N° d'ORDRE : 06/2009-M/MT

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEURE ET
DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE
« HOUARI BOUMEDIEN »
FACULTE DE MATHEMATIQUES

MEMOIRE

Présenté pour l'obtention du diplôme de MAGISTER

EN : MATHEMATIQUES

Spécialité : Recherche Opérationnelle : Mathématiques de Gestion

Par : MAKHLOUF Nouara

Sujet

Sur un problème d'optimisation non linéaire discrète multi-objectifs en variables bornées

Soutenu le 25/11/2009, devant le jury composé de :

Mr- H.AIT HADDADENE	Professeur	U.S.T.H.B	Président
Mr- M.MOULAI	Professeur	U.S.T.H.B	Dteur deThèse
Mlle- I.BOUCHEMAKH	Professeur	U.S.T.H.B	Examinateur
Mme– N.MAACHOU	Chargé de Recherche	U.S.T.H.B	Invitée

Remerciments

Je tiens à remercier particulièrement Mr Mustapha MOULAÏ, mon directeur de thèse, pour avoir été présent à tout instant, pour ses connaissances et conseils avisés. Qu'il trouve ici l'expression de ma profonde reconnaissance pour ses qualités humaines et scientifiques. Je le remercie pour toute sa confiance et la liberté qu'il m'a accordée durant la réalisation de ce travail.

Mes sincères remerciements vont également à Mr Hacène AIT HADDADENE qui m'a honoré par son accord d'être président du jury de ma thèse.

Je remercie vivement Mlle Isma BOUCHEMAKH et Mme Nacera MAACHOU d'avoir accepté d'examiner ce travail.

Je ne terminerai pas sans adresser un immense merci à mes chers parents pour le soutien et l'encouragement qu'ils m'ont apportés durant toutes mes années d'études, ainsi que mes sœurs et tous ceux qui m'ont de près ou de loin aidée, merci à tous.

Dédicaces

Je dédie ce travail à mes chers parents,

à ma grand-mère,

à mes sœurs et toute ma famille,

à mes collègues, mes amies ,

à tous ceux qui m'ont témoigné leurs soutiens.

Table des matières

Introduction générale **1**

1 Introduction à la programmation linéaire **3**

1.1 Introduction 3

1.2 Programmation linéaire en variables continues 3

1.2.1 Forme générale d'un programme linéaire 3

1.2.2 Forme canonique et forme standard 4

1.2.3 Bases, bases réalisables, solutions de base 5

1.2.4 Caractérisation des bases optimales 6

1.2.5 Dualité 7

1.2.6 Algorithme du simplexe 7

1.3 Programmation linéaire en variables bornées [18] 9

1.3.1 Formulation mathématique 9

1.3.2 Théorème d'optimalité 9

1.3.3 Algorithme du simplexe en variables bornées [18] 9

1.4 Programmation linéaire en nombres entiers 11

1.4.1 Formulation mathématique 11

1.4.2 Méthodes de résolution d'un programme linéaire en nombres entiers 12

1.4.3 Méthodes des coupes 12

1.4.4 Méthode par séparation et évaluation progressive 14

1.5 Conclusion 16

2 Programmation linéaire multi-objectifs **17**

2.1 Introduction 17

2.1.1 Formulation Générale 17

2.1.2 Concepts de base 19

2.1.3 Cônes 23

2.1.4 Détection graphique de l'efficacité 24

2.1.5 Test d'efficacité d'Eker et Kouada[13] 25

2.2 Quelques méthodes de résolutions 26

2.2.1 Méthode de la Somme pondérée 26

2.2.2 Méthode de programmation par but "Goal Programming" . . . 27

2.2.3 Méthode lexicographique 27

2.3 Programmation linéaire multi-objectifs discrète 28

2.3.1 Formulation mathématique 28

2.3.2 Méthode de D. Klein & E. Hannan [20] 28

2.3.3 Méthode de A. Crema & J. Sylva [32] 30

2.3.4 Méthode de M. Moulaï et al.[1] 34

2.4 Conclusion 41

3 Programmation fractionnaire linéaire mono-objectif **42**

3.1 Introduction 42

3.2 Programmation fractionnaire linéaire continue 42

3.2.1 Formulation mathématique 42

3.2.2 La résolution directe 45

3.2.3 La résolution par paramétrisation 45

3.2.4 La résolution d'un programme équivalent 45

3.2.5 Méthode de A. Cambini *et al.*[5] 46

3.3 Programmation fractionnaire linéaire en nombres entiers 47

3.3.1 Formulation mathématique 47

3.3.2 Méthode de M. Moulaï *et al.* [24] 47

3.3.3 Exemple numérique : 49
3.4 Programmation fractionnaire discrète en variables bornées 50
3.4.1 Formulation mathématique 50
3.4.2 Algorithme du simplexe pour la résolution des problèmes fractionnaires en variables bornées 51
3.5 Conclusion 53

4 Programmation multi-objectifs fractionnaire linéaire discrète en variables bornées (MOILFPB) 54
4.1 Introduction 54
4.2 Définitions et notations 54
4.3 Algorithme de résolution d'un problème fractionnaire en nombre entiers à objectifs multiples[24] 57
4.3.1 Algorithme 58
4.3.2 Exemple numérique 59
4.4 Une méthode d'optimisation multi-objectifs fractionnaire linéaire discrète en variables bornées 65
4.4.1 Définitions et notations 65
4.4.2 Arête incidente[18] 67
4.4.3 Test d'éfficacitée 68
4.4.4 Résultats théoriques 70
4.4.5 Description de la méthode 76
4.4.6 Exemples numériques 79
4.5 Conclusion 84

Conclusion générale 85

Bibliographie 86

Introduction générale

Les méthodes de résolution des programmes mathématiques uni-critère ont été intensivement étudiées durant les cinquante dernières années. De nos jours, les méthodes de décision mono-objectif reflètent une ère simpliste. Le monde est devenu plus complexe. Chaque problème important du monde réel implique plus d'un objectif, et plus que jamais, le décideur trouve qu'il est impératif d'évaluer les solutions alternatives selon des critères multiples. Bien qu'on soit équipé de méthodes uni-critère performantes, nous avons besoin d'extensions théoriques et pratiques de façon à pouvoir résoudre un programme multi-objectifs qu'il soit linéaire ou non linéaire. L'optimisation à objectifs multiples est, sans aucun doute, un domaine de recherche important pour les scientifiques et les ingénieurs, non seulement en raison de la nature multi-objectifs de la plupart des problèmes réels, mais également parce qu'il reste beaucoup de questions en suspens dans ce domaine. En recherche opérationnelle, plusieurs techniques ont été développées au cours des années, en vue de traiter des problèmes à objectifs multiples.
Dans ce mémoire, nous nous intéressons plus précisément au problème de la programmation fractionnaire linéaire multi-objectifs discrète en variables bornées. Il s'agit d'un domaine spécifique de l'optimisation multi-objectifs.

Ce mémoire s'articule autour de quatre chapitres dont le premier porte sur les principaux résultats de la programmation linéaire, notamment les méthodes de résolution les plus utilisées dans ce domaine. Nous distinguerons le cas où les variables sont continues et le cas où elles sont discrètes et bornées.

Le deuxième chapitre est une introduction au problème de la programmation linéaire multi-objectifs en nombres entiers. Nous indiquerons en quoi consiste ce problème et décrirons, de façon détaillée, une méthode de résolution proposée par M. Moulaï et *al* [1].

Le troisième chapitre est une introduction au problème de la programmation fractionnaire linéaire que ce soit en variables continues, bornées ou bien entières.

Après introduction et description de la problématique de la programmation multi-objectifs fractionnaire linéaire à variables discrètes, nous présentons dans le quatrième chapitre une méthode exacte de ce dernier problème décrite dans[24]. Nous avons montré que cette dernière peut être généralisée au cas où les variables sont bornées moyennant l'introduction de coupes adéquates. Une description détaillée de la méthode est présentée avec une illustration numérique.
Ce présent travail ce termine par un récapitulatif présenté sous forme de résumé.

Chapitre 1

Introduction à la programmation linéaire

1.1 Introduction

La programmation linéaire a pour objet l'étude et la résolution des "programmes linéaires" c'est à dire des problèmes où la fonction objective aussi bien que les contraintes sont exprimées de manière linéaire. Il s'agit de la technique la plus célèbre de la recherche opérationnelle, ceci est du au fait que nombreux problèmes concrets tel que l'industrie pétrochimie, la gestion ... peuvent être modéliser comme des programmes linéaires. La résolution des programmes linéaires correspondant à ces modèles a permis de réaliser des progrès substantiels. D'autre part la programmation linéaire est un outil mathématique très riche qui donne à la fois un éclairage sur les méthodes d'optimisation continue et les méthodes d'optimisation discrète.

1.2 Programmation linéaire en variables continues

1.2.1 Forme générale d'un programme linéaire

Un programme linéaire (LP) est un problème d'optimisation consistant à maximiser (ou minimiser) une fonction objectif linéaire de n variables de décision x_j $(j = 1, ..., n)$ soumises à un ensemble de contraintes exprimées sous forme d'équations ou d'inéquations linéaires appelées contraintes.

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

$$(P)\begin{cases} optimiser \quad z=\sum_{j=1}^{n} c_j x_j \\ \sum_{j=1}^{n} a_{ij}x_j \leq b_i \qquad i\in I\subseteq\{1,\ldots,m\} \\ \sum_{j=1}^{n} a_{kj}x_j \geq b_k \qquad k\in K\subseteq\{1,\ldots,m\} \\ \sum_{j=1}^{n} a_{rj}x_j = b_r \qquad r\in R\subseteq\{1,\ldots,m\} \\ x_j\geq 0 \qquad j=\overline{1,n} \end{cases}$$

Où :

- Optimiser signifie minimiser ou maximiser.
- $C=(c_j)_{j=\overline{1,n}}\in\mathbb{R}^{1\times n}$, $A=(a_{ij})_{\substack{i=\overline{1,m}\\ j=\overline{1,n}}}\in\mathbb{R}^{m\times n}$, $b=(b_i)_{i=\overline{1,m}}\in\mathbb{R}^m$.
- Les ensembles I, K, et R sont disjoints et $I\cup K\cup R=\{1,\ldots,m\}$.

1.2.2 Forme canonique et forme standard

Les deux programmes linéaires :

$$(PC)\begin{cases} Max \quad z=\sum_{j=1}^{n} c_j x_j \\ \sum_{j=1}^{n} a_{ij}x_j \leq b_i \qquad i=\overline{1,m} \\ x_j\geq 0 \qquad j=\overline{1,n} \end{cases} \qquad (PS)\begin{cases} Max \quad z=\sum_{j=1}^{n} c_j x_j \\ \sum_{j=1}^{n} a_{ij}x_j = b_i \qquad i=\overline{1,m} \\ x_j\geq 0 \qquad j=\overline{1,n} \end{cases}$$

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

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

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

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

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

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

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

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

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

$$\begin{aligned} ax \leq b \quad &\Leftrightarrow \quad ax + s = b,\ s \geq 0 \\ ax \geq b \quad &\Leftrightarrow \quad ax - s = b,\ s \geq 0 \end{aligned}$$

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

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

1.2.3 Bases, bases réalisables, solutions de base

On considère le programme linéaire :

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

tel que le système $Ax = b$ soit de plein rang,

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

À une base B du programme linéaire (P) on associe une solution du système linéaire :

$$\begin{cases} x_B = (A^B)^{-1}b \\ x_N = 0 \end{cases}$$

Cette solution est dite solution de base associé à la base B. Une telle solution est dite ***réalisable*** si $x_B \geq 0$. Une solution de base est dite ***dégénérée*** si x_B a des composantes nulles.
Une base B d'un programme linéaire (p) est dite réalisable si la solution de base correspondante est réalisable.

1.2.4 Caractérisation des bases optimales

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

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

Où :

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

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

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

1.2.5 Dualité

Soit le programme linéaire écrit sous sa forme canonique :

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

Auquel on associe un autre programme linéaire appelé le programme ***dual***

$$(D)\begin{cases} Min \quad w = yb \\ yA \geq c \\ y \geq 0 \end{cases}$$

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

L'ensemble de remarques concernant l'écriture des programmes duaux sont présentés dans le tableau suivant :

$Primal(Dual)$	$Dual(Primal)$
$Fonction \ \ objectif \ \ à \ \ maximiser$	$Fonction \ \ objectif \ \ à \ \ minimiser$
$ieme \ \ containte \geq$	$ieme \ \ variable \leq 0$
$ieme \ \ containte \leq$	$ieme \ \ variable \geq 0$
$ieme \ \ containte =$	$ieme \ \ variable \lesseqgtr 0$
$ieme \ \ varible \geq 0$	$ieme \ \ containte \geq$
$ieme \ \ varible \leq 0$	$ieme \ \ containte \leq$
$ieme \ \ varible \gtreqless 0$	$ieme \ \ containte =$

TAB. 1.1 – passage du problème primal au problème dual

1.2.6 Algorithme du simplexe

L'algorithme du simplexe, introduit en 1947 par G.B. Dantzig [11], nous permet de se déplacer d'une solution de base réalisable à une autre améliorant la valeur de la fonction objectif, jusqu'à trouver une solution optimale (si elle existe) en un nombre fini d'étapes. Soit à résoudre le programme linéaire :

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

On définit l'application linéaire *col* par :

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

1. ***Initialisation.*** Soit P le programme linéaire écrit sous forme canonique par rapport à une base réalisable de départ B.la solution de base initiale est :

$$\begin{cases} x^*_{col(i)} = bi \quad i = 1, ..., m \\ x^*_j = 0 \qquad j \in N \end{cases}$$

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

   Choisir s tel que $c^s > 0$

   (a) S'il existe aller en 3).

   (b) Sinon, terminer (la solution optimale est trouvée).

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

   Soit $I = \{i, A^i_s > 0\}$

   (a) Si $I = \emptyset$, alors **terminer** (P n'a pas de solution optimale).

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

$$\frac{\widehat{b}_r}{\widehat{A}^s_r} = \underset{i=\overline{1,m}}{min}\left\{\frac{\widehat{b}_i}{\widehat{A}^s_i} | \widehat{A}^s_i > 0\right\}$$

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

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

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

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

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

Retour à (1)

1.3 Programmation linéaire en variables bornées [18]

1.3.1 Formulation mathématique

Considérons le programme linéaire :

$$(P_B) \begin{cases} Max \quad z = cx \\ Ax = b \\ L \leq x \leq U \end{cases}$$

Un tel problème est appelé *programme linéaire en variables bornées.*

Une solution du système linéaire $Ax = b$ telle que, pour $j \in N$ $x_j = l_j$ ou bien $x_j = u_j$ est appelée solution de base associé à la base B

Une telle solution est dite ***réalisable*** si pour tous $j \in B$ $l_j \leq x_j \leq u_j$

1.3.2 Théorème d'optimalité

Une solution de base réalisable $\overline{x}$ relative à la base B est optimale si les conditions suivantes sont satisfaites :

- $c^j - \pi.A^j > 0 \Longrightarrow \overline{x}_j = u_j$.
- $c^j - \pi.A^j < 0 \Longrightarrow \overline{x}_j = l_j$.

1.3.3 Algorithme du simplexe en variables bornées [18]

Posons $Z^J = \pi.A^j$

1. Calculer $c^j - Z^j$

(a) Si les conditions d'optimalité sont vérifiées, terminer (la solution est optimale).

(b) Sinon, $\exists\ j\ c^j - Z^j < 0$ et $x_j = u_j$ ou bien $\exists\ j\ c^j - Z^j > 0$ et $x_j = l_j$, aller à 2.

2. Soit r tel que $c^r - Z^r > 0$ et $x_r = l_r$, alors x_r doit changer de valeur de sorte que : $x_r = l_r + \phi_r$ et ϕ_r est déterminé par :

$$\phi_r = min\{(u_r - l_r), (\frac{x_{B_i} - l_{B_i}}{y_{ir}}, y_{ir} > 0), (\frac{u_{B_i} - x_{B_i}}{-y_{ir}}, y_{ir} < 0)\}$$

(a) Si $\phi_r = u_r - l_r$, alors x_r atteint sa borne et reste hors base.

La solution x_B et la valeur de la fonction objectif changent comme suit :

$$\hat{x}_{B_i} = x^0_{B_i} - y_{ir}.\phi_r, \forall i = 1, 2, ..., m.$$

$$\hat{Z} = Z - \phi_r(c^r - Z^r)$$

(b) Si $\phi_r = \frac{x_{B_s} - l_{B_s}}{y_{sr}}$ alors x_r rentre dans la base et x_{B_s} quitte la base et atteint sa borne inférieure.

La solution x_B et la valeur de la fonction objectif changent comme précédemment.

(c) Si $\phi_r = \frac{u_{B_t} - x_{B_t}}{-y_{tr}}$ alors x_r rentre dans la base et x_{B_t} quitte la base et atteint sa borne supérieure.

La solution x_B et la valeur de la fonction objectif changent comme précédemment.

