

# Table des matières

<b>Table des matières</b>	<b>1</b>
<b>Introduction générale</b>	<b>2</b>
<b>1 Optimisation mono-objectif fractionnaire linéaire</b>	<b>6</b>
1.1 Introduction . . . . .	6
1.2 Notions de convexité et de convexité généralisée . . . . .	7
1.3 Programmation fractionnaire linéaire . . . . .	14
1.3.1 Propriété d'un programme fractionnaire linéaire . . . . .	14
1.4 Méthodes de résolution . . . . .	16
1.4.1 Méthode graphique . . . . .	16
1.4.2 Méthode de gradient . . . . .	18
1.4.3 La transformation de charnes et cooper . . . . .	19
1.4.4 La méthode de Dinkelbach . . . . .	20
1.4.5 La méthode de Cambini . . . . .	21
1.5 Programmation fractionnaire linéaire en nombres entiers . . . . .	22
1.5.1 Méthodes de résolution . . . . .	22
<b>2 Optimisation mono-objectif fractionnaire non linéaire</b>	<b>25</b>
2.1 Introduction . . . . .	25
2.2 Problème d'optimisation non linéaire . . . . .	25
2.2.1 Définition . . . . .	25
2.2.2 Notions de bases . . . . .	26
2.2.3 Conditions de Kuhn-Tucker . . . . .	27
2.3 Modèles fractionnaire non linéaires . . . . .	27
2.3.1 Programmation fractionnaire non linéaire . . . . .	27
2.3.2 Programmation fractionnaire quadratique . . . . .	30
2.4 Application . . . . .	32

2.4.1	Un planning d'arrimage de porte-conteneurs . . . . .	32
<b>3</b>	<b>Résolution d'un problème fractionnaire mono-objectif linéaire via la programmation linéaire</b>	<b>34</b>
3.1	Introduction . . . . .	34
3.2	Formulation du programme fractionnaire . . . . .	35
3.3	Formulation de (PFL) en (PL) . . . . .	36
3.3.1	Formulation de Charnes et Cooper (FC) . . . . .	36
3.3.2	Formulation de Bennani et Benterki (FB) . . . . .	38
3.4	Principe générale de la méthode projective de Ye-Lustig . . . . .	39
3.5	Expérimentations numériques . . . . .	43
3.5.1	Exemple à taille fixe . . . . .	44
3.5.2	Exemples à taille variable . . . . .	47
<b>4</b>	<b>Optimisation multi-objectif fractionnaire en nombre entiers</b>	<b>50</b>
4.1	Introduction . . . . .	50
4.2	Généralités sur la programmation multi-objectif . . . . .	51
4.2.1	Concepts de base et terminologie . . . . .	51
4.2.2	Notion de dominance . . . . .	52
4.2.3	Relations dérivées de la dominance . . . . .	53
4.2.4	Front de Pareto . . . . .	55
4.2.5	Les méthodes de résolutions des problèmes d'optimisation multi-objectif . . . . .	57
4.2.6	Programmation multi-objectif en nombres entiers . . . . .	60
4.3	Programme multi-objectif fractionnaire linéaire en nombres entiers .	61
4.3.1	Définition du modèle . . . . .	61
4.3.2	Approches de résolution . . . . .	62
4.3.3	Méthode de Abbas et Moulai . . . . .	62
	<b>Bibliographie</b>	<b>69</b>

# Introduction générale

L'optimisation est une branche importante des mathématiques. Elle est à la fois une science et un outil largement utilisée dans divers domaines scientifiques, en ingénierie comme en industrie, qui nous aide à prendre la meilleure décision parmi plusieurs nombres de décisions possibles.

L'optimisation consiste à chercher la solution qui maximise ou minimise la fonction objectif ( qui sert de critère pour déterminer la meilleure solution pour un problème d'optimisation). On le fait en respectant des contraintes qui sont représentées sous la forme d'un système d'inéquations. Il existe deux types de méthodes d'optimisation (mono-objectif et multi-objectif).

Dans un problème mono-objectif, un seul critère exprimé en fonction des variables de décision doit être optimisé. Cependant dans les situations rencontrée en pratique, la considération d'un seul critère ne permet pas une bonne modélisation du problème, la plupart des problèmes réels intervenant en mathématique de décision sont de nature qui impose la prise en compte de plusieurs critères qui sont souvent contradictoires. Tout décideur est obligé de tenir compte du maximum d'éléments en sa possession pour aboutir à la meilleur décision possible. En l'absence d'une solution unique qui optimise tous les objectifs simultanément, l'optimisation multi-objectifs consiste à trouver l'ensemble des solutions qui correspondent au meilleur compromis entre objectifs. La recherche d'un compromis a conduit à la définition du concept de solution efficace, également appelée solution optimale de Pareto. L'ensemble de toutes les solutions efficaces d'un problème d'optimisation multi-objectif (MOP) est appelé l'ensemble efficace, et dans l'espace des critères, on l'appelle front de Pareto.

Dans certains problèmes de décision, la modélisation de l'optimiseur implique des variables de décision avec des valeurs entières, on parlera de problèmes d'optimisation en nombres entiers. L'approche classique pour le cas continu n'est plus directement applicable, puisque le domaine des solutions réalisables change de structure.

En ce qui consiste l'optimisation multi-critère, nous nous sommes particulièrement intéressées au cas où les critères sont linéaires fractionnaires, c'est à dire on optimise un objectif qui s'écrit comme le rapport de deux fonctions linéaires ou affines sous contraintes linéaires et variables de décision entières.

Dans ce modeste travail, nous présentons des méthodes exactes pour la résolution des problèmes mono et multi objectifs fractionnaires linéaires et non linéaire .

Pour cela nous avons structuré notre mémoire comme suit :

◇ **Chapitre 1** : On s'intéresse à la programmation fractionnaire linéaire mono-objectif, en rappelant les définitions de base et quelques méthodes de résolution de ce type de problème. [17], [13], [10], [9], [7], [4], [28], [21], [18] .

◇ **Chapitre 2** : Dans ce chapitre nous allons définir un programme d'optimisation fractionnaire non linéaire mono-objectif, citer quelques applications de ces modèles et présenter quelques algorithmes de résolution. On s'intéressera plus particulièrement au cas où la fonction objectif est le rapport de deux fonctions non linéaire, chacune de ces fonction est le produit de deux fonctions linéaires. [13, 12, 25, 23, 18].

◇ **Chapitre 3** : On présente une étude détaillée sur les problèmes de programmations fractionnaires linéaires sous contraintes linéaires en donnons

des résultats d'existence et unicité d'une solution optimale et les méthodes de résolution de (PFL : Programme Fractionnaire Linéaire). Par la suite on utilise une nouvelle formulation qui permet de convertir (PFL) en un (PL : Programme Linéaire) équivalent. Ce chapitre sera achevé par des tests numériques d'ordre comparatif réalisés sur des exemples de différentes tailles à savoir : Taille fixe et Taille variable. [24, 3] .

◇ **Chapitre 4** : Il est consacré à la présentation des problèmes d'optimisation multi-objectif fractionnaire en nombres entiers, et à l'étude d'une nouvelle méthode de résolution de Abbas et Moulai [2].

Enfin nous terminons notre travail par une conclusion générale.

# Chapitre 1

## Optimisation mono-objectif fractionnaire linéaire

### 1.1 Introduction

Dans de nombreuses applications de programmation non linéaire, le rapport de deux fonctions doit être maximisé ou minimisé, et ces problèmes sont appelés programmations fractionnaires.

La formule mathématique de la programmation fractionnaire peut s'écrire comme suit :

$$(\mathbf{PF}) \begin{cases} \max Z(x) = \frac{f(x)}{g(x)}, \\ h(x) \leq 0 \end{cases} \quad (1.1)$$

Sous les hypothèse classique :

- $f$ ,  $g$  et  $h$  sont des fonctions continue sur  $\mathbb{R}^n$ ,
- L'ensemble de solutions réalisables  $X = \{x \in \mathbb{R}^n / h(x) \leq 0\}$  n'est pas vide,
- $\forall x \in X : g(x) > 0$ ,

Lorsque les deux fonctions de ce rapport sont des fonctions affines et que le domaine des solutions réalisables est décrit par des contraintes linéaires, on parle alors de **programmation fractionnaire linéaire** [21].

## 1.2 Notions de convexité et de convexité généralisée

La convexité est l'une des hypothèses les plus couramment utilisées dans la théorie de l'optimisation, en particulier dans la programmation linéaire. Cependant, pour de nombreux modèles mathématiques utilisés en économie et en ingénierie, cette propriété n'est pas vérifiée ni par les fonctions ni par les ensembles.

Par conséquent, la notion de convexité a été étendue aux notions pseudoconvexité, quasiconvexité. [28]

### Définition 1.1. Ensemble convexe [28]

L'ensemble  $S \in \mathbb{R}$  est dit convexe si le segment qui relie deux points quelconques de  $S$  est contenu dans l'ensemble  $S$ .

Autrement dit : un ensemble  $S$  est convexe si :

$$\forall x_1, x_2 \in S, \quad \forall \lambda \in [0, 1] : \quad \lambda x_1 + (1 - \lambda)x_2 \in S. \quad (1.2)$$

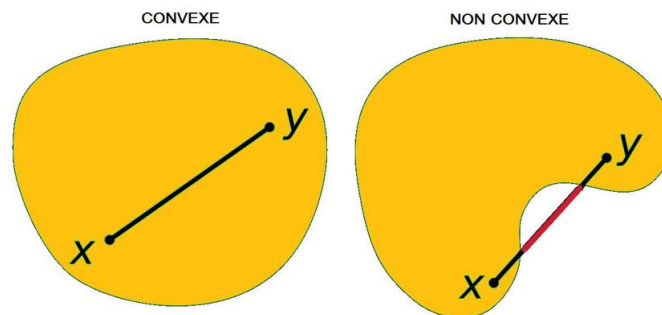


FIGURE 1.1 -- Exemples d'ensemble convexe et d'ensemble non convexe

### Définition 1.2. Fonction convexe [28]

Une fonction  $Z$  définie sur un ensemble convexe  $S$  de  $\mathbb{R}^n$  est dite convexe si

$$\forall x_1, x_2 \in S, \quad \forall \lambda \in [0, 1] : Z(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda Z(x_1) + (1 - \lambda)Z(x_2) \quad (1.3)$$

**Définition 1.3. Fonction quasiconcave.**[28]

Une fonction  $Z$  définie sur un ensemble convexe  $S$  de  $\mathbb{R}^n$  est dite quasiconcave si :

$$\forall x_1, x_2 \in S, \quad \forall \lambda \in [0, 1] : Z(x_1) \geq Z(x_2) \Rightarrow Z(\lambda x_1 + (1 - \lambda)x_2) \geq Z(x_2) \quad (1.4)$$

**Définition 1.4. Fonction quasiconvexe**[28]

Une fonction  $Z$  définie sur un ensemble convexe  $S$  de  $\mathbb{R}^n$  est dite quasiconvexe si :

$$\forall x_1, x_2 \in S, \quad \forall \lambda \in [0, 1] : Z(x_1) \geq Z(x_2) \Rightarrow Z(\lambda x_1 + (1 - \lambda)x_2) \leq Z(x_2) \quad (1.5)$$

**Remarque 1.1.** 1. Pour une fonction quasiconvexe  $Z$ , si  $Z(x_1) \geq Z(x_2)$  alors  $Z(x_1)$  est plus grand ou égale à  $Z$  en toute combinaison convexe de  $x_1$  et  $x_2$ . Donc si  $Z$  augmente de sa valeur en un point le long de n'importe quelle direction, elle doit rester croissante dans cette direction .

2. Pour une fonction quasiconcave  $Z$ , si  $Z(x_1) \geq Z(x_2)$  alors  $Z$ , en toute combinaison convexe de  $x_1$  et  $x_2$ , est plus grand ou égale à  $Z(x_2)$ .

Des définitions équivalentes d'une fonction quasiconcave sont données par le théorème suivant :

**Théorème 1.1.** [21]

Soit  $Z$  une fonction définie sur un ensemble convexe  $S$  de  $\mathbb{R}^n$ , alors les assertions suivantes sont équivalentes :

- $Z$  est une fonction quasiconcave sur l'ensemble  $S$ .
- Pour chaque nombre réel  $\lambda$ , l'ensemble  $C^+$  défini par :

$$C^+ = \{x \in S / Z(x) \geq \lambda\} \quad (1.6)$$

est un ensemble convexe.

$$- \forall x_1, x_2 \in S, \quad \forall \lambda \in [0, 1] : Z(\lambda x_1 + (1 - \lambda)x_2) \geq \min(Z(x_1), Z(x_2)).$$

De même, des définitions équivalentes d'une fonction quasiconvexe sont données par le théorème suivant :

**Théorème 1.2.** [21]

Soit  $Z$  une fonction définie sur un ensemble convexe  $S$  de  $\mathbb{R}^n$ , alors les assertions suivantes sont équivalentes :

- $Z$  est une fonction quasiconvexe sur l'ensemble  $S$ .
- Pour chaque nombre réel  $\lambda$ , l'ensemble  $C^-$  défini par :

$$C^- = \{x \in S / Z(x) \leq \lambda\} \tag{1.7}$$

est un ensemble convexe.

$$- \forall x_1, x_2 \in S, \quad \forall \lambda \in [0, 1] : Z(\lambda x_1 + (1 - \lambda)x_2) \leq \max(Z(x_1), Z(x_2)).$$

**Théorème 1.3.** [21]

soit  $S$  un polyèdre non vide de  $\mathbb{R}^n$  et soit  $Z : \mathbb{R}^n \rightarrow \mathbb{R}$  une fonction quasiconvexe et continue sur  $S$ .

On considère le problème de maximiser  $Z$  sur l'ensemble  $S$ , donc une solution optimale  $\bar{x}$  du problème existe où  $\bar{x}$  est un point extrême de  $S$

**Preuve.** Notons que  $Z$  est continue sur  $S$  et donc atteint un maximum en un point  $x' \in S$ . Soient  $x_1, \dots, x_k$  les points extrêmes de  $S$  et supposons que  $Z(x') > Z(x_j)$  pour  $j = 1, \dots, k$ .

On a  $x' = \sum_{i=1}^k \lambda_i x_i$

$\sum_{i=1}^k \lambda_i = 1, \lambda_i \geq 0$  pour  $i = 1, \dots, k$ .

comme  $Z(x') > Z(x_j)$  pour chaque  $j$ , alors

$$Z(x') > \max\{Z(x_j)\} = \alpha. \tag{1.8}$$

Maintenant, on considère la partie  $S_\alpha = \{x \in S : Z(x) < \alpha\}$ .

Notons que  $x_j \in S_\alpha$  pour  $j = 1, \dots, k$  et par la quasiconvexité de  $Z$ ,  $S_\alpha$  est convexe. D'où  $x' = \sum_{i=1}^k \lambda_j x_j \in S_\alpha$ , ce qui implique que  $Z(x') < \alpha$  ce qui contredit (1.8).

Cette contradiction montre que  $Z(x') = Z(x_j)$  pour au moins un point extrême  $x_j$ . Un résultat très important qui concerne les fonctions quasiconcaves dans la théorie de l'optimisation est le suivant, les conditions de premier ordre de Kuhn-Tucker sont "presque" suffisantes pour identifier un optimum globale d'un problème de maximisation avec des contraintes d'inégalités si toutes les fonctions impliquées dans le modèle sont quasiconcaves.

**Théorème 1.4.** [28]

Soit  $Z$  et  $h_i, i = 1, \dots, k$  des fonctions continues quasiconcave de  $U \subset \mathbb{R}^n$  dans  $\mathbb{R}$  où  $U$  est un convexe ouvert.

