

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université Abderrahmane Mira-Béjaia



Faculté des Sciences Exactes  
Département de Recherche Opérationnelle

# Mémoire de fin d'étude

pour l'obtention de Diplôme de Master  
Spécialité : Modélisation Mathématique et évaluation Des Performance Des Réseaux

---

## Thème

---

# Optimisation Fractionnaire Multiobjectifs

---

Présenté par :  
Abdelkader Toufik

Encadré par :  
 $M^{me}$  Younsi-Abbaci Leila

- Devant le jury :
- **Président** :  $M^r$  Kabyl Kamal
- **Examinatrice** :  $M^{me}$  Lazari Nassima

**Année universitaire : 2021/2022**

## *Remerciements*

Je tiens tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui m'a donné la force et la patience d'accomplir ce modeste travail.

je tiens également à remercier ma promotrice **Mme Abbaci Leila**, pour son soutien et sa disponibilité durant toute la période du travail.

j'adresse aussi Mes vifs remerciements aux membres des jurys pour avoir bien voulu examiner et juger ce travail :

- *M<sup>r</sup>* k.Kabyl qui m'a fait l'honneur d'accepter la présidence du jury.
- *M<sup>me</sup>* N.Lazari qui ma fait l'honneur d'examiner ce travail.

Je n'oublie pas ma chère famille pour leur contribution, leur soutien et leur patience.

Enfin, j'adresse mes plus sincères remerciements à tous mes proches et amis, qui m'ont toujours encouragées au cours de la réalisation de ce mémoire. . . .

# *Dédicaces*

*Je dédie ce modeste travail :*

*A ma famille, mais aucune dédicace ne serait témoin de mon profond amour, mon immense gratitude et mon plus grand respect pour eux pour leurs soutien durant tous mon cycle d'étude.*

*J'offre aussi ce travail*

*A tous mes amie, et à tous ceux que j'aime et qui ont contribué de loin ou de près afin de réaliser ce travail.*

*A ma promotrice pour ça patience avec moi.*

*Ainsi à tous le corps administratif, et à vous chers lecteurs.*

...

# Table des matières

<b>1</b>	<b>Programmation Linéaire Fractionnaire</b>	<b>10</b>
1.1	Introduction . . . . .	10
1.2	Domaines d'applications des problèmes fractionnaires . . . . .	10
1.3	Formulation du programme Linéaire fractionnaire . . . . .	10
1.4	Le modèle générale d'un programme Linéaire fractionnaire . . . . .	11
1.5	Rappel sur l'optimisation convexe . . . . .	12
1.5.1	Convexité . . . . .	13
1.6	Résolution d'un programme linéaire fractionnaire . . . . .	13
1.6.1	La résolution graphique en nombres entiers . . . . .	13
1.6.2	La résolution analytique . . . . .	15
1.6.3	Méthode de Cambini et Al [24] . . . . .	17
1.6.4	la méthode de Dinkelbach[14] . . . . .	18
1.6.5	la méthode de gradient . . . . .	19
1.7	Conclusion . . . . .	19
<b>2</b>	<b>Généralité sur l'optimisation multiobjectifs</b>	<b>20</b>
2.1	Introduction . . . . .	20
2.2	Notion fondamentale . . . . .	20
2.2.1	Définition du problème . . . . .	20
2.3	Problématique . . . . .	22
2.3.1	La relation de dominance . . . . .	23
2.3.2	Optimalité de Pareto . . . . .	24
2.3.2.1	Effcacité forte . . . . .	24
2.3.2.2	Effcacité faible . . . . .	24
2.3.3	Point particuliers . . . . .	25
2.4	Structure du front de Pareto . . . . .	27
2.5	Classification des méthodes de résolution . . . . .	28
2.5.1	Méthode $\varepsilon$ contrainte [8] . . . . .	28
2.5.2	Méthode dite "but programmé ou Goal programming" . . . . .	29
2.5.3	Méthode des sommes pondérées [30] . . . . .	30

## TABLE DES MATIÈRES

---

2.6	Approche de résolution . . . . .	31
2.6.1	Approche Scalaire . . . . .	31
2.6.2	Approche Pareto . . . . .	32
2.7	Conclusion . . . . .	32
<b>3</b>	<b>Programmation Linéaire Fractionnaire MultiObjectifs en Nombres Entiers</b>	<b>33</b>
3.1	Introduction . . . . .	33
3.2	Formulation du problème . . . . .	33
3.2.1	Complexité du problème . . . . .	35
3.2.2	Solutions supportées et non supportées . . . . .	36
3.3	Méthodes de résolution d'un programme linéaire en nombres entiers . . . .	37
3.3.1	La méthode de gomory [27] . . . . .	37
3.3.2	Méthode de Branch et Bound . . . . .	38
3.4	Résolution du problème . . . . .	40
3.4.1	Méthodes de Abbas et Moulaï [19] . . . . .	41
3.4.2	Méthode de Chergui et Moulaï [20] . . . . .	46
3.5	Conclusion . . . . .	47
<b>4</b>	<b>Approche de résolution du problème MOIFLP</b>	<b>48</b>
4.1	Introduction . . . . .	48
4.2	Définition du problème . . . . .	48
4.2.1	Problème fractionnaire multi-objectif . . . . .	48
4.2.2	Transformation du problème (MOLIFP) vers un problème mono-objectif [30] . . . . .	49
4.3	Exemple numérique . . . . .	49
4.4	Présentation du logiciel . . . . .	50
4.4.1	Interface de logiciel : . . . . .	51
4.5	Conclusion . . . . .	53

# Table des figures

1.1	Graphe de l'exemple. . . . .	14
2.1	Représentation de l'espace des décisions et l'espaces objectifs correspondant.	22
2.2	Schéma de résolution . . . . .	23
2.3	Illustration des différentes définitions. . . . .	27
2.4	La représentation de la surface de compromis . . . . .	28
2.5	Représentation graphique de la méthode de la somme pondérée . . . . .	31
3.1	Solutions supportées et non supportées . . . . .	36
3.2	Principe de Branch and Bound . . . . .	40
3.3	itération 1 . . . . .	44
3.4	itération 2 . . . . .	44
3.5	itération 3 . . . . .	45
3.6	itération 6 . . . . .	45

# Liste des tableaux

3.1	simplexe optimal associé à la base B. . . . .	37
4.1	Résultats de l'optimisation en faisant varier les poids $w_1$ et $w_2$ . . . . .	53

# Introduction générale

La Recherche Opérationnelle peut être définie comme l'ensemble des méthodes et des techniques rationnelles orientées vers la recherche du meilleur choix dans la façon d'opérer en vue d'aboutir au résultat visé ou au meilleur résultat possible.

La programmation fractionnaire a attiré l'attention de nombreux chercheurs dans le passé. La principale raison de l'intérêt dans la programmation fractionnaire provient du fait que modèles de programmation pourraient mieux répondre aux problèmes réels si l'on considère l'optimisation des le rapport entre les propriétés physiques et / ou les quantités économiques. Etude bibliographique révèle de larges applications de programmation fractionnaire dans différents domaines allant de la l'ingénierie à l'économie.

Dans un problème mono-objectifs,un seul objectifs exprimé en fonction des variable de décision doit être otimisé.ependant dans les situation rencontrées en pratique,la considération d'un seul objectifs ne permet pas une bonne modélisation du problème. Avoir comme but de se ramener à une seule fonction objectif peut aussi biaiser la modélisation. L'optimisation multiobjectifs autorise ces degrés de liberté qui manquaient en optimisation mono-objectif. La recherche ne nous donnera plus une solution unique mais une multitude de solutions. La première notion d'optimalité a été introduite par Francis Ysidro Edgeworth [21] en 1881, elle a été utilisée de manière plus formelle par l'économiste italien Vilfredo Pareto [22] en 1896. Ces solutions sont appelées solutions de Pareto et l'ensemble de solutions que l'on obtient à la fin de la recherche est la surface de compromis.

A cet effet, nous avons structuré notre travail comme suit :

- **Chapitre 1** :Nous Nous sommes intéressés à la programmation linéaire fractionnaire en nombres entiers, en rappelant les définitions de base et quelques méthodes de résolution de ce type de problème.
- **Chapitre 2** : Nous avons donné les principales définitions liées à l'optimisation linéaire multi-objectifs puis les problématiques spécifiques à ce domain sont exposées et étudiées. Parmi ces problématiques, nous avons parlé en particulier de la structure



de l'ensemble Pareto et le choix des méthodes de résolution.

- **Chapitre 3** : est consacré à la présentation des problèmes de programmation multi-objectifs linéaire fractionnaire en nombre entier (*MOILFP*) et à l'étude des méthodes principales de résolution.
- **Chapitre 4** : est consacré sur l'approche de résolution qui consiste à convertir le problème multiobjectifs fractionnaire linéaire vers un problème mono-objectif.
- Enfin, le mémoire s'achèvera par une conclusion générale.

# Programmation Linéaire Fractionnaire

## 1.1 Introduction

Les programmes linéaire fractionnaires , c'est-à-dire les programmes dont l'objectif s'exprime comme le rapport de deux fonctions, linéaires ou non linéaires, en variables réelles, en variables entières, en variables mixtes ou en variables 0-1 apparaissent dans plusieurs domaines de la recherche opérationnelle tels que les bases de données, l'ingénierie, l'optimisation combinatoire, la programmation stochastique, l'informatique et l'économie.

## 1.2 Domaines d'applications des problèmes fractionnaires

Comme domaines d'applications des problèmes fractionnaires soumis à un ensemble de contraintes, nous pouvons citer par exemple[28] :

- Le domaine des finances où il est souvent question d'optimiser un rapport de deux fonctions telles que [dette/ capitaux propres], [rendement/employé].
- Le domaine de l'économie offre une large éventail d'applications. En effet, la mesure de l'efficacité des systèmes étudiés s'exprime sous forme de rapports entre les critères techniques et/ou économiques. Par exemple, [rendement/risque].

## 1.3 Formulation du programme Linéaire fractionnaire

Les programmes fractionnaires consistent à optimiser un objectif mis sous la forme d'un rapport de deux fonctions linéaires ou non linéaires, soumis à un ensemble de contraintes[1].

Etant donné  $f, h$  et  $g_i, i=\overline{1, m}$  des fonctions réelles définies sur  $R^n$ . Désignons par  $X$  un sous-ensemble de  $R^n$  sur lequel  $h$  ne s'annule pas. Le problème de programmation fractionnaire consiste à déterminer un élément  $x^*$  de  $X$  maximisant la fonction  $f/h$  sur un domaine défini par le système de contraintes  $\{x \in X : g_i(x) \leq 0, i=1 \dots m\}$ . Un problème de programmation fractionnaire est donc de la forme suivante :

$$(PFL) = \begin{cases} \min & Z(x) = \frac{f(x)}{h(x)} \\ s.c & \\ g_i(x) \leq 0 & i = 1 \dots m \\ x \in X & \end{cases}$$

avec :

Les fonctions  $f, h$  et  $g_i, i = \overline{1, m}$ , sont continues sur  $R^n$

$h(x) > 0, \forall x \in X$

soit la solution réalisable  $S$  est donnée par :

$$S = \{g_i(x) \leq 0, x \in X\}.$$

**Définition 1.3.1.** . Le problème (PF) de maximisation est appelé problème fractionnaire concave-convexe si  $f$  est concave et les fonction  $h$  et  $g_i$  pour  $i=\overline{1, m}$  sont convexes.

. Le problème (PF) de minimisation est appelé problème fractionnaire convexe-concave si  $f$  et les fonctions  $g_i$  pour  $i=\overline{1, m}$  sont convexes et  $h$  concave [2].

## 1.4 Le modèle générale d'un programme Linéaire fractionnaire

Lorsque dans le programme fractionnaire ( $PFL$ ) les fonctions  $f, h$  et  $g_i, i=\overline{1, m}$  sont des fonctions linéaires ou affines de la variable  $x \in R_+^n$ . Alors le problème (PF) d'optimisation est appelé programme fractionnaire linéaire (PFL), ou programme hyperbolique, et prend alors la forme suivante :

$$(PFL) = \begin{cases} \min & Z(x) = \frac{p^t x + \alpha}{q^t x + \beta} \\ s.c & \\ Ax \leq b & \\ x \geq 0 & \end{cases}$$

Où  $\alpha$  et  $\beta$  sont des réels,  $p$  et  $q$  sont des vecteurs de  $R^n$  avec  $q$  est un vecteur non nul,  $A$  est une matrice réelle de format  $(m \times n)$  et  $b$  est un vecteur de  $R^m$ . Désignons par  $S$

l'ensemble

$$\{ \mathbf{x} \in R^n : A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq 0 \}$$

avec :  $qx + \beta \neq 0, \forall \mathbf{x} \in \mathbf{S}$ .

Il est clair que si  $q$  est un vecteur nul avec  $\beta \neq 0$ , alors (PFL) n'est autre qu'un problème de programmation linéaire. Si de plus, les variables sont astreintes à ne prendre que des valeurs entières ( $\mathbf{x} \in Z^n$ ), on parle dans ce cas de problème de programmation linéaire fractionnaire en nombre entiers ou encore problème hyperbolique discret, noté (ILFP), qui fait une partie d'objet de nos préoccupations dans la suite de ce travail.

## 1.5 Rappel sur l'optimisation convexe

**Définition 1.5.1.** Une fonction  $f : R^n \rightarrow R$  est dite concave si et seulement si :  $\forall (x, y) \in R^n \times R^n, \forall \lambda \in [0, 1], f(\lambda x + (1 - \lambda)y) \geq \lambda f(x) + (1 - \lambda)f(y)$

**Définition 1.5.2.** Une fonction  $f : R^n \rightarrow R$  est dite convexe si et seulement si :  $\forall (x, y) \in R^n \times R^n, \forall \lambda \in [0, 1], f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$

**Définition 1.5.3.** Soit  $f : R^n \rightarrow R$  une fonction différentiable sur  $R^n$ . Alors,  $f$  est dite pseudo-concave si et seulement si :  $(x, y) \in R^n \times R^n$  tel que  $\nabla f(x)(y - x) \leq 0$ , on a :  $f(y) \leq f(x)$ .

**Définition 1.5.4.** Soit  $f : R^n \rightarrow R$  une fonction différentiable sur  $R^n$ . Alors,  $f$  est dite pseudo-convexe si et seulement si :  $(x, y) \in R^n \times R^n$  tel que  $\nabla f(x)(y - x) \geq 0$ , on a :  $f(y) \geq f(x)$ .

**Définition 1.5.5.** On dit qu'une fonction  $f : R^n \rightarrow R$  est quasi-convexe si et seulement si :  $\forall (x, y) \in R^n \times R^n, \forall \lambda \in [0, 1], f(\lambda x + (1 - \lambda)y) \leq \max\{f(x), f(y)\}$

**Lemme 1.5.1.** Si  $\frac{f}{h}$  est une fonction fractionnaire linéaire sur un ensemble  $D$  de  $R^n$  telle que  $\forall x \in D : h(x) > 0$ , alors [3] :

- pour chaque couple  $(x, y) \in D^2$ , on a :
  - $\frac{f(x)}{h(x)} \leq \frac{f(y)}{h(y)}$  si et seulement si  $\nabla[\frac{f(x)}{h(x)}](y - x) \geq 0$
  - $\frac{f(x)}{h(x)} < \frac{f(y)}{h(y)}$  si et seulement si  $\nabla[\frac{f(x)}{h(x)}](y - x) > 0$
- $\frac{f}{h}$  est simultanément pseudo-concave et pseudo-convexe.

