

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

UNIVERSITÉ MOULOUD MAMMERI, TIZI-OUZOU



FACULTÉ DES SCIENCES
DÉPARTEMENT DE MATHÉMATIQUES

MÉMOIRE DE MASTER

SPÉCIALITÉ : RECHERCHE OPÉRATIONNELLE

OPTION : AIDE À LA DÉCISION

Présenté par *M^{lles}* :

Taous MEKHTOUB et Lynda TOUBAL

Thème :

Optimisation d'une Fonction sur l'Ensemble Pareto d'un Problème Multi-Objectifs Discret

Devant le jury d'examen composé de :

M^{me} KHEFFACHE Rezika ; Maître de conférence B ; UMMTO ; Présidente.

M^{me} BOUARAB Ouiza ; Maître de conférences A ; UMMTO ; Promotrice.

M^{me} FAHEM Karima ; Maître de conférences B ; UMMTO ; Examinatrice.

Soutenu le : 03/10/2019

Remerciements

Nous remercions tout d'abord le bon Dieu tout puissant d'avoir guide nos pas vers les portes du savoir, en nous donnant beaucoup de courage et de volonte pour mener ce travail a terme.

Nous tenons a adresser nos plus vifs remerciements a notre promotrice Mme BOUARAB Ouiza qui nous a guide et nous a oriente, nous lui exprimons notre grande gratitude pour l'attention et le temps qu'elle nous a consacre ainsi que les precieux conseils qu'elle nous a prodigue.

Nous avons eu un enorme plaisir de travailler avec elle.

Nous souhaitons egalement remercier les membres de jury qui nous ont fait l'honneur d'examiner ce travail.

Nous temoignons une reconnaissance particuliere aux enseignants qui nous ont tenu et transmis le savoir, l'esprit de recherche et de perseverance dans nos etudes durant notre cursus.

Enfin, nous remercions nos camarades de notre promotion pour leurs encouragements, ainsi que toute personne ayant contribuee de pres ou de loin a l'elaboration de ce memoire.

Dédicaces

Je dédie ce modeste travail à :

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

Je dédie ce travail à la mémoire de ma très chère soeur “Lynda” puisse Dieu, le tout puissant, l’avoir en sa sainte miséricorde !

Mes très chers frères Samir, Hamid, Ali que je remercie pour leurs encouragements et leurs soutien.

Mes très chères sœurs Tassadit et son mari Ali, Roza et son mari Ahmed sans oublier leur adorable fils Aylane.

Ma très chère amie Ouzena, merci pour tes encouragements, ton amitié, ton affection.

Ma chère binôme Lynda que je remercie pour son soutien et sa patience pour réaliser ce modeste travail.

Tous ceux qui m’aiment.

Taous

Dédicaces

Je dédie ce modeste travail à :

Mes parents, avec toute mon affection et tout mon respect, pour leur soutien et leurs encouragements tout au long de mes études.

Mes frères et sœurs que je remercie pour leurs encouragements, leurs aides ainsi qu'à toute ma grande famille.

Ma très chère binôme Taous avec laquelle j'ai partagé de bons moments.

Mes amis et camarades de ma promotion.

Lynda

Table des matières

Table des figures	4
Introduction générale	5
Chapitre 1. Programmation Linéaire Mono-Objectif en Nombres Entiers . . .	8
1.1. Introduction	8
1.2. Généralités sur l'optimisation	9
1.2.1. Définition de l'optimisation	9
1.2.2. Définition d'un problème d'optimisation	9
1.2.3. Types de problèmes	9
1.3. Classification des problèmes d'optimisation	10
1.4. Définitions	11
1.4.1. Minimum global	11
1.4.2. Minimum local	11
1.4.3. Ensemble convexe	11
1.4.4. Combinaison linéaire convexe	11
1.4.5. Fonction convexe et fonction concave	12
1.4.6. Point extrême	12
1.5. Formulation du problème	12
1.6. Méthodes de résolution d'un programme linéaire en nombres entiers	13
1.6.1. Méthodes des coupes	13
1.6.2. Méthode par séparation et évaluation (<i>Branch & Bound</i>)	18
1.7. Conclusion	21

Chapitre 2. Programmation Linéaire Multi-Objectifs	22
2.1. Introduction	22
2.2. Structure d'un problème d'optimisation multi-objectifs	23
2.3. Définitions	25
2.3.1. Dominance	25
2.3.2. Efficacité	27
2.3.3. Ensemble de Pareto	28
2.3.4. Frontière de Pareto	28
2.3.5. Point idéal	29
2.3.6. Point anti-idéal	29
2.3.7. Matrice des gains	29
2.3.8. Point nadir	30
2.3.9. Convexité	31
2.4. Problématique	31
2.5. Approches multi-critères	33
2.5.1. Approche Mono-objectif (Décideur $\implies$ Chercheur)	34
2.5.2. Approche Multi-objectifs	34
2.6. Méthodes de résolution	35
2.6.1. Méthode d'agrégation des objectifs [25]	35
2.6.2. Méthode ε - Contraintes	36
2.6.3. Méthode Goal Programming	37
2.7. Conclusion	37
Chapitre 3. Programmation Linéaire Multi-Objectifs en Nombres Entiers . . .	38
3.1. Introduction	38
3.2. Formulation générale du problème	38
3.3. Solutions efficaces supportées et solutions efficaces non supportées	39
3.4. Détection graphique des solutions efficaces	40
3.4.1. Définition du Cône	40
3.4.2. Définition du cône positif et du cône négatif	41
3.4.3. Exemple illustratif	41

3.5.	Quelques méthodes de résolution des problèmes (MOILP)	43
3.5.1.	Méthode de A.Crema et J.Sylva [13]	43
3.5.2.	Méthode de M.Abbas et M.Moulai [14]	46
3.5.3.	Méthode de F.Z.Ouail, M.E-A.Chergui et M.Moulai [20]	56
3.6.	Conclusion	64
Chapitre 4. Optimisation d'une Fonction Linéaire sur l'Ensemble Efficace		
	d'un Problème Multi-Objectifs Discret	65
4.1.	Introduction	65
4.2.	Rappels et préliminaires	66
4.3.	Test d'efficacité	68
4.4.	Méthode de résolution (Méthode de F.Z.Ouail, M.E-A.Chergui et M.Moulai) [36]	69
4.4.1.	Principe de la méthode	70
4.4.2.	Formulation de l'algorithme	70
4.5.	Résultats fondamentaux	71
4.6.	Exemple illustratif	74
4.7.	Conclusion	78
Conclusion générale		79
Bibliographie		80

Table des figures

1.1	Organigramme représentant les étapes de résolution de l'exemple par la méthode de Branch & Bound.	21
2.1	Représentation de l'espace de décision et l'espace des critères réalisables. . . .	24
2.2	Les niveaux de préférences dans la relation de dominance.	26
2.3	Illustration des différentes définitions.	32
2.4	Quel mode de résolution choisir ?	34
3.1	Illustration des solutions supportées et solutions non supportées.	41
3.2	Détection graphique des solutions efficaces de (P).	43
3.3	Représentation des différents types de solutions dans l'espace de décision. . . .	43
3.4	Représentation des différents types de solutions dans l'espace des critères. . . .	44
3.5	Graphe de l'exemple.	52
3.6	La troncature du point x_1^1	54
3.7	La troncature de l'arête E_6	55
3.8	La troncature de l'arête E_{18}	56
4.1	Graphe de l'exemple.	76

Introduction générale

Résoudre un problème d’optimisation consiste à trouver la ou les meilleures solutions vérifiant un ensemble de contraintes et d’objectifs définis par l’utilisateur. Pour déterminer si une solution est meilleure qu’une autre, il est nécessaire que le problème introduise un critère de comparaison. Ainsi, la meilleure solution, appelée aussi solution optimale, est la solution ayant obtenu la meilleure évaluation au regard du critère défini. Les problèmes d’optimisation sont utilisés pour modéliser de nombreux problèmes appliqués comme le traitement d’images, la conception de systèmes, la conception d’emplois du temps . . .etc.

Lorsqu’un seul critère est donné, par exemple un critère de minimisation de coût, la solution optimale est clairement définie, c’est celle qui a le coût minimal. Malheureusement, dans de nombreuses situations rencontrées en pratique un seul critère peut être insuffisant et le problème peut exiger la considération simultanée de plusieurs objectifs contradictoires.

On distingue en réalité deux types de problèmes d’optimisation : les problèmes combinatoires (ou problèmes “discrets”) et les problèmes à variables continues.