On définit

$$D = \{x \in U / h_i(x) \geq 0, \quad i = 1, \dots, k\}$$

On suppose qu'il existe  $x^* \in D$  et  $\lambda \in \mathbb{R}^k$  tel que les conditions de premier ordre de Kuhn-Tucker sont vérifiées :

$$\mathbf{KT-1} \quad \nabla Z(x^*) + \sum_{i=1}^k \lambda_i \nabla h_i(x^*) = 0$$

$$\mathbf{KT-2} \quad \lambda_i \geq 0, \quad \lambda_i h_i(x^*) = 0, \quad i = 1, \dots, k.$$

Alors  $x^*$  maximise  $Z$  sur  $D$  si au moins une des conditions suivantes est vérifiée

$$\mathbf{QC-1} \quad \nabla Z(x^*) = 0 .$$

$$\mathbf{QC-2} \quad \nabla Z \text{ est concave.}$$

**Définition 1.5. Fonction strictement quasiconcave**

Une fonction  $Z$  définie sur un ensemble convexe  $S$  de  $\mathbb{R}^n$  est dite strictement quasiconcave si

$$\forall x_1, x_2 \in S, \quad \forall \lambda \in ]0, 1[ : Z(x_1) > Z(x_2) \Rightarrow Z(\lambda x_1 + (1 - \lambda)x_2) > Z(x_2). \quad (1.9)$$

### Fonction strictement quasiconvexe

Une fonction  $Z$  définie sur un ensemble convexe  $S$  de  $\mathbb{R}^n$  est dite strictement quasiconvexe si

$$\forall x_1, x_2 \in S, \quad \forall \lambda \in ]0, 1[ : Z(x_1) > Z(x_2) \Rightarrow Z(\lambda x_1 + (1 - \lambda)x_2) < Z(x_1). \quad (1.10)$$

#### Théorème 1.5. [28]

Soit  $Z : \mathbb{R}^n \rightarrow \mathbb{R}$  une fonction strictement quasiconcave. On considère le problème de maximiser  $Z$  sur  $S$ , où  $S$  est un convexe non vide de  $\mathbb{R}^n$ . Si  $\bar{x}$  est une solution optimale locale alors  $\bar{x}$  est une solution optimale globale. De plus, si cet optimum global existe alors il est unique.

#### Théorème 1.6. [28]

Soit  $Z : \mathbb{R}^n \rightarrow \mathbb{R}$  une fonction strictement quasiconvexe. On considère le problème de minimiser  $Z$  sur  $S$ , où  $S$  est un convexe non vide de  $\mathbb{R}^n$ . Si  $\bar{x}$  est une solution optimale locale alors  $\bar{x}$  est une solution optimale globale. De plus, si cet optimum global existe alors il est unique.

#### Preuve.

On résout par l'absurde : Supposons qu'il existe un point  $\hat{x} \in S$  tel que  $Z(\hat{x}) < Z(\bar{x})$ . Par la convexité de  $S$ , on a pour chaque  $\lambda \in ]0, 1[$ ,  $\lambda \hat{x} + (1 - \lambda)\bar{x} \in S$ .

Comme  $\bar{x}$  est un minimum local par hypothèse,  $Z(\bar{x}) \leq Z(\lambda \hat{x} + (1 - \lambda)\bar{x})$  pour tout  $\lambda \in ]0, \delta[$  et pour  $\delta \in ]0, 1[$ .

Mais comme  $Z$  est strictement quasiconvexe et  $Z(\hat{x}) < Z(\bar{x})$ , on obtient

$Z(\lambda \hat{x} + (1 - \lambda)\bar{x}) < Z(\bar{x})$  pour  $\lambda \in ]0, 1[$ . Ceci contredit l'optimalité locale de  $\bar{x}$ , d'où la preuve.

Pour prouver l'unicité de l'optimum global  $\bar{x}$ , on suppose qu'il existe un point  $\hat{x} \in S$  tel que  $Z(\hat{x}) = Z(\bar{x})$ .

On choisit  $\lambda \in ]0, \delta[$ , soit  $y = (\lambda \hat{x} + (1 - \lambda)\bar{x})$ . Comme  $S$  est convexe,  $y \in S$ . Comme  $Z$  est strictement quasiconvexe,  $Z(y) = Z(\lambda \hat{x} + (1 - \lambda)\bar{x}) < \max(Z(\hat{x}), Z(\bar{x}))$ . Ceci contredit que  $\bar{x}$  et  $\hat{x}$  soient des optimums. D'où l'unicité de la solution optimale.

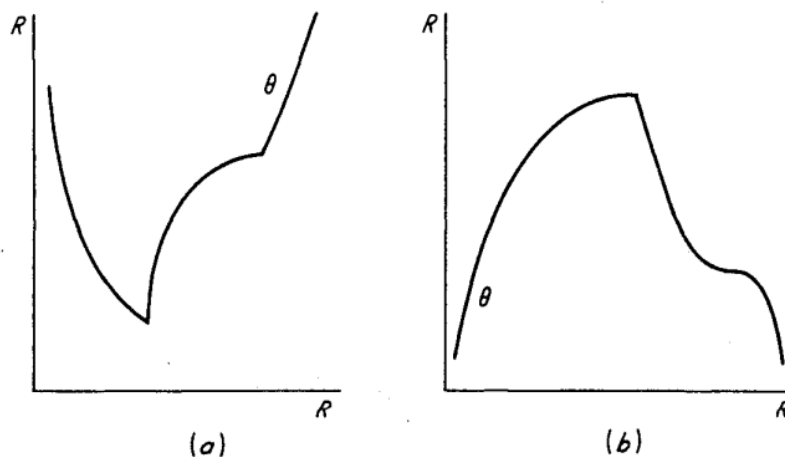


FIGURE 1.3 – Fonction strictement quasiconvexe/ strictement quasiconcave

La différentiabilité d'une fonction en un point correspond à l'existence d'une approximation linéaire de la fonction  $f$  au voisinage de ce point.

### Définition 1.6. Fonction différentiable

Soit  $X$  un ensemble non vide de  $\mathbb{R}^n$  et soit  $Z$  une fonction de  $X$  dans  $\mathbb{R}$ . On dit que  $Z$  est différentiable en un point  $\bar{x}$  s'il existe un vecteur gradient noté  $\nabla Z(\bar{x})$  et une fonction  $o : \mathbb{R}^n \rightarrow \mathbb{R}$  tels que :

$$Z(x) = Z(\bar{x}) + \nabla Z(\bar{x})(x - \bar{x}) + |x - \bar{x}|o(x - \bar{x}). \quad (1.11)$$

où

$$\lim_{x \rightarrow \bar{x}} o(x - \bar{x}) = 0 \text{ et } \nabla Z(\bar{x}) = \left( \frac{\delta Z}{\delta x_1}(\bar{x}), \dots, \frac{\delta Z}{\delta x_n}(\bar{x}) \right)$$

**Définition 1.7. fonction pseudoconvexe/ pseudoconcave**

soit  $Z : X \rightarrow \mathbb{R}$  une fonction différentiable sur un ensemble ouvert  $X \subseteq \mathbb{R}^n$  ; alors  $Z$  est pseudoconvexe sur  $X$  si

$$\forall x_1, x_2 \in X : Z(x_1) < Z(x_2) \Rightarrow (x_1 - x_2) \nabla Z(x_2) < 0, \quad (1.12)$$

et  $Z$  est une fonction pseudoconcave si

$$\forall x_1, x_2 \in X : Z(x_1) > Z(x_2) \Rightarrow (x_1 - x_2) \nabla Z(x_2) > 0, \quad (1.13)$$

**Proposition 1.1.**

1. Si une fonction  $Z$  est convexe alors elle est quasiconvexe. De plus, si  $Z$  est différentiable alors elle est pseudoconvexe.
2. De même, si une fonction  $Z$  est concave alors elle est quasiconcave. De plus, si  $Z$  est différentiable alors elle est pseudoconcave [4] .

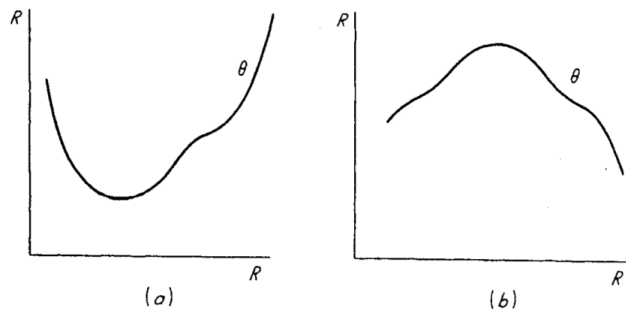


FIGURE 1.4 – Fonction Pseudoconvexe/ Pseudoconcave

### 1.3 Programmation fractionnaire linéaire

Un problème d'optimisation d'une fonction objectif qui est le rapport de deux fonction affines sous contraintes linéaires peut-être exprimer par le modèle suivant :

$$(\mathbf{FL}) \begin{cases} \max Z(x) = \frac{cx+c_0}{dx+d_0}, \\ Ax \leq b, x \geq 0, \end{cases} \quad (1.14)$$

#### 1.3.1 Propriété d'un programme fractionnaire linéaire

**Lemme 1.1.** [4]

Soit  $Z(x) = \frac{cx+c_0}{dx+d_0}$  et soit  $S$  un convexe tel que  $dx + d_0 \neq 0$  sur  $S$ .

Alors  $Z$  est à la fois pseudoconvexe et pseudoconcave sur  $S$ .

**Preuve du lemme 1.1.**

Notons d'abord que soit  $dx+d_0 < 0, \forall x \in S$ , soit  $dx+d_0 > 0, \forall x \in S$ , dans le cas contraire, il existe  $x_1$  et  $x_2$  de  $S$  tels que  $dx_1+d_0 < 0$  et  $dx_2+d_0 > 0$  et donc il existe un point  $x \in S$ ,  $x$  combinaison convexe de  $x_1$  et  $x_2$  tel que  $dx+d_0 = 0$  ce qui contredit notre hypothèse. On commence par montrer que  $Z$  est pseudoconvexe.

Supposons deux points  $x_1$  et  $x_2$  de  $S$  tels que  $(x_2 - x_1)\nabla Z(x_1) \geq 0$ , on doit montrer que  $Z(x_2) \geq Z(x_1)$ .

On note que :

$$\nabla Z(x_1) = \frac{(dx_1 + d_0)c - (cx_1 + c_0)d}{(dx_1 + d_0)^2} \quad (1.15)$$

Comme on a  $(x_2 - x_1)\nabla Z(x_1) \geq 0$  et comme  $(dx_1 + d_0)^2 > 0$ , il s'ensuit que :

$$0 \leq (x_2 - x_1)[(dx_1 + d_0)c - (cx_1 + c_0)d] = (cx_2 + c_0)(dx_1 + d_0) - (dx_2 + d_0)(cx_1 + c_0). \quad (1.16)$$

Par conséquent,  $(cx_2 + c_0)(dx_1 + d_0) \geq (dx_2 + d_0)(cx_1 + c_0)$ .

Mais comme les termes  $(dx_1 + d_0)$  et  $(dx_2 + d_0)$  sont de même signe, en divisant par  $(dx_1 + d_0) - (dx_2 + d_0) > 0$  on obtient :

$$\frac{cx_2+c_0}{dx_2+d_0} \geq \frac{cx_1+c_0}{dx_1+d_0}, \text{ c'est-à-dire } Z(x_2) \geq Z(x_1).$$

D'où  $Z$  est pseudoconvexe.

De même, on peut montrer que  $(x_2 - x_1)\nabla Z(x_1) \leq 0 \Rightarrow Z(x_2) \geq Z(x_1)$  et donc  $Z$  est pseudoconcave.

### Propriétés

Pour un problème de programmation fractionnaire linéaire, plusieurs implications de ce lemme peuvent être notées [4] :

1. Puisque la fonction objectif est à la fois pseudoconvexe et pseudoconcave, elle est alors quasiconvexe, quasiconcave, strictement quasiconvexe et strictement quasiconcave.
2. Puisque la fonction objectif est à la fois pseudoconvexe et pseudoconcave, alors un point satisfaisant les conditions **KKT** pour un problème de minimisation est aussi un minimum global sur l'ensemble des solutions réalisables. De même, un point satisfaisant les conditions de **KKT** pour un problème de maximisation est aussi un maximum global sur l'ensemble des solutions réalisables.
3. Puisque la fonction objectif est à la fois strictement quasiconvexe et strictement quasiconcave, alors un minimum local est aussi un minimum global sur l'ensemble des solutions réalisables. De même, un maximum local est aussi un maximum global sur l'ensemble des solutions réalisables.
4. Comme la fonction objectif est à la fois quasiconvexe et quasiconcave, si l'ensemble des solutions réalisables est borné, alors la fonction objectif possède un minimum en un point extrême du domaine réalisable et possède aussi un maximum en un point extrême de domaine réalisable.

Les résultats précédents concernant la fonction objectif sont très utiles et peuvent être utilisés pour développer des procédures de calcul appropriées pour résoudre le problème de programmation fractionnaire. En particulier, on peut cher-

cher parmi les points extrêmes du polyèdre jusqu'à atteindre un point satisfaisant les conditions **KKT**.

## 1.4 Méthodes de résolution

On distingue trois grandes classes de méthodes de résolutions des programmes fractionnaires.

- **La résolution directe** : Le programme reste sous sa forme originale, sans aucune modification de la fonction objectif, ni des contraintes. Ces méthodes ont été utilisées pour résoudre des programmes fractionnaires en variables continues, entières et bivalentes  $\{0, 1\}$
- **La résolution par paramétrisation** : Dans cette stratégie on modifie la fonction objectif sans toucher l'ensemble des contraintes. Le principe de cette méthode a été utilisé pour les différents programmes fractionnaires linéaires et non linéaires, en variables continues comme en variables discrètes, sur des domaines bornés.
- **La résolution d'un programme équivalent** : On change la fonction objectif et l'ensemble des contraintes. Par exemple, un programme fractionnaire concave-convexe est transformé en un programme concave, et un programme hyperbolique en un programme à objectif linéaire, avec des contraintes additionnelles.

### 1.4.1 Méthode graphique

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  $\bar{Z}$ -courbe niveau quelconque de la fonction critère :

$$\bar{Z} = \frac{cx + c_0}{dx + d_0}$$

après simplification, nous obtenons :

$$\bar{Z}(dx + d_0) = cx + c_0$$

$$(c - \bar{Z}d)x = \bar{Z}d_0 - c_0$$

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  $c \neq 0$ ,  $d \neq 0$  et  $c \neq wd$  pour tout  $w \in \mathbb{R}$ ) 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.

L'ensemble rotation est appelé point de rotation dans  $\mathbb{R}^2$  et axe de rotation dans  $\mathbb{R}^3$ .

Les points de cet ensemble sont déterminés par la résolution du système suivant :

$$cx = -c_0 \quad \text{et} \quad dx = -d_0$$

Considérons le programme fractionnaire linéaire suivant :

$$(\mathbf{PF}) \begin{cases} \max Z(x) = \frac{x_1+x_2-1}{5x_1+x_2-1} \\ s.c. \quad 5x_1 + 2x_2 \geq 6 \\ x_1 \leq 3 \\ x_2 \leq 3 \\ x_1, x_2 \geq 0 \end{cases} \quad (1.17)$$

Donc la courbe de niveau  $Z$  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 :

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

$$\bar{Z} = 1 \Leftrightarrow x_1 = 0 \text{ courbe de niveau } 1.$$

Le problème à quatre points extrêmes  $x_1, x_2, x_3$  et  $x_4$  dont les valeurs du critère sont indiquées sur **la figure**.

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  $x^4$  de valeur optimale  $Z^* = 1$  est l'intersection du domaine  $S$  avec la courbe de niveau  $Z = 1$ .

