

**MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET
DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOULOU D MAMMERI, TIZI-OUZOU
FACULTE DES SCIENCES
DEPARTEMENT DE MATHEMATIQUES**



Mémoire de fin d'étude
En vue d'obtention du diplôme master
Spécialité: Recherche Opérationnelle

Thème
Thème

Programmation mathématique multi-objectif
et applications

Proposé et dirigé par :
Mr : Chebbah Mohamed

Devant le jury :
Mr : Ouanes

Mr : Goubi

Réalisé par :
Mr : Azoug Massinissa
Melle : Moussi Samira

PROMOTION 2018/2019

REMERCIEMENT

- ❖ *Nous remercions **dieu** tout puissant d'avoir guidé nos pas vers les portes du savoir tout en illuminant notre chemin, et nous avoir donné suffisamment de courage et de persévérance pour mener notre travail à terme.*
- ❖ *Nous tenons à adresser nos plus vifs remerciements à notre promoteur Monsieur **Chebbah Mohamed** qui a su nous guider et nous orienter. Et nous tenons à lui exprimer notre profonde reconnaissance pour son très bon encadrement, sa patience, sa disponibilité, nous le remercions pour le temps qu'il a consacré à notre mémoire, sa confiance de nous proposer un thème très intéressant.*
- ❖ *Nous remercions aussi les membres de jury qui nous ont fait l'honneur d'en faire partie et qui ont eu la patience de nous écouter.*
- ❖ *Nous ne saurons oublier le grand mérite des enseignants qui ont contribué à notre cursus particulièrement ceux du département "mathématiques" et qu'ils trouvent ici le témoignage de notre profonde reconnaissance.*
- ❖ *Enfin, nous remercions tous ceux qui nous ont encouragé tout au long de notre parcours universitaire et ceux qui ont contribué de près ou de loin à la réalisation de ce mémoire.*

DÉDICACES

Je dédie mon modeste travail :

A mes chers parents, que nulle dédicace ne puisse exprimer ce que je leur dois, pour leur bienveillance, leur affection et leur soutien... Trésors de bonté de générosité et de tendresse, en témoignage de mon profond amour et ma grande reconnaissance " Que Dieu vous garde ".

A ma grande mère et à toute ma famille.

Je remercie très chaleureusement mon binôme pour sa gentillesse, et sa compréhension, Et tous mes amis : Riyou, Mehdi, Djounadi, Lil K ,Cholotou ,et tous mes proches

A tous ceux qui m'aiment.

A toute la promotion de recherche opérationnelle 2018/2019.

MASSINISSA

DÉDICACES

Je dédie mon modeste travail :

A mes chers parents, que nulle dédicace ne puisse exprimer ce que je leur dois, pour leur bienveillance, leur affection et leur soutien... Trésors de bonté, de générosité et de tendresse, en témoignage de mon profond amour et ma grande reconnaissance " Que Dieu vous garde ".

A ma grande mère, à mon cher frère Madjid, mes chères sœurs Kahina et son mari sans oublier sa petite Kayla, et Wissam, et ma chère amie haoua, ainsi que toute ma famille.

Je remercie très chaleureusement mon binôme pour sa gentillesse, et sa compréhension, Et toutes mes amies

A tous ceux qui m'aiment.

A toute la promotion de recherche opérationnelle 2018/2019.

SAMIRA

Table des Matières

INTRODUCTION

Chapitre1 : Introduction à la programmation linéaire

1.1. Introduction.....	1
1.2. Forme d'un programme linéaire.....	3
1.3. La méthode du simplexe.....	4
1.4. La M-méthode.....	9
1.5. La méthode duale du simplexe.....	11
1.6. La méthode de Branch and Bound.....	15
1.7. La méthode de Balas.....	17
1.8. La méthode des coupes (Gomory).....	21

Chapitre 2 : Programmation multi objectif

2.1. Introduction.....	25
2.2. Définitions	25
2.3. Dominance	27
2.4. Efficacité	28
2.5. Optimalité au sens de Pareto	29
2.6 Caractérisation des solutions efficaces	31
2.6.1 Ecker et Wendell.....	32
2.6.2 Benson	32
2.7. Points spéciaux et matrice des gains.....	33
2.8. Méthode de résolution d'un programme multi-objectif.....	35
2.8.1 Méthode de simplexe multi-objectif.....	35
2.8.2 Méthode d'agrégation des objectifs.....	43
2.8.3 Méthode de ε -contraintes.....	44

2.9. conclusion	45
-----------------------	----

Chapitre 3 : Programmation multi-objectif linéaire en nombre entier

3.1. Introduction.....	46
3.2 Formulation du problème.....	46
3.3 Définitions et théorèmes	46
3.4. Quelques méthodes de résolution d'un problème multi-objectif linéaire en nombres entiers.....	48
3.4.1 Méthode de Gupta	48
3.4.2 Méthode de Abbas et Moulai	53

Chapitre 4 : Programmation multi-objectif linéaire fractionnaire en nombres entiers

4.1 Introduction	57
4.2 Programmation fractionnaire linéaire uni-critère	57
4.2.1 Stratégie de résolution	58
4.2.2 Représentation graphique de la programmation fractionnaire linéaire	58
4.2.3 Méthode de A. Cambini et al	61
4.3 Programmation multi-objectif fractionnaire linéaire en nombre entiers	62
4.3.1 Formulation du problème	62
4.3.2 Méthodes de résolution d'un problème d'optimisation fractionnaire linéaire multicritère en nombres	63

❖ La méthode de MOULA ĨMustapha.....	63
--------------------------------------	----

Chapitre 5 : Logiciel (LINGO / MATLAB) et applications

• LINGO	69
A) Introduction.....	69
B) Présentation du logiciel LINGO.....	69
C) Interface de logiciel	70
D) exemple d'application	73
• MATLAB.....	75
1. Qu'est ce que MATLAB.....	75
2. Implémentation de la méthode	75

Conclusion Générale.....	78
---------------------------------	-----------

Avant propos

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

Elle fait partie des « **aides à la décision** » dans la mesure où elle propose des modèles conceptuels en vue d'analyser et de maîtriser des situations complexes pour permettre aux décideurs de comprendre, d'évaluer les enjeux et d'arbitrer ou de faire les choix les plus efficaces.

Ce domaine fait largement appel au raisonnement mathématique (logique, probabilités, analyse des données) et à la modélisation des processus. Il est fortement lié à l'ingénierie des systèmes, ainsi qu'au management du système d'information.

La Recherche Opérationnelle (R.O) est la discipline des méthodes scientifiques et techniques mathématiques utilisables pour élaborer de meilleures décisions. Elle permet de rationaliser, de simuler et d'optimiser l'architecture et le fonctionnement des systèmes de production ou d'organisation entre autres. C'est aussi la modélisation mathématique des processus de prise de décision.

La modélisation mathématique est le processus de prise de décision par lequel un problème du monde réel est interprété et représenté en équations mathématiques suivant un modèle d'optimisation.

L'optimisation est l'identification de la meilleure configuration possible d'un système selon des critères spécifiques.

La **R.O.** propose des méthodes conceptuelles pour analyser des situations complexes et permet aux décideurs de faire les choix les plus efficaces grâce à :

- Une meilleure compréhension des problèmes.
- La vision complète des données.
- La considération de toutes les solutions possibles.
- Des prédictions prudentes de résultats incluant une évaluation des risques.
- Des outils et des méthodes modernes d'aide à la décision.

La **R.O** apparaît comme une discipline carrefour associant les mathématiques, l'économie et l'informatique entre autres.

Elle est par nature en prise directe sur l'industrie et joue un rôle-clé dans le maintien de la compétitivité

Les apports de la **R.O** sont visibles tout autour de nous et dans les domaines les plus divers : de l'organisation des lignes de production de véhicules à la planification des missions spatiales, voir dans la vie de tous les jours dans l'organisation du traitement des produits recyclables, l'organisation des ramassages scolaires ou la couverture satellite...etc.

Introduction

L'une des principales missions de la recherche opérationnelle est d'aider à prendre des décisions en vue d'une gestion efficace, rationnelle et logique. La modélisation traditionnelle des problèmes de recherche opérationnelle n'a connu que des modèles d'optimisation à objectif unique jusqu'à l'apparition d'une nouvelle vision plus proche de la réalité et qui consiste à modéliser les problèmes de sorte que tous les objectifs pertinents soient pris en considération. Cette philosophie de multiplicité d'objectifs semble très raisonnable et contribue avec une efficacité très forte à l'élaboration de nouvelles méthodes pratiques de gestion plus réalistes que les méthodes classiques.

La résolution d'un problème d'optimisation multi-objectif consiste à trouver la solution Pareto optimale qui répond au mieux aux préférences. Les approches de résolution initiales consistaient à réduire le problème original en un problème mono-objectif. Néanmoins, depuis plusieurs années une des questions fondamentales de l'optimisation multi-objectif s'apparente à l'identification de l'ensemble Pareto optimale, ou à une approximation de celui-ci pour des problèmes difficiles de grande taille. Dans ce contexte, les métaheuristiques en général, et les algorithmes évolutionnaires en particulier, connaissent un intérêt grandissant.

L'optimisation multi-objectif est au cœur de plusieurs domaines, elle est largement utilisée pour modéliser de nombreuses situations comme la conception des systèmes de l'industrie, le secteur de l'énergie, les réseaux ou encore la planification.

Au lieu d'une solution unique, la solution d'un problème d'optimisation multi-objectif est généralement un assortiment de solutions, qui se distinguent par différents compromis réalisés entre les objectifs. Cet assortiment est connu comme Pareto-optimal, les solutions qui le composent sont optimales, dans le sens qu'il n'existe dans l'espace de recherche aucune solution meilleure si tous les objectifs sont considérés simultanément.

Dans le cadre de la programmation mathématique multicritère, nous sommes particulièrement intéressés aux problèmes linéaires et non linéaires pour les variables de décision réelles et /ou entières.

Le but de ce travail est d'essayer de faire une synthèse sur la programmation mathématique multi-objectif, de voir quelques méthodes de résolution et de programmer vers la fin l'un de ces algorithmes traités.

Pour cela on a partitionné le travail en cinq chapitres :

Le chapitre 1, est réservé à l'étude du problème de programmation linéaire mono-objectif. Dans ce chapitre, nous étudions quelques méthodes de résolution comme la méthode du simplexe, la dualité et la méthode duale du simplexe dans des espaces réels et discrets.

Au chapitre 2, nous présentons l'optimisation multi-objectif. Nous introduisons des concepts fondamentaux tels que la formulation du problème, le concept de dominance et d'efficacité, point idéal et point nadir. Nous décrivons aussi les principales approches de résolution pour ces problèmes comme la méthode du simplexe multi-objectifs.

Le chapitre 3 est consacré à la programmation linéaire multi-objectif en nombres entiers. Nous traitons en détail quelques méthodes de résolution de ce type de problèmes en illustrant quelques uns par un exemple numérique.

Au chapitre 4, on parlera de la programmation non linéaire multi-objectif. Ce chapitre est divisé comme suit :

- Programmation fractionnaire uni-critère.
- Programmation fractionnaire multi-objectif entière.

Le chapitre 5, traite la partie implémentation et description générale de logiciel (LINGO et MATLAB) ainsi que la présentation des résultats obtenus.

Enfin, nous terminons notre travail par une conclusion générale et une bibliographie utilisée.

CHAPITRE 1

Introduction à la programmation linéaire

1.1. Introduction

La programmation linéaire (PL) est une branche de la programmation mathématique (résolution de programmes économiques ou autres à l'aide des mathématiques). Trois types de problèmes relèvent de la programmation mathématique :

- La programmation linéaire, où les fonctions données sont linéaires.
- La programmation quadratique.
- La programmation non linéaire, où une partie des fonctions données sont représentées sous la forme de fonctions non linéaires.

Quelques rappels :

Tout problème de la programmation linéaire peut se formuler de la manière suivante :

Trouver les valeurs des variables $x = (x_j, j = \overline{1, n})$ qui maximisent ou minimisent la fonction linéaire suivante :

$$Z=Z(x_1, x_2, \dots, x_n) = c_1x_1 + c_2x_2 + \dots + c_jx_j + \dots + c_nx_n = \sum_{j=1}^n c_jx_j \rightarrow (\max / \min) \quad (1)$$

Les sous contraintes suivantes :

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1j}x_j + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2j}x_j + \cdots + a_{2n}x_n = b_2 \\ \quad \quad \quad \cdots \quad \quad \cdots \quad \cdots \quad \cdots \quad \quad \cdots \quad \cdots \\ \quad \quad \quad \cdots \quad \quad \cdots \quad \cdots \quad \cdots \quad \quad \cdots \quad \cdots \\ a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{ij}x_j + \cdots + a_{in}x_n = b_i \\ \quad \quad \quad \cdots \quad \quad \cdots \quad \cdots \quad \cdots \quad \quad \cdots \quad \cdots \\ \quad \quad \quad \cdots \quad \quad \cdots \quad \cdots \quad \cdots \quad \quad \cdots \quad \cdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mj}x_j + \cdots + a_{mn}x_n = b_m \end{array} \right. \quad (2)$$

$$x = (x_j, j = 1, \dots, n)$$

$$x_j \geq 0, j = \overline{1, n} \quad (3)$$

Où c_j , $j=\overline{1,n}$ représentent les coûts des différents produits

Les coefficients c_j et a_{ij} ($i = \overline{1, m}, j = \overline{1, n}$) sont supposés être des nombres réels, en plus on considère que l'entier m est inférieur ou égal à n , tous les nombres b_i ($i = \overline{1, m}$) sont tous positifs ou nuls et le rang du système (2) est inférieur ou égal à m .

Définition 1

- . La fonction Z est appelée fonction objective.
- . Les contraintes (2) sont appelées, contraintes principales, ou essentielles.
- . Les contraintes (3) sont dites directes (non négativité).

Remarque 1

Si l'objectif consiste à minimiser une fonction linéaire :

$$Z = Z(x_1, x_2, \dots, x_n) = c_1 x_1 + c_2 x_2 + \dots + c_j x_j + \dots + c_n x_n$$

Alors on maximisera la fonction linéaire opposée :

$$\bar{Z} = -Z(x_1, x_2, \dots, x_n) = -c_1 x_1 - c_2 x_2 - \dots - c_j x_j - \dots - c_n x_n .$$

Ecriture matricielle d'un problème de programmation linéaire

Pour écrire le problème de programmation linéaire (1)-(2)-(3) sous forme compacte, on utilise la forme matricielle (vectorielle). Pour ce faire on introduit les notations suivantes :

Soient $I = \{1, 2, \dots, m\}$ l'ensemble d'indices lignes et $J = \{1, 2, \dots, n\}$ l'ensemble des indices des colonnes.

Ainsi l'ensemble des variables $x_1, x_2, \dots, x_j, \dots, x_n$ s'écrira sous la forme vectorielle :

$$x = x(J) = (x_j, j \in J).$$

De manière analogue on aura :

$$c = c(J) = (c_j, j \in J), b = b(I) = (b_i, i \in I).$$

L'ensemble des coefficients $a_{ij}, i \in I, j \in J$, sera représenté sous forme d'une matrice A d'ordre $(m \times n)$:

$$A = A(I, J) = (a_{ij}, i \in I, j \in J) = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1j} & \dots & a_{1n} \\ \vdots & & & \ddots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mj} & \dots & a_{mn} \end{pmatrix}$$

On écrit souvent A de la manière suivante :

$$A = (a_1, a_2, \dots, a_j, \dots, a_n), \text{ Où } a_j \text{ est un vecteur colonne :}$$

$$a_j = A(I, j) = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{ij} \\ \vdots \\ a_{mj} \end{pmatrix}$$

Avec ces nouvelles notations, le problème (1)-(3) peut être écrit sous forme matricielle suivante :

$$\begin{cases} Z = Z(x) = c'x \rightarrow \max \\ Ax = b \\ x \geq 0 \end{cases}$$

$$x \in \mathbb{R}^n$$

Ici c' est le transposé de c , la matrice A est la matrice de condition du problème et b vecteur (second membre).

1.2. Forme d'un Programme linéaire

Un programme linéaire peut être écrit sous une :

Forme générale

$$(P.L.) = \begin{cases} (\min/\max) Z(x) = \sum_{j=1}^n c_j x_j \\ \text{s. c} \\ \sum_{j=1}^n a_{ij} x_j \leq, \geq \text{ ou } = b_i, i = \overline{1, m} \\ x_j \geq 0, \forall j = \overline{1, n} \end{cases}$$

Forme canonique

$$(P.L.) = \begin{cases} (\min/\max) Z(x) = c'x \\ \text{s. c} \\ Ax \leq (\text{resp } \geq) b \\ x \geq 0 \end{cases}$$

Forme standard

$$(P.L.) = \begin{cases} (\min/\max) Z(x) = C'x \\ \text{s. c} \\ Ax = b \\ x \geq 0 \\ b \geq 0 \end{cases}$$

Remarque

Les problèmes de minimisation et de maximisation sont en fait équivalents puisque $\min Z = -\max(-Z)$ et $\max Z = -\min(-Z)$.

1.3 Méthode du simplexe

La méthode du simplexe est une méthode itérative. Elle démarre d'un point extrême (sommet de départ) et passe au sommet voisin, et ceci constitue une itération de l'algorithme du simplexe. Pour cela, on doit définir le point extrême de départ et le test d'arrêt.

Soit le problème standard de programmation linéaire (avec maximisation) suivant :

$$Z(x) = c_1x_1 + c_2x_2 + \dots + c_jx_j + \dots + c_nx_n \rightarrow \max \quad (1)$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1j}x_j + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2j}x_j + \dots + a_{2n}x_n = b_2 \\ \dots \dots \dots \dots \dots \dots \\ a_{i1}x_1 + a_{i2}x_2 + \dots + a_{ij}x_j + \dots + a_{in}x_n = b_i \\ \dots \dots \dots \dots \dots \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mj}x_j + \dots + a_{mn}x_n = b_m \end{cases} \quad (2)$$

$$x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0 \quad (3)$$

Ici on suppose $b_1, b_2, \dots, b_m \geq 0$, et $\text{rang } A = m \leq n$.

Le problème (1)-(3) peut être écrit sous sa forme matricielle :

$$\begin{cases} c'x \rightarrow \max \\ Ax = b \\ x \geq 0 \end{cases} \quad (4)$$

Où $A = A(I, J)$ est la matrice des conditions $a_j = A(I, j)$ les colonnes de A .

$c' = c'(J)$ Le vecteur des coûts, $b = b(I)$ le vecteur des contraintes (second Membre), $x = x(J) = (x_j, j \in J)$ le vecteur des variables.

$I = \{1, 2, \dots, m\}, J = \{1, 2, \dots, n\}$ les ensembles d'indices des lignes et des colonnes de sa matrice A .

Problème standard et solution de base :

Définition 1

Tout vecteur x vérifiant les contraintes (2) et (3) est appelé solution réalisable (admissible) du problème (1)-(3).

Définition 2

Une solution réalisable de base x^0 de (1)-(3) est optimale si $c'x^0 = \max\{c'x\}$, pour toute solution réalisable.

Définition 3

Une solution réalisable x est dite de base si $(n - m)$ de ses composantes sont nulles, et aux autres $(x_{j_1}, x_{j_2}, \dots, x_{j_m})$, correspondent m vecteurs $a_{j_1}, a_{j_2}, \dots, a_{j_m}$ de la matrice de condition A linéairement indépendants.

L'ensemble $J_B = \{j_1, j_2, \dots, j_m\}$ est appelé ensemble des indices de base,

$J_H = J \setminus J_B$ Ensemble des indices hors base.

Autrement :

Une solution réalisable $x = x(J)$ est solution de base si $x_H = x(J_H) = 0$,

$\det A_B \neq 0$, où $A_B = A(I, J_B)$, $x(J_B) \geq 0$.

La matrice A_B est appelée la matrice de base, $x_j, j \in J_B$ les composantes de base,

$x_j, j \in J_H$ les composantes hors base.

Définition 4

Une solution réalisable de base x est dite non dégénérée si $x_j > 0, j \in J_B$.

L'accroissement de la fonction objective Z est égale à :

$$\Delta Z = Z(\bar{x}) - Z(x) = c' \bar{x} - c' x = c' \Delta x.$$

Construisons le m -vecteur $y = y(I)$ vecteur de potentiels :

$$y' = c'_B A_B^{-1}.$$

et le vecteur $\Delta = \Delta(J) = (\Delta_j, j \in J)$; le vecteur estimations :

$$\begin{cases} \Delta' = y' A - c' \\ \Delta_j = y' a_j - c_j, j \in J. \end{cases}$$

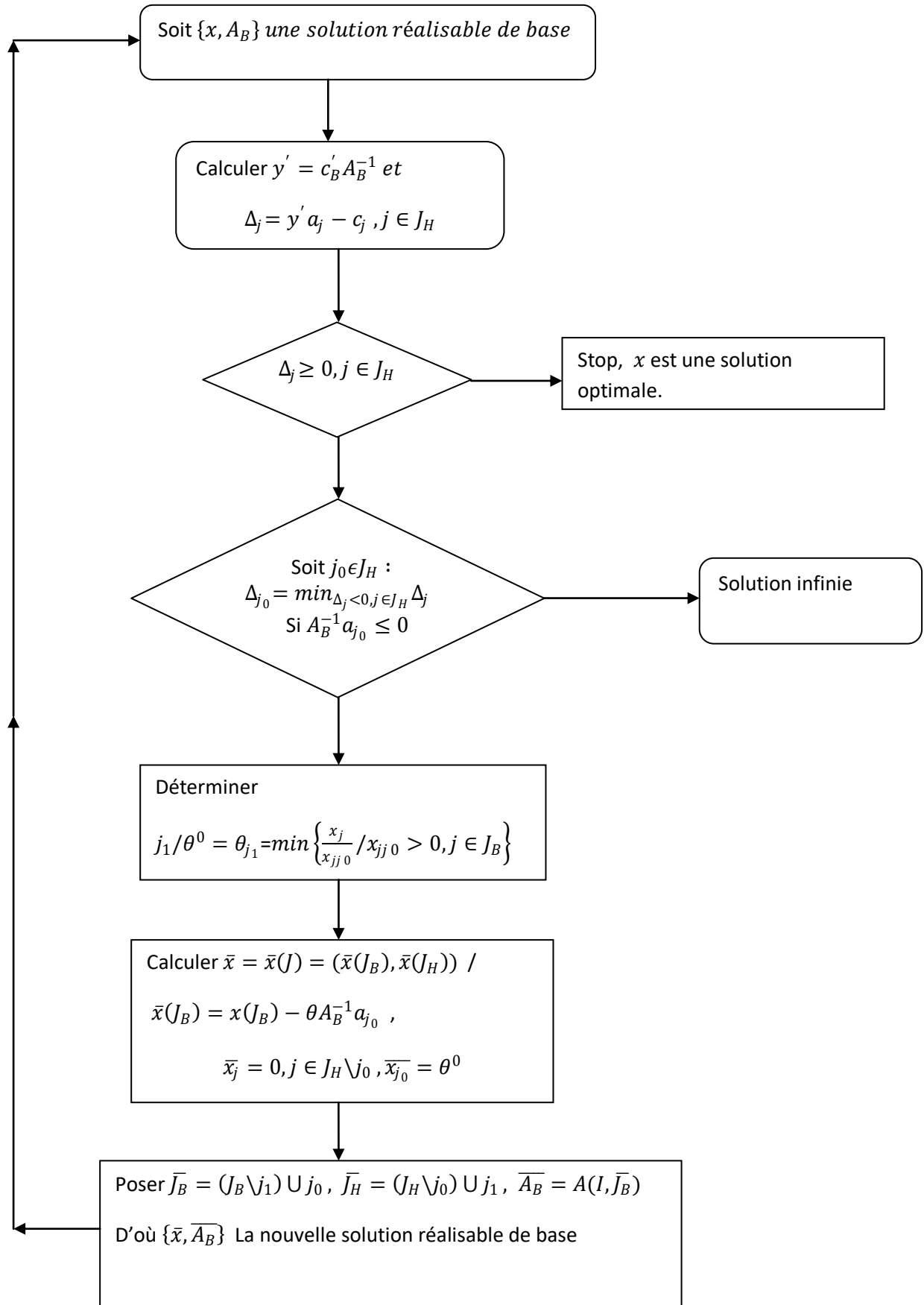
Critère d'optimalité

Théorème 1

Soit $\{x, A_B\}$ une solution réalisable de base de départ.

L'inégalité $\Delta_H = \Delta(J_H) \geq 0$ est suffisante et dans le cas de la non dégénérescence elle est nécessaire pour l'optimalité de $\{x, A_B\}$.

L'organigramme de l'algorithme de simplexe- maximisation [2]



Algorithme du simplexe [2]

1. Soit $\{x, A_B\}$ une solution réalisable de base de départ.

2. Calculer $y' = c'_B A_B^{-1}$ et $\Delta_j = y' a_j - c_j, j \in J_H$.

Si $\Delta_j \geq 0, j \in J_H$

Stop, x est solution optimale.

Sinon :

Soit $j_0 \in J_H : \Delta_{j_0} = (\min \Delta_j / \Delta_j < 0), j \in J_H$.