3. Soit l tel que $c^l - Z^{lr} < 0$ et $x_l = u_l$, alors x_l doit changer de valeur de sorte que : $\hat{x}_l = u_l - \phi_l$ et ϕ_l est déterminé par :

$$\phi_l = min\{(u_l - l_l), (\frac{x_{B_i} - l_{B_i}}{-y_{il}}, y_{il} < 0), (\frac{u_{B_i} - x_{B_i}}{y_{il}}, y_{il} > 0)\}$$

(a) Si $\phi_l = u_l - l_l$, alors x_l atteint sa borne et reste hors base.

La solution x_B et la valeur de la fonction objectif changent comme suit :

$$\hat{x}_{B_i} = x^0_{B_i} + y_{il}.\phi_l, \forall i = 1, 2, ..., m.$$

$$\hat{Z} = Z - \phi_r(c^l - Z^l)$$

(b) Si $\phi_l = \frac{x_{B_p} - l_{B_p}}{-y_{pl}}$ alors x_l rentre dans la base et x_{B_p} quitte la base et atteint sa borne inférieure.

La solution x_B et la valeur de la fonction objectif changent comme précédemment.

(c) Si $\phi_l = \frac{u_{B_q} - x_{B_q}}{-y_{ql}}$ alors x_l rentre dans la base et x_{B_q} quitte la base et atteint sa borne supérieure.

La solution x_B et la valeur de la fonction objectif changent comme précédemment.

4. **Retour à (1)**.

L'algorithme dual du simplexe en variables bornées est décrit en détail dans [19].

1.4 Programmation linéaire en nombres entiers

La programmation linéaire en nombres entiers est un domaine très riche de la programmation mathématique.les recherches dans ce domaine sont nombreuses et ont vraiment commencé avec **Gomory** en 1958.

1.4.1 Formulation mathématique

Considérons le programme linéaire :

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

Où :

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

- $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^{1\times n}$.
- S est supposé borné et non vide.

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

Le problème deviendra donc :

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

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

1.4.2 Méthodes de résolution d'un programme linéaire en nombres entiers

Les deux principales familles de méthodes actuellement connues pour résoudre les problèmes de programmation linéaire en nombres entiers sont les *méthodes de coupes* et les *méthodes arborescentes*.

1.4.3 Méthodes des coupes

a) Coupe de Dantzig [10]

La coupe de Dantzig a été proposée sur la base que dans le problème relaxé (LP), le deuxième membre de l'équation matricielle des contraintes est un vecteur positif mais non entier, et une des variables hors-base est strictement positive (supérieure ou égale à 1), cette coupe est formulée par

$$\sum_{j \in N} x_j \geq 1 \tag{1.2}$$

où N est l'ensemble des indices hors-base

b) Coupe fractionnaire de Gomory [14]

L'idée principale de cette méthode est d'ajouter des contraintes linéaires qui n'excluent aucun point entier réalisable, une par une jusqu'à ce que la solution optimale de la relaxation soit entière.

Dans une première étape, on résout le programme relaxé (LP), on cherche une solution de base optimale en utilisant la méthode de simplexe, si elle existe, on choisit une variable de base non entière et on génère une inéquation sur la contrainte associée à cette variable afin de couper la région de faisabilité courante.

Étant donnée une base optimale B du problème relaxé (LP), le tableau optimal correspondant est donné par

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

TAB. 1.2 – Tableau simplexe optimal associé à la base B.

Où :

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

$\hat{c} = c - \pi A$: est dit *vecteur coût réduit* relatif à la base B, avec $\hat{c}_B = 0$.

Si la solution optimale de (LP) est entière, elle est la solution optimale du problème (ILP). Sinon, parmi les variables de base, choisissons $x_i,\ i \in B$ dont la valeur est fractionnaire.

La $i^{\text{ème}}$ ligne du tableau optimal est donnée par

$$x_i + \sum_{j \in N} \hat{a}_{ij} x_j = \hat{b}_i. \tag{1.3}$$

Où :

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

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

Notations : étant donné un nombre réel α, on désigne par :

$\lfloor \alpha \rfloor$: le plus grand entier inférieur ou égal à α.

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

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

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

de (1.3) on a :

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

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

$$x_i + \sum_{j \in N} \lfloor \hat{a}_{ij} \rfloor x_j \leq \lfloor \hat{b}_i \rfloor. \tag{1.4}$$

en soustrayant (1.4) de (1.3) on obtient

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

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

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

Cette contrainte est introduite dans le tableau simplexe optimal et le nouveau problème formé peut être résolu en utilisant la méthode dual simplexe. Après un nombre fini d'itérations, ou bien on obtient une solution optimale entière, ou bien le problème devient impossible.

1.4.4 Méthode par séparation et évaluation progressive

La méthode de "Branch & Bound" a été spécialement élaborée pour des problèmes en variables discrètes (ILP), le principe de cette méthode consiste à subdiviser l'ensemble S (l'ensembles de solutions admissibles) en un nombre fini de sous-ensembles S^i, généralement on prend

$$\bigcup_{i=1} S^i = S \qquad avec \qquad S^i \cap S^j = \emptyset, \forall (i,j), i \neq j$$

Ces subdivisions successives sont représentées à l'aide d'une arborescence de racine S et des "nœuds" S^i présentant les sous-ensembles de solutions effectués.
Le déroulement de la méthode est constitué principalement de trois procédures :

1) Procédure de séparation

La phase de séparation consiste à diviser le problème en un certain nombre de sous-problèmes qui ont chacun leur ensemble de solution réalisables. Ainsi, en résolvant tous les sous-problèmes et en prenant la meilleur solution trouvée, on est assuré d'avoir resolu le problème initial, ce principe de séparation peut être appliqué de manière recurssive à chacun des sous-ensembles de solutions obtenus.

2) Procédure d'évaluation

l'évaluation d'un nœud de l'arborescence a pour but de déterminer l'optimum (une borne supérieure ou inférieure pour un problème de minimisation) de l'ensemble des solutions réalisables associé au nœud en question, ou au contraire, de prouver mathématiquement que cet ensemble ne contient pas de solution intéressante pour la résolution du problème initial.
La solution optimale du sous-problème associé à un nœud donné est appelée solution partielle.

3) Procédure Stérilisation

Le but de cette procédure est d'éviter l'examination de tous les nœuds de l'arborescence. Dans le cas où la borne supérieure de la solution optimale du sous-problème traité est inférieur à la borne supérieure globale (ie. la meilleure solution trouvée jusqu'a présent), on est certain que toute solution réalisable de ce sous-problème ne sera pas meilleure que l'optimum global courant, il est donc inutile d'effectuer la séparation de son ensemble de solutions. On peut également arrêter la recherche dans un nœud lorsque le sous-problème qui est lui associé est non réalisable. Un problème de programmation linéaire en nombres entiers (ILP) peut avoir plusieurs solutions optimales, on parle dans ce cas de solution alternative dont la définition est la suivante :

Définition 1.1. Soit x^* une solution optimale du problème (ILP). Une solution réalisable $x' \in D$ est dite alternative à x^* si $cx' = cx^*$.

1.5 Conclusion

Dans ce chapitre nous avons présenté les éléments nécessaires de la programmation linéaire qui vont être utilisés dans les prochains chapitres.

Chapitre 2

Programmation linéaire multi-objectifs

2.1 Introduction

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

2.1.1 Formulation Générale

Le problème de programmation multi-objectifs linéaire est définit comme un problème de décision, qui consiste à optimiser (maximiser ou minimiser) simultanément r fonctions à valeurs réelles notées f^i, $i = \overline{1,r}$ appelées critères souvent contradictoire, sur un ensemble d'actions S.
Ce problème peut être formulé mathématiquement comme suit :

$$(MOP) \begin{cases} "Optimiser" [f_1(x), ..., f_r(x)] \\ x \in \mathbb{S} \end{cases}$$

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

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

Si les r critères sont linéaires et les contraintes sont linéaires en x, on parle de problème de programmation linéaire multi-objectifs (MOLP) :

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

Où :

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

Et si $x \in Z^n$, on aura un problème de programmation Avec : $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{Z}^m$, $C = (c^1, c^2, \ldots, c^r) \in \mathbb{Z}^{r \times n}$.
multi-objectifs en nombres entiers (MOILP) :

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

Où :

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

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

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

Avec :

$A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, c^i et $d^i \in \mathbb{R}^{1 \times n}$, α^i et $\beta^i \in \mathbb{R}$, $i = \overline{1,r}$ et $d^i x + \beta^i \neq 0$ sur

$D = \{x \in \mathbb{Z}^n | Ax = b,\ x \geq 0\}$ pour tout $i = \overline{1,r}$.

2.1.2 Concepts de base

Soit le problème de programmation linéaire à objectifs multiples suivant :

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

Où : $S = \{x \in \mathbb{R}^n | Ax = b,\ x \geq 0\}$ est supposé non vide.
On considère l'application linéaire φ qui associe à chaque vecteur $x \in S$ son image $\varphi(x) = (c^1 x, c^2 x, \ldots, c^r x)$ dans l'espace des critères.

Définition 2.1. L'espace $\mathbb{R}^n$ dans lequel se situe l'ensemble des actions S est appelé *espace des décisions*. D'autre part, et dans le cadre multi-objectifs, le décideur raisonne plutôt en terme d'évaluation d'une solution pour chaque objectif et se place naturellement dans l'espace :

$$\varphi(S) = \{Z = Cx \in \mathbb{R}^r | x \in S\}$$

appelé *espace des critères*.

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

Dominance

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

- Z est au moins aussi bon que Z' pour tous les critères et,

- Z est strictement meilleur que Z' pour au moins un critère.

Dominance forte

Soient Z, Z' deux vecteurs de $\mathbb{R}^r$. On dit que Z *domine fortement* Z' ssi :
$Z > Z'$ (i.e. $z^i > z'^i \ \forall i = \overline{1, r}$).
si Z domine fortement Z', alors Z est meilleur que Z' sur tous les critères.

Efficacité

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

$$Z(x) \geq Z(x^*)$$

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

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

Efficacité faible

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

Efficacité forte

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

Il existe deux points particuliers qui apparaissent dans l'espace des critères : le point *idéal* et le point *nadir*.

Point idéal

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

Point anti-idéal

Les coordonnées de ce point sont obtenues en optimisant chaque fonction objectif séparément : $\overline{Z} = \left(\underset{x \in D}{Min}\, z_1(x),\ \underset{x \in D}{Min}\, z_2(x), \ldots, \underset{x \in D}{Min}\, z_r(x) \right) \in \mathbb{R}^r$.
Généralement, parce que les objectifs sont contradictoires, le point idéal ne correspond pas à une solution réalisable.

Point nadir

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

$$\begin{pmatrix} \overline{z_1} & z_{12} & \cdots & z_{1r} \\ z_{21} & \overline{z_2} & \cdots & z_{2r} \\ \vdots & \vdots & \vdots & \vdots \\ z_{r1} & z_{r2} & \cdots & \overline{z_r} \end{pmatrix}$$

est appelée *matrice des gains*.

Où :

- $\overline{z_i} = \underset{x \in D}{Max}\, c^i x = c^i \overline{x_i},\ \forall i = \overline{1, r}$.
- $z_{ij} = c^i \overline{x_j}\ \forall i = \overline{1, r},\ \forall j = \overline{1, r}$ avec $i \neq j$.

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

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

Remarque. Si pour un critère j, il existe plusieurs solutions optimales, la matrice des gains n'est pas unique. En effet, la colonne j de la matrice des gains dépendra de la solution $\overline{x_j}$ choisie.

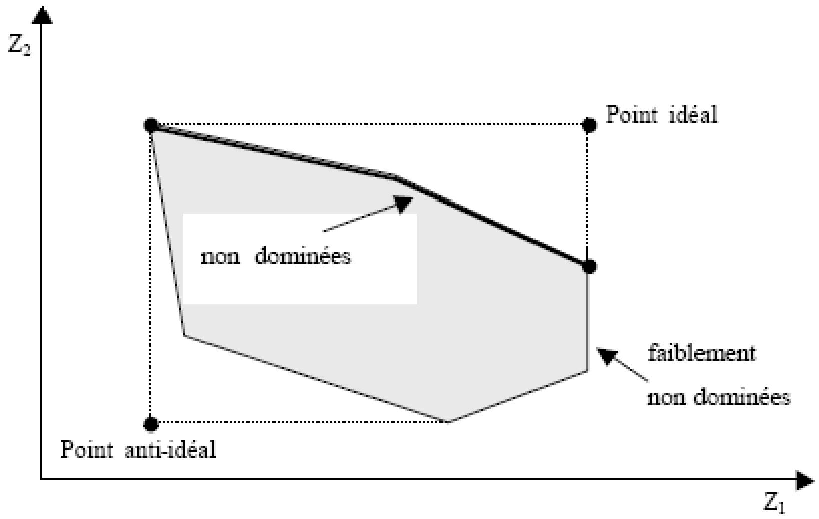


FIG. 2.1 – illustration des définitions

Solutions supportées et non supportées

Dans l'ensemble des solutions efficaces d'un problème de programmation linéaire discrète à objectifs multiples, deux types de solutions peuvent être différenciées :

- Solutions *supportées* notées $SES(P)$: Elles se trouvent sur l'enveloppe convexe [1] de D. Chacune de ces solutions peut être trouvée en optimisant une agrégation linéaire des objectifs. Cette dernière, consiste à transformer un problème multi-objectifs en un problème mono-objectif combinant les différentes fonctions objectif en une seule fonction F de façon linéaire :

$$F(x) = \sum_{i=1}^{r} \lambda_i c^i x$$

où les $\lambda_i \in [0, 1]$ et vérifiant $\displaystyle\sum_{i=1}^{r} \lambda_i = 1$.

[1] Soit X une partie de $\mathbb{R}^n$. L'enveloppe convexe de X noté $Conv(X)$ est définit comme étant la plus petite partie convexe contenant X : $Conv(X) = \left\{y \in \mathbb{R}^n | y = \sum_i \mu_i x_i,\ x_i \in X,\ \mu_i \geq 0,\ \sum_i \mu_i = 1\right\}$

Différents poids fournissent différentes solutions supportées ; une même solution pouvant être générée en utilisant des poids différents.

- Solutions *non supportées* notées $SENS(P)$: Ces solutions ne sont pas situées sur l'enveloppe convexe de D. Aucune de ces solutions ne peut être trouvée en optimisant une agrégation linéaire des objectifs.

2.1.3 Cônes

Définition 2.2 (Cône). Soit $u \in U \subset \mathbb{R}^n$, $U \neq \emptyset$. Alors, U est un *cône* si et seulement si $\alpha u \in U$ pour tout scalaire $\alpha \geq 0$. L'origine $0 \in \mathbb{R}^n$ est contenu dans chaque cône.

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

Définition 2.3 (Générateurs). Considérons $\left\{u^1, u^2, \ldots, u^k\right\}$, un ensemble de k vecteurs de $\mathbb{R}^n$ et l'ensemble U tel que :

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

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

Le seul cône pour lequel l'ensemble des générateurs est unique est $\{0 \in \mathbb{R}^n\}$.
Soit $\left\{u^1, u^2, \ldots, u^k\right\}$ un ensemble de générateurs pour le cône convexe U et soit $u^l \in \left\{u^1, u^2, \ldots, u^k\right\}$. Alors, u^l est *générateur non essentiel* si U peut être généré par $\left\{u^1, u^2, \ldots, u^k\right\} \setminus \left\{u^l\right\}$. Un générateur non essentiel est celui qui peut être exprimé comme combinaison linéaire d'autres générateurs, il est dit *essentiel* sinon.

Définition 2.4 (Dimension d'un cône). La *dimension* d'un cône $U \subset \mathbb{R}^n$ est donné par le nombre de vecteurs linéairement indépendants de U.

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

Définition 2.5 (Cône polaire). Soit $U \subset \mathbb{R}^n$ un cône. Alors, le *cône polaire* non négatif de U (notée $U^{\geq}$) est le cône convexe :

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

C'est-à-dire, tous les vecteurs de $U^{\geq}$ font un angle inférieur ou égal à 90^0 avec chaque vecteur de U.

Définition 2.6 (Cône polaire semi positif). Soit $U \subset \mathbb{R}^n$ un cône généré par l'ensemble $\left\{u^1, u^2, \ldots, u^k\right\}$. Alors, le *cône polaire semi positif* de U noté $U^{>}$ est le cône convexe :

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

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

2.1.4 Détection graphique de l'efficacité

Soit le problème de programmation linéaire multi-objectifs suivant :

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

Où : $S = \{x \in \mathbb{R}^n | Ax = b,\ x \geq 0\}$ avec : $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{Z}^m$ et $C = (c^i)_{i=\overline{1,r}} \in \mathbb{Z}^{r \times n}$. Pour tester l'efficacité en un point $x^* \in R$, Ralph E. Stewer [31] a introduit le concept d'*ensemble dominant* qui est principalement basé sur la notion du cône étudiée précédemment.

Définition 2.7 (Ensemble dominant). Soit $x^* \in S$ et $C^>$ le cône polaire semi positif du cône C généré par les gradients des r fonctions objectifs i.e. :

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

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

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

C'est à dire :

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

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

Théorème 2.1 ([31]). *Soit ED_{x^*} l'ensemble dominant en $x^* \in S$. Alors x^* est efficace si et seulement si : $ED_{x^*} \cap S = \{x^*\}$.*

Le théorème (2.1) fournit un test permettant de détecter les points efficaces qui peuvent être visualisés géométriquement :

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

2.1.5 Test d'efficacité d'Eker et Kouada[13]

pour tester l'efficacité d'une solution réalisable donnée, on résout un problème de programmation linéaire inchangé dans tous ses paramètres, sauf le vecteur second membre des contraintes, qu'il faut remplacer par la solution en question. Le théorème suivant est utilisée pour tester l'efficacité d'une solution réalisable donnée du problème (P).