Dont le graphe est donné par la figure suivante :

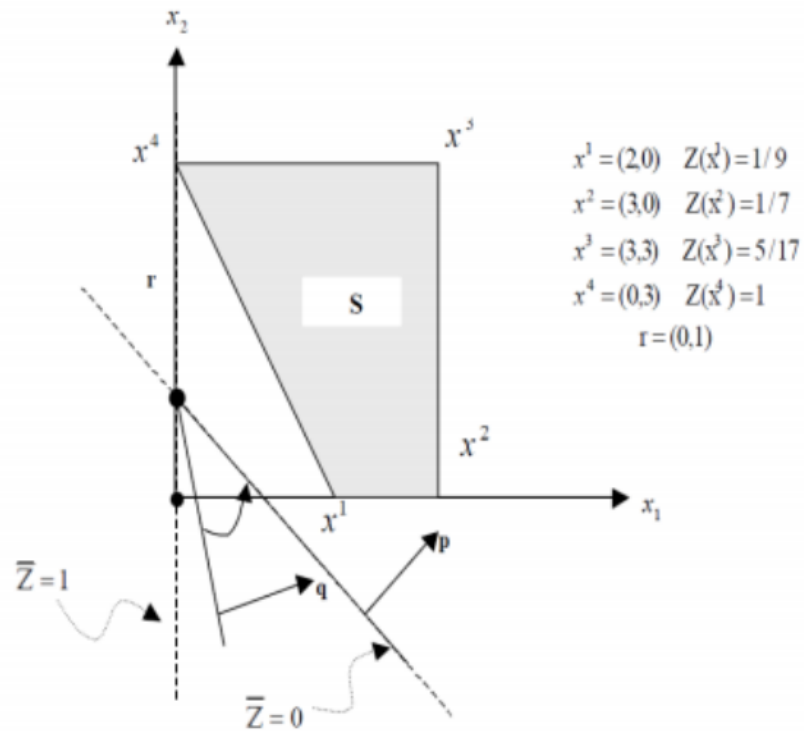


FIGURE 1.2 – Figure

### 1.4.2 Méthode de gradient

Cette méthode, ne nécessite aucun changement de variables ni l'ajout de nouvelle 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 optimal du programme

précédent [27][7] .

### 1.4.3 La transformation de charnes et cooper

En 1962, A.Charnes et W.W.Cooper ont montré que tout programme fractionnaire linéaire peut être transformé en un programme linéaire qui lui est équivalent, à condition que le domaine de solutions réalisables soit borné [10]. On introduit les nouvelles variables :

$$y_j = \frac{x_j}{D(x)}, \quad j = 1, 2, \dots, n, \quad t = \frac{1}{D(x)}$$

Où

$$D(x) = dx + d_0 \tag{1.18}$$

Avec ces nouvelles variables  $y = (y_1, y_2, \dots, y_n)$  et  $t$ , l'objectif  $Z(x)$  du programme original (1.14) peut être exprimé par :

$$L(y, t) = cy + c_0t$$

Et en multipliant l'ensemble des contraintes  $Ax \leq b$  par  $\frac{1}{D(x)}$ , on obtient les nouvelles contraintes linéaires :

$$Ay - bt \leq 0$$

Enfin, on ajoute une dernière contrainte qui exprime la relation entre les variables de décision  $x_1, x_2, \dots, x_n$  du programme originale (1.14) et les nouvelles variables  $y_1, y_2, \dots, y_n$  et  $t$ . Celle-ci est obtenue en multipliant l'égalité (1.19) par la même quantité  $\frac{1}{D(x)}$  ; la nouvelle contrainte s'écrit alors :

$$dy + d_0t = 1$$

Ainsi, pour le programme (1.14) et par la transformation de Charnes et Cooper, on obtient le programme linéaire équivalent ( avec  $n+1$  variables et  $m+1$  contraintes) suivant :

$$(P_{eq}) \begin{cases} \max L(y, t) = cy + c_0 t \\ Ay - bt \leq 0 \\ dy + d_0 t = 1 \\ y_i \geq 0, \quad i = 1, \dots, n, \quad t \geq 0 \end{cases} \quad (1.19)$$

**Lemme 1.2.** [21]

Si le vecteur  $(y_1, y_2, \dots, y_n, t)$  est une solution réalisable du problème (1.20) alors,  $t > 0$ .

**Théorème 1.7.** [21]

Si le vecteur  $(y_1^*, y_2^*, \dots, y_n^*, t^*)$  est une solution optimale du problème (1.20), alors le vecteur  $(x_1^*, x_2^*, \dots, x_n^*, t^*)$  est une solution optimale du problème (1.14), où

$$x_i^* = \frac{y_i^*}{t^*}, i = 1, \dots, n.$$

Du point de vue théorique, la relation entre les solutions optimales du problème fractionnaire linéaire et son équivalent linéaire donnée par le théorème (1.6) est utile du moment qu'elle nous permet d'utiliser quelques éléments de la théorie et des méthodes de la programmation linéaire, néanmoins cette transformation n'est pas tout à fait adéquate pour certains problèmes fractionnaires linéaires tels que les problèmes de transport ou d'affectation dont les contraintes ont une structure particulière et qui sont résolus par des méthodes spécifiques à cette structure, aussi les propriétés importantes liées à la dualité des programmes linéaires ne peuvent pas être appliquées puisque le vecteur second membre dans le programme équivalent est nul.

**1.4.4 La méthode de Dinkelbach**

Une des stratégies les plus célèbres et générales pour la résolution des programmes fractionnaires (non nécessairement linéaire) est l'approche paramétrique décrite par Dinkelbach dans [13]. 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 [13].

Soit  $\lambda \in \mathbb{R}$ , et soit le problème paramétré associé au problème fractionnaire (1.14) défini par :

$$(P(\lambda)) \begin{cases} \max Q(x) = (cx + c_0) - \lambda(dx + d_0) \\ Ax \leq b, \\ x \geq 0 \end{cases} \quad (1.20)$$

**Théorème 1.8.** [13]

Un point  $x^*$  est une solution optimale du problème fractionnaire (1.14) si et seulement si la valeur optimale de son problème paramétré associé  $(P(\lambda^*))$  est nulle, où  $\lambda^* = \frac{cx^* + c_0}{dx^* + d_0}$ .

Comme par hypothèse, on a  $(dx + d_0) > 0$ ,

$$\forall x \in \chi = \{x \in \mathbb{R}^n : Ax \leq b, \quad x \geq 0\}, \quad \frac{\delta}{\delta \lambda}((cx + c_0) - \lambda(dx + d_0)) = -(dx + d_0 < 0).$$

D'où, la fonction  $Q(x) = (cx + c_0) - \lambda(dx + d_0)$  est décroissante par rapport à  $\lambda$ .

**Algorithme**

---

**Algorithme :** Algorithme de Dinkelbach

---

**Etape 0 :** Choisir  $x^0 \in S, \lambda^1 \leftarrow Z(x^0), k \leftarrow 1$ ;

**Etape 1 :** Trouver  $x^k$  une solution optimale de  $(P(\lambda^*))$ ;

**Etape 2 :** Si la valeur optimale de  $(P(\lambda^k))$  est nulle, alors  $x^* \leftarrow x^k$ , **Terminer**;

**Etape 3 :**  $\lambda^{k+1} \leftarrow Z(x^k), k \leftarrow k + 1$ , aller à **Etape 1**;

---

#### 1.4.5 La méthode de Cambini

La méthode proposée par Cambini et al [9] utilise le concept de solution niveau optimale dont la définition est la suivante :

**Définition 1.8.** Un point  $x^*$  est une **solution niveau optimale** pour le problème (1.14) si et seulement si  $x^*$  est solution optimale du problème :

$$(P_{x^*}) \begin{cases} \max f(x) = cx + c_0 \\ Ax \leq b, \quad x \geq 0 \\ dx = dx^* \end{cases} \quad (1.21)$$

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

### Algorithme

---

**Algorithme :** Algorithme de Cambini et al.

---

**Etape 1 :**

Trouver la solution optimale niveau  $x^1$ .

Si une telle solution n'existe pas, alors  $\sup Z(x) = +\infty$ , **Terminer**.

Sinon Poser  $k \leftarrow 1$  et aller à **Etape 2**

**Etape 2 :**

Si  $J = \{j/d_j > 0\} = \emptyset$  alors  $x^k$  est une solution optimale de (1.14), **Terminer**.

Sinon, soit  $s$  tel que  $\frac{c^s}{d^s} = \max\{\frac{c_j}{d_j}\}$ ,

Si  $\gamma_s > 0$  aller à **Etape 3**

Sinon,  $x^k$  est une solution optimale de (1.14) **Terminer**.

**Etape 3 :**

La variable hors base  $x_s$  entre dans la base moyennant une opération pivot, si une telle opération existe. Poser  $k \leftarrow k + 1$  et aller à **Etape 2**

Si une telle opération n'existe pas,

$x^k$  est une solution optimale de (1.14). **Terminer**.

---

## 1.5 Programmation fractionnaire linéaire en nombres entiers

Un problème d'optimisation fractionnaire linéaire en nombres entiers sous contraintes linéaires a la forme suivante :

$$\text{FLI} \begin{cases} \max Z(x) = \frac{cx + c_0}{dx + d_0} \\ Ax \leq b, \quad x \geq 0 \\ x \in Z^n \end{cases} \quad (1.22)$$

### 1.5.1 Méthodes de résolution

#### La méthode Branch and Bound

Cette Méthode est basée sur une recherche arborescente par la résolution de problèmes fractionnaires continus, Ces derniers sont obtenus en ignorant la condition que toutes les variables soient à valeurs entières, on les appelle problèmes fractionnaires relaxés. Les étapes sont les suivantes :

1. Diviser un problème à résoudre en sous-problèmes, chaque sous-problème correspond à un noeud de l'arborescence.
2. Calculer la relaxation fractionnaire linéaire d'un sous-problème en utilisant un des algorithmes de résolution d'un programme fractionnaire linéaire en variables continues, trois cas sont possibles :
  - Le problème relaxé n'a pas de solution réalisable.
  - Le problème relaxé a une solution optimale entière. Comparez cette solution optimale avec la meilleure solution que nous connaissons jusque là.
  - Le problème relaxé a une solution optimale qui est pire que la solution courante. Dans tous les cas ci-dessus, nous savons tout ce que nous devons savoir sur ce sous-problème et le noeud associé a ce sous-problème est sondé.
3. Le problème relaxé a une solution optimale qui n'est pas entière, mais sa valeur optimale est meilleure que la meilleure solution que nous connaissons jusque là. Dans ce cas, il faudrait diviser d'avantage ce sous-problème.

La solution optimale est obtenue lorsque tous les noeuds créés sont sondés. L'exploration de l'arborescence créée se fait selon trois stratégies différentes :

1. **En largeur** : Cette stratégie favorise les noeuds les plus proches de la racine, ce qui engendre moins de séparations. Cependant les deux stratégies suivantes s'avèrent plus efficaces.
2. **En profondeur** : Dans cette stratégies, on commence par explorer le noeud le plus éloigné de la racine, c'est à dire le plus récemment créé.
3. **Le meilleur d'abord** : Cette stratégie consiste à explorer le noeud possédant la meilleur borne, ce qui permet d'éviter l'exploration des sous-problèmes qui possèdent une mauvaise évaluation par rapport à la solution optimale.

### Méthode des Coupes

Comme la méthode Branch and Bound, cette méthode consiste à ajouter itérativement au problème relaxé initial des contraintes appelées coupes de telle sorte que la solution optimale soit un point extrême. Parmi ces méthodes on peut

citer la méthode de Granot et al.[17] que nous présentons dans ce qui suit :

Dans cette méthode, on rajoute trois lignes à la matrice des contraintes  $A$  qui contient  $m$  lignes. La première ligne correspond au vecteur  $c$ , la deuxième au vecteur  $d$  et la troisième correspondra au gradient  $\gamma$  de la fonction objectif fractionnaire linéaire  $Z$ .

A chaque itération, toutes les lignes sont modifiées par des opérations de pivot sauf la dernière qui sera modifiée en utilisant la formule du gradient réduit :

$$\hat{\gamma}_i = \hat{d}_0 \hat{c}_i - \hat{c}_0 \hat{d}_i, \quad i = 1, \dots, n.$$

On calcule les composantes du gradient réduit relatives aux variables hors base de la solution courante, on vérifie :

**Si**  $\hat{\gamma}_j \leq 0, \quad \forall j \in N$ , alors la solution courante est une solution optimale du problème (1.23).

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

**Sinon** Il existe un indice  $s \in N$  tel que  $\hat{\gamma}_s > 0$ .

On calcule alors  $\mu_r$  tel que :

$$\mu_r = \frac{\hat{b}_r}{\hat{A}_{rs}} = \min_{1 \leq i \leq m} \left\{ \frac{\hat{b}_r}{\hat{A}_{rs}}, \hat{A}_{is} > 0 \right\} \quad (1.23)$$

alors la ligne  $r$  servira à générer une coupe de gomory :

$$v + \sum_{j \in n} \left| \frac{\hat{b}_i}{\hat{A}_{is}} \right| x_j = \left| \frac{\hat{b}_i}{\hat{A}_{rs}} \right|, v \geq 0. \quad (1.24)$$

Cette coupe va servir de ligne pivot et la colonne  $s$  est la colonne pivot.

Comme la valeur du pivot sera  $\frac{\hat{A}_{rs}}{\hat{A}_{rs}} = 1$ , les nouvelles valeurs des coefficients après les opérations de pivot seront toutes entières.

## Chapitre 2

# Optimisation mono-objectif fractionnaire non linéaire

### 2.1 Introduction

Nous appelons problème d'optimisation fractionnaire non linéaire un programme dont la fonction objectif est le rapport d'au moins une fonction non linéaire. Dans ce chapitre, nous présentons les définitions et les étapes nécessaires à la formulation et la résolution d'un problème d'optimisation non linéaire, nous allons citer quelques applications des modèles fractionnaires non linéaires et nous introduirons un algorithme de résolution.

### 2.2 Problème d'optimisation non linéaire

#### 2.2.1 Définition

Le problème d'optimisation non linéaire est donné sous la forme suivante :

$$\left\{ \begin{array}{l} \text{Minimiser } f(x) \\ \text{Sous les contraintes} \\ x \in X, \text{ avec, } X \subseteq \mathbb{R}^n \end{array} \right.$$

où  $f(x)$  est appelée fonction objectif ou fonction économique définie par :

$$\begin{array}{l} f : X \rightarrow \mathbb{R} \\ x \rightarrow f(x) \end{array}$$

$X$  est appelé ensemble des solutions réalisables ou admissibles, il est appelé aussi ensemble des décisions. [23]

### 2.2.2 Notions de bases

#### Définitions des solutions

Dans un problème d'optimisation non linéaire on distingue deux types de solutions :

Le minimum local et le minimum global.[23]

Soient l'ensemble  $S \in \mathbb{R}^n$  et une fonction  $f : S \rightarrow \mathbb{R}^n$ , les minimums locaux et globaux de  $f$  sur  $S$  sont définis de la manière suivante :

- **Minimum local** : Intuitivement, un vecteur  $x^* \in S$  est un minimum local de  $f$  sur  $S$  s'il a un coût plus faible que celui de ses voisins. Formellement,  $x^*$  est un minimum local de  $f$  sur  $S$  si :

$$\exists \epsilon > 0 \text{ tels que : } f(x^*) \leq f(x); \quad \forall x \in S, \text{ avec } \|x - x^*\| < \epsilon.$$

où  $\|\cdot\|$  désigne la norme.

Le minimum local est stricte si :  $f(x^*) < f(x); \quad \forall x \in S, \text{ avec } \|x - x^*\| < \epsilon.$

- **Minimum global** : Un vecteur  $x^* \in S$  est un minimum global de  $f$  sur  $S$  s'il a un coût plus faible que celui de tous les autres vecteurs dans  $S$ . Formellement,  $x^*$  est un minimum global de  $f$  sur  $S$  si :

$$f(x^*) \leq f(x); \quad \forall x \in S$$

#### Conditions d'optimalités

Avant de développer des algorithmes permettant d'identifier des solutions d'un problème d'optimisation, il faut être capable de décider si un point donné est optimal ou non[23]. On distingue deux types de conditions d'optimalité :

1. Les conditions nécessaire d'optimalité.
2. Les condition suffisantes d'optimalité.

#### L'utilité des conditions nécessaire et suffisante

[23] On s'intéresse à ces conditions pour les raisons suivantes :

- Elles permettent dans certains cas de calculer explicitement les solutions optimales.
- Elles constituent une base pour les méthodes qualitatives, c'est-à-dire les méthodes qu'étudient les propriétés des solutions du problème d'optimisation.
- Elles sont utilisées pour l'élaboration des méthodes numériques d'optimisation.

### 2.2.3 Conditions de Kuhn-Tucker

**Définition 2.1.** En mathématiques, les conditions de Karush-Kuhn-Tucker ou anciennement conditions de Kuhn-Tucker sont une généralisation des multiplicateurs de Lagrange qui permettent de résoudre des problèmes d'optimisation sous contraintes non linéaires d'inégalités [23].

**Théorème 2.1.** [18]

Soit le problème :

$$(P_k) \Leftrightarrow \begin{cases} Z(x) \rightarrow \min \\ h_i(x) \leq 0, \quad i = \overline{1, k} \end{cases}$$

1.  $Z : \mathbb{R}^n \rightarrow \mathbb{R}$  de classe  $C^1$ .
2.  $h_i : X \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$  de classe  $C^1$ .