Si $A_B^{-1} a_{j_0} \leq 0, j \in J_H$

Stop le maximum de la fonction objective tend vers l'infinie

Sinon :

Déterminer : $j_1 / \theta^0 = \theta_{j_1} = \{\min \frac{x_j}{x_{jj_0}} / x_{jj_0} > 0, j \in J_B\}$

Calculer $\bar{x} = \bar{x}(J) = (\bar{x}(J_B), \bar{x}(J_H))$

$\bar{x}(J_B) = x(J_B) - \theta A_B^{-1} a_{j_0}, \bar{x}_j = 0, j \in J_H \setminus j_0, \bar{x}_{j_0} = \theta^0$

Poser $\bar{J}_B = (J_B - j_1) \cup j_0, \bar{J}_H = (J_H - j_0) \cup j_1, \bar{A}_B = A(I, \bar{J}_B)$, d'où $\{\bar{x}, \bar{A}_B\}$, la nouvelle solution réalisable de base puis aller en (2).

Tableau du simplexe [2]

Les différents calculs qu'on aura à effectuer et les différentes étapes de résolution seront disposés dans le tableau suivant :

C			c_1	c_2	\dots	c_m	c_{m+1}	\dots	c_j	\dots	c_n	
c_B	Base	b	a_1	a_2	\dots	a_m	a_{m+1}	\dots	a_j	\dots	a_n	θ_j
c_1	a_1	$b_1 = x_1$	1	0	\dots	0	$x_{1,m+1}$	\dots	x_{1j}	\dots	x_{1n}	θ_1
c_2	a_2	$b_2 = x_2$	0	1	\dots	0	$x_{2,m+1}$	\dots	x_{2j}	\dots	x_{2n}	θ_2
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
c_m	a_m	$b_m = x_m$	0	0	\dots	1	$x_{m,m+1}$	\dots	x_{mj}	\dots	x_{mn}	θ_m
Z		Δ	Δ_1	Δ_2	\dots	Δ_m	Δ_{m+1}	\dots	Δ_j	\dots	Δ_n	

Exemple 1.1

Nous allons résoudre le problème de programmation linéaire suivant, par la méthode du simplexe :

$$\begin{cases} Z = 3x_1 + 2x_2 \rightarrow \max \\ 2x_1 + x_2 + x_3 = 5 \\ x_1 - x_2 + x_4 = 1 \\ x_1 + x_2 + x_5 = 3 \\ x_j \geq 0, j = \overline{1,5} \end{cases}$$

On a $j = \{1,2,3,4,5\}$ et $J_B = \{3,4,5\}$, $J_H = \{1,2\}$, avec $A_B = I_3$ donc solution réalisable de base est $x = (0,0,5,1,3)$, dressons alors le premier tableau du simplexe.

c			3	2	0	0	0	
c_B	Base	B	a_1	a_2	a_3	a_4	a_5	θ_j
0	a_3	5	2	1	1	0	0	5/2
0	a_4	1	1	-1	0	1	0	1
0	a_5	3	1	1	0	0	1	3
$Z=0$		Δ_j	-3	-2	0	0	0	

On remarque que la relation $\Delta_j \geq 0, \forall j \in J_H$ n'est pas vérifiée, donc la solution réalisable de base initiale n'est pas optimale, on doit alors changer la base de la manière suivante :

$$\min_{j \in J_H} \Delta_j = \Delta_1 = -3, \text{ donc } j_0 = 1$$

de là le vecteur a_1 va rentrer dans la nouvelle base .

Et calculons $\theta^0 = \min_{j \in J_B} \theta_j$:

$$\theta_3 = \frac{5}{2}, \theta_4 = \frac{1}{1} = 1, \theta_5 = \frac{3}{1} = 3, \text{ d'ou } \theta^0 = \theta_{j_1} = \min_{j \in J_B} \theta_j = \theta_4 = 1$$

De là, le vecteur a_4 va sortir de la base, et la nouvelle solution \bar{x} , dressons le 2^{ème} tableau du simplexe :

c			3	2	0	0	0	
c_B	Base	B	a_1	a_2	a_3	a_4	a_5	θ_j
0	a_3	3	0	3	1	-2	0	1
3	a_1	1	1	-1	0	1	0	/
0	a_5	2	0	2	0	-1	1	1
$\bar{Z} = 3$		$\bar{\Delta}_j$	0	-4	0	3	0	

La nouvelle solution de base est donc $\bar{x} = (1,0,3,0,2)$ de plus elle n'est pas optimale car $\bar{\Delta}_2 = -4 < 0$.

On doit alors changer la base une autre fois :

$\min_{j \in J_H} \bar{\Delta}_j = \bar{\Delta}_2 = -4$, donc le vecteur a_2 va rentrer dans la nouvelle base.

Comme $\theta_{j_1} = \min_{j \in J_B} \theta_j = \theta_3 = \theta_5$, donc le vecteur a_3 ou a_5 sortira de la base.

D'où, on obtient : $\bar{J}_B = \{3, 1, 2\}$, $\bar{J}_H = \{4, 5\}$, pour déterminer la nouvelle solution \bar{x} dressons le 3^{ème} tableau du simplexe :

c			3	2	0	0	0	
c_B	Base	B	a_1	a_2	a_3	a_4	a_5	θ_j
0	a_3	0	0	0	1	-1/2	-3/2	/
3	a_1	2	1	0	0	1/2	1/2	/
2	a_2	1	0	1	0	-1/2	1/2	/
$\bar{Z} = 8$		$\bar{\Delta}_j$	0	0	0	1/2	5/2	

La nouvelle solution de base est donc $\bar{x} = \{2, 1, 0, 0, 0\}$, comme $\bar{\Delta}_j \geq 0, \forall j \in \bar{J}_H$

L'algorithme s'arrête et la solution obtenue est optimale, avec $\bar{Z} = 8$.

1.4 La M-méthode

Le mathématicien américain Tcharness a proposé cette méthode pour résoudre les programmes linéaire.

Soit le problème :

$$\begin{cases} Z = c'x \rightarrow \max \\ Ax = b \\ x_j \geq 0, j = \overline{1, n} \end{cases} \quad (P)$$

On constitue le problème (PM) de la manière suivante :

$$\begin{cases} \bar{Z} = c'x - M \sum_{i=1}^m x_{n+i} \rightarrow \max \\ [Ax]_i + x_{n+i} = b_i, i = \overline{1, m} \\ x_j \geq 0, j = \overline{1, n+m} \end{cases} \quad (PM)$$

Où $M \gg 0$ (un nombre positif très grand).

Le vecteur $X = (0, b)' = (x = 0, x_{n+i} = b_i, i = \overline{1, m})'$ est une solution de base réalisable pour (PM).

On résout le problème (PM) par la méthode du simplexe avec une solution réalisable de base de départ $\{(0, b), A_B\}$ et on obtient une solution optimale

$\{X^0 = (x^0, x_{n+i}^0, i = \overline{1, m}), A_B^0\}$.

Principe de la M-méthode

Après la transformation de (P) en (PM), à l'optimum nous avons les cas suivants :

1. si toutes les variables artificielles sont nulles alors la solution est optimale.
De plus si $\Delta_{j_0} = 0/j_0 \in J_H$ alors on aura une infinité de solutions optimales.
2. Au moins une variable artificielle de la base strictement positive \Rightarrow contraintes contradictoires.
3. A un certain moment on ne peut pas améliorer, par manque de pivot .
On distingue deux cas :
- Toutes les variables artificielles sont nulles \Rightarrow solution infinie
- Au moins une variable artificielle strictement positive \Rightarrow contraintes contradictoires.

Exemple 1.2 : résoudre le problème suivant :

$$(P) \begin{cases} \max Z = -2x_1 + x_2 + 4x_3 \\ x_1 - x_2 + x_3 = 2 \\ -5x_2 + 2x_3 = 4 \\ -2x_2 + x_3 = 1 \\ x_j \geq 0, j = \overline{1,3} \end{cases}$$

Pour déterminer une solution de base initiale, on doit ajouter deux variables artificielles x_4 et x_5 , d'où on obtient le M-problème suivant :

$$(PM) \begin{cases} \max Z = -2x_1 + x_2 + 4x_3 - Mx_4 - Mx_5 \\ x_1 - x_2 + x_3 = 2 \\ -5x_2 + 2x_3 + x_4 = 4 \\ -2x_2 + x_3 + x_5 = 1 \\ x_j \geq 0, j = \overline{1,5} \end{cases}$$

Il admet une solution de base de départ $\bar{x} = (2, 0, 0, 4, 1)$, avec $\bar{J}_B = \{1, 4, 5\}$ et $\bar{J}_H = \{2, 3\}$.
Calculons le potentiel \bar{y} et le vecteur des estimations $\bar{\Delta}_j, j \in J$:

$$\bar{y} = \bar{c}_B \bar{A}_B^{-1} = (-2, -M, -M) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = (-2, -M, -M).$$

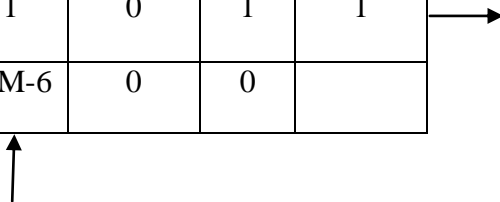
$$\bar{\Delta}_j = 0, j \in \bar{J}_B$$

$$\bar{\Delta}_2 = \bar{y} a_2 - c_2 = (-2, -M, -M) \begin{pmatrix} -1 \\ -5 \\ -2 \end{pmatrix} - 1 = 7M + 1$$

$$\bar{\Delta}_3 = \bar{y} a_3 - c_3 = (-2, -M, -M) \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} - 4 = -3M - 6$$

Dressons le premier tableau du simplexe :

C			-2	1	4	-M	-M	
\overline{C}_B	Base	B	a_1	a_2	a_3	a_4	a_5	$\overline{\theta}_j$
-2	a_1	2	1	-1	1	0	0	2
-M	a_4	4	0	-5	2	1	0	2
-M	a_5	1	0	-2	1	0	1	1
$\overline{Z} = -5M - 4$		$\overline{\Delta}_j$	0	$7M+1$	$-3M-6$	0	0	



La solution de départ n'est pas optimale car $\overline{\Delta}_3 < 0$, donc on doit faire rentrer a_3 dans la base et faire sortir a_5 , ainsi on obtient le nouveau tableau suivant :

C			-2	1	4	-M	-M	
\overline{C}_B	Base	B	a_1	a_2	a_3	a_4	a_5	$\overline{\theta}_j$
-2	a_1	1	1	1	0	0	-1	
-M	a_4	2	0	-1	0	1	-2	
4	a_3	1	0	-2	1	0	1	
$\overline{Z} = -2M + 2$		$\overline{\Delta}_j$	0	$M-11$	0	0	$3M+6$	

Le critère d'optimalité est vérifié, donc la solution du M-problème est $x = (1,0,1,2,0)$
 Comme la variable artificielle $\overline{x}_4 = 2 > 0$, donc le problème (P) n'admet pas de solution optimale (les contraintes de (P) sont contradictoires).

1.5 Méthode duale du simplexe

Etant donné le problème primal de programmation linéaire:

$$\begin{cases} c'x \rightarrow \max \\ Ax = b \\ x \geq 0 \end{cases} \quad (\mathbf{P})$$

Et son dual:

$$\begin{cases} b'y \rightarrow \min \\ A'y \geq c \\ y \in \mathbb{R}^m \end{cases} \quad (\mathbf{D})$$

Définition1 [2]

De l'ensemble J , choisissons un sous ensemble $J_B \in J$ et soit $A_B = (I, J_B)$ une sous matrice inversible de A .

En utilisant la matrice A_B , on construit le vecteur y :

$$y' = c'_B A_B^{-1} \quad (1)$$

Le vecteur y est dit plan dual basique et A_B la matrice de base si

$$A'_H y \geq c_H \quad (2)$$

Où $A_H = (I, J_H)$, $J_H = J \setminus J_B$.

Définition2 [2]

Un plan dual basique y est dit non dégénéré si $A'_H y > c_H$.

En utilisant un plan dual basique de départ y , on construit les vecteurs suivants:

$$\delta(J) = A'_J y - c, x(J) = (x(J_B), x(J_H)), x(J_B) = x_B = A_B^{-1} b$$

$$, x(J_H) = x_H = 0$$

Appelés Coplan et pseudo plan respectivement du problème(P).

Remarque1

Par construction $\delta(J_B) = 0$ et $\delta(J_H) \geq 0$.

Si y est un plan dual basique alors δ et x sont dits basiques.

Algorithme dual du simplexe [2]

Considérant un plan dual basique y avec sa matrice de base A_B .

En utilisant A_B , on calcule le pseudo plan $x = (x_B = A_B^{-1} b, x_H = 0)$.

Si $x_B \geq 0$ Alors x est optimale pour le problème (P), et y optimal du dual(D)

sinon, on calcule $x_{j_0} = \min x_j, (x_j < 0, j \in J_B)$ de la l'indice j_0 doit sortir de la base et la colonne a_{j_0} doit sortir de A_B , c'est à dire, on change de base ($A_B \rightarrow \overline{A_B}$).

Le changement de base entraine le changement du plan dual y ($y \rightarrow \bar{y}$) qui entraine aussi le changement du coplan δ ($\delta \rightarrow \bar{\delta}$).

Ce changement de coplan se fera de la manière suivante: $\bar{\delta} = \delta + \Delta \delta$, où :

$$\Delta \delta_j = \begin{cases} \sigma, j = j_0 \\ 0, j \in J_B \setminus j_0 \end{cases}$$

Où σ est le pas dual positif ou nul.

$\Delta \delta_j = \sigma x_{j0j}$, $j \in J_H$ où x_{j0j} est la $j^{\text{ème}}$ composante du vecteur $A_B^{-1} a_j$.

Pour que $\bar{\delta}$ soit un coplan, il faut avoir un pas maximal σ° :

$$\sigma^\circ = \min_{x_{j0j} < 0, j \in J_H} \left\{ \frac{-\delta_j}{x_{j0j}} \right\} = \frac{-\delta_{j_1}}{x_{j_1 0j_1}}$$

La nouvelle base sera $\bar{J}_B = (J_B \setminus j_0) \cup j_1$, et $\bar{A}_B = A(I, \bar{J}_B)$, la nouvelle itération débutera avec $\bar{x} = (\bar{x}_B) = (\bar{A}_B^{-1} b, \bar{x}_H = 0)$.

Remarque2

Les problèmes du type:

$$\begin{cases} Z = c'x \rightarrow \min \\ Ax \geq b \\ x \geq 0 \\ b \geq 0, c \geq 0 \end{cases}$$

Sont résolus dans la plupart des cas par la méthode duale du simplexe, car en ajoutant des variables d'écart, on obtient facilement la solution de base de départ. Par contre si on utilise la méthode du simplexe, on ajoute des variables d'écart et des variables artificielles et ceci, augmente la dimension du problème.

Exemple.1.3

Résoudre le problème linéaire suivant :

$$\begin{cases} \min Z = 2x_1 + x_3 \\ x_1 + x_2 - x_3 \geq 5 \\ x_1 - 2x_2 + 4x_3 \geq 8 \\ x_1 \geq 0, x_2 \geq 0 \text{ et } x_3 \geq 0 \end{cases}$$

Sous forme standard, on aura :

$$\begin{cases} \max Z = -2x_1 - x_3 \\ -x_1 - x_2 + x_3 + x_4 = -5 \\ -x_1 + 2x_2 - 4x_3 + x_5 = -8 \\ x_i \geq 0, i = \overline{1,5} \end{cases}$$

D'où $x = (0, 0, 0, -5, -8)$ et $Z(x) = 0$.

Soit $A_B = (a_4, a_5) = I_2$, une telle matrice inversible avec $J_B = \{4, 5\}$, $J_H = \{1, 2, 3\}$.

En utilisant A_B , on calcule le vecteur des potentiels y :

$$y' c'_B A_B^{-1} = (0, 0), \text{ car } c'_B = (0, 0),$$

de là $y' A_H = (0, 0, 0) \geq c'_H = (-2, 0, -1)$, donc y est un plan dual, ce qui veut dire que la matrice A_B est de base.

Le coplan basique initial est donc égale à $\delta = A'y - c = -c = (2,0,1,0,0)$.

c		-2	0	-1	0	0
$base$	b	a_1	a_2	a_3	a_4	a_5
x_4	-5	-1	-1	1	1	0
x_5	-8	-1	2	-4	0	1
	δ	2	0	1	0	0
	σ	2	/	1/4	/	/

Les composantes du pseudo plan x_B ne sont pas toutes positives ,donc le critère d'optimalité n'est pas vérifié .

On a $x_5 = -8 = \min_{x_j < 0, j \in J_B} x_j$, donc a_5 sort de la base.

De plus $\sigma_1 = \left(\frac{-\delta_1}{x_{j11}} \right) = \frac{-2}{-1} = 2$

$$\sigma_3 = \frac{-\delta_3}{x_{j13}} = \frac{-1}{-4} = 1/4$$

$$\min_{x_{j1j} < 0, j \in J_H} (\sigma_j) = 1/4$$

Donc le vecteur a_3 va rentrer dans la nouvelle base à la place de a_5 et on passe à la nouvelle itération :

c		-2	0	-1	0	0
$base$	b	a_1	a_2	a_3	a_4	a_5
x_4	-7	-5/4	-1/2	0	1	1/4
x_3	2	1/4	-1/2	1	0	-1/4
	δ	7/4	1/2	0	0	1/4
	σ	7/5	1	/	/	/

On a $x_4 = -7 < 0$, donc a_4 sort de la base et sera remplacé par a_2 , qui correspond au pas minimum $\sigma_2 = 1$, par suite on passe à l'autre itération :

c		-2	0	-1	0	0
$base$	b	a_1	a_2	a_3	a_4	a_5
x_2	14	5/2	1	0	-2	-1/2
x_3	9	3/2	0	1	-1	-2/4
	δ			0		
	σ					

On a les $x_j \geq 0$, donc la solution est optimale.

D'où $\bar{x}^* = (0,14,9,0,0)$ et $z(\bar{x}^*) = 9$.

1.6 La méthode Branch and Bound [6]

- Les méthodes de branch-and-bound sont des méthodes basées sur une énumération "intelligente" des solutions admissibles d'un problème d'optimisation combinatoire.
- Idée : prouver l'optimalité d'une solution en partitionnant l'espace des solutions.
- Exemple d'application à la programmation linéaire et non linéaire en nombres entiers
Utilise toute la puissance de la programmation linéaire ou non linéaire pour déterminer des bonnes bornes.
- On appelle relaxation linéaire d'un programme linéaire en nombres entiers le programme linéaire obtenu en supprimant les contraintes d'intégrité sur les variables.

Programme en nombres entiers

$$(P) \begin{cases} \max c'x \\ s.c \ Ax \leq b \\ x \geq 0, \text{entier} \end{cases}$$

Relaxation linéaire

$$(LP) \begin{cases} \max c'x \\ s.c \ Ax \leq b \\ x \geq 0. \end{cases}$$

Propriétés de la relaxation linéaire

- La valeur de la solution optimale de LP est une borne supérieure sur la valeur de la solution optimale de P.
- La valeur d'une solution admissible de P fournit une borne inférieure sur la valeur de la solution optimale de P.
- Si la solution optimale de LP est entière (donc admissible pour P), elle est également la solution optimale de P.

Branchement

- Si la solution de (LP) n'est pas entière, soit x_i une variable prenant une valeur fractionnaire x_i^* dans la solution optimale de (LP).
- Le problème peut être divisé en deux sous-problèmes en imposant
 $x_i \leq \lfloor x_i^* \rfloor$ Ou $x_i \geq \lfloor x_i^* \rfloor + 1$

Où $\lfloor x_i^* \rfloor$ est le plus grand entier inférieur à x_i^* .

- La solution optimale de (P) est la meilleure des solutions optimales des deux sous problèmes (P1; P2) :

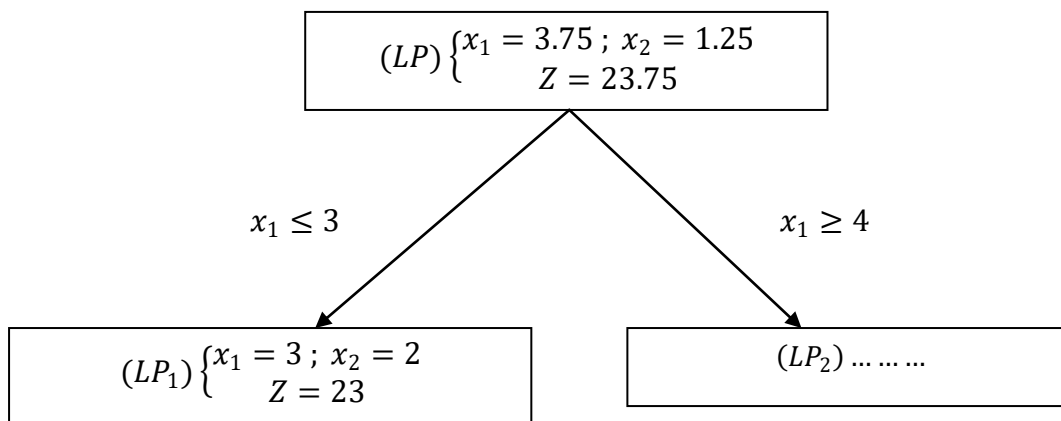
$$(P1) \begin{cases} \max c'x \\ s. c \ Ax \leq b \\ x_i \leq \lfloor x_i^* \rfloor \\ x \geq 0, \text{entier} \end{cases}$$

$$(P2) \begin{cases} \max c'x \\ s. c \ Ax \leq b \\ x_i \geq \lfloor x_i^* \rfloor + 1 \\ x \geq 0, \text{entier} \end{cases}$$

Exemple 1.4 (Branch-and-Bound)

$$(P) \begin{cases} \max Z = 5x_1 + 4x_2 \\ x_1 + x_2 \leq 5 \\ 10x_1 + 6x_2 \leq 45 \\ x_1, x_2 \geq 0, \text{entiers} \end{cases}$$

$$(LP) \begin{cases} x_1 = 3.75 ; x_2 = 1.25 \\ Z = 23.75 \end{cases}$$



La solution de (LP_1) est une solution admissible de (P) et donc $Z = 23$ est une borne inférieure sur la valeur de la solution optimale de (P) .

Le noeud correspondant peut être éliminé vu qu'une solution entière optimale satisfaisant $x_1 \leq 3$ a été trouvée (solution de LP_1).

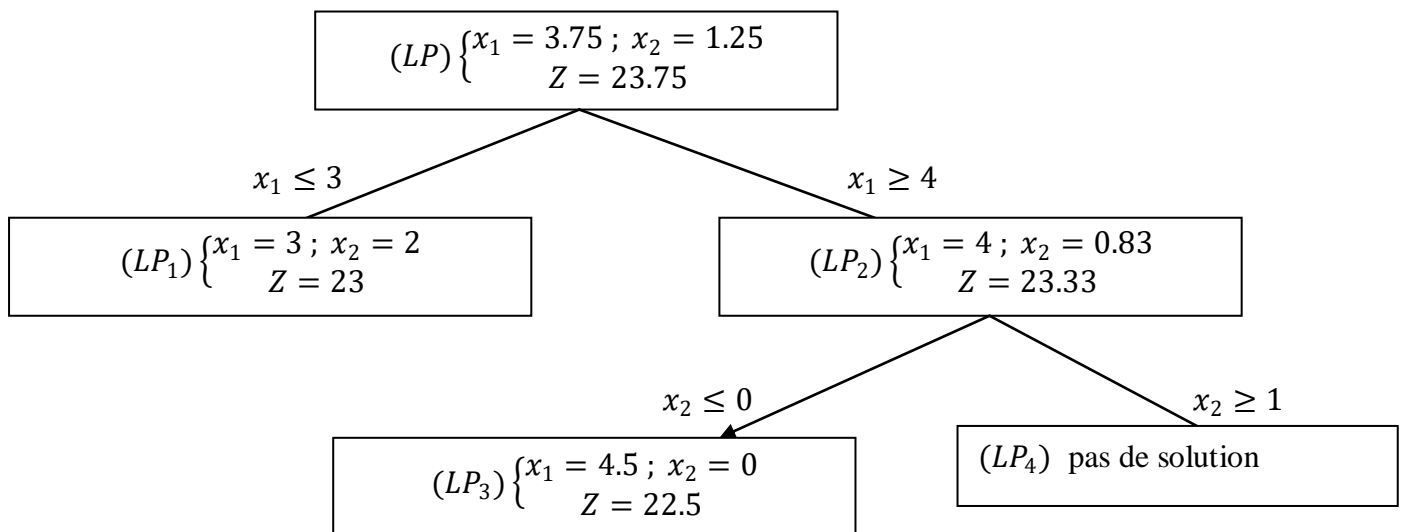
La valeur de la solution de (LP) , $Z = 23.75$ est une borne supérieure sur la valeur de la solution optimale de (P) .

Vu que tous les coefficients sont entiers, on peut en déduire que la valeur de la solution optimale de (P) est inférieure ou égale à 23.

La solution de (LP_1) est donc optimale pour (P) .

Règles de branchement

- Il n'y a pas de règle générale pour le choix de la variable de branchement et de la branche à examiner en premier.
- Ce choix peut avoir un impact important sur le nombre de noeuds à examiner dans l'arbre de branch-and-bound.



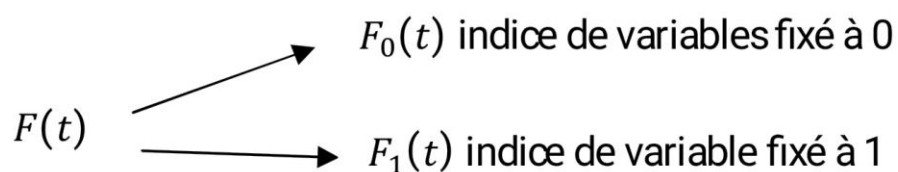
1.7 La Méthode de Balas [4]

La méthode de Balas est une méthode arborescente, comme toute autre technique arborescente (Branch and Bound) cette méthode a des limites au delà d'une dimension. Cette méthode peut être classée dans la classe des problèmes NP-difficile.