### 1.5.1 Convexité

L'ensemble  $\mathbb{X}$  est dit convexe si tout segment joignant deux points quelconques de  $\mathbb{X}$  est inclus dans  $\mathbb{X}$ .

La convexité est le premier indicateur de la difficulté du problème. En effet, certaines méthodes sont dans l'incapacité de résoudre des problèmes non convexes de manière optimale.

## 1.6 Résolution d'un programme linéaire fractionnaire

### 1.6.1 La résolution graphique en nombres entiers

Comme l'a noté Steuer[23], Les programmes fractionnaires linéaires présentent un intérêt particulier mis en évidence par la linéarité des courbes niveaux de leurs fonctions critères. En effet, pour illustrer cet aspect, considérons une  $Z$ -courbe niveau quelconque de la fonction critère :  $\bar{Z} = \frac{p^t x + \alpha}{q^t x + \beta}$

après simplification, nous obtenons :

$$\begin{aligned} \bar{Z}(q^t x + \beta) &= p^t x + \alpha \\ (p - \bar{Z}q)^t x &= \bar{Z}\beta - \alpha \end{aligned}$$

qui est une expression linéaire de la  $\bar{Z}$ -courbe niveau de la fonction critère. Puisque  $\bar{Z}$  est quelconque, on voit que chaque courbe niveau du critère fractionnaire linéaire est linéaire sur  $S$ , à condition que le dénominateur ne soit pas nul sur  $S$ . Donc, si un programme fractionnaire linéaire unicritère possède une solution optimale, alors au moins un point extrême de  $S$  est optimal.

En dépit de la linéarité de la courbe niveau de la fonction objectif, les courbes niveaux ne sont pas parallèles (lorsque  $p \neq 0$ ,  $q \neq 0$  et  $p \neq wq$  pour tout  $w \in R$ ) comme ils le sont en programmation linéaire. L'ensemble rotation est l'ensemble de tous les points d'intersection entre la 0-courbe niveau du numérateur et la 0-courbe niveau du dénominateur[23]. L'ensemble rotation est appelé point de rotation dans  $R^2$  et axe de rotation dans  $R^3$ . Les points de cet ensemble sont déterminés par la résolution du système suivant :

$$p^t x = -\alpha \text{ et } q^t x = -\beta$$

#### Exemple illustratif

Considérons le programme fractionnaire linéaire suivant :

$$(PFL) = \begin{cases} \max & Z = \frac{x_1 + x_2 - 1}{5x_1 + x_2 - 1} \\ \text{s.c.} & \\ & 5x_1 + 2x_2 \geq 6 \\ & x_1 \leq 3 \\ & x_2 \leq 3 \\ & x_1, x_2 \geq 0 \end{cases}$$

dont le graphe est donné par la figure suivante :

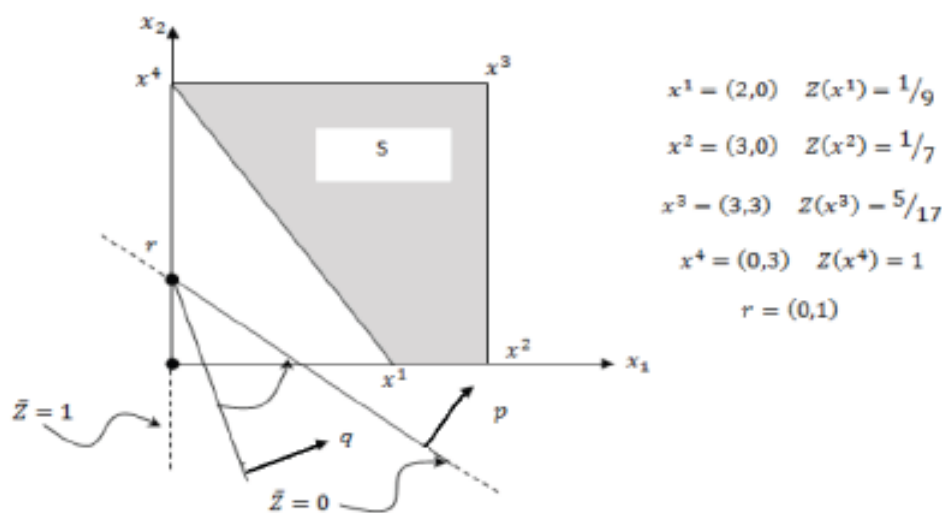


FIGURE 1.1 – Graphe de l'exemple.

Donc la courbe de niveau  $\bar{Z}$  est l'ensemble des points  $x = (x_1, x_2)$  satisfaisant l'équation :

$$(1 - 5\bar{Z})x_1 + (1 - \bar{Z})x_2 = 1 - \bar{Z}$$

donc pour :

$$\bar{Z} = 0 \iff x_1 + x_2 = 1 \quad \text{courbe de niveau 0}$$

$$\bar{Z} = 1 \iff x_1 = 0 \quad \text{courbe de niveau 1}$$

Les lignes discontinues représentent les 0-courbes niveaux du numérateur et du dénominateur dont l'intersection est le point de rotation  $r=(0,1)$ . La flèche circulaire dénote le gradient de la fonction fractionnaire linéaire critère et indique le sens et l'angle avec lesquels se déplacent les courbes de niveaux. Donc, en faisant déplacer la courbe de niveau 0 autour du point de rotation suivant le sens de rotation trigonométrique, nous pouvons voir que le point optimal  $x^4=(0,3)$  de valeur optimale  $Z^* = 1$  est l'intersection du domaine S avec la courbe de niveau  $Z = 1$ .

## 1.6.2 La résolution analytique

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

- 1- **La résolution directe** : Le programme reste sous sa forme originale.
- 2- **La résolution par paramétrisation** : Dans cette stratégie en modifient la fonction objectif sans toucher l'ensemble des contraintes.
- 3- **La résolution d'un programme équivalent** : On change la fonction objectif et l'ensemble des contraintes.

### La résolution directe

Dans cette stratégie de résolution, le programme fractionnaire est traité sous sa forme originale c'est à dire sans modifier ni la fonction objectif ni l'ensemble des contraintes, elle est utilisé pour résoudre les problèmes fractionnaires linéaires continus, entiers, bivalentes(0-1), mixtes.

Pour la recherche d'une solution optimale d'un programme hyperbolique en variables continues sans faire aucun changement au programme initial il existe plusieurs algorithmes et parmi les plus récents, celui de A. Cambini et al[24]. qui permet aussi d'optimiser le problème hyperbolique sur un domaine de solutions réalisables S non borné. On considère donc le programme hyperbolique continu (PFL) :

$$(PFL) = \begin{cases} \min & \frac{p^t x + \alpha}{q^t x + \beta} \\ s.c & \\ Ax \leq b & \\ x \geq 0 & \end{cases}$$

Où  $\alpha$  et  $\beta$  sont des réels, p et q sont des vecteurs de  $R^n$  avec q est un vecteur non nul, A est une matrice réelle de format  $(m \times n)$  et b est un vecteur de  $R^m$ . Désignons par S l'ensemble  $\{x \in R^n : Ax \leq b, x \geq 0\}$

### La résolution par paramétrisation

à l'inverse de la résolution directe, on construit un problème à objectif simplifié, combinaison linéaire du numérateur et du dénominateur par l'intermédiaire d'un paramètre, tout en gardant inchangé l'ensemble des contraintes. Une séquence de résolutions de ce type de problème fournit une solution du programme fractionnaire. Cette méthode a été utilisée pour les différents programmes fractionnaires linéaires et non linéaires, en variables continues comme en variables discrètes, sur des domaines bornés.

Afin de simplifier l'objectif du programme mathématique, un paramètre est introduit permettant de ramener un programme hyperbolique en un programme paramétré linéaire, tout en gardant l'ensemble des contraintes inchangé. Ainsi le programme obtenu peut être

résolu " paramétriquement " : une séquence de résolutions de tels programmes à objectif simplifié engendre une suite de solutions convergeant vers une solution optimale du programme fractionnaire initial.

Proposée initialement en 1956 par Isbell et Marlow [25] pour des programmes hyperboliques, ce n'est qu'à partir de 1967 que cette approche a connu un grand élan avec Dinkelbach qui l'a généralisée aux programmes fractionnaires non linéaires. Soit le programme hyperbolique suivant :

$$(PFL) = \begin{cases} \mathbf{min} & Z(x) = \frac{p^t x + \alpha}{q^t x + \beta} \\ s.c & \\ Ax \leq b & \\ x \geq 0 & \end{cases}$$

Où  $\alpha$  et  $\beta$  sont des réels,  $p$  et  $q$  sont des vecteurs de  $R^n$  avec  $q$  est un vecteur non nul,  $A$  est une matrice réelle de format  $(m \times n)$  et  $b$  est un vecteur de  $R^m$ . Désignons par  $S$  l'ensemble  $\{x \in R^n : Ax \leq b, x \geq 0\}$

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

$$(PFL) = \begin{cases} \mathbf{min} & (p^t - \lambda q^t)x + \alpha - \lambda\beta \\ s.c & \\ Ax \leq b & \\ x \geq 0 & \end{cases}$$

Remarquons que cette nouvelle fonction à maximiser est linéaire alors que celle du programme (PFL) ne l'est pas.

Soit  $x$  une solution optimale de (PFL) et  $\lambda^* = \frac{p^t x^* + \alpha}{q^t x^* + \beta}$ .

Dinkelbach[14] a montré que :

$$Z(\lambda) = 0 \Leftrightarrow \lambda = \lambda^*$$

et qu'une solution optimale de  $(PFL_{\lambda^*})$  est aussi solution optimale de (PFL).

Ainsi, résoudre (PFL) est équivalent à trouver  $\lambda$  avec  $Z(\lambda) = 0$

### La résolution d'un programme équivalent

La transformation du programme fractionnaire en un programme équivalent à objectif non fractionnaire est obtenue par un changement de variables [26]. A l'inverse de l'approche paramétrée, ce changement de variables induit l'ajout d'une contrainte et d'une variable. Plus précisément, cette transformation, proposée par Charnes et Cooper [29] pour le programme hyperbolique en variables continues, Cette transformation en un programme linéaire équivalent a pour but d'appliquer les algorithmes standards tels que la méthode du Simplexe.



Soit le problème (PFL) suivant :

$$(PFL) = \begin{cases} \mathbf{max} & Z(x) = \frac{p^t x + \alpha}{q^t x + \beta} \\ s.c & \\ Ax \leq b & \\ x \geq 0 & \end{cases}$$

En fait le changement de variable suivant :

$$z = \frac{1}{q^t x + \beta} \quad y = \left(\frac{1}{q^t x + \beta}\right)x \quad x = \frac{y}{z}$$

alors :

$$\max Z = z(p^t x + \alpha)$$

$$\max Z = z(p^t(\frac{y}{z}) + \alpha) = p^t y + \alpha z \dots\dots\dots(1)$$

$$Ax \leq b \Leftrightarrow A(\frac{y}{z}) \leq b \Leftrightarrow A_y - bz \leq 0 \dots\dots\dots(2)$$

$$\begin{aligned} z = \frac{1}{q^t x + \beta} &\Leftrightarrow (q^t x + \beta) = \frac{1}{z} \\ &\Leftrightarrow (q^t(\frac{y}{z}) + \beta)z = 1 \\ &\Leftrightarrow q^t y + \beta z = 1 \dots\dots\dots(3) \end{aligned}$$

$$x \geq 0 \Leftrightarrow \frac{y}{z} \geq 0 \Leftrightarrow y \geq 0 \text{ et } z > 0 \dots\dots\dots(4)$$

Alors de (1), (2), (3) et (4) en obtient le problème fractionnaire linéaire équivalent (PFLE) suivant :

$$(PFLE) = \begin{cases} \mathbf{max} & p^t y + \alpha z \\ s.c & \\ Ay - bz \leq 0 & \\ q^t y + \beta z = 1 & \\ y \geq 0, z > 0 & \end{cases}$$

### 1.6.3 Méthode de Cambini et Al [24]

la méthode proposé par Cambini et Al[24] utilise le concept de solution niveau optimale dont la définition est la suivante :

**Définition 1.6.1.** un point  $x^*$  est une solution niveau optimale pour le problème(PFL)si et seulement si  $x^*$  est une solution optimale du problème :

$$(P_{x^*}) \begin{cases} \mathbf{min} & f(x) = cx + c_0 \\ & Ax \leq b, x \geq 0 \\ & qx = qx^* \end{cases}$$

De plus, si  $x^*$  est un point extrême de  $\mathbb{X} = \{x \in R^n / Ax \leq b, x \geq 0\}$  on dira que  $x^*$  est une solution optimale niveau de base .

**théorème 1.6.1.** *un point  $x^0$  est une solution optimale du problème (PFL) si et seulement si le vecteur gradient réduit  $\hat{\gamma} = \hat{d}_0\hat{c} - \hat{c}_0\hat{d}$  et tel que  $\hat{\gamma}_j \leq 0$  pour tout indice hors base  $j \in N$ .*

**Algorithme 1 :** Algorithme de Cambini et Al

**Etape 1**

Trouvé la solution optimale niveau  $x^1$ .

si une telle solution n'existe pas, alors  $\sup_{x \in S} Z(x) = +\infty$ , **Terminer.**

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

**Etape 2 :**

Si  $J = j/\hat{q}_j > 0 = \emptyset$  ; , terminer.  $x^i$  est une solution optimale du problème (PFL).

Sinon, soit  $k$  tel que  $\frac{\hat{p}_k}{\hat{q}_k} = \max_{j \in J} \hat{p}_j/\hat{q}_j$

Si  $\hat{\gamma}_k > 0$ , aller à l'étape 3

Si  $\gamma \leq 0$ , terminer.  $x^i$  est une solution optimale de (PLE).

**Etape 3 :**

La variable hors base  $x_{N_k}$  entre dans la base en moyennant une opération pivot.

Si une telle opération est possible, Poser  $i := i + 1$  et aller à l'étape 2.

Si une telle opération n'est pas possible. Terminer :  $\sup_{x \in S} Z(x) = \frac{\hat{p}_k}{\hat{q}_k}$ .

### 1.6.4 la méthode de Dinkelbach[14]

Une des stratégies les plus célèbres et générales pour la résolution des programmes fractionnaire (non nécessairement linéaires) est l'approche paramétrique décrite par Dinkelbach [14]. dans le cas de la programmation fractionnaire linéaire, cette méthode réduit la solution d'un problème à la solution d'une séquence de programmes linéaires.