La principale difficulté que l’on rencontre en optimisation mono-objectif vient du fait que modéliser un problème sous la forme d’une équation unique peut être une tâche difficile. Avoir comme but de se ramener à une seule fonction objectif peut aussi biaiser la modélisation. L’optimisation multi-objectifs autorise ces degrés de liberté qui manquaient en optimisation mono-objectif. La recherche ne nous donnera plus une solution unique mais une multitude de solutions. La première notion d’optimalité a été introduite par Francis Ysidro Edgeworth [6] en 1881, elle a été utilisée de manière plus formelle par l’économiste italien Vilfredo Pareto [24] en 1896. Ces solutions sont appelées solutions de Pareto et l’ensemble de solutions que l’on obtient à la fin de la recherche est la surface de compromis.

C’est après avoir trouvé les solutions du problème multi-objectifs que d’autres difficultés surviennent, il faut sélectionner une solution dans cet ensemble. La solution qui sera choisie doit refléter les compromis opérés par le décideur vis-à-vis des différentes fonctions objectifs. Le décideur étant “humain”, il va faire des choix et l’un des buts de l’optimisation multi-objectifs va être de modéliser les choix du décideur ou plutôt ses préférences.

L'autre difficulté à laquelle on sera confronté avant d'effectuer une optimisation sera la sélection de la méthode d'optimisation. En effet, on peut répartir les méthodes d'optimisation multi-objectifs en trois grandes familles suivantes :

- Les méthodes d'optimisation a priori ;
- Les méthodes d'optimisation progressives ;
- Les méthodes a posteriori.

Ces familles sont définies en fonction de l'instant où l'on prend la décision d'effectuer ou non un compromis entre fonctions objectifs.

La résolution de problèmes multi-objectifs relève de deux disciplines assez différentes.

En effet, résoudre un problème multi-objectifs peut être divisé en deux phases :

1. La recherche des solutions de meilleur compromis. Se pose alors la question de savoir si toutes les solutions doivent être produites (elles peuvent être nombreuses) ou seulement un sous-ensemble représentatif. C'est la phase d'optimisation multi-objectifs.
2. Le choix de la solution à retenir. C'est la tâche du décideur qui, parmi l'ensemble des solutions de compromis, doit extraire celle(s) qu'il utilisera. On parle alors ici de décision multi-objectifs et cela fait appel à la théorie de la décision.

Dans le cadre de ce mémoire , on ne parlera que de la première phase qui consiste en la recherche de l'ensemble des solutions de meilleur compromis.

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. Dans notre travail, nous portons un intérêt considérable pour les problèmes discrets, l'intérêt de tels problèmes résulte du fait que dans la majorité des applications concrètes de programmation mathématique, la présence des variables entières est inévitable.

Enfin, le choix de la méthode de résolution à mettre en œuvre dépendra souvent de la complexité du problème. En effet, suivant sa complexité, le problème pourra ou non être résolu de façon optimale. Dans le cas de problèmes classés dans la classe P, un algorithme polynomial a été mis en évidence, il suffit donc de l'utiliser. Dans le cas de problèmes NP-difficiles, deux possibilités sont offertes. Si le problème est de petite taille, alors un algorithme exact permettant de trouver la solution optimale peut être utilisé (procédure de séparation et évaluation "Branch & Bound"...). Malheureusement, ces algorithmes par nature énumératifs, souffrent de l'explosion combinatoire et ne peuvent s'appliquer à des problèmes de grandes tailles. Dans ce cas, il est nécessaire de faire appel à des heuristiques permettant de trouver de bonnes solutions approchées.

Dans notre mémoire, nous traiterons principalement de la résolution de problèmes multi-objectifs linéaires discrets.

L'ensemble efficient d'un problème multi-objectifs peut être généralement très vaste et même infini dans le cas continu. Dans certaines situations, les décideurs n'ont pas besoin de l'ensemble de toutes les solutions efficaces d'un problème de programmation multi-objectifs mais uniquement de solutions efficaces qui réalisent l'optimum d'un objectif différent des objectifs déjà fixés. Ceci, nous mène vers la recherche de la solution optimale d'un critère sur l'ensemble des solutions efficaces, comme un outil pour mesurer les préférences des décideurs et distinguer parmi les nombreuses solutions efficaces.

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

◇ Chapitre 1 : Nous nous sommes intéressées à la programmation linéaire en nombres entiers, en rappelant les définitions de base et quelques méthodes de résolution de ce type de problème.

◇ Chapitre 2 : Nous avons donné les principales définitions liées à l'optimisation linéaire multi-objectifs puis les problématiques spécifiques à ce domain sont exposées et étudiées. Parmi ces problématiques, nous avons parlé en particulier de la structure de l'ensemble Pareto et le choix des méthodes de résolution suivies d'exemples numériques.

◇ Chapitre 3 : Est consacré à la présentation des problèmes d'optimisation linéaire multi-objectifs en nombres entiers et à l'étude de trois principales méthodes de résolution suivies chacune d'un exemple numérique.

◇ Chapitre 4 : Est consacré à l'optimisation d'une fonction linéaire sur l'ensemble efficace d'un problème multi-objectifs discret, nous avons donné la forme générale du problème principal puis nous avons décrit une nouvelle méthode exacte pour résoudre ce type de problème (méthode de F.Z.Ouail, M.E-A.Chergui et M.Moulai) illustré par un exemple numérique.

Enfin nous terminons notre travail par une conclusion générale.

Chapitre 1

Programmation Linéaire Mono-Objectif en Nombres Entiers

1.1 Introduction

La Programmation Linéaire en Nombres Entiers (PLNE) est un domaine des mathématiques et de l'informatique théorique dans lequel on considère des problèmes d'optimisation d'une forme particulière. Ces problèmes sont décrits par une fonction coût et des contraintes linéaires et des variables entières.

La contrainte d'intégrité sur les variables, qui différencie l'optimisation linéaire classique est nécessaire pour modéliser certains problèmes, en particulier des problèmes algorithmiques. Mais cette contrainte supplémentaire rend le problème plus complexe et demande des techniques particulières.

La Programmation Linéaire est un domaine centrale de l'optimisation, car les problèmes de (PL) sont les problèmes d'optimisation les plus faciles, toutes les contraintes y étant linéaires. Beaucoup de problèmes réels de la Recherche Opérationnelle peuvent être exprimés comme un problème de (PL), pour cette raison un grand nombre d'algorithmes pour la résolution d'autres problèmes d'optimisation sont fondés sur la résolution de problèmes linéaires.

Le terme programmation linéaire suppose que les solutions à trouver doivent être représentées en variables réelles. S'il est nécessaire d'utiliser des variables discrètes dans la modélisation du problème, on parle alors de programmation linéaire en nombre entiers (PLNE), il est important de savoir que ces derniers sont nettement plus difficiles à résoudre que les (PL) à variables continues. La programmation linéaire en nombres entiers est un domaine très riche de la programmation mathématique, les recherches dans ce domaine sont nombreuses et ont commencé depuis 1958 avec Gomory [27].

1.2 Généralités sur l'optimisation

1.2.1 Définition de l'optimisation

L'optimisation est une branche mathématique qui a pour but de chercher un extremum d'une fonction dont les entrées peuvent être soumises à des contraintes, elle traite tout problème lié à la prise de décision et elle joue un rôle très important en Recherche Opérationnelle.

1.2.2 Définition d'un problème d'optimisation

Un problème d'optimisation est défini par :

1. Un espace de décision : espace de solution de configuration constitué des différentes valeurs prises par les variables de décision.
2. Une ou plusieurs fonction(s) dites objectifs(s) à optimiser (minimiser ou maximiser).
3. Un ensemble de contraintes à respecter.

1.2.3 Types de problèmes

1. Problème de décision

C'est un problème dont la réponse est tout simplement oui ou non.

2. Problème de la classe P

Un problème est dit polynomial s'il existe un algorithme de complexité polynomiale pour le résoudre, la classe P est l'ensemble de tous les problèmes de complexité polynomiale.

3. Problème de la classe NP

Un problème est de la classe NP si pour toute instance de ce problème on peut vérifier en un temps polynomial par rapport à la taille de l'instance qu'une solution est une solution efficace.

4. Problème de la classe NP-hard

Un problème NP-hard est un problème vers lequel on peut ramener tout problème de la classe NP par une réduction polynomiale.

S'il est également dans la classe NP, on dit que c'est un problème NP-complet.

5. Problème de la classe NP complet

En théorie de la complexité, un problème NP-complet (c'est-à-dire un problème complet pour la classe NP) est un problème de décision vérifiant les propriétés suivantes :

- Il est possible de vérifier une solution efficacement (en un temps polynomial). La classe des problèmes vérifiant cette propriété est notée NP .
- Tous les problèmes de la classe NP se ramènent à celui-ci via une réduction polynomiale, cela signifie que le problème est au moins aussi difficile que tous les autres problèmes de la classe NP.

1.3 Classification des problèmes d'optimisation

On peut classer les différents problèmes d'optimisation que l'on rencontre dans la vie courante en fonction de leurs caractéristiques :

1. Nombre de variables de décision

- Une seule variable $\implies$ problème mono-variable.
- Plusieurs variables $\implies$ problème multi-variables.