Théorème 2.2. *Soit X^* un élément arbitraire de la région D. $X^* \in Eff$ si et seulement si la valeur optimale de la fonction objectif θ est nulle dans le problème de programmation linéaire mixte suivant :*

$$(P_{X^*}) \begin{cases} Max \quad \theta = \sum_{i=1}^{r} \psi_i \\ CX - I\psi = CX^* \\ X \in S \\ \psi_i \geq 0, \forall i = 1, ..., r \end{cases}$$

Où : C est une matrice $r \times n$,dont la i^{me} ligne correspond à $c^i, i = 1, ..., r$,I est la matrice identitée$(r \times r)$ et $\psi = (\psi_i)_{i=1,...,r}$.

2.2 Quelques méthodes de résolutions

L'optimisation multi-objectifs consiste à définir des méthodes efficaces pour la résolution de problèmes où tous les objectifs sont pris en compte. Nous décrivons brièvement quelques approches principale pour la détermination d'une solution efficace ou d'un ensemble des soltions efficaces.

2.2.1 Méthode de la Somme pondérée

Cette méthode consiste à additionner tous les objectifs en affectant à chacun d'eux un coefficient de poids, ce coefficient représente l'importance relative que le décideur attribue à l'objectif, cela transforme le problème multi-objectifs à un problème mono-objectif de la forme :

$$\max_{x \in S} \sum_{i=1}^{r} \lambda_i z^i(x), \lambda \in \Lambda.$$

la solution de ce problème dépend fortement du vecteur paramètre utilisé. Cette méthode à été développée pour la génération des solutions non dominée. Elle est très efficace et peut être appliquée pour produire une solution non dominée initiale qui peut être employée comme solution initiale pour d'autres techniques.

2.2.2 Méthode de programmation par but “Goal Programming”

Charnes et Cooper (1961) et Ijiri (1965) sont crédités du développement de la méthode “goal programming” pour un modèle linéaire, Dans cette méthode le décideur doit fixer un but (niveaux d'aspiration) qu'il souhaite à atteindre pour chaque critère, ces valeurs sont ajoutées au problème comme des contraintes supplémentaire, la nouvelle fonction objectif est modéfiée de façon à minimiser la somme des écarts entre les valeurs réalisées des fonctions objectifs et les buts à atteindre. La forme la plus simple de cette méthode peut être formulée comme suit :

$$\min_{x\in S}\{\sum_{i=1}^{r}|z^i(x)-\hat{z}^i|\}$$

Où $\hat{z}^i$ représente la valeur à atteindre pour le $i^{\text{ème}}$ critère.
La méthode est très facile à mettre en œuvre mais la définition des buts à atteindre est une question délicate qui détermine l'efficacité de la méthode.

2.2.3 Méthode lexicographique

Cette méthode est proposée par Fourman (1985), dans laquelle les objectifs sont préalablement rangés par ordre d'importance par le décideur. Ensuite, la solution optimale est obtenue en optimisant les objectifs l'un après l'autre selon cet ordre en commençant par le plus important, puis dans chaque itération, les objectifs du niveau supérieur sont ajouté sous forme des contraintes en fixant chaqu'un d'eux à sa valeur optimale trouvée précédemment.
Supposons que les objectifs sont rangés par l'ordre suivant :
$z^1 \succ z^2 \succ ... \succ z^r$, le premier problème à résoudre est donné par

$$\begin{cases} \max & z^1 = c^1x \\ t.q. & x\in S. \end{cases}$$

Soit x_1^* la solution optimale trouvée avec $z^{1^*} = z^1(x_1^*)$.
z^{1^*} devient une nouvelle contrainte qui est ajoutée au deuxième problème

$$\begin{cases} \max & z^2 = c^2 x \\ t.q. & x \in S. \\ & z^1(x) = z^{1^*}. \end{cases}$$

Le $k^{\text{ème}}$ problème sera le suivant :

$$\begin{cases} \max & z^k = c^k x \\ t.q. & x \in S \\ & z^1(x) = z^{1^*}, z^2(x) = z^{2^*}, ..., z^{k-1}(x) = z^{k-1^*}. \end{cases}$$

La procédure est répétée jusqu'à ce que tous les objectifs soient traités et la solution obtenue à l'étape r sera la solution du problème.

2.3 Programmation linéaire multi-objectifs discrète

2.3.1 Formulation mathématique

Un problème de programmation linéaire multi-objective en nombre entiers peut être formulé mathématiquement comme suit :

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

Où :

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

Nombreuses stratégies ont été proposées, consistant à caractériser totalement ou partiellement l'ensemble des solutions efficaces du problème de programmation linéaire multiobjectif en nombres entiers. Parmi ces méthodes, nous citons : méthode de D. Klein & E. Hannan [20], méthode de A. Crema & J. Sylva [32] et la méthode de M. Moulaï & M. Abbas [1]que nous présenterons en détail.

2.3.2 Méthode de D. Klein & E. Hannan [20]

La technique proposée par les auteurs peut être utilisée aussi bien pour identifier l'ensemble de toutes les solutions efficaces que pour en caractériser une partie seulement. Elle consiste à résoudre progressivement une séquence de programmes linéaires mono objectif en nombres entiers avec des contraintes ajoutées à chaque étape. Les contraintes supplémentaires éliminent les solutions efficaces déjà trouvées, et font en sorte que les nouvelles solutions générées soient efficaces.

Algorithme

Étape 1. Résoudre le problème (P_1) définit comme suit :

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

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

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

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

$$(P_j)\ \begin{cases} Max \quad z^s = c^s x \\ x \in D \\ \bigcap\limits_{i=1}^{q} \left(\bigcup\limits_{\substack{k=1 \\ k \neq s}}^{r} c^k x \geq c^k y^i + f_k \right) \quad (*) \end{cases}$$

Avec :

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

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

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

L'ensemble de contraintes $(*)$ assure que les solutions de (P_j) seront meilleurs que toutes les solutions dans Y^{j-1}, pour au moins un critère $k \neq s$. Il est clair que la procédure est finie étant donné qu'on élimine au moins une solution admissible à chaque étape et qu'il en existe un nombre finie (D est supposé borné).
Lorsqu'à chaque étape j, $f_k = 1$, $\forall k = \overline{1, r}$, $k \neq s$, la procédure fournit l'ensemble de toutes les solutions efficaces du problème. Cependant, si $f_k > 1$ pour certains k, $k = \overline{1, r}$, $k \neq s$, seulement un sous ensemble de points efficaces est généré.

Théorème 2.3 ([20]). *Si le problème (P_j) possède un ensemble de solutions optimales X_j^*, alors le sous ensemble de solutions efficaces dans X_j^*, l'ensemble $Eff(P_j)$, est efficace pour (P).*

Corrolaire 2.1 ([20]). *Si la solution optimale de (P_j) est unique, alors elle est efficace pour (P).*

2.3.3 Méthode de A. Crema & J. Sylva [32]

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

Algorithme

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

Deux cas se présentent :

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

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

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

Après k étapes du processus :

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

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

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

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

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

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

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

Proposition 2.1 ([32]). *Soient $x^1, x^2, \ldots, x^k$ des solutions efficaces du problème (P). Posons $\Delta_j = \left\{x \in \mathbb{Z}^n | c^i x \leq c^i x^j \quad \forall i = \overline{1,r}\right\}$ $(j = \overline{1,k})$. Si x^* est efficace pour le problème :*

$$(Q) \begin{cases} "Max" \quad c^i x \quad i = \overline{1,r} \\ x \in \left(D - \displaystyle\bigcup_{j=1}^{k} \Delta_j\right) \end{cases}$$

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

Corrolaire 2.2. *Soient* $x^1, x^2, \ldots, x^k$ *des solutions efficaces du problème* (P) *et* $\Delta_j = \left\{x \in \mathbb{Z}^n \mid c^i x \leq c^i x^j \quad \forall i=\overline{1,r}\right\}$ $(j=\overline{1,k})$. *Si* x^* *est une solution optimale pour le problème mono objectif :* $Max \left\{\sum_{i=1}^{r} \lambda_i c^i x \mid x \in \left(D - \bigcup_{j=1}^{k} \Delta_j\right)\right\}$ *pour certaines valeurs du vecteur* $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_r)$, $\lambda > 0$, *alors* x^* *est une solution efficace pour le problème* (P).

Remarque. Pour les problèmes de grande taille, la détermination de l'ensemble de toutes les solutions non dominées devient très coûteuse en terme de temps de calcul. Pour cela, une étape d'interaction avec le décideur peut être intégrée à la procédure. Cette étape a pour objectif d'éliminer des solutions efficaces que le décideur juge insatisfaisantes. Dans ce cas le problème (P_{k+1}) devient :

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

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

Pour la méthode qui va suivre, nous utilisons les notations suivantes :
On considère le problème :

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

pour $k \geq 1$,
$S_k = \{x \in Z^{nk} / A_k x \leq b_k, A_k \in R^{mk \times nk}, b_k \in R^{mk}, x \geq 0\}$

S_k est la région tronquée courante de S obtenue par des coupes de gomory successives.

$x_k^1 = x_{k,j}^1$ est la solution optimale entière du problème (P_1) obtenue sur S_k à l'étape k.

Z_i^1 est la valeur de $Z_i, i \in (2, ..., r)$ correspondant à la solution x_1^1.

B_k^1 est une base de S_k.

$a_{k,j}^1 \in R^{mk \times 1}$ sont les vecteurs activités de $x_{k,j}^1$ appropriés à la région tronquée courante S_k.

$y_{k,j}^1 = (y_{k,ij}^1) = (B_k^1)^{-1} a_{k,j}^1$ où $y_{k,j}^1 \in R^{mk \times 1}$.

$I_k = \{j : a_{k,j}^1 \in B_k^1\}$.

$N_k = \{j : a_{k,j}^1 \notin B_k^1\}$.

$c_j^1 =$ la $\grave{j}^{me}$ composante du vecteur c^1.

$d_j^1 =$ la $\grave{j}^{me}$ composante du vecteur d^1.

$c_{k,j}^1 = \sum_{i \in I_1} c_i^1 y_{k,ij}^1$.

$d_{k,j}^1 = \sum_{i \in I_1} d_i^1 y_{k,ij}^1$.

$Z_1(x_k^1) = C^1 x_k^1$

$Z_{k,1}^1 = \sum_{i \in I^k} C^1 y_{k,ij}^1$.

$\Gamma_k = \{j/j \in N_k \ et \ Z_{k,j}^1 - C_j^1 = 0\}$

Définition 2.8. Une arête E^{j_k}, $j_k \in N_k$ incidente à x^k est définie comme étant l'ensemble :

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

Où : $0 \leq \theta_{j_k} \leq \theta = \min\limits_{i \in B_k} \left\{ \dfrac{x_i^k}{\widehat{a}_{lig(i)k}^{j_k}} | \widehat{a}_{lig(i)k}^{j_k} > 0 \right\}$.

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

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

2.3.4 Méthode de M. Moulaï et al.[1]

Cette méthode, dénommée "MODILIM", a été proposée par M. Moulaï & M. Abbas pour la détermination de toutes les solutions efficaces du problème de programmation linéaire multi-objectifs en nombres entiers.

Développement de la méthode

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

Étape 1. Résoudre le problème (P_1) et trouver la solution optimale entière x^1 sur S_1. Construire l'ensemble Γ_1.

Étape 2. Tester l'ensemble Γ_1.

Cas 1. Si $\Gamma_1 = \emptyset$, x^1 est l'unique solution optimale sur S_1. Soit $(z_1^1, z_1^2, \ldots, z_1^r)$ le vecteur critère correspondant, il est enregistré dans $Eff(P)$ comme étant le premier $r-$uplet non dominé.

Tronquer le point x^1 par la coupe :

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

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

Cas 2. Si $\Gamma_1 \neq \emptyset$, choisir un indice quelconque $j_1 \in \Gamma_1$ et calculer le nombre θ de l'opération pivot.

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

$r-$uplet potentiellement non dominé.

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

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

(b) Si pour tout $j_1 \in \Gamma_1$, on a $\theta < 1$, alors choisir un indice quelconque $j_1 \in \Gamma_1$ et appliquer la coupe : $\displaystyle\sum_{j \in N_1 \setminus \{j_1\}} x_j \geq 1.$

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

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

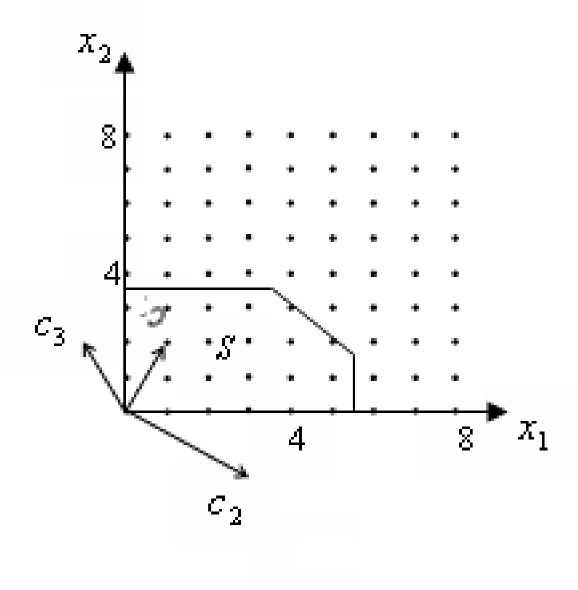
L'arête $E^{j_{k-1}}$ est tronquée par la coupe : $\displaystyle\sum_{j \in N_{k-1} \setminus \{j_{k-1}\}} x_j \geq 1.$

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

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

Pour illustrer l'algorithme précédent, on considère le problème suivant :

$$(P)\begin{cases} Max \quad z_1 = C^1 = x_1 + 2x_2 \\ \\ Max \quad z_2 = C^2 = 3x_1 - 2x_2 \\ \\ Max \quad z_3 = C^3 = -x_1 + 2x_2 \\ \\ x_1 + x_2 \leq 7 \\ 2x_1 \leq 11 \\ 2x_2 \leq 7 \\ x_1, x_2 \geq 0 \\ x_1, x_2 \in \mathbb{Z}_+ \end{cases}$$



Graphe de l'exemple

- Étape1

  Résoudre le problème relaxé P_1 suivant :

$$(P_1)\begin{cases} Max \quad z_1 = C^1 = x_1 + 2x_2 \\ \\ x_1 + x_2 \leq 7 \\ 2x_1 \leq 11 \\ 2x_2 \leq 7 \\ x_1, x_2 \geq 0 \\ x_1, x_2 \in \mathbb{Z}_+ \end{cases}$$

La solution de base réalisable optimale du problème P_1 est donnée par le tableau suivant :

base	x_1	x_2	x_3	x_4	x_5	x_6	b
x_1	1	0	1	0	0	-1	4
x_4	0	0	-2	0	0	2	3
x_2	0	1	0	0	0	1	3
x_5	0	0	0	1	1	-2	1
$z_j^1 - c_j^1$	0	0	1	0	0	1	10
$z_j^2 - c_j^2$	0	0	3	0	0	-5	6
$z_j^3 - c_j^3$	0	0	-1	0	0	3	2

TAB. 2.1 – Tableau 1

la solution optimale de P_1 est $x_1^1 = (4,3)$, et elle produit le premier triplet efficace $(10,6,2)$. Par suite $Eff_0 = \{(10,6,2)\}$.

- Étape2

  $\Gamma_1 = \emptyset$, x_1^1est unique. Ce point est tronqué en utilisant la coupe $x_3 + x_6 \geq 1$, comme le présente la figure suivante.

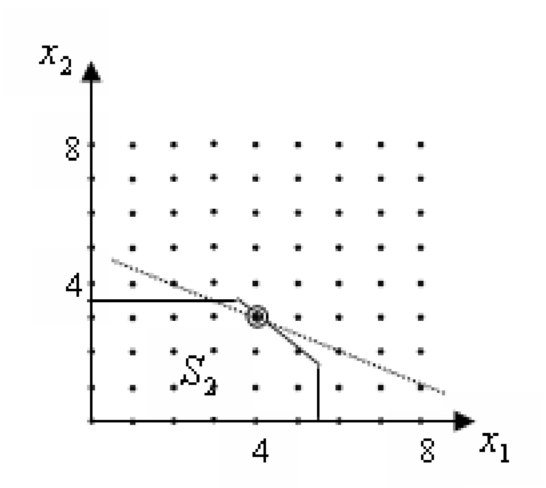


FIG. 2.2 – fig1

En utilisant l'algorithme dual du simplexe, on obtient une solution réalisable $x_2^1 = (3,3)$ donnée par le tableau 2 optimale suivant :

base	x_1	x_2	x_3	x_4	x_5	x_6	x_7	b
x_1	1	0	0	0	0	-2	1	3
x_4	0	0	0	0	0	4	-2	5
x_2	0	1	0	0	0	1	0	3
x_5	0	0	0	1	1	-2	0	1
x_3	0	0	1	1	1	1	-1	1
$z_j^1 - c_j^1$	0	0	0	0	0	0	1	9
$z_j^2 - c_j^2$	0	0	0	0	0	-8	3	3
$z_j^3 - c_j^3$	0	0	0	0	0	4	4	3

TAB. 2.2 – Tableau 2

Le triplet (9,3,3) qui correspond à $x_2^1 = (3,3)$ est non dominé, par conséquent $Eff_1 = \{(10,6,2),(9,3,3)\}$

- Étape3

  $\Gamma_2 \neq \emptyset$, x_2^1 n'est pas unique prendre, $j_2 = 6$, on a $\phi_{j2} \leq min\{\frac{5}{4}, \frac{3}{1}, \frac{1}{1}\} = 1$. La solution alternative $x_2^2 = (5,2)$ donne le triplet non dominé (9,11,-1). Donc on augmente $Eff_2 = Eff_1 \bigcup (9,11,-1)$.