Noeuds de séparation (branching)

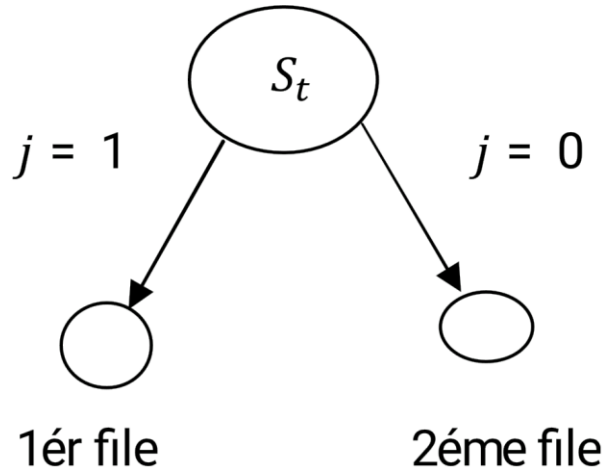
Chaque noeud de (sommet) S_t correspond à une solution partielle.
Ce noeud correspond à deux sous ensembles d'indices pour les variables.

Soient : $F(t)$ Les sous ensembles d'indices fixés à 0 ou à 1 pour les variables au niveau du noeud S_t



$L(t)$ Les sous ensembles d'indices de variables non fixés (libres).

Pour faire le branching on choisira donc une variable telle $j \in L(t)$, à ce moment on aura deux files



Remarque 1

Le problème posé c'est comment choisir sur le quel on fera le branching (séparation).

Pour cela écrivons le problème sur n'importe quel nœud (sommet) $S(t)$

$$Z_t = \sum_{j \in L(t)} c_j x_j + \sum_{j \in F_1(t)} c_j \rightarrow \min$$

$$\sum_{j \in L(t)} a_{ij} x_j \leq b_i - \sum_{j \in F_1(t)} a_{ij} = S_i$$

$$i = \{1, 2, \dots, m\}$$

Nous appelons l'évaluation $(S_t) = \sum_{j \in F_1(t)} c_j$ valeurs meilleur pour ce nœud.
 S_t est un noeud quelconque de l'arborescence.

Les tests sur les nœuds (les sommets S_t)

Après création d'un nœud S_t et son évaluation il faut faire les testes suivants :

1. On abandonne un noeud si l'évaluation (S_t) est supérieur ou égale à $Z_{trouvée}$ valeur de la solution provisoire trouvée.
2. x solution obtenu en posant $x_j = 0$ et $j \in L(t)$ est solution réalisable pour $pl_{(0,1)}$ comme avec $S_i \geq 0 \forall i \in \{1, \dots, m\}$ alors le noeud est terminal en (S_t) son évaluation $MinZ_t$.

Décision de séparation au niveau d'un sommet (noeud)

Comment séparer au niveau d'un sommet ? (quelle variable de séparation à choisir)

Balas à donné une technique pour minimiser le nombre de branching à effectuer et cela comme suivant :

Soit deux sous ensemble $Q(t)$ et $R(t)$ définis comme suit :

$$Q(t) = \{i / S_i < 0\}, i = \overline{1, m}.$$

$$R(t) = \{j \in L(t) / \exists i \in Q(t), a_{ij} < 0\}.$$

Mesure de proximité

Balas propose pour ses solutions (pour $P(t) < 0$)

$$P(t) = \sum_{i \in Q(t)} S_i = \sum_{i \in \{1, \dots, m\}} \min(0, S_i)$$

La proximité d'une solution si on fixe x_j à 1 est la suivante :

$$P(t, j) = \sum_{i=1..m} \min(0, S_i - a_{ij})$$

Le terme $(S_i - a_{ij})$ est le second membre.

Si on fixe x_j à 1 :

Pour déterminer la variable x_{j^*} à choisir pour faire le branchig (Séparation) la méthode propose de choisir l'indice $j^* \in R(t)$.

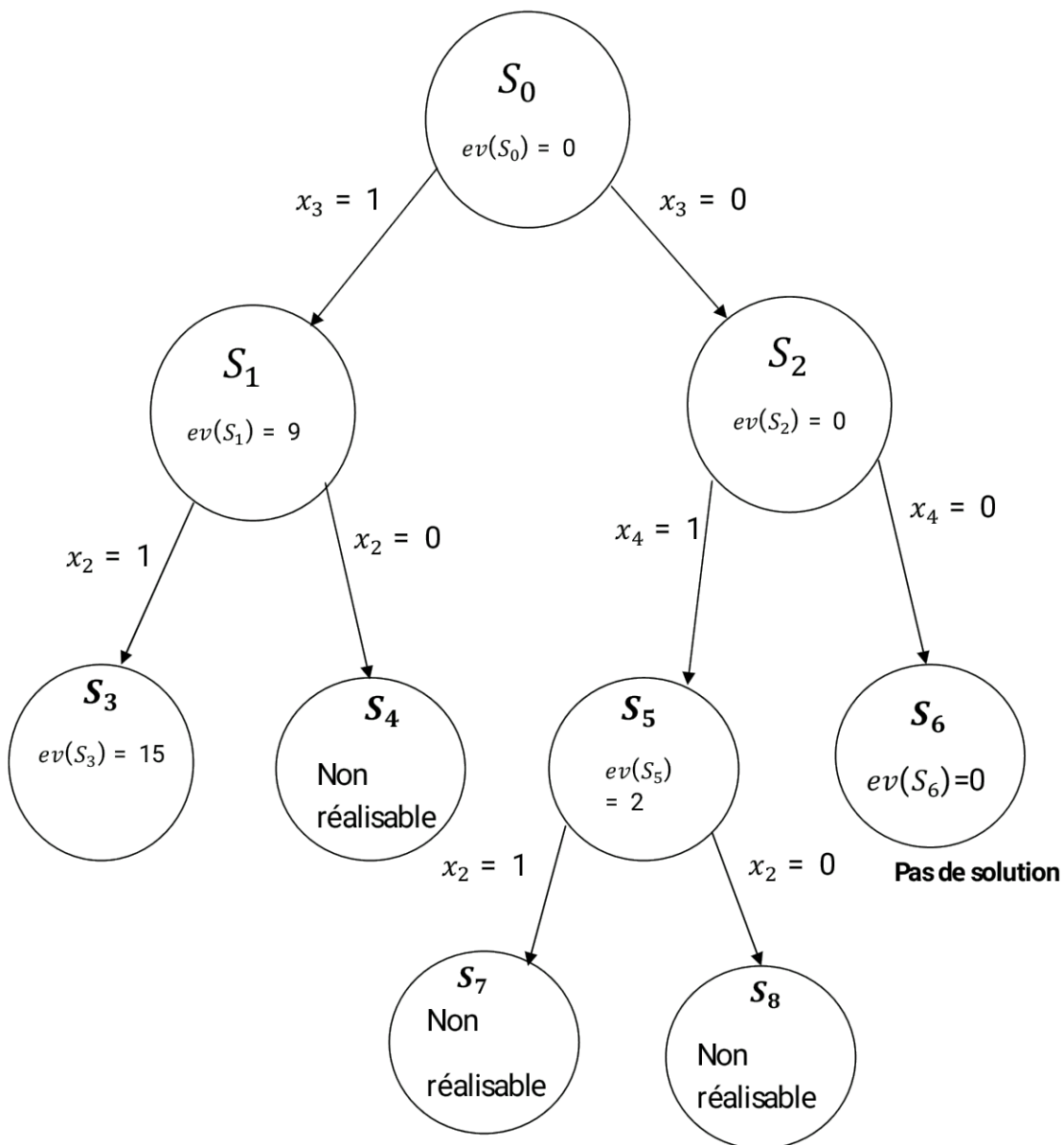
$$P(t, j^*) = \max(P(t, j)) , j \in R(t).$$

Conclusion

- $P(t, j^*) = 0$ la solution obtenue pour $x_{j^*} = 1$ et $\forall j \in L(t)$ est une nouvelle solution réalisable (neud terminal).
- En cas d'ex aequo on choisi la variable de coût faible.

Exemple.1.4

$$\begin{cases} Z = 4x_1 + 6x_2 + 9x_3 + 2x_4 + x_5 \rightarrow \min \\ -x_1 + 2x_2 - 4x_3 - x_4 + 3x_5 \leq -2 \\ x_1 - 2x_2 + x_4 + x_5 \leq -1 \\ 2x_1 - 5x_2 + 2x_3 + x_4 - 2x_5 \leq 1 \\ x_j \in \{0,1\} \quad j = \overline{1,5} \end{cases}$$



Conclusion

Avec la méthode de Balas (Algorithme) on a trouvé :

$$x_1^* = 0 \qquad x_2^* = 1$$

$$x_3^* = 1 \qquad x_4^* = 0$$

$$x_5^* = 0$$

Est solution optimale et $Z(x^*) = 15$.

1.8 Méthodes des coupes

a) Coupe de Dantzig :

La coupe de Dantzig a été proposée sur la base que dans le problème relaxé (P_r) ; 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 :

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

Où J_H est l'ensemble des indices hors base.

b) Méthode des troncatures (algorithme de Gomory) :

Description :

La méthode des troncatures [5] (appelée également "algorithme du plan sécant" ou "méthode des coupes") développé par Ralph E. Gomory (1958) fournit une méthode de résolution des programmes linéaires en nombres entiers (PLNE). L'idée principale des méthodes des coupes est d'ajouter des contraintes qui n'excluent aucun point entier réalisable. La méthode consistera à ajouter de telles contraintes linéaires une par une, jusqu'à ce que la solution optimale du problème relaxé soit entière. Les contraintes ajoutées sont appelées troncatures ou coupes.

Remarque : Notions de partie entière et partie fractionnaire d'un nombre

Pour un nombre réel « a » la somme de sa partie entière $[a]$ et de sa partie fractionnaire f s'écrit :

$$a = [a] + f$$

Où $N = [a]$ est le plus grand nombre entier inférieur ou égal au nombre a ($N \leq a$)

$0 < f < 1$: La partie fractionnaire d'un nombre réel " a " est donc la différence entre ce nombre et sa partie entière.

Algorithme de Gomory

1. Etant donné un (PLNE) ; résoudre le programme (P_r) correspondant à l'aide de l'algorithme du simplexe. Si la solution optimale de (P_r) contient uniquement des valeurs entières, elle est également une solution optimale du (PLNE) et la résolution est terminée.
2. Si une ou plusieurs variables de base dans la solution optimale de (P_r) ne sont pas entières, on doit alors générer, à partir d'une des lignes du tableau (celle dont la partie fractionnaire pour la variable de base correspondante est la plus élevée) une contrainte supplémentaire dite

coupe de Gomory. Cette contrainte (sous forme d'équation) est alors ajoutée au tableau optimal de (P_r) et on détermine le nouveau tableau optimal à l'aide de la méthode duale du simplexe.

3. Si les variables de base dans le nouveau tableau optimal sont entières, nous avons obtenu également la solution optimale du $(PLNE)$. La résolution est terminée.

4. Sinon, on doit générer (à partir du dernier tableau optimal) une nouvelle coupe de Gomory l'ajouter au dernier tableau optimal et trouver la nouvelle solution optimale à l'aide de la méthode duale du simplexe.

- Si la solution optimale obtenue est à valeurs entières, la résolution est terminée.
- Sinon, on répète la procédure jusqu'à l'obtention d'une solution optimale à valeurs entières.

Règle pratique pour générer la coupe de Gomory

1. On choisira d'abord, dans le tableau optimal, la variable de base qui présente la plus grande partie fractionnaire et on en déduit l'équation (la contrainte) correspondante exprimant toutes les variables du tableau en fonction de la valeur x_{B_j} de la variable de base.

2. Pour générer la nouvelle contrainte (coupe de Gomory), on remplace tout simplement tous les coefficients dans la contrainte obtenue du tableau optimal par sa partie fractionnaire correspondante et déclarons par la suite que l'expression résultante doit être plus grande ou égale à la partie fractionnaire du second membre c'est à dire la partie fractionnaire de la valeur x_{B_j} choisie en (1).

On appliquera cette règle chaque fois que l'on doit déterminer une nouvelle coupe de Gomory dans la résolution du $(PLNE)$.

Exemple d'application

Soit le problème linéaire en nombre entier suivant :

$$(PLNE) \begin{cases} \max z(x) = x_1 + 2x_2 \\ 2x_1 + x_2 \leq 3 \\ x_2 \leq 2 \\ x_1 \in \mathbb{N}, x_2 \in \mathbb{N} \end{cases}$$

La forme standard et relaxé de $(PLNE)$:

$$(P_r) \begin{cases} \max z(x) = x_1 + 2x_2 \\ 2x_1 + x_2 + x_3 = 3 \\ x_2 + x_4 = 2 \\ x_j \geq 0, j = 1..4 \end{cases}$$

On a $J = \{1,2,3,4\}$, $J_B = \{3,4\}$ et $J_H = \{1,2\}$.

Donc la solution de base réalisable est $x = (0,0,3,2)$.

Dressons le premier tableau du simplexe du (P_r) :

C_B	Base	B	x_1	x_2	x_3	x_4	θ_j
0	x_3	3	2	1	1	0	3
0	x_4	2	0	1	0	1	2
$Z = 0$	/	Δ	-1	-2	0	0	/

Tableau 1

On remarque que la relation $\Delta_j \geq 0, \forall j \in J_H$ n'est pas vérifiée, donc la solution réalisable de base initiale n'est pas optimale, on doit alors changer la base de la manière suivante :

$$\text{Min}(\Delta_j / \Delta_j < 0, j \in J_H) = \Delta_2 = -2$$

Donc $j_0 = 2$, de là la variable x_2 va rentrer dans la nouvelle base, et calculons :

$\theta^\circ = \min(j \in J_B, \theta_j)$, d'où $\theta^\circ = 2$, de là la variable x_4 va sortir de la base, et la nouvelle base est $J_B = \{2, 3\}$.

Dressons le deuxième tableau du simplexe pour déterminer la nouvelle solution :

C_B	Base	B	x_1	x_2	x_3	x_4	θ_j
0	x_3	1	2	0	1	-1	1/2
2	x_2	2	0	1	0	1	/
$Z = 4$	/	Δ	-1	0	0	2	/

Tableau 2

La nouvelle solution de base est $\bar{x} = (0, 2, 1, 0)$; qui n'est pas optimale, car $\Delta_1 = -1 < 0$.

Dressons le troisième tableau du simplexe :

C_B	Base	B	x_1	x_2	x_3	x_4	θ_j
1	x_1	1/2	1	0	1/2	-1/2	/
2	x_2	2	0	1	0	0	/
$Z = 9/2$	/	Δ	0	0	0	1/2	/

Tableau 3

La solution de la troisième itération est optimale pour le (P_r) mais pas optimale pour $(PLNE)$.

Pour cela on exhibe la coupe de Gomory tel que : $f_i \leq \sum_{j \in J_H} f_{ij} x_j$

$$1/2 \leq 1/2x_3 - 1/2x_4.$$

En utilisons le dual du simplexe on résout le dernier (P_r') :

1^{ère} itération :

C_B	Base	B	x_1	x_2	x_3	x_4	x_5
1	x_1	$\frac{1}{2}$	1	0	$\frac{1}{2}$	$-\frac{1}{2}$	0
2	x_2	2	0	1	0	1	0
0	x_5	$-\frac{1}{2}$	0	0	$-\frac{1}{2}$	$\frac{1}{2}$	1
$Z = 9/2$	/	Δ	0	0	$\frac{1}{2}$	$\frac{3}{2}$	0
/	/	σ	/	/	1	3	/

Le critère d'optimalité n'est pas vérifié car $\exists x_j < 0, j \in J_B$, on doit faire sortir x_5 de la base car $\min x_j = x_5 < 0$. On a $\sigma_j = -\frac{\Delta_j}{x_{5j}}$ et $\min \sigma_j = \sigma_3$, donc on doit faire rentrer x_3 dans la base.

D'où on a la 2^{ème} itération :

C_B	Base	B	x_1	x_2	x_3	x_4	x_5
1	x_1	0	1	0	0	-1	1
2	x_2	2	0	1	0	1	0
0	x_3	1	0	0	1	-1	-2
$Z = 4$	/	Δ	0	0	0	1	1

Le critère d'optimalité est vérifié, donc $\bar{x} = (0, 2, 1, 0, 0)$ est la solution optimale du problème (P_r') avec $Z^\circ = 4$ et aussi pour le $(PLNE)$.

CHAPITRE 2

Programmation multi-objectif linéaire

2.1. Introduction

La vie réelle foisonne de problèmes qui cherchent une solution. La majorité de ces problèmes ont des solutions qui ne sont pas forcément convenables selon un ou plusieurs critères bien définis. La plupart des problèmes d'optimisation réels sont décrits à l'aide de plusieurs objectifs ou critères souvent contradictoires et parfois complémentaires qui doivent être optimisés simultanément. Pour les problèmes n'incluant qu'un seul objectif, l'optimum cherché est clairement défini, celui-ci reste à formaliser pour les problèmes d'optimisation multi-objectif.

Prenons le cas d'une personne souhaitant acheter une maison. La maison idéale est celle qui est moins chère avec beaucoup d'espace et bien située si possible, mais cette maison Idyllique n'existe pas. Notre acheteur va donc devoir identifier les meilleurs compromis possibles correspondants à son budget.

Plusieurs autres problèmes peuvent être décrits de la même manière tels que l'établissement d'un emploi du temps scolaire ; c'est un problème multi-objectif de nature car il faut en même temps optimiser plusieurs objectifs tels que : le volume horaire à enseigner, l'occupation des locaux, la charge horaire par enseignant, le nombre de matière....etc. Traditionnellement, Les problèmes multi-objectifs ont été abordés comme problèmes d'optimisation mono-objectif après la combinaison de plusieurs critères dans une simple valeur scalaire. De l'autre côté et pendant les dernières années, il y a eu l'apparition d'un certain nombre de métaheuristiques multi-objectif dont le but est d'obtenir un ensemble de solutions de compromis pour des problèmes d'optimisation multi-objectif dans une seule exécution et sans besoin de convertir le problème en mono-objectif au risque que celui-ci perd sa signification. La plupart de ces techniques ont réalisé un grand succès dans l'optimisation des problèmes réels multi-objectifs.

Les problèmes multi-objectifs ont la particularité d'être beaucoup plus difficile à traiter que leurs équivalents mono-objectifs. La difficulté réside dans l'absence d'une relation d'ordre total entre les solutions.

2.2 Définitions

Un problème d'optimisation multi-objectif (**M**ultiobjective **O**ptimizaion **P**roblem(**MOP**)), ou multicritère est défini comme la maximisation (ou minimisation) de plusieurs fonctions $f(x) = F(x) = (f_1(x), f_2(x), \dots, f_k(x))$ soumise à $(g_i(x) \geq b_i, i = 1..p)$ et

$$(h_j(x) = c_j, j = 1..q), x \in \mathbb{R}^n. \text{ Avec } D = \left\{ \begin{array}{ll} g_i(x) \geq b_i & i = 1..p \\ h_j(x) = c_j & j = 1..q \end{array} ; x \in \mathbb{R}^n \right\}$$

Une solution du (MOP) minimise (ou maximise) les composantes d'un vecteur $F(x)$ où x est un vecteur de n variables de décision $x = (x_1, x_2, \dots, x_n)'$. À noter que l'ensemble des solutions réalisables D est formé par les contraintes $g_i(x) \geq b_i \forall i = 1..p$ et $h_j(x) = c_j \forall j = 1..q$ qui doivent être satisfaites tout en minimisant ou en maximisant $F(x)$.

On se limite aux problèmes de minimisation, puisque la maximisation d'une fonction $F(x)$ peut facilement être transformée en un problème de minimisation :

$$\min F(x) = -\max(-F(x))$$

Définition1 (Problème d'optimisation multi-objectif)

Généralement, un problème d'optimisation multi-objectif (MOP) est formulé comme suit :

$$(MOP) \begin{cases} F(x) = (f_1(x), \dots, f_k(x)) \rightarrow \max/\min \\ \text{s. c } x \in D \end{cases}$$

Dans notre mémoire, on considère le cas de minimisation :

$$(MOP) \begin{cases} F(x) = (f_1(x), \dots, f_k(x)) \rightarrow \min \\ \text{s. c } x \in D \end{cases}$$

$$\text{Avec : } D = \left\{ \begin{cases} g_i(x) \geq b_i & i = 1..p \\ h_j(x) = c_j & j = 1..q \end{cases} ; x \in \mathbb{R}^n \right\}$$

Définition 2 (espace des critères)

L'espace des critères est l'image de l'espace réalisable notée $F = (f(D))$. Les éléments de F sont appelés (fonctions) vecteurs objectifs ou vecteurs de critère et désignés par $F(x)$ où $F = (f_1, f_2, \dots, f_k)'$ et $F_i = f_i(x)$ pour tout $i = (1..k)$ sont des valeurs du vecteurs de critère. F est noté :

$$F = f(x) \quad x \in D.$$

Un problème d'optimisation multi-objectifs porte sur deux espaces, que sont l'espace de décision D et l'espace fonctionnel (l'espace des fonctions coût) F .

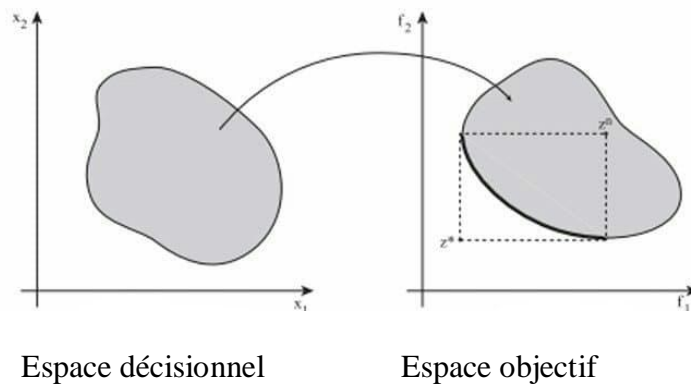


Figure 2.1 : Espace décisionnel et espace objectif d'un problème d'optimisation multi-objectif (exemple avec deux variables de décisions et deux fonctions objectifs).

Définition 3 (convexité)

Un problème d'optimisation multi-objectif est dit convexe si toutes les fonctions objectifs sont convexes et l'ensemble des solutions réalisables est convexe.

L'ensemble D est dit convexe si tout segment joignant deux points quelconques de D est inclus dans D .

$$x \in D \text{ et } y \in D \Leftrightarrow \text{segment } [x, y] \subset D$$

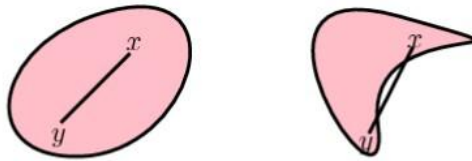


Figure 2.2 ensemble convexe (à gauche) et ensemble non convexe (à droite)

2.3 La dominance

En raison de la nature conflictuelle des objectifs, il n'y a généralement pas une solution réalisable qui optimise simultanément tous les objectifs. L'optimalité dans un contexte multi-objectif est basée sur la notion de dominance et d'efficacité au sens de Pareto.

Définition (dominance)

Soient deux vecteurs critères $w = (w_1, \dots, w_k)$ et $z = (z_1, \dots, z_k)$.

On dit que w domine z si et seulement si :

$$\begin{cases} \forall i \in [1, \dots, k], w_i \leq z_i \\ \exists j \in [1, \dots, k], w_j < z_j \end{cases}$$

C'est-à-dire :

Si w domine z , alors w est au moins aussi bon que z sur tous les critères et meilleur que lui sur au moins un des critères.

Propriétés de la relation de dominance

- N'est pas réflexive car une solution ne se domine pas elle-même.
- N'est pas symétrique, car on n'a jamais $y < z \Rightarrow z < y$.
- N'est pas anti-symétrique car on a jamais $y < z$ et $z < y$.
- Est transitive, car si $y < z$ et $z < w \Rightarrow y < w$.

2.3.1. Dominance forte

Soient deux vecteurs critères $w = (w_1, \dots, w_k)$ et $z = (z_1, \dots, z_k)$.
On dit que w domine fortement z si et seulement si :

$$\forall i \in [1, \dots, k], w_i < z_i$$

C'est-à-dire :

Si w domine fortement z , alors w est meilleur que z sur tous les critères.

2.3.2. Dominance faible

Soient deux vecteurs critères $w = (w_1, \dots, w_k)$ et $z = (z_1, \dots, z_k)$.
On dit que w domine faiblement z si et seulement si :

$$\forall i \in [1, \dots, k], w_i \leq z_i$$

2.3.3. Non-dominance

Soit w^* un vecteur critère tel que $w^* \in F$, on dit que w^* est non-dominé si et seulement s'il n'existe aucun autre vecteur critère $w \in F$ tel que :

$\forall i \in [1, \dots, k], w_i \leq w_i^*$, et $w_i < w_i^*$ pour au moins un indice i .

Dans le cas contraire on dit que w^* est dominé.

2.4 Efficacité

2.4.1 Pareto optimale

Une solution $x^* \in D$ est Pareto optimale si et seulement s'il n'existe pas une solution réalisable $x \in D$ telle que : $f_i(x) \leq f_i(x^*) \quad \forall i = \overline{1, k}$ et $f_j(x) < f_j(x^*)$ pour $j \in \{1 \dots k\}$.

La définition de la Pareto optimalité découle directement de la notion de dominance. Elle signifie qu'il est impossible de trouver une solution qui améliore les performances sur un critère sans que cela entraîne une dégradation des performances sur au moins un autre critère. Elles forment le front Pareto. Les solutions Pareto optimales sont aussi connues sous le nom de solutions efficaces, non dominées ou non inférieures.