2. Type de la variable de décision

- Variable réelle continue $x \in \mathbb{R}^n \implies$ problème continu.
- Variable entière $x \in \mathbb{N}^n \implies$ problème entier ou discret.
- Permutation sur un ensemble fini de nombre $x \in \{0, 1\}^n \implies$ problème combinatoire.

3. Type de la fonction objectif

- Fonction linéaire des variables de décision $\implies$ problème linéaire.
- Fonction quadratique des variables de décision $\implies$ problème quadratique.
- Fonction non linéaire des variables de décision $\implies$ problème non linéaire.

4. Formulation du problème

- Avec des contraintes $\implies$ problème contraint.
- Sans contraintes $\implies$ problème non contraint.

1.4 Définitions

Soit le problème d'optimisation suivant :

$$\begin{cases} \max & f(x) \\ x \in & S \end{cases}$$

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

1.4.1 Minimum global

Un point x^* est un minimum global de la fonction f si $f(x^*) \leq f(x), \forall x \in S$.

1.4.2 Minimum local

Un point x^* est un minimum local de la fonction f si $f(x^*) \leq f(x), \forall x \in V(x^*)$ tel que $V(x^*)$ est un voisinage de x^* .

1.4.3 Ensemble convexe

On dit qu'un ensemble S est convexe si et seulement si :

$$\forall x, y \in S \text{ et } \forall \mu \in [0,1] \Rightarrow \mu x + (1 - \mu)y \in S$$

1.4.4 Combinaison linéaire convexe

Un vecteur $y \in S$ est une combinaison linéaire convexe des points x_i ($i=\overline{1,p}$) s'il existe des coefficients réels μ_i tel que :

$$y = \sum_{i=1}^p \mu_i x_i \quad \text{avec } \mu_i \geq 0, i = \overline{1,p} \text{ et } \sum_{i=1}^p \mu_i = 1$$

1.4.5 Fonction convexe et fonction concave

La fonction $f : S \subset \mathbb{R}^n \rightarrow \mathbb{R}$ est dite convexe (respectivement concave) sur S convexe non vide de $\mathbb{R}^n$ si et seulement si :

$$\forall x, y \in S, \forall \mu \in [0,1] : f(\mu x + (1 - \mu)y) \leq \mu f(x) + (1 - \mu)f(y)$$

Une fonction f est dite une fonction concave si et seulement si $-f$ est convexe.

Remarque 1.1

Si $(-f)$ concave $\iff f$ convexe.

1.4.6 Point extrême

Soit S un ensemble convexe de $\mathbb{R}^n$, x est dit point extrême ou sommet de S si :

$$x = \mu x_1 + (1 - \mu)x_2, \forall x_1, x_2 \in S \text{ et } \forall \mu \in]0,1[\text{ alors } x = x_1 = x_2$$

1.5 Formulation du problème

Un problème de (PLNE) peut être mis sous deux formes classiques : la forme **canonique** et la forme **standard**.

La forme canonique

$$\begin{cases} \text{maximiser} & C^T x \\ \text{tel que} & Ax \leq b \\ & x \geq 0 \\ & x \text{ entier} \end{cases}$$

La forme standard

$$\begin{cases} \text{maximiser} & C^T x \\ \text{tel que} & Ax + S = b \\ & x \geq 0 \\ & S \geq 0 \\ & x \text{ entier} \end{cases}$$

Où :

C et b sont des vecteurs et A une matrice ayant des variables entières.

S est appelée variable d'écart.

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

Les deux méthodes les plus connues pour résoudre les problèmes (PLNE), sont les méthodes des coupes (*Gomory* et *Dantzig*) et la méthode de séparation et évaluation (*Branch & Bound*) qui se caractérisent par la possibilité de trouver une solution optimale dans un intervalle de temps bien déterminé.

1.6.1 Méthodes des coupes

1.6.1.1 Coupe de *Dantzig* [8]

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

$$\sum_{j \in N} x_j \geq 1 \tag{1.1}$$

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

1.6.1.2 Coupe fractionnaire de Gomory [27]

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

Dans une première étape, on utilise l'algorithme du simplexe pour résoudre le programme (PL) afin de chercher une solution de base optimale, si elle existe, on choisit une variable de base non entière et on génère une inéquation sur la contrainte associée à cette variable afin de couper la région de faisabilité courante, étant donnée une base optimale B du problème (PL).

Le tableau optimale correspondant est donné par :

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

TABLE 1.1 – Tableau simplexe optimal associé à la base B.

Où :

- $\pi = c_B(A^B)^{-1}$: est dit vecteur multiplicateur relatif à la base B.
- $\hat{c} = c - \pi A$: est dit vecteur coût réduit relatif à la base B avec $\hat{c}_B = 0$

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

$$x_i + \sum_{j \in N} \hat{a}_{ij} x_j = \hat{b}_i \quad (1.2)$$

Où :

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

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

B : est l'ensemble des indices de base.

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

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

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

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

De (1.2) on a

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

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

$$x_i + \sum_{j \in N} \lfloor \hat{a}_{ij} \rfloor x_j \leq \lfloor \hat{b}_i \rfloor \quad (1.3)$$

En soustrayant (1.3) de (1.2) on obtient :

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

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

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

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

Exemple illustratif

Application de l'algorithme de Gomory

Soit le programme linéaire en nombres entiers suivant :

$$(PLNE) \begin{cases} \max & z(x) = 100x_1 + 120x_2 \\ \text{s.c.} & 3x_1 + 4x_2 \leq 4100 \\ & x_1 + 3x_2 \leq 2400 \\ & 2x_1 + 2x_2 \leq 2625 \\ & x_1 \geq 0, x_2 \geq 0 \quad x_1, x_2 \in \mathbb{N} \end{cases}$$

En ajoutant les variables d'écart respectivement dans chacune des contraintes du programme linéaire relaxé, on obtient le problème linéaire continue (PLC) suivant :

$$(PLC) \begin{cases} \max & z(x) = 100x_1 + 120x_2 \\ \text{s.c} & 3x_1 + 4x_2 + x_3 = 4100 \\ & x_1 + 3x_2 + x_4 = 2400 \\ & 2x_1 + 2x_2 + x_5 = 2625 \\ & x_j \geq 0, j = \overline{1,5} \end{cases}$$

En appliquant l'algorithme du simplexe, on obtient la TABLE optimale suivante :

	c_j	100	120	0	0	0	
c_b	variable de base	x_1	x_2	x_3	x_4	x_5	solution de base
100	x_1	1	0	-1	0	2	1150
120	x_2	0	1	1	0	$-\frac{3}{2}$	$325/2$
0	x_4	0	0	-2	1	$\frac{5}{2}$	$1525/2$
z_j		100	120	20	0	20	$z=134500$
$z_j - c_j$		0	0	20	0	20	

TABLE 1.2 – Tableau simplexe.

La solution optimale est $x = (x_1, x_2)^t = (1150, 325/2)^t$. Puisque la solution optimale de (PLC) n'est pas à valeurs entières, on doit poursuivre en construisant la première contrainte additionnelle .

Construction de la première contrainte additionnelle :

On peut déterminer à partir de la TABLE optimale précédente, une contrainte (supplémentaire) qui réduit l'ensemble des solutions réalisables (on élimine la solution optimale valeurs non entières obtenues précédemment) sans enlever de solutions entières réalisables. Il s'agit de créer la contrainte supplémentaire à partir d'une des contraintes de la TABLE optimale dont la valeur x_{b_j} n'est pas entière.

Nous choisissons arbitrairement la 2^{ème} ligne de la TABLE , la variable de base que l'on veut rendre "entière" est donc x_2 .

Pour construire la contrainte additionnelle, on doit utiliser x_1, x_2, x_3, x_4, x_5 et b que l'on exprime sous forme algébrique suivante :

$$0x_1 + x_2 + x_3 + 0x_4 - \frac{3}{2}x_5 = \frac{325}{2}$$

On écrit chaque coefficient des variables comme la somme de sa partie entière et de sa partie fractionnaire :

$$0x_1 + (1 + 0)x_2 + (1 + 0)x_3 + 0x_4 + (-2 + \frac{1}{2})x_5 = (162 + \frac{1}{2})$$

En regroupant la partie entière des coefficients sur le membre de la droite, on obtient

$$\frac{1}{2}x_5 = \frac{1}{2} + (162 - x_2 - x_3 + 2x_5)$$

Supposons maintenant que toutes les variables sont des entiers non négatifs, alors le membre de gauche est une quantité non négative et le membre de droite doit être égal à $\frac{1}{2} + \text{entier}$

	$a = [a] + f$	
a	$[a]$	$f = a - [a]$
1	1	0
$-\frac{3}{2}$	-2	$-\frac{3}{2} - (-2) = \frac{1}{2}$
$\frac{325}{2}$	162	$\frac{325}{2} - 162 = \frac{1}{2}$

TABLE 1.3 –

$\frac{1}{2}x_5 = \frac{1}{2} + \text{entier non relatif et } \frac{1}{2}x_5 \geq 0$. Il s'ensuit que $\frac{1}{2}x_5 \geq \frac{1}{2}$

Cette inéquation constitue la coupe de Gomory que l'on doit ajouter au programme linéaire, elle va nous permettre d'éliminer la valeur non entière $\frac{325}{2}$ du tableau (on réduit la région des solutions réalisables)

On applique la règle pour générer cette coupe de Gomory.

