions entières réalisables générées par les arêtes.
- *seg* : le nombre de solutions entières réalisables trouvées par les coupes de Gomory.
- $|D|$  : Cardinalité de l'ensemble d'admissibilité.

#### 4.3.2 Méthode de M. Abbas & D. Chaabane [2]

$r$	$m$	$n$	$temps(s)$	$ ESE $	$iter$	$Cut2$	$Dcut$	$edge$	$seg$	$ D $
2	2	2	0.30	7	18	7	0	16	7	23
		3	1.07	29	141	29	0	116	29	145
		4	4.06	22	559	77	1	36	78	114
		5	47.73	11	1403	65	6	53	71	124
	3	2	1.40	38	118	34	9	0	43	43
		3	8.75	32	572	157	9	462	166	628
		4	22.47	6	926	77	16	187	93	280
		5	593.45	5	4059	45	20	21	65	86
3	2	2	0.95	92	54	21	4	67	25	92
		3	115.60	102	1537	135	2	341	137	478
		4	15.20	25	822	56	0	388	56	444
		5	3.24	70	422	79	0	109	80	189
	3	2	0.260	40	15	7	0	42	7	49
		3	680.4	54	2574	171	167	161	338	499
		4	6.79	20	854	42	6	1	48	49
		5	2351.49	27	5773	102	7	24	109	133

TAB. 4.2 – Résultats (2)

Où :

- *ESE* : est l'ensemble des solutions efficaces.
- *iter* : le nombre d'itérations du simplexe effectuées.
- *Cut2* : le nombre de coupes de type :  $\sum_{j \in N_k \setminus j_k} x_j \geq 1, j_k \in \Psi_k$  appliquées (voir chapitre 2, section 2.7).

- *Dcut* : le nombre de coupes de Dantzig appliquées.
- *edge* : le nombre de solutions entières réalisables générées par les arêtes.
- *seg* : le nombre de solutions entières réalisables trouvées par les coupes de Gomory.
- $|D|$  : Cardinalité de l'ensemble d'admissibilité.

4.3.3 R. Gupta & R. Malhotra (2<sup>ème</sup> procédure) [18]

$r$	$m$	$n$	$temps(s)$	$ ESE $	$iter$	$Sv$	$Sr$	
2	2	2	0.39	9	189	11	0	
		3	1.60	21	207	21	0	
		4	1.17	49	1528	49	0	
		5	0.96	92	1276	92	0	
	3	2	0.41	11	141	11	0	
		3	0.59	23	937	23	0	
		4	0.47	33	876	33	0	
		5	0.67	7	354	7	0	
	3	2	2	1.30	52	2064	103	51
			3	14.49	63	21927	755	131
4			4.07	36	6775	126	46	
5			61.63	45	64745	1171	106	
3		2	1.06	22	745	43	21	
		3	2.53	58	2495	62	3	
		4	10.09	31	5891	95	29	
		5	2.91	12	1995	27	11	
4	2	2	0.49	21	679	47	26	
		3	2.10	35	3363	91	43	
		4	5.77	61	11861	306	122	
		5	1.76	17	2760	125	71	
	3	2	1.32	19	1070	52	33	
		3	7.45	121	11662	499	273	
		4	18.87	305	15125	525	214	
		5	18.23	118	18975	570	163	

TAB. 4.3 – Résultats (4)

Où :

- $ESE$  : est l'ensemble des solutions efficaces.

- *iter* : le nombre d'itérations du simplexe effectuées.
- *sv* : le nombre de solutions entières visitées par la méthode.
- *sr* : le nombre de solutions redondantes trouvées.