On suppose que les contraintes sont qualifiées au point  $x^*$ .

Alors, une condition nécessaire pour que  $x^*$  soit une solution de  $P_k$ , et qu'il existe des nombres positifs  $\lambda_1, \dots, \lambda_n$  (appelés multiplicateurs de (Kuhn-Tucker) ou de Lagrange généralisés) tel que

$$\begin{cases} \nabla Z(x^*) + \sum_{i=1}^k \lambda_i \nabla h_i(x^*) = 0 \\ \lambda_i h_i(x^*) = 0, \quad i = \overline{1, k} \end{cases}$$

## 2.3 Modèles fractionnaire non linéaires

### 2.3.1 Programmation fractionnaire non linéaire

On considère deux fonctions continues  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  et  $g : \mathbb{R}^n \rightarrow \mathbb{R}$  définies sur un polyèdre  $\chi \subseteq \mathbb{R}^n$  tel que  $g(x) > 0, \quad \forall x \in \chi$ . Le problème de programmation fractionnaire est de trouver un point  $x^*$  satisfaisant

$$\max_{x \in \mathcal{X}} \left\{ \frac{f(x)}{g(x)} \right\} \quad (2.1)$$

### L'approche de Dinkelbach

[13] Une des approches bien connues et aussi des plus anciennes est de considérer le problème d'optimisation

$$\max_{x \in \mathcal{X}} f(x) - \lambda g(x), \quad \lambda \in \mathbb{R} \quad (2.2)$$

Cette approche génère une séquence de valeurs  $\lambda_i$  qui converge vers l'optimum global de la fonction objectif.

Nous allons décrire une méthode pour résoudre le problème (2.2), cette méthode combine entre l'algorithme de Dinkelbach et un algorithme qui résout un problème de programmation quadratique sous contraintes linéaires.

Notons qu'il existe un résultat fondamental qui relie les deux problèmes (2.2) et (2.3) qui est :

$x^*$  est une solution optimale du problème (2.2) si et seulement si  $x^*$  est une solution optimale du problème (2.3) avec  $\lambda^* = \frac{f(x^*)}{g(x^*)}$ .

L'algorithme itératif original de Dinkelbach basé sur ce résultat est décrit au chapitre 1.

On considère la fonction  $M$  définie par :

$$M(\lambda) = \max_{x \in \mathcal{X}} f(x) - \lambda g(x) \quad (2.3)$$

La fonction  $M(\lambda)$  possède deux propriétés intéressantes pour garantir la convergence des bornes inférieures et supérieures vers  $\lambda^*$ , la première est que pour chaque borne inférieure  $\lambda$  de  $\lambda^*$ ,  $M(\lambda)$  est positive et pour chaque borne supérieure  $\lambda$  de  $\lambda^*$ ,  $M(\lambda)$  est négative, la deuxième est que  $M(\lambda)$  est convexe. Autrement dit :

1.  $M(\lambda) > 0 \quad \forall \lambda < \lambda^*$  et  $M(\lambda) < 0 \quad \forall \lambda > \lambda^*$

2.  $M(\lambda)$  est convexe.

Pour obtenir une borne supérieure initiale, on résout les deux problèmes suivants

$$\max_{x \in \mathcal{X}} f(x) \quad (2.4)$$

Pour avoir une solution optimale  $f(x')$ , et

$$\min_{x \in \mathcal{X}} g(x). \quad (2.5)$$

Pour avoir une solution optimale  $g(x'')$ .

Il est évident que  $\gamma^{-1} = \frac{f(x')}{g(x'')}$  est bien une borne supérieure initiale pour le problème (2.4). En effet, selon la première propriété de la fonction  $M$ , toute valeur de  $\gamma \in \mathbb{R}$  satisfaisant  $M(\gamma) < 0$  est aussi une borne supérieure.

Ainsi, si on définit

$$\gamma^{(n)} = \gamma^{(n-1)} - M(\gamma^{(n-1)}) \frac{\gamma^{(n-1)} - \lambda^{(n)}}{M(\gamma^{(n-1)}) - M(\lambda^{(n)})}, \quad (2.6)$$

où  $\lambda^{(n)}$  est la borne inférieure la plus récente de  $\lambda^*$  et  $\gamma^{(n-1)}$  est la borne supérieure la plus récente de  $\lambda^*$ , alors la nouvelle borne supérieure de  $\lambda^*$  est donnée par  $\gamma^{(n)}$  dans la formule (2.9).

Ce résultat nous conduit à une modification importante de l'algorithme de Dinkelbach :

**Algorithme****Algorithme 3** : Algorithme de Dinkelbach Modifié

**Etape 1** : Choisir  $x^0 \in \chi$ ,  $\lambda^{(0)} \leftarrow \frac{f(x^0)}{g(x^0)}$  ;

**Etape 2** :

- Résoudre les problèmes (2.7) et (2.8) pour obtenir les valeurs  $f(x')$  et  $g(x'')$  respectivement, pour obtenir  $x^{k+1}$  ;

**Etape 4** : Si  $M(\lambda^k) = 0$ , alors  $x^* \leftarrow x^{k+1}$  et  $\lambda^* \leftarrow \lambda^k$ , Terminer ;

**Etape 5** : Résoudre le problème

$M(\gamma^{k-1}) = \max_{x \in \chi} f(x) - \gamma^{k-1}g(x)$  pour obtenir  $y^k$  ;

**Etape 6** : Si  $M(\gamma^{k-1}) = 0$  alors  $x^* \leftarrow y^k$  et  $\lambda^* \leftarrow \gamma^{k-1}$ , Terminer ;

**Etape 7** :  $\gamma^k \leftarrow \gamma^{k-1} - M(\gamma^{k-1}) \frac{\gamma^{k-1} - \lambda^k}{M(\gamma^{k-1}) - M(\lambda^k)}$  ;

**Etape 8** : Si  $\gamma^k - \lambda^k \leq \delta$  alors  $x^* \leftarrow x^{k+1}$  et  $\lambda^* \leftarrow \lambda^k$ , Terminer ;

**Etape 9** :

-  $\lambda^{k+1} \leftarrow \frac{f(x^{(k+1)})}{g(x^{(k+1)})}$

-  $k \leftarrow k + 1$

- aller à **Etape 3** ;

**2.3.2 Programmation fractionnaire quadratique****Définition du modèle**

Un problème d'optimisation mono-objectif fractionnaire quadratique s'écrit mathématiquement comme suit :

$$\begin{cases} \max f(x) = \frac{(cx+c_0)(dx+d_0)}{(px+p_0)(qx+q_0)} \\ s.c \\ Ax \leq b, \quad x \geq 0 \end{cases} \quad (2.7)$$

où  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ ,  $c \in \mathbb{R}^{1 \times n}$ ,  $d \in \mathbb{R}^{1 \times n}$ ,  $p \in \mathbb{R}^{1 \times n}$ ,  $q \in \mathbb{R}^{1 \times n}$ ,  $c_0$ ,  $d_0$ ,  $p_0$ ,  $q_0$  sont des constantes réelles.

Dans la suite, on supposera que le polyèdre  $\chi = \{x \in \mathbb{R}^n / Ax \leq b, x \geq 0\}$  est borné et non vide, les fonctions  $(cx + c_0)$ ,  $(dx + d_0)$ ,  $(px + p_0)$  et  $(qx + q_0)$  sont positives sur  $\mathbb{R}_+^n$ .

**Méthode de résolution algébrique** [21, 19]

Le problème fractionnaire quadratique (2.7) est résolu en remplaçant la fonction objectif  $f$  par les deux premiers terme de son développement de Taylor.

$$L(x) = f(a) + \frac{\partial f}{\partial x_1}(a)(x_1 - a_1) + \frac{\partial f}{\partial x_2}(a)(x_2 - a_2) + \cdots + \frac{\partial f}{\partial x_n}(a)(x_n - a_n) \quad (2.8)$$

$$x = (x_1, x_2, \dots, x_n), a = (a_1, a_2, \dots, a_n)$$

La résolution du problème (2.10) se réduit à résoudre une suite de programmes linéaires .

**Algorithme 4 : Méthode algébrique**

**Etape 0 :** Choisir  $\bar{x} \in \chi$ , une solution réalisable initiale et non nulle du problème (2.7) ;

**Etape 1 :** Calculer l'expression de  $L(x)$  au point  $\bar{x}$  en utilisant (2.8) ;

**Etape 2 :** Trouver  $x^*$  la solution optimale du programme linéaire :  $\max_{x \in \chi} L(x)$

**Etape 4 :** Si  $\bar{x} = x^*$ , alors  $x^*$  est la solution optimale de (2.7), **Terminer**.

**Etape 5 :** Si  $\bar{x} \neq x^*$ , alors  $\bar{x} \leftarrow x^*$ , aller à **Etape 1** ;

**Exemple 2.1.** On se propose de résoudre le programme quadratique fractionnaire suivant :

$$\begin{cases} \max f(x) = \frac{(x_2+1)(x_1+x_2+3)}{(x_1+4)(x_1+x_2+2)} \\ s.c \\ -x_1 + x_2 \leq 1 \\ x_1 + 2x_2 \leq 7 \\ x_1 \geq 0, x_2 \geq 0 \end{cases} \quad (2.9)$$

On choisit la solution réalisable non nulle  $\bar{x} = (1, 1)$ .

On calcule les deux dérivées partielles de  $f$  :

$$\frac{\partial f}{\partial x_1}(x) = \frac{(x_2 + 1)(x_1 + 4)(x_1 + x_2 + 2) - (x_2 + 1)(x_1 + x_2 + 3)[(x_1 + 4) + (x_1 + x_2 + 2)]}{(x_1 + 4)^2(x_1 + x_2 + 2)^2}$$

$$\frac{\partial f}{\partial x_2}(x) = \frac{[(x_2 + 1)(x_1 + x_2 + 3)](x_1 + 4)(x_1 + x_2 + 2) - (x_2 + 1)(x_1 + x_2 + 3)(x_1 + 4)}{(x_1 + 4)^2(x_1 + x_2 + 2)^2}$$

Au point  $\bar{x} = (1, 1)$ , on a  $f(1, 1) = \frac{1}{2}$ ,  $\frac{\partial f}{\partial x_1}(1, 1) = -\frac{1}{8}$  et  $\frac{\partial f}{\partial x_2}(1, 1) = \frac{9}{40}$ . D'où  $L(x) = -\frac{1}{8}x_1 + \frac{9}{40}x_2 + \frac{2}{5}$  on résoud le programme linéaire :

$$(PL) \begin{cases} \max L(x) = -\frac{1}{8}x_1 + \frac{9}{40}x_2 + \frac{2}{5} \\ s.c \\ -x_1 + x_2 \leq 1 \\ x_1 + 2x_2 \leq 7 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}$$

La solution optimale obtenue est  $x^* = (\frac{5}{3}, \frac{8}{3})$ . Comme elle est différente de  $\bar{x} = (1, 1)$ , on passe à une deuxième itération :

Au point  $\bar{x} = (\frac{5}{3}, \frac{8}{3})$ , on a  $f(\frac{5}{3}, \frac{8}{3}) = \frac{242}{323}$ ,  $\frac{\partial f}{\partial x_1}(\frac{5}{3}, \frac{8}{3}) = -\frac{15477}{104329} = -0.1483$  et  $\frac{\partial f}{\partial x_2}(\frac{5}{3}, \frac{8}{3}) = \frac{1155}{6137} = 0.1882$ .

D'où  $L(x) = -0.1483x_1 + 0.1882x_2 + cste$

on résoud le programme linéaire :

$$(PL) \begin{cases} \max L(x) = -0.1483x_1 + 0.1882x_2 + cste \\ s.c \\ -x_1 + x_2 \leq 1 \\ x_1 + 2x_2 \leq 7 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}$$

La solution optimale obtenue est  $x^* = (\frac{5}{3}, \frac{8}{3})$ . Comme elle coïncide avec  $\bar{x}$ , la solution  $x^* = (\frac{5}{3}, \frac{8}{3})$  est alors la solution optimale du problème (2.12) et sa valeur optimale est  $f^* = \frac{242}{323}$

## 2.4 Application

### 2.4.1 Un planning d'arrimage de porte-conteneurs

Un modèle de programmation linéaire peut être configuré pour trouver la combinaison optimale de cargaisons à charger sur un navire, en ayant pour objectif de maximiser le profit. Cependant selon les poids, les longueurs ou d'autres caractères des marchandises, certains types de conteneurs sont chargés, ou déchargés, plus lentement que les autres. Par exemple, pour maintenir la stabilité et la sécurité du navire, les conteneurs plus lourds doivent être placés

plus bas que les plus légers, ce qui entraîne des déplacements des conteneurs. l'un des coûts les plus importants pour une société de porte-conteneurs est le loyer annuel d'un poste d'amarrage.

Un plan d'arrimage est nécessaire pour améliorer l'efficacité du chargement/déchargement du navire, conformément aux règles de sécurité. Le but est de réduire le nombre de changement de quai dans les ports et sur les navires.

Pour cela, le modèle linéaire traditionnel de maximisation de la rentabilité peut ne pas être utile car il ne prend pas en compte le temps de chargement de différentes cargaisons et ne laisse donc pas passer une partie adéquate du loyer annuel du quai à différents types de conteneurs.

Une fonction objectif tenant compte du temps de chargement est proposée comme suit : [12]

$$\max \frac{\sum_i (p_i - C_1 K_i) x_i - C_2 T}{\sum_i K_i x_i + T} \quad (2.10)$$

où  $p_i$  est le profit unitaire pour la cargaison du type  $i$ ;  $C_1$  est le coût par unité de temps dans les ports;  $K_i$  le temps de chargement de la cargaison du type  $i$ ;  $C_2$  est le coût par unité de temps en mer, et  $T$  la durée totale du trajet. Étant donné  $K_i$ , il s'agit d'une programmation fractionnaire linéaire à un seul ratio. Si  $K_i$  sont aussi des variables de décision représentant des plans d'arrimage différents, cela devient un problème de programmation fractionnaire quadratique [25].

## Chapitre 3

# Résolution d'un problème fractionnaire mono-objectif linéaire via la programmation linéaire

### 3.1 Introduction

Les programmes mathématiques fractionnaires [24], sont des programmes dont l'objectif s'exprime comme le rapport de deux fonctions, linéaires ou non linéaires, en variables réelles, binaires ou mixtes, 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.

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

- Le domaine des finances où il est souvent question d'optimiser un rapport de deux fonctions telles que [dette/capitaux propres], [rendement/employé].
- Le domaine de l'économie offre un large éventail d'application. 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], [rendement/investissement], [coût/temps].
- Le domaine de la santé comme la planification dans un hôpital [coût/patient], [infirmière/patient], [docteur/patient]... etc

- Les processus de décision de Markov peuvent également mener à la maximisation du rapport [moyenne/écart type].

## 3.2 Formulation du programme fractionnaire

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

Soit  $f$ ,  $h$  et  $g_i$ ,  $i = [1, \dots, m]$ , des fonctions réelles définies sur  $\mathbb{R}^n$  dans  $\mathbb{R}$ , avec  $h$  ne s'annule pas sur un sous ensemble  $X$  de  $\mathbb{R}^n$ .

Le problème de programmation fractionnaire ( $PF$ ) consiste à déterminer l'élément  $x^* \in X$  optimisant la fonction  $\frac{f(x)}{h(x)}$  sur un domaine réalisable  $S = \{x \in X \subset \mathbb{R}^n / g_i(x) = 0, i = [1, \dots, m]\}$ ,

$$(PF) \begin{cases} \min G(x) = \frac{f(x)}{h(x)}, \\ x \in S = x \in X \subset \mathbb{R}^n / g_i(x) = 0, \quad i = 1, \dots, m, \end{cases} \quad (3.1)$$

vérifiant les hypothèses suivantes :

- Les fonctions  $f$ ,  $h$  et  $g_i$  sont continues sur  $\mathbb{R}^n$ .
- $S$  est un domaine non vide et borné de  $\mathbb{R}^n$ .
- $h(x) > 0, \quad \forall x \in S$ .