La 1^{ère} coupe de Gomory est donc : $\frac{1}{2}x_5 \geq \frac{1}{2}$

Déterminons la nouvelle TABLE optimale (méthode duale du simplexe) :

Il n'est pas nécessaire, en ajoutant la contrainte additionnelle (coupe de Gomory) au programme linéaire, de résoudre à nouveau tout le programme linéaire par la méthode du simplexe, il suffit d'ajouter la contrainte de Gomory en incluant la variable d'écart appropriée à la dernière TABLE optimale et de déterminer la nouvelle solution optimale à l'aide de l'algorithme dual du simplexe.

Pour utiliser la méthode duale du simplexe, il faut modifier la contrainte de Gomory en multipliant les deux membres de l'équation par (-1), ce qui donne $-\frac{1}{2}x_5 + x_6 = -\frac{1}{2}$

TABLE avec l'ajout de la coupe de Gomory :

	c_j	100	120	0	0	0	0	
c_B	variables de base	x_1	x_2	x_3	x_4	x_5	x_6	solution de base
100	x_1	1	0	-1	0	2	0	1150
120	x_2	0	1	1	0	$-\frac{3}{2}$	0	$325/2$
0	x_4	0	0	-2	1	$\frac{5}{2}$	0	$1524/2$
0	x_6	0	0	0	0	$-\frac{1}{2}$	1	$-1/2$
z_j		100	120	20	0	20	0	z=134500
$c_j - z_j$		0	0	-20	0	-20	0	

TABLE 1.4 –

Le seul $x_{B_j} \leq 0$ correspond à x_6 : c'est la variable sortante ($i = k = 4$) : la variable entrante est déterminée selon le critère suivant :

$$\min_j \left\{ \frac{c_j - z_j}{\mu_{kj}}, \mu_{kj} < 0 \right\} = \min_j \left\{ \frac{c_j - z_j}{\mu_{4j}}, \mu_{4j} < 0, j = 3 \text{ et } 5 \right\}$$

Il y a que μ_{45} qui est négatif, on obtient alors $\frac{c_5 - z_5}{\mu_{45}} = \frac{-20}{-\frac{1}{2}} = 40$

En introduisant la variable x_5 et en effectuant l'opération de pivotement, on obtient la TABLE optimale suivante :

	c_j	100	120	0	0	0	0	
c_B	variables de base	x_1	x_2	x_3	x_4	x_5	x_6	solution de base
100	x_1	1	0	-1	0	0	4	1148
120	x_2	0	1	1	0	0	-3	164
0	x_4	0	0	-2	1	0	5	760
0	x_5	0	0	0	0	1	-2	1
z_j		100	120	20	0	0	40	z=134480
$z_j - c_j$		0	0	-20	0	0	-40	

TABLE 1.5 –

La solution optimale est $(x_1, x_2)^t = (1148, 164)^t$ et $Z = 134480$ puisque cette solution optimale est entière, il n'est pas nécessaire d'effectuer une autre coupe de Gomory, nous avons obtenu la solution optimale entière du (PLNE).

1.6.2 Méthode par séparation et évaluation (*Branch & Bound*)

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

Principe de la méthode

- On construit un arbre de recherche dont le problème initial est la racine.
- On divise le problème en sous-problèmes : l'optimum peut appartenir à l'un de ces sous-problèmes.
 - On élimine tout problème infaisable.
 - Si possible, on calcule la solution du sous-problème.
 - Sinon on calcule une borne inférieure, si elle est supérieure à la meilleure solution déjà obtenue, on élimine le sous-problème considéré.

- Dans le cas restant, on subdivise à nouveau le domaine.

Étape de résolution

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

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

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

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

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

2/ Séparation

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

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

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

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

- Il s'agit de résoudre chaque sous-problème par l'algorithme du simplexe, notons par MBSD, la meilleure borne supérieure disponible de Z , elle correspond à la valeur maximale de la fonction objectif des deux sous-problèmes que nous venons de résoudre.
- Nous révisons également la borne inférieure de Z , notons cette valeur par MBID, la meilleure borne inférieure disponible, elle correspond à la valeur maximale de la fonction objectif de toutes les solutions entières obtenues jusqu'à présent.

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

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

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

5/ Critère d'optimalité

Toutes les branches sont saturées, donc on termine avec la procédure de séparation.

La solution optimale de (PLNE) est la solution entière qui donne la meilleure borne inférieure pour $Z(MBID)$, à l'optimum on aura $MBID = MBSD = Z_{opt}$ de (PLNE).

6/ Continuer le processus de séparation

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

Exemple illustratif

Soit le programme linéaire en nombres entiers suivant :

$$(P) \left\{ \begin{array}{l} \max \quad z(x) = 2x_1 + x_2 \\ s.c. \quad x_1 + 2x_2 \leq 12 \\ \quad \quad 2x_1 - x_2 \leq 8 \\ \quad \quad 4x_1 - 3x_2 \leq 8 \\ \quad \quad x_1 \geq 0, x_2 \geq 0 \\ \quad \quad x_1, x_2 \text{ entiers} \end{array} \right.$$

Sa solution est donnée dans la FIGURE (1.1) où la méthode de Branch & Bound est utilisée.

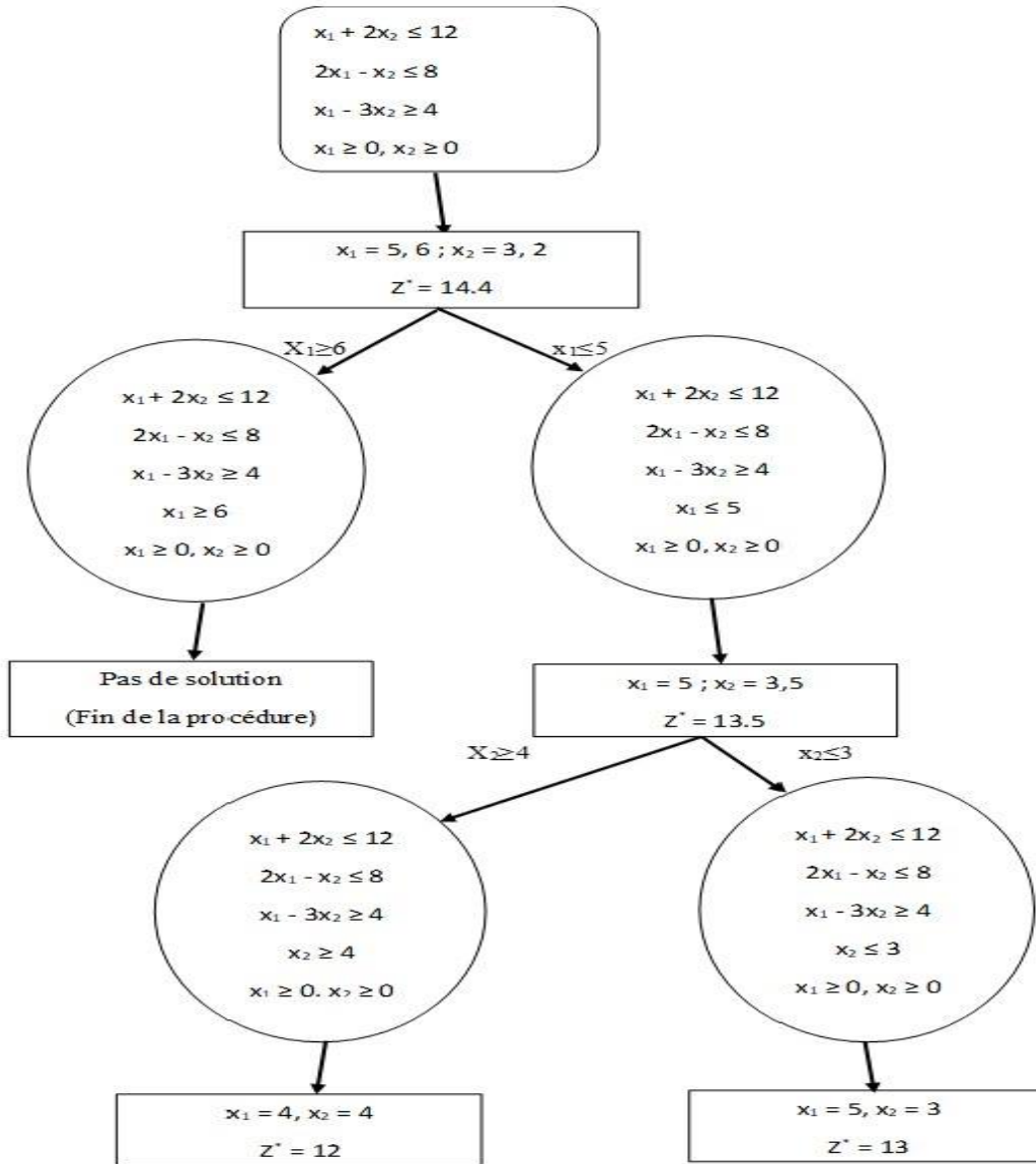


FIGURE 1.1 – Organigramme représentant les étapes de résolution de l'exemple par la méthode de Branch & Bound.