Nous avons programmé cette méthode après avoir apporté les rectifications nécessaires (correction du test d'arrêt). Nous avons constaté que les sous problèmes traités par la méthode ne sont pas disjoints ; un même point peut être solution de plusieurs sous problèmes, d'où l'inconvénient. Néanmoins, au vue des résultats fournis par cette méthode, elle est quand même meilleure que celle de Sylva & Crema [35]. Aussi, on remarque qu'elle est indépendante du nombre de solutions non dominées du problème étudié mais son efficacité dépend fortement de celle des coupes de Gomory sur lesquelles elle s'appuie. Il est utile de noter que dans toutes les étapes de la procédure, seulement 2 contraintes sont rajoutées au problème initial et aucune variable supplémentaire n'est rajoutée.

### 4.3.4 Méthode de A. Crema & J. Sylva [35]

$r$	$m$	$n$	$temps(s)$	$ SND $	$iter$	$Vs$	$Cs$	$f_i, i = \overline{1, r}$
2	2	2	0.37	9	776	18	27	1
		3	3.22	21	2191	42	63	1
		4	298.40	49	29572	98	147	1
		5	9704.52	92	161489	184	276	1
	3	2	1.36	11	945	22	33	1
		3	13.33	23	7671	46	69	1
		4	30.95	33	8898	66	99	1
		5	0.49	7	745	14	21	1
3	2	2	2430.56	52	98284	156	208	1
		3	17864.78	63	384262	189	252	1
		4	736.07	36	78587	108	144	1
		5	383.89	45	26596	135	180	1
	3	2	10.37	22	4764	66	88	1
		3	13607.28	58	248575	174	232	1
		4	140.32	31	15148	93	124	1
		5	3.66	12	2412	36	48	1
4	2	2	1877.19	21	288264	84	105	1
		3	1280.25	35	53708	140	175	1
		4	30060.82	61	249675	244	305	1
		5	21.81	17	6964	68	85	1
	3	2	72.37	19	15572	76	95	1
		3	2877.82	39	82789	156	195	2
		4	1038.42	24	74539	96	120	12
		5	727.87	26	55905	104	130	6

TAB. 4.4 – Résultats (3)

Où :

- $SND$  : est l'ensemble des solutions non dominées.

- *iter* : le nombre d'itérations du simplexe effectuées.
- *Vs* : le nombre de variables supplémentaires rajoutées au problème initial.
- *Cs* : le nombre de contraintes supplémentaires rajoutées au problème initial.

Comme nous l'avons déjà signalé au chapitre 2, cette méthode est intimement liée au nombre de solutions non dominées du problème étudié ainsi qu'au nombre de critères considérés. En effet, chaque étape  $k$  de la procédure correspond à une solution non dominée (sauf la dernière étape), où  $r$  variables et  $r + 1$  contraintes sont rajoutées au programme considéré à l'étape  $k - 1$ . A la fin de la procédure, on est amené à résoudre le programme linéaire en nombres entiers initial, augmenté de  $r \times |SND|$  variables et  $(r + 1) \times |SND|$  contraintes, ce qui peut être astronomique. Cependant, vue l'architecture de la méthode, cette dernière est facilement intégrable à n'importe quel logiciel du commerce qui résout des programmes linéaires en nombres entiers tel que Cplex et Lingo, car à chaque étape de l'algorithme, un problème de programmation linéaire en nombres entiers est résolu connaissant les résultats fournis par les étapes précédentes.

Nous rapportons dans ce qui suit, quelques résultats obtenus par les auteurs [35]. La méthode a été testée sur des problèmes types sac à dos en 0 – 1 avec contraintes multiples et 2 critères. Les coefficients des fonctions objectifs ainsi que ceux des contraintes sont des entiers générés aléatoirement de façon uniforme dans l'intervalle  $[1,99]$ . Pour chaque contrainte, la valeur du second membre est fixé à 50% de la somme de ses coefficients. Pour tous les problèmes traités, l'ensemble de toutes les solutions non dominées est généré. La table suivante montre les résultats cumulatifs sur une batterie de 30 instances de différentes tailles :

$m$	$n$	$NPLE$		$iter$	
		<i>Moyenne</i>	<i>Max</i>	<i>Moyenne</i>	<i>Max</i>
5	15	8.2	14	7625.2	36191
5	20	13.8	23	48730.4	153042
5	25	15.5	26	127435.0	936581
5	30	21.8	37	377892.4	1115816
10	15	8.3	20	13177.2	116644
10	20	14.3	31	127575.2	929335
10	25	17.6	29	186130.5	597954
10	30	21.0	33	702280.9	2698297

TAB. 4.5 – Problèmes sac à dos en 0 – 1 avec 2 critères

Où :

- $NPLE$  : représente le nombre de programmes linéaires en nombres entiers résolus,
- $iter$  : le nombre d'itérations effectuées par le simplexe.

