

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université M'hamed Bougara Boumerdes
Faculté des Sciences
Département de Mathématiques



Mémoire Présenté
Pour L'Obtention Du Diplôme De Master
En Recherche Opérationnelle
Option : Recherche Opérationnelle Modélisation et Aide à la Décision
(ROMAD)

Par : CHELOUTI Sara
Et : KAIDI Karima

***Résolution d'un problème d'optimisation
multi-objectif fractionnaire Linéaire flou en
nombres entiers***

Soutenu à l'UMBB, le 19/06/2016, devant le jury composé de :

M ^r F.CHEURFA	M.A.A	Président	à l'UMBB -Boumerdes.
M ^{elle} S.ZOUAOUI	M.A.A	Promotrice	à l'UMBB - Boumerdes
M ^r L.BOURECHE	M.A.A	Examineur	à l'UMBB - Boumerdes.

Année Universitaire 2015 – 2016

Remerciements

Nous tenons à exprimer nos plus vifs remerciements à ALLAH pour nous avoir facilité ce travail..

Nous tenons, avant tout, à exprimer nous profonde gratitude à madame *ZOUAOUI Selma*, qui a assumé la direction de ce travail. Qu'elle veuille bien trouver ici l'expression de nous reconnaissance pour son dévouement, sa patience, sa disponibilité, ses conseils et son aide constant qu'il nous a apporté tout au long de ce travail.

Nous remercions les membres de jury qui ont accepté de juger ce travail et d'y apporter leur caution.

Nous adressons notre vifs remerciements à tous les enseignants qui, par leur enseignement, leur encouragement et leur aide, ont contribués à notre formation durant toutes notre études à l'université de Boumerdes.

A tous ceux qui ont participé de près ou de loin à l'accomplissement de ce travail.

Dédicace

Je remercie Allah pour m'avoir donné la force d'accomplir ce travail pour aller plus loin

Je dédie ce travail à mes parents, ma mère pour ses encouragements et ses prières tout au long de mes études, mon père pour tous ce qu'il avait fait pour avoir ce résultat.

Je le dédie à mes frères et sœurs surtout mon ange "youcef" (rabi yarahmou), et je les remercie pour leurs encouragements et leurs aides ainsi que toute ma famille.

A ma grand-mère que j'aimes .

A tous mes amies avec lesquelles j'ai partagé mes moments de Joie et de bonheur

karima

Dédicace

Je remercie Allah tout puissant clément et miséricordieux de m'avoir soigné et aidé pour accomplir ce travail .

Je dédie ce travail à la memoire de mon père(allah yerhmo),à ma mère pour toute son assistance et sa présence dans ma vie, ses encouragements et ses prières tout au long de mes études, .

À mes frères et sœurs, et je les remercie pour leurs encouragements et leurs aides ainsi que toute ma grande famille.

À l'homme qui m'a choisi pour me donner son nom à jamais.

*À tous mes amies qui m'ont partegé les bon mouments comme les mauvais parme eux amina , bouchra , karima , nedjma , hanane , nesrine , sarah , hafida , ahlam , meriem , hassiba ,
katia , feriel , imane ,.....*

À tous ceux qui aiment Sara et ceux qui Sara aime.

sara

Table des matières

<i>Introduction générale</i>	2
1 Programmation Linéaire	4
1 Programmation linéaire continue	4
1.1 Programme linéaire	4
1.2 Forme d'un Programme linéaire	4
1.3 La notion de dualité	5
1.4 Résultats fondamentaux et méthodes de résolution de la programmation linéaire	7
1.5 Méthodes de résolution	8
2 Programmation linéaire en nombres entiers	10
2.1 Forme générale d'un programme linéaire en nombres entiers	10
2.2 Méthodes de résolution	11
2 Optimisation Multi-Objectif	16
1 Définition :	16
2 La dominance	17
2.1 Introduction et définitions	18
2.2 Optimalité de Pareto	18
3 Efficacité (ou Pareto optimale)	19
3.1 Efficacité faible	19
3.2 Efficacité forte	19
4 Front Pareto et surface de compromis	19
4.1 Ensemble des solutions non dominées :	20
4.2 Front Pareto	20
4.3 Point Idéal	21
4.4 Point Nadir	22
5 Complexité et difficulté de l'optimisation combinatoire multi-objectif	22
6 Méthodes de résolution des problèmes (PMO)	23
6.1 Méthodes de résolution exacte	23
6.2 Méta-heuristiques multi-objectif	25
6.3 Méthodes hybrides de résolution des problèmes (POML)	27
6.4 Méthodes floues ou brouillées	28

3	<i>Programmation fractionnaire linéaire</i>	31
1	Formulation du programme fractionnaire	31
2	Rappelle sur l'optimisation convexe	32
2.1	Notions sur la convexité	32
3	Le modèle générale d'un programme fractionnaire linéaire	33
4	Résolution d'un programme fractionnaire linéaire	34
4.1	La résolution graphique	34
4.2	La résolution analytique	35
5	Programmation fractionnaire linéaire discrète	39
5.1	Forme générale d'un programme fractionnaire linéaire discrète (PFLNE)	40
5.2	Relaxation	40
5.3	Méthodes de résolution des programmes fractionnaires linéaires discrets	41
4	<i>Notions de base de la logique floue</i>	45
1	Idées cle sur la théorie du flou [ZALILA, 2003]	45
2	Les nombres et les ensembles flous	46
2.1	Définitions	46
2.2	Opérations sur les ensembles flous	49
2.3	Nombres flous	50
2.4	Défuzzification	51
5	<i>Résolution d'un problème multi-objectif fractionnaire linéaire flou en nombres entiers</i>	53
1	Formulation du problème	53
2	Modification de la méthode "Isbell-Marlow"	55
3	Algorithme Solution proposée pour la résolution de problèmes (α – PMFLNE)	57
4	Un exemple illustratif	58
6	<i>Implémentation, Résultats et Discussion</i>	65
1	Choix du langage	65
1.1	Présentation de langage de programmation : MATLAB	66
2	les fonctions programmées	66
2.1	La fonction "Isabelle-Marlow" :	66
2.2	La fonction de resolution de chaque (PL) :	66
2.3	La fonction qui regroupe les deux fonctions" ε –contrainte" et " α –coupe" ainsi que le résultat finale de programme :	67
3	Exemple concret	67
4	Application de l'exemple sur Matlab	68
	Conclusion generale	70
	Bibliographie	73

Table des figures

1.1	Les etapes de la méthode de branch and bound	13
2.1	Espace décisionnel et espace objectif d'un problème d'optimisation multi-objectif(exemple avec deux variables de décisions et deux fonctions objectifs). .	17
2.2	Exemple d'optimalité locale	20
2.3	Exemple d'optimalité locale	21
2.4	Interprétation graphique de la méthode.	24
2.5	Fonction d'adhésion de Reardon	29
3.1	Exemple d'un ensemble convexe.	32
3.2	Graphe de l'exemple 2.	35
4.1	Les fonctions d'appartenances les plus utilisées	48
4.2	Fonction d'appartenance trapézoïdale (TrFN)	50
4.3	Fonction d'appartenance triangulaire (TFN)	51
4.4	La coupe α	52

Introduction générale

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 du décideur. L'optimisation multi-objectif est donc étroitement liée au domaine de la prise de décision multicritère. 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 multiobjectif s'apparente à l'identification de l'ensemble Pareto optimal, 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.

Dans de nombreuses situations pratiques, il est nécessaire d'utiliser des variables discrètes dans la modélisation du problème, par exemple, dans un problème de planification manufacturière, le nombre d'unités produites doit être un nombre entier. On parle alors d'optimisation discrète ou programmation en nombres entiers.

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

La théorie probabiliste a toujours été l'unique approche capable de représenter toutes les incertitudes présentes dans les problèmes de la vie courante. par ailleurs durant les

trois dernières décennies la recherche dans le domaine de l'intelligence artificielle, des mathématiques et des systèmes en ingénierie a démontré que la théorie probabiliste n'était efficace que pour un type spécifique d'incertitude, et que d'autres approches étaient possibles pour la compléter. Parmi ces nouvelles approches, la théorie des ensembles flous est particulièrement pertinente pour ce type de problèmes, elle offre une structure pour représenter l'information imprécise.

Dans ce mémoire, une tentative pour couvrir le problème multi-objectif fractionnaire linéaire flou en nombre entiers (*PMFLFNE*). Après un aperçu de certains concepts préliminaires concernant les nombres flous, nous introduisons le concept de α -solution efficace du problème à étudier. Le problème initial de programmation floue (*PMFLFNE*) est transformé en version non floue basée sur α spécifié l'ensembles α de niveau des nombres flous. Un algorithme a été développé pour obtenir α solution efficace au problème (*PMFLFNE*).

A cet effet ; nous avons structuré ce présent mémoire en sept chapitres ; nous introduisons dans le premier chapitre les différentes notions et concept de base de la programmation linéaire et les différentes méthodes de résolution, ensuite dans le deuxième chapitre nous présentons les concepts fondamentaux de l'optimisation multi-objectif et un état de l'art des méthodes de résolution de ces problèmes, nous définissons dans le troisième chapitre l'essentiel des éléments de la programmation fractionnaire linéaire telle que la convexité ...etc, nous traitons aussi les différentes méthodes de résolution, dans le quatrième chapitre comprend les notions des ensembles flous, le cinquième chapitre est consacré à la résolution de problème multi-objectif fractionnaire linéaire flou en nombres entiers suivie d'un exemple illustratif, la résolution du ce problème posé à fait l'objet du dernier chapitre, ont été explicité et implémenté sous matlab ; Finalement, nous clôturons ce mémoire par une conclusion générale.

Chapitre 1

Programmation Linéaire

Introduction

Les modèles de programmation linéaire permettent d'aborder un grand nombre de problèmes d'optimisation en apparence très différents, dans des contextes très divers. On peut considérer que, plutôt que d'être à proprement parler du domaine de la gestion, la programmation linéaire relève des mathématiques de la Recherche Opérationnelle et a des applications en gestion ainsi qu'en économie, en statistique, en physique,...etc. Il s'agit d'un outil versatile et puissant, régulièrement cité par les entreprises comme un des modèles les plus utilisés de la Recherche Opérationnelle.

1. Programmation linéaire continue

1.1. Programme linéaire

Un *programme linéaire* (PL) est un problème d'optimisation consistant à maximiser (minimiser) une fonction objectif linéaire à n variables soumises à un ensemble de contraintes exprimées sous forme d'équations ou d'inéquations linéaires.

1.2. Forme d'un Programme linéaire

Un programme linéaire peut être écrit sous une :

Forme générale

$$(\text{PL}) = \begin{cases} \text{Opt } Z_{LP}(X) = \sum_{j=1}^n c_j x_j \\ s.c \\ \sum_{j=1}^n a_{ij} x_j \leq, \geq \text{ ou } = b_i, \forall i = \overline{1, m} \\ x_j \geq 0, \forall j = \overline{1, n} \end{cases} \quad (1.1)$$

avec :

- Opt pouvant signifier Minimiser ou Maximiser selon le problème traité,
- Un vecteur $C = (c_1, c_2 \dots c_j)^t$ définissant les coefficients appliqués à chaque variable de décision dans la fonction objectif Z_{LP} ,
- Un ensemble de variables de décision représenté par un vecteur $X = (x_1, x_2 \dots x_j)^t$,
- Un vecteur $B = (b_1, b_2 \dots b_i)$ définissant les seconds membres,
- Un ensemble de contraintes représenté par une matrice $A = (a_{ij})$,

Forme canonique

$$(\text{PL}) = \begin{cases} \text{Opt } Z_{LP}(X) = C^T x \\ s.c \\ Ax \leq (\text{resp } \geq) b \\ x \geq 0 \end{cases} \quad (1.2)$$

Forme standard

$$(\text{PL}) = \begin{cases} \text{Opt } Z_{LP}(X) = C^T x \\ s.c \\ Ax = (\text{resp } =) b \\ x \geq 0 \end{cases} \quad (1.3)$$

? Théorème 1

§ On peut toujours écrire un problème canonique sous une forme standard et vis versa.

! Remarque

- ★ Les problèmes de minimisation et de maximisation sont en fait équivalents puisque
- ★ Min $Z = -\text{Max}(-Z)$ et $\text{Max } Z = \text{Min}(-Z)$.

1.3. La notion de dualité

La notion de dualité est un aspect très important de la programmation linéaire. Elle fut développée par John Newmann en 1947 où il a démontré qu'à tout modèle linéaire qu'on appelle programme primal (P) correspond un autre appelé programme dual (D).

Problème dual d'un problème de programmation linéaire Considérons le problème

de PL suivant :

$$(\mathbf{PL}) = \begin{cases} Z(\max) = C^T x \\ s.c \\ Ax \leq b \\ x \geq 0 \end{cases} \quad (1.4)$$

Son dual est le programme linéaire suivant :

$$(\mathbf{D}) = \begin{cases} W(\min) = b^T y \\ s.c y A^T \leq c \\ y \geq 0 \end{cases} \quad (1.5)$$

Définition du dual dans le cas général

Evidemment, on peut définir le *dual* de programme linéaire quelconque. le tableau suivant résume les correspondances entre primal et dual et permet d'écrire directement le dual d'un programme linéaire quelconque.

$Z(\text{Max}) = C^T x$	$W(\text{Min}) = b^T y$
$i^{eme} \text{Contrainte} \geq$	Variable $y_i \leq$
$i^{eme} \text{Contrainte} \leq$	Variable $y_i \geq$
$i^{eme} \text{Contrainte} =$	$y_i \in R$
$x_j \geq 0$	$j^{eme} \text{Contrainte} \geq$
$x_j \leq 0$	$j^{eme} \text{Contrainte} \leq$
$x_j \in R$	$j^{eme} \text{Contrainte} =$

- Soient x et y des solutions réalisables respectivement du primal et du dual. Si $c^t x = y^t b$ alors, x et y sont solutions optimales.
- Si un problème possède une valeur optimale infinie, son dual n'a pas de solutions.



Propriété 1 (dualité faible)

Si x et y sont des solutions réalisables respectivement du primal (PL) et de son dual (D), elle vérifient : $C^t x \leq y^t b$



Propriété 2 (dualité forte)

Si le problème primal possède une solution optimale, alors il en est de même pour le problème dual et de plus, les deux problèmes ont la même valeur optimale $Z^* = W^*$.

1.4. Résultats fondamentaux et méthodes de résolution de la programmation linéaire

Convexité et ensemble convexe

Définition 1.1

Un ensemble D non vide est dit *convexe* si et seulement si pour tout élément x et y de D et pour tout $\lambda \in [0, 1]$ $\lambda x + (1 - \lambda)y \in D$. [5]

Définition 1.2

L'intersection d'un nombre fini de demi-espaces fermés de $\mathbb{R}^n$ est appelé *polytope* fermé. [5]

Définition 1.3

On appelle *polyèdre*, un polytope convexe borné (c'est à dire compact convexe de $\mathbb{R}^n$). [5]

Base et solution de base

Soit le (PL) suivant :

$$(PL) = \begin{cases} Opt Z_{LP}(X) = C^T x \\ s.c \\ Ax = b \dots (1) \\ x \geq 0 \dots (2) \end{cases} \quad (1.6)$$

Définition 1.4

$D = \{x \in \mathbb{R}^n, Ax = b, x \geq 0\}$ est dit ensemble des solutions réalisables. [5]

Définition 1.5

Tout vecteur vérifiant les contraintes (1) et (2) est appelé "solution réalisable" (admissible) du problème (P). [5]

Définition 1.6

Une solution réalisable x^* est dite optimale si : $C^T x^* = \{ \max C^T x, \forall x \in D \}$. [5]

Définition 1.7

Une solution réalisable x est dite de base si $(m \times n)$ de ses composantes sont nulles et les autres $x_{j1}, x_{j2}, \dots, x_{jm}$, correspondent au m vecteurs $a_{j1}, a_{j2}, \dots, a_{jm}$, de la matrice qui sont linéairement indépendants. [5]

Optimalité en un point extrême

? Théorème 2

L'ensemble des *points extrêmes* du polygone convexe $X = \{x \in \mathbb{R} / Ax=b, x \geq 0\}$, est égale à l'ensemble des solutions de bases réalisables.[9]

? Théorème 3

La fonction objectif du problème, atteint son maximum en un *point extrême* de l'ensemble X des solutions réalisables.[6]

Si la fonction objectif atteint son maximum en plusieurs points, elle prendra cette même valeur maximale en tout points qui est combinaison linéaire convexe de ces points extrêmes.

? Théorème 4(théorème d'optimalité)

Soit un (PL) mis sous forme canonique relativement à une base réalisable.
Si le vecteur d'estimation E est positif ou nul alors la solution de base est optimale.
Une telle base est dite base optimale.[9]

1.5. Méthodes de résolution

Méthode de simplexe

L'algorithme du simplexe est une méthode générale de résolution des programmes linéaires dont les fondements ont été énoncés et publiés en 1947 par G.Danzig. C'est une technique informatique d'une grande richesse conceptuelle et d'une extraordinaire efficacité qui résulte d'expériences établies par la résolution de milliers de problèmes pratiques (qui trouvaient d'ailleurs leurs origines dans des secteurs variés de la vie économique) pouvant se modéliser comme des programmes linéaires. De plus, la méthode du simplexe avait une certaine profondeur mathématique et se révélait un outil de démonstration de résultats théoriques nouveaux, elle apporte donc cette fameuse synthèse "théorique - pratique" qui donne à la recherche opérationnelle et à la programmation linéaire, en particulier, ses premières lettres de noblesse en tant que discipline scientifique.

Principe de la méthode du simplexe

La recherche systématique d'une solution optimale à l'aide d'un algorithme du simplexe peut se résumer comme suit :

1. Déterminer une première solution de base réalisable, cette solution initiale sert de départ au cheminement vers la solution optimale si elle existe.
2. Si la solution obtenu en 1 n'est pas optimale déterminer une autre solution de base

réalisable qui permettrait d'améliorer la fonction objectif (augmentation pour une maximisation ou diminution pour une minimisation)

3. On répète cette procédure itérative jusqu'à ce qu'il ne soit plus possible d'améliorer la fonction objectif. La dernière solution de base réalisable obtenue constitue la solution optimale au programme linéaire.

Algorithm 1 L'algorithme du simplexe

Soit B une base réalisable de départ, alors l'écriture canonique de (PL) par rapport à B est :

I_m	$\hat{A}_N = A_B^{-1} A_N$	$\hat{b} = A_B^{-1} b$
$0_{\mathbb{R}^n}$	$\hat{c}_N^t = c_N^t - c_B^t A_B^{-1} A_N$	$z - c_B^t A_B^{-1} b$

Etape 1 : si $\hat{c}_N \leq 0$, terminer ; la solution courante est optimale.

Etape 2 : Déterminer quelle variable x_s , $s \in N$, rentre en base en sélectionnant

$$\hat{c}_s = \max_{j \in N} \{\hat{c}_j\}.$$

Etape 3 : si $\forall i = 1, \dots, m$, $\hat{a}_{is} < 0$, terminer ; la fonction objectif n'est pas bornée Sinon, choisir une ligne r pour déterminer la variable qui doit quitter la base telle que :

$$\frac{\hat{b}_r}{\hat{a}_{rs}} = \min_{i=1, \dots, m} \left\{ \frac{\hat{b}_i}{\hat{a}_{is}} \mid \hat{a}_{is} > 0 \right\}$$

$$B \leftarrow B \cup \{s\} \setminus \{f(r)\}$$

où $f(i) \in \{1, \dots, n\}$ est l'indice de la variable de base associée à la ligne i, $i \in \{1, \dots, m\}$. L'élément $\hat{a}_{rs}$ est appelé pivot.