La solution optimale de (P) est $x = (5, 3)^t$ et $Z^* = 13$.

1.7 Conclusion

Ce chapitre avait pour objectif d'introduire dans un premier temps, les concepts fondamentaux des problèmes d'optimisation. Puis nous avons présenté les principales définitions nécessaires à l'optimisation mono-objectif en nombres entiers. Ainsi on a présenté quelques méthodes de résolution comme la méthode des coupes (Gomory et Dantzig) et la méthode de Branch & Bound. Ces méthodes sont les plus utilisées dans la résolution d'un (PLNE).

Chapitre 2

Programmation Linéaire Multi-Objectifs

2.1 Introduction

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

L'optimisation multi-objectifs, appelée aussi optimisation multi-critères, a un impact important dans l'industrie. En effet, la plupart des problèmes rencontrés en pratique sont de nature multi-objectifs, car il ne s'agit pas d'optimiser seulement un seul objectif, comme est le cas dans l'optimisation mono-objectif, mais plutôt d'optimiser simultanément plusieurs objectifs qui sont souvent contradictoires¹ et non commensurables². On cherche la meilleure solution selon les paramètres du problème. Dans ce cas, un problème d'optimisation multi-objectifs est en fait un problème de décision³ multi-objectifs.

La solution d'un problème multi-objectifs n'est pas une solution unique mais un ensemble de solutions, connu comme l'ensemble des solutions Pareto optimale [24]. Toute solution de cet ensemble est optimale dans le sens qu'aucune amélioration ne peut être faite sur un composant du vecteur sans dégradation d'au moins un autre composant du vecteur. Autrement dit, la résolution d'un problème multi-objectifs consiste à trouver l'ensemble des solutions optimales au sens de Pareto, c'est à dire les solutions telles qu'il n'existe pas de solutions au moins aussi bonnes sur tous les critères et meilleures sur au moins un critère.

-
1. La diminution d'un objectif implique l'augmentation d'un autre objectif.
 2. Les objectifs sont exprimés dans des unités différentes.
 3. C'est un problème où la décision finale revient au décideur.

2.2 Structure d'un problème d'optimisation multi-objectifs

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

- **Fonction objectif** : La fonction objectif est la fonction à optimiser (maximiser ou minimiser), appelée aussi fonction d'adaptation, fonction coût, ou encore performance. On la note $F(x)$.
- **Paramètre** : Le paramètre est une variable qui exprime une donnée quantitative ou qualitative au problème (exemple : coût, temps, taux d'erreur, ...etc). Ce sont les variables de la fonction objectif, appelée aussi variables de conception, variables d'optimisation ou encore variables de projet.
- **Contrainte** : La contrainte est une condition qui doit être respectée par le vecteur de décision pour trouver la solution admissible.

Un problème d'optimisation multi-objectifs (MOP) : "Multiobjective Optimization Problem" est un problème de programmation mathématique posé sous la forme suivante :

$$(\text{MOP}) \begin{cases} \text{"Optimiser"} & F(x) = (f_1(x), f_2(x), \dots, f_k(x)) \\ \text{s.c.} & x \in S \subset \mathbb{R}^n \end{cases} \quad (2.1)$$

Où :

- $x = (x_1, x_2, x_3, \dots, x_n)^t \in \mathbb{R}^n$ représente le vecteur de décision avec x_i $i = \overline{1, n}$ les variables de décision et n le nombre de ces variables.
- $\mathbb{R}^n$ s'appelle espace de décision.
- $F(x) = (f_1(x), f_2(x), \dots, f_k(x)) \in \mathbb{R}^k$ représente le vecteur des fonctions objectifs avec f_i les fonctions objectifs ($k \geq 2$).
- $\mathbb{R}^k$ s'appelle espace des critères ou espace des objectifs.

Où :

$$F : \mathbb{R}^n \longrightarrow \mathbb{R}^k$$

$$x \longmapsto F(x) = (f_1(x), f_2(x), \dots, f_k(x))$$

Et :

$$f_i : \mathbb{R}^n \longrightarrow \mathbb{R}$$

$$x \longmapsto f_i(x) , \quad i = \overline{1, k}.$$

- S est l'ensemble des solutions réalisables, $S \subset \mathbb{R}^n$.
- $D = F(S)$ est l'image de S par F , il s'appelle espace des critères réalisables

$D \subset \mathbb{R}^k$, il est donné par : $D = F(S) = \{z \in \mathbb{R}^k / z = F(x), x \in S\}$.

où z est le vecteur de l'espace des critères qui correspond à la solution réalisable x .

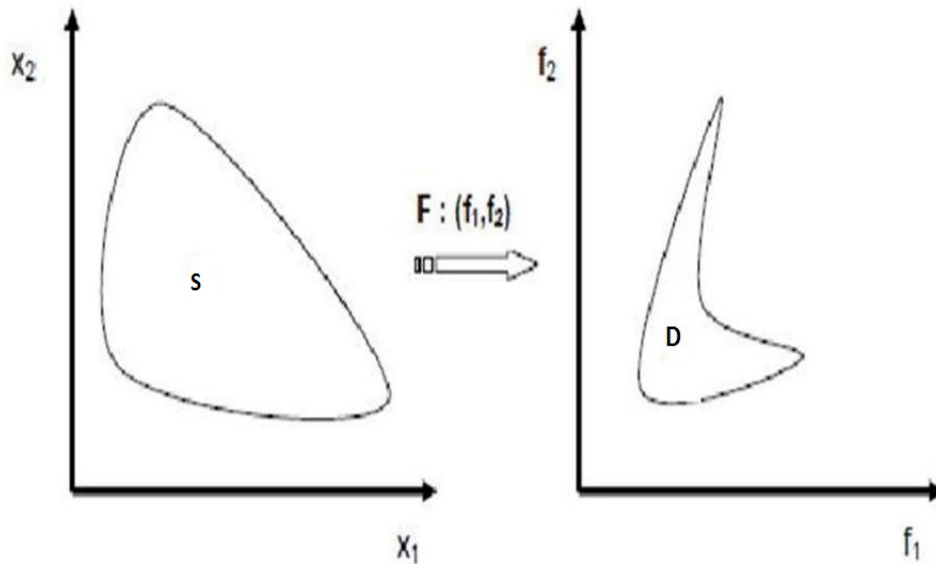


FIGURE 2.1 – Représentation de l'espace de décision et l'espace des critères réalisables.

Remarque 2.1

Le symbole " " dans (2.1) indique qu'on ne peut pas trouver en général une solution admissible qui optimise (maximise ou minimise) simultanément les k fonctions objectifs.

Remarque 2.2

Si les k critères sont linéaires et les contraintes sont linéaires en x , on parle alors de problème de Programmation Linéaire Multi-Objectifs (**MOLP**) (Multiple Objective Linear Programming Program) écrit sous la forme suivante :

$$(\text{MOLP}) \begin{cases} \min & F(x) = (c^1x, c^2x, \dots, c^kx) \\ \text{s.c.} & x \in S = \{x \in \mathbb{R}^n / Ax = b, x \geq 0\} \end{cases} \quad (2.2)$$

Avec :

$A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ et $C = (c^1, c^2, \dots, c^k) \in \mathbb{R}^{k \times n}$.

2.3 Définitions

Dans la suite, on considère, sans perte de généralité, que toutes les fonctions objectifs sont à minimiser. En effet, on peut aisément transformer un problème de maximisation en un problème de minimisation grâce à l'équivalence suivante : $\max(F) = -\min(-F)$

2.3.1 Dominance

2.3.1.1 Dominance

Soient $F(x^*)$; $F(x) \in F(S)$ deux vecteurs critères. On dira que $F(x)$ domine $F(x^*)$ si seulement si :

$$\forall i = \overline{1, k}, c^i x \leq c^i x^* \text{ et } \exists i = \overline{1, k} / c^i x < c^i x^* \quad (2.3)$$

Autrement dit, $F(x)$ domine $F(x^*)$ si et seulement si $F(x)$ est au moins aussi bon que $F(x^*)$ sur tous les objectifs et $F(x)$ est meilleur que $F(x^*)$ sur au moins un objectif.