  L'arête E_6 est tronqué par la coupe : $x_7 \geq 1$, i.e.$-x_7 + x_8 = -1$, comme le présente la figure suivante.

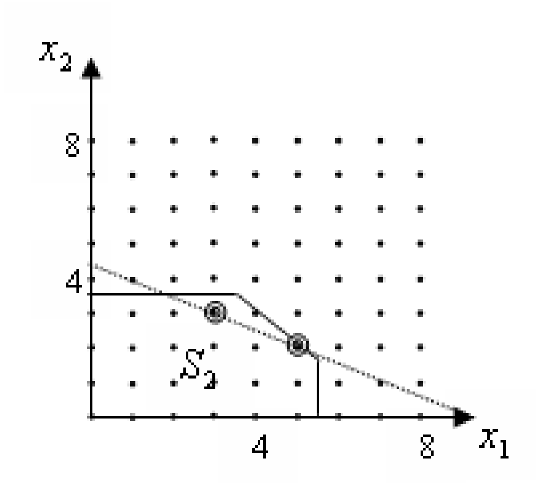


FIG. 2.3 – fig2

En appliquant l'algorithme dual du simplexe on obtient une solution entière réalisable $x_3^1 = (2,3)$ le triplet (8,0,4) correspondant à la solution x_3^1 est non dominé, donc $Eff_2 = \{Eff_2 \bigcup (8,0,4)\}$

- Étape4

  $\Gamma_3 \neq \emptyset$,x_3^1 n'est pas unique prendre, $j_2 = 6$, on a $\phi_{j3} \leq min\{\frac{7}{4}, \frac{3}{1}, \frac{2}{1}\} = \frac{7}{4}$. La solution alternative $x_3^2 = (4,2)$ donne le triplet non dominé(8,8,0). Donc on augmente $Eff_3 = Eff_2 \bigcup (8,8,0)$
  de la même manière, on obtient
  $Eff_9 = \{(10,6,2),(9,3,3),(9,11,-1),(8,0,4),(8,8,0),(7,-3,5)$
  $,(7,13,-3),(6,-6,6),(5,15,-5)\}$ et une solution entière réalisable $x_{10}^1 = (1,0)$, le triplet (1,3,-1) correspondant à la solution x_{10}^1 est dominé, donc $Eff_{10} = Eff_9$

- Étape11

  $\Gamma_{10} \neq \emptyset$,$x_{10}^1$ n'est pas unique prendre, $j_{10} = 18$, on a $\phi_{j10} \leq min\{\frac{9}{4}, \frac{0}{1}, \frac{6}{1}\} = 0$. La solution x_{10}^1 ne possède pas de solution alternative après sa troncture, le dual du simplexe donne une solution entière réalisable $x_{11}^1 = (0,0)$ donnée par le tableau optimale suivant :

base	x_1	x_2	x_3	x_4	x_5	x_6	x_{18}	x_{19}	b
x_1	1	0	0	0	0	0	-2	1	0
x_4	0	0	0	1	0	0	4	-2	11
x_2	0	1	0	0	0	0	1	0	0
x_5	0	0	0	0	1	0	-2	0	7
x_3	0	0	1	0	0	0	1	-1	7
x_6	0	0	0	0	0	1	-1	0	3
$z_j^1 - c_j^1$	0	0	0	0	0	0	0	1	0
$z_j^2 - c_j^2$	0	0	0	0	0	0	-8	3	0
$z_j^3 - c_j^3$	0	0	0	0	0	0	4	-1	0

TAB. 2.3 – Tableau 11

- Étape12

  $\Gamma_{11} \neq \emptyset$,$x_{11}^1$ n'est pas unique prendre, $j_{11} = 18$, on a $\phi_{j11} \leq min\{\frac{11}{4}, \frac{0}{1}, \frac{7}{1}\} = 0$. La solution x_{11}^1 ne possède pas de solution alternative après la troncature de l'arete E_{18}, comme le présente la figure suivante .

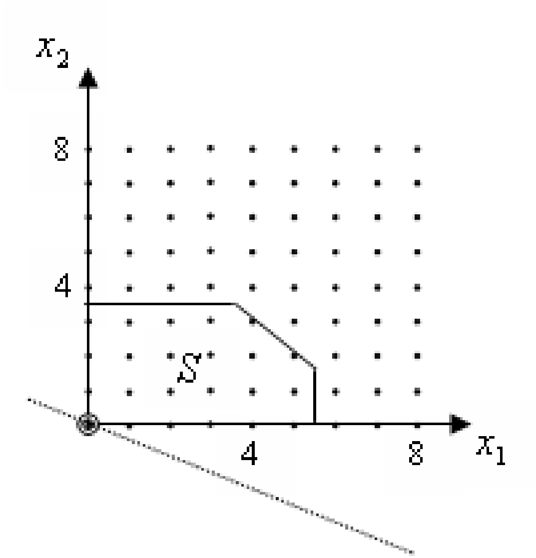


FIG. 2.4 – fig11

Le dual du simplexe donne une solution non réalisable donnée par le tableau optimale suivant :

base	x_1	x_2	x_3	x_4	x_5	x_6	x_{18}	x_{19}	b
x_1	1	0	0	0	0	0	-2	1	0
x_4	0	0	0	1	0	0	4	-2	11
x_2	0	1	0	0	0	0	1	0	0
x_5	0	0	0	0	1	0	-2	0	7
x_3	0	0	1	0	0	0	1	-1	7
x_6	0	0	0	0	0	1	-1	0	3
$z_j^1 - c_j^1$	0	0	0	0	0	0	0	1	0
$z_j^2 - c_j^2$	0	0	0	0	0	0	-8	3	0
$z_j^3 - c_j^3$	0	0	0	0	0	0	4	-1	0

TAB. 2.4 – Tableau 12

la variable x_2 est sélectionnée pour sortir de la base, mais l'opération pivot est impossible car la ligne $a_{3,j}$ est positive ou nul. Donc il n'y a plus de solutions entières réalisables .

d'où l'ensemble des solution efficace est

$Eff = \{(10, 6, 2), (9, 3, 3), (9, 11, -1), (8, 0, 4), (8, 8, 0), (7, -3, 5),$
$(7, 13, -3), (6, -6, 6), (5, 15, -5)\}$

2.4 Conclusion

Dans ce chapitre, nous avons présenté les concepts de base de la programmation multi-objectifs, ainsi que quelques méthodes de résolution dans le cas continu et entier.

Chapitre 3

Programmation fractionnaire linéaire mono-objectif

3.1 Introduction

Le terme *programmation fractionnaire* est utilisé pour désigner un type de problèmes d'optimisation où la fonction objectif est un quotient $f(x)/g(x)$, soumis à un ensemble de contraintes. Différentes versions de ce modèle existent, linéaires ou non linéaires, en nombres entiers, en continus ou en bornés.

3.2 Programmation fractionnaire linéaire continue

3.2.1 Formulation mathématique

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

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

Où :

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

selon Stewer [31], les programmes fractionnaires linéaires présentent l'intérêt particulier d'avoir des courbes de niveaux linéaires de leurs fonctions objectif.

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

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

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

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

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

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

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

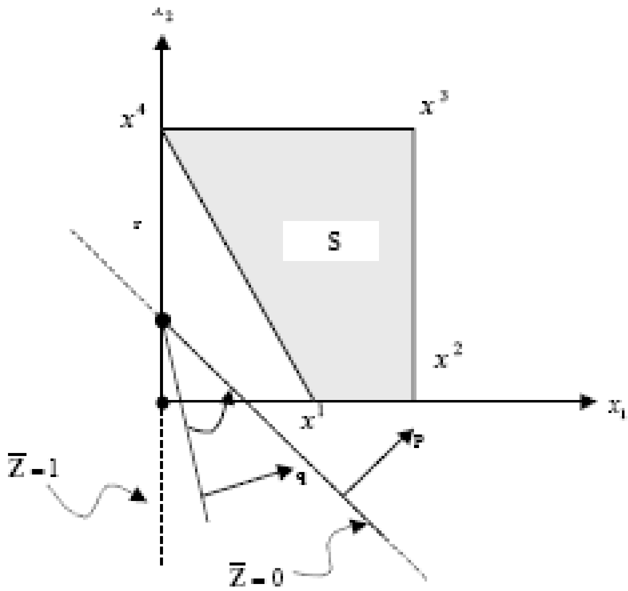
Les éléments de cet ensemble sont déterminés par la résolution du système :

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

Exemple illustratif

On Considère le programme fractionnaire linéaire suivant :

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



Graphe de l'exemple

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

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

. Donc pour :

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

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

Dans la littérature, il existe plusieurs stratégies de résolution d'un programme fractionnaire mono-objectif :

3.2.2 La résolution directe

Dans cette stratégie, le programme fractionnaire est traité sous sa forme originale, c'est-à-dire sans modifier ni l'objectif ni l'ensemble des contraintes. Cette approche est utilisée pour résoudre les programmes hyperboliques tant qu'en variables continues qu'en variables entières.

3.2.3 La résolution par paramétrisation

A l'inverse de la résolution directe, on construit un problème à objectif simplifié, combinaison linéaire du numérateur et du dénominateur par l'intermédiaire d'un paramètre, tout en gardant inchangé l'ensemble de contraintes. Une séquence de résolutions de ce type de problème fournit une solution optimale du programme fractionnaire.

3.2.4 La résolution d'un programme équivalent

Un changement de variables permet de simplifier aussi l'objectif mais en faisant augmenter le nombre de variables et de contraintes. Charnes and Cooper [7] sont les premiers a avoir linéarisé un problème hyperbolique en un problème linéaire équivalent :

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

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

Proposition 3.1 (Charnes & Cooper [7]). *Si (y^*, z^*) est une solution optimale de (PE), alors $z^* > 0$ et $x^* = \dfrac{y^*}{z^*}$ est une solution optimale de (P).*

3.2.5 Méthode de A. Cambini *et al.*[5]

On considère le programme fractionnaire linéaire continu (P) :

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

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

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

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

L'algorithme de Cambini *et al.* génère une séquence finie x^k, $k = \overline{1,l}$ de solutions optimales niveau dont la première est trouvée de la façon suivante :
Résoudre le programme linéaire (P_0) $\{Min\ dx+\beta \mid x \in S\}$, soit x^0 la solution optimale (car sa fonction objectif est bornée). Si x^0 est unique, alors elle est une solution niveau, sinon résoudre le programme linéaire (P_1) $Max\ \{cx+\alpha \mid dx = dx^0;\ x \in S\}$. Si (P_1) n'admet pas de solutions, alors la valeur de la fonction objectif est infinie ; sinon une solution optimale x^1 de (P_1) est aussi une solution optimale niveau.

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

Algorithme

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

Si une telle solution n'existe pas, alors $sup \left\{\dfrac{cx+\alpha}{dx+\beta} | x \in S\right\} = +\infty$, terminer.
Sinon, poser $k = 1$ et aller à l'étape 2.

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

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

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

3.3 Programmation fractionnaire linéaire en nombres entiers

3.3.1 Formulation mathématique

Le problème de programmation fractionnaire linéaire en nombre entiers a la forme suivante :

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

Où :

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

3.3.2 Méthode de M. Moulaï *et al.* [24]

Dans cette section, une méthode par séparation et évaluation progressive pour la résolution des problèmes de programmation fractionnaire linéaire discrète $(ILFP)$ est présenté. Elle est basée sur la méthode de résolution des problèmes(LFP)de Cambini et Martein .

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

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

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

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

$$(PR)\begin{cases} Max \quad \dfrac{\displaystyle\sum_{j\in N} \widehat{c}_j x_j + \widehat{\alpha}}{\displaystyle\sum_{j\in N} \widehat{d}_j x_j + \widehat{\beta}} \\ x_B + \widehat{A}^N x_N = \widehat{b} \\ x_B, x_N \geq 0, \ entiers \end{cases}$$

Où : B est l'ensemble des indices de base et N est l'ensemble des indices hors base.

- Calcule de pénalités : après l'obtention de la solution optimale continue de (P), la fonction objectif associée est donnée dans le tableau optimale du simplexe par (PR). Soit b_k la valeur non entière de x_k pour un certain $k \in B$. Pour sélectionner la nouvelle branche qui doit être ajoutée à (PR), il faut calculer les pénalités π_r et $\pi_{r'}$ des contraintes $x_k \leq \lfloor e_k \rfloor$ et $x_k \geq \lceil e_k \rceil$ respectivement, donnée par :

  $\pi_r = \frac{b.\Delta_r}{\bar{B}(\bar{B}+\frac{b.\bar{q}_r}{\bar{a}_{kr}})}$, $\pi_{r'} = \frac{(1-b).\Delta_r}{\bar{B}(\bar{B}+\frac{(1-b).\bar{q}_r}{\bar{a}_{kr}})}$ où

  $\Delta_r = min\{\frac{\bar{\gamma}_j}{-\bar{a}_{kj}}/\bar{a}_{kj} > 0\}$, $\Delta_{r'} = min\{\frac{\bar{\gamma}_j}{-\bar{a}_{kj}}/\bar{a}_{kj} < 0\}$ et $b = b_k - \lfloor b_k \rfloor$

  la branche correspondante à la plus petite valeur des pénalités est sélectionnée.

Algorithme de résolution :

Étape 1. Trouver la solution optimale niveau x^0 du problème(P) .

Si une telle solution n'existe pas, alors

– *soit* $sup \left\{\dfrac{cx+\alpha}{dx+\beta} | x \in S\right\} = +\infty$, lorsque la solution niveau de base optimale de (ILFP) n'existe pas.

– *soit* $sup \left\{ \frac{cx+\alpha}{dx+\beta} | x \in S \right\} = \max_{j \in J} \frac{\hat{c}}{\hat{d}}$, lorsque l'opération de pivot est impossible.

Sinon, poser $k = 1$, $l = 0$ et aller à l'étape 2.

Étape 2 .

– Si x^0 est entier, terminer. x^0 est une solution optimale du problème (P).

– Sinon, soit $x_k, k \in B$, une composante non entière de x^0 ayant pour valeur b_k. Poser $\pi_l = 0$ et aller à l'étape 3.

Étape 3 .

– Calculer π_{2k-1} et π_{2k}. Poser $\pi_{2k-1} = \pi_{2k-1} + \pi_l, \pi_{2k} = \pi_{2k} + \pi_l$et $\pi_l = +\infty$

– Calculer $\pi_l = \max_{1 \leq j \leq 2k}\{\pi_j\}$

– Ajouter la contrainte au tableau du simplexe optimale ,effectuer les opérations de pivot et aller à l'étape 4.

Étape 4 .

– Si x^l est une solution entière, terminer. x^l est une solution optimale du problème(P).

– Sinon,

1. Le problème augmenté n'a pas de solutions, terminer.
2. Soit x_k^l une composante non entière de x^layant pour valeur b_k^l. Poser $k = k + 1$ et aller à l'étape 3.

3.3.3 Exemple numérique :

Considérons le problème fractionnaire linéaire en nombre entier (P) suivant :

$$(P) \begin{cases} Max \quad z = \dfrac{7x_1 + 9x_2 + 3}{3x_1 + 4x_2 + 2} \\ 2x_1 + 3x_2 \leq 6 \\ 3x_1 + 2x_2 \leq 5 \\ x_1, x_2 \geq 0, entiers \end{cases}$$

En résolvant le problème relaxé,on obtient le tableau du simplexe optimale qui est représenté par le programme fractionnaire linéaire suivant :

$$(PR') \begin{cases} Max \quad z = \dfrac{13x_3 + 3x_4 + 108}{6x_3 + x_4 + 51} \\ x_1 - (2/5)x_3 + (3/5)x_4 = (3/5) \\ x_2 - (3/5)x_3 - (2/5)x_4 = (8/5) \\ x_j \geq 0, j = 1, 2, 3, 4 \end{cases}$$

La solution continue optimale est $x_1 = (3/5), x_2 = (8/5)$ et $x_j = 0, j = 3, 4$; avec une valeur de Z égale à 36/17. Puisque x_1, x_2 ne sont pas entières,cette solution n'est pas réalisable pour (P). Donc en calculant les pénalités π_1 et π_2 des contraintes $x \leq \lfloor 3/5 \rfloor$ et $x \geq \lceil 3/5 \rceil$, respectivement on obtient $\pi_1 = 3/170$ et $\pi_2 = 1/153$. la branche dont la valeur de pénalité est la plus petite est sélectionnée et la contrainte respective est rajoutée au problème (PR').
la solution obtenu est $x_1 = 1, x_2 = 1, x_3 = 1$ et $x_j = 0, j \in N$. Maintenant la solution est entière et donc elle est optimale pour le problème (P).