Considérons le problème de programmation fractionnaire suivant :

$$(PFL) \begin{cases} \mathbf{min} & Z(x) = \frac{f(x)}{h(x)} \\ & s.c \\ & x \in S \end{cases}$$

L'ensemble  $S$  est supposé non vide, compact (ent un ensemble fermé) dans  $R^n$ , les fonctions  $f(x), h(x)$  sont des fonctions continues à valeurs réelles dans  $S, h(x) > 0, \forall x \in S$  Le problème paramétré associé  $Q(\lambda)$  consiste à simplifier l'objectif en combinant linéairement le numérateur et le dénominateur par l'intermédiaire d'un paramètre réel  $\lambda$ .

$$(Q(\lambda)) \begin{cases} \mathbf{min} & v(\lambda) = f(x) - \lambda h(x) \\ s.c & \\ x \in S, \lambda \in R & \end{cases}$$

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

**théorème 1.6.2.** [14] *Toute solution optimale  $\lambda^*$  de problème  $Q(\lambda^*)$  est une solution optimale du problème  $(PFL)$ .*

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

**proposition 1.6.1** :[14]

La fonction  $\lambda \rightarrow v(\lambda)$  est continue, strictement décroissante, convexe  $v(0) > 0$  et  $v(\lambda)$  tend vers  $-\infty$  quand  $\lambda$  tend vers  $+\infty$ .

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

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

**proposition 1.6.2** :[14]

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

### 1.6.5 la méthode de gradient

ne nécessite aucun changement de variables ni l'ajout de nouvelles variables ou de contraintes et l'utilisation de la méthode classique du simplexe reste toujours possible. en effet dans cette méthode une séquence de programmes linéaires sont résolus, dans chacun de ces programmes linéaires l'objectif fractionnaire est remplacé par le gradient de ce dernier au point optimale du programme précédent

## 1.7 Conclusion

Dans ce chapitre nous avons fait un rappel sur quelques notions mathématiques, telle que les définitions, les notions de convexité, et les fonctions convexe généralisées. nous avons donné un aperçu sur la programmation fractionnaire linéaire telle que le modèle générale, les domaines d'applications, et les quatre grandes méthodes de résolution : graphique, directe, par paramétrisation, et la résolution avec un programme équivalent.

# Chapitre 2

## Généralité sur l'optimisation multiobjectifs

### 2.1 Introduction

Dans la vie réelle, les ingénieurs sont face à des problèmes de complexité augmentante qui surgissent quotidiennement dans divers secteurs (transport, télécommunication, économie, ...etc). Parmi ces problèmes, on retrouve les problèmes d'optimisation multi-objectifs décrits sous plusieurs objectifs à optimiser.

Un problème d'optimisation multi-objectifs consiste à rechercher les meilleures solutions compromises entre les critères à optimiser, connues par les solutions Pareto optimales, qui correspondent mieux aux préférences du décideur. L'une des questions les plus difficiles est donc liée à l'identification de ces solutions ou d'une approximation de celles-ci pour des problèmes complexes.

Ce chapitre présente le contexte de l'optimisation multi-objectifs qui sera le cadre de notre travail, Nous introduisons les principales définitions telles que la dominance, la surface de compromis ainsi les principales approches de résolution.

### 2.2 Notion fondamentale

#### 2.2.1 Définition du problème

Tout d'abord définissons trois notions de base d'un problème d'optimisation multi-objectifs :

- **Fonction objectif** : La fonction objectif est la fonction à optimiser (maximiser ou minimiser), appelée aussi fonction d'adaptation, fonction coût, ou encore performance. On la note  $Z(x)$ .
- **Paramètre** : Le paramètre est une variable qui exprime une donnée quantitative

ou qualitative au problème (exemple : coût, temps, taux d'erreur, ...etc). Ce sont les variables de la fonction objectif, appelée aussi variables de conception, variables d'optimisation ou encore variables de projet.

- **Contrainte** : La contrainte est une condition qui doit être respectée par le vecteur de décision pour trouver la solution admissible.

Un problème d'optimisation multiobjectifs consiste à optimiser  $k$  fonctions objectifs simultanément ( $k \geq 2$ ). Il est défini sous la forme suivante :

$$(PMO) \begin{cases} \text{optimiser} & Z(x) = (Z_1(x), Z_2(x), \dots, Z_k(x)) \\ \text{s.c} & x \in S \end{cases}$$

Où :

- $x = (x_1, x_2, x_3, \dots, x_n)^t \in R^n$  représente le vecteur de décision avec  $i = \overline{1, n}$
- $R^n$  s'appelle espace de décision.
- $Z(x) = (Z_1(x), Z_2(x), \dots, Z_k(x)) \in R^k$  représente le vecteur des fonctions objectifs avec  $Z_i$  les fonctions objectifs ( $k \geq 2$ ).
- $R^k$  s'appelle espace des critères ou espace des objectifs.

Avec :

$$Z : R^n \rightarrow R^k \\ x \mapsto Z(x) = (Z_1(x), Z_2(x), \dots, Z_k(x))$$

Et :

$$Z_i : R^n \rightarrow R \\ x \mapsto Z_i(x), i = \overline{i, k}.$$

- $S$  est l'ensemble des solutions réalisables,  $S \subset R^n$ .
  - $D = Z(S)$  est l'image de  $S$  par  $Z$ , il s'appelle espace des critères réalisables  $D \subset R^k$ , il est donnée par :  $D = Z(S) = \{z \in R^k / z = Z(x); x \in S\}$
- Où  $z$  est le vecteur de l'espace des critères qui correspond à la solution réalisable  $x$ .  
pour cela nous avons présenté l'espace de décision et l'espace des critères dans la figure suivante :

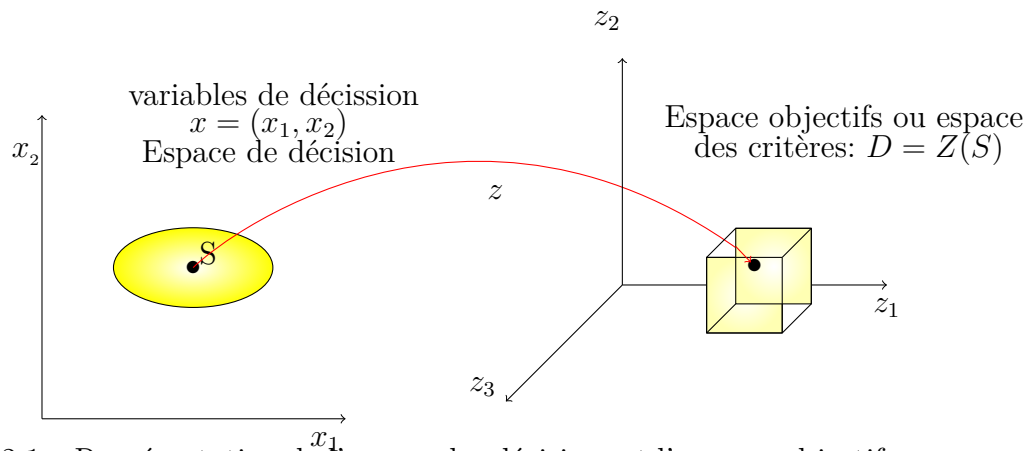


FIGURE 2.1 – Représentation de l'espace des décisions et l'espaces objectifs correspondant.

## 2.3 Problématique

La difficulté principale d'un problème multiobjectif est qu'il n'existe pas de définition de la solution optimale. Le décideur peut simplement exprimer le fait qu'une solution est préférable à une autre mais il n'existe pas une solution meilleure que toutes les autres.

Dés lors résoudre un problème multiobjectif ne consiste pas à rechercher la solution optimale mais l'ensemble des solutions satisfaisantes pour lesquelles on ne pourra pas effectuer une opération de classement. Les méthodes de résolution de problème multiobjectif sont donc des méthodes d'aide à la décision car le choix final sera laissé au décideur.

Pour répondre à ce problème, il existe deux types de comportement. Le premier est de ramener un problème multiobjectif à un problème uni-critère au risque d'enlever toute signification au problème. Le second comportement est d'apporter des réponses au problème en prenant en compte l'ensemble mono-objectif. La différence entre ces deux communautés s'exprime dans le schéma ci-dessous. Soit le décideur intervient dès le début de la définition du problème, en exprimant sa préférence, afin de transformer un problème multiobjectif en un problème uni-critère. Soit le décideur effectue son choix dans l'ensemble des solutions proposées par le solveur multi-objectif.

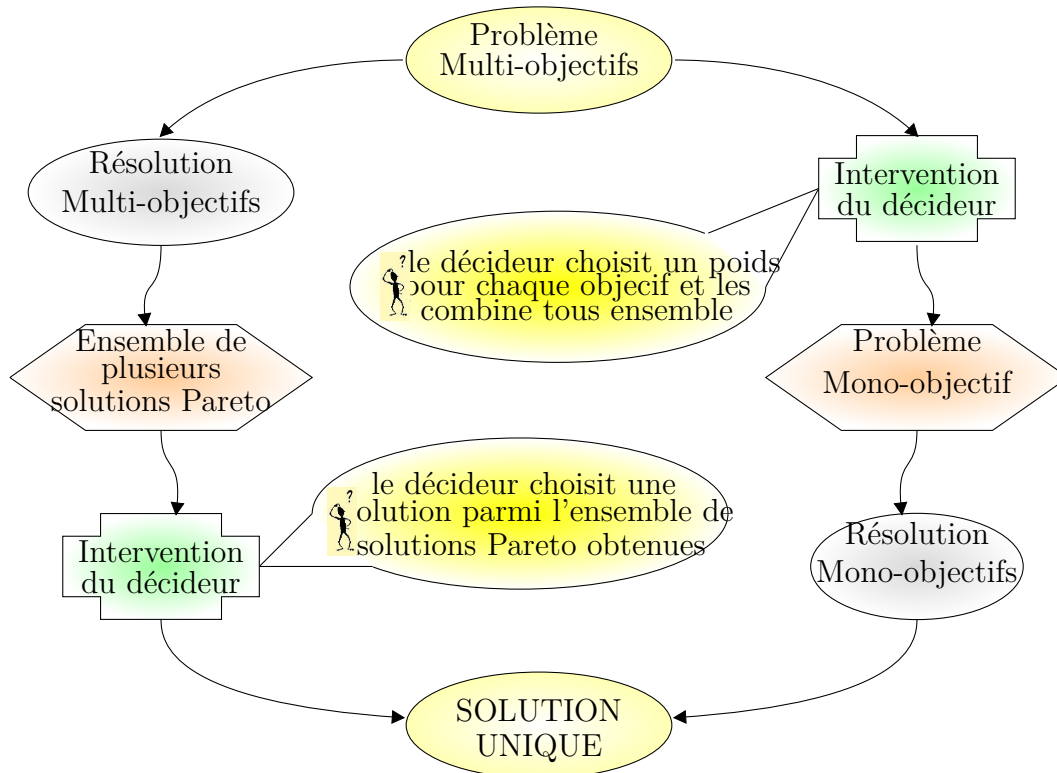


FIGURE 2.2 – Schéma de résolution

Les problèmes multiobjectif ont en général un ensemble de solutions optimales dont les valeurs des fonctions sont en fait les meilleurs compromis possibles dans l'espace des fonctions objectif. Il faut donc utiliser une autre définition de la "meilleure solution", afin de déterminer exactement quelle solution peut être considéré meilleure par rapport à une autre. Le concept de "l'optimalité de Pareto" (Pareto, 1896)[4] est ainsi utilisé pour établir une hiérarchie entre les solutions d'un problème multi-objectif en vue de déterminer si une solution appartient réellement à l'ensemble des meilleurs compromis.

### 2.3.1 La relation de dominance

**Notation :** Si le vecteur  $U$  domine le vecteur  $V$ , on note alors  $U \prec V$  la relation binaire de dominance.

**Définitions :** La relation binaire de dominance notée  $\prec$  :

- n'est pas réflexive, car un vecteur ne se domine pas lui même.
- n'est pas symétrique, car on n'a jamais  $U \prec V$  et  $V \prec U$ .

- est transitive, car  $U \prec V$  et  $V \prec U$  implique  $U \prec W$ .
- n'est pas antisymétrique, en raison de l'existence de solutions Pareto-équivalentes.

Au final, on conclut que la relation de dominance est donc une relation d'ordre partiel strict sur l'espace des critères réalisables [5].

**Remarque 2.3.1.** Notons que  $U \not\prec V$  n'implique pas  $V \prec U$ .

## 2.3.2 Optimalité de Pareto

Les problèmes rencontrés dans la vie réelle sont souvent décrits à l'aide de plusieurs critères ou objectifs, c'est l'optimisation multiobjectifs qui consiste à optimiser simultanément plusieurs fonctions[6],[7]. Cependant, ces fonctions objectifs ont la particularité d'être souvent conflictuelles et contradictoires entre elles, ce qui explique que dans la plupart du temps on trouve une multitude de solutions en résolvant un problème d'optimisation multiobjectifs. La solution n'est pas unique contrairement au problème mono-objectif, mais un ensemble de solutions connu comme l'ensemble des solutions "Pareto optimales". En effet, il définit la formule suivante :

"Dans un problème multiobjectifs, il existe un équilibre tel que l'on ne peut pas améliorer un objectif sans détériorer au moins un des autres objectifs".

Cet équilibre est appelé optimum de Pareto, point non dominé ou encore solution efficace. La définition de l'ensemble de Pareto optimalité provient directement de la notion de dominance d'après cette équivalence :

Une solution  $x^*$  est Pareto optimale si et seulement s'il n'existe aucun point  $x \in S$  telle que :

$$c_i x \leq c_i x^*, \forall i = \overline{1, k}, \text{ avec } c \in R$$

**Et**

$$c_i x < c_i x^*, \text{ pour au moins un } i = \overline{1, k}, \text{ avec } c \in R$$

### 2.3.2.1 Efficacité forte

Une solution  $x^*$  est fortement efficace si et seulement s'il n'existe pas un vecteur de décision  $x$  tel que :  $c^i x < c^i x^*, \forall i = \overline{1, k}$

Autrement dit, une solution est fortement efficace si son vecteur objectif est fortement dominé.

### 2.3.2.2 Efficacité faible

Une solution  $x^*$  est faiblement efficace si et seulement s'il n'existe pas un vecteur de décision  $x$  tel que  $c^i x \leq c^i x^*, \forall i = \overline{1, k}$ .



Autrement dit, une solution est faiblement efficace si son vecteur objectif est faiblement dominé.

**Remarque 2.3.2.** Toute solution efficace est faiblement efficace et l'inverse est faux.

### 2.3.3 Point particuliers

#### Point idéal

Le point idéal est un vecteur de  $R^k$  dont les coordonnées sont les valeurs minimales de chacune des fonctions objectifs pris séparément. D'une autre manière, c'est le vecteur qui minimise chacune des fonctions objectifs  $f_i$  pris séparément.

$$Z^* = (z_1^*, z_2^*, \dots, z_k^*)$$

avec

$$Z_i^* = \min_{x \in S} (f_i(x)), \forall i = \overline{1, k}$$

#### Remarque 2.2.1

- Le point idéal n'appartient pas à l'espace des critères réalisables car les objectifs sont contradictoires.
- Le point idéal est appelé aussi vecteur des valeurs les plus préférées de chaque objectif  $f_i$ , ainsi le point idéal peut être utilisé comme point de référence.
- Le point de référence est un vecteur qui définit les valeurs souhaitables (buts) qu'il faut atteindre par chaque objectif  $f_i$ .