**Définition 3.1.** [24]

Le problème de minimisation ( $PF$ ) est appelé programme fractionnaire convexe-concave si  $f$  et  $g_i$  pour  $i = 1, \dots, m$ , sont convexes et  $h$  est concave.

**Lemme 3.1.** [24] Si  $f$  est convexe et  $h$  linéaire telle que  $h(x) > 0$ , pour tout  $x \in S$ , alors  $G = \frac{f}{h}$  est quasiconvexe.

**Preuve.**  $\forall \lambda \in [0, 1], \quad \forall x, y \in S$ , on a :

$$G((1 - \lambda)x + \lambda y) \leq \frac{(1 - \lambda)f(x) + \lambda f(y)}{(1 - \lambda)h(x) + \lambda h(y)}$$

$$\begin{aligned}
 &= \frac{(1 - \lambda)G(x)h(x) + \lambda G(y)h(y)}{(1 - \lambda)h(x) + \lambda h(y)} \\
 &\leq \max\{G(x); G(y)\} \frac{(1 - \lambda)h(x) + \lambda h(y)}{(1 - \lambda)h(x) + \lambda h(y)} \\
 &= \max\{G(x); G(y)\}
 \end{aligned}$$

D'où le résultat.

**Lemme 3.2.** *Si  $f$  est convexe et  $h$  est linéaire telle que  $h(x) > 0, \forall x \in S$ , et  $S$  est compact (fermé et borné), alors le problème (PF) admet au moins une solution optimale.*

### 3.3 Formulation de (PFL) en (PL)

Soit le problème de programmation fractionnaire linéaire suivant :

$$(PFL) \left\{ \begin{array}{l} \min \frac{c^t x + \alpha}{d^t x + \beta} \\ x \in P = \{x \in \mathbb{R}^n / Ax = b; x \geq 0\}, \end{array} \right.$$

où  $c, d \in \mathbb{R}^n, \alpha, \beta \in \mathbb{R}; A \in \mathbb{R}^{m \times n}$  de plein rang;  $b \in \mathbb{R}^m$  et  $x \in \mathbb{R}^n$  avec  $d^t x + \beta > 0$ .

#### 3.3.1 Formulation de Charnes et Cooper (FC)

Charnes et Cooper [9] ont considéré (PFL) en introduisant le changement de variables suivant :

$$y = tx \geq 0 \quad \text{avec} \quad t = \frac{1}{d^t x + \beta} > 0.$$

Ce qui induit aux nouvelles données suivantes :

##### 1) Transformation de l'objectif

On a

$$\frac{c^t x + \alpha}{d^t x + \beta} = (c^t x + \alpha) \frac{1}{d^t x + \beta} = (c^t x + \alpha)t.$$

On remplace  $x$  par  $\frac{y}{t}$ , on obtient un objectif linéaire équivalent.

$$\frac{c^t x + \alpha}{d^t x + \beta} = c^t y + \alpha t.$$

## 2) Transformation des contraintes

Les contraintes de (PFL) sont données par :

$$Ax = b.$$

On remplace  $x$  par  $\frac{y}{t}$ , on obtient

$$A \frac{y}{t} = b \Leftrightarrow Ay - bt = 0.$$

On a aussi

$$d^t x + \beta = \frac{1}{t} \Leftrightarrow d^t \frac{y}{t} + \beta = \frac{1}{t}$$

$$\Leftrightarrow d^t y + \beta t = 1.$$

Alors, (PFL) devient équivalent au programme linéaire suivant :

$$(PL) \begin{cases} \min (c, \alpha)^t \begin{pmatrix} y \\ t \end{pmatrix} \\ \begin{pmatrix} A & -b \\ d^t & \beta \end{pmatrix} \begin{pmatrix} y \\ t \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ (y, t) \geq 0. \end{cases}$$

**Proposition 3.1.** [10]

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

**Remarque 3.1.** L'approche de l'équivalence de Charnes et Cooper nécessite l'augmentation des contraintes et des variables. Le problème linéaire obtenu (PL) est de taille  $(m + 1, n + 1)$  contrairement au (PFL) qui est de taille  $(m, n)$ .

Dans cette partie, Nous présentons une nouvelle formulation permettant de transformer (PFL) en un programme linéaire (PL) équivalent tout en gardant la même dimension du problème initial  $(m, n)$ .

Le problème linéaire obtenu sera traité et résolu par une approche de points intérieurs.

On note que la nouvelle formulation fonctionne bien pour tout  $\beta \in \mathbb{R}$ ; contrairement à d'autres méthodes existantes dans la littérature qui échouent dans le cas  $\beta = 0$ [4].

**3.3.2 Formulation de Bennani et Benterki (FB)**

**Lemme 3.3.** [5] *Le programme fractionnaire linéaire (PFL) est équivalent au programme linéaire (PL) suivant :*

$$(PL) \begin{cases} \min c'^t x + \alpha' \\ Ax = b; \\ x \geq 0 \end{cases},$$

où

$$c' = c - z^k d \in \mathbb{R}^n, \quad z^k = \frac{c^t x^k + \alpha}{d^t x^k + \beta} \in \mathbb{R} \text{ et } \alpha' = \alpha - z^k, \quad \beta \in \mathbb{R}$$

Où  $z^k$  est une borne supérieure de la valeur optimale de l'objectif de (PFL) et  $x^k$  est une solution strictement réalisable de (PFL) i.e.,  $Ax^k = b$  et  $x^k > 0$ .

**Preuve.** Soit  $z^*$  la valeur optimale de (PFL), alors :

$$(PFL) \begin{cases} \min \frac{c^t x + \alpha}{d^t x + \beta} = z^* \\ Ax = b; \\ x \geq 0. \end{cases}$$

qui est équivalent à :

$$\begin{cases} \min c^t x + \alpha - z^*(d^t x + \beta) \\ Ax = b; \\ x \geq 0. \end{cases}$$

Ce qui nous donne

$$\begin{cases} \min(c^t - z^*d^t)x + (\alpha - z^*\beta) = 0 \\ Ax = b; \\ x \geq 0. \end{cases}$$

Puisque  $z^*$  est généralement inconnue, dans l'implémentation numérique on l'approxime à chaque itération par une borne supérieure  $z^k$ ; c'est à dire :

$$z^* \leq z^k$$

où

$$z^k = \frac{c^t x^k + \alpha}{d^t x^k + \beta}$$

avec  $x^k$  est une solution strictement réalisable connue de (PFL).

Alors, (PFL) devient équivalent au programme linéaire suivant :

$$(PL) \begin{cases} \min c'^t x + \alpha' \\ Ax = b; \\ x \geq 0. \end{cases}$$

où

$$c' = c - z^k d \quad \text{et} \quad \alpha' = \alpha - z^k \beta.$$

### 3.4 Principe générale de la méthode projective de Ye-Lustig

[20]

Le programme linéaire traité par Ye-Lustig est le suivant :

$$(PL) \begin{cases} \min c^t x = z^* \\ Ax = b \\ x \geq 0. \end{cases}$$

où  $c \in \mathbb{R}^n$ ;  $A \in \mathbb{R}^{m \times n}$  de plein rang ( $\text{rang}(A) = m < n$ ).

Dans ces méthodes, à chaque itération  $k$ , on utilise une transformation projective qui ramène la région admissible polyédrique  $\{Ax = b, x \geq 0\}$  à un simplexe

$$S_{n+1} = \left\{ x \in \mathbb{R}_+^{n+1}, \sum_{i=1}^{n+1} x_i = 1 \right\}.$$

Cette transformation est définie par :  $T_a : \mathbb{R}^n \rightarrow S_{n+1}$

$$T_a(x_k) = y = \begin{cases} y_i = \frac{\frac{x_i^k}{a_i}}{1 + \sum_{i=1}^n \frac{x_i^k}{a_i}}, & i = 1, \dots, n \\ y_{n+1} = 1 - \sum_{i=1}^n y_i, \end{cases}$$

avec  $a \in \mathbb{R}_+^n$  une solution strictement réalisable et  $T_a(a) = \frac{e_{n+1}}{n+1}$  et  $e_{n+1} = (1, \dots, 1)^t \in \mathbb{R}^{n+1}$ .

Donc, (PL) se transforme via la transformation  $T_a$  au problème linéaire suivant :

$$PL(z^*) \begin{cases} \min \begin{pmatrix} D_k c \\ -z^* \end{pmatrix}^t y \\ B_k y = 0 \\ y \in S_{n+1} = \{y \in \mathbb{R}^{n+1}, y \geq 0, \sum_{i=1}^{n+1} y_i = 1\}. \end{cases}$$

Où  $B_k = [AD_k, -b] \in \mathbb{R}^{m \times (n+1)}$   $D_k + \text{diag}(x^k)$  matrice diagonale,  $x^k$  est une solution strictement réalisable et  $z^*$  la valeur optimale de (PL).

Comme le calcul d'une solution optimale d'un programme linéaire, sur une sphère est évident, on introduit à chaque itération la plus grande sphère inscrite dans le simplexe  $S_{n+1}$ .

On obtient le sous problème de (PL) suivant :

$$PL_s(z^*) \begin{cases} \min \begin{pmatrix} D_k c \\ -z^* \end{pmatrix}^t y \\ B_k y = 0 \\ \|y - e_{n+1}\|^2 \leq r^2 < 1 \end{cases}$$

Où

- $r = \frac{1}{\sqrt{n(n+1)}}$  est le rayon de la sphère .
- $e_{n+1} = (1, \dots, 1)^t \in \mathbb{R}^{n+1}$

Comme la valeur de  $z^*$  est généralement inconnue, Ye-Lustig approxime  $z^*$  à chaque itération par des bornes supérieures  $z^k = c^t x^k \geq z^*$  .

Donc,  $PL_s(z^*)$  devient  $PL_k$  :

$$PL_k \begin{cases} \min \begin{pmatrix} D_k c \\ -z^* \end{pmatrix}^t y \\ B_k y = 0 \\ \|y - e_{n+1}\|^2 \leq r^2 < 1 \end{cases}$$

En utilisant les conditions d'optimalité de (K.K.T), la solution optimale est donnée par :

$$y^{(n+1)} = \frac{e_{n+1}}{n+1} - \alpha^k r d_k.$$

Où

- $\alpha^k$  est le pas de déplacement ;  $0 < \alpha^k < 1$
- $d_k$  est la direction telle que  $d_k = \frac{P_k}{\|P_k\|}$  ; avec  $P_k$  la projection du vecteur coût  $(D_k c - c^t x^k)^t$  de  $(PL_k)$  sur le noyau de la matrice des contraintes  $B_k$  définie par :

$$P_k = [I - B_k^t (B_k B_k^t)^{-1} B_k] \begin{pmatrix} D_k c \\ -c^t x^k \end{pmatrix}$$

On revient au problème initiale ( $PL$ ) par la transformation inverse  $T_a^{-1}$ , on obtient un candidat tel que :

$$x^{k+1} = T_a^{-1}(y^{(k+1)}) = \frac{D_k y^{(k+1)}[n]}{y_{n+1}^{(k+1)}}$$

### Calcul d'une solution réalisable initiale

Le calcul d'une solution réalisable initiale, est un problème difficile qui se manifeste dans les différentes variantes des méthodes de points intérieurs. Plusieurs alternatives sont proposées pour le régler. Dans notre cas, on s'est intéressé à la procédure de la variable artificielle.

En effet, trouver une solution strictement réalisable pour  $(PL)$  revient à résoudre le problème suivant :

$$(F) \begin{cases} Ax = b, \\ x > 0. \end{cases}$$

qui est équivalent au problème auxiliaire suivant :

$$(APL) \begin{cases} \min \lambda \\ Ax + \lambda(b - Ax^0) = b, & x^0 \in \mathbb{R}_+^n \quad (\text{arbitraire}) \\ (x, \lambda) \geq 0 \end{cases}$$

Le problème  $(APL)$  peut s'écrire aussi sous la forme :

$$(APL) \begin{cases} \min \bar{c}^t \bar{x} \\ B\bar{x} = b \\ \bar{x} \geq 0 \end{cases}$$

Où

- $\bar{c} = (0, 0, \dots, 1)^t \in \mathbb{R}^{n+1}$ .
- $B = [Ab \quad -Ax^0] \in \mathbb{R}^{(n+1) \times m}$ .
- $\bar{x} = (x, \lambda)^t \geq 0$ , *telque*  $\bar{x} \in \mathbb{R}^{n+1}$ .

Le problème  $(APL)$  possède une solution strictement réalisable triviale  $\bar{x} = (x^0, 1)^t$ ,  $x^0 \in \mathbb{R}_+^n$  (l'orthant positif), exemple  $x^0 = (1, 1, \dots, 1)^t$ .

Le problème  $(APL)$  est équivalent au problème  $(F)$ , au sens :

**Lemme 3.4.**  *$(x^*, \lambda^*)$  est une solution optimale de  $(APL)$  si et seulement si  $x^*$  est une solution de  $(F)$  ( avec  $\lambda^* \leq \varepsilon$ ,  $\varepsilon$  étant une très petite précision positive donnée). En pratique, si  $\lambda^*$  reste loin de zéro, on calcule que le problème  $(F)$  n'a pas de solution et alors  $(PL)$  est non réalisable.*

Maintenant, on résume l'algorithme de Ye-Lustig.

**Description de l'algorithme :**[6]

– **Initialisation**

$k = 0, x^0 > 0$ , strictement réalisable ( $Ax^0 = b, x^0 > 0$ ) pour  $(PL)$ ,  $\varepsilon$  est une précision donnée.

– **Tant que**  $\frac{\|P_k\|}{|c^t x^0|} > \varepsilon$  faire :

– **Etape 0** : (Construire la matrice des contraintes)

$$B_k = [A_k, -b] \text{ où } A_k = AD_k \text{ et } D_k = \text{diag}(x^k)$$

– **Etape 1** : (Calculer la projection  $P_k$ )

$$P_k = [I - B_k^t(B_k B_k^t)^{-1} B_k] \begin{matrix} D_k c \\ -c^t x^k \end{matrix} \text{ et poser } d_k = \frac{P_k}{\|P_k\|}$$

– **Etape 2** Calculer l'itéré suivant :

$$y^{k+1} = \frac{e_{n+1}}{n+1} - \alpha_k r d_k, \quad r = \frac{1}{\sqrt{n(n+1)}} \text{ avec } 0 < \alpha_k < 1$$

$\alpha_k$  est le pas de déplacement.

– **Etape 3** : (Revenir au problème initial  $(PL)$  par  $T_a^{-1}$ )

$$x^{k+1} = T_a^{-1}(y^{(k+1)}) = \frac{D_k y^{k+1}[n]}{y_{n+1}^{k+1}}, \quad k = k + 1$$

– **Fin tant que**

**Fin d'algorithme.**

### 3.5 Expérimentations numériques

Les testes numériques réalisés sont effectués sur des exemples de la littérature. La précision choisie étant  $\varepsilon = 10^{-6}$ . On résout les exemples (à taille fixe et variable) via les deux formulations équivalentes **FC** et **FB** de  $(PFL)$  afin de distinguer l'efficacité de l'une par rapport à l'autre : Dans chaque exemple, on donne la solution optimale  $x^*$  trouvée par l'algorithme de Ye-Lustig et la valeur optimale  $z^*$  ainsi que le nombre des itérations qu'on note  $k$  et le temps d'exécution nécessaire pour trouver une solution optimale qu'on note temps en secondes.

Signalons que lorsqu'il s'agit d'un programme fractionnaire linéaire à contraintes d'inégalités, on utilise des variables d'écart pour l'écrire sous sa forme standard. De même s'il s'agit d'un problème de maximisation la relation suivante permet de le convertir en un problème de minimisation :  $\max(f(x)) = -\min(-f(x))$ .

### 3.5.1 Exemple à taille fixe

**Exemple** On considère le programme fractionnaire linéaire suivant :

$$(PFL1) \begin{cases} \min \left( \frac{-2x_1 + x_2 + 2}{x_1 + 3x_2 + 4} \right) \\ -x_1 + x_2 \leq 4 \\ 2x_1 + x_2 \leq 14 \\ x_2 \leq 6 \\ x_1, x_2 \geq 0. \end{cases}$$

**Résultats obtenus via FC :**

Solution optimale trouvée après  $k = 4$  itérations

$$x^* = (6.99999, 0.00000)^t$$

La valeur optimale est  $z^* = 1,09090$ .