Etape 4 : Les m premières lignes du tableau 1 ainsi que la dernière ligne sont alors calculées de la façon suivante :

$$\begin{aligned} l_i &\leftarrow l_i - \frac{l_r \hat{a}_{is}}{\hat{a}_{rs}} ; i = 1, \dots, m, i \neq r \\ l_r &\leftarrow l_r - \frac{l_r}{\hat{a}_{rs}} \\ l_{m+1} &\leftarrow l_{m+1} - \frac{l_r \hat{a}_{is}}{\hat{a}_{rs}} \end{aligned}$$

Aller à l'étape 2.

Méthode de dual du simplexe

La méthode de dual du simplexe a été développée par C.E.Lemke. Le principe de cette méthode consiste à partir d'une solution non réalisable ($x_B < 0$) obtenir une solution réalisable ($x_B \geq 0$).

Algorithm 2 Algorithme dual du simplexe

Les étapes pour appliquer l'algorithme dual du simplexe sont les suivantes :

Etape 1 : Ecrire le problème sous forme standard

Etape 2 : Obtenir une solution de base de départ

avec $-z_j \leq 0 \quad j \in N$.

Etape 3 : Critère de sortie d'une variable : la variable x_r sort de la base d'après

$$x_r = \min \{x_{\hat{b}_i} / x_{\hat{b}_i} < 0\}$$

(a) S'il n'existe pas terminer.

(b) Sinon aller à (4)

Etape 4 : Critère d'entrée d'une variable : la variable x_s introduite dans la base d'après :

$$x_s = \min \{ -z_j / v_{rj}, v_{rj} < 0 \}$$

telle que le pivot est v

Etape 5 : pivoter sur v_{rs} en utilisant les mêmes règle que l'algorithme de simplexe.

Etape 6 : Critère d'optimalité : On obtient une solution réalisable optimale si tous les $x_{br} \geq 0$ et tous les $-z_j \leq 0$.

Etape 7 : Si la solution de base n'est pas réalisable, on poursuit l'algorithme (allez à 4).

Etape 8 : Sinon terminer.

2. Programmation linéaire en nombres entiers

La programmation linéaire en nombres entiers est un domaine très riche de la programmation mathématique. Les recherches dans ce domaine sont nombreuses et ont vu leurs naissances en 1958. Un problème de programmation linéaire en nombres entiers (*PLNE*) est un programme linéaire, sous des contraintes linéaires, dans lequel il y a la contrainte supplémentaire que les variables sont entières.

2.1. Forme générale d'un programme linéaire en nombres entiers

La forme générale d'un programme linéaire en nombres entiers se présente comme suit :

$$(\text{PLNE}) = \begin{cases} \text{Opt } Z_{LP}(X) = \sum_{j=1}^n c_j x_j \\ s.c \\ \sum_{j=1}^n a_{ij} x_j \leq, \geq \text{ ou } = b_i, \forall i = \overline{1, m} \\ x_j \geq 0, \forall j = \overline{1, n} \\ x_i \text{ entier } j = \overline{1, k} (k \leq n). \end{cases} \quad (1.7)$$

Relaxation Soit la forme générale du (*PLNE*) suivante :

$$(\text{PLNE}) = \begin{cases} \text{Opt } Z_{LP}(X) = \sum_{j=1}^n c_j x_j \\ \text{s.c} \\ \sum_{j=1}^n a_{ij} x_j \leq, \geq \text{ ou } = b_i, \forall i = \overline{1, m} \\ \geq 0 \quad j = \overline{1, n}, \forall j = \overline{1, n} x_j \\ x_i \in Z^+ \quad j = \overline{1, k} \quad (k \leq n). \end{cases}$$

(1.8)

2.2. Méthodes de résolution

Ces méthodes sont capables de trouver une solution optimale d'un problème donné dans un intervalle de temps bien déterminé mais exponentiel sur les problèmes NP-complets (notamment les (PLNE)). Parmi ces méthodes on retrouve la Méthode des coupes, la méthode de séparation et d'évaluation (Branch and Bound), et la méthode que combine les deux dernières (Branch-and-Cut).

Coupes de Gomory

L'algorithme utilisant les coupes de Gomory procède comme suit :

Soit $\bar{x}$ une solution optimale du problème relaxé (PL) trouvée par la méthode du simplexe. Si $\bar{x}$ est une solution entière, elle est optimale pour (PLNE). Sinon, choisir une variable x_j telle que la valeur $\bar{x}_j$ est fractionnaire et considérer la ligne correspondante du tableau du simplexe, par exemple la ligne i :

$$\sum_{k \in N} \hat{a}_{ik} x_k = \bar{x}_j \quad (1.9)$$

Où N est l'ensemble des indices des variables hors-base

La contrainte

$$\sum_{k \in N} f(\hat{a}_{ik}) x_k \geq f(\bar{x}_j) \quad (1.10)$$

est alors déduite de l'expression précédente, où $f(r) = r - \lfloor r \rfloor$ désigne la partie fractionnaire du nombre réel r .

Cette coupe, appelée coupe fractionnaire de Gomory, ou coupe fondamentale, peut être rajouter au tableau courant du simplexe.

Dans la pratique, les coupes fractionnaires de Gomory convergent très lentement.

D'autres coupes ont été introduites dont l'intention de les rendre plus performantes.

Méthode par séparation et évaluation

Principe de la méthode :

Le principe des procédures par séparation et évaluation est basé sur l'exploration d'un arbre de solutions. Toutes les solutions possibles du problème sont séparées en deux sous ensembles ou plus, chacun d'eux est représenté par une branche de l'arbre de décision.

En utilisant cette méthode, nous examinons, initialement, tout l'ensemble de solutions S . Dans la phase d'évaluation, nous admettons des solutions n'appartenant pas à S (non réalisables). Cette étape permet de déterminer une borne inférieure au problème et consiste en une énumération des solutions du problème par exploration partielle de l'arborescence des solutions obtenues par décomposition successives du problème.

L'efficacité de cette exploration arborescente peut être grandement accrue si on a un moyen d'éviter l'examen de toutes les parties en lesquelles l'ensemble des solutions est décomposé.

Dans les procédures par S-E, c'est la fonction d'évaluation qui permet d'éliminer des branches complètes de l'arborescence et de localiser la solution optimale, on peut dire qu'il s'agit d'une "exploration intelligente" du domaine des solutions réalisables du problème d'optimisation combinatoire considéré. Cette méthode opère comme suit :

- Décomposer l'ensemble des solutions réalisables en plusieurs sous-ensembles.
- Associer à chaque sous-ensemble une évaluation qui est une borne inférieure de la fonction objectif pour un problème de minimisation (respectivement une borne supérieure de la fonction objectif pour un problème de maximisation).

Si la plus petite évaluation (respectivement la plus grande) est associée à une solution réalisable, alors cette solution est optimale, sinon, séparer l'ensemble auquel correspond l'évaluation.

Présentation des procédures par séparation et évaluation

Les procédures par S-E ne constituent qu'un cadre général à l'intérieur duquel les différents algorithmes sont spécifiés, quand on précise d'une part comment sont définis les sous-ensembles en lequel l'ensemble fondamental des solutions d'un problème d'optimisation combinatoire S sera progressivement décomposé d'une part, et comment on effectue le choix du sous-ensemble à traiter, les modes d'évaluation et de séparation, d'autre part. Cependant la méthode générale peut être définie de manière plus précise.

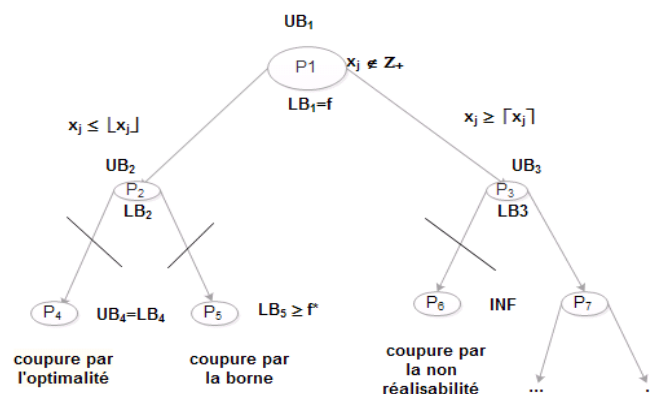


FIGURE 1.1 : Les etapes de la méthode de branch and bound

On utilisera un schéma de résolution du type suivant :

-Chaque fois que de nouveaux sous-ensembles de S sont engendrés, on leur associe une évaluation, borne inférieure (respectivement), une borne supérieure pour un problème de maximisation) du coût des solutions appartenant à ce sous-ensemble.

-Si l'on dispose d'une solution réalisable dont le coût est inférieur (respectivement supérieur pour un problème de maximisation) ou égal aux évaluations des différents sous-ensembles constituant la partition courante de S , terminer : cette solution est optimale.

Sinon, on choisit un sous-ensemble de S susceptible de contenir une solution optimale et on le sépare (on le partitionne) en sous-ensembles plus petits. Et on recommence.

La première chose à faire, quand on veut définir une procédure par S-E, est de spécifier comment seront caractérisés les sous-ensembles qui, à chaque instant, recouvrent l'ensemble S . Le principe de séparation est évidemment lié de manière étroite, le plus souvent, à la définition de ces sous-ensembles.

Algorithme par S-E pour la résolution de (PLNE)

Soit le programme linéaire (PLNE) issu de la modélisation du problème posé, soit (PLNE)' le programme linéaire relaxé obtenu à partir de (PLNE) en relâchant les contraintes d'intégrité sur les variables Y_{it} (remplacer les conditions $Y_{it} \in 0,1$ par $0 \leq Y_{it} \leq 1$. et soit S l'ensemble des solutions réalisables de (PLNE)' qui peuvent être obtenues soit par tâtonnement ou fournies par l'entreprise sans utiliser l'algorithme du simplexe et sans faire aucun pivotage, on peut effectuer les évaluations de manière directe, augmentant ainsi considérablement l'efficacité de l'algorithme.

A l'optimum de (PLNE)' ; $Z((PLNE)')$ qu'on notera Z' est une évaluation de l'ensemble S , si la solution de (PLNE)' est à composantes entières alors Z' est une évaluation exacte.

Ainsi pour la résolution de P , on utilise le schéma suivant :

(1) Procédure d'initialisation

Y^0 : meilleure solution actuelle.

$v = +\infty$

(2) Procédure de choix

Pour choisir le nœud qu'on doit traiter, on doit parcourir toutes les branches de l'arborescence afin de repérer la branche dont la valeur de la fonction objectif est minimum, ce qui signifie que la sélection se fera selon la stratégie du meilleur d'abord.

(3) Procédure d'évaluation

Chaque nœud de l'arborescence représente un programme linéaire qui est résolu à l'aide de la méthode du simplexe. La fonction objectif sera évaluée pour chaque nœud et qui sera utile par la suite pour la sélection de la branche qui doit être explorée.

Si les composantes de la solution obtenue sont entières, terminer (évaluation exacte).

$v := Z^0$

$Y^0 := Y$

Stériliser.

(4) Procédure de séparation

Pour séparer l'ensemble S , choisir l'indice r tel que la $r^{ième}$ composante associée à Y^n n'est pas entière car la n n'est pas entière.

Séparer S en S_1 et S_2 avec : S_1 : l'ensemble des solutions réalisables du programme $(PLNE)'_1$ obtenu en rajoutant la contrainte $Y_{it} \geq \lfloor Y_{it} \rfloor$

S_2 : l'ensemble des solutions réalisables du programme $(PLNE)'_2$ obtenu en rajoutant la contrainte $Y_{it} \leq \lfloor Y_{it} \rfloor + 1$.

Associer au sommet de l'arborescence deux sommets fils.

Stérilisation :

Evaluation exacte :

stériliser les sous-ensembles correspondant et tout sous-ensemble dont l'évaluation est supérieure à celle ci. Cet algorithme se termine quand tous les sommets sont soit terminaux ou éliminés, donc chaque sommet de l'arborescence sera classé dans un des trois états suivants :

1. Exploré : ses successeurs sont créés par séparation et sont évalués.
2. Actif : non encore exploré.
3. Éliminé son évaluation est supérieure à la valeur de la meilleure solution trouvée.

Méthode "branch and cut"

Cette méthode couple la méthode par séparation et évaluation avec des coupes pour résoudre des programmes linéaires en nombres entiers ou mixtes. Le principe est de résoudre la relaxation continue du programme linéaire en nombres entiers à l'aide de l'algorithme du simplexe. Si la solution optimale du problème relaxé n'est pas entière, on construit une coupe (ou plusieurs) à partir du tableau optimal du simplexe qu'on rajoute à ce dernier et on optimise de nouveau le tableau du simplexe obtenu. On répète ce procédé jusqu'à ce qu'une solution entière, soit trouvée, (solution optimale dans le cas d'un (PLNE)), ou un test d'arrêt est vérifié. A cette étape, si on n'a pas encore trouvé de solution entière, alors la partie séparation et évaluation de l'algorithme commence.

Conclusion

On a intéressé dans ce chapitre à la présentation de programmation linéaire (continue et discrète) d'une façon générale.

Chapitre 2

Optimisation Multi-Objectif

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-objectif 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-objectif.

1. Définition :

Un problème d'optimisation multi-objectifs consiste à rechercher les meilleures solutions qui minimisent un nombre M de fonctions, appelées fonctions coût, $f_m[X], m = 1...M$ par rapport à un vecteur X , qui est le vecteur des n variables de contrôle (ou paramètres) :

$X = [x_1 \ x_2 \dots x_n]^T$, en satisfaisant un certain nombre de contraintes, explicites comme les contraintes de borne ($x_i^I \leq x_i \leq x_i^S, i = 1, \dots, n$) ou implicites, d'égalité ($h_k(x) = 0, k = 1, \dots, K$) ou d'inégalité ($g_j(x) \geq 0, j = 1, \dots, J$). On se limite aux problèmes de minimisation, puisque la maximisation d'une fonction $f(X)$ peut facilement être transformée en un problème de minimisation :

$$\max(f(X)) = -\min(-f(X))$$

Un problème d'optimisation peut s'écrire sous la forme générale suivante :

$$(PM) \begin{cases} Opt f_m(X) , m \geq 1 \\ s.c \\ g_j(x) \geq 0, j = 1, \dots, J \\ x_i \geq 0 \end{cases}$$

Dans notre mémoire, on considère le cas de maximisation.

Quand le problème d'optimisation contient une seule fonction coût ($M = 1$), il est appelé mono-objectif. Ce problème, utilise un seul espace de recherche, connu aussi sous le nom d'espace de décision S . Il s'agit de l'ensemble des combinaisons possibles des paramètres qui satisfont toutes les contraintes. En revanche, si le problème d'optimisation contient plusieurs fonctions coût ($M > 1$), et si les fonctions coût sont antagonistes, il est appelé multiobjectif. Un problème d'optimisation multi-objectifs porte sur deux espaces, que sont l'espace de décision S et l'espace fonctionnel (l'espace des fonctions coût) F .

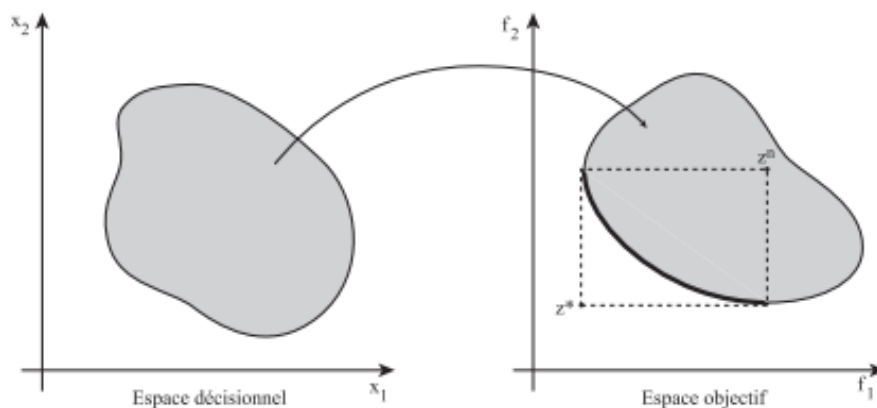


FIGURE 2.1 : Espace décisionnel et espace objectif d'un problème d'optimisation multi-objectif(exemple avec deux variables de décisions et deux fonctions objectifs).

2. La dominance

2.1. Introduction et définitions

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

Définition 2.1

Une solution $y = (y_1, \dots, y_m)^T$ domine faiblement une solution $z = (z_1, \dots, z_m)$ Ssi $\forall i \in 1 \dots m, f_i(y) \leq f_i(z)$
si la solution y domine faiblement la solution z nous allons noter $y \preceq z$. [4]

Définition 2.2

Une solution $y = (y_1, \dots, y_m)$ domine une solution $z = (z_1, \dots, z_m)$ Ssi :
 $\forall i \in \{1 \dots m\} f_i(y) \leq f_i(z)$, et $\exists j \in \{1 \dots m\}$ tq $f_j(y) < f_j(z)$
Si la solution y domine la solution z nous notons alors $y < z$. [4]

Notons que pour toute paire de solution y et z un et un seul des cas suivants peut se présenter :

- y domine z
- y est domine par z
- y et z sont équivalentes au sens de la dominance.

Les solutions équivalentes au sens de la dominance sont appelées dans ce qui suit, solutions équivalentes au sens de Pareto ou solutions Pareto équivalentes ou encore, solutions non-dominées.



Propriétés de la relation de dominance

- La relation de dominance telle qu'elle est définie ci-dessus : [4]
- n'est pas réflexive car une solution ne se domine pas elle même.
 - n'est pas symétrique, car on n'a jamais $y < z$ et $z < y$.
 - est transitive, car si $y < z$ et $z < w$ alors $y < w$.

2.2. Optimalité de pareto

Soit un ensemble (P) de solution candidates d'un problème d'optimisation multi-objectif. L'ensemble $P' \subseteq P$, composé de tous les éléments de P qui ne sont dominés par aucun élément de P est dit sous-ensemble non dominé de l'ensemble P .

- Optimalité locale au sens de pareto : Une solution y est optimale localement au sens de pareto s'il existe un réel $\delta > 0$ telqu'il n'ait pas une solution z dominant y et vérifiant $\|z - y\| \leq \delta$
- Optimalité globale au sens de pareto : Une solution y est optimale globalement au sens de pareto, ou optimale au sens de pareto, ou encore preto-optimale s'il n'existe aucun point de l'espace faisable D qui la domine. L'ensemble des solutions Pareto optimale est

appelé l'ensemble de Pareto ou également l'ensemble de compromis optimaux. L'image de l'ensemble de Pareto dans l'espace des critères est appelé la surface de pareto (ou le front de pareto pour les cas bi-objectif) ou encore la surface des compromis optimaux.