#### Matrice des gains

Soit  $\bar{x}_i$  la solution optimale en minimisant le critère  $f_i$ . La matrice carrée de dimension  $(k \times k)$  formée par :

$$z_i^* = \min_{x \in S} (f_i(x)) = f_i(\bar{x}_i), \forall i = \overline{1, k}$$

Et

$$z_{ij} = f_i(\bar{x}_j), \forall i = \overline{1, k}, \forall j = \overline{1, k} \text{ et } i \neq j.$$

est appelée matrice des gains ("payoff matrix").

- Les éléments de la diagonale principale de cette matrice représentent en fait le point idéal.

$$\begin{pmatrix} Z_{11} = Z_1(\bar{x}_1) & Z_{12} & \cdots & Z_{1k} \\ Z_{21} & Z_{22} = Z_2(\bar{x}_2) & \cdots & Z_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ Z_{1k} & Z_{k2} & \cdots & Z_{kk} = Z_k(\bar{x}_k) \end{pmatrix}$$

avec :

$\bar{x}_1$  est la solution optimale de  $z_1^* = \min_{x \in S} Z_1(x)$ .

$\bar{x}_2$  est la solution optimale de  $z_2^* = \min_{x \in S} Z_2(x)$ .

$\vdots$

$\bar{x}_k$  est la solution optimale de  $z_k^* = \min_{x \in S} Z_k(x)$ .

**Remarque 2.3.3.** Si pour un objectif  $Z_i$ , le problème admet plusieurs solutions optimales, la matrice des gains n'est donc pas unique. En effet, la colonne  $j$  de la matrice dépendra de la solution choisie  $x_i$ .

#### Point Nadir

le vecteur nadir  $z^{nad}$  correspond aux bornes supérieures de chaque objectifs sur la surface de Pareto et non pas dans tout l'espace réalisable  $S$ .

$$z^{nad} = (z_1^{nad}, z_2^{nad}, \dots, z_k^{nad})$$

avec

$$z_i^{nad} = \max_{x \in X_{Pareto}} (Z_i(x)), \forall i = \overline{1, k}$$

#### Remarque 2.3.4.

- On peut définir le point nadir de la matrice des gains, où  $z^{nad}$  se trouve sur le maximum de la ligne de la matrice des gains pour un problème à minimiser.
- Le point nadir est le vecteur des valeurs les plus mauvaises de chaque critères  $Z_i$ , il est utilisé comme point de réservation.
- Le point nadir sert à restreindre l'espace de recherche, il est utilisé dans certaines méthodes d'optimisation interactives.
- A titre d'exemple, on peut utiliser le point idéal et le point nadir pour la normalisation des valeurs des objectifs par :

$$z_i^{norm} = \frac{Z_i - z_i^*}{z_i^{nad} - z_i^*}$$

Une visualisation de l'ensemble des différentes définitions est illustrée sur la Figure suivante :

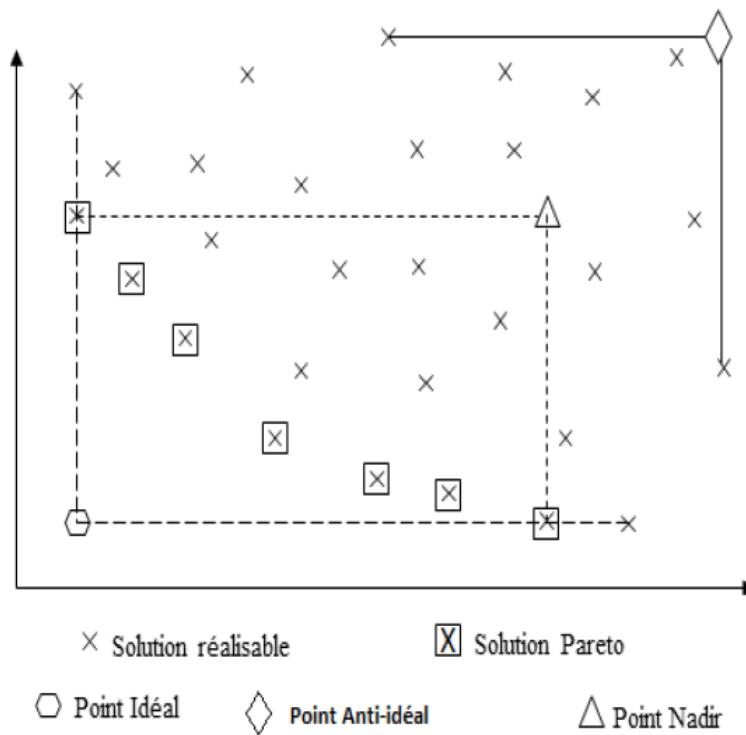


FIGURE 2.3 – Illustration des différentes définitions.

## 2.4 Structure du front de Pareto

L'objectif est donc de fournir aux décideurs un ensemble de solution de Pareto, afin qu'ils puissent ensuite choisir les solutions qui les intéressent le plus.

### Représentation du front Pareto

Toutes les représentations de surface de compromis ou front de Pareto ne sont pas équivalentes. En effet, la représentation idéale du front Pareto devra être constituée de points "solution" de notre problème répartis de manière uniforme sur la surface de compromis (voir figure)

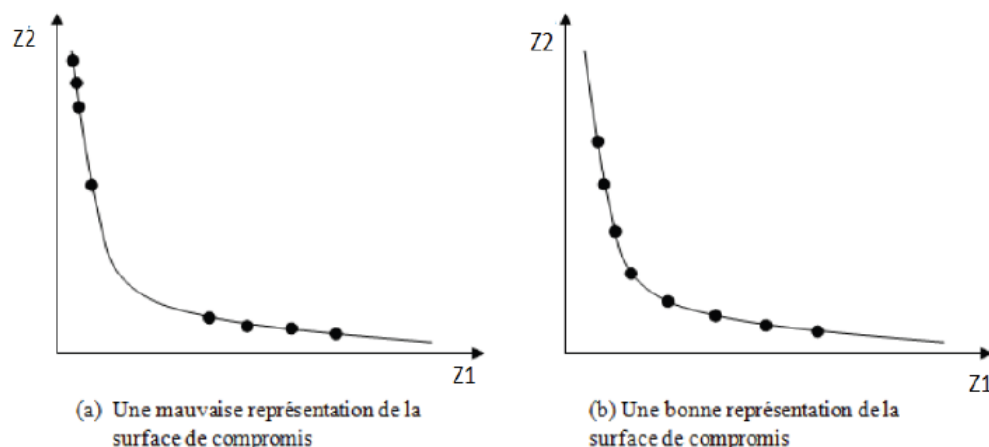


FIGURE 2.4 – La représentation de la surface de compromis

## 2.5 Classification des méthodes de résolution

L'optimisation multiobjectifs consiste à définir des méthodes efficaces de résolution où tous les objectifs sont pris en compte, et d'après la classification des méthodes de résolution multiobjectifs on constate qu'il n'existe pas de méthodes de résolution efficaces et adaptées à tous les types de problèmes.

### 2.5.1 Méthode $\varepsilon$ contrainte [8]

Cette méthode permet de transformer le problème d'optimisation multiobjectif en un problème mono objectif. La méthode consiste à convertir  $(m - 1)$  des  $m$  objectifs du problème en contraintes et d'optimiser séparément l'objectif restant [8]. La démarche est la suivante :

- Nous choisissons un critère à optimiser prioritairement.
- Nous transformons le problème conservant l'objectif prioritaire et nous transformons les autres objectifs en des contraintes d'inégalité Le problème peut être reformulé de la manière suivante :

$$(P_\varepsilon) \begin{cases} \mathbf{maximiser} & Z_i(x) \\ \text{s.c} & \\ Z_j(x) \leq \varepsilon_j, \forall j \neq i, j = \overline{1, k} & \\ x \in S & \end{cases}$$

L'approche par  $\varepsilon$ -contrainte doit aussi être appliquée plusieurs fois en faisant varier le vecteur  $\varepsilon$  pour trouver un ensemble de points Pareto optimaux. Cette approche a l'avantage

par rapport à la précédente de ne pas être trompée par les problèmes non convexes[9].

**théorème 2.5.1.** [10] *Pour tout vecteur  $\varepsilon = (\varepsilon_1; \varepsilon_2; \dots, \varepsilon_k)$  tel que le problème  $(P_\varepsilon)$  a une solution optimale alors cette solution est Pareto optimale pour le problème  $(PMO)$ .*

### Exemple illustratif

Soit le problème multiobjectifs suivant :

$$(P) \begin{cases} \mathbf{max} & Z_1(x) = -3x_1 + x_2 \\ \mathbf{max} & Z_2(x) = x_1 + 2x_2 \\ \mathbf{max} & Z_3(x) = 2x_1 + x_2 \\ & 3x_1 - x_2 \leq 6 \\ & x_2 \leq 2 \\ & x = (x_1, x_2) \in R_+^2 \end{cases}$$

Résoudre le problème en considérant le critère 3 comme prioritaire avec  $\varepsilon = (\varepsilon_1, \varepsilon_2)^t = (1, 3)^t$

$$(P_\varepsilon) \begin{cases} \mathbf{max} & Z(x) = 2x_1 + x_2 \\ & 3x_1 - x_2 \leq 6 \\ & x_2 \leq 2 \\ & -3x_1 + x_2 \leq 1 \\ & x_1 + 2x_2 \leq 3 \\ & x \in R_+^2 \end{cases}$$

La résolution de  $(P_\varepsilon)$  par la méthode du simplexe nous donne la solution optimale  $x^* = (x_1, x_2)^t = (0, 0)^t$

Donc, cette solution est Pareto optimale pour le problème initial d'après le théorème (2.5.1).

### 2.5.2 Méthode dite "but programmé ou Goal programming"

Cette méthode, qui relève d'une théorie très avancée dans le domaine des problèmes multiobjectifs, a été initialement conçue par (Charnes et Cooper [11]) cette méthode dans laquelle le décideur fixe un but  $T_i$ ,  $i = \overline{1, k}$  à atteindre pour chaque objectif  $Z_i$ . Ces valeurs sont ensuite ajoutées au problème comme des contraintes supplémentaires. La nouvelle fonction objectif est modifiée de façon à minimiser la somme des écarts entre les résultats et les buts à atteindre.

$$\min_{x \in S} \sum_{i=1}^k (Z_i(x) - T_i)$$

**Remarque 2.5.1.**

1. Cette méthode a l'avantage de fournir un résultat même si un mauvais choix initial a conduit le décideur à donner un ou plusieurs buts non réalisables.
2. La méthode est facile à mettre en oeuvre mais la définition des buts à atteindre est une question délicate.

### 2.5.3 Méthode des sommes pondérées [30]

Cette méthode de résolution d'un problème de programmation multiobjectifs est la plus évidente et, probablement, la plus couramment utilisée en pratique. Elle consiste à transformer un problème multiobjectifs en un problème qui combine les différentes fonctions objectif du problème en une seule fonction  $Z$  de façon linéaire, l'idée de cette méthode est d'associer une pondération  $\omega_k$  à chaque objectif  $Z^k$  et de résoudre le problème  $P_\omega$  suivant :

$$(P_\omega) \begin{cases} \mathbf{max} & Z(x) = \sum_{k=1}^p \omega_k Z_k(x) \\ s.c & \\ x \in S & \end{cases}$$

Où  $\omega$  est un vecteur de :

$$\Lambda = \left\{ \omega \in \mathbb{R}^p, \sum_{k=1}^p \omega_k = 1, \omega_k \geq 0, k = 1, 2, \dots, p \right\}$$

où les poids  $\omega_k$  sont compris dans l'intervalle  $[0, 1]$ . Différents poids fournissent différentes solutions. La solution de ce problème est une solution Pareto optimale. Il a été montré que pour toute solution supportée  $x^*$  il existe un vecteur  $\omega$  telle que  $x^*$  soit une solution de  $P_\omega$ . Une solution supportée étant une solution appartenant à fermeture convexe de l'ensemble des solutions. En revanche, les solutions non supportées ne peuvent être trouvées par cette méthode. Ainsi, pour les problèmes pour lesquels le front Pareto est convexe il sera possible de trouver l'ensemble du front Pareto avec cette méthode.

Graphiquement, la solution optimale donnée par la méthode de la somme pondérée est le point tangent d'un hyperplan d'équation  $\sum_{k=1}^p \omega_k Z_k(x) = c$  avec le front de Pareto du problème considéré, tel que  $c$  est minimum et  $x$  est une action. (voir la figure)

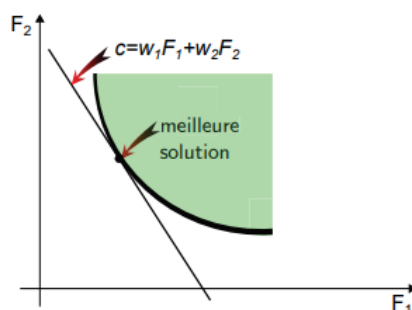


FIGURE 2.5 – Représentation graphique de la méthode de la somme pondérée

## 2.6 Approche de résolution

### 2.6.1 Approche Scalaire

#### Méthode d'agrégation des objectifs [12]

Cette méthode est la plus largement utilisée en pratique et la plus efficace du point de vu informatique car elle est très facile à implémenter. D'ailleurs, elle est l'une des premières méthodes utilisées pour résoudre les problèmes multiobjectifs [12].

Elle consiste à ramener le problème multiobjectifs en un problème mono-objectifs. Elle somme tous les objectifs en affectant à chacun d'entre eux un coefficient de poids. Le décideur choisit les poids en fonction de l'importance relative des objectifs. Le problème multiobjectifs (*PMO*) devient sous la forme suivante :

$$(P_\lambda) \begin{cases} \max & \sum_{i=1}^k \lambda_i Z_i(x) \text{ combinaison linéaire des } Z_i \\ x \in S \end{cases}$$

Où :

$$\lambda_i \geq 0, \forall i = \overline{1, k} \text{ et } \sum_{i=1}^k \lambda_i = 1$$

La résolution du problème multi-objectifs (*PMO*) consiste à résoudre le problème paramétrique ( $P_\lambda$ ). La solution obtenue dépend fortement des coefficients choisis pour les poids. Cette méthode a été développée pour la généralisation des solutions non dominées. Elle est très efficace et applicable pour produire une solution non dominée initiale et peut être employée comme solution initiale pour d'autre technique.

**théorème 2.6.1.** [31]

*Toute solution optimale du problème  $(P_\lambda)$  est une solution Pareto optimale pour le problème multi-objectifs (PMO) si tous les poids sont strictement positifs.*

**théorème 2.6.2.** [31]

*Si le problème multiobjectifs (PMO) est convexe et si  $x^*$  est une solution Pareto optimale alors :*

$\exists \lambda = (\lambda_1, \lambda_2, \dots, \lambda_k)$ , tel que  $x^*$  est une solution optimale pour  $(P_\lambda)$ .

*Cette méthode est très simple à mettre en oeuvre et elle est d'une grande efficacité, sa difficultés est la détermination du poids de chaque critère par le décideur.*

### 2.6.2 Approche Pareto

Les approches Pareto utilisent la notion de dominance pour comparer les solutions et leur affecter un score ou sélectionner des solutions [13].