3.4 Programmation fractionnaire discrète en variables bornées

3.4.1 Formulation mathématique

Le problème de programmation fractionnaire linéaire discrète en variables bornées a la forma suivante :

$$(P) \begin{cases} Max \quad z = \dfrac{cx + \alpha}{dx + \beta} \\ x \in S. \end{cases}$$

Où :

- $S = \{x \in \mathbb{Z}^n \mid Ax = b,\ L \leq x \leq U\}$, avec α et β sont des réels, $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{Z}^m$, c et d des vecteurs de $\mathbb{Z}^{1 \times n}$.
- $dx + \beta \neq 0$, $\forall x \in S$.

3.4.2 Algorithme du simplexe pour la résolution des problèmes fractionnaires en variables bornées

Dans cette section on présente l'algorithme du simplexe pour la programmation fractionnaire en variables bornées qui est une extension de l'algorithme du simplexe pour la résolution des programmes fractionnaire dans le cas continu.
considéron le problème de programmation fractionnaire suivant :

$$(P)\begin{cases} Max \quad z = \dfrac{cx+\alpha}{dx+\beta} = \dfrac{n(X)}{d(X)} \\ AX = b \\ \text{L}\leq\text{x}\leq\text{U} \end{cases}$$

Définitions et notations :

X_B =la solution de base réalisable pour le problème P.
B = la matrice de base correspondante à la solution de base.
$I = \{i/a_i \in (B)\}$.
$N_1 = \{j|a_j \notin B \ et \ x_j \ est \ \ sa \ borne \ inf\}$.
$N_2 = \{j|a_j \notin B \ et \ x_j \ est \ \ sa \ borne \ sup\}$.
pour tous $j = 1, 2, ..., n$,$Y_j = \{y_{ij}\} = (B)^{-1}a_j$.
$Z_j^1 = \sum_{i\in I} c_{B_I} y_{ij}$.
$Z_j^2 = \sum_{i\in I} d_{B_I} y_{ij}$.
$Z(X) = \frac{n(X)}{d(X)}$.
$\delta_j = n(Z_j^2 - d_j) - d(Z_j^1 - cj)$.

Théorème d'optimalité :

Une solution de base réalisable $X = (X_B \ X_{N_1} \ X_{N_2})$ est optimale pour (P) si $\delta_j \leq 0, \forall j \in N_1$; $\delta_j \geq 0, \forall j \in N_2$ et $\delta_j = 0$ pour les variables de base.

Algorithme de résolution des programmes fractionnaires en variables bornées

phase 1. La phase 1 de cette méthode trouve une solution de base réalisable initiale pour P par la résolution d'un programme linéaire avec des variables bornées.

Étape1. Introduire les variables artificielles $x_{n+1}, x_{n+2}, ..., x_{n+m}$ à chaque équation dans P. Dans la solution de base obtenu durant l'algorithme ,les variables de base sont soit à leurs bornes inf ou à leurs bornes sup. À la phase1 on maximise

$$(P_2)\begin{cases} Max \quad Z^* = -x_{n+1} - x_{n+2} - ... - x_{n+m} \\ AX + IX_\alpha = b \\ L \leq X \leq U \\ X_\alpha \geq 0 \end{cases}$$

où :

- $X_\alpha = (x_{n+1}, x_{n+2}, ..., x_{n+m})^T$, et I est la matrice identité.

Une solution de base initiale $X_B = (x_{B_I})_{i=1,2,...,m}$ pour ce problème est donner par :

$$x_{B_i} = x_{n+i} = (b) - \sum_{j \in N_1} a_{ij} l_j - \sum_{j \in N_2} a_{ij} u_j \tag{3.1}$$

Étape2. Appliquer l'algorithme du simplexe en variables bornée.

Étape3. D'après la valeur de Z^*, on a trois cas :

1. Max $Z^* < 0$,et une variable artificielle est présente dans la base avec une valeur positive, dans ce cas le problème original (P) n'a pas de solution réalisable.
2. Max $Z^* = 0$, et une variable artificielle est présente dans la base avec une valeur nulle. Dans ce cas le problème original(P) a une solution réalisable. pour obtenir une solution réalisable on passe à la phase 2, ou bien éliminer la variable artificielle et puis passé à l'étape 2.
3. Max $Z^* = 0$,et aucune variable artificielle n' est présente dans la

base. Dans ce cas une solution de base réalisable est obtenue pour le problème(P). Aller à la phase2.

phase2. On considère la solution optimale obtenue à la phase 1 comme une solution de base réalisable initiale pour le problème originale(P), terminer, x^k est une solution optimale du problème (P). Calculer δ_j pour tous j. Si les conditions d'optimalité sont vérifiées alors la solution courante est optimale pour le problème(P). Sinon il existe un certain $x_r,\ r \in N_1$ tel que $\delta_r > 0$ alors x_r doit changer de valeur comme dans (b) dans l'étape 2 de l'algorithme du simplexe en variables bornées et la valeur de la fonction objectif et le solution de base changent comme suit :
$Z(\hat{X}) = \frac{n(X^0)-\phi_r(Z_r^1-c_r)}{d(X^0)-\phi_r(Z_r^2-d_r)}$ $\ et\ \ \hat{x}_{B_i} = x_{B_i}^0 - y_{ir}\phi_r, \forall i = 1, 2, ..., m.$
Ou bien il existe un certain $x_l,\ l \in N_2$ tel que $\delta_r < 0$ alors x_r doit changer de valeur comme dans (c) dans l'étape2 de l'algorithme du simplexe en variables bornées et la valeur de la fonction objectif et la solution de base changent comme suit :
$Z(\hat{X}) = \frac{n(X^0)+\phi_l(Z_l^1-c_l)}{d(X^0)+\phi_l(Z_l^2-d_l)}$ $\ et\ \ \hat{x}_{B_i} = x_{B_i}^0 + y_{il}\phi_l, \forall i = 1, 2, ..., m.$

3.5 Conclusion

Dans ce chapitre nous nous sommes intéressés au problème de l'optimisation fractionnaire linéaire continue, discrète et bornée. La méthode décrite pour ce dernier va être utilisé dans la méthode que nous allons proposer pour la détermination de toutes les solutions efficaces du problème de programmation multi-objectif discrète fractionnaire linéaire en variables bornées ($MOILFPB$).

Chapitre 4

Programmation multi-objectifs fractionnaire linéaire discrète en variables bornées (MOILFPB)

4.1 Introduction

La programmation fractionnaire a été étudiée par plusieurs auteurs pour le problème à un seul objectif, il n'existe que quelque articles traitant le cas de plusieurs fonctions objectifs fractionnaires, moins encore, pour le cas ou les variables sont entières et bornées. Le but de ce chapitre est de présenter une méthode pour la détermination de l'ensemble des solutions efficaces pour un problème d'optimisation multi-objectifs discrète fractionnaire linéaire en variables bornées ($MOILFPB$).

4.2 Définitions et notations

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

$$
(P)\begin{cases}
Max \quad z^1 = \dfrac{c^1x+\alpha^1}{d^1x+\beta^1} \\
Max \quad z^2 = \dfrac{c^2x+\alpha^2}{d^2x+\beta^2} \\
\qquad\qquad \vdots \\
Max \quad z^r = \dfrac{c^rx+\alpha^r}{d^rx+\beta^r} \\
x \in D \\
x \;\; entier
\end{cases}
$$

Où :

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

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

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

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

Afin de résoudre (P), une approche consistant à résoudre une séquence finie de problèmes de programmation fractionnaire linéaire discrète. Considérons donc, le problème de programmation fractionnaire linéaire entier unicritère donnée sous la forme suivante :

$$
(P_1)\begin{cases}
Max \quad z^1 = \dfrac{c^1x+\alpha^1}{d^1x+\beta^1} \\
x \in S \\
x \;\; entier
\end{cases}
$$

Pour résoudre P_1, on a besoin des notations suivantes :

$S_1 = \{x \in R^{n1}/A_1 x \leq b_1, A_1 \in R^{m1\times n1}, b_1 \in R^{m1}, x \geq 0\}$

S_1 est la région tronquée courante de S obtenue par des coupes de gomory successives.

Z_1^1 est la valeur optimale de Z_1 dans le problème (P_1).

$x_1^1 = x_{1,j}^1$ est la solution optimale de Z_1 dans le problème (P_1).

Z_i^1 est la valeur de $Z_i, i \in (2, ..., r)$ correspondant à la solution x_1^1.

B_1^1 est une base de S_1.

$a_{1,j}^1 \in R^{m1\times 1}$ sont les vecteurs activités de $x_{1,j}^1$ appropriés à la région tronquée courante S_1.

$y_{1,j}^1 = (y_{1,ij}^1) = (B_1^1)^{-1} a_{1,j}^1$ où $y_{1,j}^1 \in R^{m1\times 1}$.

$I_1 = \{j : a_{1,j}^1 \in B_1^1\}$.

$N_1 = \{j : a_{1,j}^1 \notin B_1^1\}$.

$c_j^1 =$ la $j^{\grave{}me}$ composante du vecteur c^1.

$d_j^1 =$ la $j^{\grave{}me}$ composante du vecteur d^1.

$c_{1,j}^1 = \sum_{i\in I_1} c_i^1 y_{1,ij}^1$.

$d_{1,j}^1 = \sum_{i\in I_1} d_i^1 y_{1,ij}^1$.

$Z_1(x_1^1) = \frac{Z_{1,1}^1}{Z_{1,2}^1}$

où

$Z_{1,1}^1 = c^1 x_1^1 + \alpha^1$.

$Z_{1,2}^1 = d^1 x_1^1 + \beta^1$.

$\bar{\gamma}_{1,j}^1 = Z_{1,2}^1 (c_j^1 - c_{1,j}^1) - Z_{1,1}^1 (d_j^1 - d_{1,j}^1)$ le coût réduit relatif à la $j^{\grave{}me}$ composante du vecteur $\bar{\gamma}_1^1$.

$\Gamma_1 = \{j/j \in N_1 \ et \ \bar{\gamma}_1^1 = 0\}$

pour $k \geq 2$,

$S_k = \{x \in R^{nk}/A_k x \leq b_k, A_k \in R^{mk\times nk}, b_k \in R^{mk}, x \geq 0\}$

S_k est la région tronquée courante de S obtenue par des coupes de gomory successives.

$x_k^1 = x_{k,j}^1$ est la solution optimale entière du problème (P_1) obtenue sur S_k à l'étape k.

Z_i^1 est la valeur de $Z_i, i \in (2, ..., r)$ correspondant à la solution x_1^1.

B_k^1 est une base de S_1.

$a^1_{k,j} \in R^{mk\times 1}$ sont les vecteurs activités de $x^1_{k,j}$ appropriés à la région tronquée courante S_k.

$y^1_{k,j} = (y^1_{k,ij}) = (B^1_k)^{-1}a^1_{k,j}$ où $y^1_{k,j} \in R^{mk\times 1}$.

$I_k = \{j : a^1_{k,j} \in B^1_k\}$.

$N_1 = \{j : a^1_{k,j} \notin B^1_k\}$.

$c^1_j =$ la $\grave{j}^{me}$ composante du vecteur c^1.

$d^1_j =$ la $\grave{j}^{me}$ composante du vecteur d^1.

$c^1_{k,j} = \sum_{i\in I_1} c^1_i y^1_{k,ij}$.

$d^1_{k,j} = \sum_{i\in I_1} d^1_i y^1_{k,ij}$.

$Z_1(x^1_k) = \frac{Z^1_{k,1}}{Z^1_{k,2}}$

où

$Z^1_{k,1} = c^1x^1_k + \alpha^1$.

$Z^1_{k,2} = d^1x^1_k + \beta^1$.

$\bar{\gamma}^1_{k,j} = Z^1_{k,2}(c^1_j - c^1_{k,j}) - Z^1_{k,1}(d^1_j - d^1_{k,j})$ le coût réduit relatif à la $\grave{j}^{me}$ composante du vecteur $\bar{\gamma}^1_k$.

$\Gamma_1 = \{j/j \in N_1 \ et \ \bar{\gamma}^1_k = 0\}$

Théorème 4.1. *[24] Le point x^1_k de S est une solution optimale du problème fractionnaire P_1 si et seulement si le vecteur gradient $\bar{\gamma}$ est tel que $\bar{\gamma}^1_{k,j} \leq 0$ pour tout indice $j \in N_k$.*

Corrolaire 4.1. *[24] La solution x^1_k du problème fractionnaire P_1 est unique si et seulement si le vecteur gradient $\bar{\gamma}$ est tel que $\bar{\gamma}^1_{k,j} < 0$ pour tout indice $j \in N_k$.*

4.3 Algorithme de résolution d'un problème fractionnaire en nombre entiers à objectifs multiples[24]

Un algorithme qui donne l'ensemble des solutions efficaces d'un problème fractionnaire en nombre entiers à objectifs multiple est présenté dans les étapes suivantes :

4.3.1 Algorithme

Étape 1 : Résoudre P_1 par n'importe quelle méthode directe de la programmation fractionnaire discrète. Autrement dit, rajouter aux m premières lignes de la matrice des contraintes A, les trois lignes $(m+1), (m+2)$ *et* $(m+3)$ relative respectivement aux vecteurs numérateur c, dénominateur$d$et au valeurs du gradient réduit $\bar{\gamma}_j$ de la fonction objectif où :

$$\bar{\gamma}_j = \bar{\beta c_j} - \bar{\alpha d_j}, \quad \forall j \in \{1, ..., n\}.$$

Donner une solution optimale entière x_1^1 sur S_1, de manière analogue, on peut considérer le problèmeP_i avec un objectif Z_i pour un indice quelconque$i \in \{2, ..., r\}$. Construire l'ensembleΓ_1.

Étape 2 : **Cas 1.** Si $\bar{\gamma}_{i,j}^1 < 0$ pour tout indice $j \in N_1$, x_1^1 est l'unique solution optimale sur S_1 et Z_1^1 est la valeur optimale de Z_1. Calculer les valeurs Z_i^1 de Z_i donnée par $x_1^1, i \in \{2, ..., r\}$. Enregistrer le premier vecteur efficace comme $(Z_1^1, ..., Z_r^1)$ pour construire l'ensemble des solutions efficaces Eff_0. Puisque $\Gamma_1 = \emptyset$, l'arête E_{j1} ne contient pas de solutions réalisables entières alternatives à x_1^1.

Tronquer le point x_1^1 par la coupe de Dantzig suivante :

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

Par application de la méthode dual fractionnaire relative à la programmation fractionnaire, on obtient une solution réalisable entièrex_2^1 dans la région tronquée S_2. Rajouter le vecteur correspondant $(Z_1^1, ..., Z_r^1)$ à Eff_0 s'il n'est pas dominé par l'un des précédents vecteurs critères éfficaces. D'où Eff_0 devient Eff_1.

Cas 2. Sinon, il existe un indice $j_1 \in N_1$ pour lequel $\bar{\gamma}_{1,j}^1 = 0$. Déterminer dans ce cas toutes les solutions qui lui sont alternatives, éliminer celles qui ne sont pas efficaces et mettre à jour l'ensemble Eff_0.

Appliquer la coupe suivante pour éliminer toutes les solutions alternatives trouvées à cette étape :

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

Étape k **;** $k \geq 2$**.** Choisir un indice $j_{k-1} \in \Gamma_{k-1}$ et explorée l'arête E_{jk-1} pour déterminer d'éventuelles solutions entières réalisables x_{k-1}^q, $q \in 2, ..., p_{k-1}$ alternative à x_{k-1}^1. Augmenter l'ensemble Eff_{k-1} par les vecteurs critères non dominés correspondant pour construire Eff_{k-1}.

l'arête E_{jk-1} est tronquée par la coupe

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

Étape finale La procédure s'arrête lorsque la méthode dual 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.

4.3.2 Exemple numérique

Pour illustrer l'algorithme précédent, on considère le problème suivant tiré de [24] :

$$(P) \begin{cases} Max \quad z_1(x) = \dfrac{x_1 - 4}{-x_2 + 3} \\ Max \quad z_2(x) = \dfrac{-x_1 + 4}{x_2 + 1} \\ Max \quad z_3(x) = -x_1 + x_2 \\ -x_1 + 4x_2 \leq 0 \\ x_1 - 1/2x_2 \leq 4 \\ x_1, x_2 \geq 0 \\ x_1, x_2 \in \mathbb{Z}_+ \end{cases}$$

- Étape1

Résoudre le problème relaxé P_1 suivant :

$$(P_1)\begin{cases} Max \quad z_1(x) = \dfrac{x_1 - 4}{-x_2 + 3} \\ -x_1 + 4x_2 \leq 0 \\ x_1 - 1/2x_2 \leq 4 \\ x_1, x_2 \geq 0 \\ x_1, x_2 \in \mathbb{Z}_+ \end{cases}$$

La solution optimale réalisable de base de S_1 domaine tronqué de S du problème est donnée par le tableau suivant :

base	x_1	x_2	x_3	x_4	x_5	b
x_3	0	4	1	1	1	4
x_4	0	$-1/2$	0	0	-1	0
x_1	1	0	0	0	1	4
c_j	0	0	0	0	-1	0
d_j	0	-1	0	0	0	-3
$\bar{\gamma}_j$	0	0	0	0	-3	0

TAB. 4.1 – Tableau 1

$I_1 = \{1, 3, 4\}, \quad N_1 = \{2, 5\}, \quad \Gamma_1 = \{2\}$

$x_1^1 = (4, 0)$

$Z_1^1 = Z_1(x_1^1) = 0; Z_2^1 = Z_2(x_1^1) = 0; Z_3^1 = Z_3(x_1^1) = -4$

Le premier triplet correspondant à x_1^1 est égale à $(0, 0, -4)$.