2.4.2 Efficacité forte

Une solution $x^* \in D$ est dite fortement efficace, s'il n'existe aucun vecteur $x \in D$ tel que $x \neq x^*$ et $f_i(x) \leq f_i(x^*) \quad \forall i = \{1 \dots k\}$

Une solution est fortement efficace si son vecteur critère est fortement non dominé.

2.4.3 Efficacité faible

Une solution $x^* \in D$ est dite faiblement efficace, s'il n'existe aucun vecteur $x \in D$ tel que $f_i(x) < f_i(x^*) \quad \forall i = \{1 \dots k\}$

Une solution est faiblement efficace si son vecteur critère est faiblement non dominé.

Dominance au sens Geoffrion [22]

Une autre forme de dominance importante dans le monde de l'optimisation multi-objectif est la dominance au sens de Geoffrion, elle est plus forte que l'autre, les solutions optimales obtenues par ce type de dominance sont appelées solutions optimales propres.

Définition

Une solution $x^* \in D$ est appelé solution Pareto optimale propre pour le problème multi-objectif qu'on a noté (MOP) si :

1. x^* est Pareto optimal.

2. S'il existe un nombre $M > 0$ tel que $\forall i = \overline{1, k}, \forall x \in D$ vérifiant

$$f_i(x) < f_i(x^*) \text{ Alors } \exists \text{ au moins un } j \in \{1..k\} \text{ tel que } f_j(x^*) < f_j(x) \text{ et } \frac{f_i(x^*) - f_i(x)}{f_j(x) - f_j(x^*)} \leq M$$

Considérons le problème paramétrique mono-objectif suivant :

$$(P_\lambda) \left\{ \begin{array}{l} \min \sum_{i=1}^k \lambda_i f_i(x) \\ x \in D \end{array} \right.$$

Où λ_i sont des poids relatif à chaque objectif f_i ($i = \overline{1, k}$)

Tel que $\lambda_i \geq 0$ et $\sum_{i=1}^k \lambda_i = 1$. ($i = \overline{1, k}$)

Théorème de Geoffrion (propre) [22]

Si x^* est une solution optimale du problème (P_λ) avec $[\lambda_i > 0 \text{ et } \sum_{i=1}^k \lambda_i = 1]$, ($i = \overline{1, k}$) alors x^* est Pareto optimale propre pour le problème (MOP).

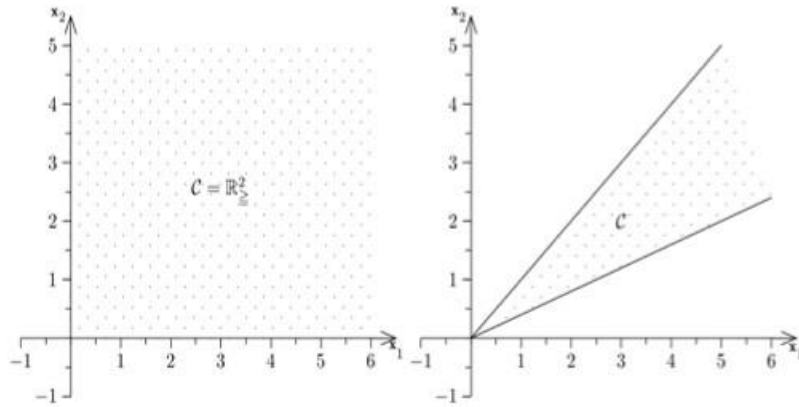
2.5 Optimalité au sens de Pareto

Les solutions qui dominent les autres mais ne se dominent pas entre elles sont appelées solutions optimales au sens de Pareto (ou solutions non dominées). Ensemble des solutions optimales au sens Pareto sont situées sur la limite de l'espace de critère F .

Définition (cône)

Un sous-ensemble $K \subseteq \mathbb{R}^n$ ($K \neq \emptyset$) est appelé un cône si et seulement si :

$$\forall x \in K, \forall \alpha \in \mathbb{R}_+, \alpha x \in K$$



1. Cône négatif

Est défini par :

$$C^- = \{x \in \mathbb{R}^n / f(x) \in \mathbb{R}^k \text{ avec } f(x) \leq 0\}$$

2. Cône positif

Est défini par :

$$C^+ = \{x \in \mathbb{R}^n / f(x) \in \mathbb{R}^k \text{ avec } f(x) \geq 0\}$$

Remarques

- le vecteur $0_{\mathbb{R}^n}$ est contenu dans chaque cône.
- Tous les cônes sont non bornés (sauf le singleton $\{0_{\mathbb{R}^n}\}$ qui contient uniquement l'origine).

Théorème du contact

Un vecteur $x \in \mathbb{R}^n$ est optimal au sens de Pareto pour un problème d'optimisation multi-objectif donné si et seulement si :

$$\{C^- + x\} \cap D = \{x\}$$

$$\{C^+ + x\} \cap D = \{x\}$$

Où D désigne l'espace des solutions réalisables.

L'utilisation de ce théorème est illustrée par la figure ci-dessus :

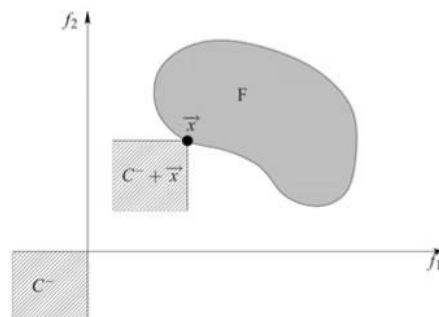


Figure 2.3 théorème de contact

Définition (optimalité locale au sens de Pareto)

Un vecteur $x \in \mathbb{R}^n$ est optimal localement au sens de Pareto s'il existe un réel $r > 0$ tel qu'il n'y ait pas de vecteur x' qui domine le vecteur x avec $x' \in \mathbb{R}^n \cap B(x, r)$ où $B(x, r)$ représente une boule de centre x et de rayon r . Un vecteur x est donc optimal localement au sens de Pareto s'il est optimal au sens de Pareto sur une restriction de l'ensemble D . Cette définition est illustrée par la figure suivante :

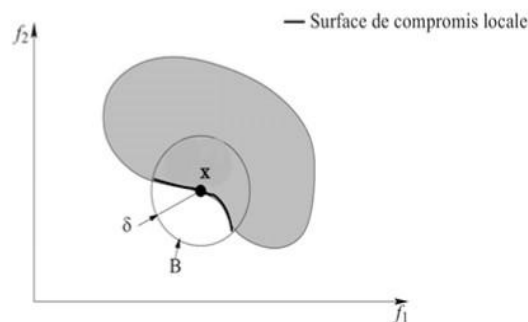


Figure 2.4 l'optimalité locale au sens de Pareto

Définition (optimalité globale au sens de Pareto)

Un vecteur x est optimal globalement au sens de Pareto (ou optimal au sens de Pareto) s'il n'existe pas de vecteur x' tel que x' domine le vecteur x .

La différence entre cette définition et celle de l'optimalité locale tient dans le fait que l'on ne considère plus une restriction de l'ensemble D . Une version "graphique" de la précédente définition utilise le théorème du contact.

2.6 Caractérisations des solutions efficaces

Nous présentons quelques caractérisations qui permettent de tester l'efficacité d'une solution réalisable :

2.6.1 Théorème d' Ecker et wendell

Soit $x^* \in D$ un vecteur de décision donné et \tilde{x} un vecteur quelconque de D .

x^* est efficace pour (MOP) $\Leftrightarrow x^*$ est solution optimale du problème auxiliaire :

$$(P_{\tilde{x}}) \left\{ \begin{array}{l} \text{minimiser } \sum_{i=1}^k f_i(x) \\ x \in D \\ f_i(x) \leq f_i(\tilde{x}), \forall i = 1..k \end{array} \right.$$

Exemple

Soit le problème suivant :

$$(P) \left\{ \begin{array}{l} \min_{(x_1, x_2)} (-3x_1 - x_2 ; x_1 + 2x_2) \\ 3x_1 - x_2 \leq 6 \\ x_2 \leq 2 \\ (x_1, x_2) \in \mathbb{R}_+^2 \end{array} \right.$$

On a Pour $\tilde{x} = (1,1)$ solution réalisable.

$$(P_{\tilde{x}}) \left\{ \begin{array}{l} \min \sum_{i=1}^2 f_i(x) \\ 3x_1 - x_2 \leq 6 \\ x_2 \leq 2 \\ (x_1, x_2) \in \mathbb{R}_+^2 \end{array} \right.$$

$$f_1(x) \leq f_1(\tilde{x}) = f_1(1,1) \Rightarrow -3x_1 - x_2 \leq -4.$$

$$f_2(x) \leq f_2(\tilde{x}) = f_2(1,1) \Rightarrow x_1 + 2x_2 \leq 3.$$

En résolvant $(P_{\tilde{x}})$ par la méthode du simplexe, sa solution optimale est $x^* = (2,0)'$.

Donc x^* est efficace pour (P) .

2.6.2 Théorème de Benson

Soit le problème uni-critère suivant :

$$(P_{\varepsilon}(x^*)) \left\{ \begin{array}{l} \text{maximiser } \theta = \sum_{i=1}^k \varepsilon_i \\ x \in D \\ f_i(x) + \varepsilon_i = f_i(x^*) \quad \forall i = 1..k \\ \varepsilon_i \geq 0; \forall i = 1..k \end{array} \right.$$

Soit x^* une solution réalisable donné, alors :

1- x^* est efficace pour le problème (MOP) \Leftrightarrow la valeur optimale de la fonction objectif est nulle dans $(P_\varepsilon(x^*))$.

2- Si $\theta_{opt} \neq 0$ pour la solution \tilde{x} de $(P_\varepsilon(x^*))$ alors : \tilde{x} est efficace pour (MOP).

Exemple

Soit le problème multi-objectif (MOP) suivant :

$$(MOP) \begin{cases} \text{minimiser } (x_1 - 2x_2 ; -x_1 + 4x_2 ; 2x_1 + x_2) \\ -2x_1 + x_2 \leq 0 \\ x_1 \leq 3 \\ x_2 \leq 2 \\ (x_1, x_2) \in \mathbb{R}_+^2 \end{cases}$$

Vérifions si la solution réalisable $x_1 = (0,0)'$ est efficace en utilisant le test de Benson .

$$(P_\varepsilon(x_1)) \begin{cases} \max \theta = \varepsilon_1 + \varepsilon_2 + \varepsilon_3 \\ -2x_1 + x_2 \leq 0 \\ x_1 \leq 3 \\ x_2 \leq 2 \\ f_1(x) + \varepsilon_1 = f_1(x_1) \Rightarrow x_1 - 2x_2 + \varepsilon_1 = 0 \\ f_2(x) + \varepsilon_2 = f_2(x_1) \Rightarrow -x_1 + 4x_2 + \varepsilon_2 = 0 \\ f_3(x) + \varepsilon_3 = f_3(x_1) \Rightarrow 2x_1 + x_2 + \varepsilon_3 = 0 \\ x_1, x_2, \varepsilon_1, \varepsilon_2, \varepsilon_3 \geq 0 \end{cases}$$

La solution optimale de $(P_\varepsilon(x_1))$ est $x_1 = 0, x_2 = 0, \varepsilon_1 = 0, \varepsilon_2 = 0, \varepsilon_3 = 0$

$\Rightarrow \theta_{opt} = 0 \Rightarrow x_1 = (0,0)'$ est efficace pour (MOP).

2.7 Points spéciaux et matrice des gains

On pose $Z_i = F_i$

2.7.1 Le point idéal

Le vecteur idéal $Z^* = (z_1^*, z_2^*, \dots, z_k^*) \in \mathbb{R}^k$ est le vecteur qui minimise chacune des fonctions objectifs F_i séparément :

C'est-à-dire : $Z_i^* = \min(F_i(x), x \in D)$

Il est clair que si le vecteur idéal est réalisable, il est la solution du problème (MOP). Mais ce n'est pas en général possible à cause des conflits qui existent entre les critères.

2.7.2 Le point anti-idéal

Le vecteur anti-idéal $Z^a = (z_1^a, z_2^a, \dots, z_k^a) \in \mathbb{R}^k$ est le vecteur qui maximise chacune des fonctions objectifs F_i :

C'est-à-dire : $Z^a = \max(F_i(x), x \in D)$

2.7.3 Le point de référence :

C'est un vecteur $(\bar{f}_1, \bar{f}_2, \dots, \bar{f}_k)$ dont les coordonnées sont les valeurs souhaitables que l'on doit atteindre.

2.7.4 matrice des gains :

Nous appelons matrice des gains, une matrice qui est formée en utilisant les vecteurs de décision obtenus lors du calcul du vecteur critère idéal.

La $i^{\text{ème}}$ ligne de la matrice des gains correspond aux valeurs de toutes les fonctions objectifs calculées au point où f_i atteint sa valeur minimale.

Alors $Z_i^* = M_i$ le point idéal est sur la diagonale principale de la matrice :

$$\begin{pmatrix} M_1 & f_1(\bar{x}^2) \cdots f_1(\bar{x}^k) \\ f_2(\bar{x}^1) & M_2 & \cdots f_2(\bar{x}^k) \\ \vdots & \vdots & \ddots & \vdots \\ f_k(\bar{x}^1) & f_k(\bar{x}^2) \cdots & M_k \end{pmatrix}$$

2.7.5 Le point Nadir

Le point critère nadir est le vecteur dont chaque composante Z_i^{nad} correspond à la valeur maximale de la $i^{\text{ème}}$ ligne de la matrice des gains.

2. 7.6 Front de Pareto

C'est l'ensemble des vecteurs de décision qui ne sont pas dominé.

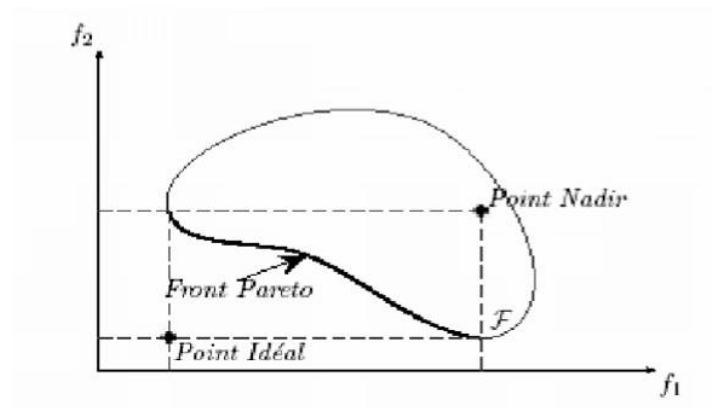


Figure 2.5 point nadir et point Idéal

2.8 Méthodes de résolution d'un programme multi-objectif:

2.8.1 Méthode du simplexe multi-objectif [8]

La méthode du simplexe multicritère consiste à générer un premier point efficace à partir d'une solution de base réalisable, puis à recenser (énumérer) tous les autres points efficaces. Cependant cette méthode ne teste pas toutes les bases car certaines sont dominées de manières évidentes.

Etant donnée le problème:

$$\begin{cases} \max F_1 = c'^1 x \\ \max F_2 = c'^2 x \\ \vdots \\ \max F_k = c'^k x \\ Ax = b, \quad x \geq 0 \end{cases}$$

$$D = \{x \in \mathbb{R}^n / Ax = b, x \geq 0\}$$

Le tableau du simplexe (en considérant séparément chaque objectif) est donné par le tableau (1).

Dans ce tableau, on suppose sans perte de généralité que les m première variables sont dans la base, soient:

J_B : Ensemble des indices basiques.

$J_H = J \setminus J_B$: Ensemble des indices hors base.

						C^1	C_1^1	C_m^1	C_{m+1}^1	C_j^1
						C^2	C_1^2	C_m^1	C_{m+1}^2	C_j^2
						\vdots				\vdots		\vdots	
						C^k	C_1^k	C_m^k	C_{m+1}^k	C_j^k
C_B^1	C_B^2	C_B^j	C_B^k	base	B	a_1		a_m	a_{m+1}	a_j
C_1^1	C_1^2	C_1^j	C_1^k	a_1	b_1	1		0	x_{1m+1}	x_{1j}
\vdots	\vdots	\vdots	\vdots	\vdots		0	\vdots	0	\vdots	\vdots		\vdots	
\vdots	\vdots	\vdots	\vdots	\vdots		0	\vdots	0	\vdots	\vdots		\vdots	
C_m^1	C_m^2	C_m^j	C_m^k	a_m	b_m	0		1	x_{mm+1}	x_{mj}
						0	0	0	Δ_{m+1}^1		Δ_j^1		
						\vdots	0	0	Δ				
						0	0	\vdots	Δ_{m+1}^i	Δ_j^1
						\vdots	0	\vdots	\vdots				
						0	0	0	Δ_{m+1}^k	Δ_j^k	

$$F_0 = \begin{pmatrix} F_0^1 = C^1 b \\ F_0^2 = C^2 b \\ \vdots \\ F_0^k = C^k b \end{pmatrix}$$

Le teste d'efficacité :

Théorème 1 Soit $\bar{x} \in D$ est efficace (respectivement faiblement efficace)

Si et seulement s'il existe :

$$\lambda \in \Lambda = \left\{ \lambda \in \mathbb{R}^k / \sum_{j=1}^k \lambda_j = 1, \lambda_j > 0 \right\}$$

(Respectivement $\exists \lambda \in \bar{A} = \{\lambda \in \mathbb{R}^k / \sum_{j=1}^k \lambda_j = 1, \lambda_i \geq 0\}$) tel que \bar{x} minimise le problème des sommes pondérées donné par :

$$(P_\lambda) \quad \begin{cases} \min \lambda' C x \\ \text{s.c. } x \in D \end{cases}$$

Théorème 2 [9]

Si D possède un point efficace, alors au moins un point extrême de D est efficace.

Preuve

Soit \bar{x} un point efficace de D . D'après le théorème 1, il existe $\lambda \in A$ tel que $\lambda C' \bar{x} = \min_{x \in D} \lambda C' x$

Comme une fonction linéaire atteint son maximum en un point extrême, donc \bar{x} est extrême efficace.

Théorème 3

Soit $x \in D$ un point extrême associé à une base efficace B alors x est efficace.

Preuve

Puisqu'il existe un $\lambda \in A$ pour lequel B est une base optimale par le théorème 2, x est efficace.

Théorème 4

Soient B et \bar{B} deux bases efficaces adjacentes obtenues à partir d'un pivot efficace, et soient x et \bar{x} les points extrêmes associés à B et à \bar{B} respectivement. Alors, l'arête (x, \bar{x}) est efficace.

Théorème 5 [9]

Soit (x, v) une arête efficace infinie de D .

Alors, x est un point extrême associé à une base efficace B .

De la théorie du simplexe uni-critère on a :

$$1- \Delta_j^i = \sum_{r \in J_B} C_r^i x_{rj} - C_j^i \quad j \in J_H, \forall i = \overline{1, k}$$

$$\text{Et } F_0^i = \sum_{r \in J_B} C_r^i b_r \quad \forall i = \overline{1, k}$$

Si :

$\Delta_j^i \geq 0, \forall j \in J_H$, alors $x^0 = (b, 0)'$, ($b \in \mathbb{R}_+^m, 0 \in \mathbb{R})^{n-m}$ est une solution optimale pour le critère i .

2- si on introduit la $j^{ième}$ variable dans la base, nous obtenons une nouvelle solution x_1 et un vecteur :

$$\hat{F}_0 = F_0 - \theta_j \Delta_j$$

$$\text{Où : } \hat{F}_0 = \begin{pmatrix} \hat{F}_0^1 \\ \hat{F}_0^2 \\ \vdots \\ \hat{F}_0^k \end{pmatrix} = \begin{pmatrix} F_0^1 \\ F_0^2 \\ \vdots \\ F_0^k \end{pmatrix} - \theta_j \begin{pmatrix} \Delta_j^1 \\ \Delta_j^2 \\ \vdots \\ \Delta_j^k \end{pmatrix}$$

$$3- \theta_j = \min \left\{ \frac{b_r}{x_{rj}} / x_{rj} > 0 \right\} \quad \forall j \in J_H, r \in J_B.$$

Remarque 1

1- Soit x° une solution basique réalisable.

a- S'il existe un $j \in J_H$ tel que tous les $\Delta_j^i \leq 0$, avec au moins une inégalité stricte et si $\theta_j > 0$, alors la solution courante x° est dominée.

En effet

Si on introduit la $j^{ième}$ variable à la base, on obtient un point extrême adjacent x^1

Pour lequel $\hat{F}_0 \geq F_0$ avec au moins une inégalité stricte, car :

$$\theta_j \Delta_j \leq 0, \text{ donc } \hat{F}_0 = F_0 - \theta_j \Delta_j \geq F_0$$

b- S'il existe un $j \in J$ tel que $\Delta_j^i \geq 0$, avec au moins une inégalité stricte et si de plus $\theta_j > 0$ alors l'introduction de la $j^{ième}$ variable à la base mène à une solution dominée.

En effet

si on introduit la variable j dans la base, on aura un nouveau point extrême x_1 pour lequel :

$$\hat{F}_0 = F_0 - \theta_j \Delta_j \leq F_0 \text{ avec au moins une inégalité stricte.}$$

2- Soit x° une solution basique réalisable s'il existe $j_1, j_2 \in J_H$ tel que :

$\theta_{j_1} \Delta_{j_1} \leq \theta_{j_2} \Delta_{j_2}$ avec au moins une inégalité stricte, alors l'introduction de la variable d'indice j_2 dans la base conduit à une solution dominée par celle résultante de l'introduction de la variable d'indice j_1 .

En effet

$\theta_{j_1} \Delta_{j_1} \leq \theta_{j_2} \Delta_{j_2}$ avec au moins une inégalité stricte, cela implique que :

$$F_0 - \theta_{j_2} \Delta_{j_2} \leq F_0 - \theta_{j_1} \Delta_{j_1} \text{ avec au moins une inégalité stricte.}$$

Remarque 2

1- S'il existe un indice $i \in \{1, \dots, k\}$ tel que $\Delta_j^i \geq 0 \quad \forall j \in J_H$, alors la $i^{ième}$ critère est à son maximum et la solution basique correspondante est non dominée à condition qu'il n'existe pas de colonne k' avec $\Delta_{k'}^i = 0$.

2- D'après la remarque précédente, seules les colonnes (variables) non comparables à zéro et les variables x_j , et x tel que :

$\theta_k \Delta_k$ incomparable à $\theta_j \Delta_j$ sont admissibles pour une introduction dans la base.

Dans ce cas, on ne peut pas dire si la solution correspondante est dominée ou pas.

Pour cela on considère le test dit de non dominance énoncé par le théorème suivant :

Théorème : (Test de dominance)

soit le problème :

$$\begin{aligned} \max v &= \sum_{i=1}^k \varepsilon_i \\ C'x - \delta &= C'\bar{x} \\ x \in D &= \{x \in \mathbb{R}^n / Ax = b, x \geq 0\} \\ \delta \in \mathbb{R}^n, \delta &\geq 0, \bar{x} \in D. \end{aligned}$$

Alors :

- \bar{x} est efficace si et seulement si : $\max v = 0$
- \bar{x} est dominée si et seulement si : $\max v > 0$

Exemple

Soit le problème multi-critères suivant :

$$\begin{aligned} \max Z_1 &= 0.4x_1 + 0.3x_2 \\ \max Z_2 &= x_1 \\ (1) \quad \begin{cases} x_1 + x_2 \leq 400 \\ 2x_1 + x_2 \leq 500 \\ x_1 \geq 0, x_2 \geq 0 \end{cases} \end{aligned}$$

$$\begin{aligned} (1) \Leftrightarrow \max Z_1 &= 0.4x_1 + 0.3x_2 \\ \max Z_2 &= x_1 \\ \begin{cases} x_1 + x_2 + x_3 &= 400 \\ 2x_1 + x_2 + x_4 &= 500 \\ x_j \geq 0 \quad \forall j = \overline{1,4} \end{cases} \end{aligned}$$

$x_1 = (0,0,400,500)'$ est une solution de base réalisable associée à la base.

$$(a_3, a_4) = I_2 \quad \text{et} \quad z(x) = \begin{pmatrix} z_1(x) \\ z_2(x) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Le tableau de simplexe initial est :

			C^1	0.4	0.3	0	0
			C^2	1	0	0	0
C_B^1	C_B^2	Base	B	a_1	a_2	a_3	a_4
0	0	a_3	400	1	1	1	0
0	0	a_4	500	2	1	0	1
		$Z_0 = \begin{pmatrix} z_0^1 = 0 \\ z_0^2 = 0 \end{pmatrix}$	Δ^1	-0.4	-0.3	0	0
			Δ^2	-1	0	0	0

$$J_H = \{1,2\} \quad \text{et} \quad J_B = \{3,4\}$$

$\nexists i \in \{1,2\}$ tel que $\Delta_j^i \geq 0$ tel que $\forall j \in J_H = \{1,2\}$, donc aucun objectif n'est à son maximum.