Ces approches ont connu un important développement en conjonction avec les algorithmes évolutionnistes à population à partir de la seconde moitié des années 90. Elles sont devenues la principale approche employée pour résoudre les problèmes multiobjectifs du fait de leur capacité à trouver un ensemble potentiellement efficace via la recherche menée sur une population de solutions. De manière générale, les algorithmes évolutionnistes multiobjectifs affectent un score à une solution selon qu'elle est dominée ou non par d'autres solutions de la population courante et éventuellement si elle domine d'autres solutions.

## 2.7 Conclusion

Dans ce chapitre, nous Nous sommes intéressées principalement au problème d'optimisation multiobjectifs. En premier lieu, nous avons présenté le concept de base du problème multiobjectifs linéaire. Ensuite nous avons introduit les principales définition et notions nécessaires à la présentation des problèmes multiobjectifs, parmi ces définition nous citons le concept de dominance. En dernier lieu, nous avons présenté quelques méthodes de résolution. ainsi que la caractérisation des approches de résolution.



# Programmation Linéaire Fractionnaire MultiObjectifs en Nombres Entiers

## 3.1 Introduction

La plupart des problèmes rencontrés en pratique sont de nature multiobjectifs. En effet, un problème d'optimisation multiobjectifs fractionnaire consiste à optimiser simultanément plusieurs objectifs souvent contradictoires et parfois complémentaires. Contrairement à l'optimisation mono-objectif, la résolution d'un problème multiobjectifs nous donne une multitude de solutions, 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. Si les fonctions objectifs sont linéaires, que l'ensemble des solutions réalisables est défini par des contraintes linéaires, et si les variables  $x$  prennent des valeurs entières, alors on parle d'un programme linéaire fractionnaire multiobjectif en Nombres Entiers.

A travers ce chapitre, qui a pour objectif principal la présentation du contexte de la programmation linéaire multiobjectifs fractionnaire en nombres entiers (*MOLFP*), nous présentons les notions de base et les définitions nécessaires, ainsi que les principaux résultats liés à la théorie de la programmation linéaire multiobjectifs en nombres entiers[15]. Nous introduisons aussi quelques méthodes de résolution pour ce type de problème et nous donnons une description détaillée de ces méthodes accompagnées chacune par un exemple illustratif.

## 3.2 Formulation du problème

Le problème de la programmation linéaire fractionnaire multiobjectifs (MOLFP :Multiple Objective Linear Fractional Programming problem) est l'un des modèles les plus populaire utilisés dans la prise de décision à critères multiples. La formule générale de ce problème peut être donnée comme suit :

$$(MOLIFP) \left\{ \begin{array}{l} \max Z_1(x) = \frac{c^1x + \alpha^1}{d^1x + \beta^1} \\ \max Z_2(x) = \frac{c^2x + \alpha^2}{d^2x + \beta^2} \\ \max Z_3(x) = \frac{c^3x + \alpha^3}{d^3x + \beta^3} \\ \vdots \\ \max Z_p(x) = \frac{c^px + \alpha^p}{d^px + \beta^p} \\ s.c \\ Ax \leq b, \quad x > 0 \end{array} \right.$$

où le nombre d'objectifs est  $p \geq 2, A \in R^{m \times n}, b \in R^m$ , et pour chaque fonction objectif  $Z^k$ ,  $k = \overline{1, p}$ , on a  $c^k = (c_1^k, c_2^k, \dots, c_n^k) \in R^{1 \times n}$ ,  $d^k = (d_1^k, d_2^k, \dots, d_n^k) \in R^{1 \times n}$ .

On suppose que la région admissible non vide  $S = \{x \in R^n / Ax \leq b, x \geq 0\}$  est borné, et la fonction  $d^kx + \beta^k > 0$  est positif sur  $S$  pour chaque  $k = \overline{1, p}$ .

Dans de nombreuses situation réelles modélisables par la programmation mathématique, les variables intervenant dans la modélisation sont soumises à être totalement en nombre entiers, on parle de problème de programmation multi-objectifs linéaire fractionnaire en nombre entiers (MOILFP : Multiple Objective integer Linear Fractional Programming problem)

$$(MOLIFP) \left\{ \begin{array}{l} \mathbf{max} \quad Z_1(x) = \frac{c^1x + \alpha^1}{d^1x + \beta^1} \\ \mathbf{max} \quad Z_2(x) = \frac{c^2x + \alpha^2}{d^2x + \beta^2} \\ \mathbf{max} \quad Z_3(x) = \frac{c^3x + \alpha^3}{d^3x + \beta^3} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{max} \quad Z_p(x) = \frac{c^px + \alpha^p}{d^px + \beta^p} \\ s.c \\ x \in D \end{array} \right.$$

où  $D = S \cap Z$ ,  $Z$  est l'ensemble des entiers relatifs.

Comme pour les problème de programmation linéaire multiobjectifs, la résolution des problèmes de programmation linéaire fractionnaire multiobjectifs est de déterminer toutes les solution efficaces.

**théorème 3.2.1.** *Soit le problème unicritère linéaire suivant[16] :*

$$(P_\lambda) \begin{cases} \mathbf{max} & \sum_{k=1}^p \lambda_k Z^k(x) \\ \text{s.c} & \\ x \in S, & \text{et } \lambda_k \in \Lambda, \quad \forall k = \overline{1, p} \end{cases}$$

La solution  $x^*$  est efficace pour le problème  $(P_R)$  si et seulement si  $x^*$  est une solution optimale du problème paramétrique  $(P_\lambda)(\Lambda = \{\lambda_k : \sum_{k=1}^k \lambda_k = 1 \text{ et } 0 \leq \lambda_k \leq 1 \quad \lambda_k > 0 \quad \forall k = \overline{1, p}\})$

D'après ce théorème, l'ensemble efficient du problème  $(P_R)$  sans les contraintes d'intégrité est bien caractérisé par les solutions du problème paramétrique  $(P_\lambda), \lambda \in \Lambda$ . Ces solutions se trouvent sur la frontière de  $S$ , elles sont appelées solution supportées.

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

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

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

### 3.2.1 Complexité du problème

La complexité d'un algorithme est le nombre d'opérations élémentaires (affectations, comparaisons, opérations arithmétiques) effectuées par cet algorithme. Ce nombre s'exprime en fonction de la taille  $n$  des données.

On s'intéresse au coût exact quand c'est possible, mais également au coût moyen (que se passe-t-il si en moyenne sur toutes les exécutions du programme sur des données de taille  $n$ ), au cas le plus favorable, ou bien au pire cas. On dit que la complexité de l'algorithme est  $O(f(n))$  où  $f$  est d'habitude une combinaison de polynômes, logarithmes ou exponentielles. Ceci reprend la notation mathématique classique, et signifie que le nombre d'opérations effectuées est borné par  $cf(n)$ , où  $c$  est une constante, lorsque  $n$  tend vers l'infini.

Les algorithmes usuels peuvent être classés en un certain nombre de grandes classes de complexité.

- Les algorithmes sous-linéaires, dont la complexité est en général en  $o(\log n)$ . C'est le cas de la recherche d'un élément dans un ensemble ordonné fini de cardinal  $n$ .
- Les algorithmes linéaires en complexité  $o(n)$  ou en  $o(n \log n)$  sont considérés comme rapides, comme l'évaluation de la valeur d'une expression composée de  $n$  symboles ou les algorithmes optimaux de tri .
- Plus lents sont les algorithmes de complexité située entre  $o(n^2)$  et  $o(n^3)$ , c'est le cas de la multiplication des matrices et du parcours dans les graphes.
- Au delà, les algorithmes polynômiaux en  $o(n^k)$  pour  $k > 3$  sont considérés comme lents, sans parler des algorithmes exponentiels (dont la complexité, est supérieure à tout polynôme en  $n$ ), que l'on s'accorde à dire inutilisables dès que la taille des données est supérieure à quelques dizaines d'unités.

### 3.2.2 Solutions supportées et non supportées

Les problèmes de programmation linéaire multiobjectifs en nombres entiers ont la particularité d'avoir le domaine de décision non convexe, ce qu'il en résulte que sa résolution est complexe et difficile à traiter. En effet, les méthodes classiques qui permettent de trouver les solutions efficaces ne sont plus valables pour les problèmes (*MOILP*) car certaines solutions ne peuvent pas être obtenues.

De ce principe, on répartie l'ensemble des solutions efficaces en deux sous ensembles : les solutions supportées et les solutions non supportées.

Solutions Supportées notées SES : elles appartiennent aux portions convexes du front de Pareto (elles se situent sur l'enveloppe convexe de la frontière de Pareto). On obtient ces solutions en optimisant une agrégation linéaire des objectifs, c'est à dire transformer un problème multiobjectifs en un problème mono-objectifs.

Solutions Non Supportées notées SENS : elles se trouvent dans les parties non convexe.

**Remarque 3.2.1.** L'ensemble des solutions du problème (*MOILP*) est égal à l'union des solutions supportées SES et des solutions non supportées SENS.

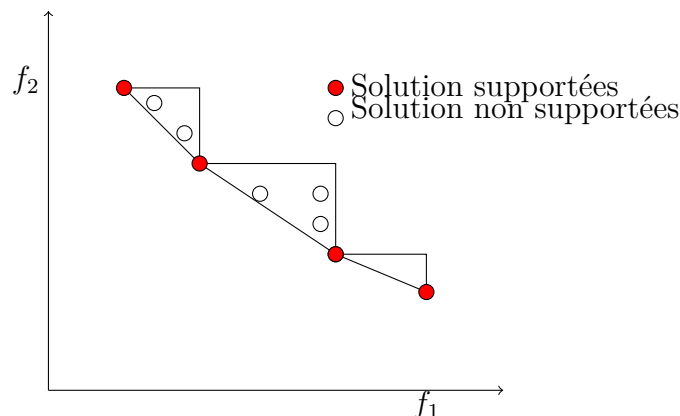


FIGURE 3.1 – Solutions supportées et non supportées

### 3.3 Méthodes de résolution d'un programme linéaire en nombres entiers

#### 3.3.1 La méthode de gomory [27]

C'est une méthode qui a été développée par Ralph E : Gomory[27] en 1958 a  
fin de résoudre les problèmes (*PLNE*). L'idée principale de cette méthode est d'ajouter des contraintes linéaires qui n'excluent aucun point entier réalisable, une par une jusqu'à ce que la solution optimale de la relaxation soit entière.

Dans une première étape, on utilise l'algorithme du simplexe pour résoudre le programme (*PL*) a

fin de chercher une solution de base optimale, si elle existe, on choisit une variable de base non entière et on génère une inéquation sur la contrainte associée à cette variable a

fin de couper la région de faisabilité courante, étant donnée une base optimale B du problème (*PL*).

Le tableau optimale correspondant est donné par :

$x_B$	$\hat{A} = B^{-1}A$	$\hat{b} = B^{-1}b$
$-Z$	$\hat{c} = c - \Pi A$	$Z - c_B B^{-1}b$

TABLE 3.1 – simplexe optimal associé à la base B.

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 réduit relatif à la base B avec  $\hat{c}_B = 0$

Si la solution optimale de (PL) est entière, donc elle est solution optimale du problème (ILP). Sinon, parmi les variables de base, choisissons  $x_i, i \in B$  dont la valeur est fractionnaire. La  $i^{\text{ème}}$  ligne du tableau optimal est donnée par :

$$x_i + \sum_{j \in N} \hat{a}_{ij} x_j = \hat{b}_i \dots\dots\dots(1)$$

Où :

$\hat{a}_{ij}$  : est un élément de la matrice optimale des contraintes  $\hat{A}$ .

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

$B$  : est l'ensemble des indices de base.

Etant donné un nombre réel  $\alpha$ , on désigne par  $\lfloor \alpha \rfloor$  le plus grand entier inférieur ou égal à  $\alpha$ .

$\langle \alpha \rangle = \alpha - \lfloor \alpha \rfloor$  est appelée la partie fractionnaire de  $\alpha$  et  $\lfloor \alpha \rfloor$  sa partie entière.

Puisque toutes les variables sont positives ou nulles, on a :

$$\sum_{j \in N} \lfloor \hat{\alpha}_{ij} \rfloor x_j \leq \sum_{j \in N} \hat{\alpha}_{ij} x_j$$

de (1) on a :

$$x_i + \sum_{j \in N} \lfloor \hat{\alpha}_{ij} \rfloor x_j \leq \hat{b}_i.$$

Comme le membre gauche est entier dans cette inégalité, la partie droite (second membre) peut être remplacée par sa partie entière.

$$x_i + \sum_{j \in N} \lfloor \hat{\alpha}_{ij} \rfloor x_j \leq \lfloor \hat{b}_i \rfloor \dots\dots(2)$$

En soustrayant (1) de (2) on obtient :

$$\sum_{j \in N} \langle \hat{\alpha}_{ij} \rangle x_j \geq \langle \hat{b}_i \rangle$$

En ajoutant une variable d'écart  $x_s$  à cette dernière inéquation, on obtient la coupe de Gomory définie par :

$$-\sum_{j \in N} \langle \hat{\alpha}_{ij} \rangle x_j + x_s = -\langle \hat{b}_i \rangle$$

On introduit cette contrainte dans le tableau optimal du simplexe et en utilisant le dual du simplexe, on résout le nouveau problème formé qui nous permet d'obtenir soit une solution optimale entière ou bien une impossibilité de continuation de l'itération.

### 3.3.2 Méthode de Branch et Bound

La méthode de Branch et Bound appelée également méthode par séparation et évaluation, est la méthode la plus utilisée pour résoudre d'une façon exacte des problèmes (PLNE). En effet, cette méthode est basée principalement sur deux procédures. En premier lieu la séparation ou appelée encore le branchement, qui consiste à subdiviser l'ensemble des solutions réalisables en un sous-ensemble. En deuxième lieu l'évaluation qui nous permet de déterminer les solutions qui n'appartiennent pas à l'ensemble des solutions réalisables. Cette méthode est très efficace, en effet son énumération des solutions d'une manière intelligente permet d'éliminer des solutions partielles qui ne mènent pas à la solution recherchée.

#### Principe de la méthode

- On construit un arbre de recherche dont le problème initial est la racine.
- On divise le problème en sous-problèmes : l'optimum peut appartenir à l'un de ces sous-problèmes.

- On élimine tout problème infaisable.
- Si possible, on calcule la solution du sous-problème.
- Sinon on calcule une borne inférieure, si elle est supérieure à la meilleure solution déjà obtenue, on élimine le sous-problème considéré.
- Dans le cas restant, on subdivise à nouveau le domaine.

## Etape de résolution

On présente les étapes nécessaires de résolution comme suit : (on donne ici la démarche pour un problème de maximisation).

### 1/ Résolution de (PL) avec l'algorithme du simplexe

Résoudre le (PL) correspondant (sans contraintes de variables entières), la valeur de  $Z$  du tableau optimal donne une borne supérieure au (PLNE) notée par  $Z_{BS}$ .