- Étape2

$\Gamma_1 \neq \emptyset$, $j_1 = 2$ et parcourir l'arête E_2.

cela signifie que x_1^1 n'est pas unique, et pour $\varphi_{j1} = 1$, la solution $x_1^2 = (4, 1)$ est une solution alternative à la solution x_1^1. et le triplet correspondant $0, 0, -3)$ domine le triplet $(0, 0, -4)$ ce qui entraîne $Eff_0 = \{(4, 1)\}$.

l'arête E_2 est tronqué par la coupe $x_5 \geq 1$ en utilisant l'algorithme dual du simplexe , on obtient une solution réalisable entière $x_2^1 = (3,0)$ donnée par le tableau 2 optimale suivant :

base	x_1	x_2	x_3	x_4	x_5	x_6	b
x_3	0	4	1	0	0	1	3
x_4	0	$-1/2$	0	1	0	-1	1
x_1	1	0	0	0	0	1	3
x_5	0	0	0	0	1	-1	1
c_j	0	0	0	0	0	-1	1
d_j	0	-1	0	0	0	0	-3
$\bar{\gamma}_j$	0	-1	0	0	0	-3	$-1/3$

TAB. 4.2 – Tableau 2

Le triplet $(-1/3, 1, -3)$ qui correspond à $x_2^1 = (3,0)$ est non dominé,par conséquent $Eff_1 = \{(4,1),(3,0)\}$,$N_2 = \{2,6\}$, $\Gamma_2 = \emptyset$

- Étape3

$\Gamma_2 = \emptyset$, x_2^1 est une solution unique. Ce point est tronqué par la coupe :$x_2+x_6 \geq 1$, i.e.$-x_7 - x_6 + x_7 = -1$.

L'application de la méthode dual du simplexe produit la solution $x_3^1 = (2,0)$ donnée par le tableau suivant :

base	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	b
x_3	0	0	1	0	0	0	1	-3	2
x_4	0	0	0	1	0	0	-1	$-1/2$	2
x_1	1	0	0	0	0	0	1	1	2
x_5	0	0	0	0	1	0	1	-1	2
x_6	0	0	0	0	0	0	-1	-1	1
x_2	0	1	0	0	0	0	0	1	0
c^j	0	7	0	0	0	0	-1	-1	2
d^j	0	8	0	0	0	0	0	1	-3
$\bar{\gamma}_j$	0	0	0	0	0	0	-3	-1	$-2/3$

TAB. 4.3 – Tableau 3

Le triplet $(-2/3, 2, -2)$ correspondant à la solution x_3^1 est non dominé, d'où$Eff_2 = \{(4,1),(3,0),(2,0)\}$,$N_3 = \{7,8\}$, $\Gamma_3 = \emptyset$

- Étape4

$\Gamma_3 = \emptyset$, x_3^1 est unique, ce point est tronqué par la coupe :$x_7 + x_8 \geq 1$, i.e $-x_7 - x_8 + x_9 = -1$.

l'application de la méthode dual du simplexe produit la solution $x_4^1 = (1,0)$ donnée par le tableau suivant :

base	x_1	x_2	x_3	x_4	x_5	x_6	x_9	x_{10}	b
x_3	0	0	1	0	0	0	1	4	1
x_4	0	0	0	1	0	0	-1	$-1/2$	3
x_1	1	0	0	0	0	0	1	0	1
x_5	0	0	0	0	1	0	1	2	3
x_6	0	0	0	0	0	1	-1	0	2
x_2	0	1	0	0	0	0	0	-1	0
c_j	0	0	0	0	0	0	-1	0	3
d_j	0	0	0	0	0	0	0	-1	-3
$\bar{\gamma}_j$	0	0	0	0	0	0	-3	-3	-1

TAB. 4.4 – Tableau 4

Dont le triplet $(- - 1, 3, -1)$ correspondant est non dominée. D'où $Eff_3 = Eff_2 \bigcup \{(3,0)\}$, $N_4 = \{9,10\}$, $\Gamma_4 = \emptyset$

- étape10

$\Gamma_4 = \emptyset$, x_4^1 est unique, ce point est tronqué par la coupe :$x_9 + x_{10} \geq 1$, i.e $-x_9 - x_{10} + x_{11} = -1$.

L'application de la méthode dual du simplexe produit la solution $x_5^1 = (0,0)$ donnée par le tableau suivant :

$base$	x_1	x_2	x_3	x_4	x_5	x_6	x_{10}	x_{11}	b
x_3	0	0	1	0	0	0	3	1	0
x_4	0	0	0	1	0	0	1/2	−1	4
x_1	1	0	0	0	0	0	−1	1	0
x_5	0	0	0	0	1	0	1	1	2
x_6	0	0	0	0	0	1	1	−1	3
x_2	0	1	0	0	0	0	−1	1	0
c_j	0	0	0	0	0	0	1	−1	4
d_j	0	0	0	0	0	0	−1	0	−3
$\bar{\gamma}_j$	0	0	0	0	0	0	−1	−3	4/3

TAB. 4.5 – Tableau 5

Dont le triplet $(-4/3, 4, 0)$ correspondant est non dominée. D'où $Eff_4 = Eff_3 \bigcup \{(2,0)\}$, $N_5 = \{10, 11\}, \quad \Gamma_5 = \emptyset$

- étape11

  $\Gamma_5 = \emptyset$, x_{54}^1 est unique, ce point est tronqué par la coupe : $x_{10} + x_{11} \geq 1$, i.e.$-x_{10} - x_{11} + x_{12} = -1$.

  Lors de l'application de la méthode dual du simplexe, l'opération usuel de pivot devient impossible indiquant que le domaine tronqué restant ne contient aucun point entiers réalisable et que toute les solutions efficaces ont été obtenues. Cette impossibilité de pivotage est donnée par le tableau suivant :

base	x_1	x_2	x_3	x_4	x_5	x_6	x_{11}	x_{12}	b
x_3	0	0	1	0	0	0	2	1	-1
x_4	0	0	0	1	0	0	3/2	-1	5
x_1	1	0	0	0	0	0	-2	1	-1
x_5	0	0	0	0	1	0	0	1	1
x_6	0	0	0	0	0	1	2	-1	4
x_2	0	1	0	0	0	0	-1	0	0
c_j	0	0	0	0	0	0	2	-1	5
d_j	0	0	0	0	0	0	-1	0	-3
$\bar{\gamma}_j$	0	0	0	0	0	0	1	-3	$-5/3$

TAB. 4.6 – Tableau 6

La variable x_3 est sélectionnée pour sortir de la base, mais l'opération pivot est impossible car la ligne $a_{3,j}$ est positive ou nul. Donc il n'y a plus de solutions entières réalisables. D'où l'ensemble des solutions efficaces est

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

4.4 Une méthode d'optimisation multi-objectifs fractionnaire linéaire discrète en variables bornées

4.4.1 Définitions et notations

Considérons le problème de programmation fractionnaire linéaire discrète en variables bornées à objectifs multiples ($MOILFPB$) suivant :

$$(P)\begin{cases} Max \quad z^1 = \dfrac{C^1x+\alpha^1}{D^1x+\beta^1} \\ Max \quad z^2 = \dfrac{C^2x+\alpha^2}{D^2x+\beta^2} \\ \qquad\qquad \vdots \\ Max \quad z^r = \dfrac{C^rx+\alpha^r}{D^rx+\beta^r} \\ x \in D \\ x \ \ entier \\ L \leq x \leq U \end{cases}$$

Où :

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

Avec : $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, C^i et $D^i \in \mathbb{R}^{1\times n}$, α^i et $\beta^i \in \mathbb{R} \quad i = \overline{1,r}$.

Afin de résoudre P ,les solutions réalisables entières du problème P_1,définit ci-dessous sont recherchées et rangées dans le sens décroissant des valeurs respectives de la fonction objectif Z_1.

Considérons donc le programme fractionnaire linéaire entiers uni-critère à variables bornées donné sous la forme suivante :

$$(P_1)\begin{cases} Max \quad z^1 = \dfrac{C^1x+\alpha^1}{D^1x+\beta^1} \\ x \in D \\ x \quad entier \\ L \leq x \leq U \end{cases}$$

La recherche des solutions réalisables entières du problème P_1 nécessite l'introduction des notations suivantes :

$c_j^1 =$ la $\grave{j}^{me}$ composante du vecteur C^1.

$d_j^1 =$ la $\grave{j}^{me}$ composante du vecteur D^1.

X^k c'est la k^{me}solution optimale du problème (P^1) à l'étape k

X^k est un point extrême de $S(k)$ qui est obtenu à partir de D après les coupes successive. soit $S_k = \{X/A_kX = b_k, L \leq X \leq U\}$

$X^k = x_{,j}^k$ est laj^{me} composante deX^k .

B^k= la base correspondante à X^k.

$a_j =$ le vecteurs activités de x_j^k appropriés à la région tronquée courante S_k.

$Y_j = (y_{ij}) = (B^k)^{-1}a_j$

$I_k = \{j : a_j \in B^k\}$.

$N_1^k = \{j : a_j \notin B^k \quad et \quad x_j \quad est \quad sa \quad borne \quad inf\}$.

$N_2^k = \{j : a_j \notin B^k \quad et \quad x_j \quad est \quad sa \quad borne \quad sup\}$.

$X^k = (X_{B^k} \quad X_{N_1^K} \quad X_{N_2^K})$

$C_{B^k}^1$ =coût associé aux variables de base dans le numérateur .

$C_{N_1^k}^1$ =Coût associé aux variables hors base dans le numérateur qui sont à leur borne inf.

$C_{N_2^k}^1$ =Coût associé aux variables hors base dans le numérateur qui sont à leur borne sup.

$D_{B^k}^1$ =Coût associé aux variables de base dans le dénominateur .

$D_{N_1^k}^1$ =Coût associé aux variables hors base dans le dénominateur qui sont à leur borne inf.

$D_{N_2^k}^1$ =Coût associé aux variables hors base dans le dénominateur qui sont à leur borne sup.

$n^{1k} = n(X^k) = C^{1T}X^k + \alpha_1.$

$d^{1k} = d(X^k) = D^{1T}X^k + \beta_1.$

$Z_j^1 = \sum_{i\in I_k} c^1_{B_i^k} y_{ij}.$

$Z_j^2 = \sum_{i\in I_k} d^1_{B_i^k} y_{ij}.$

$Z(X^k) = \frac{n(X^k)}{d^{X^k}}$

$\gamma_j^k = n^k(Z_j^2 - d_j^1) - d^k(Z_j^1 - c_j^1)$ le coût réduit relatif à la $\grave{j}^{me}$ composante du vecteur γ^k.

$J_1^K = \{j/j \in N_1^k \ et \ \gamma^k = 0\}.$

$J_2^K = \{j/j \in N_2^k \ et \ \gamma^k = 0\}.$

$T_1^K = \{j/j \in N_1^k \ et \ \gamma^k \neq 0\}.$

$T_2^K = \{j/j \in N_2^k \ et \ \gamma^k \neq 0\}.$

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

4.4.2 Arête incidente[18]

Une arête E^{j_k} incidente à une solution x^k est définit comme suit :

1. Si $j_k \in T_1^k$

$$E^{j_k} = \left\{ X = (x_i) \in \mathbb{R}^{n_k} \;\middle|\; \begin{array}{l} x_{B_i} = x_{B_i} - \phi_{j_k} y_{ij_k}, i \in I^k \\ \\ x_{j_k} = \phi_{j_k} + l_{j_k}, j_k \in T_1^k \\ \\ x_j = l_j, \forall j \in (T_1^k \backslash \{j_k\}) \bigcup J_1^k \\ \\ x_j = u_j, j \in N_2^k \end{array} \right\}$$

Où :

$$\phi_{j_k} = min\{(u_{j_k} - l_{j_k}), (\frac{x_{B_i} - l_{B_i}}{y_{ij_k}}, y_{ij_k} > 0), (\frac{u_{B_i} - x_{B_i}}{-y_{ij_k}}, y_{ij_k} < 0), i \in I^k\} \quad (4.1)$$

2. Si $j_k \in T_2^k$

$$E^{j_k} = \left\{ X = (x_i) \in \mathbb{R}^{n_k} \;\middle|\; \begin{array}{l} x_{B_i} = x_{B_i} + \phi_{j_k} y_{ij_k}, i \in I^k \\ \\ x_{j_k} = u_{j_k} - \phi_{j_k}, j_k \in T_2^k \\ \\ x_j = l_j, j \in N_2^k \\ \\ x_j = u_j, \forall j \in (T_2^k \backslash \{j_k\}) \bigcup J_2^k \end{array} \right\}$$

Où :

$$\phi_{j_k} = min\{(u_{j_k} - l_{j_k}), (\frac{x_{B_i} - l_{B_i}}{-y_{ij_k}}, y_{ij_k} < 0), (\frac{u_{B_i} - x_{B_i}}{y_{ij_k}}, y_{ij_k} > 0), i \in I^k\} \qquad (4.2)$$

4.4.3 Test d'éfficacitée

Le test d'efficacité décrit par Ecker et Kouada dans [13] présenté dans le chapitre 3 peut être appliqué dans le cas des ***problèmes multi-objectifs fractionnaire*** :

Théorème 4.2. *: soit x^* un élément arbitraire de la région D. $x^* \in Eff$ si et seulement si la valeur optimale de la fonction objectif θ est nulle dans le problème de programmation fractionnaire mixte suivant :*

$$
(Px^*)\begin{cases}
Max \quad \theta = \sum_{i=1}^{p} \psi_i \\
\frac{C^1x+\alpha^1}{D^1x+\beta^1} - \psi_1 = \frac{C^1x^*+\alpha^1}{D^1x^*+\beta^1} \\
\frac{C^2x+\alpha^2}{D^2x+\beta^2} - \psi_2 = \frac{C^2x^*+\alpha^2}{D^2x^*+\beta^2} \\
\qquad\qquad \vdots \\
\frac{C^rx+\alpha^r}{D^rx+\beta^r} - \psi_r = \frac{C^rx^*+\alpha^r}{D^rx^*+\beta^r} \\
X \in D \\
\psi_i \geq 0, \forall i = 1, ..., p
\end{cases}
$$

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

Avec : $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, C^i et $D^i \in \mathbb{Z}^{1\times n}$, α^i et $\beta^i \in \mathbb{R}$ $\quad i = \overline{1, r}$. et $\psi = (\psi_i)_{i=1,...,p}$.

Démonstration. Notons par $f_i(x) = \frac{C^ix+\alpha^i}{D^ix+\beta^i}$,$i \in \{1, 2, 3, ..., r\}$

- Soit $(x^*) \in Eff(P)$, comme $\psi_i \geq 0$ pour tout $i \in \{1, 2, 3, ..., r\}$, alors $f_i(x) \geq f_i(x^*)$, $\forall i \in \{1, 2, 3, ..., r\}$ dans $P(x^*)$. Supposons $\exists\, x \in\ D$ et $\psi > 0$ tel que $max\ \theta \neq 0$, alors il existe un i tel que $\psi_i > 0$, ce qui implique $f_i(x) > f_i(x^*)$. Ce qui est en contradiction avec l'hypothèse selon laquelle x^* est efficace.
- Soit $max\theta = 0$; supposons que x^* est non efficace. Il existe un point $x \in D$ tel que $f_i(x) \geq f_i(x^*)$ et $f_i(x) \neq f_i(x^*)$ par conséquent, il existe $x \in D$ tel que $f_i(x) - f_i(x^*) \geq 0$ et $f_i(x) - f_i(x^*) \neq 0$, d'où il existe un i tel que $\psi_i > 0$ qui est en contradiction avec l'hypothèse $max\theta = 0$.