On a :

$$\Delta_1 \begin{pmatrix} -0.4 \\ -1 \end{pmatrix} \leq 0, x_1 \text{ est dominé.}$$

$$\theta_1 = \min \left\{ \frac{400}{1}, \frac{500}{2} \right\} = 250 \quad (\text{atteint pour } r = 4)$$

$$\theta_2 = \min \left\{ \frac{400}{1}, \frac{500}{1} \right\} = 400 \quad (\text{atteint pour } r = 3)$$

$$\theta_1 \Delta_1 = 250 \begin{pmatrix} -0.4 \\ -1 \end{pmatrix} = \begin{pmatrix} -100 \\ -250 \end{pmatrix}$$

$$\theta_2 \Delta_2 = 400 \begin{pmatrix} -0.3 \\ 0 \end{pmatrix} = \begin{pmatrix} -120 \\ 0 \end{pmatrix}$$

Donc $\theta_1 \Delta_1$ est incomparable à $\theta_2 \Delta_2$

Introduisons a_1 dans la base à la place de a_4 , on aura alors le tableau du simplexe suivant :

			C^1	0.4	0.3	0	0
			C^2	1	0	0	0
C_B^1	C_B^2	Base	B	a_1	a_2	a_3	a_4
0	0	a_3	150	0	$1/2$	1	$-1/2$
0.4	1	a_1	250	1	$1/2$	0	$1/2$
		$Z_0 = \begin{pmatrix} z_1^1 = 100 \\ z_1^2 = 250 \end{pmatrix}$	Δ^1	0	-0.1	0	0.2
			Δ^2	0	$1/2$	0	$1/2$

→

↑

$\exists i = 2$ tel que $\Delta_j^i > 0 \forall j \in J_H$, ce qui implique que le deuxième critère est à son maximum.

Le point extrême $x_2 = (250, 0, 150, 0)'$ est non dominé.

$$\theta_2 = \min \left\{ \frac{150}{0.5}, \frac{250}{0.5} \right\} = 300 \text{ (atteint pour } r = 3)$$

$$\theta_4 = \min \left\{ \frac{250}{0.5} \right\} = 500 \text{ (atteint pour } r = 1)$$

$$\theta_2 \Delta_2 = 300 \begin{pmatrix} -0.1 \\ 0.5 \end{pmatrix} = \begin{pmatrix} -30 \\ 150 \end{pmatrix}$$

$$\theta_4 \Delta_4 = 500 \begin{pmatrix} 0.2 \\ 0.5 \end{pmatrix} = \begin{pmatrix} 100 \\ 250 \end{pmatrix}$$

$$\theta_2 \Delta_2 < \theta_4 \Delta_4.$$

Donc a_2 rentre dans la base et a_3 sort de la base d'après la remarque.

Dressons alors le troisième tableau simplexe :

			C^1	0.4	0.3	0	0
			C^2	1	0	0	0
C_B^1	C_B^2	Base	B	a_1	a_2	a_3	a_4
0.3	1	a_2	300	0	1	2	-1
0.4	1	a_1	100	1	0	-1	1
		$Z_0 = \begin{pmatrix} z_1^1 = 130 \\ z_1^2 = 100 \end{pmatrix}$	Δ^1	0	0	0.2	0.1
			Δ^2	0	0	-1	1

↑

$\Delta^1 \geq 0$, alors le premier objectif est à son maximum et puisqu'il n'existe pas de colonne k telle que $\Delta_k^1 = 0$ avec $k \in J_H$, alors $x_3 = (100, 300, 0, 0)^t$ est non dominée.

$$\theta_3 = \min \left\{ \frac{300}{2} \right\} = 150 \text{ (atteint pour } r=2)$$

$$\theta_4 = \min \left\{ \frac{100}{1} \right\} = 100 \text{ (atteint pour } r=1)$$

$$\theta_3 \Delta_3 = 150 \begin{pmatrix} 0.2 \\ -1 \end{pmatrix} = \begin{pmatrix} 30 \\ -150 \end{pmatrix}$$

$$\theta_4 \Delta_4 = 100 \begin{pmatrix} 0.1 \\ 1 \end{pmatrix} = \begin{pmatrix} 10 \\ 100 \end{pmatrix}$$

On remarque que $\theta_3 \Delta_3$ est incomparable à $\theta_4 \Delta_4$.

Puisque l'introduction de a_3 dans la base conduit à une base déjà explorée, donc on introduira a_4 .

Dressons alors le quatrième tableau du simplexe :

			C^1	0.4	0.3	0	0
			C^2	1	0	0	0
C_B^1	C_B^2	Base	B	a_1	a_2	a_3	a_4
0.3	0	a_2	400	1	1	1	0
0	0	a_4	100	1	0	-1	1
		$Z_0 = \begin{pmatrix} z_1^1 = 120 \\ z_1^2 = 0 \end{pmatrix}$	Δ^1	-0.1	0	0.3	0
			Δ^2	-1	0	0	0

$\nexists i \in \{1,3\}$ tel que $\Delta_j^i \geq 0 \forall J_H \in \{1,3\}$, donc aucun objectif n'est à son maximum.

$\Delta_1 = \begin{pmatrix} -0.1 \\ -1 \end{pmatrix} < 0 \Rightarrow x_4 = (0,400,0,100)'$ est dominé.

Finalement,

En considérant le problème initial (1) seuls les points extrêmes $x^2 = (250,0)$.

Et $x^3 = (100,300)$ sont non dominés avec :

$z(x_*^2) = \begin{pmatrix} 100 \\ 250 \end{pmatrix}$ et $z(x_*^3) = \begin{pmatrix} 130 \\ 100 \end{pmatrix}$.

On a recensé deux solutions efficaces.

La méthode du simplexe multicritère nous apporte clarté et précision dans toutes ses étapes donc facilité d'application, cependant en pratique, elle peut être non efficace, car la solution désirée peut être sur arête et non un point extrême efficace.

2.8.2 Méthode somme pondérée ou méthode d'agrégation des objectifs [21]

Cette méthode, probablement la plus largement utilisée dans la pratique. Elle consiste à ramener le problème (MOP) au problème de l'optimisation d'une combinaison des objectifs initiaux suivant :

$$(P_\lambda) \left\{ \begin{array}{l} \min/\max \sum_{i=1}^k \lambda_i f_i(x) \\ x \in D \end{array} \right.$$

Où les poids $\lambda_i \geq 0$; $i = \overline{1, k}$ et $\sum_{i=1}^k \lambda_i = 1$.

(Les poids λ_i sont choisis en fonction de l'importance relative des objectifs).

Théorème 1

Toute solution optimale du problème (P_λ) est une solution Pareto-optimale pour le problème (MOP) si tous les λ_i sont strictement positifs. $i = \overline{1, k}$

Théorème 2

Si le problème (MOP) est convexe (en particulier linéaire), si x^* est solution Pareto-optimale de (MOP) alors $\exists \lambda = (\lambda_1, \lambda_2, \dots, \lambda_k)$ avec : $(\lambda_i > 0 \quad i = \overline{1, k})$ chaque x^* est solution optimale de (P_λ) .

2.8.3 Méthode ε -contraintes

Une autre façon de transformer le problème (MOP) en un problème simple objectif est de minimiser (respectivement maximiser) l'objectif f_i jugé prioritaire par le décideur et de convertir les autres objectifs en contraintes.

Le problème transformé s'écrit :

$$(P_\varepsilon) \begin{cases} \text{minimiser } f_i(x) \\ x \in D \\ f_j(x) \leq \varepsilon_j, \forall i \neq j \quad (j = \overline{1, k}) \end{cases}$$

Les $\varepsilon_j \in \mathbb{R}$ sont appelés « seuil de satisfaction »

Pour un problème (MOP) à maximum :

$$(P_\varepsilon) \begin{cases} \text{maximiser } f_i(x) \\ x \in D \\ f_j(x) \geq \varepsilon_j, \forall i \neq j \quad (j = \overline{1, k}) \end{cases}$$

Théorème 3

Pour tout vecteur $\varepsilon = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_{i-1}, \varepsilon_{i+1}, \dots, \varepsilon_k)$ tel que le problème (P_ε) a une solution optimale alors cette solution est Pareto-optimale pour le problème (MOP).

Remarques

1. Les paramètres ε_j sont choisis par le décideur.
2. En faisant varier les ε_j dans les deux méthodes, on peut générer un sous-ensemble de l'ensemble Pareto du (MOP).

Exemple

Résolution de problème (MOP) suivant avec la méthode ε -contraintes en considérant le critère 3 comme prioritaire et $\varepsilon = (\varepsilon_1, \varepsilon_2) = (1, 3)$.

$$(MOP) \left\{ \begin{array}{l} \min f(x) = (f_1(x) = -3x_1 + x_2; f_2(x) = x_1 + 2x_2; f_3(x) = 2x_1 + x_2) \\ 3x_1 - x_2 \leq 6 \\ x_2 \leq 2 \\ x = (x_1, x_2) \in \mathbb{R}_+^2 \end{array} \right.$$

$$(P_\varepsilon) \left\{ \begin{array}{l} \text{minimiser } f_{i=3}(x) \\ x \in D \\ f_j(x) \leq \varepsilon_j, \forall j \neq 3 \ (j = \overline{1,3}) \end{array} \right.$$

$$(P_\varepsilon) \left\{ \begin{array}{l} \min 2x_1 + x_2 \\ 3x_1 - x_2 \leq 6 \\ x_2 \leq 2 \\ f_1(x) \leq \varepsilon_1 \rightarrow -3x_1 + x_2 \leq 1 \\ f_2(x) \leq \varepsilon_2 \rightarrow x_1 + 2x_2 \leq 3 \\ x = (x_1, x_2) \in \mathbb{R}_+^2 \end{array} \right.$$

En résolvant (P_ε) par le simplexe, sa résolution optimale $x^* = (x_1, x_2) = (0,0)'$.
Donc d'après le théorème 3, $x^* = (0,0)'$ est Pareto-optimale pour le problème (MOP).

Conclusion

Dans les trente dernières années, la plupart des travaux réalisés dans ce domaine ont porté sur la programmation linéaire multi-objectif. Les raisons principales de cet intérêt sont d'une part le développement de la programmation linéaire mono-objectif en recherche opérationnelle et la facilité relative de traiter de tels problèmes, et d'autre part l'abondance de cas pratiques pouvant être formulés sous cette forme.

CHAPITRE 3

Programmation multi-objectif linéaire en nombre entiers

3.1 Introduction

Dans un problème multi-objectif, lorsque, les fonctions $Z_i(x)$ pour $i = 1, \dots, r$ sont linéaires et que l'ensemble des solutions réalisables D est défini par des contraintes linéaires, alors, on parle de problème linéaire multi-objectif (MOLP). Ainsi, si les variables x prennent des valeurs entières dans le problème (MOLP), on parle alors de problème d'optimisation multi-objectif linéaire en nombres entiers noté (MOLPE).

3.2 Formulation du problème

Un problème (MOLPE) est formulé comme suit :

$$(\text{MOLPE}) \begin{cases} \max(Z_1, Z_2, \dots, Z_r) \\ Z_i = C_i'x \\ x \in D \end{cases}$$

Où : $D = \{x \in \mathbb{R}^n / Ax \leq b, x \geq 0, x \text{ entier} \}$

r : Nombres d'objectifs à maximiser.

C_i : Vecteur des coefficients pour $i = 1, \dots, r$.

A : La (m, n) matrice des contraintes, $rgA = m \leq n$.

Z_i : Critère à maximiser pour $i = 1, \dots, r$.

3.3 Définitions et théorèmes [12]

Définition 1 (l'Optimalité de Pareto)

$x^* \in D$ Est efficace si et seulement s'il n'existe aucun autre vecteur

$x \in D$ tel que :

$$c'x \geq c'x^*, c'x \neq c'x^*.$$

Autrement dit :

$x^* \in D$ est efficace si et seulement s'il n'existe aucun autre vecteur $x \in D$

Tel que $Z_i(x) \geq Z_i(x^*)$ pour $i = 1, \dots, r$ et $Z_i(x) < Z_i(x^*)$ pour au moins un indice i .

Définition 2 (l'optimalité de Slater ou faible efficacité)

$x^* \in D$ est dite faiblement efficace si et seulement s'il n'existe aucun autre vecteur $x \in D$ tel que $c'x > c'x^*$.

Autrement dit :

$x^* \in D$ est faiblement efficace si et seulement s'il n'existe aucun autre vecteur $x \in D$ tel que $Z_i(x) > Z_i(x^*)$ pour $i = 1, \dots, r$.

Remarque 1

Dans le cas où le décideur veut minimiser les critères Z_i , $i = 1, \dots, r$, on a des définitions analogues aux définitions 1 et 2 il suffit d'inverser les inégalités correspondantes car $\min Z_i(x) = -\max(-Z_i(x))$.

Notons par X^s l'ensemble des solutions de Slater et par X^p l'ensemble des solutions de Pareto.

Théorème 1

Soit $x^0 \in D$, alors :

S'il existe $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_r) \geq 0$ tel que

$$\sum_{i=1}^r \lambda_i Z_i(x^0) = \max_{x \in D} \sum_{i=1}^r \lambda_i Z_i(x), \text{ alors } x^0 \in X^s, \quad \sum_{i=1}^r \lambda_i = 1.$$

S'il existe $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_r) > 0$ tel que,

$$\sum_{i=1}^r \lambda_i Z_i(x^0) = \max_{x \in D} \sum_{i=1}^r \lambda_i Z_i(x), \quad x^0 \in X^p \text{ et } \sum_{i=1}^r \lambda_i = 1.$$

Ce résultat signifie que pour trouver un optimum de Pareto ou de Slater, il suffit d'optimiser la fonction représentant une combinaison linéaire pondérée des critères. Cependant, le théorème 1 ne permet pas la caractérisation de toutes les solutions optimales selon Pareto ou selon Slater.

3.4. Quelques méthodes de résolution d'un problème multi-objectif linéaire en nombres entiers

3.4.1 Méthode de Gupta [13]

Gupta a proposé une méthode avec deux techniques pour la construction de l'ensemble des solutions efficaces d'un problème (MOLPE). La première procédure de cette méthode est basée sur la méthode des coupes qui consiste à restructurer l'ensemble S des solutions réalisables du problème relaxé correspondant (sans la condition d'intégrité). La deuxième procédure est basée sur le contrôle de fonctions objectives.

Soit le problème (P_1) qui optimise le premier objectif (ou un objectif quelconque) de (MOLPE), qui est défini comme suit :

$$(P_1) \begin{cases} \max Z_1 = C'x \\ x \in D \end{cases}$$

Où $D' = \{x \in \mathbb{R}^n / Ax = b, x \geq 0\}$ i.e c'est l'ensemble des solutions réalisables du problème relaxé.

Notons par :

S_1 : La région obtenue après application des coupes de Gomory.

$Z_1^* = Z_1^1$ = La valeur optimale de Z_1 dans le problème (P_1) .

$x_1^1 = (x_{1,j})$: La solution entière correspondante à Z_1^* . C'est la solution basique réalisable de la région obtenue après les coupes de Gomory sur S_1 .

Z_i^* : La valeur de Z_i pour $i \in I' = \{2, \dots, k, \dots, r\}$ correspondant à x_1^1 .

$(Z_1^1, Z_2^1, \dots, Z_r^1)$ Représente le premier r-uplet efficace.

x_1 = Solution correspond aux $(Z_1^1, Z_2^1, \dots, Z_r^1)$.

B_1 = base associée à S_1 .

$a_{1,j}$: Vecteur activité de $x_{1,j}$ approprié à la région tronquée.

$y_{1,j} : (B_1^{-1})a_{1,j}$.

$I_1 = \{j/a_1, j \in B_1\}$.

$N_1 = \{j/a_1, j \notin B_1\}$.

$J_1 = \{j/j \in N_1 \text{ et } Z_{1,j}^{c^1} - C_j^1 = 0\}$ où :

$Z_{1,j}^{c^1} = C_{B_1}^1 y_{1,j}$, C_j^1 est la J^{eme} composante du vecteur C^1 et $C_{B_1}^1$ est le vecteur des coûts des variables basique associées à B_1 , dans C^1 .

Pour $k \geq 2$,

S_k est la région de S_1 obtenue après l'application de la coupe :

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

$x_k = (x_{k,j})$ est la solution entière obtenue dans la région tronquée de la coupe

par $\sum_{j \in N_{k-1} \setminus \{j_{k-1}\}} x_j \geq 1$, ou $j_{k-1} \in \Gamma_{k-1}$.

B_k : base associée à x_k .

a_{kj} : Vecteur activité de $x_{k,j}$ approprié à la région tronquée.

$$y_{kj} = B_k^{-1} a_{kj}.$$

$$I_k = \{j/a_{k,j}, j \in B_k\}$$

$$\Gamma_k = \{j / j \in N_k, Z_{k,j}^{c^1} - C_j^1 > 0 \text{ et } Z_{k,j}^{c^i} - C_j^i < 0 \text{ pour au moins un indice } i \in I'\}$$

Où : $Z_{k,j}^{c^i} = C_{Bk}^i y_{kj}$, C_{Bk}^i est un vecteur des coûts des variables basiques associées à B_k dans C^i .

Γ_k : L'ensemble correspondant à la solution x_k .

Définition 3

Une arête E_{jk} incidente à x_k est définie par l'ensemble :

$$E_{jk} = \left\{ x = (x_i) \in \mathbb{R}^n / \begin{cases} x_i = x_{ki} - \theta_{jk} y_{k,i_{jk}}, i \in I_k \\ x_{jk} = \theta_{jk} \\ x_i = 0 \text{ pour tout } i \in N_k - \{j_k\} \end{cases} \right\}$$

Où : $0 \leq \theta_{jk} \leq \min \left\{ \frac{x_{ki}}{y_{k,i_{jk}}}, y_{k,i_{jk}} > 0 \right\}$ et θ_{jk} et $\theta_{jk} y_{k,i_{jk}}$ sont des entiers $\forall i \in I_k$.

Définition 4

L'arête incidente à un point entier réalisable x est dite dominée si toutes les solutions le long de cette arête rendent dominé le r-uplet (Z_1, \dots, Z_r) .

Théorème 6

Toutes les solutions entières réalisables du problème (P_1) alternatives à la solution x_1 sur une arête E_{j_1} incidente à x_1 , ($j_1 \in \Gamma_1$) dans S' (ou dans une région tronquée S_k) sont dans le demi espace ouvert :

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

Théorème 7

La solution entière réalisable du problème (P_1) qui n'est pas sur l'arête E_{j_1} , incidente à x_1 , ($j_1 \in \Gamma_1$) est dans le demi espace fermé :

$$\sum_{j \in N_1 - \{j_1\}} x_j \geq 1. \quad (1)$$

Corollaire 2

La solution entière réalisable qui n'est pas sur l'arête E_{j_k} incidente à x_k dans la région tronquée de S_k , $j_k \in \Gamma_k$, $k \geq 2$, est dans le demi espace fermé

$$\sum_{j \in N_k - \{j_k\}} x_j \geq 1. \quad (2)$$

Remarque

1-Les coupes (1) et (2) sont des générations des coupes de Dantzig. Dans le cas où Γ_1 (ou bien Γ_k) est vide, les coupes (1) et (2) correspondant aux de Dantzig sont

$$\sum_{j \in N_1 - \{j_1\}} x_j \geq 1 \quad \text{où} \quad \sum_{j \in N_k - \{j_k\}} x_j \geq 1.$$

Les coupes (1) et (2) sont préférables à celle de Dantzig car elles éliminent seulement un point à la fois.

2- Si pour un $a_j \in \Gamma_1 (\neq 0)$, $\theta = \min \left\{ \frac{x_{ki}}{y_{k,i_{jk}}}, y_{k,i_{jk}} > 0 \right\}$ où $y_{1,j}$ correspond à la solution X_1 , alors aucune solution entière réalisable ne peut être obtenue sur l'arête E_j .

Ci-dessus sont données les différentes étapes pour la résolution du problème (P) .

Algorithme

Etape1 : Calculer la solution optimale entière X_1^1 du problème (P_1)

- Si $\Gamma_1 = \emptyset$, X_1^1 est l'unique solution optimal entière du problème (P_1) de valeur Z_1^1 . Calculer $z_i^1, i \in I_k$ et enregistrer $(z_1^1, z_2^1, \dots, z_r^1)$ comme étant le premier r-uplet.
- Si $\Gamma_1 \neq \emptyset$, déterminer toutes les solutions alternative X_1^1 ainsi des r-uplets. Supprimer tous les r-uplets dominés pour obtenir l'ensemble des r-uplets efficace Eff_0 . Enregistrer $(z_1^1, z_2^1, \dots, z_r^1)$ comme étant le premier r-uplet efficace pour lequel z_2 a une valeur maximal pour tous les r-uplets appartenant à Eff_0 .
- Si toutes les deuxièmes composantes sont identiques, choisir un r-uplet avec la plus petite valeur z_3 . Continuer le processus chaque fois que cette contrainte se présente. Soit $(z_1^1, z_2^1, \dots, z_r^1)$ le r-uplet enregistré

Etape2 : Choisir un indice $j_1 \in \Gamma_1$. Trouver le plus petite θ pour le prochain pivot.

- Si $\theta_{j_1} < 1$, aucune solution réalisable ne peut être sur l'arête E_{j_1} .
- Si $\theta_{j_1} \geq 1$, déterminer toutes les solutions entières réalisables le long de l'arête E_{j_1} .

Soit S_1 l'ensemble de tous les r -uplets correspondant à l'arrête E_{j_1} . L'ensemble Eff_1 est l'ensemble de r -uplets de la forme $(z_1^1, z_2^1, \dots, z_r^1)$ potentiellement non dominées à l'étape 2.

-Tronquer l'arrête E_{j_1} par la coupe : $\sum_{j \in N_1 - \{j_1\}} x_j \geq 1$.

-Appliquer la méthode duale du simplexe et les coupes de Gomory si c'est nécessaire pour avoir une solution entière réalisable $x = (x_{2,j})$ dans la région tronquée S_2 .

Etape3 :

- Choisir $j_2 \in \Gamma_2$ et balayer toutes les solutions qui sont sur l'arrête E_{j_2} correspondant à j_2 .
- Lire les nouveaux r -uplets obtenus.
- Augmenter l'ensemble Eff_1 avec ces r -uplets pour construire l'ensemble Eff_2 .
- Supprimer tous les r -uplets dominés et poser Eff_2 l'ensemble restant.
- Tronquer l'arrête E_{j_2} par la coupe $\sum_{j \in N_2 - \{j_2\}} x_j \geq 1$.

Etape k :

- Choisir $j_{k-1} \in \Gamma_{k-1}$ et explorer $E_{j_{k-1}}$ pour avoir les solutions entières possibles.
- Lire les r -uplets correspondants et augmenter l'ensemble Eff_{k-2} en ajoutant ces r -uplets pour construire Eff_{k-1} .
- Tronquer l'arrête $E_{j_{k-1}}$ par la coupe : $\sum_{j \in N_{k-1} - \{j_{k-1}\}} x_j \geq 1$.
- Obtenir la solution entière optimale X_k de la région. Ceci marque le début de l'état $(k + 1)$.

Etape (k+1) : Ce processus se termine à l'étape $(k + 1)$ quand :

- L'ensemble $\Gamma_k = \emptyset$ et $z_{k,j}^{c^1} - c_j^1 > 0, \forall j \in N_k$.
- $\Gamma_k \neq \emptyset$ mais tout les $j \in \Gamma_k$ produisent des l'arrête dominés.

Remarque : Il arrive parfois que l'algorithme développé par Gupta s'arrête sans avoir énuméré toutes les solutions efficaces. Par conséquent des solutions efficaces pouvant intéresser le décideur peuvent être ratées.

Exemple 3.4.1

considérons le problème suivant :

$$(P) \begin{cases} \min Z_1(x) = -x_1 + 4x_2 \\ \min Z_2(x) = -2x_1 - 5x_2 \\ \text{s. c. } x_1 + 3x_2 \leq 9 \\ 2x_1 + x_2 \leq 7 \\ x_1 - 2x_2 \leq 1 \\ x_1, x_2 \geq 0, \text{ entier} \end{cases}$$

Les résultats de la première étape sont données dans le tableau 3.1 :

La solution optimale entière $x_1 = (1,0)$ est unique dans le premier couple non dominé est $(Z_1, Z_2) = (-1, -2)$. $Eff_0 = \{(-1, -2)\}$

Base	Valeurs des variables de base	x_1	x_2	x_3	x_4	x_5
x_3	8	0	5	1	0	-1
x_4	5	0	5	0	1	-2
x_1	1	1	-2	0	0	1
$Z_{1j}^1 - C_{1j}^1$		0	-2	0	0	-1
$Z_{1j}^2 - C_{1j}^2$		0	9	0	0	-2

Tableau 3.1

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

$\Gamma_1 = \{j \in N_1, Z_{1j}^1 - C_{1j}^1 < 0 \text{ et } Z_{2j}^1 - C_{2j}^1 > 0\} = \{2\}$

$j_1 = 2$, $\min\{\frac{8}{5}, \frac{5}{5}\} = 1 \Rightarrow \theta_2 = 1$. Il existe une seule solution entière alternative à x_1 sur l'arête E_2 . Elle est donnée par $x_1^1 = (3,1)$ avec $(Z_1, Z_2) = (1, -11)$ qui est non dominé, donc $Eff_1 = \{(-1, -2), (1, -11)\}$.

L'ajout de la coupe $x_5 \geq 1$ nous donne la solution optimale $x_2 = (0,0)$.

Le couple $(Z_1, Z_2) = (0,0)$ est dominé par $(-1, -2)$. (voir tableau 3.2).

$I_2 = \{1,3,4,5\}$ et $N_2 = \{2,6\}$

Base	Valeurs des variables de base	x_1	x_2	x_3	x_4	x_5	x_6
x_3	9	0	5	1	0	0	-1
x_4	7	0	5	0	1	0	-2
x_1	0	1	-2	0	0	0	1
x_5	1	0	0	0	0	1	-1
$Z_{1j}^1 - C_{1j}^1$		0	-2	0	0	0	-1
$Z_{1j}^2 - C_{1j}^2$		0	9	0	0	0	-2

Tableau 3.2

$\Gamma_1 = \{j \in N_1, Z_{1j}^1 - C_{1j}^1 < 0 \text{ et } Z_{2j}^1 - C_{2j}^1 > 0\} = \{2\}$

$\min\{\frac{9}{5}, \frac{7}{5}\} = 1 \Rightarrow \theta_2 = 1$. Il existe une seule solution entière alternative à x_2 sur l'arête E_2

C'est $x_2^1 = (2,1)$ avec $(Z_1, Z_2) = (2, -9)$ qui est dominé par le couple $(1, -11)$.