Arrondir cette solution aux valeurs entières les plus proches et réalisables et calculer à nouveau la valeur de la fonction objectif du (PLNE). Appelons cette valeur  $Z_{BI}$ .

ainsi  $Z_{BI} \leq Z_{opt}$  du (PLNE)  $\leq Z_{BS}$

### 2/ Séparation

- Construire deux sous-ensembles à l'aide d'une variable non entière, soit  $x_k$  cette variable, représentons par  $[x_k^*]$  la partie entière de  $x_k$ .
- Pour éliminer la solution non entière  $x_k$ , on crée deux branches (deux sous-problèmes), on obtient une branche avec  $x_k \leq [x_k^*]$  et l'autre avec  $x_k \geq [x_k^*] + 1$ .

Les deux sous-problèmes sont obtenus en ajoutant au programme linéaire (PL) précédent (celui d'où provient la séparation), la contrainte  $x_k \leq [x_k^*]$  pour un des sous-problèmes et la contrainte  $x_k \geq [x_k^*] + 1$  pour l'autre sous-problème.

Toutes les solutions entières réalisables sont maintenant contenues dans l'un ou l'autre des sous-problèmes.

### 3/ Résolution des sous-problèmes et détermination de nouvelles bornes pour Z, s'il y a lieu

- Il s'agit de résoudre chaque sous-problème par l'algorithme du simplexe, notons par  $MBSD$ , la meilleure borne supérieure disponible de  $Z$ , elle correspond à la valeur maximale de la fonction objectif des deux sous-problèmes que nous venons de résoudre.

- Nous révisons également la borne inférieure de  $Z$ , notons cette valeur par  $MBID$ , la meilleure borne inférieure disponible, elle correspond à la valeur maximale de la fonction objectif de toutes les solutions entières obtenues jusqu'à présent.

#### 4/ Critère d'arrêt de l'exploitation d'une branche

La procédure de branchement est achevée lorsque l'une des conditions suivantes est satisfaite :

- (a) Le sous-problème de la branche considérée n'admet pas de solution réalisable.
- (b) La solution entière obtenue pour le sous-problème de la branche est réalisable mais la valeur  $Z$  obtenue est plus petite ou égale à  $MBID$  ( $Z^* \leq MBID$ ), ainsi une branche est terminée aussitôt qu'on obtient une solution entière.
- (c) Le sous-problème de la branche considérée admet une solution optimale non entière mais sa valeur  $z^*$  est inférieure ou égale à celle d'un autre sous-problème à solution entière.

#### 5/ Critère d'optimalité

Toutes les branches sont saturées, donc on termine avec la procédure de séparation. La solution optimale de ( $PLNE$ ) est la solution entière qui donne la meilleure borne inférieure pour  $Z(MBID)$ , à l'optimum on aura  $MBID = MBSD = Z_{opt}$  de ( $PLNE$ ).

#### 6/ Continuer le processus de séparation

Dans le cas où la solution obtenue dans l'un des sous-ensembles de chaque branche est non entière, on poursuit le processus de séparation à partir du sous-ensemble ayant la valeur de  $Z^*$  la plus élevée.

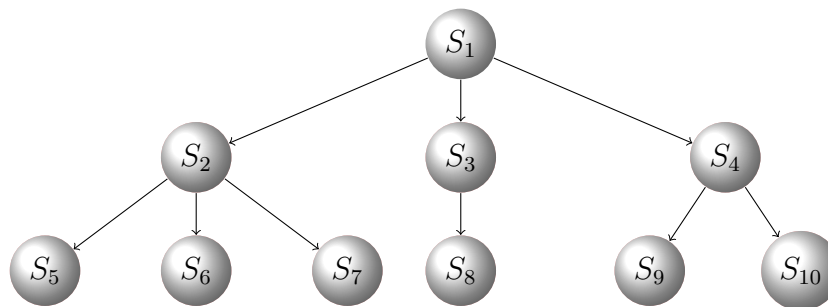


FIGURE 3.2 – Principe de Branch and Bound

### 3.4 Résolution du problème

Le problème de la programmation multi-objectifs linéaire fractionnaire (MOLFP) a été largement étudié par plusieurs auteurs. De nombreuses études et applications ont été



rapportées dans la littérature, nous citons entre autre[17]. Cependant, le problème multi-objectifs linéaire fractionnaire à variables entières (MOILFP) n'a pas reçu autant d'attention que le problème (MOLFP). On ne trouve que très peu de méthodes dédiées au problème (MOILFP), voir par exemple [18].

### 3.4.1 Méthodes de Abbas et Moulaï [19]

La méthode proposée par (Abbas et Moulaï)[19] pour résoudre le problème (MOLIFP) est une généralisation de leur méthode du problème(MOLIFP),détaillée dans [19] Afin de résoudre le problème (MOLIFP) une approche consistant à résoudre une séquence finie de problèmes de programmation fractionnaire linéaire discrète. Considérons le problème de programmation linéaire fractionnaire entier mono-objectif donné sous la forme suivante.

$$\begin{cases} \max & Z^1(x) = \frac{c^1x + \alpha^1}{d^1x + \beta^1} \\ s.c & \\ Ax \leq b, & x > 0 \end{cases}$$

**Remarque 3.4.1.** Notons qu'à la place de  $Z^1$ , on peut de façon similaire considérer une autre fonction objectif  $Z^k$  pour tout  $k = \overline{1, p}$   
La recherche des solutions réalisables entières du problème (MOLIFP) nécessite l'introduction des notations suivantes :

- $S_1 = \{x \in R^{n_1}/A_1x \leq b_1, A_1 \in R^{m_1 \times n_1}, b_1 \in R^{m_1}, x \geq 0\}$ .  $S_1$  est la région tronquée courante de S obtenue par des coupes de Gomory successives.
- $x_1^1 = (x_{1,j}^1)$  est la solution optimale entière donnant  $Z_1^1$  obtenue sur  $S_1$ .
- $B_1^1$  est une base de  $S_1$ .
- $a_{1,j}^1 \in R^{m_1 \times 1}$  sont les vecteurs activités de  $(x_{1,j}^1)$  appropriés à la région tronquée courante  $S_1$ .
- $y_{1,j}^1 = x_{1,kj}^1 = (B_1^1)^{-1}a_{1,j}^1$  où  $y_{1,j}^1 \in R^{m_1 \times 1}, k \in R^p$ .
- $I_1 = \{j/a_{1,j}^1 \in B_1^1\}$ , et  $N_1 = \{j/a_{1,j}^1 \notin B_1^1\}$ .
- $c_j^1$  est la  $j^{\text{ème}}$  composante du vecteur  $c_1$ ,  $d_j^1$  est la  $j^{\text{ème}}$  composante du vecteur  $d^1$ .
- $c_{1,1}^1 = \sum_{k \in I_1} c_k^1 y_{1,kj}^1$  et  $d_{1,1}^1 = \sum_{k \in I_1} d_k^1 y_{1,kj}^1$ .
- $Z^1(x_1^1) = \frac{Z_{1,1}^1}{Z_{1,2}^1}$  Où  $Z_{1,1}^1 = c^1 x_1^1 + \alpha^1$  et  $Z_{1,2}^1 = d^1 x_1^1 + \beta^1$ .
- $\bar{\gamma}_{1,j}^1 = Z_{1,2}^1(c_j^1 - c_{1,j}^1) - Z_1^1(d_j^1 - d_{1,j}^1)$  le coût réduit relatif à la  $j^{\text{ème}}$  composante du vecteur gradient réduit  $\bar{\gamma}_1^1$ .
- $\Gamma_1 = \{j/j \in N_1 \text{ et } \bar{\gamma}_{1,j}^1 = 0\}$

pour  $k \geq 2$

- $S_k = \{x \in R^{n_k} = A_k x \leq b_k, A_k \in R^{m_k \times n_k}, b_k \in R^{m_k}, x \geq 0\}$

- $S_k$  est la région courante tronquée de  $S$  obtenues après l'application de la coupe  $\sum_{j \in N_{k-1}} x_j \geq 1$ .  
Où  $j_{k-1} \in \Gamma_k - 1$  et d'éventuelles successives coupes de Gomory.
- $x_k^1 = x_{1,j}^1$  est la  $k^{\text{ème}}$  solution optimale entière du problème (MOLIFP) obtenue sur  $S_k$  à l'étape  $k$
- $x_k^q = \{x_{k,j}^q\}$  sont les  $(P_{k-1})$  solution entière, lorsqu'elles existes alternatives à  $x_k^1$ .  
Où  $p_k$  est un nombre entier et  $q \in \{2, \dots, p^k\}$
- $B_k^1$  est une base de de  $S_k$ .
- $a_{k,j}^1 \in R^{m_k \times 1}$  sont les vecteurs activités de  $x_{k,j}^1$  sous  $S_k$ .
- $y_{k,j}^1 = (y_{k,i,j}^1) = (B_k^1)^{-1} a_{k,j}^1$  Où  $y_{k,j}^1 \in R^{m_k \times 1}$ .
- $I_k = \{j : a_{k,j}^1 \in B_k^1\}$  indices de base de  $B_k^1$ .
- $N_k = \{j : a_{k,j}^1 \notin B_k^1\}$  indices hors base de  $B_k^1$ .
- $Z_{k,j}^1 = \sum_{i \in I_k} c_i^1 y_{k,i,j}^1$
- $Z_i(x_k^1) = c^i x_k^1$
- $\Gamma_k = \{j : j \in N_k \text{ et } Z_{k,j}^1 - C_k^1 = 0\}$  Où  $Z_{k,j}^1 = c_{B_k^1}^1 y_{k,j}^1$ ,  $c_j^1$  est la  $j^{\text{ème}}$  correspondante du vecteur  $c^1$  et  $c_{B_k^1}^1$  est le vecteur coût des variables de base associées à  $B_k^1$  dans le vecteur  $c^1$ .  
Ainsi,  $\Gamma_k$  est l'ensemble correspondant à la solution  $x_k^1$ .

**théorème 3.4.1.** *Le point  $x_k^1$  de  $S$  est une solution optimale du problème fractionnaire (MOLIFP) si et seulement si le vecteur gradient réduit  $\bar{\gamma}$  est tel que  $\bar{\gamma}_{k,j}^1 \leq 0$  pour tout indice  $j \in N_k$ .*

**Corollaire 3.1** La solution  $x_k^1$  du problème (MOLIFP) est unique si et seulement si le vecteur gradient réduit  $\bar{\gamma}$  est tel que  $\bar{\gamma}_{k,j}^1 < 0$  pour tout indice  $j \in N_k$ .

**Remarque 3.4.2.** Il peut arriver que la solution optimale  $x^0$  du problème (MOLIFP) ne soit pas unique, dans ce cas, il existe une autre solution réalisable  $x^1 \neq x^0$  avec  $Z^1(x^1) = Z^1(x^0)$ . On dit alors que  $x^1$  est une solution optimale alternative de  $x^0$ .

## Algorithme de la Méthode

Pour résoudre un problème fractionnaire linéaire en nombres entiers à objectifs multiples (MOLIFP), une procédure basée sur une technique de coupe plane est présentée dans les étapes suivantes :

**Etape(1) :** Résoudre le problème (MOLIFP) par n'importe quelle méthode directe de la programmation fractionnaire discrète. Soit  $x_1^1$  sa solution optimale entière sur  $S_1$ , construire l'ensemble  $\Gamma_1$ .

**Etape (2) :** 1/ - Si  $\bar{\gamma}_{1,j}^1 < 0$  pour tout indice  $j \in N_1$ ,  $x_1^1$  est l'unique solution optimale sur

CHAPITRE 3. PROGRAMMATION LINÉAIRE FRACTIONNAIRE  
MULTIOBJECTIFS EN NOMBRES ENTIERS

---

$S_1$ .

Enregistrer le premier vecteur efficace (non dominé) par  $(Z^1(x_1^1), \dots, Z^p(x_1^1))$  pour construire l'ensemble des points efficaces  $Eff_0$ . Tronquer le point  $x_1^1$  par la coupe de Dantzig  $\sum_{j \in N_1} x_j \geq 1$ .

Par application de la méthode duale du simplexe relative à la programmation fractionnaire, on obtient une solution réalisable entière  $x_1^2 = (x_{2,j}^1)$  dans la région tronquée  $S_2$ . Rajouter le vecteur correspondant  $(Z^1(x_1^2), \dots, Z^p(x_1^2))$  à  $Eff_0$  s'il n'est pas dominé par l'un des précédents vecteurs critères efficaces (non dominés). Enregistrer l'ensemble  $Eff_1$ .

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

**Etape** ( $k \geq 2$ ) : Choisir un indice  $j_{k-1} \in \Gamma_{k-1}$  et explorer l'arête  $E_{j,k-1}$  pour déterminer d'éventuelles solutions entières réalisables  $x_{k-1}^1$ . Augmenter l'ensemble  $Eff_{k-2}$  par les vecteurs critères non dominés correspondants pour construire  $Eff_{k-1}$ . Tronquer l'arête  $E_{j,k-1}$  par la coupe  $\sum_{j \in N_{k-1}/\{j,k-1\}} x_j \geq 1$ .

**Etape finale** : Le 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.

## Exemple illustratif

Soit le problème de programmation multi-objectifs linéaire fractionnaire en nombres entiers suivant :

$$(P) \begin{cases} \max & Z_1(x) = \frac{x_1 - 4}{-x_2 + 3} \\ \max & Z_2(x) = \frac{-x_1 + 4}{x_2 + 1} \\ \max & Z_3(x) = -x_1 + x_2 \\ \text{s.c} & -x_1 + 4x_2 \leq 0 \\ & x_1 - \frac{1}{2}x_2 \leq 4 \\ & 2x_2 \leq 7 \\ & x_1, x_2 \geq 0, \text{ et entiers} \end{cases}$$

**Etape 1** : Résoudre le problème linéaire fractionnaire unicritère suivant :

$$(P_1) \begin{cases} \max & Z_1(x) = \frac{x_1 - 4}{-x_2 + 3} \\ \text{s.c} & -x_1 + 4x_2 \leq 0 \\ & x_1 - \frac{1}{2}x_2 \leq 4 \\ & x_1, x_2 \geq 0 \text{ et entiers} \end{cases}$$

CHAPITRE 3. PROGRAMMATION LINÉAIRE FRACTIONNAIRE  
MULTIOBJECTIFS EN NOMBRES ENTIERS

---

Les solution entière optimal  $x_1^1 = (4, 0)$  sur le domaine  $S_1 = S$  est obtenue au tableau 1 et tableau 2 suivants.

Tableau 1	$x_B$	$x_2$	$x_5$
$x_3$	4	4	1
$x_4$	0	-1/2	-1
$x_1$	4	0	1
$-p$	0	0	-1
$-q$	-3	-1	0
$\bar{\gamma}_j$	0	0	-3

FIGURE 3.3 – itération 1

Tableau 2	$x_B$	$x_2$	$x_6$
$x_3$	3	4	1
$x_4$	1	-1/2	-1
$x_1$	3	0	1
$x_5$	1	0	-1
$-p$	1	0	-1
$-q$	-3	-1	0
$\bar{\gamma}_j$	-1/3	-1	-3

FIGURE 3.4 – itération 2

On a  $I_1 = \{1, 3, 4\}$ ,  $N_1 = \{2, 5\}$ ,  $\Gamma_1 = \{j \in N_1 / \bar{\gamma}_j^1 = 0\} = \{2\}$ , le premier triplet correspondant à  $x_1^1 = (4, 0)$  est  $(Z_1(4, 0), Z_2(4, 0), Z_3(4, 0)) = (0, 0, -4)$ .