3. Efficacité(ou Pareto optimale)

Définition 2.3

Une solution $x^* \in X$ est une solution *efficace* s'il n'existe pas de $x \in X$ tel que $z(x)$ domine $z(x^*)$. [4]

3.1. Efficacité faible

Définition 2.4

Une solution $x^* \in X$ est une solution *faiblement efficace* s'il n'existe pas de $x \in X$ telle que $z(x) < z(x^*)$. Une solution est faiblement efficace si son vecteur critère n'est pas fortement dominé. [4]

3.2. Efficacité forte

Définition 2.5

Une solution $x^* \in X$ est une solution *fortement efficace* s'il n'existe pas de $x \in X$ telle que $x \neq x^*$ et $z(x) = z(x^*)$. Une solution x est fortement efficace s'il n'existe pas une autre solution telle que le vecteur critère, qui lui est associé, soit aussi bon que celui de x . Remarquons que l'efficacité forte implique l'efficacité qui implique à son tour l'efficacité faible. [4]

4. Front Pareto et surface de compromis

Nous rappelons que la solution que nous cherchons á un problème d'optimisation multi-objectif n'est pas un point unique mais un ensemble de points que nous avons appelé l'ensemble des meilleurs compromis

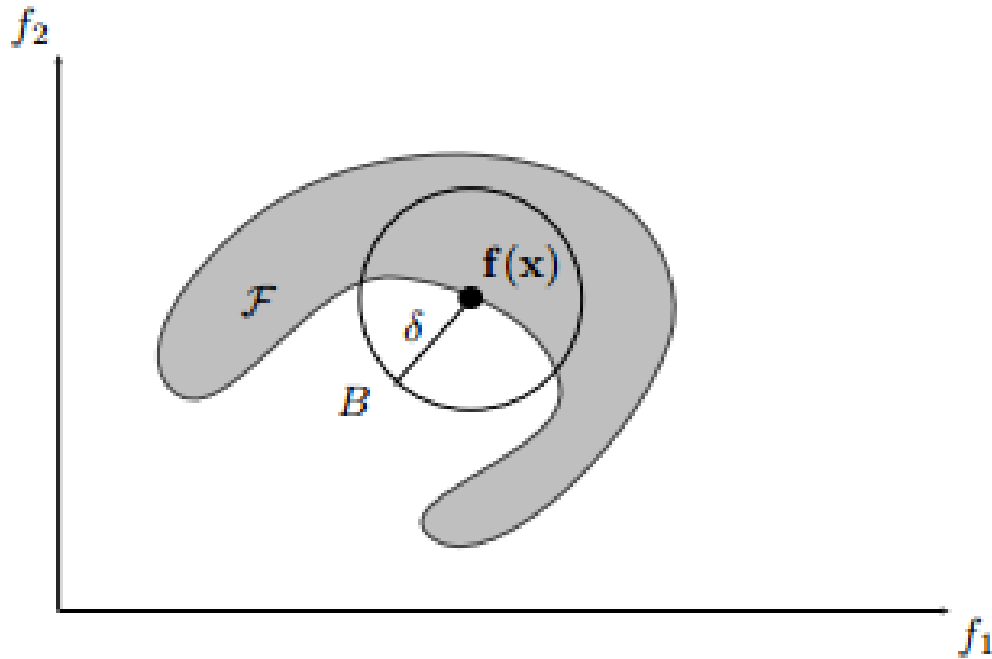


FIGURE 2.2 : Exemple d'optimalité locale

Nous avons défini jusqu'ici les notions de dominance (au sens de Pareto) et d'optimalité. Il nous reste maintenant à définir la solution d'un problème multi-objectif utilisant ces concepts. Cet ensemble des meilleurs compromis, appelé aussi surface de compromis ou front Pareto, est composé des points qui ne sont dominés par aucun autre. Nous définissons d'abord formellement l'ensemble des solutions non dominées :

4.1. Ensemble des solutions non dominées :

Définition 2.6

[4] Soit F l'image dans l'espace des objectifs de l'ensemble réalisable X . L'ensemble des solutions non dominées de X , est défini par l'ensemble $ND(X)$:

$$ND(X) = \{x \in X \mid x \text{ est non dominé par rapport à } X\}$$

Nous pouvons définir le front Pareto de F de manière analogue :

4.2. Front Pareto

Définition 2.7

Soit F l'image dans l'espace des objectifs de l'ensemble réalisable X . Le front Pareto $ND(F)$ de F est défini comme suit :

$$ND(F) = \{y \in F \mid \nexists z \in F, z < y\}$$

Le front Pareto est aussi appelé l'ensemble des solutions efficaces ou la surface de compromis.[4]

Un exemple de surface de compromis (front Pareto) en dimension 2 est montré dans la **figure 2.3**. Dans cet exemple, le problème considéré est un problème de minimisation avec deux critères. Deux points particuliers apparaissent clairement : le point idéal et le point Nadir. Ces deux points sont calculés à partir du front Pareto. Le point idéal (resp. le point Nadir) domine (resp. est dominé par) tous les autres points de la surface de compromis. Bien que ces points ne soient pas forcément compris dans la zone réalisable.

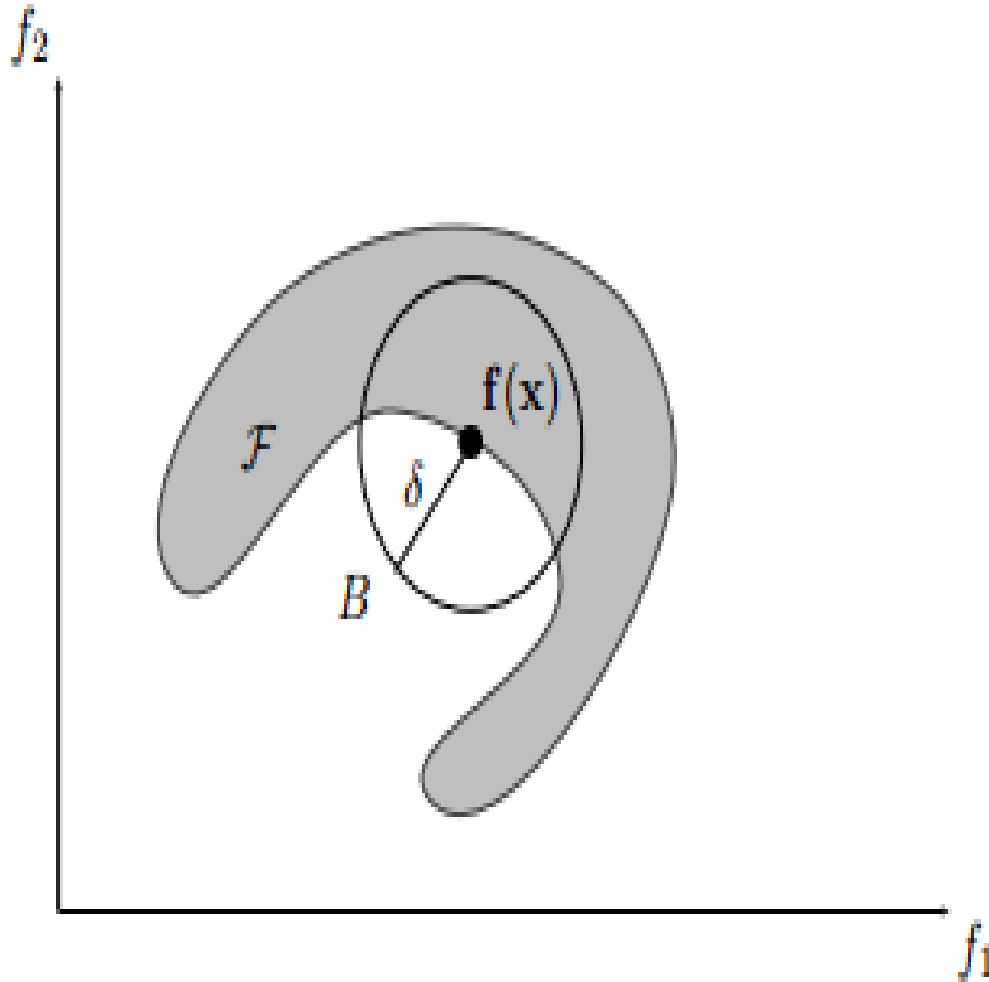


FIGURE 2.3 : Exemple d'optimalité locale

Les coordonnées de ces deux points sont maintenant définies formellement.

4.3. Point Idéal

[46] Les coordonnées du point *idéal* (p_i) correspondent aux meilleures valeurs de chaque objectif des points du front Pareto. Les coordonnées de ce point correspondent aussi aux valeurs obtenues en optimisant chaque fonction objectif séparément.

$$p_{ii} = \min y_i | y \in ND(F)$$

4.4. Point Nadir

Les coordonnées du *point Nadir* (pn) correspondent aux pires valeurs de chaque objectif des points du front Pareto.

$$p_{ni} = \max y_i | y \in ND(F)$$

La **figure 2.3** nous indique aussi que le front Pareto peut avoir des propriétés particulières quant à sa forme. La principale caractéristique utilisée pour comparer les formes de ces courbes est la convexité.[46]

Nous rappelons donc la définition de la convexité :

Convexité :

Un ensemble A est *convexe*, si et seulement si l'équivalence suivante est vérifiée :

$$x \in A \wedge y \in A \iff \text{Segment}(x, y) \subset A$$

La convexité est le premier indicateur de la difficulté du problème. En effet, certaines méthodes sont dans l'incapacité de résoudre des problèmes non convexes de manière optimale. Mais il existe d'autres indicateurs tout aussi importants, notamment la continuité, la multimodalité, la nature des variables de décision (entières ou réelles),...[16]

5. Complexité et difficulté de l'optimisation combinatoire multi-objectif

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

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

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

- Les algorithmes sous-linéaires, dont la complexité est en général en $o(\log n)$. C'est le cas de la recherche d'un élément dans un ensemble ordonné fini de cardinal n ;
- Les algorithmes linéaires en complexité $o(n)$ ou en $o(n \log n)$ sont considérés comme rapides, comme l'évaluation de la valeur d'une expression composée de n symboles ou

les algorithmes optimaux de tri ;

- Plus lents sont les algorithmes de complexité située entre $o(n^2)$ et $o(n^3)$, c'est le cas de la multiplication des matrices et du parcours dans les graphes ;
- Au delà, les algorithmes polynômiaux en $o(n^k)$ pour $k > 3$ sont considérés comme lents, sans parler des algorithmes exponentiels (dont la complexité, est supérieure à tout polynôme en n), que l'on s'accorde à dire inutilisables dès que la taille des données est supérieure à quelques dizaines d'unités.

6. Méthodes de résolution des problèmes (PMO)

6.1. Méthodes de résolution exacte

Il y a très peu de travaux sur les méthodes exactes dans le contexte de la résolution des problèmes d'optimisation multi-objectif, sans doute, à cause de la grande difficulté de ce type de problème.

Les références existantes et qui présentent la plupart des méthodes exactes sont Ulungu et Ehrgott et Gandibleux 2000.

L'agrégation linéaire

Cette méthode populaire transforme le problème multi-objectif en un problème mono-objectif en combinant linéairement les différents objectifs. Ainsi, le nouveau problème obtenu, car il s'agit alors d'un problème différent, consiste à optimiser $\sum_i \lambda_i f_i$

Le théorème de Geoffrion indique qu'en utilisant différentes valeurs pour le vecteur λ , il est possible d'obtenir toutes les solutions supportées du problème multi-objectif initial.

Par contre, aucune solution non supportée peut être trouvée par cette méthode.

La méthode d'agrégation linéaire a donc ses limites. Toutefois, elle est intéressante pour des problèmes ayant de nombreux objectifs et/ou un grand nombre de solutions supportées bien réparties. Dans ce contexte, il peut être suffisant de générer les solutions supportées.[51]

Méthode ϵ contrainte

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

La démarche est la suivante :

- Nous choisissons un critère à optimiser prioritairement
- Nous transformons le problème conservant l'objectif prioritaire et nous nous transfor-

mons les autres objectifs en des contraintes d'inégalité Le problème peut être reformulé de la manière suivante :

$$\left\{ \begin{array}{l} \max[f_i(x)] \\ tq : \\ f_1(x) \leq \epsilon_1 \\ \vdots \\ f_{i-1}(x) \leq \epsilon_{i-1} \\ f_{i+1}(x) \leq \epsilon_{i+1} \\ \vdots \\ f_m(x) \leq \epsilon_m \\ \text{et que } g(x) \leq 0 \\ \text{avec, } x \in \mathbb{R}^n, f(x) \in \mathbb{R}^m, g(x) \in \mathbb{R}^q \end{array} \right.$$

L'approche par ϵ -contrainte doit aussi être appliquée plusieurs fois en faisant varier le vecteur ϵ pour trouver un ensemble de points Pareto optimaux.

Cette approche a l'avantage par rapport à la précédente de ne pas être trompée par les problèmes non convexes.

Ainsi la figure suivante illustre, en dimension 2, le cas où un point (ϵ, f_{1min}) , de la partie non convexe, est trouvé. Cette figure montre aussi comment cette approche procède.

En transformant des fonctions objectifs en contraintes, elle diminue la zone réalisable par paliers. Ensuite, le processus d'optimisation trouve le point optimal sur l'objectif restant. L'inconvénient de cette approche réside dans le fait qu'il faille lancer un grand nombre de fois le processus de résolution. De plus, pour obtenir des points intéressants et bien répartis sur la surface de compromis, le vecteur ϵ doit être choisi judicieusement.

Il est clair qu'une bonne connaissance du problème a priori est requise.[1]

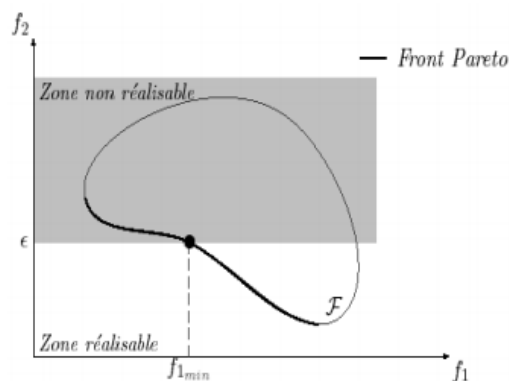


FIGURE 2.4 : Interprétation graphique de la méthode.

6.2. Méta-heuristiques multi-objectif

La classe des Métaheuristiques pour l'optimisation multi-objectif sont ceux qui emploient méthodes approchées et utilisent souvent la recherche locale ou l'exploration du voisinage pour conduire la recherche ou comme composant important du processus (approches d'hybride). Plusieurs métaheuristiques multi-objectif employant la recherche locale ont été proposées dans la littérature.

Les métaheuristiques ont été appliquées avec succès sur un grand nombre de problèmes académiques et réels : problème d'affectation quadratique, coloriage de graphe, voyageur de commerce, ... ect.

Ehr Gott et Gandibleux classent ces méthodes essentiellement en deux classes : les algorithmes évolutionnaires (avec plusieurs solutions) et les algorithmes d'exploration de voisinage (à solution unique).

Les algorithmes d'exploration de voisinage (à solution unique)

La Recherche Tabou

La recherche Tabou est une métaheuristique développée par Glover ([Glover, 86], [Glover, 90], [Glover, 97]). Elle est basée sur des idées simples, mais elle est néanmoins très efficace. Elle combine une procédure de recherche locale avec un certain nombre de règles et de mécanismes pour surmonter l'obstacle des optima locaux, tout en évitant de cycliser. Elle a été appliquée avec succès pour résoudre de nombreux problèmes difficiles d'optimisation combinatoire. Le principe de la méthode Tabou est simple : On part d'une solution initiale x , puis on cherche le meilleur voisin $s(x)$ dans le voisinage de x [$s(x)$], l'originalité de la méthode réside dans le fait que l'on retient le meilleur voisin de x même si celui-ci est plus mauvais que x . Ce critère autorisant les dégradations de la fonction objectif évite à l'algorithme d'être piégé dans un minimum local, mais il induit un risque de cyclage. Pour régler ce problème, l'algorithme mémorise la liste des transformations interdites (telle que de $s(x)$ à x) dans la liste Tabou interdisant la transformation inverse d'une transformation faite récemment.[51]

Algorithm 3 Algorithme Recherche Tabou

```

1 :  $L = \emptyset$  (liste des points tabous)
2 : Init  $x$  (point de départ)
3 : Init  $k_{fin}$  (nb d'itérations max total )
4 : Init  $k = 0$  et  $l = 0$ 
5 : While (not(end))
6 :    $y = \text{voisinage}(x) - L$  (on prend un voisinage sans points tabous)
7 :    $DC = C(y) - C(x)$ 
8 :    $x = y$  avec  $y \in \text{voisinage}(x) - L$  tel que  $\min C(y) - C(x)$ 
9 :    $L = L + x$ 
10 :    $k = k + 1$ 
11 :   Update L (on retire des mouvements tabou de L suivant des critères d'aspiration)
12 :   If  $k = k_{fin}$  then end(while) repeat

```

Le Recuit Simulé

L'origine de la méthode du Recuit Simulé provient de la métallurgie, où, pour atteindre les états de basse énergie d'un solide, on chauffe celui-ci jusqu'à des températures très élevées, après on le laisse refroidir lentement. Ce processus est appelé le recuit. La méthode du recuit simulé a été développée simultanément par Kirk en 1983 et Cerny en 1985.[41]

Algorithm 4 Algorithme Recuit Simulé

```

13 : Init  $T$  (température initial), Init  $x$  (point de départ), Init  $\Delta T$  (temperature) ;
2 : tantque not end faire
3 :    $y = \text{Voisin}(x), \Delta C = C(y) - C(x)$ ;
4 :   si  $\Delta C < 0$  alors
5 :      $y = x$ ;
6 :   sinon
7 :     si  $\text{alea}(0,1) < e^{\Delta C}$  alors    $y = x$ ;  $T = \alpha(T)$ ;
8 :   finsi
9 :   finsi
10 :  finsi
11 :  si  $T < \Delta T$  alors
12 :    finsi
13 :  finsi
14 : fin tantque
15 : REPEAT(while);

```

Les algorithmes évolutionnaires(à solution multiple)

On peut distinguer trois grandes classes d'algorithmes évolutionnaires : les algorithmes génétiques [Holland, 1975 ; Goldberg, 1989], les stratégies d'évolution [Schwefel, 1981] et la programmation évolutive [Fogel, 2000]. Ces méthodes se distinguent par la manière de représenter l'information et par la façon de faire évoluer la population d'une génération à l'autre. Un algorithme évolutionnaire est typiquement composé de trois éléments fondamentaux :

- une **population** constituée de plusieurs individus représentant des solutions potentielles (configurations) du problème donné,
- un **mécanisme d'évaluation des individus** permettant de mesurer l'adaptation de l'individu à son environnement,
- un **mécanisme d'évolution de la population** permettant, grâce à des opérateurs prédéfinis, d'éliminer certains individus et d'en créer de nouveaux.

Parmi les composants d'un algorithme évolutionnaire, l'individu et la fonction d'évaluation correspondent respectivement à la notion de configuration et à la fonction d'évaluation dans les méthodes de voisinage. Le mécanisme d'évolution est composé de plusieurs opérateurs tels que la sélection, la mutation et le croisement.