2.3.1.2 Dominance forte

Soient $F(x^*)$; $F(x) \in F(S)$ deux vecteurs critères. On dira que $F(x)$ domine fortement $F(x^*)$ si seulement si :

$$\forall i = \overline{1, k}, c^i x < c^i x^* \quad (2.4)$$

Autrement dit, $F(x)$ domine fortement $F(x^*)$ si et seulement si $F(x)$ est meilleur que $F(x^*)$ sur tous les objectifs.

2.3.1.3 Dominance faible

Soient $F(x^*)$; $F(x) \in F(S)$ deux vecteurs critères. On dira que $F(x)$ domine faiblement $F(x^*)$ si seulement si :

$$\forall i = \overline{1, k}, \quad c^i x \leq c^i x^* \quad (2.5)$$

Remarque 2.3

Pour toute paire de vecteurs objectifs Y et $F(x^*)$, un et un seul des cas suivants est vrai :

- $F(x)$ domine $F(x^*)$.
- $F(x)$ est dominé par $F(x^*)$.
- $F(x)$ et $F(x^*)$ sont équivalents au sens de la dominance.

Les solutions équivalentes au sens de la dominance sont appelées aussi solutions ***Pareto équivalentes*** ou encore ***solutions non dominées***.

Remarque 2.4

En appliquant la définition de la dominance, on peut définir quatre régions auxquelles on peut attribuer des niveaux de préférences dans la relation de dominance pour un problème à minimiser. Ces régions sont représentées dans la FIGURE 2.2

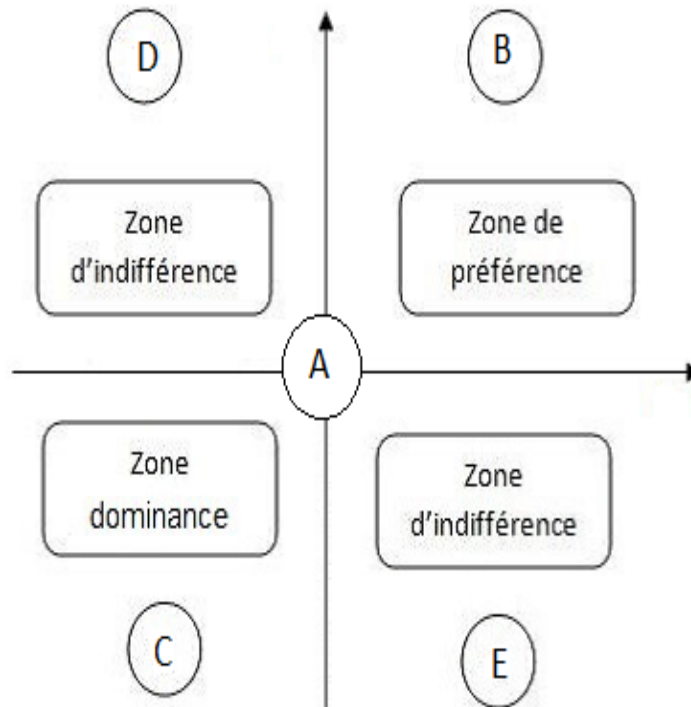


FIGURE 2.2 – Les niveaux de préférences dans la relation de dominance.

Cette figure nous illustre les cas suivants :

- A domine B , donc il y’a une amélioration des deux objectifs.
- A dominé par C , donc il y’a une détérioration des deux objectifs.
- A est incomparable à E et à D , c.à.d. on ne peut pas se prononcer sur la préférence.

2.3.1.4 Propriété de la relation de dominance

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

Définition : La relation binaire de dominance notée $\prec$:

- n’est pas réflexive, car un vecteur ne se domine pas lui même.
- n’est pas symétrique, car on n’a jamais $U \prec V$ et $V \prec U$.
- est transitive, car $U \prec V$ et $V \prec U$ implique $U \prec W$.
- n’est pas antisymétrique, en raison de l’existence de solutions Pareto-équivalentes.

Au final, on conclut que la relation de dominance est donc une relation d’ordre partiel strict sur l’espace des critères réalisables [29].

Remarque 2.5

Notons que $U \succ V$ n’implique pas $V \prec U$.

2.3.2 Efficacité

2.3.2.1 Efficacité ou Pareto Optimalité

Les problèmes rencontrés dans la vie réelle sont souvent décrits à l’aide de plusieurs critères ou objectifs, c’est l’optimisation multi-objectifs qui consiste à optimiser simultanément plusieurs fonctions [25], [1], [30]. Cependant, ces fonctions objectifs ont la particularité d’être souvent conflictuelles et contradictoires entre elles, ce qui explique que dans la plupart du temps on trouve une multitude de solutions en résolvant un problème d’optimisation multi-objectifs. La solution n’est pas unique contrairement au problème mono-objectif, mais un ensemble de solutions connu comme l’ensemble des solutions ”**Pareto optimales**”.

Cette notion d’optimalité est introduite par l’économiste Irlandais Francis Ysidro Edgeworth en 1881 [6]. Ensuite généralisée au 19^{ème} siècle, par l’économiste et sociologue Italien Vilfredo Pareto en 1896 [24], qui introduit le concept d’efficacité, d’où le nom ”**Pareto optimalité**”. En effet, il définit la formule suivante :

“Dans un problème multi-objectifs, il existe un équilibre tel que l’on ne peut pas améliorer un objectif sans détériorer au moins un des autres objectifs”.

Cet équilibre est appelé optimum de Pareto, point non dominé ou encore solution efficace.

La définition de la Pareto optimalité provient directement de la notion de dominance d'après cette équivalence :

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

$$\begin{aligned} c^i x &\leq c^i x^*, \quad \forall i = \overline{1, k}, \\ \text{et} \\ c^i x &< c^i x^*, \quad \text{pour au moins un } i = \overline{1, k}. \end{aligned} \tag{2.6}$$

2.3.2.2 Efficacité forte

Une solution x^* est fortement efficace si et seulement s'il n'existe pas un vecteur de décision x tel que $c^i x < c^i x^*, \forall i = \overline{1, k}$ (2.7)

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

2.3.2.3 Efficacité faible

Une solution x^* est faiblement efficace si et seulement s'il n'existe pas un vecteur de décision x tel que $c^i x \leq c^i x^* \forall i = \overline{1, k}$. (2.8)

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

Remarque 2.6

Toute solution efficace est faiblement efficace et l'inverse est faux.

Remarque 2.7

La notion de dominance se réfère à l'espace des critères alors que la notion d'efficacité se réfère à l'espace de décision.

2.3.3 Ensemble de Pareto

L'ensemble de Pareto est l'ensemble des solutions optimales au sens de Pareto, il est appelé aussi ensemble des solutions non dominées ou ensemble des compromis ou encore ensembles des solutions efficaces .

2.3.4 Frontière de Pareto

La frontière de Pareto est l'image de l'ensemble de Pareto dans l'espace des critères réalisables. Il est appelé aussi la surface de Pareto ou encore le front de Pareto.

2.3.5 Point idéal

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

$$Z^* = (z_1^*, z_2^*, \dots, z_k^*) \quad \text{avec } z_i^* = \min_{x \in S} (f_i(x)), \forall i = \overline{1, k}. \quad (2.9)$$

Remarque 2.8

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

2.3.6 Point anti-idéal

Le point anti-idéal est un vecteur de $\mathbb{R}^k$ dont les coordonnées sont les valeurs maximales de chacune des fonctions objectifs pris séparément. D'une autre manière, c'est le vecteur qui maximise chacune des fonctions objectifs f_i séparément.

$$Z^a = (z_1^a, z_2^a, \dots, z_k^a) \quad \text{avec } z_i^a = \max_{x \in S} (f_i(x)), \forall i = \overline{1, k}. \quad (2.10)$$

2.3.7 Matrice des gains

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

$$\begin{aligned} z_i^* &= \min_{x \in S} (f_i(x)) = f_i(\bar{x}_i), \forall i = \overline{1, k} \\ \text{et} \\ z_{ij} &= f_i(\bar{x}_j), \forall i = \overline{1, k}, \forall j = \overline{1, k} \text{ et } i \neq j. \end{aligned} \quad (2.11)$$

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

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

$$\begin{bmatrix} z_{11} = f_1(\bar{x}_1) & z_{12} & \cdots & z_{1k} \\ z_{21} & z_{22} = f_2(\bar{x}_2) & \cdots & z_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ z_{k1} & z_{k2} & \cdots & z_{kk} = f_k(\bar{x}_k) \end{bmatrix}$$

avec :

$\bar{x}_1$ est la solution optimale de $z_1^* = \min_{x \in S} f_1(x)$.

$\bar{x}_2$ est la solution optimale de $z_2^* = \min_{x \in S} f_2(x)$.