Notons que le nombre maximum de programmes linéaires en nombres entiers (PLE) résolus varie entre 14 et 37, ce qui signifie que le nombre maximum de solutions non dominées varie lui aussi entre 13 et 36. Nous estimons que ces valeurs sont relativement faibles par rapport aux tailles des instances traitées, ce qui a engendré un nombre réduit d'itérations  $iter$ .

Les résultats que nous avons rapportés précédemment (Tab. 4.4), montrent que pour des instances de petites tailles, le nombre de solutions non dominées peut être important, ce qui justifie le nombre important d'itérations  $iter$ .

#### 4.4 Résultats pour la méthode d'optimisation multiobjectif fractionnaire discrète

Résultats donnés en moyenne arithmétique									
$r$	$m$	$n$	$temps(s)$	$ ESE $	$Cut3$	$Cut4$	$edge$	$seg$	$ D $
2	2	2	0.11	16	13	3	7	16	43
		3	2.52	18	72	8	17	80	236
		4	35.49	12	137	3	5	139	352
		5	10.58	9	78	2	1	80	156
	3	2	0.09	15	7	4	13	11	38
		3	3.35	16	44	16	10	60	115
		4	36.65	8	117	5	3	122	320
		5	53.73	8	67	1	1	68	126
3	2	2	0.25	22	9	3	13	12	42
		3	3.95	33	42	17	36	59	120
		4	2.79	31	47	15	6	62	200
		5	35.65	22	68	27	10	95	149
	3	2	0.12	16	8	3	7	11	31
		3	3.80	33	35	34	24	69	141
		4	17.73	34	67	22	9	88	150
		5	23.04	16	58	4	2	62	116

TAB. 4.6 – Résultats (5)

$r$	$m$	$n$	Itérations simplexe			Partie non explorée du domaine (%)		
			moyenne	min	max	moyenne	min	max
2	2	2	39	12	83	45.31	0	76.36
		3	256	131	511	44.91	7	83.21
		4	502	144	1052	56.70	33.33	78.34
		5	493	283	713	47.54	28.85	79.58
	3	2	29	14	72	33.92	0	78.43
		3	293	36	639	30.96	0	78.76
		4	474	94	1011	56.50	20	82.07
		5	620	370	1005	45.47	41.93	58.09
3	2	2	26	14	48	44.62	0	74.07
		3	185	32	532	27.59	2.94	59.25
		4	249	137	474	43.00	2.43	89.58
		5	728	583	920	30.53	0	46.93
	3	2	31	13	71	43.48	10.71	75.67
		3	240	73	547	32.78	5.47	63.52
		4	416	87	908	28.88	1.72	59.95
		5	575	323	895	44.41	37.5	52.89

TAB. 4.7 – Résultats (6)

Où :

- $ESE$  : est l'ensemble des solutions efficaces.
- $iter$  : le nombre d'itérations du simplexe effectuées.
- $Cut3$  : le nombre de coupes de type  $\sum_{j \in N_k \setminus j_k} x_j \geq 1, j_k \in T_k$  utilisées (voir chapitre 3).
- $Cut4$  : le nombre de coupes  $\sum_{j \in N_k \setminus j_k} x_j \geq 1, j_k \in N_k$  appliquées (voir chapitre 3).
- $edge$  : le nombre de solutions entières réalisables générées par les arêtes.
- $seg$  : le nombre de solutions entières réalisables trouvées par les Coupes de Gomory.
- $|D|$  : cardinalité de l'ensemble d'admissibilité.