**Résultats obtenus via FB :**

Solution optimale trouvée après  $k = 5$  itérations

$$x^* = (7.00000, 0.00000)^t.$$

La valeur optimale est  $z^* = 1,09090$ .

**Exemple 02 [3]** On considère le programme fractionnaire linéaire suivant :

$$(PFL2) \begin{cases} \max \left( \frac{x_1 + 2x_2 + 3.5x_3 + x_4 + 1}{2x_1 + 2x_2 + 3.5x_3 + 3x_4 + 4} \right) \\ 2x_1 + x_2 + 3x_3 + 3x_4 \leq 10 \\ x_1 + 2x_2 + x_3 + x_4 \leq 14 \\ x_1, x_2, x_3, x_4 \geq 0. \end{cases}$$

**Résultats obtenus via FC :**

Solution optimale trouvée après  $k = 7$  itérations

$$x^* = (0.00001, 6.00004, 1.19984, 0.00002)^t.$$

La valeur optimale est  $z^* = 0,85713$ .

**Résultats obtenus via *FB* :**

Solution optimale trouvée après  $k = 6$  itérations

$$x^* = (0.00000, 6.39999, 1.20000, 0.00000)^t$$

La valeur optimale est  $z^* = 0,85714$

**Exemple 03** On considère le programme fractionnaire linéaire suivant :

$$(PFL3) \left\{ \begin{array}{l} \max \left( \frac{x_1 + 2x_2 + 4x_3 + 5x_4 + 8x_5}{2x_1 + 5x_2 + 3x_3 + 4x_4 + 6x_5 + 1} \right) \\ 2x_1 + x_2 + 3x_3 + x_4 + x_5 \leq 15 \\ x_1 + 2x_2 + x_5 \leq 8 \\ 3x_1 + 5x_2 + 2x_4 \leq 10 \\ 2x_2 + 4x_3 + 3x_4 + x_5 \leq 21 \\ x_1, x_2, x_3, x_4, x_5 \geq 0. \end{array} \right.$$

**Résultats obtenus via *FC* :**

Solution optimale trouvée après  $k = 7$  itérations

$$x^* = (0.00001, 0.00002, 3.66665, 0.00059, 3.99924)^t$$

La valeur optimale est  $z^* = 1,29628$ .

**Résultats obtenus via *FB* :**

Solution optimale trouvée après  $k = 5$  itérations

$$x^* = (0.00000, 0.00000, 2.33333, 0.00001, 7.99999)^t.$$

La valeur optimale est  $z^* = 1,30952$ .

**Exemple 04 [24]** On considère le programme fractionnaire linéaire suivant :

$$(PFLA) \begin{cases} \min \left( \frac{\sum_{j=1}^n c_j x_j}{\sum_{j=1}^n d_j x_j} \right) \\ \sum_{j=1}^n x_j = N \\ 0 \leq x_j \leq 1, \quad j = 1, \dots, n. \end{cases}$$

1. **n=5**

$$c^t = (2, -1, -3, 5, -2), d^t = (1, 2, 2, 3, 4), N = 3$$

**Résultats obtenus via FC :**

solution optimale trouvée après  $k = 8$  itérations

$$x^* = (0.00000, 0.99999, 0.99999, 0.00000, 0.99999)^t.$$

La valeur optimale est  $z^* = -0,74999$ .

**Résultats obtenus via FB :**

solution optimale trouvée après  $k = 5$  itérations

$$x^* = (0.00000, 1.00000, 1.00000, 0.00000, 1.00000)^t.$$

La valeur optimale est  $z^* = -0,74999$ .

2. **n=10**

$$c^t = (1, 1, 1, 2, 2, 3, 0, 0, 1, 1), d^t = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1), N = 5$$

**Résultats obtenus via FC :**

solution optimale trouvée après  $k = 5$  itérations

$$x^* = (0.59999, 0.59999, 0.59999, 0.00000, 0.00000, 0.00001, 0.99999, 0.99999, 0.59999, 0.59999)^t.$$

La valeur optimale est  $z^* = 0,60000$ .

**Résultats obtenus via FB :**

solution optimale trouvée après  $k = 5$  itérations

$$x^* = (0.60000, 0.60000, 0.60000, 0.00001, 0.00001, 0.00001, 1.00000, 1.00000, 0.60000, 0.60000)^t. \text{ La valeur optimale est } z^* = 0,60000.$$

3. **n=15**

$$c^t = (1, 2, 3, 3, 0, 0, 1, 6, 0, 4, 0, 1, 5, 2, 1),$$

$$d^t = (1, 2, 2, 4, 6, 1, 1, 0, 0, 5, 8, 2, 3, 1, 1), N = 8$$

**Résultats obtenus via FC :**

solution optimale trouvée après  $k = 5$  itérations

$$x^* = (0.99995, 0.00009, 0.00000, 0.00003, 0.99998, 0.99998, 0.99995, 0.00000, 0.99999, 0.00001, 0.99998, 0.99998, 0.00000, 0.00002, 0.99995)^t.$$

La valeur optimale est  $z^* = 0,20001$ .

**Résultats obtenus via  $FB$  :**

solution optimale trouvée après  $k = 7$  itérations

$$x^* = (0.99994, 0.00019, 0.00000, 0.00003, 1.00000, 0.99999, 0.00000, 0.99998, 0.00001, 0.99999, 0.99997, 0.00000, 0.00002, 0.99994)^t.$$

La valeur optimale est  $z^* = 0,20001$ .

Le tableau suivant résume les résultats obtenus via les 2 formulations  $FC$  et  $FB$  pour les exemples testés précédemment. On note que le temps d'exécution est négligeable pour les exemples précédents.

<i>Exemple</i>	<i>Taille(m, n)</i>	<i>k(FC)</i>	<i>k(FB)</i>
<i>PFL1</i>	(3, 2)	4	5
<i>PFL2</i>	(2, 4)	7	6
<i>PFL3</i>	(4, 5)	7	13
<i>PFL4</i>	(6, 5)	8	5
	(11, 10)	5	5
	(16, 15)	10	7

### 3.5.2 Exemples à taille variable

**Exemple 01** : On considère le programme fractionnaire linéaire suivant :

$$x_i^* = \begin{cases} 2 & i = 1, \dots, m \\ 0 & i = m + 1, \dots, n. \end{cases}$$

Le tableau suivant résume les résultats obtenus via les 2 formulations  $FC$  et  $FB$  suivant des valeurs différentes de  $m$ .

$$(PFL_m) \begin{cases} \max \frac{c^t x + 2m}{d^t x + 1}, \\ Ax = b \\ x \geq 0 \end{cases}$$

$$\text{où} \begin{cases} c_i = \begin{cases} -1 + 2m, & i = 1, \dots, m \\ 2m, & i = m + 1, \dots, n \end{cases} \\ d_i = 1, & i = 1, \dots, n \quad n = 2m \\ A[i, i] = A[i, m + i] = 1, & i = 1, \dots, m \\ b_i = 2, & i = 1, \dots, m. \end{cases}$$

La solution optimale obtenue par les deux formulations étant :

<i>Taille</i>	<b>FC</b>		<b>FB</b>	
	<i>k</i>	<i>temps</i>	<i>k</i>	<i>temps</i>
(50, 100)	3	*	3	*
(100, 200)	3	*	3	*
(200, 400)	2	1	3	*
(300, 600)	2	1	3	1
(500, 1000)	2	7	3	3
(1000, 2000)	2	52	3	29

La notion \* désigne que le temps d'exécution dans ce cas est négligeable.

### Commentaires

A travers les tests numériques et pour les différentes dimensions, les résultats montrent l'efficacité de la nouvelle formulation **FB** en particulier pour les exemples de grande taille, exprimée soit par la réduction de nombre des itérations ou le temps d'exécution.

L'introduction des méthodes de points intérieurs pour résoudre les programmes linéaires connues par son efficacité, reste préservée pour la programmation fractionnaire linéaire pour les exemples à grande taille.

## Chapitre 4

# Optimisation multi-objectif fractionnaire en nombre entiers

### 4.1 Introduction

La plupart des problèmes d'optimisation réels sont décrits à l'aide de plusieurs objectifs ou critères souvent contradictoires devant être optimisés simultanément. Alors que, pour les problèmes n'incluant qu'un seul objectif, l'optimum recherché est clairement défini, celui-ci reste à formaliser pour les problèmes d'optimisation multi-objectif. En effet, pour un problème à deux objectifs contradictoires, la solution optimale cherchée est un ensemble de points correspondant aux meilleurs compromis possibles pour résoudre notre problème. Par exemple dans le cas de deux objectifs à minimiser, toute amélioration de l'un des objectifs se fait au détriment de l'autre et que la solution optimale ou proche de l'optimum est un compromis entre les deux. Dans l'achat d'une voiture d'occasion, la voiture idéale est celle qui est peu chère (critère économique) avec peu de kilomètres (critère qualitatif), il n'est pas évident de pouvoir regrouper en un seul objectif ces deux critères non commensurables. Ainsi il n'existe plus une solution optimale unique mais un ensemble de solutions. Nous allons donc devoir identifier les meilleurs compromis possibles suivant notre budget. Les problèmes multi-objectifs ont la particularité d'être beaucoup plus difficiles à traiter que leur équivalent mono-objectif. La difficulté réside dans l'absence d'une relation d'ordre total entre les solutions. Une solution peut être meilleure qu'une autre sur certains objectifs et

moins bonne sur les autres. Donc il n'existe généralement pas une solution unique qui procure simultanément la solution optimale pour l'ensemble des objectifs. Voilà pourquoi le concept de solution optimale devient moins pertinent en optimisation multi-objectif. Dans ce cas la solution optimale ou de bonne qualité n'est plus une solution unique mais, un ensemble de solutions compromis entre les différents objectifs à optimiser. Il est vital pour identifier ces meilleurs compromis de définir une relation d'ordre entre ces éléments. La plus célèbre et la plus utilisée est la relation de dominance au sens Pareto. L'ensemble des meilleurs compromis est appelé le front Pareto, la surface de compromis ou l'ensemble des solutions efficaces. Cet ensemble de solutions constitue un équilibre, dans le sens qu'aucune amélioration ne peut être faite sur un objectif sans dégradation d'au moins un autre objectif. La solution Pareto consiste à obtenir le front de Pareto.

## 4.2 Généralités sur la programmation multi-objectif

### 4.2.1 Concepts de base et terminologie

**Définition 4.1.** [11] Un problème d'optimisation multi-objectif peut être formulé comme suit :

$$\max_{x \in \chi} \{f_1(x), f_2(x), \dots, f_r(x)\} \quad (4.1)$$

où  $r \geq 2$ ,  $r$  est le nombre de fonctions objectifs  $f_i$  à optimiser,  $x$  est le vecteur de variable de décision et  $\chi \in \mathbb{R}$  est l'espace des solutions réalisables appelé aussi **espace de décisions**.

Notons que l'**espace des critères** :  $Z = f(\chi) = \{(f_1(x), f_2(x), \dots, f_r(x)), x \in \chi\}$  est un espace de dimension  $r$ .

**Définition 4.2.** L'**image d'une solution**  $x$  notée  $z^x$  dans l'espace des critères est le point :

$$z^x = (f_1(x), f_2(x), \dots, f_r(x))^T.$$

Dans la FIGURE 4.1 sont présentées l'espace des décisions et l'espace des objectifs pour un problème à deux objectifs et deux variables.

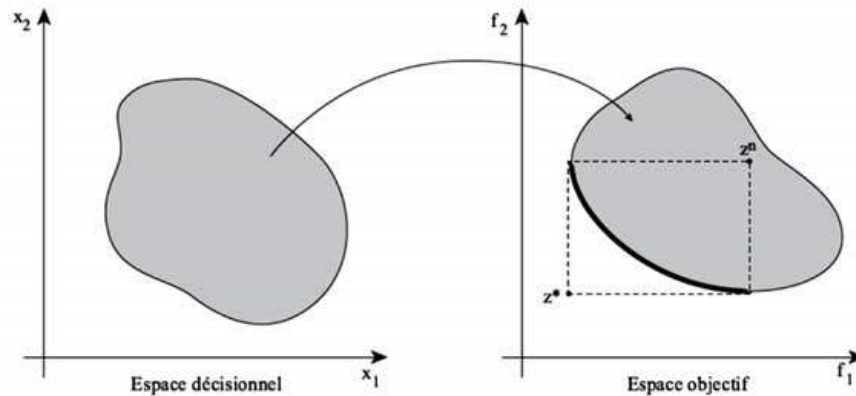


FIGURE 4.1 - L'espace des décisions et l'espace des objectifs

Comme les objectifs sont conflictuels, il n'existe pas une solution réalisable  $x$  qui soit simultanément optimale pour chacun des  $r$  objectifs, la résolution du problème (4.1) consiste à trouver l'ensemble des points efficaces, dans la section suivante nous définirons la notion de solution efficace.

#### 4.2.2 Notion de dominance

Lorsque nous avons résolu notre problème d'optimisation multi-objectif, nous avons obtenu une multitude de solutions. Seul un nombre restreint de ces solutions va nous intéresser. Pour qu'une solution soit intéressante, il faut qu'il existe une relation de dominance entre la solution considérée et les autres solutions, dans le sens suivant :

**Définition 4.3.** [11] On dit que le vecteur  $z^y$  est **faiblement dominé** s'il n'existe aucune solution  $x \in \chi$  tel que  $f_i(x) > f_i(y), \forall i \in \{1, \dots, r\}$ .

**Définition 4.4.** [11] On dit que le vecteur  $z^x$  **domine au sens de Pareto** le vecteur  $z^y$  ( $z^x \succ z^y$ ) si et seulement si  $f_i(x) \geq f_i(y), \forall i \in \{1, \dots, r\}$  et  $x \neq y$  et  $\exists 1 \leq i_0 \leq n$  tel que  $f_{i_0}(x) > f_{i_0}(y)$ .

**Définition 4.5.** [11] On dit que le vecteur  $z^x$  **domine strictement au sens de Pareto** le vecteur  $z^y$  ( $z^x \succ z^y$ ) si et seulement si  $f_i(x) > f_i(y), \forall i \in \{1, \dots, r\}$  et  $x \neq y$ .

**Définition 4.6.** [11] Une solution  $x$  domine (strictement/faiblement) une solution  $y$  au sens de Pareto si et seulement si  $z^x$  domine (strictement/faiblement) le vecteur  $z^y$  au sens de Pareto.

**Définition 4.7.** [11] Une solution  $x$  est dite **efficace** ou **Pareto optimale** s'il n'existe pas une solution réalisable qui la domine, l'image d'une solution efficace est **un point non dominé**

### 4.2.3 Relations dérivées de la dominance

Dans la définition précédente de la domination de Pareto, il n'était pas possible d'inclure une préférence pour un objectif plutôt qu'un autre. C'est pour contrer ce manque de souplesse que se développent des relations dérivées des relations de dominance. Les solutions qui permettent de trouver ces relations dérivées dominantes sont toutes optimales au sens de Pareto. La plus grande différence que nous rencontrons avec ces relations est que l'ensemble des solutions que nous obtenons grâce à ces relations est un sous-ensemble de l'ensemble des solutions obtenues grâce aux relations de dominance de Pareto [11].

#### Optimalité lexicographique

Cette définition de l'optimalité permet d'inclure une préférence entre objectifs.

Soient  $x, y \in \chi$ , on dit que  $x \succeq_{lex} y$ , s'il existe un indice  $k \in \{1, \dots, r\}$  tel que  $f_i(x) = f_i(y)$  pour  $i \in \{1, \dots, k-1\}$  et  $f_k(x) > f_k(y)$ . Les relations entre  $f_i(x)$  et  $f_i(y)$  pour  $i > k$  ne sont pas considérées puisque nous nous arrêtons à l'indice  $k$ . C'est le premier indice pour lequel on a  $f_k(x) > f_k(y)$ .

Cette définition implique que l'utilisateur ait rangé par ordre d'importance les différents objectifs. La comparaison entre les deux solutions se fera dans l'ordre de classement des objectifs.