La sélection a pour objectif de sélectionner des individus qui vont pouvoir se reproduire pour transmettre leurs caractéristiques à la génération suivante. Le croisement ou recombinaison est un opérateur permettant de construire de nouveaux individus enfants à partir des caractéristiques d'individus parents sélectionnés. La mutation effectue des légères modifications de certains individus.

Algorithmes génétiques

Les algorithmes génétiques (AGs) ont été introduits par Holland [Holland, 1975] comme un modèle de méthode adaptative. Le but principal d'un AG est de chercher, dans une space de solutions E , un élément de cet espace ayant la meilleure adaptation à l'environnement posé. Chaque élément de E est un individu, noté x . Une population P est donc un ensemble de N éléments (individus) de E : $P = (x_1, x_2, \dots, x_n)$. La mesure du degré d'adaptation de chaque individu constitue la fonction de performance $F(\text{fitness})$. L'objectif d'un AG est de faire évoluer cette population P afin de trouver le meilleur individu x^* . Dans ce but, à chaque génération t , les individus de la population $P(i)$ sont sélectionnés, croisés et mutés selon des probabilités prédéfinies respectivement p_c (probabilité de croisement) et p_m (probabilité de mutation).[1]

6.3. Méthodes hybrides de résolution des problèmes (POML)

Comme nous allons le voir, il est possible d'exploiter plusieurs méthodes pour en former une nouvelle. Dans ce cas, nous obtenons une "méthode hybride".

Présentation de la méthode

La méthode hybride la plus connue est la méthode de Corley. Cette méthode utilise la méthode de pondération des fonctions objectif et la méthode du compromis. On part du problème (P). On le transforme de la manière suivante :

$$(P_{cor}) : \begin{cases} \text{Maximiser } \sum_{i=1}^m w_i \cdot f_i(x) \\ f_i(x) \leq \epsilon_j, j \in 1, \dots, m \\ \text{et } g(x) \leq 0 \\ x \in \mathbb{R}^n \end{cases}$$

6.4. Méthodes floues ou brouillées

Ces méthodes font intervenir la logique floue à tous les niveaux (sur les paramètres du problème ainsi que sur les paramètres des contraintes). L'ensemble de solutions que l'on va trouver, lui aussi, fera intervenir la logique floue.

Méthode de Sakawa

Cette méthode fait intervenir la logique floue à tous les niveaux (sur les paramètres du problème ainsi que sur les paramètres des contraintes). L'ensemble des solutions que l'on va trouver, lui aussi, fera intervenir la logique floue. Ces solutions auront un niveau d'appartenance. Ce seront des solutions qui auront une corrélation variable avec l'objectif initial (le niveau de corrélation avec l'objectif initial est fixé par l'utilisateur).

Algorithme de la méthode de Sakawa

Algorithm 5 Algorithme de la méthode de Sakawa

Etape 1 : sous la contrainte $g(x) \leq 0$, on minimise, puis on maximise, chaque fonction objectif séparément, de manière à obtenir la plage de variation de la fonction d'appartenance des fonctions objectif.

Etape 2 : On définit les fonctions d'appartenance de la manière suivante :

$$\mu_i(x) = \frac{f_{i1} - f_i(x)}{f_{i1} - f_{i0}};$$

où

- f_{i0} est la valeur de la fonction d'appartenance la moins intéressante,
- f_{i1} est la valeur de la fonction d'appartenance la plus intéressante.

Etape 3 : On définit les fonctions de prise de décision DM_i (Decision Making) comme suit :

$$DM_i(x) = \begin{cases} 0, & \text{si } \mu_i(x) \leq 0 \\ \mu_i(x), & \text{si } 0 \leq \mu_i(x) \leq 1 \\ 1, & \text{si } \mu_i(x) \geq 1 \end{cases} \quad (2.1)$$

où

- $DM_i(x) = 0$, quand l'objectif n'est pas atteint,
- $DM_i(x) = 1$, quand l'objectif est atteint.

Etape 4 : On maximise les fonctions DM_i .

Etape 5 : S'il n'y a pas de solutions, ou si la solution ne satisfait pas l'utilisateur, alors il faut modifier les fonctions d'appartenance, et retourner à l'étape 3.

Etape 6 : Arrêt et affichage des résultats.

fin.

Méthode de reardon

On présente une méthode simplifiée de traitement multiobjectif utilisant la logique floue.

Présentation de la méthode

On part du problème P. Pour chaque fonction objectif, on définit une fonction d'appartenance, qui aura la forme représentée à la figure 2.5. Cette fonction d'appartenance permet "d'informer" l'algorithme que la fonction objectif a sa valeur située dans l'intervalle $[O_i - E_i, O_i + E_i]$ (zone où la fonction d'appartenance vaut 0). Si la valeur de la fonction objectif est située en dehors de cette zone, on pénalise de plus en plus la fonction objectif. Dans ce cas, la pénalité varie entre 0 et S_{min} et S_{max} .

La signification des différents paramètres de la fonction d'appartenance est la suivante :

S_{min} et S_{max} : facteurs d'échelle flous.

f_{min} : valeur minimale de la fonction objectif numéro i.

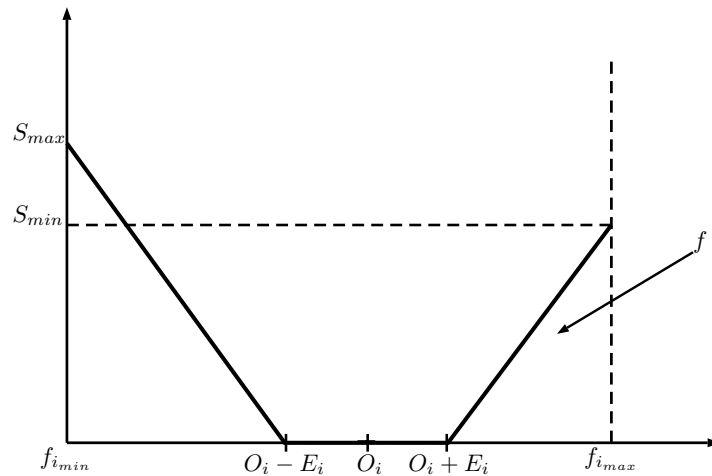


FIGURE 2.5 : Fonction d'adhésion de Reardon

f_{max} : valeur maximale de la fonction objectif numéro i.

O_i : valeur expérimentale de la fonction objectif numéro i : on voudrait que $f_i = O_i$.

E_i : marge d'erreur acceptable (l'objectif O_i est défini à $2 \cdot E_i$ près)

- si $f_i \leq (O_i - E_i)$ alors $f'(f_i) = \left[\frac{S_{max}}{f_{min} - (O_i - E_i)} \right] \cdot f_i - (O_i - E_i)$
- si $(O_i - E_i) \leq f_i \leq (O_i + E_i)$ alors $f'(f_i) = 0$
- si $f_i \geq (O_i + E_i)$ alors $f'(f_i) = \left[\frac{S_{min}}{(O_i + E_i) - f_{max}} \right] \cdot f_i - (O_i + E_i)$.

Conclusion

Dans ce chapitre nous avons donné un aperçu sur les problèmes d'optimisation multiobjectif ainsi que les méthodes de résolution proposées.

Chapitre 3

Programmation fractionnaire linéaire

Introduction

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

1. Formulation du programme fractionnaire

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

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

$$(PF) \left\{ \begin{array}{l} \text{maximiser } \frac{f(x)}{h(x)} \\ \text{s.c.} \\ g_i(x) \leq 0, i = \overline{1, m} \\ x \in X \end{array} \right.$$

avec :

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

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

L'ensemble des solutions réalisables de (PF) est non vide $S(PF) \neq \emptyset$

$$\exists x \in S(PF) : f(x) \geq 0.$$

Définition 3.1

1. Le problème (PF) de maximisation est appelé problème fractionnaire concave-convexe si f est concave et les fonction h et g_i pour $i = \overline{1, m}$ sont convexes.
2. Le problème (PF) de minimisation est appelé problème fractionnaire convexe-concave si f et les fonctions g_i pour $i = \overline{1, m}$ sont convexes et h concave.[49]

2. Rappel sur l'optimisation convexe

$K \subset \mathbb{R}^n$ est compact si et seulement s'il est fermé et borné.

2.1. Notions sur la convexité

L'importance de la convexité en optimisation peut se résumer dans la phrase de Rockafellar

Ensemble convexe

Définition 3.2

K est un ensemble convexe si et seulement si

$$\forall x, y \in K; \forall \lambda \in [0, 1] \text{ on a } \lambda x + (1 - \lambda)y \in K$$

C-à-d le segment $[x, y]$ est entièrement continue dans K .

Autrement dit, toute combinaison convexe d'éléments de K est dans K . i.e. $\forall x_i \in K, \forall \lambda_i \geq 0, i = \overline{1, n} \text{ avec } \sum_{i=1}^n \lambda_i = 1 \text{ alors } \sum_{i=1}^n \lambda_i x_i \in K$. [16]

Exemple :

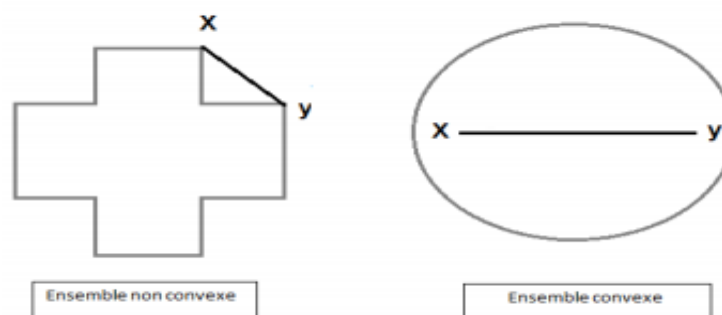


FIGURE 3.1 : Exemple d'un ensemble convexe.



Propriétés

- L'ensemble $\mathbb{R}^n$ est convexe.
- Un singleton $\{a\}$ est convexe.
- Un intervalle $[a, b]$ est convexe dans $\mathbb{R}$.
- Une réunion disjoint d'intervalle de $\mathbb{R}$ n'est pas convexe $]-\infty, 0] \cup [1, +\infty[$.[49]

Fonction convexe



Définition 3.3

On dit qu'une fonction $f : K \rightarrow \mathbb{R}$, $K \subseteq \mathbb{R}^n$ défini sur un ensemble K convexe et non vide, est convexe si elle vérifie :

$$\forall (x, y) \in K, \forall \lambda \in [0, 1] \quad f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y).$$

[16]

Fonction concave



Définition 3.4

On dit qu'une fonction $f : K \rightarrow \mathbb{R}$ définie sur un ensemble convexe non vide K de $\mathbb{R}^n$ est concave si elle vérifie :

$$\forall x, y \in K \quad \lambda \in [0, 1] \quad f(\lambda x + (1 - \lambda)y) \geq \lambda f(x) + (1 - \lambda)f(y).$$

[16]

3. Le modèle générale d'un programme fractionnaire linéaire

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

$$(PFL) \left\{ \begin{array}{l} \text{maximiser } \frac{p^t x + \alpha}{q^t x + \beta} \\ \text{s.c.} \\ Ax \leq b \\ x \geq 0 \end{array} \right.$$

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

4. Résolution d'un programme fractionnaire linéaire

4.1. La résolution graphique

Les programmes fractionnaires linéaires présentent un intérêt particulier mis en évidence par la linéarité des courbes niveaux de leurs fonctions critères. En effet, pour illustrer cet aspect, considérons une $\bar{Z}$ -courbe niveau quelconque de la fonction critère :

$$\bar{Z} = \frac{p^t x + \alpha}{q^t x + \beta}$$

après simplification, nous obtenons :

$$\bar{Z}(q^t x + \beta) = p^t x + \alpha$$

$$(p - \bar{Z}q)^t x = \bar{Z}\beta - \alpha$$

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

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

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

Considérons le programme fractionnaire linéaire suivant :

$$(PFL) \left\{ \begin{array}{l} \text{maximiser } Z = \frac{x_1 + x_2 - 1}{5x_1 + x_2 - 1} \\ \text{s.c. } 5x_1 + 2x_2 \geq 6 \\ x_1 \leq 3 \\ x_2 \leq 3 \\ x_1, x_2 \geq 0 \end{array} \right.$$

dont le graphe est donné par la figure suivante :

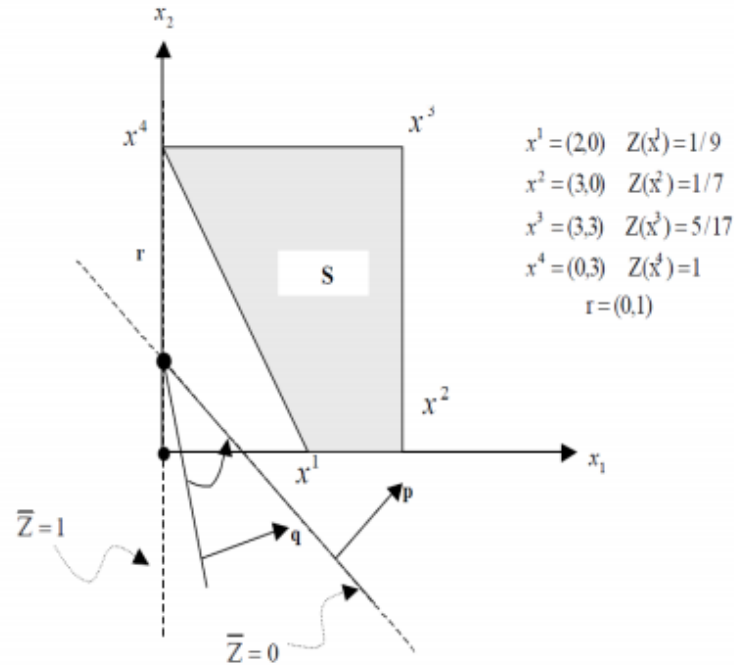


FIGURE 3.2 : Graphe de l'exemple 2.

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

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

donc pour :

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

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

Le problème à quatre points extrêmes x_1, x_2, x_3 et x_4 dont les valeurs du critère sont indiquées sur la **figure 3.1**.

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

4.2. La résolution analytique

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

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

La résolution directe

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

Pour la recherche d'une solution optimale d'un programme hyperbolique en variables continues sans fait aucun changement au programme initial il existe plusieurs algorithmes et parmi les plus récentes, celle de A. Cambini et al. qui permet aussi d'optimiser le problème hyperbolique sur un domaine de solutions réalisables S non borné.

On considère donc le programme hyperbolique continu (PFL) :

$$(PFL) \left\{ \begin{array}{l} \text{maximiser } \frac{p^t x + \alpha}{q^t x + \beta} \\ \text{s.c.} \\ Ax \leq b \\ x \geq 0 \end{array} \right.$$

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

? Théorème

Le point x^0 de S est une solution optimale du problème (PFL) si et seulement si le vecteur gradient réduit $\bar{\gamma} = \bar{\beta} \bar{p}_j - \bar{\alpha} \bar{q}_j$ est tel que $\bar{\gamma}_j \leq 0$ pour tout indice hors base $j \in N_k$.

L'algorithme de A. Cambini génère une séquence finie $x_i, i = \overline{1, l}$ de solutions niveau optimales dont la première est trouvée de la façon suivante :

Résoudre le problème linéaire¹ $P_0 : \min_{x \in S} (q^t x + \beta)$ et considérant x^0 comme une solution optimale. Si x^0 est unique, alors elle est aussi une solution niveau² optimale, sinon résoudre le problème linéaire (P1) :

$$(PFL) \begin{cases} \text{maximiser } p^t x + \alpha \\ \text{s.c.} \\ qx = qx^0 \\ x \in S \end{cases}$$

Si P_1 n'a pas de solutions, alors la valeur de la fonction objectif est infinie ; sinon une solution optimale¹ x de P_1 est aussi une solution niveau optimale.[29]

Algorithm 6 L'Algorithme de A. Cambini et al.

Etape 1 : : Trouver la solution optimale niveau x^1 .

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

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

Etape 2 : : Si $J = \{j / \bar{q}_j > 0\} = \emptyset$, terminer. x^i est une solution optimale du problème (P).

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

Si $\gamma_k > 0$, aller à l'étape 3

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

Etape 3 : : La variable hors base x_{N_k} entre dans la base en moyennant une opération pivot.

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

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

La résolution par paramétrisation

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

Afin de simplifier l'objectif du programme mathématique, un paramètre est introduit

permettant de ramener un programme hyperbolique en un programme paramétré linéaire, tout en gardant l'ensemble des contraintes inchangé. Ainsi le programme obtenu peut être résolu " paramétriquement " : une séquence de résolutions de tels programmes à objectif simplifié engendre une suite de solutions convergeant vers une solution optimale du programme fractionnaire initial.

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

Soit le programme hyperbolique suivant :

$$(PFL) \left\{ \begin{array}{l} \text{maximiser } \frac{p^t x + \alpha}{q^t x + \beta} \\ \text{s.c.} \\ Ax \leq b \\ x \geq 0 \end{array} \right.$$

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

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

$$(PFL_\lambda) \left\{ \begin{array}{l} \text{maximiser } (p^t - \lambda q^t) x + \alpha - \lambda \beta \\ \text{s.c.} \\ Ax \leq b \\ x \geq 0 \end{array} \right.$$

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

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

Dinkelbach a montré que :

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

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

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

La résolution d'un programme équivalent

La transformation du programme fractionnaire en un programme équivalent à objectif non fractionnaire est obtenue par un changement de variables. A l'inverse de l'approche paramétrée, ce changement de variables induit l'ajout d'une contrainte et d'une variable.

Plus précisément, cette transformation, proposée par Charnes et Cooper pour le programme hyperbolique en variables continues, Cette transformation en un programme linéaire équivalent a pour but d'appliquer les algorithmes standards tels que la méthode du Simplexe.

Soit le problème (PFL) suivant :

$$(PFL) \left\{ \begin{array}{l} \text{maximiser } \frac{p^t x + \alpha}{q^t x + \beta} \\ \text{s.c.} \\ Ax \leq b \\ x \geq 0 \end{array} \right.$$

En fait le changement de variable suivant :

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

Alors :

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

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

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

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

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

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

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

5. Programmation fractionnaire linéaire discrète

Les programmes fractionnaires linéaires en nombres entiers ou discrets consistent à optimiser un objectif mis sous la forme d'un rapport de deux fonctions linéaires soumis à

un ensemble de contraintes avec des variables entières. Notons $(PFLNE)$ Le programme fractionnaire linéaire en nombres entiers.

5.1. Forme générale d'un programme fractionnaire linéaire discrète $(PFLNE)$

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

avec $D = \{x \in \mathbb{R}^n / Ax \leq b, x \geq 0\}$, p et x sont des vecteurs de $\mathbb{R}^n$, q un vecteur non nul de $\mathbb{R}^n$, α et β des scalaires, A une $(m \times n)$ -matrice réelle et b un vecteur de $\mathbb{R}^m$. Il est clair que si q est un vecteur nul avec $\beta \neq 0$ alors, $(PFLNE)$ n'est autre qu'un problème classique de la programmation linéaire, de plus, les variables sont astreintes à ne prendre que des valeurs entières.