□

4.4.4 Résultats théoriques

Théorème 4.3. *[18] Toute les solutions entières alternatives à la solution courante x^k du problème P appartiennent au demi espace fermé :*

$$\sum_{j\in J_1^k}(x_j - l_j) + \sum_{j\in J_2^k}(u_j - x_j) \geq 1$$

Démonstration. Soit x^k la solution entière réalisable du problème (P),

$$B^K X_{B^k} + \sum_{j\in N_1^k}(a_j l_j) + \sum_{j\in N_2^k}(a_j u_j) = b$$

i.e

$$\sum_{i\in I^k}(a_i x_{B^i}^k) + \sum_{j\in N_1^k}(a_j l_j) + \sum_{j\in N_2^k}(a_j u_j) = b$$

où $X_{B^i}^k = (x_{B^i}^k)$

pour certain $a_{j_1}, j_1 \in J_1^k$,

$$\sum_{i\in I^k}(a_i x_{B^i}^k) + \sum_{j\in N_1^k}(a_j l_j) + \sum_{j\in N_2^k}(a_j u_j) + \phi_{j_1} a_{j_1} - \phi_{j_1} a_{j_1} = b$$

où ϕ_{j_1} est un scalaire non nul.

comme $a_j = B^K Y_j$, en substituant $a_{j_1} = \sum_{i\in I^k} a_i y_{ij_1}$ dans l'équation suivante :

$$\sum_{i\in I^k}(a_i x_{B^i}^k) + \sum_{j\in N_1^k}(a_j l_j) + \sum_{j\in N_2^k}(a_j u_j) + \phi_{j_1} a_{j_1} - \phi_{j_1}[\sum_{i\in I^k} a_i y_{ij_1}] = b$$

$$\sum_{i\in I^k}(a_i x_{B^i}^k - \phi_{j_1} y_{ij_1}) + \sum_{j\in N_1^k\setminus\{j_1\}}(a_j l_j) + \sum_{j\in N_2^k}(a_j u_j) + a_{j_1}(\phi_{j_1} + l_{j_1}) \Rightarrow= b$$

$\Rightarrow Ax^{*k} = b$

où

$$(X^{*k})\begin{cases} x^{*k}_{B_i} = x^k_{B_i} - \phi_{j_1} y_{ij_1}, i \in I^k \\ \\ x^{*k}_{j_1} = \phi_{j_1} + l_{j_1}, j_1 \in J^k_1 \\ \\ x^{*k}_j = l_j, j \in N^k_1 \backslash \{j_1\} \\ \\ x^{*k}_j = u_j, j \in N^k_2 \end{cases}$$

ce qui implique que X^{*k}est une nouvelle solution entière réalisable d'après :

- ϕ_{j_1} est un entier positive.
- $\phi_{j_1} y_{ij_1}$ est entier pour tout $i \in I^k$et
- $l_{B_i} \leq x^{*k}_{B_i} \leq u_{B_i}, \forall i \in I^k \text{et} l_{j_2} \leq x^{*k}_{j_2} \leq u_{j_2}$

on a :

$$\begin{aligned} n^{*k} &= n(X^{*k}) \\ &= C^T_{B^k}[(B^k)^{-1}b] - \sum_{j \in N^k_1} (Z^1_j - c_j) l_j - (Z^1_{j_1} - c_{j_1})(l_{j_1} + \phi_{j_1}) - \sum_{j \in N^k_2} (Z^1_j - c_j) u_j + \alpha \\ &= [C^T_{B^k}[(B^k)^{-1}b] - \sum_{j \in N^k_1} (Z^1_j - c_j) l_j - \sum_{j \in N^k_2} (Z^1_j - c_j) u_j + \alpha] - \phi_{j_1}(Z^1_{j_1} - c_{j_1}) \\ &= n^k - \phi_{j_1}(Z^1_{j_1}) - c_{j_1} \end{aligned}$$

de la même façon on obtient : $d^{*k} = d(X^{*k}) = d^k - \phi_{j_1}(Z^2_{j_1} - c_{j_1}$

$$Z(X^{*k}) = \frac{n^{*k}}{d^{*k}} = \frac{n^k - \phi_{j_1}(Z^1_{j_1} - c_{j_1}}{d^k - \phi_{j_1}(Z^2_{j_1} - d_{j_1}}$$

$$\begin{aligned} Z(X^{*k}) - Z(X^k) &= \frac{n^k - \phi_{j_1}(Z^1_{j_1} - c_{j_1}}{d^k - \phi_{j_1}(Z^2_{j_1} - d_{j_1}} - \frac{n^k}{d^k} \\ &= \frac{\phi_{j_1}[n^k(Z^2_{j_1} - d_{j_1}) - d^k(Z^1_{j_1} - c_{j_1})]}{d^k(d^k - \phi_{j_i}(Z^2_{j_1} - d_{j_1}))} \\ &= \frac{\phi_{j_1}\delta^k_{j_1}}{d^k(d^k - \phi_{j_i}(Z^2_{j_1} - d_{j_1}))} \\ &= 0 \end{aligned}$$

Ce qui implique que X^{*k} est une solution alternative à X^k. Donc X^{*k} satisfait :

$$\sum_{j\in J_1^k}(x_j - l_j) \geq 1 \tag{4.3}$$

pour certain $a_{j_2}, j_2 \in J_2^k$,

$$\sum_{i\in I^k}(a_i x_{B^i}^k) + \sum_{j\in N_1^k}(a_j l_j) + \sum_{j\in N_2^k}(a_j u_j) + \phi_{j_2}a_{j_2} - \phi_{j_2}a_{j_2} = b$$

où ϕ_{j_1} est un scalaire non nul.

Comme $a_j = B^K Y_j$, en substituant$a_{j_2} = \sum_{i\in I^k} a_i y_{ij_2}$ dans l'équation suivante :

$$\sum_{i\in I^k}(a_i x_{B^i}^k) + \sum_{j\in N_1^k}(a_j l_j) + \sum_{j\in N_2^k}(a_j u_j) - \phi_{j_2}a_{j_2} + \phi_{j_2}[\sum_{i\in I^k} a_i y_{ij_2}] = b$$

$$\sum_{i\in I^k}(a_i x_{B^i}^k + \phi_{j_2}y_{ij_2}) + \sum_{j\in N_1^k}(a_j l_j) + \sum_{j\in N_2^k\backslash\{j_2\}}(a_j u_j) + a_{j_2}(u_{j_2} - \phi_{j_2}) = b$$

$$\Rightarrow Ax^{*k} = b$$

où

$$(X^{*k}) \begin{cases} x_{B_i}^{*k} = x_{B_i}^k + \phi_{j_2}y_{ij_2}, i \in I^k \\ x_{j_1}^{*k} = u_{j_2} - \phi_{j_2}, j_2 \in J_2^k \\ x_j^{*k} = l_j, j \in N_1^k \\ x_j^{*k} = u_j, j \in N_2^k\backslash\{j_2\} \end{cases}$$

Ce qui implique que X^{*k}est une nouvelle solution entière réalisable d'après :

- ϕ_{j_2} est un entier positive.
- $\phi_{j_2}y_{ij_2}$ est entier pour tout $i \in I^k$et
- $l_{B_i} \leq x_{B_i}^{*k} \leq u_{B_i}, \forall i \in I^ketl_{j_1} \leq x_{j_1}^{*k} \leq u_{j_1}$

On a :

$$\begin{aligned}
n^{*k} &= n(X^{*k}) \\
&= C_{B^k}^T[(B^k)^{-1}b] - \sum_{j\in N_1^k\setminus\{j_1\}} (Z_j^1 - c_j)l_j - (Z_{j_2}^1 - c_{j_2})(u_{j_2} - \phi_{j_2}) - \sum_{j\in N_2^k\setminus\{j_2\}} (Z_j^1 - c_j)u_j + \alpha \\
&= [C_{B^k}^T[(B^k)^{-1}b] - \sum_{j\in N_1^k} (Z_j^1 - c_j)l_j - \sum_{j\in N_2^k} (Z_j^1 - c_j)u_j + \alpha] + \phi_{j_2}(Z_{j_2}^1 - c_{j_2}) \\
&= n^k + \phi_{j_2}(Z_{j_2}^1) - c_{j_2}
\end{aligned}$$

De la même façon on obtient : $d^{*k} = d(X^{*k}) = d^k + \phi_{j_2}(Z_{j_2}^2 - c_{j_2}$

$$Z(X^{*k}) = \frac{n^{*k}}{d^{*k}} = \frac{n^k - \phi_{j_1}(Z_{j_1}^1 - c_{j_1}}{d^k - \phi_{j_1}(Z_{j_1}^2 - d_{j_1}}$$

$$\begin{aligned}
Z(X^{*k}) - Z(X^k) &= \frac{n^k + \phi_{j_2}(Z_{j_2}^1 - c_{j_2}}{d^k + \phi_{j_2}(Z_{j_2}^2 - d_{j_2}} - \frac{n^k}{d^k} \\
&= -\frac{\phi_{j_2}[n^k(Z_{j_2}^2 - d_{j_2}) - d^k(Z_{j_2}^1 - c_{j_2})]}{d^k(d^k + \phi_{j_2}(Z_{j_2}^2 - d_{j_2}))} \\
&= -\frac{\phi_{j_2}\delta_{j_2}^k}{d^k(d^k + \phi_{j_2}(Z_{j_2}^2 - d_{j_2}))} \\
&= 0
\end{aligned}$$

Ce qui implique que X^{*k} est une solution alternative à X^k. Donc X^{*k} satisfait :

$$\sum_{j\in J_2^k} (u_j - x_j) \geq 1 \tag{4.4}$$

En combinant (4.3) et (4.4) en obtient :

$$\sum_{j\in J_1^k} (x_j - l_j) + \sum_{j\in J_2^k} (u_j - x_j) \geq 1$$

□

Corrolaire 4.2. *[18] Toute les solutions entières alternatives à la solution courante*

x^k du problèmr P appartiennent au demi espace fermé :

$$\sum_{j\in T_1^k}(x_j - l_j) + \sum_{j\in T_2^k}(u_j - x_j) < 1$$

Démonstration. Pour toute solution entière réalisable du problème (P_1) alternative à X^k, on a $x_j = l_j$ pour tous $j \in T_1^k$ et $x_j = u_j$ pour tous $j \in T_2^k$ □

Théorème 4.4. *[18] Toute les solutions entières réalisables du problème (P_1) qui ont un valeur de l'objectif inférieur de $Z(X^k)$ sont dans le demi espace fermé*

$$\sum_{j\in T_1^k}(x_j - l_j) + \sum_{j\in T_2^k}(u_j - x_j) \geq 1$$

Démonstration. D'une solution entière réalisable X^k sur $S(k)$ à la k^{me} étape, on peut obtenir une autre solution $\hat{X}^k$ en entrant une certaine variable $x_r, r \in T_1^k$ ou bien $x_s, s \in T_2^k$.

Soit $x_r, r \in T_1^k$ la variable qui va renter dans la base,la nouvelle solution $\hat{X}^k$est donnée par :

$$(\hat{X}^k)\begin{cases}\hat{x}_{B_i}^k = x_{B_i}^k - \phi_r y_{ir}, i \in I^k \\ \hat{x}_r^k = \phi_r + l_r, r \in T_1^k \\ \hat{x}_j^k = l_j, j \in (T_1^k \backslash \{r\}) \bigcup J_1^k \\ \hat{x}_j^k = u_j, j \in N_2^k \end{cases}$$

$\hat{X}^k$ est une solution entière car :

1. $\phi_r = min\{(u_r - l_r), (\frac{x_{B_i} - l_{B_i}}{y_{ir}}, \quad y_{ir} > 0, i \in I^k), (\frac{u_{B_i} - x_{B_i}}{-y_{ir}}, y_{ir} < 0, i \in I^k)\}$
2. ϕ_r est un entier positive et
3. $\phi_r y_{ir}$ est entier pour tout $i \in I^k$

alors $\hat{X}^k$ satisfait

$$\sum_{j\in T_1^k}(x_j - l_j) \geq 1$$

$$\begin{aligned} Z(\hat{X}^k) - Z(X^k) &= \frac{n^k - \phi_r(Z_r^1 - c_r}{d^k - \phi_r(Z_r^2 - d_{j_2}} - \frac{n^k}{d^k} \\ &= \frac{\phi_r[n^k(Z_r^2 - d_r) - d^k(Z_r^1 - c_r)]}{d^k(d^k - \phi_r(Z_r^2 - d_r))} \\ &= \frac{\phi_r \delta_r^k}{d^k(d^k - \phi_r(Z_r^2 - d_r))} \\ &< 0 (car \delta_r^k < 0 \quad pour \quad r \in T_1^k) \end{aligned}$$

$$\text{d'où } Z(\bar{X}^k) < Z(X^k)$$

Même pour toute solution réalisable entière obtenu à partir de X^k en se déplaçant dans la direction de $x_r, r \in T_1^k$ aura une valeur de l'objectif inférieur à celle de X^k et se trouve dans le demi espace fermé :

$$\sum_{j \in T_1^k} (x_j - l_j) \geq 1 \tag{4.5}$$

Soit $x_s, r \in T_2^k$ la variable qui va renter dans la base, la nouvelle solution $\hat{X}^k$est donnée par :

$$(\bar{X}^k) \begin{cases} \bar{X}_{B_i}^k = x_{B_i}^k + \phi_s y_{is}, i \in I^k \\ \\ \bar{X}_r^k = u_s - \phi_s, s \in T_2^k \\ \\ \bar{X}_j^k = l_j, j \in N_1^k \\ \\ \bar{X}_j^k = u_j, j \in (T_2^k \backslash \{s\}) \bigcup J_2^k \end{cases}$$

$\bar{X}^k$ est une solution entière car :

1. $\phi_s = min\{(u_s - l_s), (\frac{x_{B_i} - l_{B_i}}{-y_{is}}, y_{is} < 0, i \in I^k), (\frac{u_{B_i} - x_{B_i}}{y_{ir}}, y_{ir} > 0, i \in I^k)\}$

2. ϕ_s est un entier positive et

3. $\phi_s y_{is}$ est entier pour tout $i \in I^k$

Alors $\bar{X}^k$ satisfait

$$\sum_{j \in T_2^k} (l_j - x_j) \geq 1$$

$$\begin{aligned} Z(\bar{X}^k) - Z(X^k) &= \frac{n^k + \phi_s(Z_s^1 - c_s}{d^k + \phi_s(Z_s^2 - d_s} - \frac{n^k}{d^k} \\ &= -\frac{\phi_s[n^k(Z_s^2 - d_s) - d^k(Z_s^1 - c_s)]}{d^k(d^k + \phi_s(Z_s^2 - d_s))} \\ &= -\frac{\phi_s \delta_s^k}{d^k(d^k - \phi_s(Z_s^2 - d_s))} \\ &< 0(car\delta_s^k > 0 \quad pour \quad s \in T_2^k) \end{aligned}$$

$$\text{d'où } Z(\bar{X}^k) < Z(X^k)$$

Même pour toute solution réalisable entière obtenu à partir de X^k en se déplaçant dans la direction de $x_s, s \in T_1^k$ aura une valeur de l'objectif inférieur à celle de X^k et se trouve dans le demi espace fermé :

$$\sum_{j \in T_2^k} (u_j - x_j) \geq 1 \tag{4.6}$$

en combinant (4.5) et (4.6), on obtient :

$$\sum_{j \in T_1^k} (x_j - l_j) + \sum_{j \in T_2^k} (u_j - x_j) \geq 1$$

□

4.4.5 Description de la méthode

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

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

Résoudre P_1 par la méthode décrite dans le chapitre(2)pour les problèmes fractionnaires mono-objectif en variables bornées. Construire les ensembles J_1, J_2.

Cas 1. Si $J_1^1 = \emptyset, et J_2^1 = \emptyset$, alors la solution optimale de (P_1), soit x^1 est unique. Tester l'efficacité de x^1 .

- Si x^1 est efficace, mettre à jour l'ensemble $Eff(P)$.
- Rajouter la coupe :

$$\sum_{j\in T_1^1}(x_j - l_j) + \sum_{j\in T_2^1}(u_j - x_j) \geq 1$$

aller à l'étape 2.

Cas 2. Sinon, la solution optimale de (P_1) n'est pas unique. Déterminer dans ce cas toutes les solutions qui lui sont alternatives, éliminer celles qui ne sont pas efficaces et mettre à jour l'ensemble $Eff(P)$.

Appliquer la coupe suivante pour éliminer toutes les solutions alternatives trouvées à cette étape :

$$\sum_{j\in T_1^1}(x_j - l_j) + \sum_{j\in T_2^1}(u_j - x_j) \geq 1$$

et utiliser la méthode dual du simplexe pour les problèmes fractionnaires en variables bornées pour obtenir une solution optimale entière, aller à l'étape 2.

Étape k ; $k \geq 2$. Construire les ensembles J_1^{k-1}, J_2^{k-1}.

Cas 1. Si $J_1^{k-1} = \emptyset, et J_2^{k-1} = \emptyset$, alors la solution optimale de (P_{k-1}), soit x^{k-1} de (P_{k-1}) est unique. Tester l'efficacité de x^{k-1} .

- Si x^{k-1} est efficace, mettre à jour l'ensemble $Eff(P)$.
- Rajouter la coupe :

$$\sum_{j\in T_1^k}(x_j - l_j) + \sum_{j\in T_2^k}(u_j - x_j) \geq 1$$

aller à l'étape $k + 1$.

Cas 2. Sinon, la solution optimale de (P_{k-1}) n'est pas unique. Déterminer dans ce cas toutes les solutions qui lui sont alternatives, éliminer celles qui ne sont pas efficaces et mettre à jour l'ensemble $Eff(P)$.

Appliquer la coupe suivante pour éliminer toutes les solutions alternatives trouvées à cette étape :

$$\sum_{j\in T_1^k}(x_j - l_j) + \sum_{j\in T_2^k}(u_j - x_j) \geq 1$$

et utiliser la méthode dual du simplexe pour les problèmes fractionnaires en variables bornées pour obtenir une solution optimale entière, aller à l'étape $k+1$.