- % : le pourcentage de points de  $D$  qui n'ont pas été explorés.

Les résultats obtenus lors de l'implémentation de la méthode proposée sont satisfaisants, car plusieurs points entiers réalisables sont supprimés sans être évalué. Néanmoins, elle présente deux faiblesses :

- La faiblesse majeure concerne les coupes de Gomory qui ralentissent énormément les calculs. En fait, toutes les méthodes d'optimisation combinatoire basées sur les coupes de Gomory possèdent cet inconvénient.
- La deuxième faiblesse a une incidence moindre sur l'efficacité de la méthode, compte tenu du fait qu'elle n'apparaît que lorsqu'à chaque étape  $k$  de la procédure, l'ensemble  $T_k$  est vide. La méthode explore alors, tout le domaine des solutions réalisables.

## 4.5 Conclusion

La présente étude expérimentale aurait été caduque sans l'étude approfondie des différentes références bibliographiques. L'apprentissage du logiciel de calcul Matlab a facilité la programmation des différentes procédures. Cependant, la difficulté principale lors de la réalisation des différents codes était la maîtrise de la propagation des erreurs d'arrondis dus à l'utilisation des coupes de Gomory. Pour cela, l'utilisation de coupes profondes est envisageable pour remédier à ce genre de problèmes. L'analyse des résultats montre, que même si toutes les méthodes évoquées donnent toutes la réponse favorable aux problèmes posés, elles se distinguent par leurs performances par rapport aux divers paramètres ciblés dans cette étude, à savoir ; temps de calcul, nombre de solutions réalisables générées et nombre de coupes de Gomory utilisées. Celles basées sur les coupes de Gomory convergent très lentement et arrivent juste à résoudre des instances de petite taille. Les faiblesses de la méthode de Sylva et Crema [35] ne sont surmontées même si elle basée sur une méthode Branch & Bound, le nombre de programmes linéaires en nombres entiers à résoudre croît rapidement en fonction du nombre de critères et de celui des solutions non dominées.

## Conclusion générale

Inconsciemment, nous résolvons tous au quotidien des problèmes d'optimisation plus ou moins complexes. Nos achats, nos déplacements ne sont faits qu'après réflexion afin d'aboutir à la décision qui nous arrange le mieux. Par exemple, pour un trajet en véhicule, la décision peut être basée sur des caractéristiques telles que : prendre le trajet le plus court possible, le moins coûteux possible et le plus confortable possible tout en respectant certaines conditions (par exemple passer par certaines villes, arrêts obligatoires ou prévisible, etc.). Ces raisonnements, même s'ils paraissent innocents, font appel au concept de compromis, en ce sens que les décisions prises le sont rarement dans un contexte mono objectif. Plusieurs critères sont simultanément intégrés à la réflexion, afin de dégager un choix final présentant le meilleur compromis entre tous les objectifs. Cette approche nous conduit à considérer une autre catégorie de problèmes d'optimisation : les problèmes multiobjectif.

Dans ce mémoire, nous avons présenté les concepts de base de l'optimisation multiobjectif telles que : la dominance, l'efficacité faible, le concept du cône pour la détection graphique de l'efficacité, etc. En fait, la résolution des problèmes multiobjectif est divisée, fondamentalement, en deux étapes : détermination des solutions efficaces, puis étape de décision (concerne la sélection d'une solution de meilleur compromis entre les solutions efficaces trouvées à la première étape). La première étape consiste à rechercher les solutions efficaces dans l'espace des solutions réalisables, où sont nombreuses les approches proposées. Parmi celles-ci nous avons étudié et examiné