On suppose que toutes les données du problème sont entières et que l'on dispose initialement d'une solution réalisable entière pour $(PFLNE)$.

$$(PFLNE) \begin{cases} \max Z = \frac{\sum_{j \in N} \bar{p}_j x + \bar{\alpha}}{\sum_{j \in N} \bar{q}_j x + \bar{\beta}} \\ s.c \\ x_B + \bar{A}^N x_N = \bar{b} \\ x_B, x_N \geq 0, x \text{ entier.} \end{cases}$$

5.2. Relaxation



Définition 3.5

Soit la forme générale du $(PFLNE)$ suivante :

$$(PFLNE) \begin{cases} \max (\min) Z = \frac{p^t x + \alpha}{q^t x + \beta} \\ s.c. \\ Ax \leq b \\ \forall x_j \in x, x_j \geq 0 \quad (j = \overline{1, n}) \\ x_j \in \mathbb{Z}^+ \text{ pour } (j = \overline{1, k}) \text{ avec } k \leq n \end{cases}$$

Notez bien que si on remplace les dernies restrictions par $x_j \geq 0$ pour $j = \overline{1, k}$ avec $(k \leq n)$ on aura un programme linéaire qui sera une relaxation du problème original. Ce fait est très important. ça implique qu'on peut résoudre la relaxation par des méthodes qu'on a déjà considérés.

5.3. Méthodes de résolution des programmes fractionnaires linéaires discrets

Méthode de Granot

[17]

Principe de l'algorithme

L'algorithme s'applique directement au programme fractionnaire linéaire en nombres entiers (hyperbolique discrète) (*PFLNE*).

Considérons le problème suivant :

$$(PFLNE) \begin{cases} \max Z = \frac{p^t x + \alpha}{q^t x + \beta} \\ s.c \\ A x \leq b \\ x \geq 0, x \text{ entier.} \end{cases}$$

dans cette méthode, la structure originale des contraintes est maintenues et les opérations sont réalisées dans un tableau de simplexe augmenté qui comprend m et 3 lignes supplémentaires. les m premières lignes correspondant aux contraintes d'origine, les lignes (m+1), (m+2) correspondant aux numérateur et dénominateur de la fonction objectif fractionnaire du problème (*PFLNE*) et la ligne (m+3) correspond au vecteur gradient réduit γ_j . A chaque itération de l'algorithme, les (m+2) lignes sont modifiées à travers les opérations ordinaires de pivot, par contre la dernière ligne est modifiée selon la formule du gradient réduit.

– Une fois les valeurs de gradient réduit calculées par la formule suivante :

$$\bar{\gamma}_j = \bar{\beta} \bar{p}_j - \bar{\alpha} \bar{q}_j, \quad \forall j \in \{1, \dots, N\} \quad (N \text{ Ensemble des indices hors base})$$

– Faite le teste suivant :

✓ si $\gamma_j \leq 0, \quad \forall j \in N$ Alors terminer et la solution obtenue est optimal.

✓ sinon $\exists$ un indice k, $k \in N$ pour les quel $\gamma_k > 0$, soit θ_k tel que

$$\theta_k = \min \left\{ \frac{b_i}{A_{ik}}, A_{ik} > 0 \right\}$$

Alors toute ligne v pour laquelle $[b_v/A_{vk}] \leq \theta_k$, peut servir comme une ligne source pour générer une coupe de Gomory de la forme :

$$s + \sum_{j \in N} \lfloor \frac{A_{vj}}{A_{vk}} \rfloor x_j = \lfloor \frac{b_v}{A_{vk}} \rfloor, \quad s \geq 0.$$

Cette coupe peut être rajoutée au problème initial et servir comme ligne pivot, avec la k^{ieme} colonne comme une colonne pivot. Puisque la valeur du pivot dans ce cas est $[A_{vk}/A_{vk}] = 1$, les nouveaux coefficients obtenus après l'opération de pivot usuelle sont tous entiers.

La coupe de Dantzig

Etant donné le programme relaxé (*PFL*) de (*PFLNE*) c'est à dire sans les contraintes d'intégrité et supposons le vecteur seconde membre b des contraintes est un vecteur positif mais non entier. La coupe de Dantzig définit par :

$$\sum_{j \in N} x_j \geq 1, \text{ Où } N \text{ étant l'ensemble des indices hors base}$$

a été proposée sur la base que dans le problème relaxé (*PFL*), une des variable hors base est strictement positive et dans le problème en variables entières (*PFLNE*) cette variable est supérieure ou égale à 1. Cette coupe n'est utilisée qu'en cas de nécessité vu ces conditions d'utilisation et son inefficacité (son rôle est de tronquer juste un seul point du domaine réalisable). Une des coupes qui nous utiliserons dans notre travail est une généralisation de cette coupes de Dantzig appelée coupe de Dantzig améliorée. Elle ressemble quelque peu à cette coupe mais est plus efficace. Elle permet de réduire plus fortement le domaine d'admissibilité.

Les coupes de Gomory

L'algorithme utilisant les coupes de Gomory consiste d'abord a résoudre le problème relaxé (*PFL*) par l'algorithme du simplexe, considérons le tableau obtenu a l'optimum, parmi les variables de base B choisissons une variable x_r , $r \in B$, fractionnaire (s'il n'y en a pas, on est à l'optimum). La contrainte correspondante à x_r (i^{eme} ligne de tableau optimal) s'écrit :

$$x_r + \sum_{j \in N} A_{ij} x_j = b_i$$

où N est l'ensemble des indices des variables hors-base.

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

$$x_r + \sum_{j \in N} \langle A_{ij} \rangle x_j \geq \langle b_i \rangle$$

où $\langle r \rangle = r - [r]$ désigne la partie fractionnaire du nombre réel r . cette contrainte est introduite dans le tableau simplexe optimal et le nouveau problème formé peut être résolu en utilisant la méthode duale du simplexe. Après un nombre fini d'itérations, ou bien une solution optimale entière de problème fractionnaire discret est obtenue, ou bien le problème devient impossible. Dans la pratique, les coupes fractionnaires de Gomory convergent très lentement.

Méthode par séparation et évaluation

C'est une des méthodes les plus utilisées pour la résolution de (*PFLNE*). Elle combine de façon élégante la méthode primale-duale du simplexe et le principe de séparation et évaluation dans une arborescence de recherche. Elle repose sur les étapes suivantes :

Au nœud 0 de l'arborescence :

On pose $U = \underline{z}$ un minorant de la valeur optimale de la fonction objectif et on résout le problème relaxé (*PFL*) avec la méthode primale du simplexe.

Séparation :

Si la solution optimale, disons $\hat{x}$, a ses coordonnées $\hat{x}_j \in N$, pour tout $j = 1, \dots, n$, terminer. Cette solution est optimale pour (*PFLNE*).

Sinon, choisir une variable x_j qui n'est pas entière. Partitionner l'ensemble des solutions en deux sous-ensembles en ajoutant l'une ou l'autre des contraintes :

$$x_j \leq \lfloor \hat{x}_j \rfloor \quad \text{ou} \quad x_j \geq \lceil \hat{x}_j \rceil$$

où $\lfloor a \rfloor$ indique la partie entière du nombre réel a , et $\lceil a \rceil = \lfloor a \rfloor + 1$

Deux nouveaux nœuds sont alors créés. Notons que $\hat{x}$ viole à la fois ces deux contraintes et que le coût optimal de chacun des deux sous-problèmes sera inférieur à celui de (*PFL*) :

Evaluation :

Résoudre le programme (*PFL*) avec la nouvelle contrainte en utilisant la méthode duale du simplexe.

Test de stérilisation :

1. Si la nouvelle solution optimale x est telle que $z_{PFL}(x) \leq U$
2. Si le dual du simplexe implique qu'il n'y a pas de solutions réalisables.
3. Si la solution optimale x est réalisable pour le problème initial (*PFLNE*), c'est-à-dire entière. Si de plus, $z_{PFL} > U$, alors poser $U = z_{PFL}$ et garder la solution x comme meilleure solution de la branche courante.

A un nœud k :

Faire : l'évaluation, le test de stérilisation et éventuellement, la séparation.

Plusieurs choix sont possibles pour le choix d'un nœud à évaluer : profondeur d'abord, meilleur d'abord et largeur d'abord. Le plus utilisé est profondeur d'abord pour trouver rapidement une solution réalisable et ne pas trop encombrer la mémoire du calculateur. La méthode s'arrête lorsque tous les nœuds pendants de l'arborescence (nœuds de degré 1) sont sondés (le test de stérilisation est positif).

Conclusion

Dans ce chapitre nous avons fait un rappel sur quelques notions mathématiques, telle que les définitions, les notions de convexité, et les fonctions convexe généralisées. nous avons

donné un aperçu sur la programmation fractionnaire linéaire telle que le modèle générale, les domaines d'applications, et les quatre grandes méthodes de résolution : graphique, directe, par paramétrisation, et la résolution avec un programme équivalent. nous avons aussi défini la programmation fractionnaire linéaire en nombres entiers (*PFLNE*) et ses méthodes de résolution.

Chapitre 4

Notions de base de la logique floue

Introduction

Les mathématiques du flou, de plus en plus désignées sous le terme générique de théorie du flou, regroupent plusieurs théories qui sont des généralisations ou des extensions de leurs homologues classiques : la théorie des sous-ensembles flous étend celle des ensembles, la logique floue étend la logique binaire, la théorie des quantités floues étend celle des nombres et intervalles, la théorie des possibilités étend celle des probabilités, plus généralement la théorie des mesures floues étend celle de la mesure. elles ont toutes pour objectif de proposer des concepts, des techniques et des méthodes formellement rigoureuses pour recueillir, représenter et traiter des connaissances et des données floues c'est-à-dire contenant de l'imprécision, de l'incertitude ou de la subjectivité ; ces trois facettes principales du flou étant souvent coexistantes. D'autres synonymes, tels que connaissance mal spécifiée, mal décrite, imparfaite, vague, qualitative, linguistique, partielle, incomplète, approximative ou approchée, recouvrent cette même acception du flou.

1. Idées cle sur la théorie du flou [ZALILA, 2003]

Le Flou fédère trois facettes souvent coexistantes : Imprécision, Incertitude, Subjectivité. Dans la vie réelle, nous prenons régulièrement de bonnes décisions à partir d'informations floues. La théorie du flou est un cadre formel qui permet la modélisation et le traitement rigoureux de ce type d'information.

La connaissance floue est riche en information : la transformer dès le début du traitement en une connaissance nette induit un biais qui se répercute inmanquablement sur la qualité de la décision. Il est au contraire préférable de maintenir le flou tout au long du traitement et de ne trancher qu'en fin de processus (passage de la décision floue vers la décision nette).

La logique floue permet une modélisation graduelle et nuancée de la connaissance experte et propose un mode de raisonnement approché et par analogie. En appréhendant à la

fois l'imprécision et l'incertitude, elle autorise la conception de systèmes d'aide à la décision plus efficaces que les systèmes experts classiques : un expert sera d'autant plus certain de ses affirmations qu'on l'autorise à être imprécis et sera d'autant plus incertain qu'on l'oblige à être précis. Elle offre aussi une approche alternative performante pour la modélisation de processus complexes non linéaires.

L'arithmétique floue permet une modélisation et un traitement des quantités numériques imprécises. Elle autorise la conception de modèles analytiques prévisionnels plus fidèles à la réalité.

La mesure de possibilité remplace celle de probabilité lorsque le décideur doit évaluer l'occurrence d'un événement, sur lequel il possède peu de données historiques ou des données de mauvaise qualité. Ceci apparaît notamment dans les problèmes de décision multicritère ou de sûreté de fonctionnement, lorsque l'agent décisionnaire fait appel à des informations provenant de capteurs humains (jugement, avis d'expert). Elle autorise notamment l'estimation de l'occurrence d'événements imprécis.

De cette manière, la théorie du flou étend la palette des méthodes de modélisation, de prédiction et d'évaluation, tout en simplifiant la formalisation de problèmes complexes.

2. Les nombres et les ensembles flous

2.1. Définitions



Définition 4.1

En général, en logique floue, on rattache toujours la variable (A, B ou C) à une donnée extérieure (une information ou une mesure, par exemple). Ce rattachement s'effectue par l'intermédiaire de ce que l'on appelle *une fonction d'appartenance*. Cette fonction permet de faire correspondre une information à une variable logique continue variant de 0 à 1.[44]

Définition 4.2

Soit X un ensemble ordinaire. Un ensemble flou $\tilde{A}$ de X est défini par sa fonction d'appartenance $\mu_{\tilde{A}}$.

$$\mu_{\tilde{A}} : X \rightarrow [0, 1]$$

$$x \rightarrow \mu_{\tilde{A}}(x) \in [0, 1]$$

La valeur de $\mu_{\tilde{A}}(x)$ représente *le degré d'appartenance* de x dans X et est interprété comme la mesure dans laquelle x appartient à $\tilde{A}$. Par conséquent, plus la valeur de $\mu_{\tilde{A}}(x)$ est proche à 1, plus x appartient à $\tilde{A}$.

Ainsi un sous-ensemble flou $\tilde{A}$ dans X peut être représenté par un ensemble de couples ordonnés :

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) \mid x \in X\}$$

Un ensemble ordinaire A de X peut être considéré comme un ensemble flou en X avec une fonction d'appartenance identique à sa fonction caractéristique $\mu_{\tilde{A}}$ donnée par :

$$\mu_{\tilde{A}(x)} = \begin{cases} 1 & \text{si } x \in A \\ 0 & \text{si } x \notin A \end{cases} \quad (4.1)$$

Définition 4.3

Soit $\tilde{A}$ un ensemble flou sur X , alors *le support* de $\tilde{A}$, noté $\text{supp}(\tilde{A})$, est l'ensemble ordinaire donné par :

$$\text{supp}(\tilde{A}) = \{x \in X \mid \mu_{\tilde{A}}(x) > 0\}. [21]$$

Définition 4.4

Le noyau d'un sous-ensemble flou A dans un univers du discours X est le sous-ensemble des éléments de X pour lesquels la fonction d'appartenance vaut 1. C'est l'ensemble des points qui appartiennent intégralement à :

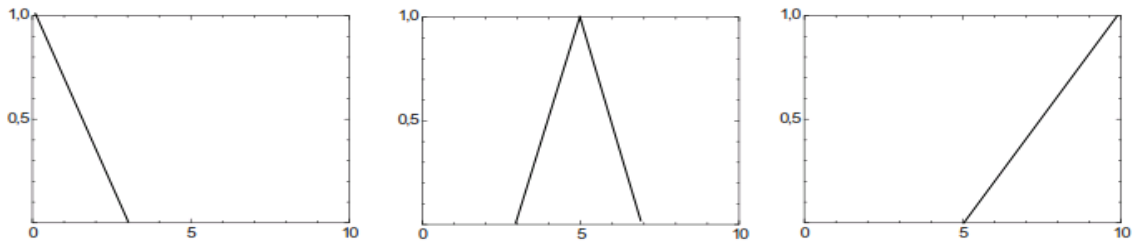
$$N(\tilde{A}) = \{x \mid \mu_{\tilde{A}}(x) = 1\}. [21]$$

Définition 4.5

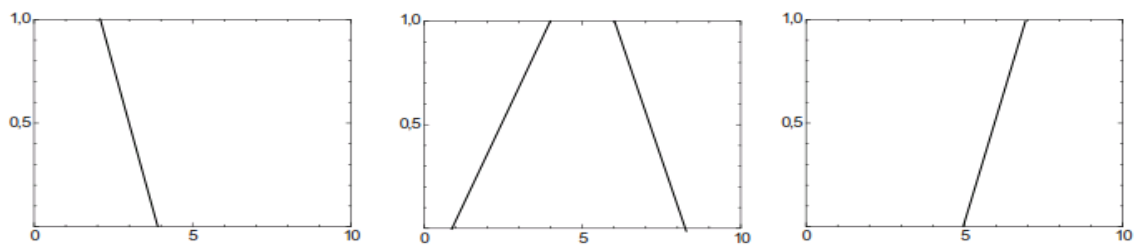
Soit $\tilde{A}$ un ensemble flou sur X . La hauteur de $\tilde{A}$, notée $\text{htr}(\tilde{A})$, est définie comme

$$\text{h}(\tilde{A}) = \sup_{x \in X} \mu_{\tilde{A}}(x)$$

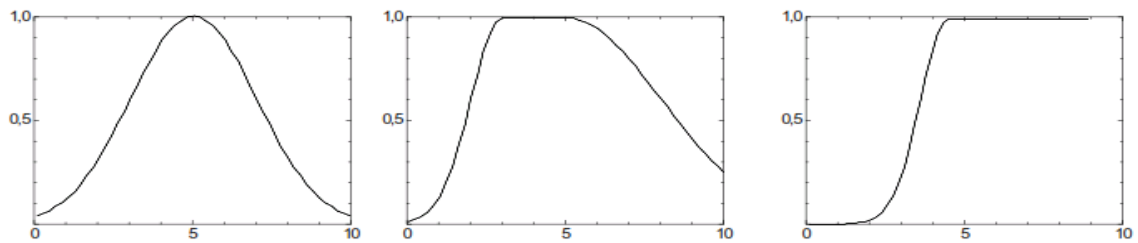
Si $\text{htr}(\tilde{A}) = 1$, alors l'ensemble flou $\tilde{A}$ est appelé un ensemble flou normal, sinon il est dit subnormal.[31]



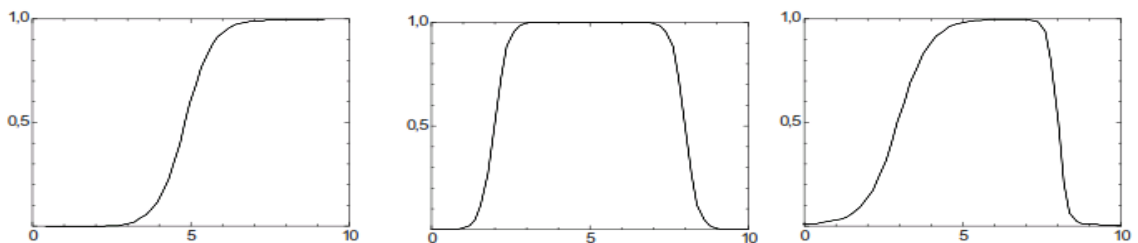
(a) Fonctions d'appartenance triangulaires



(b) Fonctions d'appartenance trapézoïdales



(c) Fonctions d'appartenance Gaussiennes



(d) Fonctions d'appartenance sigmoïdes

FIGURE 4.1 : Les fonctions d'appartenances les plus utilisées

2.2. Opérations sur les ensembles flous

Les opérations sur les ensembles flous sont généralement des extensions des opérations connues sur les ensembles classiques (égalité, réunion, intersection, complément, etc.). Elles s'appliquent d'ailleurs aux ensembles classiques lorsque les fonctions d'appartenance se réduisent à des fonctions caractéristiques.