**Définition 4.8.** Une solution  $x^* \in \chi$  est dite

**optimale au sens lexicographique**

si

$$x^* \geq_{lex} y, \quad \forall y \in \chi \setminus \{x^*\}$$

**Optimalité extrême**

Comme pour la relation d'optimalité lexicographique. Cette relation permet d'établir une préférence entre critères. Cette préférence est établie en utilisant des poids. Plus un objectif sera important. Plus son poids sera élevé [14].

**Définition 4.9.** Une solution  $x^* \in \chi$  est dite **extrême-optimale** si, étant donné un vecteur de poids  $\lambda \in \mathbb{R}^r$  tel que  $\sum_{i=1}^r \lambda^i = 1$ ,  $x^*$  est une solution optimale du problème de maximisation monocritère

$$\max_{x \in \chi} \sum_{i=1}^r \lambda^i f_i(x)$$

Donc

$$\sum_{i=1}^r \lambda^i f_i(x^*) \geq \sum_{i=1}^r \lambda^i f_i(x), \quad \forall x \in \chi \setminus \{x^*\}$$

**La dominance au sens de Geoffrion**

[14, 22]

Une dernière forme de dominance importante dans le monde de l'optimisation multi-objectif est la dominance au sens de Geoffrion. Les solutions optimales obtenues par ce type de dominance sont appelées les solutions Pareto optimales propres.

**Définition 4.10. La dominance au sens de Geoffrion** [16] Une solution  $x^* \in \chi$  est appelée solution Pareto optimale propre si :

- a. elle est Pareto optimale,
- b. il existe un nombre  $M > 0$  tel que  $\forall i \in \{1, \dots, r\}$  et  $\forall x \in \chi$  vérifiant  $f_i(x^*) > f_i(x)$ , il existe un indice  $j$  tel que  $f_j(x^*) > f_j(x) \frac{f_i(x) - f_i(x^*)}{f_j(x) - f_j(x^*)} \leq M$

Cette relation n'est quasiment jamais utilisée telle quelle. En général, on utilise plutôt un résultat qui découle de cette définition. En effet, l'interprétation de ce théorème faite dans [15] est la suivante :

”Les solutions Pareto optimales propres ont des compromis bornés suivant leurs objectifs.”

un théorème relatif à la méthode de pondération des fonctions objectifs utilisant ce résultat est le suivant :

**Théorème 4.1.** *Soit la méthode d'agrégation des fonctions objectifs suivantes :*

$$F(x) = \sum_{i=1}^r \lambda^i f_i(x)$$

*Supposons que  $\forall i \in \{1, \dots, r\}, \lambda^i > 0$  et  $\sum_{i=1}^r \lambda^i = 1$*

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

**Définition 4.11.** Deux solutions  $x$  et  $y$  sont dites **Pareto équivalentes** si l'une ne domine pas l'autre.

#### 4.2.4 Front de Pareto

**Définition 4.12.** L'ensemble de toutes les solutions efficaces du problème 3.1 est appelé **ensemble efficace** de 3.1, il est noté  $\chi_E$ .

**Définition 4.13.** L'image de l'ensemble des solutions efficaces noté  $\mathcal{Z}_E$  est le **front Pareto optimal** et aussi **surface de compromis**, Ces solutions sont Pareto équivalentes. Autrement dit, on a :

$$\mathcal{Z}_E = f(\chi_E) = \{(f_1(x), f_2(x), \dots, f_r(x)), \quad x \in \chi_E\}$$

#### Les bornes du front Pareto

On définit la notion de point idéal, point anti-idéal et point Nadir. C'est points sont utilisés comme des points de références dans les méthodes de résolution, dont le but est de trouver une solution la plus satisfaisante pour le décideur.

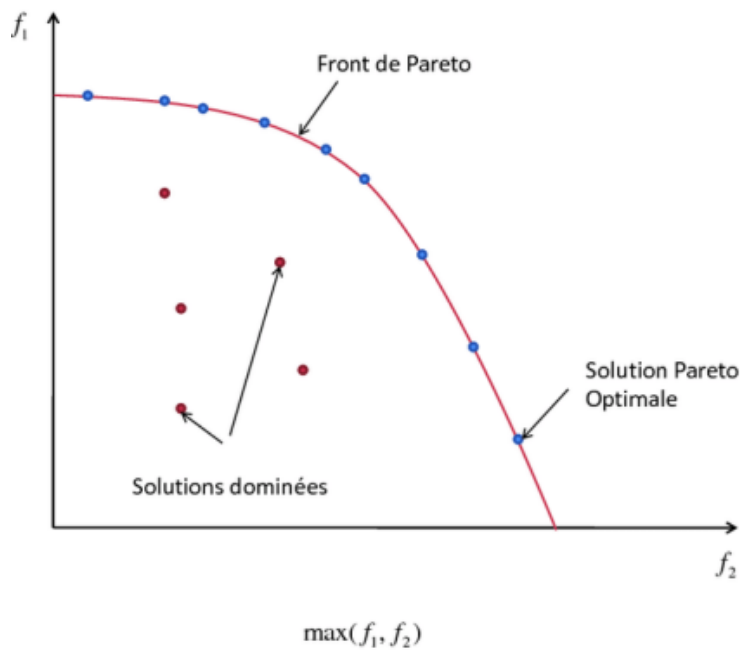


FIGURE 4.1 – Figure 4.2- Front Pareto

**Définition 4.14. (Le point idéal)**

Le vecteur idéal  $z^* = (z_1^*, z_2^*, \dots, z_r^*)$  est le vecteur qui optimise chacune des fonctions objectives  $f_i$ , i.e. :  $z_i^* = \min(f_i(x))$ ,  $x \in S$

**Définition 4.15. Le point anti-idéal**

Le vecteur  $z^* = (z_1^*, z_2^*, \dots, z_r^*)$  défini par :

$$z_i^* = \max(f_i(x)), \quad x \in S \text{ est le point anti-idéal.}$$

**Définition 4.16. Le point Nadir**

est un point  $z^N = (z_1^N, z_2^N, \dots, z_r^N)$  construit à partir des pires valeurs que peut prendre chacune des fonctions objectifs dans l'ensemble des points du front de Pareto.

**Définition 4.17. La matrice des gains**

Soit  $x_i^*$  du problème mono-objectif où seule la fonction  $f_i$  est considérée comme objectif. La matrice de dimension  $(r \times r)$  formée des éléments de  $z_{ki} = f_k(x_i^*)$  est

appelée matrice des gains.

$$\begin{pmatrix} z_1^* & \cdots & z_{1i} & \cdots & z_{1r} \\ \vdots & \ddots & \vdots & & \vdots \\ z_{i1} & \cdots & z_i^* & \cdots & z_{ir} \\ \vdots & & \vdots & \ddots & \vdots \\ z_{r1} & \cdots & z_{ri} & \cdots & z_r^* \end{pmatrix} \quad (4.2)$$

Les composantes du point idéal apparaissent sur la diagonale de cette matrice. La matrice des gains est univoque quand chaque solution optimale  $x_i^*$  trouvée est unique.

#### 4.2.5 Les méthodes de résolutions des problèmes d'optimisation multi-objectif

On constate un nombre important de méthodes classées en cinq groupes :

- Les méthodes scalaires.
- Les méthodes interactives.
- Les méthodes floues.
- Les méthodes exploitant une métaheuristique.
- Les méthodes d'aide à la décision.

Ces méthodes peuvent être regroupées en trois familles de méthodes d'optimisation multi-objectif :

- **Les méthodes à préférence à priori** : Dans ces méthodes, le décideur exprime ses préférences avant l'exécution de la méthode ainsi la méthode est exécutée une seule fois pour trouver la solution recherchée. Cependant le décideur peut ne pas être satisfait de la solution obtenue il faut donc relancer la recherche avec un ou plusieurs autres paramètres de préférences du décideur.

- **Les méthodes à préférence progressive** : Le décideur intervient dans le processus de recherche des solutions efficaces et ceci en exprimant ses préférences a fin d'orienter la recherche de la sol de façon interactive.
- **Les méthodes à préférence à posteriori** : Génèrent d'abord l'ensemble des solutions efficaces (ou une partie) qui seront ensuite présenter au décideur pour choisir sa solution préférée parmi les inconvénients de ces méthodes et que le processus de recherche des solutions efficaces nécessitent un temps considérable ajouter à ça le nombre de solutions efficaces qui peut être très grand ce qui rend très difficile le choix du meilleur compromis.

**Les méthodes scalaires** C'est une classer où le problème est transformé en un problème mono-objectif, on cite :

**a. La méthode de pondération des fonctions objectifs :**

Cette approche de résolution est la plus évidente appelée aussi méthode "l'approche naïve" de l'optimisation multi- objectif. Le but est de revenir à un problème d'optimisation mono-objectif, dont il existe de nombreuses méthodes de résolution. Elle consiste à associer à chacune des fonctions objectifs, un coefficient de pondération relatif et à faire la somme des fonctions objectifs pondérées. On aura alors une nouvelle fonction objectif. Le problème original (4.1) est transformé en :

$$\max_{x \in \chi} \sum_{i=1}^r \lambda_i f_i(x) \quad (4.3)$$

A chaque fois qu'on change les valeurs des coefficients de pondération, les solutions optimales de (4.3) pour de différentes valeurs des coefficients de pondération forment la surface de compromis. Cette méthode est applicable quand l'ensemble  $\chi$  est convexe.

Dans le cas contraire, il existe des solutions efficaces qu'on ne pourra pas prouver pour aucune valeur de  $\lambda = (\lambda_i)$ , par conséquent, on n'obtient pas la totalité de la surface de compromis.

**b. La méthode de Keeney-Raiffa :**

Dans cette méthode, on remplace les fonctions objectif par leurs produits pour

se ramener à un problème d'optimisation mono-objectif. L'approche utilisée ici est semblable à celle utilisée dans la méthode de pondération des fonctions objectif. La fonction objectif ainsi obtenue s'appelle la fonction d'utilité de Keeney-Raiffa.

### c. La méthode du compromis : ( $\varepsilon$ - contrainte)

Cette méthode permet de transformer le problème d'optimisation multi-objectif 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.

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 reformuler de la manière suivante :

$$\left\{ \begin{array}{l} \max[f_i(x)] \\ \text{tq :} \\ f_1(x) \leq \varepsilon_1 \\ \vdots \\ f_{i-1}(x) \leq \varepsilon_{i-1} \\ f_{i+1}(x) \leq \varepsilon_{i+1} \\ \vdots \\ f_m(x) \leq \varepsilon_m \\ \text{et que } g(x) \leq 0 \\ \text{avec, } x \in \mathbb{R}^n, \quad g(x) \in \mathbb{R}^q \end{array} \right.$$

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.

### Les méthodes interactives

Les méthodes interactives permettent de chercher une et une seule solution efficace du problème multi-objectif. Elles forment la famille des méthodes progressives et permettent à l'utilisateur de déterminer ses préférences vis-à-vis d'un compromis

entre objectifs au cours de l'optimisation.

#### 4.2.6 Programmation multi-objectif en nombres entiers

La programmation multi-objectif mixte est très utile pour de nombreux domaines d'application tels que la communication, les transports et la localisation.

Les variables entières sont nécessaires dans un modèle du monde réel chaque fois que l'on cherche à incorporer des phénomènes discrets ; par exemple, les choix d'investissement, les niveaux de production, les charges fixes, les conditions logiques ou les contraintes disjonctives.

La plupart des travaux de recherche sur l'optimisation multi-objectif ont jusqu'à présent été consacrés à la programmation linéaire en variables continues (MOLP). L'introduction de phénomènes discrets dans les modèles multi-objectifs conduit à des problèmes dont les variables sont totalement ou partiellement des nombres entiers.

Ces problèmes sont plus difficiles à résoudre et ne peuvent pas être traités par la plupart des approches (MOLP) car l'ensemble réalisable n'est plus convexe. En outre, il existe des approches multi-objectifs conçues pour les problèmes relatifs aux nombres entiers qui ne s'appliquent pas au cas entier mixte.

Comme le domaine des solutions réalisables n'est plus convexe, des solutions non dominées non supportées peuvent exister. Une solution non dominée est non supportée si elle est dominée par une combinaison linéaire d'autres vecteurs critères non dominés.

Contrairement aux programmes multi-objectifs linéaires continus (MOLP), l'ensemble efficace d'un programme multi-objectif en nombres entiers ou mixtes ne peut pas être entièrement déterminé par la résolution de problèmes mono-objectifs paramétrés (dont la fonction objectif est la somme pondérée).

De nouvelles caractérisations des solutions non dominées ont été établies pour la programmation multi-objectif discrètes, en particulier dans le cas linéaire. Il existe deux caractérisations principales de l'ensemble efficace. L'une consiste à introduire des contraintes supplémentaires dans le programme mono-objectif de la somme pondérée. Généralement, ces contraintes imposent des bornes aux valeurs de la fonction objectif. Cette forme de caractérisation peut être considérée comme un cas particulier d'une caractérisation générale définie par R.M. Soland [26]. La deuxième est basée sur la norme de Tchebycheff, son fondement théorique provient de V.J. Bowman [8].

Bien que ces caractérisations fournissent des résultats théoriques importants, elles ne constituent pas un moyen explicite de l'aide à la décision. Les recherches ne sont plutôt intéressés à la résolution de ces problèmes. C'est à dire trouver l'ensemble efficace correspondant.

### **4.3 Programme multi-objectif fractionnaire linéaire en nombres entiers**

La programmation fractionnaire linéaire a été étudiée par de nombreux auteurs, beaucoup plus dans le cas continu que le cas discret. Dans le cas multi-objectif en variables continues, de nouveaux travaux de recherches basés sur les méthodes floues sont apparus. D'autres approches sont proposées pour la programmation multi-objectif fractionnaire linéaire en nombres entiers [2] .

#### **4.3.1 Définition du modèle**

Mathématiquement, un problème d'optimisation multi-objectif linéaire fractionnaire en nombres entiers s'écrit comme suit :

$$(MOILFP) \left\{ \begin{array}{l} \max f_1(x) = \frac{c^1 x + c_0^1}{d^1 x + d_0^1} \\ \max f_2(x) = \frac{c^2 x + c_0^2}{d^2 x + d_0^2} \\ \vdots \\ \max f_r(x) = \frac{c^r x + c_0^r}{d^r x + d_0^r} \\ s.c \\ Ax \leq b, \quad x \geq 0, \quad x \text{ entier} \end{array} \right. \quad (4.4)$$

où le nombre d'objectifs est  $r \geq 2$ ,  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ , et pour chaque fonction objectif  $f_i$ ,  $i \in \{1, 2, \dots, r\}$ , on a  $c^i = (c_1^i, c_2^i, \dots, c_n^i) \in \mathbb{R}^{1 \times n}$ ,  $d^i = (d_1^i, d_2^i, \dots, d_n^i) \in \mathbb{R}^{1 \times n}$ ,  $c_0^i, d_0^i$  sont des constantes réelles.

Dans la suite, on supposera que le polyèdre  $\chi = \{x \in \mathbb{R}^n / Ax \leq b, \quad x \geq 0\}$  est borné et non vide, la fonction  $(d^i x + d_0^i)$ , est positive sur  $\chi$  pour chaque  $i \in \{1, \dots, r\}$ .

On supposera aussi que l'ensemble des solutions réalisables  $D = \chi \cap \mathbb{Z}^n$  est non vide,  $\mathbb{Z}$  est l'ensemble des entiers relatifs.

### 4.3.2 Approches de résolution

Les suggestions de solution de MOLPF sont fréquemment rapporté dans la littérature, en raison de leur structure particulière. Cependant, les difficultés résultant à la fois de la multi-objectivité du problème d'optimisation et la non-linéarité des objectifs conduisent généralement les chercheurs à élaborer différents approches de résolutions.

Certaines des méthodes actuelles ne suggèrent qu'un seul point, et certaines d'entre elles connaissances initiales ne sont pas satisfaisantes pour le décideur.

### 4.3.3 Méthode de Abbas et Moulai

[2] La méthode proposée par les auteurs pour résoudre le problème (MOILFP) est une généralisation au cas fractionnaire de l'algorithme de résolution d'un programme linéaire en nombre entiers à objectifs multiples (MOLP), détaillé dans [1].