$\vdots$

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

Remarque 2.9

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

2.3.8 Point nadir

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

$$Z^{nad} = (z_1^{nad}, z_2^{nad}, \dots, z_k^{nad}) \quad \text{avec } z_i^{nad} = \max_{x \in X_{Pareto}} (f_i(x)), \forall i = \overline{1, k} \quad (2.12)$$

Remarque 2.10

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

$$f_i^{norm} = \frac{f_i - z_i^*}{z_i^{nad} - z_i^*} \quad (2.13)$$

Une visualisation de l'ensemble des différentes définitions est illustrée sur la FIGURE (2.3) suivante :

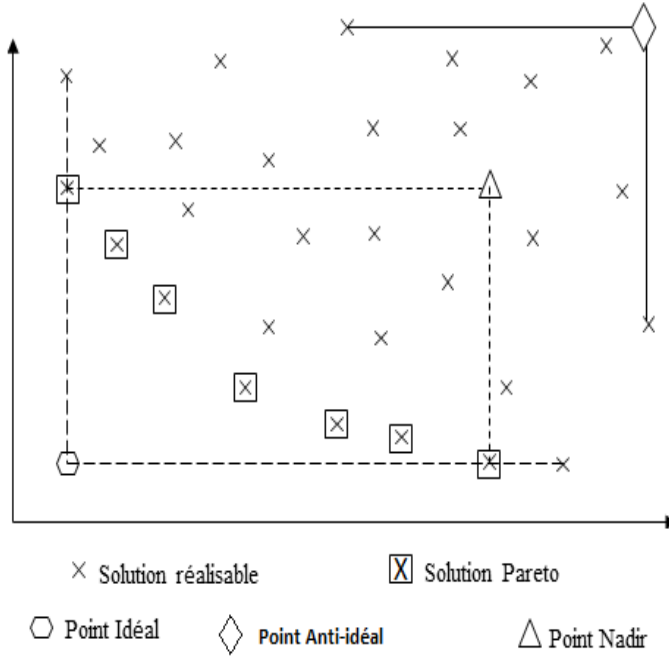


FIGURE 2.3 – Illustration des différentes définitions.

2.3.9 Convexité

Définition 2.1

— Une fonction f est dite convexe sur l'ensemble E non vide de $\mathbb{R}^n$ si et seulement si :

$$\forall x, y \in E, \forall \lambda \in [0, 1] \text{ on a : } f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y) \quad (2.14)$$

La convexité joue un rôle important dans la reconnaissance de la difficulté du problème. En effet, plusieurs méthodes d'optimisation sont incapables de résoudre d'une façon optimale des problèmes non convexes

2.4 Problématique

L'optimisation multi-objectifs est apparue au 19^{ème} siècle par **Edgeworth** (1881) [6] puis généralisée par **Pareto** (1896) [24], elle a été initialement utilisée dans l'économie et le management. A présent, elle occupe une grande place dans les sciences de l'ingénierie depuis 1980, on peut citer l'exemple des problèmes de transport, les problèmes d'aéronautique, ...etc. La résolution des problèmes multi-objectifs consiste à optimiser simultanément les fonctions objectifs qui sont contradictoires entre elles, contrairement à l'optimisation mono-objectif, de ce fait on est amené donc à chercher l'optimum de

plusieurs composantes d'un vecteur fonction, ainsi on obtient un ensemble de solutions optimales.

Les problèmes multi-objectifs ont la particularité d'être beaucoup plus difficiles à résoudre que leurs équivalents mono-objectif. En effet, la difficulté réside dans le fait qu'il n'existe pas de définition de la solution optimale. Le décideur intervient alors, il exprime ses préférences, c'est à dire il exprime si une solution est préférable à une autre solution mais il n'existe pas une solution meilleure que toutes les autres.

Dès lors, résoudre un problème multi-objectif ne consiste pas à rechercher la solution optimale mais l'ensemble des solutions satisfaisantes pour lesquelles on ne pourra pas effectuer une opération de classement. Les méthodes de résolution des problèmes multi-objectifs sont donc des méthodes d'aide à la décision car le choix final revient uniquement au décideur.

Pour répondre à ce problème, il existe deux types d'approches de résolution multi-objectifs :

- La première, consiste à ramener un problème multi-objectif à un problème mono-objectif.
- La deuxième, consiste à proposer des solutions au problème en prenant en compte l'ensemble des objectifs.

La différence entre ces deux approches s'exprime dans la FIGURE (2.4) suivante :

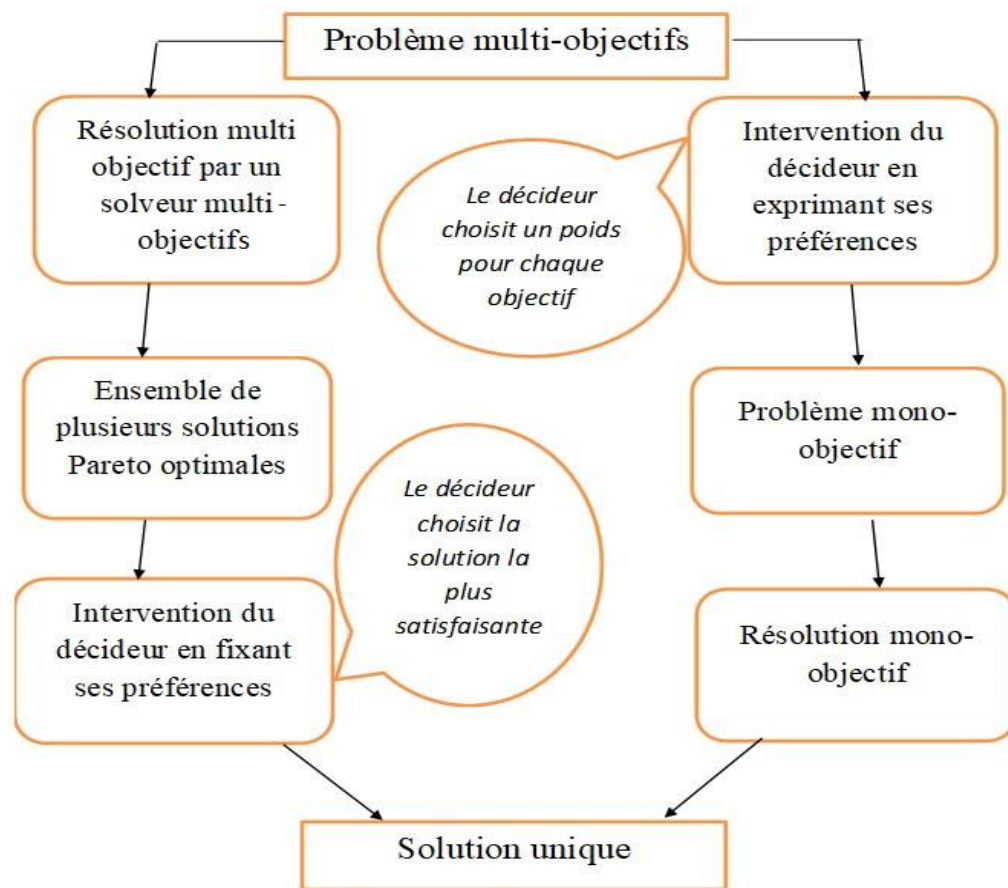


FIGURE 2.4 – Quel mode de résolution choisir ?

Soit le décideur intervient dès le début de la définition du problème, en exprimant ses préférences, afin de transformer un problème multi-objectif en un problème mono-objectif. Soit le décideur effectue son choix dans l'ensemble des solutions proposées par le solveur multi-objectifs.

2.5 Approches multi-critères

Résoudre un problème d'optimisation multi-objectifs consiste à passer par deux disciplines bien précises et importantes

- Phase d'optimisation multi-objectifs : C'est la phase de la recherche des solutions efficaces.
- Phase de décision multi-objectifs : C'est le rôle du décideur, en effet il intervient et choisit une solution unique parmi l'ensemble des solutions efficaces trouvées auparavant.

2.5.1 Approche Mono-objectif (Décideur $\implies$ Chercheur)

Elle caractérise les méthodes dites “ *a priori* ” utilisées pour leur simplicité de mise en œuvre. En effet, les objectifs du problème multi-objectifs sont transformés en une seule fonction objectif et dans ce cas le décideur est supposé quantifier à priori l'importance de chaque critère, c'est à dire connaître à priori le poids de chaque objectif afin de construire une fonction unique et cela revient à résoudre un problème simple objectif par des méthodes d'optimisation classique. Le processus d'optimisation mono-objectif est ensuite lancé afin de déterminer la solution optimale. Cependant dans la plupart des cas, la fonction d'utilité n'est pas connue à priori du processus d'optimisation et les différents objectifs ne sont pas commensurables. De plus, l'espace de recherche défini peut ne pas représenter le problème initial. Si le décideur n'est pas à même d'indiquer à priori le type de compromis qu'il souhaite réaliser entre les critères, il n'est pas pertinent de chercher une et une seule solution efficace réalisant une agrégation entre ces critères.