les méthodes décrites dans ([2], [1], [35], [18], [21]), en montrant clairement les avantages et les inconvénients de chacune à travers une étude comparative numérique. Un contre exemple est donné montrant une faille dans la deuxième méthode proposée par R. Gupta & R. Malhotra [18] et à laquelle nous avons proposé des corrections. Aussi, nous avons été amené à décrire une méthode de détermination de toutes les solutions alternatives (adjacentes) d'un problème de programmation linéaire en nombres entiers aux fins de déterminer initialement un sous ensemble de solutions efficaces d'un programme linéaire multiobjectif. Ensuite, notre attention s'est focalisée sur la maîtrise des concepts de la programmation linéaire fractionnaire mono objectif discrète dans le but de surmonter la difficulté de résolution du problème linéaire fractionnaire discret multiobjectif représentant une partie importante de notre travail. A cet effet, en se basant sur un concept introduit par Granot *et al.* [16], et sur une technique de coupes planes due à R. Gupta & R. Malhotra [18], une nouvelle méthode de résolution exacte du problème linéaire fractionnaire à objectifs multiples en variables entière est mise au point dont l'algorithme converge en un nombre fini d'étapes. Une implémentation des différentes méthodes traitées dans ce mémoire en présentant une synthèse des caractéristiques principales de ces méthodes et de leurs difficultés de mise en œuvre est présentée. Il nous a été possible à travers la programmation informatique, de constater la convergence très lente des méthodes utilisant les coupes de Gomory, ce qui a conduit à l'étude expérimentale sur des exemples donnant les résultats en un temps raisonnable. L'adaptation des principes des méthodes de Branch & Bound et Branch & Cut pourrait apporter un gain précieux en temps d'exécution de ces méthodes et permettre le traitement d'instances de tailles plus grandes.

## Bibliographie

- [1] M. Abbas and M. Moulai, *Solving multiple objective integer linear programming*, Journal of the Italian Operations Research Society (Ricerca Operativa) 29, pp. 15-38, (1999).
- [2] M. Abbas and D. Chaabane, *An algorithm for solving multiple objective integer linear programming problem*, RAIRO Operations Research 36, pp. 351-364, (2002).
- [3] P. Armand, *Une brève introduction à Matlab*, [http://www.unilim.fr/page\\_perso/paul.armand/matlab/doc/d.pdf](http://www.unilim.fr/page_perso/paul.armand/matlab/doc/d.pdf).
- [4] G.R. Bitran and G. Novaes, *Linear Programming with a fractional objective function*, Operations Research 24, pp. 675-699, (1976).
- [5] V.J. Bowman, *On the relationship of the tchebycheff norm and the efficient frontier of multiple-criteria objectives*. In H. Thieriez, editor, Multiple Criteria Decision Making, volume 130 of Lecture Notes in Economics and Mathematical Systems. Springer Verlag, Berlin (1976).
- [6] A. Cambini and L. Martein, *A modified version of Martos's algorithm for the linear fractional problem*, Methods of Oper. Res. 53, pp. 33-44, (1986).
- [7] D. Chaabane, *Contribution à l'optimisation multicritère en variables discrètes*, thèse de doctorat, Université des Sciences et de la Technologie Houari Boumediene, Algérie, (2005).

- [8] V. Chankong and Y.Y. Haimes, *Multiobjective decision making : theory and methodology*, North Holland Series in System Science and Engeneering 8, p. 406, (1983).
- [9] A. Charnes and W.W. Cooper, *Programming with linear fractional functionals*, Naval Res. Logist. Quart. 9, pp. 181-186, (1962).
- [10] S. Correia, *Cours d'introduction à Matlab*,  
[http ://s.correia.free.fr/data/matlab.ps](http://s.correia.free.fr/data/matlab.ps).
- [11] R. Dakin, *A tree search algorithm for mixed integer programming problems*, Computer Journal 8, pp. 250-255, (1965).
- [12] G.B. Dantzig, *Linear programming and extention*, Princeton University Press, (1963).
- [13] W. Dinkelbach, *On nonlinear fractional programming*, Management Science 13, pp. 492-498, (1967).
- [14] A.M.Geoffrion, *Proper efficiency and the theory of vector maximization*, Journal of Mathematical Analysis and Applications 22 pp. 618-630 (1968).
- [15] R.E. Gomory, *Outline of an algorithm for integer solutions to linear programs*, Bulletin of the AMS 64, pp. 275-278, (1958).
- [16] D. Granot and F. Granot, *On integer and mixed integer fractional programming problems*, Ann. Discrete Math 1, pp. 221-231, (1977).
- [17] M. Grunspan, *Fractional programming : A survey*, Technical Report 50, Department of Industrial and Systems Engineering, University of Florida, (1971).
- [18] R. Gupta and R. Malhotra, *Multi-criteria integer linear programming problem*, Cahiers de CERO 34, pp. 51-68, (1992).
- [19] R. Gupta and R. Malhotra, *Multi-criteria integer linear fractional programming problem*, Optimization 35, pp. 373-389, (1995).
- [20] J.R. Isbell and W.H. Marlow, *Attribution games*, Naval Res. Logist. Quart. 3, pp. 71-94, (1956).