**Etape 2 :** On a  $\Gamma \neq \emptyset$ , prendre l'indice  $j_1 = 2$  et parcourir l'arête  $E_{j_1} = E_2$ , on calcule d'abord le nombre  $\theta_{j_1} = \min\{\frac{3}{4} = 1 \neq 0\}$  ce qui prouve qu'il existe une solution  $x_1^2$  alternative à  $x_1^1$  sur l'arête  $E_{j_1}$  et sa valeur est  $x_1^2 = (4, 1)$  et son triplet correspondant  $(0, 0, -3)$  domine le triplet  $(0, 0, -4)$  d'où  $Eff_0 = \{(0, 0, -3)\}$ .

l'arête  $E_{j_1}$  est tronquée par la coupe  $\sum_{j \in N_1 / \{2\}} x_j \geq 1$  ce qui donne  $x_5 \geq 1$ .

on rajoute cette coupe au tableau 1 et on applique la méthode Dual de Simplexe pour obtenir une nouvelle solution réalisable entière optimal  $x_1^2 = (3, 0)$  donnée au tableau 2. le triplet correspondant  $(\frac{-1}{3}, 1, -3)$  est non dominé d'où  $Eff_1 = \{(0, 0, -3), (\frac{-1}{3}, 1, -3)\}$ .

**Etape 3 :** On a  $N_2 = \{2, 6\}$  et comme  $\Gamma_2 = \emptyset$  donc la solution  $x_1^2 = (3, 0)$  est unique. cette solution est tronquée par la coupe de Dantzig  $\sum_{j \in N_2} x_j \geq 1$  c'est à dire  $x_1 + x_6 \geq 1$ . Après application de la méthode Dual de Simplexe, la nouvelle solution entière optimale  $x_1^3 = (2, 0)$  est obtenue au tableau 3 suivant :

CHAPITRE 3. PROGRAMMATION LINÉAIRE FRACTIONNAIRE  
MULTIOBJECTIFS EN NOMBRES ENTIERS

---

<b>Tableau 3</b>	$x_B$	$x_7$	$x_8$
$x_3$	2	1	-3
$x_4$	2	-1	-1/2
$x_1$	2	1	1
$x_5$	2	1	-1
$x_6$	1	-1	-1
$x_2$	0	0	1
$x_9$	-1	-1	-1
$-p$	2	-1	-1
$-q$	-3	0	1
$\bar{\gamma}_j$	-2/3	-3	-1

FIGURE 3.5 – itération 3

le triplet non dominé  $(\frac{-2}{3}, 2, -2)$  obtenu est rajouté à  $Eff_1$  pour construire l'ensemble  $Eff_2 = \{(0, 0, -3), (\frac{-1}{3}, 1, -3), (\frac{-2}{3}, 2, -2)\}$ .

De manière similaire, les itérations se poursuivent jusqu'à l'étape (6) où on obtient le tableau 6 final suivant :

<b>Tableau 6</b>	$x_B$	$x_{10}$	$x_{11}$
$x_3$	-3	-2	3
$x_4$	7/2	-3/2	1/2
$x_1$	1	2	-1
$x_5$	-1	0	1
$x_6$	2	-2	1
$x_7$	2	-1	0
$x_8$	-1	-1	1
$x_9$	0	-2	1
$x_2$	1	1	-1
$-p$	3	-2	1
$-q$	-2	1	-1
$\bar{\gamma}_j$	-3/2	-1	-1

FIGURE 3.6 – itération 6

Lors de l'application de la méthode Dual du Simplexe, l'opération usuelle de pivot devient impossible indiquant que le domaine tronqué restant ne contient aucun point entier réalisable et que toutes les solution efficaces du problème étudié ont été obtenues. D'où l'ensemble  $Eff$  des points non dominés du problème ( $P_1$ ) est :

$Eff = \{(0, 0, -3); (\frac{-1}{3}, 1, -3); (\frac{-2}{3}, 2, -2); (-1, 3, -1); (\frac{-4}{3}, 4, 0)\}$  ce qui correspond à l'ensemble des solutions efficaces suivant  $Eff = \{(4, 1); (3, 0); (2, 0); (1, 0); (0, 0)\}$ .

**Discussion :**

Dans la première étape, un calcul considérable est nécessaire pour obtenir une première solution entière optimale du problème linéaire fractionnaire en nombre entiers ( $P_1$ ) est la méthode doit passer en revue tous les points entières de l'ensemble des solutions réalisables ce qui alourdit d'avantages la complexité de l'algorithme.

### 3.4.2 Méthode de Chergui et Moulaï [20]

L'approche adoptée dans cette méthode [20] pour générer toutes les solutions efficaces du problème ( $MOIFLP$ ) est basée sur la résolution du programme linéaire fractionnaire (P) suivant, à chaque étape (k) de l'algorithme :

$$(P) \begin{cases} \mathbf{max} & Z^1(x) = \frac{c^1x + \alpha^1}{d^1x + \beta^1} \\ s.c & \\ x \in S_k & \end{cases}$$

où  $S_0 = S$  et sans contraintes d'intégrité des variables.

Soit  $x_k^*$  la solution entière obtenue après résolution du problème (P) on note  $B_k$  l'ensemble des indices des variables de base et  $N_k$  l'ensemble des indices des variables hors base de  $x_k^*$ .

Soit  $\bar{\gamma}_j^k$  la  $j^{\text{ème}}$  composante du vecteur gradient réduit  $\bar{\gamma}^k$  défini pour chaque critère  $k, k \in \{1, \dots, p\}$  par :

$\bar{\gamma}^k = \bar{\beta}^k \bar{c}^k - \bar{\alpha}^k \bar{d}^k$  Où  $\bar{c}^k, \bar{d}^k, \bar{\beta}^k$  et  $\bar{\alpha}^k$  sont les valeur mises à jour de  $c^k, d^k, \beta^k$  et  $\alpha^k$  respectivement.

On définit l'ensemble  $H_k$  en  $x_i^*$  par :

$$H_k = \{j \in N_k, |\exists k \in \{1, \dots, p\} : \bar{\gamma}_j^k > 0\} \cup \{j \in N_k, |\bar{\gamma}_j^k = 0, \forall k \in \{1, \dots, p\}\}$$

Ainsi, l'ensemble  $S_{k+1}$  est défini par  $S_{k+1} = \{x \in S_k | \sum_{j \in H_k} x_j \geq 1\}$ .

## Algorithme de la méthode

L'algorithme de génération de toutes les solutions entères efficaces du programme ( $MOILFP$ ) est présenté dans les étapes suivants :

**Étape(0) : Initialization :** Créer le premier noeud avec le programme  $(P)(Eff = \emptyset Eff$  étant l'ensemble des des solutions efficaces du problème multi-objectif ( $MOILFP$ ))

. **Étape(1) :Étape générale :** Tant qu'il existe un noeud  $k$  non encore sondé qui correspond à la région  $S_k$ , faire :

Résoudre le programme linéaire fractionnaire correspondant  $(P)$  par la méthode duale du simplexe (la méthode,est utilisée juste au début pour la résolution du programme initial  $(p)$  (c'est à dire pour  $S_0 = S$  ).

- Si le programme  $(P)$  n'a pas de solutions réalisables, alors le noeud correspondant est sondé.
- Sinon, soit  $\bar{x}_k$  la solution optimale. Si  $\bar{x}_k$  n'est pas entière, aller à l'étape(a), sinon aller à l'étape (b).

**Étape(1.a) :** Séparation Choisir une coordonnée non entière  $\bar{x}_{k,j}$  de  $\bar{x}_k$  et séparer le noeud  $k$  actuel en deux noueds  $r, r \geq k + 1$ , et  $h = r + 1$  ,Dans le tableau courant du simplexe, la contrainte  $x_j \leq \lfloor \bar{x}_{k,j} \rfloor$  est rajoutée et un nouveau domaine est considéré au noeud  $r$  et de façon similaire, la contrainte  $x_j \geq \lfloor \bar{x}_{k,j} \rfloor + 1$  est rajoutée pour obtenir un autre domaine au noeud  $h = r + 1$  (chaque programme créé doit être résolu en utilisant le même processus jusqu'à ce qu'une solution entière réalisable soit trouvée).

**Étape(1.b) :** Mise à jour de l'ensemble  $Eff$  Si le vecteur  $Z(\bar{x}_k)$  n'est pas dominé par le vecteur  $Z(x)$  pour toute solution  $x \in Eff$  , alors  $Eff = Eff \cup \{\bar{x}_k\}$ .

S'il existe  $x \in Eff$  tel que  $Z(\bar{x}_k)$  domine  $Z(x)$ ,  $Eff = Eff / \{x\} \cup \{\bar{x}_k\}$ .

Déterminer les ensembles  $N_k$  et  $H_k$  ;

- Si  $H_k = \emptyset$ , alors le noeud correspondant est sondé. Aller à l'étape (1) ;
- Sinon, rajouter la coupe  $\sum_{j \in H_k} x_j \geq 1$  au programme  $(P)$ . Aller à l'étape (1).

## 3.5 Conclusion

A travers ce chapitre, nous avons pris connaissance des problèmes de programmation linéaire fractionnaire multi-objectifs en nombres entiers et la formule mathématique de ce type de problème. Ainsi, Nous avons cité quelques méthodes classiques scalarisantes qui transforment le problème fractionnaire multiobjectifs en problème mono-objectif. Puis, nous avons introduit dans ce chapitre les méthodes de résolution d'un problème ( $MOILFP$ ) ont été présentées, dont la dernière servira comme support pour la résolution de notre problème principal.

# Chapitre 4

## Approche de résolution du problème MOIFLP

### 4.1 Introduction

La programmation est un ensemble d'outils et de technique permettant de résoudre des problèmes mathématiques par ordinateurs, elle sert a trouver une solution optimale de n'importe quel type de problème. Le processus de résoudre un problème mathématique exige un grand nombre de calculs donc il est mieux de l'exécuter par machine . Pour cela on a choisit le logiciel LINGO qui est spécifié pour résoudre des problèmes d'optimisations (linéaire,non linéaire,convexe ,non convexe,..etc). Il comporte un éventail de commandes qui peuvent être appelées à tout moment.

### 4.2 Définition du problème

Dans cette partie, nous nous sommes particulièrement intéressés aux problèmes multi-objectifs fractionnaire linéaire en nombre entier. L'approche de résolution proposée consiste à convertir le problème étudié en un problème mono-objectif en utilisant la méthode de la somme pondérée. Dans cette étape, les poids relatifs sont attribués en fonction de l'importance relative de la fonction objectif. Enfin, la solution obtenue est un ensemble non dominé.

#### 4.2.1 Problème fractionnaire multi-objectif

Considérons le problème(MOLIFP) :

$$(MOLIFP) \begin{cases} \mathbf{max} & Z_k(x) = \frac{c^k x + \alpha^k}{d^k x + \beta^k}, \forall k = \overline{1, p} \\ s.c & \\ x \in D & \end{cases}$$

avec D l'ensemble des entiers relatifs



### 4.2.2 Transformation du problème (MOLIFP) vers un problème mono-objectif [30]

En utilisant la méthode de la somme pondérée pour résoudre le problème (MOLIFP). Elle est à la fois la méthode la plus simple et la plus connue dans le domaine de prise de décision multicritère. Comme son nom l'indique, la méthode de la somme pondérée consiste à pondérer les différents critères du problème de décision multicritères avec des nombres réels appelés poids  $w_k$ , qui représentent l'importance de chaque critère dans le processus de décision. Une fois l'importance des différents critères quantifiée, la méthode choisit l'action qui minimise ou maximise la somme pondérée des critères. La méthode de la somme pondérée consiste à maximiser une fonction  $U$  définie comme suit :

$$U(x) = \sum_{k=1}^p \omega_k Z^k. \quad (4.1)$$

où  $\omega_1, \omega_2, \dots, \omega_k$  et les valeurs pondérées pour  $k = 1, 2, \dots, p$ , de telle sorte que  $\sum_{k=1}^p \omega_k = 1$ .

Enfin, le problème multiojectifs (MOLIFP) est transformé vers un problème mono-objectifs de la manière suivante :

$$(P_\omega) \begin{cases} \mathbf{max} & U_k(x) = \sum_{k=1}^p \omega_k \frac{c^k x + \alpha^k}{d^k x + \beta^k} \\ s.c & \\ x \in D & \\ \sum_{k=1}^p \omega_k = 1 & \end{cases}$$

le problème obtenu est un problème quadratique fractionnaire (c'est à dire la fonction objectif  $U(x)$  n'est pas linéaire)

**théorème 4.2.1.** [30] Si  $\hat{x}$  est une solution optimale du problème paramétrique  $(P_\omega)$  pour un certain  $\omega > 0$ , alors  $\hat{x}$  est efficace pour le problème (MOILFP).

## 4.3 Exemple numérique

Considérons le problème de programmation fractionnaire multi-objectif suivant :

$$\left\{ \begin{array}{l} \max Z^1(x) = \frac{x_1 + 3x_2 + \frac{3}{2}x_3}{\frac{1}{2}x_1 + \frac{3}{4}x_2 + \frac{7}{8}x_3} \\ \max Z^2(x) = \frac{3x_1 + x_2 + x_3}{\frac{1}{2}x_1 + x_2 + \frac{2}{9}x_3} \\ \text{s.c} \\ x_1 + 2x_2 - x_3 \leq 26 \\ -2x_1 + 3x_2 + x_3 \leq 32 \\ x_1 + x_2 + x_3 \leq 37 \\ x_1 + 2x_2 + x_3 \leq 30 \\ -2x_1 + 4x_3 \leq 25 \\ x_i \geq 0, \text{entiers} \quad i = 1, 2, 3 \end{array} \right. \quad (4.2)$$

Après, la transformation du problème multiobjectifs vers le problème mono-objectifs, en utilisant la méthode de somme pondérée on aura le problème suivant :

$$\left\{ \begin{array}{l} \max U(x) = w_1 \frac{x_1 + 3x_2 + \frac{3}{2}x_3}{\frac{1}{2}x_1 + \frac{3}{4}x_2 + \frac{7}{8}x_3} + w_2 \frac{3x_1 + x_2 + x_3}{\frac{1}{2}x_1 + x_2 + \frac{2}{9}x_3} \\ \text{s.c} \\ x_1 + 2x_2 - x_3 \leq 26 \\ -2x_1 + 3x_2 + x_3 \leq 32 \\ x_1 + x_2 + x_3 \leq 37 \\ x_1 + 2x_2 + x_3 \leq 30 \\ -2x_1 + 4x_3 \leq 25 \\ x_i \geq 0, \text{entiers} \quad i = 1, 2, 3 \end{array} \right. \quad (4.3)$$

Nous utiliserons le logiciel lingo pour résolution des problèmes fractionnaire quadratique en nombre entier

## 4.4 Présentation du logiciel

LINGO nous permet de formuler rapidement et facilement nos problèmes d'optimisation linéaire, non-linéaire ou en nombres entiers. Grâce à ses outils de modélisation, les modèles sont exprimés de manière transparente à l'aide de sommes et de variables indicées.