D'après gupta le processus de résolution doit s'arrêter alors qu'il existe encore d'autres solutions efficaces non dominées telles que $x = (0,3)$ et $x = (1,2)$ dont les couples non dominés correspondants sont $(12, -15)$ et $(7, -12)$.

3.4.2 Méthode de Abbas et Moulai [14]

C'est une méthode de coupes. Elle est appropriée à la résolution des problèmes (MOLPE). La première itération consiste à déterminer une solution entière réalisable pour le problème relaxé. La seconde itération nous permet de trouver toutes les solutions entières alternatives à la solution optimale obtenue (si elles existent) et obtenir la première solution entière efficace du problème principal. Dans la prochaine étape, nous introduisons une coupe afin de calculer les autres solutions entières efficaces restantes en utilisant la méthode duale du simplexe. L'algorithme s'arrête quand tout le domaine de décision est exploré. Soit le problème (MOLPE) suivant:

$$(P) \begin{cases} \max Z_1(x) = C^1 x \\ \max Z_2(x) = C^2 x \\ \vdots \\ \max Z_r(x) = C^r x \\ \text{s.c} \\ x \in D \\ x \text{ entier} \end{cases}$$

Cet algorithme adopte les mêmes notations que celles de la méthode de Gupta.

Procédure [14]

Etape1 : Résoudre le problème (P_1) et trouver la solution optimal entière x_1^1 du problème dans S_1 et construire Γ_1 .

Si $\Gamma_1 = \emptyset$, x_1^1 est l'unique solution optimale entière du problème (P_1) de valeur z_1^1 .

- Calculer les valeurs de z_i^1 , $i \in 2, \dots, r$. et enregistrer $(z_1^1, z_2^1, \dots, z_r^1)$

Comme étant le premier r-uplet.

- Tronquer le point x_1^1 par la coupe $\sum_{j \in N_1} x_j \geq 1$.

En appliquant la méthode duale du simplexe et des coupes de Gomory si Nécessaire, on obtient une solution réalisable entière x_2^1 dans la région tronquée S_2 .

- Lire le r-uplet correspondant $(z_1^1, z_2^1, \dots, z_r^1)$ et l'ajouter à Eff_0 s'il n'est pas dominé par l'un des r-uplets efficaces pour obtenir Eff_1 .

Si $\Gamma_1 \neq \emptyset$, choisir un indice $j_1 \in \Gamma_1$, et calculer le nombre θ_{j_1} correspond à x_1^1 .

- Si $\theta_{j_1} < 1$ choisir un indice $j_1 \in \Gamma_1$ et appliquer la coupe : $\sum_{j \in N_1 - j_1} x_j \geq 1$.

- En appliquant la méthode duale du simplexe et si nécessaire des coupes de Gomory successives, on obtient une solution réalisable

entière x_2^1 dans la région tronquée S_2 .

- Enregistrer $(z_1^1, z_2^1, \dots, z_r^1)$ r-uplet dans Eff_0 . s'il n'est pas dominé par l'un des r-uplets efficaces précédemment déterminés pour obtenir Eff_1 .

- Si $\theta_{j_1} \geq 1$, déterminer toutes les solutions entières réalisables $x_1^q, q \in \{2, \dots, p_k\}$. Alternatives à x_1^1 le long de l'arête E_{j_1} .

- Tronquer le point x_1^q par la coupe $\sum_{j \in N_1 - j_1} x_j \geq 1$.

- En appliquant la méthode duale du simplexe et des coupes de Gomory si nécessaire, on obtient une solution réalisable entière x_2^1 dans la région tronquée S_2 . Lire le r-uplet correspondant $(z_1^1, z_2^1, \dots, z_r^1)$ et l'ajouter à Eff_0 s'il n'est pas dominé par l'un des r-uplets efficaces pour obtenir Eff_1 .

Exemple d'application (méthode Abbas & Moulai)

Soit le problème multicritère suivant :

$$(P) \begin{cases} \max Z_1(x) = 2x_1 + 3x_2 \\ \max Z_2(x) = -4x_1 - x_2 \\ \text{s. c} \quad x_1 + 2x_2 \leq 4 \\ \quad \quad 2x_1 + x_2 \leq 2 \\ x_1 \geq 0, \quad x_2 \geq 0 \\ x_1 \in \mathbb{N}, \quad x_2 \in \mathbb{N} \end{cases}$$

Etape 1

Résolution du problème (P_1) :

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

La solution de base efficace (0,2) de la région tronquée S_1 est donnée par le tableau 1 suivant :

x_B	B	x_1	x_2	x_3	x_4
x_2	2	2	1	0	1
x_3	0	-3	0	1	-2
$Z_1^1 = 6$	$Z_j^1 - C_j^1$	4	0	0	3
$Z_1^2 = -2$	$Z_j^2 - C_j^2$	2	0	0	-1

Tableau 1

On a la solution optimale $x_1^1 = (0,2)$ produit le premier r- uplet $(6, -2)$ efficace par suite, $Eff_0 = (6, -2)$.

Ce point entier est tronqué en utilisant la coupe $x_1 + x_4 \geq 1$ on obtient le tableau 2 suivant :

x_B	B	x_1	x_2	x_3	x_4	x_5
x_2	1	1	1	0	0	1
x_3	2	-1	0	1	0	-2
x_4	1	1	0	0	1	-1
$Z_2^1 = 3$	$Z_j^1 - C_j^1$	1	0	0	0	3
$Z_2^2 = -1$	$Z_j^2 - C_j^2$	3	0	0	0	-1

Tableau 2

Le r-uplet $(3, -1)$ correspondant à x_2^1 est non dominée par rapport au précédent de Eff_0 par conséquent, il est ajouté à Eff_0 pour obtenir $Eff_1 = \{(6, -2), (3, -1)\}$ avec : $I = \{2,3,4\}$, $N_1 = \{1,5\}$, $\Gamma_1 = \{j/j \in N_1, Z_j^1 - C_j^1 = 0\} = \emptyset$.

$$Z_2^1(x_2^1) = 3, \quad Z_2^2(x_2^1) = -1$$

Etape 2 :

$\Gamma_1 = \emptyset$, la solution $x_2^1 = (0,1)$ est l'unique solution. Ce point entier est tronqué en utilisant la coupe $x_1 + x_5 \geq 1$.

En utilisant le dual du simplexe, on obtient une solution réalisable : $x_3^1 = (1,0)$ mais n'est pas efficace car $Z_1(x_2^1) > Z_1(x_3^1)$ et $Z_2(x_3^1) > Z_2(x_3^1)$ donnée par le tableau 3 suivant :

x_B	B	x_1	x_2	x_3	x_4	x_5	x_6
x_1	1	1	0	0	0	1	-1
x_2	0	0	1	0	0	0	1
x_3	3	0	0	1	0	-1	-1
x_4	0	0	0	0	1	-2	1
$Z_3^1 = 2$	$Z_j^1 - C_j^1$	0	0	0	0	2	1
$Z_3^2 = -4$	$Z_j^2 - C_j^2$	0	0	0	0	-4	3

Tableau 3

Donc le r-uplet $(2, -4)$ correspondant à x_3^1 est dominée par rapport au précédent de Eff_0 et Eff_1 .

Etape 3 :

$\Gamma_2 = \emptyset$. $x_3^1 = (1,0)$ est l'unique solution. Ce point entier est tronqué en utilisant la coupe $x_5 + x_6 \geq 1$.

En utilisant le dual du simplexe, on obtient une solution efficace $x_4^1 = (0,0)$ donnée par le tableau 4 optimal suivant :

x_B	B	x_1	x_2	x_3	x_4	x_5	x_6	x_7
x_1	0	1	2	0	0	0	0	1
x_6	0	0	1	0	0	0	1	0
x_3	4	0	0	1	0	0	0	0
x_4	2	0	-3	0	1	0	0	-2
x_5	1	0	-1	0	0	1	0	-1
$Z_4^1 = 0$	$z_j^1 - C_j^1$	0	1	0	0	0	0	2
$Z_4^2 = 0$	$z_j^2 - C_j^2$	0	-7	0	0	0	0	-4

Tableau 4

Le r-uplet $(0,0)$ correspondant à x_4^1 est non dominé par rapport aux précédents de Eff_0 et Eff_1 par conséquent, il est ajouté à Eff_0 et Eff_1 pour obtenir $Eff_2 = \{(6, -2); (3, -1); (0,0)\}$ avec : $I_2 = \{1,6,3,4,5\}$; $N_2 = \{2,7\}$.

Etape 4 :

$\Gamma_3 = \emptyset$. La solution $x_4^1 = (0,0)$ est l'unique solution. Le point est tronqué en utilisant la coupe $x_2 + x_7 \geq 1$. L'application du dual du simplexe provoque l'impossibilité de l'opération de pivot. Donc une infaisabilité indiquant ainsi que toutes les solutions efficaces du problème (P) ont été trouvées.

Cette infaisabilité est donnée par le tableau 5 suivant :

x_B	B	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	0	1	2	0	0	0	0	1	0
x_6	0	0	1	0	0	0	1	0	0
x_3	0	0	0	1	0	0	0	2	0
x_4	2	0	-3	0	1	0	0	-2	0
x_5	1	0	-1	0	0	1	0	-1	0
x_8	-1	0	-1	0	0	0	0	-1	1
$Z_5^1 = 0$	$z_j^1 - C_j^1$	0	1	0	0	0	0	2	0
$Z_5^2 = 0$	$z_j^2 - C_j^2$	0	-7	0	0	0	0	-4	0

Tableau 5

Le vecteur ligne est positif ou nul. Ceci implique que l'opération pivot est impossible. donc la convergence de la méthode de Abbas & Moulay nous donne toutes les solutions efficaces $\{(0,2); (0,1); (0,0)\}$.

Conclusion : la méthode de Abbas & Moulay nous donne toutes les solutions efficaces contrairement à celle de Gupta qui peut s'arrêter sans nous donner toutes les solutions efficaces.

CHAPITRE 4

Programmation multi-objectif linéaire fractionnaire en nombres entiers

4.1 Introduction

Le problème de la programmation multi objectif linéaire fractionnaire (*MOLFP*: Multiple Objective Linear fractional Programming program) est l'un des modèles les plus utilisés dans la prise de décision à objectifs multiples. Dans de nombreuses situation réelles modélisables par la programmation mathématique, les variables intervenant dans la modélisation sont soumises à être totalement en nombres entiers dans (*MOLFP*).

On parle alors de programmation multi-objectif linéaire fractionnaire en nombres entiers (*MOILFP*: Multiple Objective Integer Linear fractional Programming program). Une variété très grande de problèmes réels est formulée et résolue en utilisant (*MOILFP*).

Parmi les problèmes connus, on note les problèmes de voyageur de commerce, problème de sac à dos, problème de gestion, problème de planification de la production, problème d'optimisation de réseaux de télécommunication.

Dans ce chapitre on présente le problème fractionnaire linéaire uni critère et la résolution du problème de programmation multicritère linéaire fractionnaire en nombres entiers (*MOILFP*) basées sur le processus des coupes de Gomory et branch and bound.

4.2 Programmation fractionnaire linéaire uni-critère

Les programmes fractionnaires consistent à optimiser un objectif donné sous forme d'un quotient de deux fonctions soumises à un ensemble de contraintes.

le problème fractionnaire linéaire uni-critère se modélise comme suit :

$$\max \frac{M'x + M_0}{N'x + N_0}$$

Sous les contraintes :

$$\begin{cases} Ax \leq b \\ x \geq 0 \end{cases}$$

Où

$$M = \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_n \end{bmatrix}, N = \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_n \end{bmatrix}, X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

Où M_0, N_0 sont des scalaires réels, M et N sont des vecteurs de \mathbb{R}^n , A est une matrice d'ordre $(m \times n)$, b est un vecteur de \mathbb{R}^m .

Comme cela est prouvé, si une solution optimale pour un programme fractionnaire linéaire existe alors un point extrême optimal existe, en outre tout minimum local est un minimum global et tout maximum local est un maximum global, d'où une procédure qui se déplace d'un point extrême à un point extrême adjacent est une approche réalisable pour résoudre un tel problème.

Lemme :

Soit : $g(x) = \frac{M'x + M_0}{N'x + N_0}$

Et soit $S = \{x \in \mathbb{R}^n, Ax = b, x \geq 0\}$ un ensemble convexe tel que $N'X + N_0 \neq 0$ sur S alors, g est à la fois pseudo-convexe et pseudo-concave sur S .

4.2.1 Stratégie de résolution

La forme particulière des programmes fractionnaires a fait que de nombreux auteurs ont élaboré des méthodes de résolution spécifiques qui se sont avérées plus efficaces que l'application directe de méthodes générales de la programmation non linéaire.

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

- **La résolution directe :** le programme est traité sous sa forme originale. Elle a été utilisée pour les programmes hyperboliques en variables continues et entières.
- **La résolution par paramétrisation :** à l'inverse de la résolution directe, on construit un problème à objectif simplifié, combinaison linéaire du numérateur et du dénominateur par l'intermédiaire d'un paramètre, tout en gardant inchangé l'ensemble des contraintes. Une séquence de résolution de ce type de problème fournit une solution du programme fractionnaire. Cette méthode a été utilisée pour les différents programmes fractionnaires linéaires et non linéaires, en variables continues comme en variables discrètes, sur des domaines bornés.
- **La résolution d'un programme équivalent :** un changement de variables permet lui aussi de simplifier l'objectif, mais en transportant la difficulté sur l'ensemble des contraintes. Par exemple, un programme fractionnaire concave-convexe est transformé en un programme concave, et un programme hyperbolique en un programme à objectif linéaire, avec des contraintes additionnelles.

4.2.2 Représentation graphique de la programmation fractionnaire linéaire

Comme l'a noté Steuer[17], les programmes linéaires fractionnaires présentent un intérêt particulier mis en évidence par la linéarité des courbes niveaux de leurs fonctions critères. En effet, pour illustrer cet aspect, considérons une $\bar{\mathbb{Z}}$ -courbe niveau quelconque de la fonction critère :

$$\bar{\mathbb{Z}} = \frac{p'x + \alpha}{q'x + \beta}$$

Après simplification, nous obtenons :

$$\bar{\mathbb{Z}}(q'x + \beta) = p'x + \alpha$$

Ce qui entraîne :

$$\bar{\mathbb{Z}}\beta - \alpha = (p - \bar{\mathbb{Z}}q)'x$$

Qui est une expression linéaire de la $\bar{\mathbb{Z}}$ -courbe niveau de la fonction critère.

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

En dépit de la linéarité de la courbe niveau de la fonction objectif, les courbes niveaux ne sont pas parallèles (pour $p \neq 0$, $q \neq 0$ et $\neq \omega q, \forall \omega \in \mathbb{R}$).

Comme ils le sont en programmation linéaire. L'ensemble rotation est l'ensemble de tous les points d'intersection entre la 0-courbe niveau du numérateur et la 0-courbe niveau du dénominateur.

L'ensemble rotation est appelé **point de rotation** dans \mathbb{R}^2 et axe de rotation dans \mathbb{R}^3 . Les points de cet ensemble sont déterminés par la résolution du système suivant :

$$\begin{aligned} p'x &= -\alpha \\ q'x &= -\beta \end{aligned}$$

Exemple illustratif

Soit le programme fractionnaire linéaire suivant :

$$\text{maximiser } z = \frac{x_1 + x_2 - 1}{2x_1 + x_2 - 1}$$

$$\text{s. c } \begin{cases} 5x_1 + x_2 \geq 5 \\ x_1 \leq 3 \\ x_2 \leq 5 \\ x_1, x_2 \geq 0 \end{cases}$$

Dont le graphe est donné par la figure suivante :

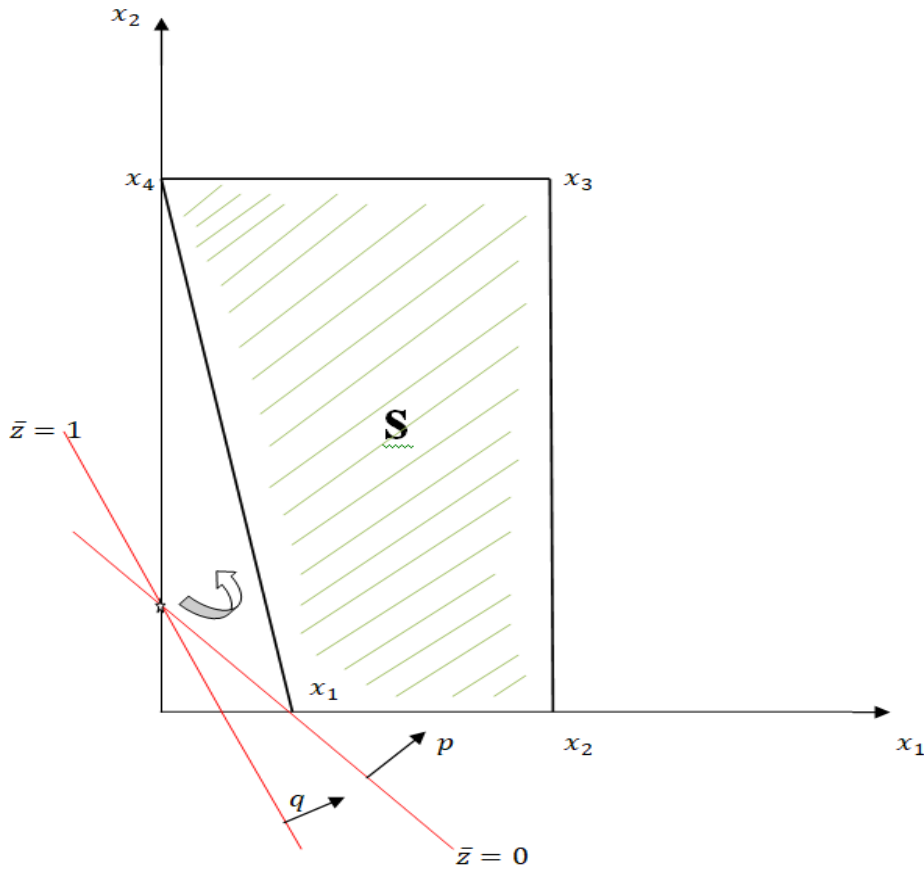


Figure1

Le problème a quatre extrémités x^1, x^2, x^3 et x^4 dont les valeurs du critère sont comme suit :

$$\begin{aligned} x^1 &= (1,0) & \mathbb{Z}(x^1) &= 0 \\ x^2 &= (3,0) & \mathbb{Z}(x^2) &= 2/5 \\ x^3 &= (3,5) & \mathbb{Z}(x^3) &= 7/10 \\ x^4 &= (0,5) & \mathbb{Z}(x^4) &= 1 \\ r &= (0,1) \end{aligned}$$

Alors la courbe de niveau $\bar{\mathbb{Z}}$ est l'ensemble des points $x = (x_1, x_2)$ satisfaisant l'équation :
 $\bar{\mathbb{Z}}\beta - \alpha = (p - \bar{\mathbb{Z}}q)'x \Leftrightarrow (-\bar{\mathbb{Z}} + 1) = (1 - 2\bar{\mathbb{Z}})x_1 + (1 - \bar{\mathbb{Z}})x_2$

Donc pour :

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

$$\bar{\mathbb{Z}} = 1 \Leftrightarrow -x_1 = 0 : \text{courbe de niveau } (1).$$

Les lignes vertes représentent les 0-courbes niveaux du numérateur et du dénominateur dont l'intersection est le point de rotation $r = (0,1)$.

La flèche circulaire dénote le gradient de la fonction fractionnaire linéaire critère et indique le sens et l'angle avec lesquels se déplacent les courbes de niveaux, donc en faisant déplacer la

courbe de niveau 1 autour du point de rotation suivant le sens de rotation trigonométrique, nous pouvons voir que le point optimal est x^4 de valeur optimale $\mathbb{Z}^* = 1$.

4.2.3 Méthode de A. Cambini et al. [20]

On considère le programme fractionnaire linéaire continu (P) :

$$(P) \begin{cases} \max z = \frac{c'x + \alpha}{d'x + \beta} \\ x \in S \end{cases}$$

Où $S = \{x \in \mathbb{R}^n, Ax = b, x \geq 0\}$ 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 4.1

On dira que x^* est solution optimale niveau pour le problème (P) si et seulement si x^* est solution optimale du problème (P_θ) pour certaine valeur de θ :

$$(P_\theta) \begin{cases} \max c'x + \alpha \\ x \in S \\ d'x + \beta = \theta \end{cases}$$

L'algorithme de Cambini et al.[20] génère une séquence finie $x^k, k = \overline{1, l}$ de solutions optimales niveau dont la première est trouvé de la façon suivante:

Résoudre le programme linéaire $(P_0) \begin{cases} \min d'x + \beta \\ x \in S \end{cases}$

Soit x^0 est unique, alors elle est une solution niveau, sinon résoudre le programme linéaire

$$(P_1) \begin{cases} \max c'x + \alpha \\ x \in S \end{cases}$$

Si (P_1) n'admet pas de solution, 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 4.1 [20]

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

Algorithme

Étape 1 :

Trouvé la solution optimale niveau x^1 .

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

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

Étape2 :

Si $J = \{j \in N / \hat{d}_j > 0\} = \emptyset$, terminer x^k est une solution optimale du problème (P) .

Sinon soit w tel que $\frac{\hat{c}_w}{\hat{d}_w} = \max \left(\frac{\hat{c}_j}{\hat{d}_j} \right)$.

Si $\hat{\gamma}_j > 0$, aller en 3.

Sinon, terminer, x^k est une solution optimale de (P) .

Etape3 :

La variable hors base x_w entre dans la base au moyen d'une opération pivot, poser $k = k + 1$ et aller à l'étape 2.

si une telle opération n'est pas possible, terminer : $\sup \left\{ \frac{cx + \alpha}{dx + \beta} / x \in S \right\} = \frac{\hat{c}_w}{\hat{d}_w}$.

4.3 Programmation multi-objectif fractionnaire linéaire en nombre entiers

4.3.1 Formulation du problème

Considérons, le problème d'optimisation fractionnaire linéaire multi-objectif en nombres entiers (MOILFP) donné sous la forme mathématique suivante :

$$(P) \begin{cases} \max Z_1(x) = \frac{c^1 x + \alpha^1}{d^1 x + \beta^1} \\ \vdots \\ \max Z_r(x) = \frac{c^r x + \alpha^r}{d^r x + \beta^r} \\ \text{s. c} \\ x \in S \\ x \text{ entier} \end{cases}$$

Où : $k = \{1, \dots, r\}$ représente le nombre d'objectifs ($r \geq 2$).

$S = \{x \in \mathbb{R}^n / Ax \leq b, x \geq 0\}$ est l'ensemble des solutions réalisables.

A est une matrice $m \times n$, $b \in \mathbb{R}^m$ et c^i et d^i sont des $1 \times n$ vecteurs, α^i et β^i sont des scalaires pour tout $i \in \{1, \dots, r\}$. $d^i x + \beta^i > 0$; Pour tout $i \in \{1, \dots, r\}$.

Le but quant à traiter ce type de problème (MOILFP) est de déterminer toutes les solutions efficaces. Plusieurs approches utilisent le concept basé sur la résolution d'une séquence finie de programmes uni-critères fractionnaires linéaires en nombres entiers.

Considérons donc le programme fractionnaire linéaire uni-critère en nombres entiers donné sous la forme suivante :

$$(P_1) \begin{cases} \max Z_1(x) = \frac{c^1 x + \alpha^1}{d^1 x + \beta^1} \\ x \in S, x \text{ entier} \end{cases}$$

Notons qu'au lieu de Z_1 , on peut de la même manière considérer le problème (P_l) avec un autre objectif Z_l , $l \in \{1, \dots, r\}$.

4.3.2 Méthodes de résolution d'un problème d'optimisation fractionnaire linéaire multicritère en nombres entiers

❖ La méthode de MOULAÏ Mustapha [1]

Considérons le problème fractionnaire linéaire en nombres entiers à objectifs multiples donné par :

$$(P) \left\{ \begin{array}{l} \max Z_1(x) = \frac{c^1 x + \alpha^1}{d^1 x + \beta^1} \\ \quad \cdot \\ \quad \cdot \\ \quad \cdot \\ \max Z_r(x) = \frac{c^r x + \alpha^r}{d^r x + \beta^r} \\ \text{s.c} \\ x \in S \\ x \text{ entier} \end{array} \right.$$

Afin de résoudre (P), on cherche l'ensemble des solutions réalisables de (P_1) suivant :

$$(P_1) \left\{ \begin{array}{l} \max Z_1(x) = \frac{c^1 x + \alpha^1}{d^1 x + \beta^1} \\ \text{s.c} \\ x \in S \\ x \text{ entier} \end{array} \right.$$

Les notations suivantes sont adoptées :

$S_1 = \{x \in \mathbb{R}^{n_1} / A_1 x \leq b_1, A_1 \in \mathbb{R}^{m_1 \times n_1}, b_1 \in \mathbb{R}^{m_1}, x \geq 0\}$ est la région tronquée de S obtenue par des coupes successives de Gomory.

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 entière correspondante à Z_1 obtenue sur S_1 .