Définition 4.6

Soient $\tilde{A}$ et $\tilde{B}$ deux ensembles flous sur X . L'ensemble flou $\tilde{A}$ est appelé *un sous ensemble* de $\tilde{B}$ (ou $\tilde{A}$ est contenu dans $\tilde{B}$), noté $\tilde{A} \subset \tilde{B}$, si la condition suivante est vérifiée :

$$\mu_{\tilde{A}}(x) \leq \mu_{\tilde{B}}(x) \quad \forall x \in X. [31]$$

Définition 4.7

Soient $\tilde{A}$ et $\tilde{B}$ deux ensembles flous sur X . $\tilde{A}$ et $\tilde{B}$ sont *égaux*, noté par $\tilde{A} = \tilde{B}$ si les deux conditions suivantes sont vérifiées :

$$\tilde{A} \subset \tilde{B} \quad \text{et} \quad \tilde{B} \subset \tilde{A}. [21]$$

Définition 4.8

Soient $\tilde{A}$ et $\tilde{B}$ deux ensembles flous sur X . L'*union* de $\tilde{A}$ et $\tilde{B}$, notée $\tilde{A} \cup \tilde{B}$, est un ensemble flou sur X dont la fonction d'appartenance est donnée par :

$$\begin{aligned} \forall x \in X \quad \mu_{\tilde{A} \cup \tilde{B}}(x) &= \max(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)) \\ &= \mu_{\tilde{A}}(x) \vee \mu_{\tilde{B}}(x). [21] \end{aligned}$$

Définition 4.9

Soient $\tilde{A}$ et $\tilde{B}$ deux ensembles flous sur X . L'*intersection* de $\tilde{A}$ et $\tilde{B}$, notée $\tilde{A} \cap \tilde{B}$, est un ensemble flou sur X dont la fonction d'appartenance est donnée par :

$$\begin{aligned} \forall x \in X \quad \mu_{\tilde{A} \cap \tilde{B}}(x) &= \min(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)) \\ &= \mu_{\tilde{A}}(x) \wedge \mu_{\tilde{B}}(x). [21] \end{aligned}$$

Définition 4.10

Soit $\tilde{A}$ un ensemble flou sur X . Le *complémentaire* de $\tilde{A}$, noté $\tilde{A}^c$, est un ensemble flou sur X dont la fonction d'appartenance est donnée par :

$$\forall x \in X, \quad \mu_{\tilde{A}^c}(x) = 1 - \mu_{\tilde{A}}(x). [31]$$

2.3. Nombres flous

✚ Définition 4.11

Un ensemble flou sur $\mathbb{R}$ est appelé un *nombre flou* s'il satisfait les conditions suivantes

- 1 - $\tilde{A}$ est normal, i.e., $\exists x_0 \in \mathbb{R} \mu_{\tilde{A}}(x_0) = 1$.
- 2 - A_α est un intervalle fermé pour chaque $\alpha \in (0, 1]$, noté par $[A_\alpha^L, A_\alpha^R]$.
- 3 - Le $\text{supp } \tilde{A}$ est borné.[21]

✚ Définition 4.12

Un *nombre flou trapézoïdal (TrFN)* $\tilde{A}$ sur $\mathbb{R}$ est défini par un quadruplet (a, b, c, d) , avec la fonction d'appartenance $\mu_{\tilde{A}}(x)$ donnée par (4.2) et illustrée par la figure 4.1

$$\mu_{\tilde{A}}(x) = \begin{cases} \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & b \leq x \leq c \\ \frac{d-x}{d-c} & c \leq x \leq d \\ 0 & \text{sinon} \end{cases}$$

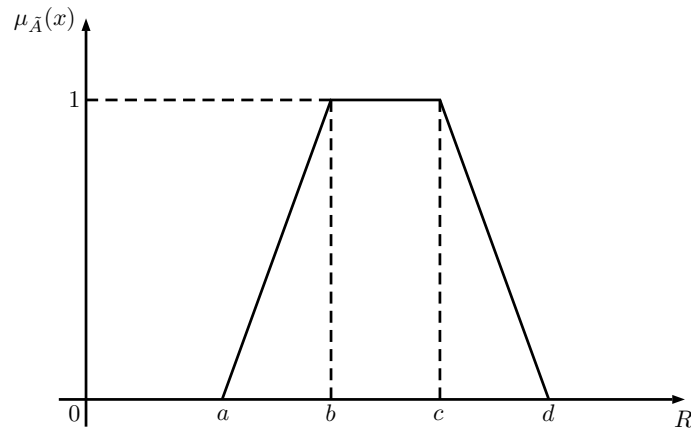


FIGURE 4.2 : Fonction d'appartenance trapézoïdale (TrFN)

§ Définition 4.13

[21] Un *nombre flou triangulaire (TFN)* $\tilde{A}$ sur R est un cas particulier de *(TrFN)* où ($b = c$), qui est défini par un triplet (a, b, c) avec la fonction d'appartenance $\mu_{\tilde{A}}(x)$ donnée par (4.2) et illustrée par la figure 4.2

$$\mu_{\tilde{A}}(x) = \begin{cases} \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{c-x}{c-b} & b \leq x \leq c \\ 0 & \text{sinon} \end{cases}$$

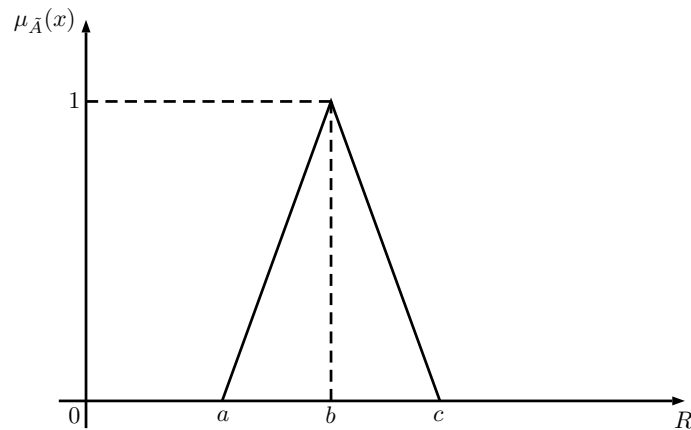


FIGURE 4.3 : Fonction d'appartenance triangulaire (TFN)

§ Définition 4.14

Pour toute valeur α de l'intervalle $[0,1]$ on appelle *la coupe α* d'un sous-ensemble flou de X , le sous-ensemble noté $\tilde{A}_\alpha$ des éléments de X pour lesquels la fonction d'appartenance est supérieure ou égale à α :

$$\tilde{A}_\alpha = \{x \in X \mid \mu_{\tilde{A}}(x) \geq \alpha\}$$

Si nous choisissons $\alpha = 0$, alors $\tilde{A}_\alpha$ est l'univers du discours X . Si nous choisissons $\alpha = 1$, alors A est le noyau de $\tilde{A}(N)$.

2.4. Défuzzification

La *défuzzification* est essentielle d'une part pour ramener toutes les unités de mesures vers une unité de mesure commune (i.e., la normalisation).

Dans notre cas, la défuzzification de chaque (TrFN) se fait en utilisant une coupe α .

? Théorème de décomposition

[31] Le *théorème de décomposition* montre qu'il est possible de reconstituer un sous-ensemble flou à partir de ses coupes α . En fait, le théorème permet de définir la fonction d'appartenance μ_A à partir des fonctions caractéristiques des coupes α :

$$\mu_{\tilde{A}}(x) = \sup_{\alpha \in [0,1]} \alpha \chi_{\tilde{A}_\alpha(x)}$$

la coupe α définit un intervalle de confiance $[B^L, B^U]$ de niveau $\alpha \in [0,1]$ comme le montre la figure 4.3

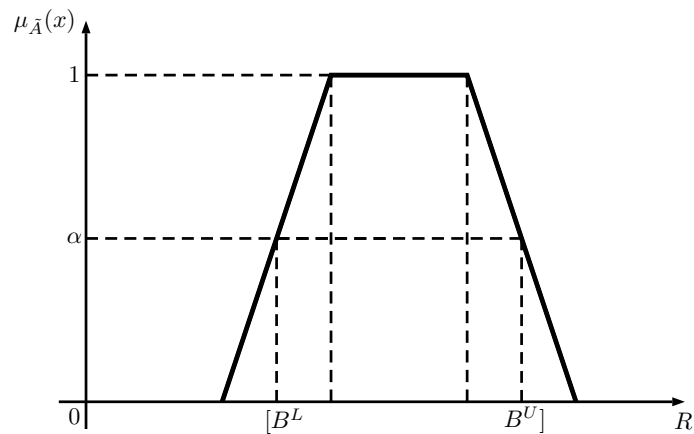


FIGURE 4.4 : La coupe α

Les deux bornes supérieure et inférieure sont définies comme suit :

$$B^L = (b - a)\alpha + a \quad (4.2)$$

$$B^U = d - (d - c)\alpha \quad (4.3)$$

Conclusion

Nous avons vu dans ce chapitre les concepts de base des ensembles flous, nombres flous et comment faire la défuzzification.

Chapitre 5

Résolution d'un problème multi-objectif fractionnaire linéaire flou en nombres entiers

Introduction

Dans ce chapitre, un algorithme suggéré pour résoudre le problème multi-objectif fractionnaire linéaire flou en nombres entiers (*PMFLFNE*) est décrit, telle que les coefficients flous sont dans le second membre des contraintes et peuvent être caractérisés par des nombres flous par une fonction d'appartenance trapézoïdale.

1. Formulation du problème

La forme d'un problème multi-objectif fractionnaire linéaire flou en nombres entiers (*PMFLFNE*) est donnée comme suit :

$$(PMFLFNE) = \left\{ \begin{array}{l} \max Z_1(x) = \frac{c^1 x + \alpha^1}{d^1 x + \beta^1} \\ \max Z_2(x) = \frac{c^2 x + \alpha^2}{d^2 x + \beta^2} \\ \cdot \\ \cdot \\ \cdot \\ \max Z_k(x) = \frac{c^k x + \alpha^k}{d^k x + \beta^k} \\ s.c \\ x \in M(\tilde{b}) \end{array} \right. \quad (5.1)$$

Où $k \geq 2$, c^r , d^r sont $1 * n$ vecteurs

$M(\tilde{b}) = \{x \in \mathbb{R}^n / \sum_{i=1}^n A_i x \leq \tilde{b}, x \geq 0 \text{ et entier}\}$,

A est une $(m * n)$ matrice réelle, $\tilde{b} \in \mathbb{R}^m$ est un vecteur de paramètres flous et nous supposons qu'ils sont donnés par des nombres flous estimés à partir des informations

fournies par le décideur. Tout au long de ce mémoire, nous supposons que $M(\tilde{b})$ est non vide.

nous avons choisi parmi les divers types de nombres flous les trapézoïdaux (4.11), car ces nombres sont assez simples (linéaires par morceaux), ce qui les rend particulièrement utiles en résolution de notre problème.

Le vecteur des paramètres flous $\tilde{b}$ dans le problème ($PMFLFNE$) est un vecteur de nombres flous dont la fonction d'appartenance est notée $\mu_{\tilde{b}}(b)$.

À l'aide de la coupe (4.13) α nous avons transformé le problème ($PMFLFNE$) vers un problème ($PMFLNE$) puis le résoudre.

pour un certain degré $\alpha = \alpha^* = [0.1]$, estimé par le décideur. le problème peut être exprimer par un problème α -multi-objectif fractionnaire linéaire non flou en nombre entier ($\alpha - PMFLNE$).

$$(\alpha - PMFLNE) = \begin{cases} \max Z_1(x) = \frac{c^1 x + \alpha^1}{d^1 x + \beta^1} \\ \max Z_2(x) = \frac{c^2 x + \alpha^2}{d^2 x + \beta^2} \\ \cdot \\ \cdot \\ \cdot \\ \max Z_k(x) = \frac{c^k x + \alpha^k}{d^k x + \beta^k} \\ s.c \\ x \in M(b) \end{cases} \quad (5.2)$$

où $M(b) = \{x \in \mathbb{R}^n / Ax \leq b, x \geq 0 \text{ et entier} / b \in L_\alpha(\tilde{b})\}$,

tq : $L_\alpha(\tilde{b}) = \{b \in \mathbb{R}^m / \mu_{\tilde{b}}(b) \geq \alpha\}$ (4.13)

une fonction d'appartenance du vecteur floue $\tilde{b}$ est formule comme suit :

$$\mu_{\tilde{b}}(b) = \begin{cases} 0, & b \leq b_1 \\ [\frac{b-b_1}{b_2-b_1}] & b_1 \leq b \leq b_2 \\ 1 & b_2 \leq b \leq b_3 \\ [\frac{b_4-b}{b_4-b_3}] & b_3 \leq b \leq b_4 \\ 0, & \text{autrement} \end{cases}$$

Défuzzification

soit un nombre Trapzoïdale $\tilde{T}$ défini par $\tilde{T} = (b_1, b_2, b_3, b_4)$, d'après les équations (4,2) et (4,3) les bornes inférieure et supérieure respectivement sont définies comme suit :

$$\underline{b} = \alpha(b_2 - b_1) + b_1, \quad \overline{b} = b_4 - \alpha(b_4 - b_3)$$

Alors l'intervall flou de $\tilde{T} : \tilde{T}^\alpha = [\underline{b}, \overline{b}]$.

le problème ($\alpha - PMFLNE$) peut être réécrit comme suit :

$$(\alpha - PMFLNE) = \begin{cases} \max Z_1(x) = \frac{c^1 x + a^1}{d^1 x + \beta^1} \\ \max Z_2(x) = \frac{c^2 x + a^2}{d^2 x + \beta^2} \\ \cdot \\ \cdot \\ \cdot \\ \max Z_k(x) = \frac{c^k x + a^k}{d^k x + \beta^k} \\ s.c \\ x \in M(b) \end{cases} \quad (5.3)$$

où $\{x \in \mathbb{R}^n / Ax \leq b, x \geq 0, \text{ et entier}\}$
et $\underline{b}^\alpha \leq b_i \leq \bar{b}^\alpha$

2. Modification de la méthode "Isbell-Marlow"

Une modification est effectuée sur la méthode Isbell-Marlow pour résoudre le problème fractionnaire linéaire en nombres entiers ($PFLNE$) suivant :

$$(PFLNE) = \begin{cases} \max(f(x)) = \frac{c^t x + c_0}{d^t x + d_0} \\ s.c \\ s = \{x \in \mathbb{R}^n / Ax \leq b, x \geq 0 \text{ et entier}\} \end{cases} \quad (5.4)$$

où $c, d \in \mathbb{R}^n, c_0, d_0 \in \mathbb{R}, A$ est $(m * n)$ matrice réelle, $b \in \mathbb{R}^m, d \neq 0, d_0 \neq 0, d^t x + d_0 > 0$ pour tous $x \in s$, et le rang. $A=m < n$.

? Théorème

[35] x^* est une optimale pour le problème (PFL) si elle est optimale pour le problème linéaire :

$$(PLNE) = \begin{cases} \max_{x \in S_I} \phi^*(x) = (c^t x + c_0) - f(x^*)(d^t x + d_0) \\ s.c \ s = \{x \in \mathbb{R}^n / Ax \leq b, x \geq 0 \text{ et entier}\} \end{cases} \quad (5.5)$$

L'algorithme modifié pour résoudre le problème de programmation fractionnaire linéaire en nombres entiers ($PFLNE$), formulé ci-dessus peut être résumé dans les étapes suivantes :

Algorithm 7 Algorithme I

Etape 0 : Initialisation $i = 0$. Trouver une solution réalisable entière $x^i = x^0$ de base pour le problème (PFLNE).

Etape 1 : Résoudre le problème linéaire en nombres entiers.

$$\max_{x \in S} \phi^i(x) = (c^t x + c_0) - f(x^i)(d^t x + d_0)$$

pour obtenir la solution x^{i+1} .

Etape 2 : Si x^{i+1} est un point entier, passer à l'étape 3. Sinon, utiliser la Méthode Branch and Bound.

Etape 3 : Si $\phi^i(x^{i+1}) = 0$, alors, arrêter et x^{i+1} est une solution optimale entière pour le problème (PFLNE). Sinon, $i = i + 1$ et aller à l'étape 1.

Vu la nature du problème ($\alpha - PMFLNE$) on peut dire que la technique adaptée de scalarisation de ce problème est d'utiliser la méthode ϵ -contrainte.

$$P_s(\epsilon) = \begin{cases} \max Z_s(x) = \frac{c^s x + \alpha^s}{d^s x + \beta^s} \\ Z_r(x) = \frac{c^r x + \alpha^r}{d^r x + \beta^r} \geq \epsilon_r, r = 1, 2, 3, \dots, k; r \neq s, \\ Ax \leq b \\ \underline{b}^\alpha \leq b_i \leq \bar{b}^\alpha, i = (1, 2, \dots, m) \\ x \geq 0 \text{ et entier.} \end{cases} \quad (5.6)$$

Si x^* est une solution optimale entière unique de problème $P_s(\epsilon^*)$, alors x^* devient une α -solution efficace au problème de ($\alpha - PMFLNE$).

3. Algorithme Solution proposée pour la résolution de problèmes (α - PMFLNE)

L'algorithme générant une α - solution efficace au problème (α -PMFLNE) est présenté dans les étapes suivantes.

Algorithm 8 Algorithme II

Etape 1 : Commencer avec un premier α -niveau définies avec $\alpha = \alpha^* = 0$.

Etape 2 : Déterminer les points $(b_1, b_2, \dots, b_m)$ pour le vecteur des paramètres flous $\tilde{b}$ dans le problème (α -PMFLNE) pour susciter une fonction d'appartenance $\mu_{\tilde{b}}(b)$.

Etape 3 : Convertir le problème (α -PMFLNE) dans sa version non floue.

Etape 4 : Formuler le problème fractionnaire linéaire en nombre entier avec une seule fonction objectif $P_s(\epsilon)$.

Etape 5 : Résoudre le problème fractionnaire linéaire en nombre entier k individuel P_r ($r = 1, 2, \dots, k$) en utilisant la méthode "Isbell-Marlow" modifié (Alg. I).

Etape 6 : Construire le tableau des gains et déterminer n_r, M_r (le plus petit et le plus grand nombres dans la colonne dans le tableau des gains).

Etape 7 : Déterminer ϵ'_i

Etape 8 : Trouver $D = \{\epsilon \in R^{k-1} | n_r \leq \epsilon_r < M_r, r = 1, 2, \dots, k; r \neq s\}$ où t est le nombre de toutes les partitions de l'intervalle $[n_r, M_r]$.

Etape 9 : Choisir $\epsilon_r^* \in D$ et résoudre le problème fractionnaire linéaire en nombres entiers $P_s(\epsilon^*)$. on trouve sa solution optimale entière x^* avec α - niveau optimal.

Etape 10 : Régler $\alpha = (\alpha^* + \epsilon) \in [0, 1]$ et passez à l'étape 2.

Etape 11 : Répéter la procédure ci-dessus jusqu'à ce que l'intervalle $[0, 1]$ est complètement épuisé. Puis, arrêter.



Remarque :



1. Le pas de α c'est un choix de décideur.
2. Pour chaque calcul sur un niveau de α , l'intervalle de "b" va diminuer sachant que $\mu_{\tilde{b}}(b) \geq \alpha$.
3. Si l'intervalle trouvé contient 2 entiers on prend le min.