La méthode ne diffère guère de la méthode traditionnelle avec crayon et papier, mais les modèles seront plus faciles à réutiliser et mettre à jour. Il possède quatre solveurs qu'il utilise afin de résoudre les différents types de modèles :

- Solveur direct.
- Solveur linéaire
- Solveur non linéaire.
- Méthode de type séparation et évaluation.

de plus, LINGO est :

- Un moyen pour confirmer que l'optimum trouver est l'optimum global.

- Possibilité pour résoudre les problèmes plus rapidement.
- Un moyen amélioré pour résoudre beaucoup de type de problèmes.
- Un moyen de décomposition si un modèle contient des sous-modèles.

LINGO est livré avec un jeu de solveurs pour l'optimisation linéaire, non-linéaire convexe ou non convexe), quadratique, sous contraintes, et en nombre entier. Nous avons même pas à nous préoccuper du choix du solveur : en effet, LINGO interprète lui-même nos formulations et sélectionne automatiquement le solveur adapté à chaque problème. Un modèle d'optimisation se compose de trois parties :

- La fonction objective : c'est la formule simple qui décrit exactement ce que le modèle devrait optimiser.
- Les variables : ce sont les quantités qui peuvent être changés pour déterminer la valeur optimale de la fonction objective.
- Les contraintes : ce sont les formules qui définissent les limites sur les valeurs des variables.

### Les fonctions utilisées dans un modèle de LINGO sont :

@*FOR* : utilisée pour produire des contraintes.

@*SUM* : calcul de la somme.

@*MAX* : recherche de maximum.

@*MIN* : recherche de minimum.

### Type de variables dans LINGO :

Toutes les variables dans un modèle de LINGO sont considérées non négatives et continus, les fonctions variables d'un modèle de LINGO sont :

@*GIN* : toute valeur positive de nombre entiers.

@*BIN* : une valeur binaire (0 ou 1). @*FREE* : toute valeur positive ou négative réelle.

@*BND* : toute valeur bornée par des limites indiquées.

LA forme générale pour la déclaration d'un variable  $x$ , en utilisant les fonctions variables @*GIN*, @*BIN*, @*FREE* est : @*fonction*( $x$ ).

La fonction @*BND* inclut les limites inférieure et supérieure des variables, sa forme générale est : @*BND* (limite inférieure,  $x$ , limite supérieure).

#### 4.4.1 Interface de logiciel :

Après,avoir entré le problème de programmation fractionnaire multiobjectifs,avec les poids  $\omega_1, \omega_2$  tel que  $\omega_1 = 0.4, \omega_2 = 0.6$  on aura la fenêtre suivante :

## CHAPITRE 4. APPROCHE DE RÉOLUTION DU PROBLÈME MOIFLP

```

Lingo Model - Lingoexemple master
max=0.4*((x1+3*x2+(3/2)*x3)/((1/2)*x1+(3/4)*x2+(7/8)*x3))+0.6*((3*x1+x2+x3)/((1/2)*x1+x2+(2/9)*x3));
z1=(x1+3*x2+(3/2)*x3)/((1/2)*x1+(3/4)*x2+(7/8)*x3);
z2=(3*x1+x2+x3)/((1/2)*x1+x2+(2/9)*x3);

x1+2*x2-x3<=26;
-2*x1+3*x2+x3<=32;
x1+x2+x3<=37;
-2*x1+4*x3<=25;
x1+2*x2+x3<=30;
@GIN (x1);
@GIN (x2);
@GIN (x3);

end

```

Par conséquent, on obtient une solution non dominée, où  $x^* = (x_1^*, x_2^*, x_3^*) = (26, 0, 0)$  avec  $U(x^*) = 4.4$  et  $(Z^1, Z^2) = (2, 6)$ . Cette solution est une solution optimale de (4.3), avec  $w_1 = 0.4$  et  $w_2 = 0.6$ .

The screenshot shows the 'Solution Report' window in Lingo 18.0. It displays the following information:

- Objective value: 4.400000
- Objective bound: 4.400000
- Infeasibilities: 0.000000
- Extended solver steps: 0
- Total solver iterations: 33
- Elapsed runtime seconds: 0.05
- Model Class: MINLP
- Total variables: 5
- Nonlinear variables: 3
- Integer variables: 3
- Total constraints: 8
- Nonlinear constraints: 3
- Total nonzeros: 25
- Nonlinear nonzeros: 9

The variable values and reduced costs are as follows:

Variable	Value	Reduced Cost
X1	26.000000	0.2307692E-01
X2	0.000000	0.2307692
X3	0.000000	0.000000
Z1	2.000000	0.000000
Z2	6.000000	0.000000

The dual prices for the constraints are:

Row	Slack or Surplus	Dual Price
1	4.400000	1.000000
2	0.000000	0.000000
3	0.000000	0.000000
4	0.000000	0.2307692E-01
5	54.00000	0.000000
6	11.00000	0.000000

Cette fenêtre nous donne la valeur de l'optimum qui produit la valeur optimale de la fonction objectif.

Le tableau(4.1) donne l'ensemble des solutions non dominées et les valeurs optimales correspondantes de la fonction objectif. Ces résultats sont obtenus à partir du logiciel Lingo en faisant varier les poids  $w_1$  et  $w_2$ . Les résultats obtenus peuvent donner une certaine flexibilité au décideur quand au choix des poids de la fonction objectif.

TABLE 4.1 – Résultats de l'optimisation en faisant varier les poids  $w_1$  et  $w_2$

$w_1$	$w_2$	$x_1$	$x_2$	$x_3$	$U(x)$	$(Z^1, Z^2)$
0	1	26	0	0	6	(2, 6)
0.1	0.9	26	0	0	5.6	(2, 6)
0.2	0.8	26	0	0	5.2	(2, 6)
0.3	0.7	26	0	0	4.8	(2, 6)
0.4	0.6	26	0	0	4.4	(2, 6)
0.5	0.5	26	0	0	4	(2, 6)
0.6	0.4	26	0	0	3.6	(2, 6)
0.7	0.3	0	10	0	3.1	(4, 1)
0.8	0.2	0	10	0	3.4	(3, 0)
0.9	0.1	0	10	0	3.7	(4, 1)
1	0	0	10	0	4	(4, 1)

## 4.5 Conclusion

Dans ce chapitre, Nous sommes intéressées principalement sur l'approche de résolution du problème d'optimisation fractionnaire multi-objectifs en nombre entier. En premier lieu, nous avons convertir le problème fractionnaire multiobjectifs vers un problème mono-objectif. Ensuite nous avons présenté le logiciel lingo et les interfaces nécessaires pour exécuté un problème fractionnaire quadratique. En dernier lieu, nous avons présenté un exemple sur le logiciel lingo.

# Conclusion générale

Les problèmes de programmation multiobjectifs sont de plus en plus étudiés car ils ont un grand intérêt dans l'industrie, où plusieurs objectifs concurrents doivent être satisfaits. Ils sont la plupart du temps NP-difficiles même si leur version mono-objectif ne l'est pas. Contrairement aux problèmes d'optimisation mono-objectif, il existe généralement plusieurs solutions Pareto optimales ou efficaces en optimisation multi-objectifs, c'est-à-dire des solutions qu'on ne peut améliorer sur un objectif sans les détériorer sur un autre. Comme il n'existe aucune solution meilleure en tout point qu'une autre, un compromis différent selon les personnes doit être choisi. Le choix est donc subjectif, et il est indispensable de proposer l'ensemble des choix possibles afin de ne pas exclure une possibilité. L'optimisation multi-objectifs est donc avant tout un outil d'aide à la décision, et c'est une personne qui prendra la décision finale.

Nous avons présenté dans ce mémoire les travaux de recherche sur les méthodes d'optimisation multi-objectifs fractionnaire linéaires.

Dans notre étude, nous nous sommes particulièrement intéressés aux problèmes de programmation multi-objectifs fractionnaire linéaires en nombres entiers pour lesquels relativement peu de travaux ont été réalisés. Le noyau du thème de notre mémoire est principalement focalisé sur l'optimisation linéaire multi-objectifs en nombres entiers et le travail présenté s'articule autour de deux volets :

- Le premier volet consiste à rappeler la notion d'optimisation multi-objectifs fractionnaire linéaire en nombre entier (MOILFP).
- Le deuxième volet consiste à résoudre le problème d'optimisation fractionnaire multi-objectifs en nombre entier en utilisant la méthode de somme pondérée.

# Notation

- (**PLE**) :problème linéaire en nombre entier.
- (**PFE**) :problème linéaire fractionnaire en nombre entier.
- (**PLMO**) :problème linéaire multiobjectifs.
- (**PLMOE**) :problème linéaire multiobjectifs en nombre entier.
- (**PLFMO**) :problème linéaire fractionnaire multiobjectifs.
- (**PLFMOE**) :problème linéaire fractionnaire multiobjectifs en nombre entier.
- (**PLIFMOE**) :problème linéaire fractionnaire integer multiobjectifs en nombre entier.
- D** :l'ensemble des entiers relatifs.
- S** :l'ensemble des solutions réalisable.
- SES** :solution efficace supportée.
- SENS** :solution efficace non supportée.
- $x^*$  :la solution optimale

# Bibliographie

- [1] **M.Moulaï**. Thèse de doctorat : Optimisation Multicritere Fractionnaire Linéaire en Nombres entiers 2002.
- [2] **O.Zerdani**. Thèse de doctorat : L'optimisation non linéaire multi objectif 2013,.
- [3] **E.U. Choo**. Proper efficiency and the linear fractional vector maximum problem. Operations Research, 32(1) :216–220, 1984.
- [4] **V. Pareto**. The new theories of economics. Journal of political economy, 5(4) :485–502, 1897.
- [5] **V.Chankong and Y.Y.Haimes**, "Multiobjective Decision Making : Theory and Methodology", North Holland Series in System Science and Engeneering, p.406, 1983.
- [6] **A.Berro**, "Optimisation Multi-objectifs et Stratégie d'Evolution en Environnement Dynamique", Thèse de doctorat, université des sciences sociale de Toulouse, 2001.
- [7] **Y.Collette, P.Siarry**, "Optimisation Multi-objectifs", édition Eyrolles 61, 2002.
- [8] **Y.Y.Haimes**, "Modeling and Control of the Pollution of Water Resources Systems via Multilevel Approach", New-York, 1971.
- [9] **L.ASLI** . Approche hybride pour les problèmes d'optimisation combinatoire multiobjectif : cas des problèmes de type sac-à-dos 2010.
- [10] **Y.Y.Haimes, L.S.Lasdon and D.A.Wisner**, "Formulation of the Problems of Integrated System Identification and System Optimization", 1986.
- [11] **A. Charnes and W.W. Cooper**. Management models and industrial applications of linear programming. Management science,
- [12] **F.Kebaili, H.Mouloud**, "Optimisation de le Gestion de Portefeuille d'Actifs Financiers, mémoire de fin d'étude", université de USTHB, 2008.
- [13] **E.Zitzler, L. Thiele** : Multiobjective evolutionary algorithms : A comparative case study and the strength pareto approach. IEEE Transactions on Evolutionary Computation 3(4), 257–271 (1999)
- [14] **W.Dinkelbach** , "On nonlinear fractional programming ", Management Science, 1967, vol, 13, pp.492-498



- [15] **V.Chankong and Y.Y.Haimes**, "Multiobjective Decision Making : Theory and Methodology", North Holland Series in System Science and Engineering, p.406, 1983.
- [16] **A.M. Geoffrion**. Proper efficiency and the theory of vector maximization. Journal of mathematical analysis and applications, 22(3) :618–630, 1968.
- [17] **J.P. Costa**. Computing non-dominated solutions in molfp. European Journal of Operational Research, 181(3) :1464–1475, 2007.
- [18] **M.E.A. Chergui and M. Moulaï**. An exact method for a discrete multiobjective linear fractional optimization. Journal of Applied Mathematics et Decision Sciences, 2008(1), 2008.
- [19] **M.Abbac and M.Moulai** Integer linear fractional programming with multiple objective. Journal of the Italian Operations Research Society, 32(103-104) :15–38, 2002.
- [20] **M.E.A. Chergui and M. Moulaï**. An exact method for a discrete multiobjective linear fractional optimization. Journal of Applied Mathematics and Decision Sciences, 2008(1), 2008.
- [21] **F.Y.Edgeworth**, "Mathematical Physics", Keagan.P, Londres, Angleterre, 1881.
- [22] **Pareto Vilfredo**, "cours d'économie politique" **J.R. Isbell and W.H. Marlow**. Attrition games. Naval Research Logistics Quarterly, Vol.1 et 2, Rouge Lausanne, 1896.
- [23] **R.E. Steuer**. Multiple criteria optimization. Theory, computation and applications, 1986.
- [24] **A. Cambini and L. Martein**. A modified version of martos' algorithm. Methods of Operation Research, 53 :33–44, 1986.
- [25] 3(1-2) :71–94, 1956.
- [26] **A. Nagih and G. Plateau**. Problèmes fractionnaires : tour d'horizon sur les applications et méthodes de résolution. RAIRO-Operations Research-Recherche Opérationnelle, 33(4) :383–419, 1999.
- [27] **R.E.Gomory**, "Outline of an Algorithm for Integer Solutions to Linear Programs", Bulletin of the AMS 64, p.275-278, 1958.
- [28] **B. Bereanu**. Decision regions and minimum risk solutions in linear programming. In colloquium on applications of mathematics to economics, Budapest, pages 37–42, 1963.
- [29] **A.Charnes,W.cooper**.programming with linear fractional functionals,naval ref,logist.quart.9,181-186,1962
- [30] **A.Merakeb**.optimisation multicritères en contrôle optimal :application au véhicule électrique, thèse de doctorat,université de Tizi-ouzou,2011
- [31] **K.Miettinen**, "non linear multiobjective Optimization", Kluwer, Boston, 1999.

# Résumé

La résolution des problèmes multi-objectifs, particulièrement les problèmes d'optimisation fractionnaire multi-objectifs se fait ou bien dans l'espace des critères ou dans l'espaces des décisions. La notion d'optimalité disparaît pour les problèmes de ce type au profit de la notion d'efficacité. Une solution efficace est une solution à partir de laquelle, il est impossible d'augmenter la valeur d'un critère sans diminuer celle d'au moins un autre. La solution choisie par un décideur sera un compromis dépendant d'un grand nombre de paramètres variant d'un décideur à un autre et donc difficile à modéliser. Notre objectif vise la résolution des problèmes multi-objectifs fractionnaires en présence de variables discrètes et l'approche de résolution est basée sur la méthode de la somme pondérée.

## Abstract

The resolution of multi-objective problems, particularly multi-objective fractional optimization problems, is done either in the criterion space or in the decision space. The notion of optimality disappears for problems of this type in favor of the notion of efficiency. An effective solution is a solution from which it is impossible to increase the value of one criterion without decreasing that of at least one other. The solution chosen by a decision maker will be a compromise depending on a large number of parameters varying from one decision maker to another and therefore difficult to model. Our goal is to solve fractional multi-objective problems in the presence of discrete variables and the resolution approach is based on the weighted sum method.