Z_i^1 est la valeur de $Z_i, i \in \{2 \dots r\}$ correspondant à la solution x_1^1 .

β_1^1 est la base de S_1 .

$\alpha_{1,j}^1 \in \mathbb{R}^{m_1 \times 1}$ sont les vecteurs d'activités de $x_{1,j}^1$ appropriés à la région tronquée S_1 .

$y_{1,j}^1 = (y_{1,ij}^1) = (\beta_1^1)^{-1} \alpha_{1,j}^1$ où $y_{1,j}^1 \in \mathbb{R}^{m_1 \times 1}$.

$I_1 = \{j: \alpha_{1,j}^1 \in \beta_1^1\}$.

$N_1 = \{j: \alpha_{1,j}^1 \notin \beta_1^1\}$.

c_j^1 = la $j^{\text{ème}}$ composante de vecteur c^1 .

d_j^1 = la $j^{ème}$ composante de vecteur d^1 .

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

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

$$Z_1(x_1^1) = \frac{z_{1,1}^1}{z_{1,2}^1} \quad \text{où} \quad z_{1,1}^1 = c^1 x_1^1 + \alpha^1 \quad \text{et} \quad z_{1,2}^1 = d^1 x_1^1 + \beta^1$$

$y_{1,j}^1 = z_{1,2}^1(c_j^1 - c_{ij}^1) - z_{1,1}^1(d_j^1 - d_{ij}^1)$ le coût réduit relatif à la $j^{ème}$ composante du vecteur gradient réduit \overline{y}_1^1 .

$$\Gamma_1 = \{j/j \in N_1 \text{ et } \overline{y}_{1,j}^1 = 0\}$$

Pour $k \geq 2$

$S_k = \{x \in \mathbb{R}^{n_k} / A_k x \leq b_k, A_k \in \mathbb{R}^{m_k \times n_k}, b_k \in \mathbb{R}^{m_k}, x \geq 0\}$ est la région tronquée courante de S obtenue par l'application de la coupe $\sum_{j \in N_{k-1} - \{j_{k-1}\}} x_j \geq 1$ où $j_{k-1} \in \Gamma_{k-1}$

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

β_k^1 est la base S_k .

$\alpha_{k,j}^1 \in \mathbb{R}^{m_k \times 1}$ sont les vecteurs activités de $x_{k,j}^1$ appropriés à la région courant S_{k-1} .

$$y_{k,j}^1 = (y_{k,ij}^1) = (\beta_k^1)^{-1} \alpha_{k,j}^1 \quad \text{où} \quad y_{k,j}^1 \in \mathbb{R}^{m_k \times 1}$$

$$I_k = \{j/\alpha_{k,j}^1 \in \beta_k^1\}.$$

$$N_k = \{j/\alpha_{k,j}^1 \notin \beta_k^1\}.$$

c_j^1 est la $j^{ème}$ composante de vecteur p^1 .

d_j^1 est la $j^{ème}$ composante de vecteur q^1 .

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

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

$$Z_1(x_k^1) = \frac{z_{k,1}^1}{z_{k,2}^1} \quad \text{où} \quad z_{k,1}^1 = c^1 x_k^1 + \alpha^1 \quad \text{et} \quad z_{k,2}^1 = d^1 x_k^1 + \beta^1$$

$\overline{y}_{k,j}^1 = z_{k,2}^1(c_j^1 - c_{ij}^1) - z_{k,1}^1(d_j^1 - d_{ij}^1)$ le coût réduit relatif à la $j^{ème}$ composante du vecteur gradient réduit \overline{y}_k^1 .

$$\Gamma_k = \{j/j \in N_k \text{ et } \overline{y_{k,j}^1} = 0\}.$$

Théorème 3 : Le point x_k^1 de S est une solution optimale du problème fractionnaire (P_1)

Si et seulement si le vecteur gradient réduit $\overline{y_{k,j}^1} \leq 0$ pour tout indice $j \in N_k$.

Corollaire 2 : La solution x_k^1 du problème (P_1) est unique si et seulement si le vecteur gradient réduit $\overline{y_{k,j}^1} < 0$ pour tout indice $j \in N_k$.

Théorème 4 : Toutes les solutions réalisables $x_k^q, q \in \{2, \dots, p_k\}$ du problème fractionnaire linéaire (P_1) alternatives à x_k^1 sur l'arête E_{jk} de la région S (ou la région tronquée S_1) dans la direction d'un vecteur $\alpha_{k,jk}^1, j_k \in \Gamma_k$, existent dans le demi-espace ouvert $\sum_{j \in N_k - \{j_k\}} x_j < 1$

Théorème 5 : Une solution réalisable entière du problème fractionnaire linéaire (P_1) n'appartient pas à l'arête $E_{jk}, j_k \in \Gamma_k$ de la région S_k à travers une solution réalisable entière x_k^1 du problème (P_1) appartient au demi-espace fermé $\sum_{j \in N_k - \{j_k\}} x_j \geq 1$.

Algorithme

Etape 1 :

- Résoudre le problème (P_1) par la méthode de Cambini.
- Donner la solution optimale entière x_1^1 sur S_1 , notons qu'à la place du problème (P_1) , on peut considérer d'une manière analogue le problème (P_i) avec un objectif Z_i pour un indice quelconque $i \in \{2, \dots, r\}$, construire l'ensemble Γ_1 .

Etape 2 :

Si $\overline{y_{1,j}^1} < 0$ 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 données par $x_1^1, i \in \{2, \dots, r\}$.

Enregistrer le premier vecteur efficace comme $(Z_1^1, Z_2^1, \dots, Z_r^1)$ pour construire l'ensemble Eff_0 . Si $\Gamma_1 = \emptyset$, l'arête E_{j_1} ne contient pas de solutions réalisables entières alternatives à x_1^1 .

Tronquer le point x_1^1 par la coupe suivante : $\sum_{j \in N_k} x_j > 1$.

Par application de la méthode duale du simplexe, on obtient une solution réalisable entière $x_2^1 = (x_{2,j}^1)$ dans la région tronquée S_2 . Ajouter la valeur correspondante (Z_1, Z_2, \dots, Z_r) à l'ensemble des solutions efficaces Eff_0 s'il n'est pas dominé par l'un des précédents vecteurs entiers efficaces. Par conséquent Eff_0 devient Eff_1 .

S'il existe au moins un indice $j_1 \in N_1$ pour lequel $\overline{y_{1,j}^1} = 0$, calculer le nombre

$$\theta_{j_1} = \min_{i \in I_1} \left\{ \frac{x_{1,i}^1}{y_{1,i,j_1}^1} / y_{1,i,j_1}^1 > 0 \right\} \text{ correspondant à la solution } x_1^1.$$

- Si $\theta_{j_1} \geq 1$, déterminer toutes les solutions entières réalisables $x_1^q, q \in \{2, \dots, p_1\}$ alternatives à x_1^1 le long de l'arête E_{j_1} et leurs vecteurs critères correspondants. Tronquer l'arête E_{j_1} par la coupe $\sum_{j \in N_1 - \{j_1\}} x_j > 1$. Une solution réalisable entière

$x_2^1 = (x_{2,j}^1)$ dans la région tronquée S_2 est obtenue et son vecteur critère correspondant est ajouté à l'ensemble Eff_0 s'il n'est pas dominé par l'un des vecteurs critères efficaces précédemment déterminés pour obtenir Eff_1 .

- b) Si $\theta_{j_1} < 1$ pour tout indice $j_1 \in \Gamma_1$, choisir un indice quelconque j_1 et appliquer directement la coupe $\sum_{j \in N_1 - \{j_1\}} x_j \geq 1$ pour déterminer une solution réalisable entière $x_2^1 = (x_{2,j}^1)$ dans la région tronquée S_2 . Son vecteur critère correspondant est ajouté à l'ensemble Eff_0 s'il n'est pas dominé par l'un des vecteurs précédemment déterminés. Eff_0 devient Eff_1 .

Etape k :

Choisir un indice $j_{k-1} \in \Gamma_{k-1}$ et explorer l'arête $E_{j_{k-1}}$ pour déterminer les solutions entières réalisables x_{k-1}^q , $q \in \{2, \dots, p_k\}$ alternatives à x_{k-1}^1 . Augmenter l'ensemble Eff_{k-1} par les vecteurs critères non dominés correspondants pour construire Eff_k , l'arête $E_{j_{k-1}}$ est tronquée par la coupe $\sum_{j \in N_{k-1} - \{j_{k-1}\}} x_j \geq 1$.

La solution entière obtenue sur la région tronquée S_k sera soit une solution réalisable pour le problème (P_1) alternative à x_{k-1}^1 ou la prochaine meilleure solution x_k^1 ou un point non entier. Ceci marque le début de l'étape $k + 1$.

Etape finale :

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

Exemple d'application

Soit le problème de la programmation fractionnaire linéaire en nombres entiers à trois critères suivant :

$$(P) \left\{ \begin{array}{l} \max Z_1(x) = \frac{4x_1 - 2x_2 - 23}{2x_1 + 3x_2 + 3} \\ \max Z_2(x) = \frac{-3x_1 + 3}{2x_2 + 2} \\ \max Z_3(x) = 2x_1 + x_2 \\ \text{s.c} \\ 4x_1 + x_2 \leq 2 \\ x_1 - 3x_2 \leq 6 \\ x_1 \geq 0 ; x_2 \geq 0 \\ x_1 \text{ et } x_2 \text{ entiers} \end{array} \right.$$

Etape 1 :

Résoudre le problème relaxé (*) :

$$(*) \left\{ \begin{array}{l} \max Z_1(x) = \frac{4x_1 - 2x_2 - 23}{2x_1 + 3x_2 + 3} \\ \text{s.c} \\ 4x_1 + x_2 \leq 2 \\ x_1 - 3x_2 \leq 6 \\ x_1, x_2 \geq 0 \end{array} \right.$$

La solution de base réalisable optimale du domaine tronqué est donnée par le tableau 4.1 :

$$I_1 = \{2; 4\}, N_1 = \{1; 3\}, \Gamma_1 = \{j/j \in N_1, \bar{y}_j = 0\} = \emptyset$$

$$Z_1^1 = Z_1(x_1^1) = -3$$

$$Z_2^1 = Z_2(x_1^1) = 1/2$$

$$Z_3^1 = Z_3(x_1^1) = 2.$$

L'unique solution optimale $x_1^1 = (0; 2)$ produit le premier triplet efficace $\left(-3; \frac{1}{2}; 2\right)$ par suite,
 $Eff_0 = \left\{\left(-3; \frac{1}{2}; 2\right)\right\}.$

B	x_B	x_1	x_2	x_3	x_4
x_2	2	4	1	1	0
x_4	12	13	0	3	1
p_j	-27	-12	0	-2	0
q_j	9	10	0	3	0
\bar{y}_j	-3	-162	0	-53	0

Tableau 4.1

Etape 2 :

$\Gamma_1 = \emptyset$, donc on tronque le point par la coupe : $x_1 + x_3 \geq 1$ i.e $-x_1 - x_3 + x_5 = -1$ où x_5 est une variable d'écart supplémentaire.

B	x_B	x_1	x_2	x_3	x_4	x_5
x_2	1	3	1	0	0	1
x_4	9	10	0	0	1	3
x_3	1	1	0	1	0	-1
p_j	-25	-10	0	0	0	-2
q_j	6	7	0	0	0	3
\bar{y}_j	-25/6	-115	0	0	0	-63

Tableau 4.2

Comme $\Gamma_2 = \emptyset$, $x_2^1 = (0; 1)$ est une solution unique.

x_2^1 produit le r-uplet efficace $\left(-\frac{25}{6}; \frac{3}{4}; 1\right)$ qui n'est pas dominé par le précédent donc on augmente $Eff_1 = \left\{\left(-3; \frac{1}{2}; 2\right), \left(-\frac{25}{6}; \frac{3}{4}; 1\right)\right\}.$

Etape 3 :

Comme $\Gamma_2 = \emptyset$, $x_2^1 = (0; 1)$ est une solution unique. Ce point entier est tronqué par la coupe $x_1 + x_5 \geq 1$ i.e $-x_1 - x_5 + x_6 = -1$ où x_6 est une variable d'écart supplémentaire

B	x_B	x_1	x_2	x_3	x_4	x_5	x_6
x_2	0	2	1	0	0	0	1
x_4	6	7	0	0	1	0	3
x_3	2	2	0	1	0	0	-1
x_5	1	1	0	0	0	1	-1
p_j	-23	-8	0	0	0	0	-2
q_j	3	4	0	0	0	0	3
\bar{y}_j	-23/3	-68	0	0	0	0	-63

Tableau 4.3

$\Gamma_3 = \emptyset$, $x_3^1 = (0; 0)$ est une solution unique.

$x_3^1 = (0; 0)$ produit le r-uplet efficace $\left(-\frac{23}{3}; \frac{3}{2}; 0\right)$ qui n'est pas dominé par les précédents donc on aura $Eff_2 = Eff_1 \cup \left\{\left(-\frac{23}{3}; \frac{3}{2}; 0\right)\right\}$.

Etape 4 :

Comme $\Gamma_3 = \emptyset$, x_3^1 est une solution unique. Ce point entier est tronqué par la coupe

$x_1 + x_6 \geq 1$ i.e $-x_1 - x_6 + x_7 = -1$ où x_7 est une variable d'écart supplémentaire.

Lors de l'application de la méthode duale du simplexe, l'opération usuelle de pivot devient impossible indiquant que le domaine restant ne contient aucun point entier réalisable et que toutes les solutions efficaces du problème étudié ont été obtenues.

Cette impossibilité de pivoter est donnée par le tableau du simplexe suivant :

B	x_B	x_1	x_2	x_3	x_4	x_5	x_6	x_7
x_2	-1	1	1	0	0	0	0	1
x_4	3	4	0	0	1	0	0	3
x_3	3	3	0	1	0	0	0	-1
x_5	2	2	0	0	0	1	0	-1
x_6	1	1	0	0	0	0	1	-1
p_j	-21	-6	0	0	0	0	0	-2
q_j	0	1	0	0	0	0	0	3
\bar{y}_j	/	-21	0	0	0	0	0	-63

Tableau 4.4

La variable x_2 est sélectionnée pour sortir de la base.

Et x_7 est la variable qui doit rentrer dans la base. Seulement, le vecteur ligne toutes ses composantes positives ce qui implique l'impossibilité de l'opération de pivot et par suite l'arrêt de l'algorithme.

CHAPITRE 5

5 Logiciel (LINGO / MATLAB) et Applications

• Logiciel LINGO

A) Introduction

La programmation informatique est un ensemble d'outils et de techniques permettant de résoudre des problèmes mathématiques par ordinateurs, elle sert à trouver une solution optimale de n'importe quel type de problème par exemple.

Le processus de résolution d'un problème mathématique exige un grand nombre de calculs donc il est mieux de l'exécuter par machine.

Pour cela on a choisi le logiciel LINGO qui est spécifique pour résoudre des problèmes d'optimisations (linéaire, non linéaire, convexe, non convexe,...etc).

Il comporte un éventail de commandes qui peuvent être appelées à tout moment.

B) Présentation du logiciel LINGO

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

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

_ Solveur direct.

_ Solveur indirect.

_ Solveur linéaire.

_ Solveur non linéaire.

-Méthode de type séparation et évaluation. De plus, LINGO est :

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

- Possibilités pour résoudre les problèmes plus rapidement.

Un moyen amélioré pour résoudre beaucoup de types de problèmes.

Un moyen de décomposition si un modèle contient des sous-modèles.

LINGO est livré avec un jeu de solveurs pour l'optimisation linéaire, non-linéaire convexe ou non convexe), simplicial, sous contraintes, et en nombre entier.

Nous n'avons même pas à nous préoccuper du choix du solveur : en effet, LINGO réinterprète lui-même nos formulations et sélectionne automatiquement le solveur adapté à chaque problème. Un modèle d'optimisation se compose de trois parties :

1. La fonction objective : c'est la formule simple qui décrit exactement ce que le modèle devrait optimiser.

2. Les variables : ce sont les quantités qui peuvent être changées pour déterminer la valeur optimale de la fonction objective.

3. Les contraintes : ce sont les formules qui définissent les limites sur les valeurs des variables.

Les fonctions utilisées dans un modèle de LINGO sont :

@FOR - utilisée pour produire des contraintes.

@ SUM- calcul de la somme.

@MAX - recherche de maximum.

@MIN - recherche de minimum.

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

GIN- toute valeur positive de nombre entiers.

BIN - une valeur binaire (0 ou 1).

FREE - toute valeur positive ou négative réelle.

BND - toute valeur bornée par des limites indiquées.

La forme générale pour la déclaration d'une variable x, en utilisant les fonctions variables

GIN, BIN, FREE est : fonction (x) ; La fonction BND inclut les limites inférieures et supérieure des variables, sa forme générale est :

BND (limite inférieure, x, limite supérieure).

C) Interface de logiciel

Lors du lancement de LINGO après l'installation, certaines versions nous demandent d'entrer la clé de la licence, donc on voit apparaître la fenêtre suivante :

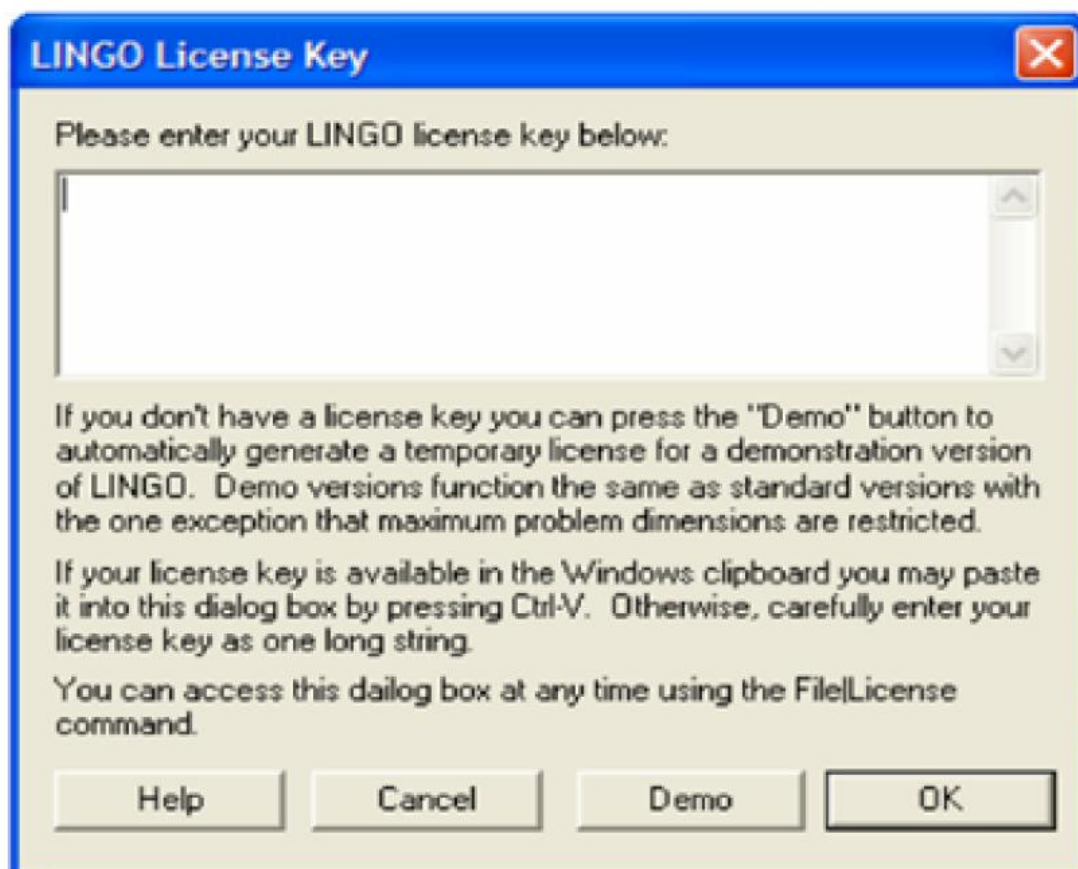


Figure 5.1.1 : Licence du logiciel

Donc on fait entrer la clé et on clique sur _ OK _, si on n'a pas la clé de la licence on peut utiliser LINGO sous le mode « Demo », mais la licence de ce mode expire dans 30 jours.

Juste après la barre des menus LINGO, une fenêtre pour saisir le modèle de problème à résoudre apparaît :

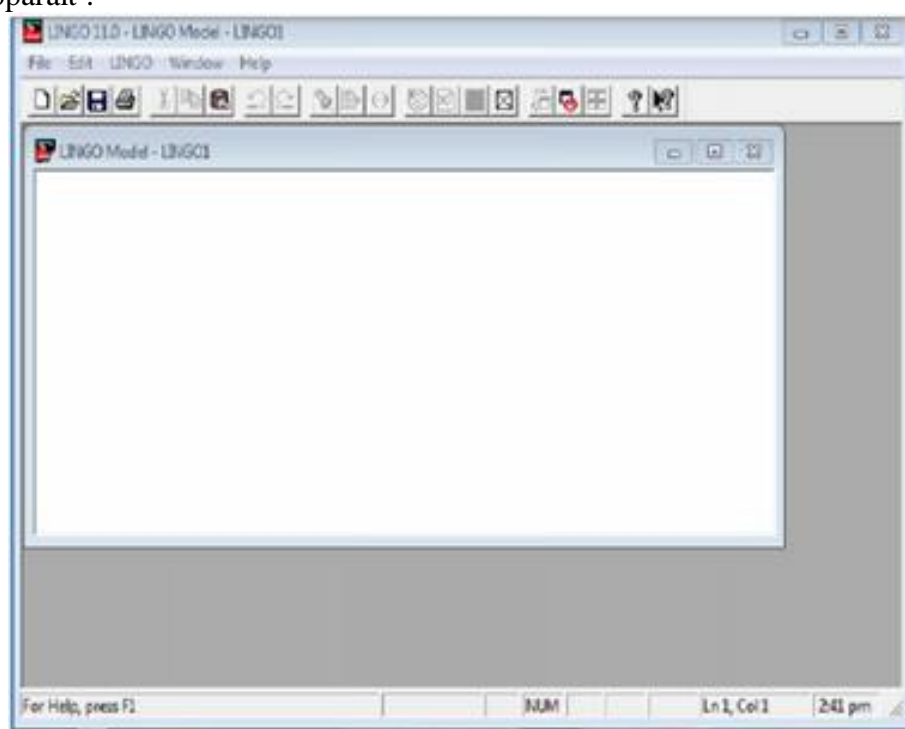
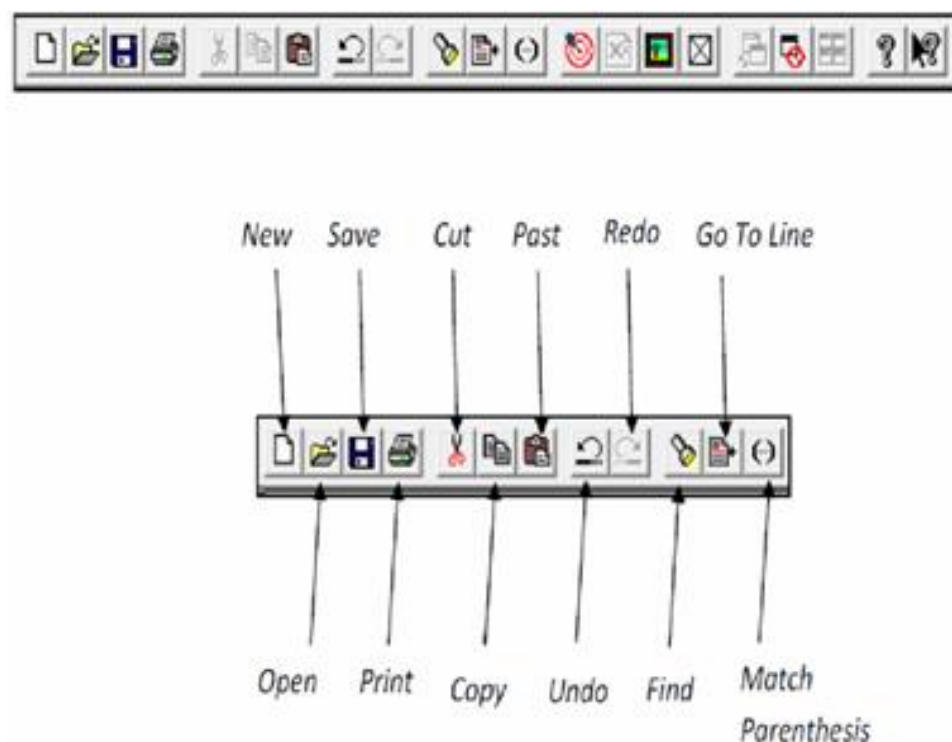


Figure 5.1.2 : La barre des menus du logiciel.