Afin de résoudre le problème (4.4) une approche consistant à discrète est présentée. Considérons le problème de programmation linéaire fractionnaire entier mono-objectif donné sous la forme suivante.

$$\begin{cases} \text{Maximiser} & f_1(x) = \frac{c_1x+c_0^1}{d_1x+d_0^1} \\ \text{sc} & Ax \leq b, \quad x \geq 0 \\ & x \text{ vecteur entier} \end{cases} \quad (4.5)$$

La recherche des solutions réalisables entières de problème (4.5) nécessite l'introduction des notations suivantes ;

- $\chi_1 = \{x \in \mathbb{R}^{n_1}, \quad A_1x \leq b_1, \quad A_1 \in \mathbb{R}^{m_1 \times n_1}, \quad b_1 \in \mathbb{R}^{m_1}, \quad x \geq 0\}$ .  $\chi_1$  est la région tronquée courante de  $\chi$  obtenue par des coupes de Gomory successives :
- $x_1^1 = (x_{1,j}^1)$  est la solution optimale entière donnant  $f_1^1$  obtenue sur  $\chi_1$  ;
- $B_1^1$  est une base de  $\chi_1$  ;
- $a_{1,j}^1 \in \mathbb{R}^{m_1 \times 1}$  sont les vecteurs activités de  $(x_{1,j}^1)$  appropriés à la région tronquée  $\chi_1$  ;
- $y_{1,j}^1 = (y_{1,ij}^1) = (B_1^1)^{-1}a_{1,j}^1$  où  $y_{1,j}^1 \in \mathbb{R}^{m_1 \times 1}$  ;
- $I_1 = \{j : a_{1,j}^1 \in B_1^1\}$  ; et  $N_1 = \{j : a_{1,j}^1 \notin B_1^1\}$  ;
- $c_j^1$  est la jème composante du vecteur  $c^1$ ,  $d_j^1$  est la jème composante du vecteur  $d^1$  ;
- $c_{1,1}^1 = \sum_{i \in I_1} c_i^1 y_{1,ij}^1$  et  $d_{1,1}^1 = \sum_{i \in I_1} d_i^1 y_{1,ij}^1$  ;
- $f_1(x_1^1) = \frac{f_{1,1}^1}{f_{1,2}^1}$  où  $f_{1,1}^1 = c^1 x_1^1 + c_0^1$  et  $f_{1,2}^1 = d^1 x_1^1 + d_0^1$  ;
- $\bar{\gamma}_{1,j}^1 = f_{1,2}^1(c_j^1 - c_{1,j}^1) - f_{1,1}^1(d_j^1 - d_{1,j}^1)$  le coût réduit relatif à la jème composante du vecteur gradient réduit  $\bar{\gamma}_1^1$  ;
- $\Gamma_1 = \{j; \quad j \in N_1 \text{ et } \bar{\gamma}_{1,j}^1 = 0\}$ .
- Pour  $k \geq 2$  :
- $\chi_k = \{x \in \mathbb{R}^{n_k}, \quad A_kx \leq b_k, \quad A_k \in \mathbb{R}^{m_k \times n_k}, \quad b_k \in \mathbb{R}^{m_k}, \quad x \geq 0\}$ .  $\chi_k$  est la région tronquée courante de  $\chi$  obtenue par application de la coupe  $\sum_{j \in N_{k-1}} \{j_{k-1}\} x_j \geq 1$  où  $j_{k-1} \in \Gamma_{k-1}$  et par des coupes de Gomory successives éventuellement ;

- $x_k^1 = (x_{k,j}^1)$  est la kème solution optimale entière du problème (4.5) obtenue par  $\chi_k$  à l'étape ( $k$ );
- $B_k^1$  est une base de  $\chi_k$ ;
- $a_{k,j}^1 \in \mathbb{R}^{m_k \times 1}$  sont les vecteurs activités de  $(x_{k,j}^1)$  qui correspondent à la région tronquée courante  $\chi_k$ ;
- $y_{k,j}^1 = (y_{k,i,j}^1) = (B_k^1)^{-1} a_{k,j}^1$  où  $y_{k,j}^1 \in \mathbb{R}^{m_k \times 1}$ ;
- $I_k = \{j : a_{k,j}^1 \in B_k^1\}$ ; et  $N_k = \{j : a_{k,j}^1 \notin B_k^1\}$ ;
- $c_j^1$  est la jème composante du vecteur  $c^1$ ,  $d_j^1$  est la jème composante du vecteur  $d^1$ ;
- $c_{k,j}^1 = \sum_{i \in I_k} c_i^1 y_{k,i,j}^1$  et  $d_{k,j}^1 = \sum_{i \in I_k} d_i^1 y_{k,i,j}^1$ ;
- $f_1(x_k^1) = \frac{f_{k,1}^1}{f_{k,2}^1}$  où  $f_{1,1}^1 = c^1 x_k^1 + c_0^1$  et  $f_{1,2}^1 = d^1 x_k^1 + d_0^1$ ;
- $\bar{\gamma}_{k,j}^1 = f_{k,2}^1 (c_j^1 - c_{k,j}^1) - f_{k,1}^1 (d_j^1 - d_{k,j}^1)$  le coût réduit relatif à la jème composante du vecteur gradient réduit  $\bar{\gamma}_k^1$ ;
- $x_k^\mu = (x_{k,j}^\mu)$  sont les  $t_k - 1$  solutions entières adjacentes à  $x_k^1$ , lorsqu'elles existent, où  $t_k$  est un nombre entier et  $\mu \in \{2, 3, \dots, t_k\}$ ;
- $\Gamma_k = \{j; \quad j \in N_k \quad \text{et} \quad \bar{\gamma}_{k,j}^1 = 0\}$ .

**Théorème 4.2.** *Le point  $x_k^1$  de  $\chi$  est une solution optimale du problème fractionnaire (4.9) si et seulement si le vecteur gradient réduit  $\bar{\gamma}_k^1$  est tel que  $\bar{\gamma}_{k,j}^1 \leq 0$  pour tout indice  $j \in N_k$ .*

**Corollaire 4.1.** *Une solution optimale  $x_k^1$  du problème (4.5) est unique si seulement si le vecteur gradient réduit  $\bar{\gamma}_k^1$  est tel que  $\bar{\gamma}_{k,j}^1 < 0$  pour tout indice  $j \in N_k$ .*

**Définition 4.18.** Soit  $x_k^1$  une solution optimale du problème (4.5). Supposons que  $j_k \in N_k$ , une arrête  $E_{j_k}$  incidente à la solution  $x_k^1$  est définie par l'ensemble :

$$E_{j_k} = \left\{ x = x_i \in \chi_k : \begin{cases} x_i = x_i^k - \theta_{j_k} y_{k,i,j_k}^k & \forall i \in I_k \\ x_{j_k} = \theta_{j_k} \\ x_v = 0 & \forall v \in N_k \setminus \{j_k\} \end{cases} \right. \quad (4.6)$$

où  $0 \leq \theta_{j_k} \leq \min_{i \in I_k} \left\{ \frac{x_i^k}{y_{i,j_k}^k} \mid y_{i,j_k}^k > 0 \right\}$ ,  $\theta_{j_k}$  est un entier positif et  $\theta_{j_k} y_{k,i,j_k}^k$  est un entier pour tout  $i \in I_k$ , si de telles valeurs de  $\theta_{j_k}$  existent.

**Remarque 4.1.** Il peut y arriver que la solution optimale  $x^0$  du problème (4.5) ne soit pas unique, dans ce cas il existe une autre solution réalisable  $x^1 \neq x^0$  avec  $f_1(x^1) = f_1(x^0)$ . On dit alors que  $x^1$  est une solution optimale alternative de  $x^0$ . La relation (4.6) précédente permet de déterminer cette solution alternative.

**Algorithme de la méthode de Abbas et Moulaï** Pour résoudre un problème linéaire fractionnaire en nombres entiers à objectifs multiples (4.4), une procédure basée sur une technique de coupe plane est présentée dans les étapes suivantes :

**Etape (1) :**

Résoudre le problème fractionnaire (4.6) par n'importe quelle méthode directe de la programmation fractionnaire discrète. Soit  $x_1^1$  sa solution optimale entière sur  $\chi_1$ , construire l'ensemble  $\Gamma_1$ .

**Etape (2) :**

- Si  $\bar{\gamma}_{1,j}^1 < 0$  pour  $\forall j \in N_1$ , alors  $x_1^1$  est l'unique solution optimale sur  $\chi_1$ . Enregistrer le premier vecteur non dominé par  $(f_1(x_1^1), \dots, f_1(x_1^1))$  pour construire l'ensemble des vecteurs non dominés  $Ef f_0$ . 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 fractionnaire relative à la programmation fractionnaire, on obtient une solution réalisable entière  $x_2^1 = (x_{2,j}^1)$  dans la région tronquée  $\chi_2$ . Rajouter le vecteur correspondant  $(f_1(x_2^1), \dots, f_1(x_2^1))$  à  $Ef f_0$  s'il n'est pas dominé par l'un des précédents vecteurs critères non dominés. Enregistrer l'ensemble  $Ef f_1$ .

- Sinon, il existe un indice  $j_1 \in N_1$  pour lequel  $\bar{\gamma}_{1,j_1}^1 = 0$ . Déterminer dans ce cas toutes les solutions qui lui sont alternatives, éliminer celles qui ne sont pas efficaces et mettre à jour l'ensemble  $Ef f_0$ . Appliquer la coupe  $\sum_{j \in N_1} \{j_1\} x_j \geq 1$  pour tronquer l'arête  $E_{j_1}$ .

**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 alternatives à la solution courante  $x_{k-1}^1$ .

Augmenter l'ensemble  $Eff_{k-2}$  par des vecteurs critères non dominés correspondants pour construire  $Eff_{k-1}$ .

Tronquer l'arête  $Eff_{k-1}$  par la coupe  $\sum_{j \in N_1} x_j \geq 1$

**Etape finale :**

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

# Conclusion générale et perspectives

Dans la plupart des problèmes pratiques, il ne s'agit pas d'optimiser seulement un seul critère mais plutôt d'optimiser simultanément plusieurs critères et qui sont généralement conflictuels. Dans les problèmes de conception, par exemple, il faut le plus souvent trouver un compromis entre des besoins technologiques et des objectifs de coût. L'optimisation multi-objective consiste donc à optimiser simultanément plusieurs fonctions. La notion de solution optimale unique dans l'optimisation mono-objective disparaît pour les problèmes d'optimisation multi-objective au profit de la notion d'ensemble de solution Pareto optimales.

Nous avons présenté dans ce mémoire les travaux de recherche sur les méthodes d'optimisation mono et multi objectif fractionnaire linéaire et non linéaire.

Dans notre étude, nous nous sommes particulièrement intéressés aux problème de programmation multi-objectif fractionnaire linéaire en nombres entiers pour lequel peu de travaux ont été réalisés. Le travail présenté se compose de trois volets.

- ◇ Le premier volet consiste à rappeler les notions de bases et les méthodes de résolution des problèmes d'optimisation mono-objectif fractionnaire linéaire et non linéaire.
- ◇ Le deuxième volet consiste à résoudre le problème fractionnaire mono-objectif linéaire (PFL) via la programmation linéaire (PL).

◇ Dans le dernier volet nous nous sommes focalisées sur l'optimisation multi-objectif fractionnaire en nombres entiers en citant les concepts de base et les nouvelles méthodes de résolution.

# Bibliographie

- [1] Abbas, M., Moulai M., "Solving multiple objective integer linear programming problem", Journal of the Italian Operations Research Society, (ricerca operativa), Vol. 29, pp. 15-38, (1999).
- [2] Abbas, M., Moulai, M., "Integer linear fractional programming with multiple objective", Journal of the Italian Operation Research Society, Vol. 32.103-104, pp. 15-38, (2002).
- [3] M. Babul Hasan, S. Acharjee : Solving LFP by converting it into a single LP. International Journal of Operations Research 8, pp.1-14, (2011).
- [4] Bazaraa, M.S., Sherali, H.D., Shetti, C.M., Nonlinear Programming Third edition. New Jersey : Wiley, 853p. (2006).
- [5] Bennani, A., Benterki, D., : Efficient projective algorithm for linear fractional programming problem based on a linear programming formulation. International Journal of Computing Science and Mathematics, (à paraître).
- [6] Bennani, A., Benterki, D., Grar, h : adaptive projection methods for linear fractional programming. RAIRO Operations Research. 55p, (2021).
- [7] Bitran, G.R., Novaes, A.J., "Linear Programming with a fractional objective function", Operations Research, Vol.21, pp.22-29, (1973).
- [8] Bowman, V.J., " On the relationship of the Tchebycheff norm and the efficient frontier of multiple-criteria objectives", In : Thiriez, H, Zionts, S (eds) Multiple Criteria Decision Making. Lecture notes Economics and Math Systems. Springer, Berlin, pp. 76-86, (1976).
- [9] Cambini, A., Martein, L., A modified version of Martos's algorithm for the linear fractional problem., Methods of Oper., Res., Vol. 53, pp.33-44, (1986).
- [10] Charnes, A., Cooper, W.W., Programming with Linear Fractional Functionals, Naval., Res., Logistics Quart., Vol. 9.3-4, pp.181-186, (1962).

- 
- [11] Collette, T., Siarry, P., *Optimisation Multiobjectif*, Paris : Eyrolles, 294p, (2002).
- [12] Craven, B.D., *Fractional programming*, In : *Sigma series in applied mathematics*, Vol.4. Berlin : Heldermann Verlag, <https://doi.org/10.1002/zamm.19890691019>, (1988).
- [13] Dinkelbach, W., *On nonlinear fractional programming*, *Management Science*, Vol. 13, pp.492-498, (1967).
- [14] Ehrgott, M., "A characterization of Lexicographic Max-ordering Solutions", *Methods of Multicriteria Decision Theory : Proceedings of the 6th Workshop of the DGOR Working Group Multicriteria and Decision Theory*, Egelsbach, Häsel-hohenhausen, pp. 193-202, (1997).
- [15] Ehrgott, M., *Multicriteria Optimization*, *Lecture Notes in Economics and Mathematical Systems*, Vol. 491, Springer Verlag. doi : 10.1007/978-3-662-22199-0, (2000).
- [16] Geoffrion, A.M., "Proper efficiency and the theory of vector maximization", *Journal of Mathematical Analysis and Applications*, Vol. 22, pp. 618-630, [https://doi.org/10.1016/0022-247X\(68\)90201-1](https://doi.org/10.1016/0022-247X(68)90201-1), (1968).
- [17] Granot, J., Granot, F., *On integer and mixed integer fractional programming problems*, *Ann. Discrete Math*, Vol. 1, pp.221-231, (1977).
- [18] Leslous, F., "Problème d'optimisation non convexe et optimisation DC", *Thèse doctorat*, pp.21-22, (2023).
- [19] Leslous, F., Goubi, M., Ouanes, M., *A new approach for non-convex optimisation problems applied to Hump and benchmark functions*; *Int. J. Mathematics in Operational Research*, Vol. 25, No. 3, (2023).
- [20] Lustig, I., J., *A practical approach to Karmarkar's algorithm*. Technical report sol 85-5 system optimization laboratory; department of Operations Research Stanford. University Stanford California 94305 (1985).
- [21] Mekhilef, A., "Optimisation multi-objectif fractionnaire quadratique en nombres entiers", *Thèse doctorat*, pp. 4-9, (2020).
- [22] Miettinen, K., *Nonlinear multiobjective optimization*, Boston : Kluwer Academic Publishers, 298p, (1999).
- [23] Mokhtar S.Bazarra, Hanif D, N. Sherali, C.M.Shetty, "Nonlinear programming, Theory and Algorithms ", John Willy et Son, Inc,1993.

- 
- [24] A. Nagih, G. Plateau : Problèmes fractionnaires : Tour d'horizon sur les applications et méthodes de résolution. *RAIRO Operations Research* **33**, pp. 383-419, (1999).
- [25] Sheu, R.L., *Fractional Programming : a survey, recent developments, and applications*. Institute of Applied Mathematics National Cheng Kung University, Tainan, Taiwan.
- [26] Soland, R.M., "Multicriteria optimization : A general characterization of efficient solutions", *Decision Science*, Vol. 10, pp.26-38, (1979).
- [27] Steuer, R., *Multiple Criteria Optimization : Theory, Computation and Applications*, New-York : John Wiley and Sons, 568p, (1985).
- [28] Sundaram, R.K., *A First Course in Optimization Theory*, Cambridge University Press, 376p, (1996).