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

Théorème 4.5. *L'algorithme décrit ci-dessus converge en un nombre fini d'itérations et donne l'ensemble de toute les solutions efficaces.*

Démonstration. L'ensemble D des solutions réalisables de $(MOILFPB)$ étant fermé et borné, il contient un nombre fini de solutions entières. A chaque étape k de l'algorithme au moins la solution entière x^k générée est éliminée, et éventuellement toute les solutions entières incidente à x^k après avoir tester leurs efficacité. De plus aucune solution réalisable du problème initiale (P) n'est raté en rajoutant la coupe :

$$\sum_{j \in T_1^k} (x_j - l_j) + \sum_{j \in T_2^k} (u_j - x_j) \geq 1$$

□

4.4.6 Exemples numériques

Exemple1

Pour illustrer l'algorithme précédent, nous considérons le problème de programmation fractionnaire linéaire en nombres entiers bornés à trois objectifs suivant :

$$(P) \begin{cases} Max \quad z_1(x) = \dfrac{5x_1 + 7x_2 + 4}{2x_1 + 8x_2 + 1} \\ Max \quad z_2(x) = \dfrac{5x_1 + 8x_2 + 9}{8x_1 + 10x_2 + 4} \\ Max \quad z_3(x) = \dfrac{7x_1 + 2x_2 + 6}{8x_1 + 10x_2 + 7} \\ 7x_1 + 10x_2 \leq 34 \\ 7x_1 + 6x_2 \leq 26 \\ x_1, x_2 \geq 0 \\ x_1, x_2 \in \mathbb{Z}_+ \\ x_1 \leq 4 \\ x_2 \leq 2 \end{cases}$$

- Étape1

  Résoudre le problème P_1 par la méthode décrite au chapitre 3 pour le problème de programmation discrète fractionnaire en variables bornées :

$$
(P_1)\begin{cases} Max \quad z_1(x) = \dfrac{5x_1 + 7x_2 + 4}{2x_1 + 8x_2 + 1} \\ 7x_1 + 10x_2 \leq 34 \\ 7x_1 + 6x_2 \leq 26 \\ x_1, x_2 \geq 0 \\ x_1, x_2 \in \mathbb{Z}_+ \\ x_1 \leq 4 \\ x_2 \leq 2 \end{cases}
$$

La solution optimale réalisable de base du problème P_1 est $x_1^1 = (0, 0)$ donnée par le tableau suivant :

base	x_1	x_2	x_3	x_4	b
x_3	7	10	1	0	34
x_4	7	6	0	1	26
$Z_J^1 - c_j$	-5	-7	0	0	0
$Z_J^2 - d_j$	-2	-8	0	0	0
γ_j	-3	-25	0	0	0

TAB. 4.7 – Tableau 1

$J_1^1 = \emptyset, \quad J_2^1 = \emptyset; \quad T_1^1 = \{1, 2\}, \quad T_2^1 = \emptyset$

Le premier triplet correspondant à x_1^1 est égale à $(4, 9/4, 6/7)$. après avoir appliqué le test d'efficacité on déduit que x_1^1 est efficace, donc $Eff_0 = \{(0, 0)\}$

- Étape2

$J_1^1 = \emptyset, \quad J_2^1 = \emptyset$, cela signifie que x_1^1 est unique. Ce point est tronqué par la coupe : $x_1 + x_2 \geq 1$, i.e $-x_1 - x_2 + x_5 = -1$.

En utilisant l'algorithme dual du simplexe pour les problèmes fractionnaire en variables bornée, on obtient une solution réalisable entière $x_2^1 = (1, 0)$ donnée par le tableau 2 optimale suivant :

base	x_1	x_2	x_3	x_4	x_5	b
x_3	0	3	1	0	7	27
x_4	0	-1	0	1	7	19
x_1	1	1	0	0	-1	1
$Z_J^1 - c_j$	0	-2	0	0	-5	0
$Z_J^2 - d_j$	0	-6	0	0	-2	0
γ_j	0	-48	0	0	-3	0

TAB. 4.8 – Tableau 2

$J_1^2 = \emptyset, \quad J_2^2 = \emptyset; \quad T_1^2 = \{2, 5\}, \quad T_2^2 = \emptyset$

Après avoir appliqué le test d'efficacité on déduit que x_1^2 est efficace , Par conséquent $Eff_1 = \{(0,0),(1,0)\}$,

- Étape3

$J_1^2 = \emptyset$, $J_2^2 = \emptyset$, x_2^1 est une solution unique. Ce point est tronqué par la coupe : $x_2 + x_5 \geq 1$, i.e.$-x_2 - x_5 + x_6 = -1$.

L'application de la méthode dual du simplexe produit la solution $x_1^3 = (2, 0)$ donnée par le tableau suivant :

base	x_1	x_2	x_3	x_4	x_5	x_6	b
x_3	0	-4	1	0	0	7	20
x_4	0	-8	0	1	0	7	12
x_1	1	2	0	0	0	-1	2
x_5	0	1	0	0	1	-1	1
$Z_J^1 - c_j$	0	3	0	0	0	-5	0
$Z_J^2 - d_j$	0	-4	0	0	0	-2	0
γ_j	0	-71	0	0	0	-3	0

TAB. 4.9 – Tableau 3

$J_1^3 = \emptyset, \quad J_2^3 = \emptyset; \quad T_1^3 = \{2, 6\}, \quad T_2^3 = \emptyset$

Après avoir appliqué le test d'efficacité on déduit que x_1^3 est efficace , par conséquent $Eff_2 = \{(0,0),(1,0),(2,0)\}$

- Étape4

$J_1^3 = \emptyset, \quad J_2^3 = \emptyset$, x_3^1 est une solution unique. Ce point est tronqué par la coupe : $x_2 + x_6 \geq 1$, i.e.$-x_2 - x_6 + x_7 = -1$.

L'application de la méthode dual du simplexe produit la solution $x_1^4 = (3,0)$ donnée par le tableau suivant :

base	x_1	x_2	x_3	x_4	x_5	x_6	x_7	b
x_3	0	-11	1	0	0	0	7	13
x_4	0	-15	0	1	0	0	7	5
x_1	1	3	0	0	0	0	-1	3
x_5	0	2	0	0	1	0	-1	2
x_6	0	1	0	0	0	1	-1	1
$Z_J^1 - c_j$	0	2	0	0	0	0	-5	0
$Z_J^2 - d_j$	0	-8	0	0	0	0	-2	0
γ_j	0	-94	0	0	0	0	-3	0

TAB. 4.10 – Tableau 4

$J_1^4 = \emptyset, \quad J_2^4 = \emptyset; \quad T_1^4 = \{2,7\}, \quad T_2^4 = \emptyset$

Après avoir appliqué le test d'efficacité on déduit que x_1^3 est efficace , Par conséquent $Eff_3 = \{(0,0),(1,0),(2,0),(3,0)\}$

- Étape5

$J_1^4 = \emptyset, \quad J_2^4 = \emptyset$,$x_3^1$ est une solution unique. Ce point est tronqué par la coupe :$x_2 + x_7 \geq 1$,i.e.$-x_2 - x_7 + x_8 = -1$.

Lors de l'application de la méthode dual du simplexe l'opération pivot devient impossible indiquant que le domaine tronqué ne contient aucun point réalisable et que toute les solutions efficaces du problème ont été obtenues. D'ou l'ensemble des solutions efficaces est

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

Et si on change les bornes des variables comme suit, on aura un autre ensemble de solutions efficaces. exemple

$$(P)\begin{cases} Max \quad z_1(x) = \dfrac{5x_1 + 7x_2 + 4}{2x_1 + 8x_2 + 1} \\ Max \quad z_2(x) = \dfrac{5x_1 + 8x_2 + 9}{8x_1 + 10x_2 + 4} \\ Max \quad z_3(x) = \dfrac{7x_1 + 2x_2 + 6}{8x_1 + 10x_2 + 7} \\ 7x_1 + 10x_2 \leq 34 \\ 7x_1 + 6x_2 \leq 26 \\ x_1, x_2 \geq 0 \\ x_1, x_2 \in \mathbb{Z}_+ \\ x_1 \leq 2 \\ x_2 \leq 1 \end{cases}$$

on obtiendra l'ensemble des solutions efficaces suivant :$Eff(P) = \{(0,0),(1,0),(2,0)\}$

Exemple2

En appliquant notre méthode sur cette exemple

$$(P)\begin{cases} Max \quad z_1(x) = \dfrac{x_1 - 4}{-x_2 + 3} \\ Max \quad z_2(x) = \dfrac{-x_1 + 4}{x_2 + 1} \\ Max \quad z_3 = C^3 = -x_1 + x_2 \\ -x_1 + 4x_2 \leq 0 \\ x_1 - 1/2x_2 \leq 4 \\ x_1, x_2 \geq 0 \\ x_1 \leq 4 \\ x_2 \leq 2 \\ x_1, x_2 \in \mathbb{Z}_+ \end{cases}$$

On obtiendra l'ensemble des solutions efficaces $Eff(P) = \{(4,1),(3,0),(2,0),(1,0),(0,0)\}$, et si on change les bornes des variables comme suit

$$(P)\begin{cases} Max \quad z_1(x) = \dfrac{x_1 - 4}{-x_2 + 3} \\ Max \quad z_2(x) = \dfrac{-x_1 + 4}{x_2 + 1} \\ Max \quad z_3 = C^3 = -x_1 + x_2 \\ -x_1 + 4x_2 \leq 0 \\ x_1 - 1/2x_2 \leq 4 \\ x_1, x_2 \geq 0 \\ x_1 \leq 2 \\ x_2 \leq 1 \\ x_1, x_2 \in \mathbb{Z}_+ \end{cases}$$

On obtiendra l'ensemble des solutions efficaces $Eff(P) = \{(2,0),(1,0),(0,0)\}$

4.5 Conclusion

Dans ce chapitre nous nous somme intéressés au problème de l'optimisation fractionnaire discrète multi-objectifs en variables bornées ($MOILFPB$). Nous avons proposé une méthode pour la détermination de toutes les solutions efficaces de ce problème.

Conclusion générale

Les problèmes d'optimisation issus de la réalité sont, la plupart du temps, de nature multi-objectifs car, plusieurs critères, souvent contradictoires, sont à considérer simultanément. Pour ce type de problèmes, la notion d'optimalité disparaît au profit de la notion d'efficacité. Il s'agit de chercher un ensemble de solutions réalisant le meilleur compromis entre les critères considérés.

Dans ce mémoire, on s'est focalisé sur les problèmes de programmation discrète multi-objectifs fractionnaire linéaire en variables bornées et on s'est intéressé à la recherche d'une méthode pour la détermination de l'ensemble des solutions efficaces du problème considéré.
Nous avons commencé ce travail par introduire les notions de base concernant la programmation linéaire mono-objectif.

Nous nous somme ensuite penché sur la description du problème multi-objectifs linéaire ainsi que quelques méthodes de résolutions.

Afin d'aborder la problématique de la programmation multi-objectifs fractionnaire linéaire discrète en variables bornées ($MOIFLPB$), nous avons jugé utile de rappeler des notions de la programmation fractionnaire uni-critère. Pour ce faire, nous avons fait appel à une méthode récemment développée par Kalpana et Vatina [18] pour la résolution d'un programme fractionnaire linéaire à variables bornées. Un programme esclave est alors appelé pour tester l'efficacité de telles solutions. Ceci est rendu possible après avoir généraliser le test d'efficacité d'Ecker et Kouada [13] pour les problèmes linéaires au cas des problèmes fractionnaires. L'adjonction de coupes adéquates nous a permis de mettre au point une nouvelle méthode pour la résolution du problème $(MOILFPB)$. Un algorithme est alors décrit pour trouver l'ensemble des solutions efficaces du problème $(MOILFPB)$ en un nombre fini d'itérations. L'exposé détaillé des étapes de l'algorithme sur un exemple numérique est illustrée.

Un programme informatique écrit à l'aide du langage de programmation $MATLAB$ mettant en oeuvre cet algorithme, a été testé avec succès sur différents exemples.

Bibliographie

[1] M. Abbas and M. Moulaï, *Solving multiple objective integer linear programming*, Journal of the Italian Operations Research Society (Ricerca Operativa) 29, pp. 15-38, (1999).

[2] M. Abbas and D. Chaabane, *An algorithm for solving multiple objective integer linear programming problem*, RAIRO Operations Research 36, pp. 351-364, (2002).

[3] G.R. Bitran and G. Novaes, *Linear Programming with a fractional objective function*, Operations Research 24, pp. 675-699, (1976).

[4] D. Chaabane, *Contribution à l'optimisation multicritère en variables discrètes*, thèse de doctorat, Université des Sciences et de la Technologie Houari Boumediene, Algérie, (2005).

[5] A. Cambini and L. Martein, *A modified version of Martos's algorithm for the linear fractional problem*, Methods of Oper. Res. 53, pp. 33-44, (1986).

[6] V. Chankong and Y.Y. Haimes *Multiobjective decision making : theory and methodology*, North Holland Series in System Science and Engeneering 8, p. 406, (1983).

[7] A. Charnes and W.W. Cooper, *Programming with linear fractional functionals*, Naval Res. Logist. Quart. 9, pp. 181-186, (1962).

[8] A. Crema and J. Sylva, *A method for finding well-dispersed subsets of non-dominated vectors for multiple objective mixed integer linear programs*, European

Journal of Operational Research, (2006).

[9] R. Dakin, *A tree search algorithm for mixed integer programming problems*, Computer Journal 8, pp. 250-255, (1965).

[10] G.B. Dantzig, On a Linear Programming Combinatorial Approach to the Traveling Salesman Problem, Operations Research, 7, pp.58-66,(1959).

[11] G.B. Dantzig, *Linear programming and extention*, Princeton University Press, (1963).

[12] W. Dinkelbach, *On nonlinear fractional programming*, Management Science 13, pp. 492-498, (1967).

[13] J.G. Ecker and I.A. Kouada, *Finding efficient points for multi-objective linear programs* , Mathematical programming 8, pp. 375-377, (1975).

[14] R.E. Gomory, *Outline of an algorithm for integer solutions to linear programs*, Bulletin of the AMS 64, pp. 275-278, (1958).

[15] D. Granot and F. Granot, *On integer and mixed integer fractional programming problems*, Ann. Discrete Math 1, pp. 221-231, (1977).

[16] M. Grunspan, *Fractional programming : A survey*, Technical Report 50, Department of Industrial and Systems Engineering, University of Florida, (1971).

[17] R. Gupta and R. Malhotra, *Multi-criteria integer linear fractional programming problem*, Optimization 35, pp. 373-389, (1995).

[18] D. Kalpana and V. Vatina, *Ranking in integer programming with bounded variables*, Journal of the Indian Operations Research Society 42, pp. 252-279, (2005).

[19] D. Kalpana and V. Vatina, *Post-optimality in bounded variables problem*, Journal of the Indian Operations Research Society .

[20] D. klein and E. Hannan, *An algorithm for multiple objective integer linear programming problem*, European Journal of Operational Research 9, pp. 378-385, (1982).

[21] A.H. Land and A.G. Doig, *An automatic method for solving discrete programming problems*, Econometrica 28, pp. 497-520, (1960).

[22] B. Martos, *Hyperbolic programming*, Naval Res. Logist. Quart 11, pp. 135-155, (1964).

[23] M. Moulaï, *Optimisation multicritère fractionnaire linéaire en nombres entiers*, thèse de doctorat, Université des Sciences et de la Technologie Houari Boumediene, Algérie, octobre (2002).

[24] M. Moulaï and M. Abbas, *Integer linear fractional programming with multiple objective*, Journal of the Italian Operations Research Society, vol 1 N°1. 103-104, pp. 15-38, (2002).

[25] S.S. Rao, *Engineering optimization : theory and practice* , Wiley, John & Sons, Incorporated, (1996).

[26] B. Roy, *Des critères multiples en recherche opérationnelle : pourquoi ?*, in Rand (ed.), Operational Research 87, North-Holland, Amsterdam, pp. 829-842, (1988).

[27] M. Saad Omar and B. Hughes John, *Bicriterion integer linear fractional programs with parameters in the objective functions*, J. Inform. Optim. Sci. 19, pp. 97-108, (1998).

[28] M. Sakarovitch, *Optimisation combinatoire. Méthodes mathématiques et algorithmiques. Tome 1 : Graphes et Programmation Linéaire ; Tome 2 : Programmation Discrète*, Paris, Hermann, (1984).

[29] C.R. Seshan and V.G. Tikekar, *Algorithms for Integer Fractional Programming*, J. of the Indian Institute of Science 62 (B), pp. 9-16, (1980).

[30] M. Simonnard, *Programmation linéaire : technique du calcul économique*, Paris, (1972).

[31] R.E. Steuer, *Multiple criteria optimization : theory, computation and application*, Wiley Series in Probability and Mathematical Statistics : Applied Probability and Statistics. New York : John Wiley & Sons, Inc. XXII, p. 546, (1986).

[32] J. Sylva,A. Crema *A method for finding the set of non-dominated vectors for multiple objective integer linear programs*, European Journal of Operational Research, 158, pp. 46-55,(2004).

[33] E.L. Ulungu, Optimisation combinatoire multicritère : Détermination de l'Ensemble des Solutions Efficaces et Méthodes Interactives, thèse de doctorat, université de Mons-Hainaut, octobre (1993).