4. Un exemple illustratif

Nous considérons le problème multi-objectif fractionnaire linéaire flou en nombres entiers (PMFLFNE) suivant :

$$(PMFLFNE) = \begin{cases} \max Z_1(x) = \frac{x_1-4}{-x_2+3} \\ \max Z_2(x) = \frac{-x_1+4}{x_2+1} \\ \max Z_3(x) = -x_1 + x_2 \\ s.c \\ -x_1 + 4x_2 \leq \tilde{b}_1 \\ 2x_1 - x_2 \leq \tilde{b}_2 \\ x_1, x_2 \geq 0, \text{ et entier} \end{cases} \quad (5.7)$$

Les paramètres flous $\tilde{b}_1, \tilde{b}_2$ sont donnés par les nombres flous suivants énumérés dans le tableau ci-dessous :

	a ₁	a ₂	a ₃	a ₄
b ₁	0	0.2	0.3	0.6
b ₂	8	8.2	8.3	0.6

Voici les fonctions d'appartenances pour convertir au-dessus de la logique floue Le problème (PMFLFNE) dans sa version non floue.

$$\mu_{\tilde{b}_1}(b_1) = \begin{cases} \left[\frac{b_1-a_1}{a_2-a_1} \right] & a_1 \leq b_1 \leq a_2 \\ 1 & a_2 \leq b_1 \leq a_3 \\ \left[\frac{a_4-b_1}{a_4-a_3} \right] & a_3 \leq b_1 \leq a_4 \\ 0, & \text{sinon.} \end{cases}$$

$$\mu_{\tilde{b}_2}(b_2) = \begin{cases} \left[\frac{b_2-a_1}{a_2-a_1} \right] & a_1 \leq b_2 \leq a_2 \\ 1 & a_2 \leq b_2 \leq a_3 \\ \left[\frac{a_4-b_1}{a_4-a_3} \right] & a_3 \leq b_2 \leq a_4 \\ 0, & \text{sinon.} \end{cases}$$

posons $\alpha = \alpha^* = 0$ dans cet exemple le nombre trapzoïdal flou est :

pour $\tilde{b}_1$: $T_1 = (0, 0.2, 0.3, 0.6)$.

pour $\tilde{b}_2$: $T_2 = (8, 8.2, 8.3, 8.6)$.

D'où :

$$1. \left[\frac{b_1 - a_1}{a_2 - a_1} \right] = \alpha \Rightarrow \underline{b}_1 = 0 \text{ (borne inf);}$$

$$\left[\frac{a_4 - b_1}{a_4 - a_3} \right] = \alpha \Rightarrow \overline{b}_1 = 0.6 \text{ (borne sup);}$$

$$\underline{b}_1 \leq b_1 \leq \overline{b}_1 \Rightarrow 0 \leq b_1 \leq 0.6 .$$

$$2. \left[\frac{b_2 - a_1}{a_2 - a_1} \right] = \alpha \Rightarrow b_1 = 8 \text{ (borne inf);}$$

$$\left[\frac{a_4 - b_2}{a_4 - a_3} \right] = \alpha \Rightarrow \overline{b}_2 = 8.6 \text{ (borne sup);}$$

$$\underline{b}_2 \leq b_2 \leq \overline{b}_2 \Rightarrow 8 \leq b_1 \leq 8.6 .$$

le problème flou (*PMFLFNE*) est converti à la version non floue ($\alpha - PMFLNE$) comme dans la forme suivante :

$$(\alpha - PMFLFNE) = \begin{cases} \max Z_1(x) = \frac{x_1 - 4}{-x_2 + 3} \\ \max Z_2(x) = \frac{-x_1 + 4}{x_2 + 1} \\ \max Z_3(x) = -x_1 + x_2 \\ s.c \\ -x_1 + 4x_2 \leq b_1 \\ 2x_1 - x_2 \leq b_2 \\ 0 \leq b_1 \leq 0.6 \\ 8 \leq b_2 \leq 8.6 \\ x_1, x_2 \geq 0, \text{ et entier} \end{cases} \quad (5.8)$$

nous avons choisi $b_1 = 0$ et $b_2 = 8$ (car ils sont les petits entiers dans ces intervalles).

en utilisant l'algorithme "Isbell-Marlow" modifié (Alg. I) pour résoudre les problème (*PFLNE*).

Tout d'abord, nous résolvons le problème (P_1) :

$$(P_1) = \begin{cases} \max Z_1(x) = \frac{x_1 - 4}{-x_2 + 3} \\ s.c \\ -x_1 + 4x_2 \leq 0 \\ 2x_1 - x_2 \leq 8 \\ x_1, x_2 \geq 0, \text{ et entier} \end{cases} \quad (5.9)$$

nous commençons par une solution initiale entière de base $x^0 = (0,0)$, nous avons :

$$(P'_1) = \begin{cases} \max \phi^0(x) = [(x_1 - 4) - z_1(x^0)(-x_2 + 3)] \\ \quad \quad \quad = x_1 - \frac{4}{3}x_2 \\ s.c \\ -x_1 + 4x_2 \leq 0 \\ 2x_1 - x_2 \leq 8 \\ x_1, x_2 \geq 0, \text{ et entier} \end{cases} \quad (5.10)$$

Solution de base :

	x_1	x_2	x_3	x_4	b
x_3	-1	4	1	0	0
x_4	2	-1	0	1	8
-Z	1	-4/3	0	0	0

En résolvant le programme linéaire avec la méthode du simplexe, on avait obtenu le tableau optimal suivant :

Tableau optimal au P'_1 :

	x_1	x_2	x_3	x_4	b
x_2	0	1	2/7	1/7	8/7
x_1	1	0	1/7	8/14	32/7
-Z	0	0	-1/7	-8/14	-4/7

La solution optimale est $(\frac{8}{7}, \frac{32}{7})$

cette solution n'est pas entière donc on applique Branch and Bound avec $4 \leq x_1 \leq 5$

La séparation :

En intégrant la contrainte $x_1 \leq 4$ à la solution optimale précédente nous obtenons le sous problème (SP'_1)

$$(SP'_1) = \begin{cases} \max \phi^0(x) = [(x_1 - 4) - z_1(x^0)(-x_2 + 3)] \\ \quad \quad \quad = x_1 - \frac{4}{3}x_2 \\ s.c \\ -x_1 + 4x_2 \leq 0 \\ 2x_1 - x_2 \leq 8 \\ x_1 \leq 4 \\ x_1, x_2 \geq 0, \text{ et entier} \end{cases} \quad (5.11)$$

L'évaluation :

Réolvons le sous problème (SP'_1), On doit ajouter la contrainte x_1 au (SP'_1); sous forme d'équation, on a $x_1 + x_5 = 4$ On obtient alors le tableau suivant en ajoutant cette contrainte au tableau précédent.

	x_1	x_2	x_3	x_4	x_4	b
x_2	0	1	$2/7$	$1/7$	0	$8/7$
x_1	1	0	$1/7$	$8/14$	0	$32/7$
x_5	1	0	0	0	1	4
-Z	0	0	$-1/7$	$-8/14$	0	$-4/7$

On doit toute fois modifier ce tableau puisque x_1 est une variable dans la base et il doit lui correspondre un vecteur unitaire dans le tableau.

	x_1	x_2	x_3	x_4	x_4	b
x_2	0	1	$2/7$	$1/7$	0	$8/7$
x_1	1	0	$1/7$	$8/14$	0	$32/7$
x_5	0	0	$-1/7$	$-8/7$	1	$-4/7$
-Z	0	0	$-1/7$	$-8/14$	0	$-4/7$

Le tableau optimal du SP'_1 :

	x_1	x_2	x_3	x_4	x_4	b
x_2	0	1	0	-1	2	0
x_1	1	0	0	0	0	4
x_5	0	0	1	4	-7	4
-Z	0	0	0	0	-1	0

Tous les $c_j \leq 0$, et la solution optimale entière a été trouvé (4.0) avec $\phi^0(x) = 0$, arrêter.

$$(P_2) = \begin{cases} \max Z_2(x) = \frac{-x_1 + 4}{x_2 + 1} \\ s.c \\ -x_1 + 4x_2 \leq 0 \\ 2x_1 - x_2 \leq 8 \\ x_1, x_2 \geq 0, \text{ et entier} \end{cases} \quad (5.12)$$

La solution entière de base initiale $x^0 = (0,0)$ est utilisé pour résoudre :

$$(P'_2) = \begin{cases} \max \phi^0(x) = [(-x_1 + 4) - z_2(x^0)(x_2 + 1)] \\ \quad \quad \quad = -x_1 - 4x_2 \\ s.c \\ -x_1 + 4x_2 \leq 0 \\ 2x_1 - x_2 \leq 8 \\ x_1, x_2 \geq 0, \text{ et entier} \end{cases} \quad (5.13)$$

Tous les $c_j \leq 0 \ j=\overline{1,2}$, donc la solution optimale entière pour le problème (P'_2) est $(0,0)$ avec $\phi^0(x) = 0$

Enfin, nous résolvons :

$$(P_3) = \begin{cases} \max Z_3(x) = -x_1 + x_2 \\ s.c \\ -x_1 + 4x_2 \leq 0 \\ 2x_1 - x_2 \leq 8 \\ x_1, x_2 \geq 0, \text{ et entier} \end{cases} \quad (5.14)$$

Solution de base :

	x_1	x_2	x_3	x_4	b
x_3	-1	4	1	0	0
x_4	2	-1	0	1	8
-Z	-1	1	0	0	0

Tableau optimal du P_3 :

	x_1	x_2	x_3	x_4	b
x_2	-1/4	1	1/4	0	0
x_4	7/4	0	1/4	1	0
-Z	-3/4	0	-1/4	0	0

Tous les $c_j \leq 0 \ j=\overline{1,2}$, donc la solution optimale entière trouvée est $(0,0)$ et puisque x_1 et x_2 sont entiers, arrêter.

Maintenant, la méthode " ϵ -contrainte" est utilisé pour convertir le problème multi-objectif fractionnaire linéaire non floue en nombres entiers ($\alpha - PMFLNE$) à un problème mono-objectif fractionnaire linéaire en nombres entiers ($PFLNE$) et le tableau des gains est donné ci-dessous.

	$Z_1(x) = \frac{x_1-4}{-x_2+3}$	$Z_2(x) = \frac{-x_1+4}{x_2+1}$	$Z_3 = -x_1 + x_2$
$x_{z_1(4,0)}$	0	0	-4
$x_{z_2(0,0)}$	$\frac{-4}{3}$	4	0
$x_{z_3(0,0)}$	$\frac{-4}{3}$	4	0
max	0	4	0
min	$\frac{-4}{3}$	0	-4
ϵ	$\frac{-4}{3} \leq \epsilon_1 \leq 0$	$0 \leq \epsilon_2 \leq 4$	$-4 \leq \epsilon_3 \leq 0$

Soit $\epsilon_2 = 1$, $\epsilon_3 = -3$, alors nous avons :

Solution de base :

$$(P_1(\epsilon^*)) = \begin{cases} \max Z_1(x) = \frac{x_1-4}{-x_2+3} \\ s.c \\ -x_1 + 4x_2 \leq 0 \\ 2x_1 - x_2 \leq 8 \\ \frac{-x_1+4}{x_2+1} \geq 1 \\ -x_1 + x_2 \geq -3 \\ x_1, x_2 \geq 0, \text{ et entier} \end{cases} \quad (5.15)$$

En utilisant l'algorithme "Isbell-Marlow" modifié (Alg. I) et en laissant la solution entière de base initiale $x^0 = (0,0)$, le problème sera :

$$(P'_1(\epsilon^*)) = \begin{cases} \max \omega^0(x) = [(x_1 - 4) - z_1(x^0)(-x_2 + 3)] \\ \quad \quad \quad = x_1 - \frac{4}{3}x_2 \\ s.c \\ -x_1 + 4x_2 \leq 0 \\ 2x_1 - x_2 \leq 8 \\ x_1 + x_2 \leq 3 \\ x_1 - x_2 \leq 3 \\ x_1, x_2 \geq 0, \text{ et entier} \end{cases} \quad (5.16)$$

Solution de base :

	x_1	x_2	x_3	x_4	x_5	x_6	b
x_3	-1	4	1	0	0	0	0
x_4	2	-1	0	1	0	0	8
x_5	1	1	0	0	1	0	3
x_6	1	-1	0	0	0	1	3
-Z	1	-4/3	0	0	0	0	-3

Tableau optimal de $(P_1(\epsilon^*))$

	x_1	x_2	x_3	x_4	x_5	x_6	b
x_3	0	5	1	0	1	0	3
x_4	0	-3	0	1	-2	0	2
x_1	1	1	0	0	1	0	3
x_6	0	-2	0	0	-1	1	0
-Z	0	-7/3	0	0	-1	0	-3

La solution entière est trouvée (3,0) avec $\omega^0(x) = 3$. puisque $\omega^0(x) \neq 0$, puis utiliser $x^* = (3,0)$ comme une solution entière initiale de base pour résoudre le problème suivant :

$$(P_1(\epsilon^*)) = \begin{cases} \max \omega^0(x) = [(x_1 - 4) - z_1(x^1)(-x_2 + 3)] \\ \quad = x_1 - \frac{1}{3}x_2 - 3 \\ \text{s.c} \\ -x_1 + 4x_2 \leq 0 \\ 2x_1 - x_2 \leq 8 \\ x_1 + x_2 \leq 3 \\ x_1 - x_2 \leq 3 \\ x_1, x_2 \geq 0, \text{ et entier} \end{cases} \quad (5.17)$$

Soluion de base :

	x_1	x_2	x_3	x_4	x_5	x_6	b
x_3	-1	4	1	0	0	0	0
x_4	2	-1	0	1	0	0	8
x_1	1	1	0	0	1	0	3
-Z	1	-4/3	0	0	0	0	0

Tableau optimal de $P_1(\epsilon^*)$

	x_1	x_2	x_3	x_4	x_5	x_6	b
x_3	0	5	1	0	1	0	3
x_4	0	-3	0	1	-2	0	2
x_1	1	1	0	0	1	0	3
x_6	0	-2	0	0	-1	1	0
-Z	0	-7/3	0	0	-1	0	0

la solution optimale entière est trouvée (3,0) avec $\omega^1(x)=0$. puisque $\omega^1(x) = 0$. alors arrêter. Alors l'ensemble des solution efficaces entières du problème ($\alpha - PMFLNE$) telle que ($\alpha=0$) est Eef = {(4.0) , (0.0) , (3.0) }

Pour toutes valeurs de $\alpha \in]0.1]$ les b_i ne peuvent être entières. donc la résolution de notre problème est faite seulement pour $\alpha = 0$.

Conclusion

Dans ce chapitre nous avons présenté un algorithme qui a été proposé pour résoudre le problème de programmation multi-objectif fractionnaire linéaire flou en nombres entiers ($PMFLFNE$). Certains concepts flous ont été donnés pour convertir le problème ($PMFLFNE$) à un problème non flou ($PMFLNE$), la version est une modification de l'algorithme "Isbell-Marlow" (Alg. I) qui a été utilisé pour achever le processus de solution.

Chapitre 6

Implémentation, Résultats et Discussion

Introduction

Dans ce chapitre, nous aboutissons à l'étape finale de notre travail, à savoir l'élaboration d'un logiciel aussi convivial que possible. nous expliquerons son fonctionnement afin de faciliter son utilisation et de mettre en application la méthode de résolution proposée, nous avons alors implémenté cette méthode en utilisant Matlab de la société The Mathwork.

1. Choix du langage

Le choix s'est porté sur l'emploi du langage du logiciel Matlab R2015a, car il répond aux critères suivants :

-La maniabilité du langage : constitué d'un ensemble de possibilités faisant en sorte que le programmeur travaille avec aisance.

-Le bagage du langage : il contient une interface graphique et des fonctions prédéfinies telle que :

intlinprog :

permet de résoudre un problème linéaire en nombres entiers.

La fonction length :

Cette fonction prend comme entrée un vecteur et retourne sa taille. Son code Matlab est : `k=length(vec)`, telle que " vec " est un vecteur, et " k " est la taille qu'on cherche.

La fonction size :

Cette fonction prend comme entrée une matrice et retourne sa taille. Le code Matlab est : `[MN]=size (A)`

A est une matrice.

M nombre de ligne.

N nombre de colonne.

La fonction zeros :

Cette fonction génère une matrice de taille $M \times N$.

le code Matlab : `A= zeros(M,N)`.

La fonction floor :

retourne la première valeur entière inférieure à x, dans la direction de $-\infty$

le code Matlab : `a=floor(b)`.

b : est la valeur qu'on cherche a trouvé leur partie entière inferieur. a : la valeur obtenue.

1.1. Présentation de langage de programmation : MATLAB

MATLAB est un logiciel parfaitement dédié à la résolution de problèmes d'analyse numérique ou de traitement du signal. permet d'effectuer des calculs matriciels ou de visualiser les résultats sous forme graphique. la formulation des problèmes s'apparente à la formulation mathématique des problèmes à résoudre. l'utilisation de ce logiciel consiste à lancer des lignes de commandes, qui peuvent le plus souvent ressembler à la programmation en C. le nom MATLAB vient de MATrix LABoratory, les éléments de donnée de base manipulés par MATLAB étant des matrices (mais peuvent évidemment se réduire à des vecteurs et des scalaires) qui ne nécessitent ni dimensionnement ni déclaration de type.

2. les fonctions programmées

2.1. La fonction "Isbelle-Marlow" :

Cette fonction représente la méthode de "Isbelle-Marlow", elle prend comme paramètre la fonction a modifie "Z" et la valeur initiale de "x" et elle retourne la la fonction "Z" modifie.

le code Matlab : `function [Z]=IsaMar( Z,x )`.

2.2. La fonction de resolution de chaque (PL) :

Cette fonction prend comme paramètre le vecteur des paramètre de la fonction objectif "Z", la matrice "C" associe aux contraintes, et la valeur de second membre " b", Elle retourne " fval1 " la valeur de la fonction objectif et le vecteur " sol1 " qui représente la solution optimale de chaque.

le code Matlab : `function [ sol1,fval1 ] = mintprog( Z, C, B )`.

2.3. La fonction qui regroupe les deux fonctions " ε -contrainte" et " α -coupe" ainsi que le résultat finale de programme :

cette fonction fait la resolution de probleme pour chaque niveau $\alpha \in [0, 1]$ afin de trouver la valeur déterministe de "b" pour trouver les solutions optimales de chaque (PL) de problème initiale pour enfin trouver l'intervalle de " ε " puis faire la résolution de " ε -contrainte" qui résoudre le problème initiale dont le resultat est une solution optimale pour chaque " ε " choisi dans l'intervalle trouvé.

le code Matlab : `function [ ] = master(Z, C )`.