La barre des outils de LINGO est la suivante :



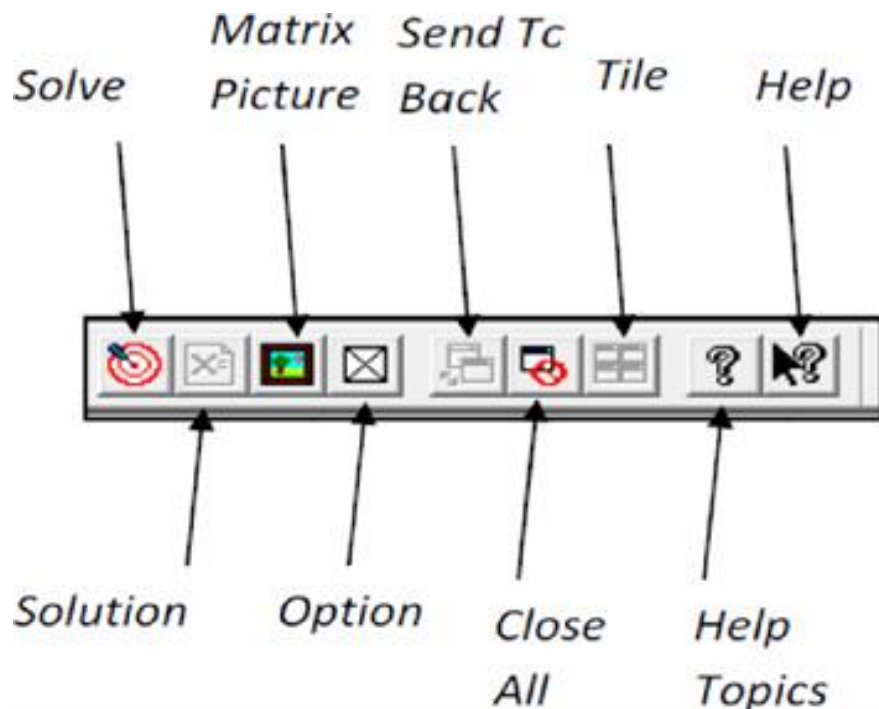


Figure 5.1.3 : La barre des outils de LINGO

Signification :

- New** : ouvrir une nouvelle fenêtre.
- Open** : ouvrir un dossier enregistré.
- Save** : enregistrer un modèle.
- Print** : imprimer la fenêtre courante.
- Cut** : copier et déplacer le texte sélectionné.
- Copy** : copier le texte sélectionné.
- Past** : coller le contenu sélectionné dans le document.
- Undo** : refaire l'action précédente.
- Redo** : revenir à l'action suivante.
- Find** : chercher un document.
- Go To Line** : déplacer le curseur à un certain nombre de ligne.
- Match Parenthesis** : trouvez la parenthèse étroite qui correspond à la parenthèse ouverte que vous avez choisi.
- Solve** : résoudre le modèle.
- Solution** : faire la fenêtre de rapport de la solution du modèle courant.
- Option** : ensemble des options de système.
- Send To Back** : placer la fenêtre courante derrière les autres fenêtres ouvertes.
- Close All** : fermer toutes les fenêtres ouvertes.
- Tile** : placer les fenêtres ouvertes dans la même place dans la fenêtre de programme LINGO.
- Help Topics** : ouvrir le manuel de LINGO.
- Help** : Ouvrir l'aide de LINGO.

D) Exemple d'application

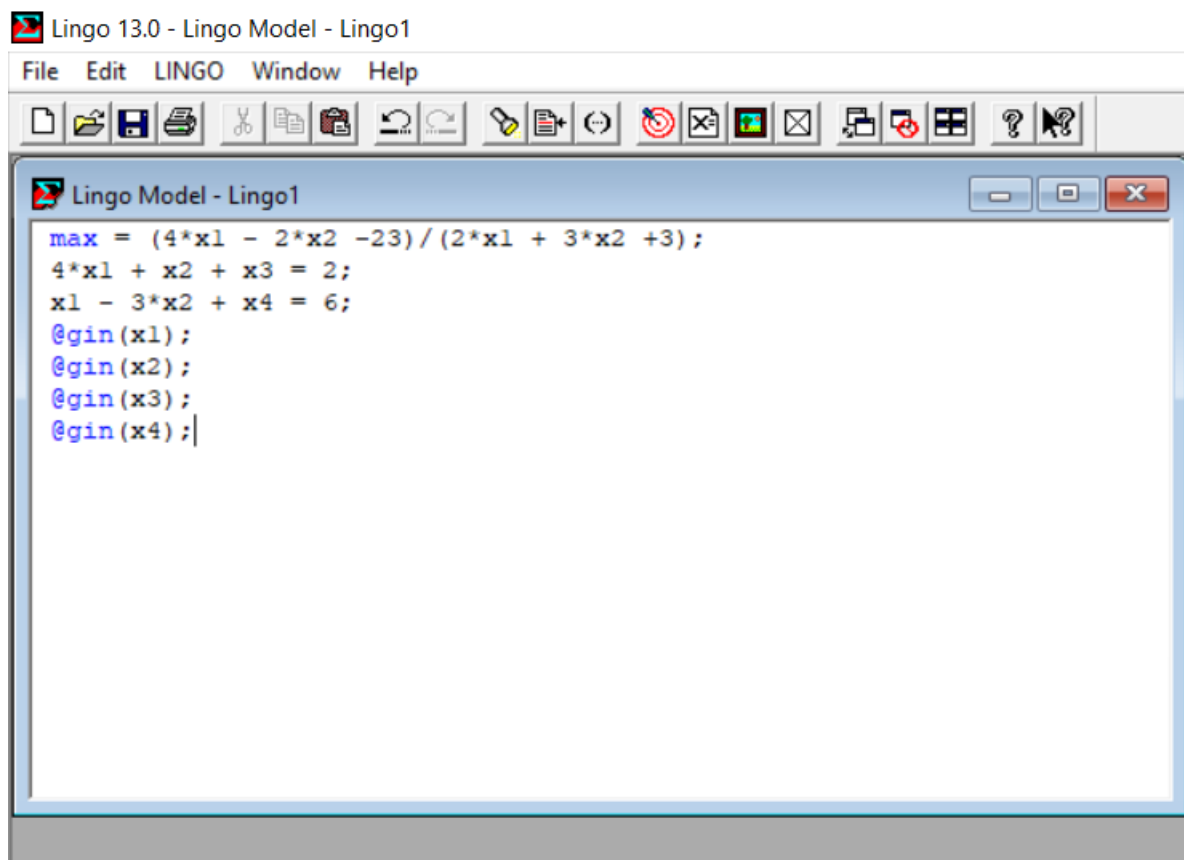
Soit le problème de la programmation fractionnaire linéaire en nombres entiers à trois critères suivant :

$$(P) \left\{ \begin{array}{l} \max Z_1(x) = \frac{4x_1 - 2x_2 - 23}{2x_1 + 3x_2 + 3} \\ \max Z_2(x) = \frac{-3x_1 + 3}{2x_2 + 2} \\ \max Z_3(x) = 2x_1 + x_2 \\ \text{s.c} \\ 4x_1 + x_2 \leq 2 \\ x_1 - 3x_2 \leq 6 \\ x_1 \geq 0 ; x_2 \geq 0 \\ x_1 \text{ et } x_2 \text{ entiers} \end{array} \right.$$

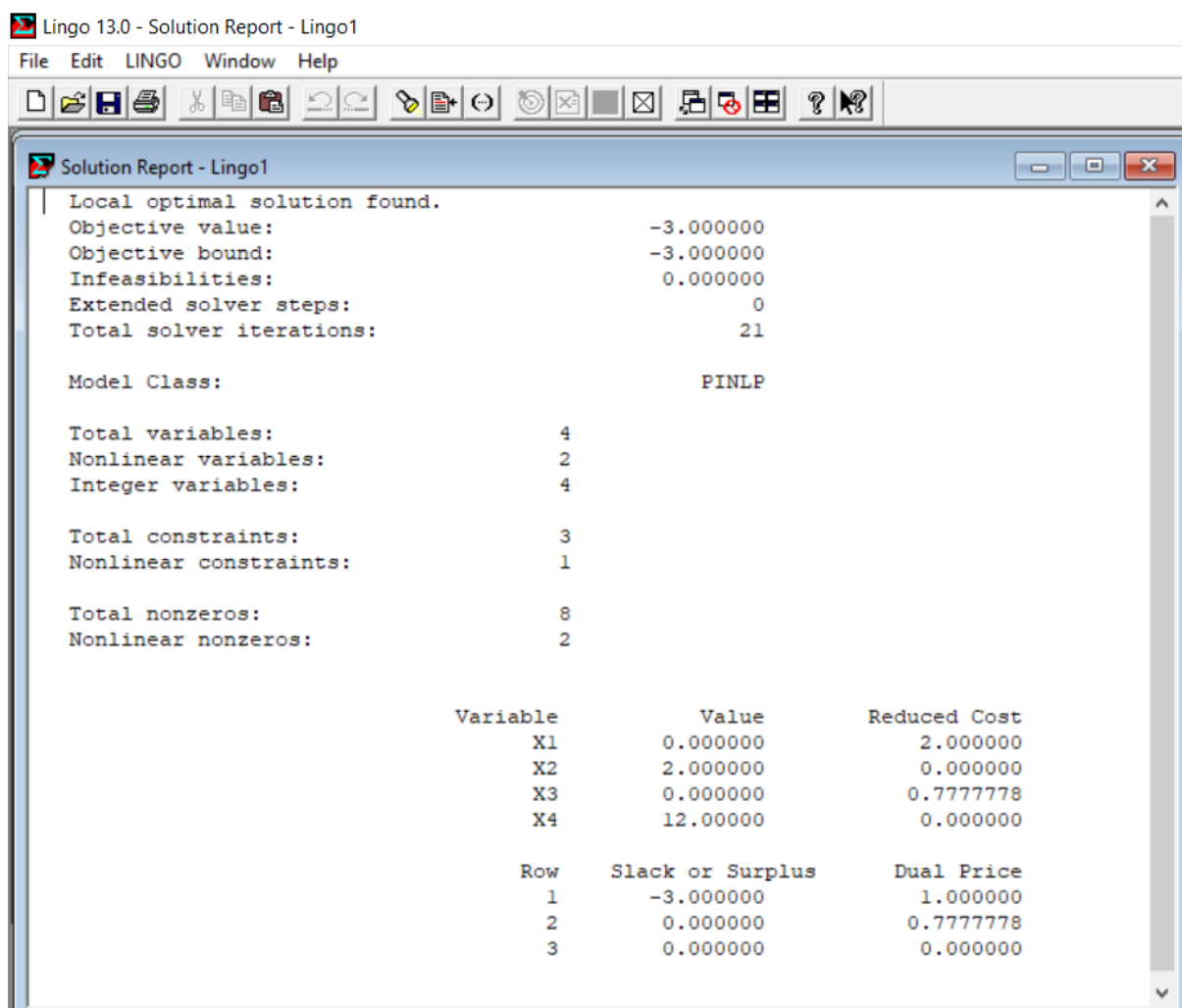
Etape 1 :

Résoudre le problème relaxé (*) :

$$(*) \left\{ \begin{array}{l} \max Z_1(x) = \frac{4x_1 - 2x_2 - 23}{2x_1 + 3x_2 + 3} \\ \text{s.c} \\ 4x_1 + x_2 \leq 2 \\ x_1 - 3x_2 \leq 6 \\ x_1, x_2 \geq 0 \end{array} \right.$$



On clique sur solve dans la barre des outils :



• Logiciel MATLAB

1. Qu'est ce que MATLAB ?

MATLAB est une application qui fournit un environnement de calcul matriciel simple, efficace, interactif permettant la mise en œuvre des algorithmes développés dans le cadre des projets *linpack* et *eispack*.

MATLAB est constitué d'un noyau relativement réduit, capable d'interpréter puis d'évaluer des expressions numériques matricielles qui lui sont adressées :

- ✓ Soit directement au clavier depuis une fenêtre de commande.
- ✓ Soit sous forme de séquences d'expressions ou **scripts** enregistrées dans des fichiers-texte appelés *m-files* et exécutées depuis la fenêtre de commandes.
- ✓ Soit plus rarement sous forme de fichiers binaires appelés *mex-files* ou fichiers *.mex* générés à partir d'un compilateur C ou Fortran.

Ce noyau est complété par une bibliothèque de fonctions prédéfinies, très souvent sous forme de fichiers *m-files*, et regroupés en paquetages ou *toolboxes*. A côté des *toolboxes* requises *local* et *matlab*, il est possible d'ajouter des *toolboxes* spécifiques à tel ou tel problème mathématique, *Optimization Toolbox*, *Signal Processing Toolbox* par exemple ou encore des

toolboxes créées par l'utilisateur lui même. Un système de chemin d'accès ou *path* permet de préciser la liste des répertoires dans lesquels MATLAB trouvera les différents fichiers *mfiles*.

MATLAB comporte un très grand nombre d'opérateurs, de commandes et de fonctions permettant de réaliser plusieurs opérations.

Dans MATLAB, on peut programmer et créer les fonctions et fichiers nous même sur M-file, puis les sauvegarder et les exécuter sur l'espace de travail. Il suffit juste d'introduire le nom du fichier ou de la fonction.

2. Implémentation de l'exemple

Pour notre implémentation nous avons pris l'exemple précédent.
Après avoir trouver la solution optimale avec LINGO $x_1^1 = (0,2)$, en utilisant MATLAB nous calculerons le premier tableau de **MOULAÏ Mustapha** de la manière suivante :

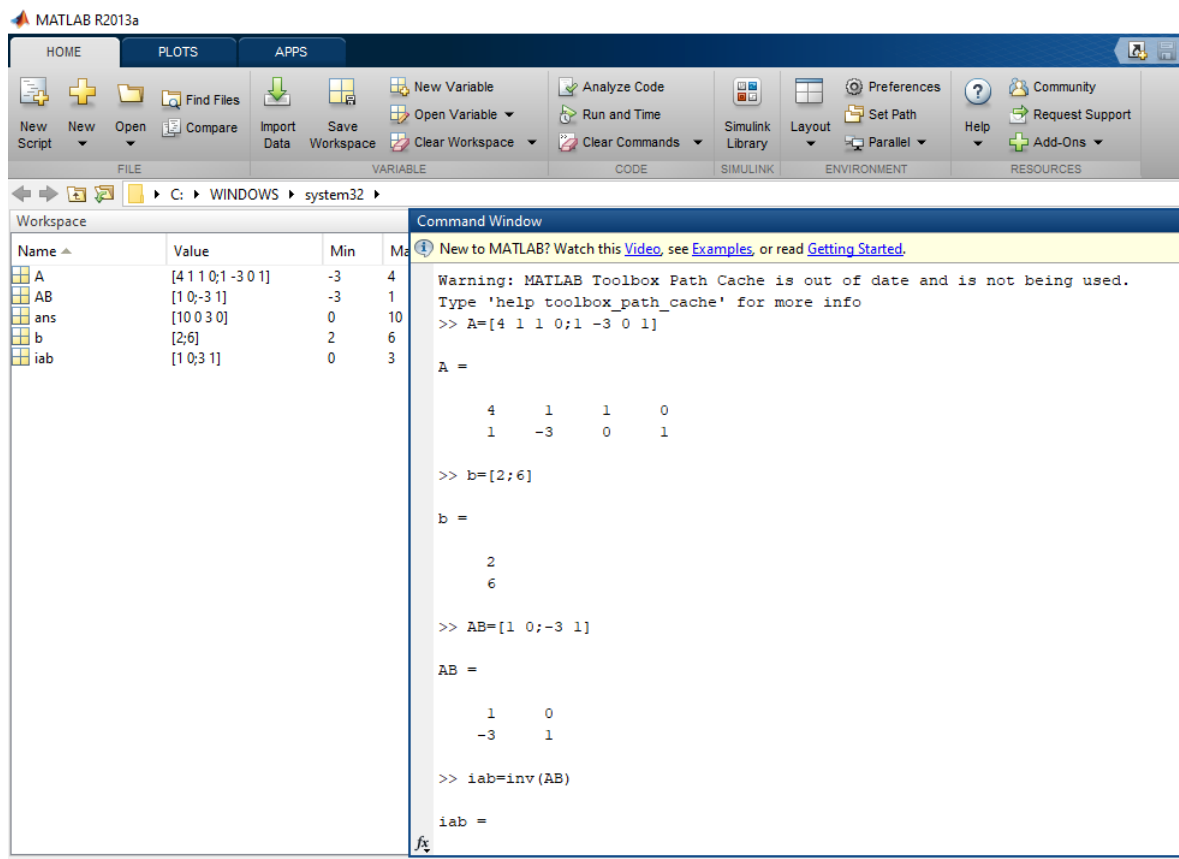


Figure 5.2.1

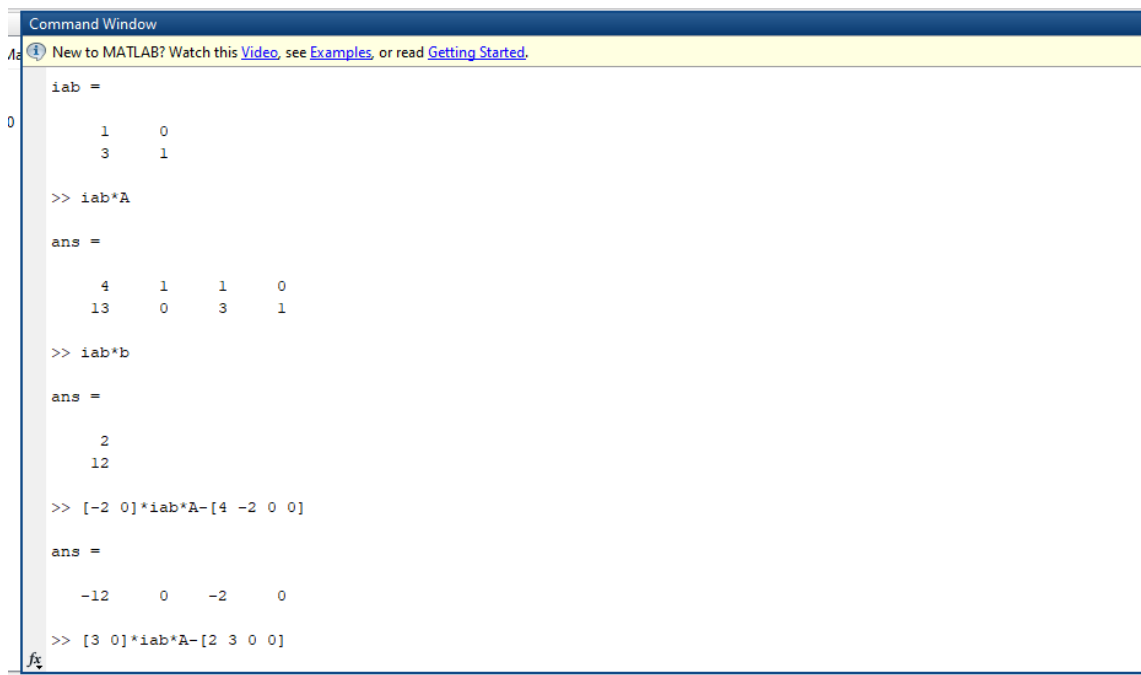


Figure 5.2.2

```

>> [3 0]*iab*A-[2 3 0 0]

ans =

    10     0     3     0

fx >> |

```

Figure 5.2.3

On aura le résultat suivant :

B	x_B	x_1	x_2	x_3	x_4
x_2	2	4	1	1	0
x_4	12	13	0	3	1
p_j	-27	-12	0	-2	0
q_j	9	10	0	3	0
\bar{y}_j	-3	-162	0	-53	0

Et de même manière on trouvera les trois autres tableaux.

Conclusion

L'optimisation multicritère est sans doute un axe de recherche très intéressant pour les scientifiques et les ingénieurs, non seulement à cause de la nature multicritère de la plupart des problèmes réels, mais aussi parce que de nombreuses questions restent ouvertes dans ce domaine.

Pour ce type de problèmes, il ne s'agit plus de rechercher une solution optimale, mais un ensemble de solutions Pareto optimales qui se distinguent par les différents compromis réalisés entre les différentes fonctions objectifs considérées.

Dans les trente dernières années, la plupart des travaux réalisés dans ce domaine ont porté sur la programmation linéaire multi-objectif. Les raisons principales de cet intérêt sont d'une part le développement de la programmation linéaire mono-objectif en recherche opérationnelle et la facilité relative de traiter de tels problèmes, et d'autre part l'abondance de cas pratiques pouvant être formulés sous cette forme.

Dans cette étude, nous sommes intéressés aux problèmes de programmation linéaire multi-objectifs.

Nous avons commencé ce travail par l'étude de la méthode du simplexe pour la programmation linéaire mono-objectif, ce qui a fait l'objet du premier chapitre de ce mémoire, et la méthode duale du simplexe, la méthode de branch and bound, la méthode des coupes...etc.

Le deuxième chapitre introduit les notions de base concernant l'optimisation multi-objectif comme la dominance, l'efficacité, les points particuliers. nous avons étudié trois méthodes de résolution de tels problèmes, la méthode du simplexe multicritère, la méthode d'agrégation et la méthode des contraintes et nous avons illustré chacune de ces méthodes par un exemple numérique détaillé.

Nous avons aussi donné une vue panoramique des méthodes d'optimisation fractionnaire linéaire discrète uni-critère et multicritère. On peut conclure que ces méthodes sont basées sur l'optimisation continue, vu l'utilisation successives des algorithmes de la méthode du simplexe, et la méthode duale du simplexe.

Les deux méthodes exactes concernant l'optimisation fractionnaire linéaire en nombres entiers à objectifs multiples représentent un résultat important dans la littérature compte tenu des propriétés spécifiques du problème (MOILFP). Bien que l'avantage principal de ces méthodes soit le fait qu'aucune optimisation non linéaire n'est exigée, elle présente néanmoins un inconvénient alourdissant la complexité de l'algorithme. Cet inconvénient est caractérisé par la recherche de toutes les solutions réalisables entières du problème de programmation fractionnaire linéaire entier relaxé (*Pr*).

Donc le problème reste ouvert aux autres chercheurs pour développer de meilleures méthodes de résolution de ce type de problèmes.

Bibliographie :

- [1] Abbas M. et Moulai M., Integer linear fractional programming with multiple objective, Journal of the Italian operations research society, Ricerca Operativa, vol 2; pp. 256- 265 (2002).
- [2] Aiden M.Oukacha B. « les manuels de l'étudiant Recherche Opérationnelle Programmation linéaire ».Copyright Eurl Pages Bleues internationales,Maison d'édition pour l'enseignement et la formation,2005.
- [3] Chergui M.E.A and Moulai M., An exact method for a discrete multiobjective linear fractional optimization, Hindawi Publishing Corporation, Journal of Applied Mathematics and Decision Sciences, pp. 1-12 (2008).
- [4] 1.7 La Méthode de Balas: C. Guéret C. Prins M. Sevaux Programmation linéaire,N° éditeur : 6186,Février 2003.
- [5] G.BAILLARGEON. .Programmation linéaire Appliquée., Les éditions SMG.
- [6] C.Guéret C.Prins M.sevaux Programmation linéaire ,N° éditeur :6186,Février 2003.
- [7] Chergui M.E.A and Moulai M., An exact method for a discrete multiobjective linear fractional optimization, Hindawi Publishing Corporation, Journal of Applied Mathematics and Decision Sciences, pp. 1-12 (2008).
- [8] Yann C., Siarry P., Optimisation multiobjectif, groupe Eyrolles, 2002.
- [9] : Klein D. and Hannan E., An algorithm for multiple objective integer linear programming problem, European journal of operational research 9, pp. 152- 159, (1982).
- [10] : Abbas M. et Moulai M., Solving multiple objective integer linear programming, Journal of the Italian operations research society (Ricerca operativa), Vol. 29 n° 89 pp.15-38 (1999).
- [11] Cambini A. and Martein L. *A modified version of Martos's algorithm for the linear fractional problem*, Methods of Oper.Res. 53, 33-44, 1986.
- [12] Yann C., Siarry P., Optimisation multiobjectif, groupe Eyrolles, (2002).
- [13] Gupta R. and Malhorta R., Multi Criteria integer linear programming problem, cahier de CERO 34 (1992).
- [14]Abbas M. et Moulai M., Solving multiple objective integer linear programming, Journal of the Italian operations research society (Ricerca operativa), Vol. 29 n° 89 pp.15-38 (1999).
- [15] : Kornbluth, J. S. H., and Steuer, R. E., Multiple objective linear fractional programming, Management Science 27 (1981), pp. 1024-1039.

- [16] : Nykowski I. and Zolkiewski Z., A compromise procedure for the multiple objective linear fractional programming problem, *European Journal of Operational Research* 19(1), pp. 91-97, (1985).
- [17] : Steuer R., *Multiple Criteria Optimization : Theory, Computation and Applications*, Wiley & Sons, New-York (1985).
- [18] Sylva J. and Crema A., (2004), A method for finding the set of nondominated vectors for multiple objective integer linear programs, *European Journal of Operational Research*, 158 (1), 46-55.
- [19] Pareto A., *Cours d'économie politique*, vol. 1 et 2, F. Rouge, Lausanne, (1896).
- [20] Bitran G.R., and Novaes G., Linear Programming with a fractional objective function *Operations Research* 24, pp. 675-699, (1976).
- [21] A. Berro, *Optimisation multiobjectif et stratégies d'évolution en environnement dynamique*, thèse doctorat en informatique, Toulouse I, (2001).
- [22] A.M. Geoffrion, Proper Efficiency and the theory of Vector Maximization, *Journal of Mathematical and Advances Algebra* 22 (1968), pp. 618-630.

Résumé

L'une des principales missions de la recherche opérationnelle est d'aider à prendre des décisions en vue d'une gestion efficace, rationnelle et logique.

L'optimisation multicritère est sans doute un axe de recherche très intéressant pour les scientifiques et les ingénieurs, non seulement à cause de nature multicritère de la plupart des problèmes réels, mais aussi parce que de nombreuses questions restent ouvertes dans ce domaine. Cette expérience nous a permis de trouver la solution Pareto optimale qui répond au mieux aux préférences du décideur. L'optimisation multi objectif est donc étroitement liée au domaine de la prise de décision multicritère.

Mots clés : PL, MOP, indice hors base, Gupta, entiers, coupe, fractionnaire.