- [21] D. Klein and E. Hannan, *An algorithm for multiple objective integer linear programming problem*, European Journal of Operational Research 9, pp. 378-385, (1982).
- [22] A.H. Land and A.G. Doig, *An automatic method for solving discrete programming problems*, Econometrica 28, pp. 497-520, (1960).
- [23] B. Martos, *Hyperbolic programming*, Naval Res. Logist. Quart 11, pp. 135-155, (1964).
- [24] M. Michel, *Programmation mathématique*, vol. 1 xxxi, pp. 294, -Paris : Dunod- (1983).
- [25] M. Michel, *Programmation mathématique*, vol. 2 xxxi, pp. 236, -Paris : Dunod- (1983).
- [26] M. Moulaï, *Optimisation multicritère fractionnaire linéaire en nombres entiers*, thèse de doctorat, Université des Sciences et de la Technologie Houari Boumediene, Algérie, octobre (2002).
- [27] M. Moulaï and M. Abbas, *Integer linear fractional programming with multiple objective*, Journal of the Italian Operations Research Society, vol 1 N°1. 103-104, pp. 15-38, (2002).
- [28] S.S. Rao, *Engineering optimization : theory and practice* , Wiley, John & Sons, Incorporated, (1996).
- [29] B. Roy, *Des critères multiples en recherche opérationnelle : pourquoi ?*, in Rand (ed.), Operational Research 87, North-Holland, Amsterdam, pp. 829-842, (1988).
- [30] M. Saad Omar and B. Hughes John, *Bicriterion integer linear fractional programs with parameters in the objective functions*, J. Inform. Optim. Sci. 19, pp. 97-108, (1998).
- [31] M. Sakarovitch, *Optimisation combinatoire. Méthodes mathématiques et algorithmiques. Tome 1 : Graphes et Programmation Linéaire ; Tome 2 : Programmation Discrète*, Paris, Hermann, (1984).
- [32] C.R. Seshan and V.G. Tikekar, *Algorithms for Integer Fractional Programming*, J. of the Indian Institute of Science 62 (B), pp. 9-16, (1980).

- [33] M. Simonnard, *Programmation linéaire : technique du calcul économique*, Paris, (1972).
- [34] R.E. Steuer, *Multiple criteria optimization : theory, computation and application*, Wiley Series in Probability and Mathematical Statistics : Applied Probability and Statistics. New York : John Wiley & Sons, Inc. XXII, p. 546, (1986).
- [35] J. Sylva and A. Crema, *A method for finding the set of non-dominated vectors for multiple objective integer linear programs*, European Journal of Operational Research, in press, (2003).
- [36] H.A. Taha, *Integer programming theory, applications and computations*, Academic Press, New York, (1975).
- [37] E.L. Ulungu, *Optimisation combinatoire multicritère : détermination de l'ensemble des solutions efficaces et méthodes interactives*, thèse de doctorat, université de Mons-Hainaut, octobre (1993).
- [38] V. Verma, *Constrained integer linear fractional programming problem*, Optimization 21 pp. 749-757 (1990).