3. Exemple concret

Le problème à résoudre ici est un problème muti-objectif fractionnaire linéaire flou en nombres entiers suivant :

$$(\text{PMFLFNE}) = \begin{cases} \max Z_1(x) = \frac{x_1 - 4}{-x_2 + 3} \\ \max Z_2(x) = \frac{-x_1 + 4}{x_2 + 1} \\ \max Z_3(x) = -x_1 + x_2 \\ s.c \\ -x_1 + 4x_2 \leq \tilde{b}_1 \\ 2x_1 - x_2 \leq \tilde{b}_2 \\ x_1, x_2 \geq 0, \text{ et entier} \end{cases} \quad (6.1)$$

avec deux fonctions objectifs fractionnaires linéaires et une fonction linéaire. en declarant la matrice Z comme suit :

$$Z = \begin{pmatrix} 1 & 0 & -4 \\ -1 & 0 & 3 \\ -1 & 0 & 4 \\ 0 & 1 & 1 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

la matrice des contraintes :

$$C = \begin{pmatrix} -1 & 4 \\ 2 & -1 \end{pmatrix}$$

les vecteurs des nombres flous Trapzoïaux pour b_1 et b_2 :

$$b_1 = \begin{pmatrix} 0 & 0.2 & 0.3 & 0.6 \end{pmatrix}$$

$$b_2 = \begin{pmatrix} 8 & 8.2 & 8.3 & 8.6 \end{pmatrix}$$

4. Application de l'exemple sur Matlab

nous faisons la résolution sous Matlab par les fonctions cite au-dessus, en déclarant le problème comme suit :

```
Z=[1 0 -4 ; 0 -1 3 ; -1 0 4 ; 0 1 1 ; -1 1 0 ; 0 0 1 ]
```

```
Z =
```

```

    1    0   -4
    0   -1    3
   -1    0    4
    0    1    1
   -1    1    0
    0    0    1
```

```
C=[-1 4 ; 2 -1]
```

```
C =
```

```

   -1    4
    2   -1
```

nous donnons le vecteur flou de "b" sous forme d'un (*TrFN*) puis nous précisant le pas de α ainsi que le choix de la fonction fixé pour la résolution de la fonction ε -contrainte.

```

Donnez moi la matrice des flou SVP [0 0.2 0.3 0.6 ; 8 8.2 8.3 8.6]
Donnez moi le pas de alpha SVP 0.1
Donnez moi la fonction objectif à fixé SVP 1
```

pour le résultat finale nous avons les deux premiers colonnes sont les " ε " choisi pour la résolution de ε -contrainte. les deux derniers colonnes représente la solution optimale pour chaque " ε " choisi.

```

Le vecteur b = 0 0.6 pour alpha = 0
Le vecteur b = 8 8.6 pour alpha = 0
Les solutions entières efficaces de (alpha-PM FLNE)
La solution optimale du chaque problème linéaire : 4 0
La solution optimale du chaque problème linéaire : 0 0
La solution optimale du chaque problème linéaire : 0 0
```

La fonction objectif fixé est : 1

ans =

	'E'	'E'	'X'	'X'
[0]	[-4]	[4]	[0]	
[0]	[-3]	[3]	[0]	
[0]	[-2]	[2]	[0]	
[0]	[-1]	[1]	[0]	
[0]	[0]	[0]	[0]	
[1]	[-4]	[3]	[0]	
[1]	[-3]	[3]	[0]	
[1]	[-2]	[2]	[0]	
[1]	[-1]	[1]	[0]	
[1]	[0]	[0]	[0]	
[2]	[-4]	[2]	[0]	
[2]	[-3]	[2]	[0]	
[2]	[-2]	[2]	[0]	
[2]	[-1]	[1]	[0]	
[2]	[0]	[0]	[0]	
[3]	[-4]	[1]	[0]	
[3]	[-3]	[1]	[0]	
[3]	[-2]	[1]	[0]	
[3]	[-1]	[1]	[0]	
[3]	[0]	[0]	[0]	
[4]	[-4]	[0]	[0]	
[4]	[-3]	[0]	[0]	
[4]	[-2]	[0]	[0]	
[4]	[-1]	[0]	[0]	
[4]	[0]	[0]	[0]	

Conclusion

Dans ce chapitre nous avons commencé par décrire le logiciel Matlab puisque nous l'avons choisi pour la programmation de notre méthode, puis nous avons appliqué notre programme sur un exemple, enfin nous avons testé le programme sur tous l'intervalle de ε .

Conclusion générale

Dans ce mémoire, nous nous sommes intéressés à l'étude d'un problème d'optimisation multi-objectif fractionnaire linéaire flou en nombre entier. Les fonctions objectifs sont fractionnaires linéaires et les paramètres flous sont associés aux second membre des contraintes. Ils sont caractérisés par des nombres flous trapézoïdales. Dans ce sens nous avons deux objectifs à atteindre :

Le premier est de rendre notre problème multi-objectif fractionnaire linéaire flou en un problème multi-objectif linéaire non flou. Pour cela, nous avons transformé le problème flou en un problème non flou, en se basant sur des fonctions d'appartenance prédéfinies ou en obtenant les "b" dans des intervalles. Puis nous avons effectué une modification de telle sorte que nous ayons transformé notre problème fractionnaire en un problème linéaire à l'aide de la méthode "Isbelle-Merlow".

Deuxième objectif est de rendre le problème, d'un problème multi-objectif en un problème mono-objectif en utilisant la méthode d'/ε -Contrainte puis le résoudre.

Enfin, nous nous sommes attachés à décrire l'application construite, ainsi une vue d'ensemble sur le logiciel de programmation que nous avons utilisé qui est Matlab 2015.

Bibliographie

- [1] ASLI, L. . Approche hybride pour les problèmes d'optimisation combinatoire multiobjectif : cas des problèmes de type sac-à-dos 2010.
- [2] BARICHARD, V. . Approches hybrides pour les problèmes multiobjectifs 2003.
- [3] BECTOR, C. R. ,Suresh Chandra , Fuzzy Mathematical Programming and Fuzzy Matrix Games Springer-Verlag Berlin Heidelberg, 2005 ,169,.
- [4] BENLAHRACHE, N. . Optimisation Multi-Objectif Pour l'Alignement Multiple de Séquences 2007.
- [5] BEZOUÏ, M. . Support de Cours :programmation linéaire *Bèjaia, 2008-2009*.
- [6] C.GUÉRARD. « Programmation Linéaire » *Paris, Eyrolles 1976*.
- [7] CHAABANE, D. Contribution à l'optimisation multicritère en variables discrètes Mons 2007,.
- [8] CHANKONG, V., AND Y.Y. Haimes, Multiobjective Decision-Making : Theory and Methodology, System Science and Engineering. *North Holland, 1983*.
- [9] CHERGUI, M.-A., AND MOULAÛ, M. An Exact Method for a Discrete Multiobjective Linear Fractional Optimization, *Applied Mathematics and Decision Sciences1-12 (2008)*.
- [10] COELLO COELLO. C. A. .An updated survuey of g.a based multiobjective optimization techniques *Technical report Lania-RI-98-08 Laboratorio Nacional Avanzada, Xalapa, Veracruz, Mexico (Deceumber 1998)*.
- [11] COLLETTE, Y., AND P. . Siarry, Optimisation multiobjectif, Edition Eyrolles 75240 Paris Cedex 05, France, 2002.
- [12] DANTZIG, G. Linear programming and extensions *Princeton University Press, Prince- ton, N.J., 1963*.
- [13] ET I, O. . Optimisation multicritère : Fondements et Concepts *PhD thesis, Université de Grenoble, 1998*.
- [14] ET L.HAMADACHE, A. B. Interprétation économique de la dualité et l'analyse de post-optimale Boumerdes 2013,.
- [15] ET N.ZEMOURENE, A. Programmation fractionnaire linéaire en nombres entiers : méthode de Granot. Boumerdes 2015,.
- [16] FABIAN, B. Modèles de Recherche Opérationnelle, bastin@iro.umontreal.ca Université de Montréal 2010.

- [17] FRENK, J. B. G., AND S. Schaible, Fractional Programming : Introduction and Applications, in Encyclopedia of Optimization *Kluwer Academic Publishers, The Netherlands, 2001.*
- [18] GRANOT, D., AND F. Granot, On Integer and Mixed Integer Fractional Programming *Annals of Discrete Mathematics, 1 (1977), 221-231.*
- [19] HACHEMI, N. E. .L'analyse postoptimale. www.almohandiss.com 17 avril 2011..
- [20] HAIMES. Y. Y ; W. A. HALL ; H. T. FREEDMAN. .A multiobjective optimization in water resources systems *Elsevier Scientific (1975).*
- [21] IGOULALENE, I. Développement d'une approche floue multicritère d'aide à la coordination des décideurs pour la résolution des problèmes de sélection dans les chaines logistiques 2014,.
- [22] I.M, S.-M. . Fractional Programming : Theory, Methods and Applications, *Kluwer Academic Publishers, 1997.*
- [23] I.M, S.-M. . A fifth bibliography of fractional programming, *Optimization 45, 1-4, 343-367, 1999.*
- [24] I.M, S.-M. . A sixth bibliography of fractional programming, *Optimization 55, 4, 405-428, 2006.*
- [25] ISBELL, J., AND MARLOW, W. Attrition Games, *Naval Research Logistics Quarterly Vol. 3 (1956), 71-94.*
- [26] ISBELL, J., AND MARLOW, W. Attrition Games, *Naval Research Logistics Quarterly Vol. 3 (1956), 71-94.*
- [27] IZAZ ULLAH KHAN, TAHIR AHMAD, N. M. A Simplified Novel Technique for Solving Fully Fuzzy Linear Programming Problems *Journal of Optimization Theory and Applications* .November 2013,159, pg 536-546,.
- [28] MAHDI.K, M. L'optimisation multiobjectif et l'informatique quantique Constantine,.
- [29] MARTOS, B. "Hyperbolic programming", *Naval Res. Logist. Quart* 1964, 11, pp. 135-155,.
- [30] MINASIAN, I. S. Fractional Programming : Theory ; Methods and Applications, *Mathematics and Its Applications 409 (1997).*
- [31] MOKEDDEM, D.
- [32] MOULAÏ, M. Mèmoire de doctorat : OPTIMISATION MULTICRITERE FRACTIONNAIRE LINEAIRE EN NOMBRES ENTIERS 2002,.
- [33] MOULAÏ, M. Optimisation multicritère fractionnaire linéaire en nombres entiers 2002,.
- [34] M.SAKAROVITCH. graphes et programmation linéaire paris,.
- [35] O. M. SAAD, A. A., AND F, E. Abdellah, On Solving Single - Objective Fuzzy Integer Linear Fractional Programs, *Applied Mathematics Information Sciences to appear during summer (2010).*

- [36] REARDON. B. J. Fuzzy logic us niched pareto multi-objective genetic algorithm optimization : Part i : Schaffer's problem. *Technical report la- ur-97-3675, Los Alamos National Laboratory, 1997.*
- [37] SAAD, O. M. n Stability of Proper Efficient Solutions in Multiobjective Fractional Programming under Fuzziness, *Mathematical and Computer Modeling 45 (2007), 221-231.*
- [38] SAKAWA. M; H. yano. An interactive fuzzy satisficing method for multiobjective linear programming problems with fuzzy parametres, vol *volume 28. Fuzzy sets and Systems, 1988.*
- [39] SCHAIBLE, S., AND SHI, J. Recent Developments in Fractional Programming : Single and Max-Min Case, *A Gary Anderson Graduate School of Management, California, 2004.*
- [40] SESHAN, C. R., AND V.G. Tikekar, Algorithms for Integer Fractional Programming *Journal of the Indian Institute of Science, 62 (1980), 9-16.*
- [41] SIARRY. P. .La méthode du recuit simulé en èlectronique *Habilitation report, Paris-sud University(Orsay) (1994).*
- [42] SIMONNARD, M. « Programmation Linéaire » *Dunod 1962.*
- [43] STEUER, R. E. Multiple Criteria Optimization : Theory ; Computation and Application *Willey, New York, 1986.*
- [44] SUGENO, M., K. G. Structure identification of fuzzy model. *Fuzzy Sets Syst 28 : 15–33, 1988.*
- [45] TEGHEM, J. « Programmation Linéaire » , *OM Partners.*
- [46] VERMA H.C., B., AND M.C. . Puri Ranking in integer linear fractional programming problems, *Zeitschrift f' er Operations Research- Methods and Models of Operations Research 34, 325-334, 1990.*
- [47] ZADEH, L. A. Soft computing and fuzzy logic. *IEEE Software 11(6), pp 48-56, 1994.*
- [48] ZADEH, L. A. le concept d'une variable linguistique et son application à rapprocher le raisonnement (1975).
- [49] ZERDANI, O. Thèse de doctorat : L'optimisation non linéaire multi objectif 2013,.
- [50] ZOUAOUI, S. .Mèmoire de magistère : Approche multicritère pour le problème de sac à dos. *USTHB,Algerie,alger 2011.*

Bibliographie

- [1] ASLI, L. . Approche hybride pour les problèmes d'optimisation combinatoire multiobjectif : cas des problèmes de type sac-à-dos 2010.
- [2] BARICHARD, V. . Approches hybrides pour les problèmes multiobjectifs 2003.
- [3] BECTOR, C. R. ,Suresh Chandra , Fuzzy Mathematical Programming and Fuzzy Matrix Games Springer-Verlag Berlin Heidelberg, 2005 ,169,.
- [4] BENLAHRACHE, N. . Optimisation Multi-Objectif Pour l'Alignement Multiple de Séquences 2007.
- [5] BEZOUÏ, M. . Support de Cours :programmation linéaire *Bèjaia, 2008-2009*.
- [6] C.GUÉRARD. « Programmation Linéaire » *Paris, Eyrolles 1976*.
- [7] CHAABANE, D. Contribution à l'optimisation multicritère en variables discrètes Mons 2007,.
- [8] CHANKONG, V., AND Y.Y. Haimes, Multiobjective Decision-Making : Theory and Methodology, System Science and Engineering. *North Holland, 1983*.
- [9] CHERGUI, M.-A., AND MOULAÛ, M. An Exact Method for a Discrete Multiobjective Linear Fractional Optimization, *Applied Mathematics and Decision Sciences1-12 (2008)*.
- [10] COELLO COELLO. C. A. .An updated survuey of g.a based multiobjective optimization techniques *Technical report Lania-RI-98-08 Laboratorio Nacional Avanzada, Xalapa, Veracruz, Mexico (Deceumber 1998)*.
- [11] COLLETTE, Y., AND P. . Siarry, Optimisation multiobjectif, Edition Eyrolles 75240 Paris Cedex 05, France, 2002.
- [12] DANTZIG, G. Linear programming and extensions *Princeton University Press, Prince- ton, N.J., 1963*.
- [13] ET I, O. . Optimisation multicritère : Fondements et Concepts *PhD thesis, Université de Grenoble, 1998*.
- [14] ET L.HAMADACHE, A. B. Interprétation économique de la dualité et l'analyse de post-optimale Boumerdes 2013,.
- [15] ET N.ZEMOURENE, A. Programmation fractionnaire linéaire en nombres entiers : méthode de Granot. Boumerdes 2015,.
- [16] FABIAN, B. Modèles de Recherche Opérationnelle, bastin@iro.umontreal.ca Université de Montréal 2010.

- [17] FRENK, J. B. G., AND S. Schaible, Fractional Programming : Introduction and Applications, in Encyclopedia of Optimization *Kluwer Academic Publishers, The Netherlands, 2001.*
- [18] GRANOT, D., AND F. Granot, On Integer and Mixed Integer Fractional Programming *Annals of Discrete Mathematics, 1 (1977), 221-231.*
- [19] HACHEMI, N. E. .L'analyse postoptimale. www.almohandiss.com 17 avril 2011..
- [20] HAIMES. Y. Y ; W. A. HALL ; H. T. FREEDMAN. .A multiobjective optimization in water resources systems *Elsevier Scientific (1975).*
- [21] IGOULALENE, I. Développement d'une approche floue multicritère d'aide à la coordination des décideurs pour la résolution des problèmes de sélection dans les chaines logistiques 2014,.
- [22] I.M, S.-M. . Fractional Programming : Theory, Methods and Applications, *Kluwer Academic Publishers, 1997.*
- [23] I.M, S.-M. . A fifth bibliography of fractional programming, *Optimization 45, 1-4, 343-367, 1999.*
- [24] I.M, S.-M. . A sixth bibliography of fractional programming, *Optimization 55, 4, 405-428, 2006.*
- [25] ISBELL, J., AND MARLOW, W. Attrition Games, *Naval Research Logistics Quarterly Vol. 3 (1956), 71-94.*
- [26] ISBELL, J., AND MARLOW, W. Attrition Games, *Naval Research Logistics Quarterly Vol. 3 (1956), 71-94.*
- [27] IZAZ ULLAH KHAN, TAHIR AHMAD, N. M. A Simplified Novel Technique for Solving Fully Fuzzy Linear Programming Problems *Journal of Optimization Theory and Applications* .November 2013,159, pg 536-546,.
- [28] MAHDI.K, M. L'optimisation multiobjectif et l'informatique quantique Constantine,.
- [29] MARTOS, B. "Hyperbolic programming", *Naval Res. Logist. Quart* 1964, 11, pp. 135-155,.
- [30] MINASIAN, I. S. Fractional Programming : Theory ; Methods and Applications, *Mathematics and Its Applications 409 (1997).*
- [31] MOKEDDEM, D.
- [32] MOULAÏ, M. Mèmoire de doctorat : OPTIMISATION MULTICRITERE FRACTIONNAIRE LINEAIRE EN NOMBRES ENTIERS 2002,.
- [33] MOULAÏ, M. Optimisation multicritère fractionnaire linéaire en nombres entiers 2002,.
- [34] M.SAKAROVITCH. graphes et programmation linéaire paris,.
- [35] O. M. SAAD, A. A., AND F, E. Abdellah, On Solving Single - Objective Fuzzy Integer Linear Fractional Programs, *Applied Mathematics Information Sciences to appear during summer (2010).*

- [36] REARDON. B. J. Fuzzy logic us niched pareto multi-objective genetic algorithm optimization : Part i : Schaffer's problem. *Technical report la- ur-97-3675, Los Alamos National Laboratory, 1997.*
- [37] SAAD, O. M. n Stability of Proper Efficient Solutions in Multiobjective Fractional Programming under Fuzziness, *Mathematical and Computer Modeling 45 (2007), 221-231.*
- [38] SAKAWA. M; H. yano. An interactive fuzzy satisficing method for multiobjective linear programming problems with fuzzy parametres, vol *volume 28. Fuzzy sets and Systems, 1988.*
- [39] SCHAIBLE, S., AND SHI, J. Recent Developments in Fractional Programming : Single and Max-Min Case, *A Gary Anderson Graduate School of Management, California, 2004.*
- [40] SESHAN, C. R., AND V.G. Tikekar, Algorithms for Integer Fractional Programming *Journal of the Indian Institute of Science, 62 (1980), 9-16.*
- [41] SIARRY. P. .La méthode du recuit simulè en èlectronique *Habilitation report, Paris-sud University(Orsay) (1994).*
- [42] SIMONNARD, M. « Programmation Linéaire » *Dunod 1962.*
- [43] STEUER, R. E. Multiple Criteria Optimization : Theory ; Computation and Application *Willey, New York, 1986.*
- [44] SUGENO, M., K. G. Structure identification of fuzzy model. *Fuzzy Sets Syst 28 : 15–33, 1988.*
- [45] TEGHEM, J. « Programmation Linéaire » , *OM Partners.*
- [46] VERMA H.C., B., AND M.C. . Puri Ranking in integer linear fractional programming problems, *Zeitschrift f' er Operations Research- Methods and Models of Operations Research 34, 325-334, 1990.*
- [47] ZADEH, L. A. Soft computing and fuzzy logic. *IEEE Software 11(6), pp 48-56, 1994.*
- [48] ZADEH, L. A. le concept d'une variable linguistique et son application à rapprocher le raisonnement (1975).
- [49] ZERDANI, O. Thèse de doctorat : L'optimisation non linèaire multi objectif 2013,.
- [50] ZOUAOUI, S. .Mèmoire de magistère : Approche multicritère pour le problème de sac à dos. *USTHB,Algerie,alger 2011.*