2.5.2 Approche Multi-objectifs

Elle est constituée d'approches “*a posteriori*” et d'approches “*interactive*”.

2.5.2.1 Approche a Posteriori (Chercheur $\implies$ Décideur)

Le décideur prend sa décision d'après un ensemble de solutions calculées par un solveur. Dans ce cas, la quantité de solutions dépend du choix de la méthode de résolution, car celle-ci va devoir donner un ensemble de résultats le plus représentatif de l'espace des objectifs efficaces.

2.5.2.2 Approche Interactive (Décideur $\iff$ Chercheur)

Une méthode interactive consiste en une série d'étapes de calculs et de dialogue entre le décideur et le solveur (chercheur). L'étape de calcul fournit une première solution, celle-ci est présentée au décideur qui réagit en apportant des informations supplémentaires sur ses préférences (étape de dialogue avec l'analyste). Il modifie certaines variables ou contraintes afin de diriger le processus d'optimisation. Ces informations, une fois injectées dans le modèle utilisé, permettent de construire une nouvelle solution. Dans cette classe d'approche le décideur intervient dans chaque étape de résolution du problème.

Les processus de calculs et de dialogue sont alternés. Le décideur modifie ainsi interactivement le compromis entre ses préférences et les résultats trouvés, cette approche exige une connaissance approfondie des outils utilisés par le décideur.

2.6 Méthodes de résolution

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

2.6.1 Méthode d'agrégation des objectifs [25]

Cette méthode est la plus largement utilisée en pratique et la plus efficace du point de vu informatique car elle est très facile à implémenter. D'ailleurs, elle est l'une des premières méthodes utilisées pour résoudre les problèmes multi-objectifs [5], elle est appelée aussi méthode de la somme pondérée.

Elle consiste à ramener le problème multi-objectifs en un problème mono-objectif. Elle somme tous les objectifs en affectant à chacun d'entre eux un coefficient de poids. Le décideur choisit les poids en fonction de l'importance relative des objectifs.

Le problème multi-objectif (MOP) devient sous la forme suivante :

$$(P_\lambda) \begin{cases} \min & \sum_{i=1}^k \lambda_i f_i(x) \quad \text{combinaison linéaire des } f_i \\ x \in E \end{cases} \quad (2.16)$$

Où :

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

La résolution du problème multi-objectifs (MOP) consiste à résoudre le problème paramétrique (2.16). La solution obtenue dépend fortement des coefficients choisis pour les poids. Cette méthode a été développée pour la généralisation des solutions non dominées. Elle est très efficace et applicable pour produire une solution non dominée initiale et peut être employée comme solution initiale pour d'autre technique.

Théorème 2.1

Toute solution optimale du problème (2.16) est une solution Pareto optimale pour le problème multi-objectifs (MOP) si tous les poids sont strictement positifs.

Théorème 2.2 [18]

Si le problème multi-objectifs (MOP) est convexe et si x^* est une solution Pareto optimale alors

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

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

2.6.2 Méthode ε - Contraintes

Cette méthode est probablement l'une des méthodes les plus connues pour la résolution des problèmes multi-objectifs. Elle a été introduite par Haines en 1971 [31] et depuis elle est vue comme un outil très utilisé dans la résolution de ce type de problème, où elle transforme le problème multi-objectifs (MOP) en un problème mono-objectif. Elle est basée sur la minimisation d'un objectif jugé par le décideur prioritaire et de convertir les autres objectifs en contraintes supplémentaires au problème.

Le problème (P_ε) s'écrit comme suit :

$$(P_\varepsilon) \begin{cases} \text{minimiser} & f_i(x) \\ \text{s.c.} & x \in E \\ & f_j(x) \leq \varepsilon_j, \forall j \neq i, j = \overline{1, k} \end{cases} \quad (2.17)$$

Théorème 2.3 [32]

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

Remarque 2.11

- Les poids λ_i et les paramètres ε_j sont choisis par le décideur.
- En faisant varier les poids et les paramètres dans les deux méthodes, on peut générer un sous ensemble de l'ensemble Pareto du problème multi-objectifs.

Exemple illustratif

Soit le problème multi-objectifs suivant :

$$(P) \begin{cases} \min & f(x) = (f_1(x) = -3x_1 + x_2, f_2(x) = x_1 + 2x_2, f_3(x) = 2x_1 + x_2) \\ & 3x_1 - x_2 \leq 6 \\ & x_2 \leq 2 \\ & x = (x_1, x_2) \in \mathbb{R}_+^2 \end{cases}$$

Résoudre le problème en considérant le critère 3 comme prioritaire avec

$$\varepsilon = (\varepsilon_1, \varepsilon_2)^t = (1, 3)^t$$

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

La résolution de (P_ε) par la méthode du simplexe nous donne la solution optimale

$$x^* = (x_1, x_2)^t = (0, 0)^t$$

Donc, cette solution est Pareto optimale pour le problème initial d'après le théorème 2.3.

2.6.3 Méthode Goal Programming

Cette méthode est également appelée “Target Vector Optimization“. Charnes et Cooper proposent en 1961 cette méthode dans laquelle le décideur fixe un but T_i , $i = \overline{1, k}$ à atteindre pour chaque objectif f_i [2]. Ces valeurs sont ensuite ajoutées au problème comme des contraintes supplémentaires. La nouvelle fonction objectif est modifiée de façon à minimiser la somme des écarts entre les résultats et les buts à atteindre.

$$\min_{x \in E} \sum_{i=1}^k (f_i(x) - T_i) \quad (2.18)$$

Remarque 2.12

- Cette méthode a l'avantage de fournir un résultat même si un mauvais choix initial a conduit le décideur à donner un ou plusieurs but(s) non réalisable(s).
- La méthode est facile à mettre en œuvre mais la définition des buts à atteindre est une question délicate.

2.7 Conclusion

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

Chapitre 3

Programmation Linéaire Multi-Objectifs en Nombres Entiers

3.1 Introduction

La plupart des problèmes rencontrés en pratique sont de nature multi-objectifs. En effet, un problème d'optimisation multi-objectifs consiste à optimiser simultanément plusieurs objectifs souvent contradictoires et parfois complémentaires. Contrairement à l'optimisation mono-objectif, la résolution d'un problème multi-objectifs nous donne une multitude de solutions, un ensemble de solutions connu comme l'ensemble des solutions efficaces. Toute solution de cet ensemble est optimale dans le sens qu'aucune amélioration ne peut être faite sur un critère sans dégradation d'au moins un autre critère. Si les fonctions objectifs sont linéaires, que l'ensemble des solutions réalisables est défini par des contraintes linéaires, et si les variables x prennent des valeurs entières, alors on parle d'un problème d'Optimisation Multi-Objectifs Linéaire en Nombres Entiers.

A travers ce chapitre, qui a pour objectif principal la présentation du contexte de la programmation linéaire multi-objectifs en nombres entiers (MOILP), nous présentons les notions de base et les définitions nécessaires, ainsi que les principaux résultats liés à la théorie de la programmation linéaire multi-objectifs en nombres entiers [29], [26]. Nous introduisons aussi quelques méthodes de résolution pour ce type de problème et nous donnons une description détaillée de ces méthodes accompagnées chacune par un exemple illustratif.

3.2 Formulation générale du problème

Un programme linéaire multi-objectifs en nombres entiers est constitué d'un système de contraintes linéaires définissant un domaine discret de solutions réalisables, et d'un ensemble de fonction linéaires à maximiser ou à minimiser définissant des objectifs conflic-

tuels entre eux. Le problème consiste à déterminer toute solution réalisable entière telle qu'il n'existe aucune autre solution réalisable entière qui fournisse des valeurs au moins aussi bonnes que celles sur chaque objectif et même meilleures sur au moins un objectif.

Un problème de programmation linéaire multi-objectifs en nombres entiers (MOILP) "Multiple Objective Integer Linear Program" peut être formulé mathématiquement comme suit :

$$(\text{MOILP}) \begin{cases} \text{"Optimiser"} & Z_i(x) = c^i x, \ i = \overline{1, r} \\ \text{s.c.} & x \in D = S \cap \mathbb{Z}^n \end{cases} \quad (3.1)$$

Où :

- $r \geq 2$.
- $S = \{x \in \mathbb{R}^n | Ax \leq b, x \geq 0\}$.
- $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$.
- C est une matrice de dimension $(k \times n)$ formée par $c^i, i = \overline{1, r}$.
- D est l'ensemble des solutions entières dans S .
- $\mathbb{Z}$ est l'ensemble des entiers relatifs.

Remarque 3.1

- Dans la suite, nous supposerons que toutes les fonctions objectifs sont à **maximiser**.

3.3 Solutions efficaces supportées et solutions efficaces non supportées

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

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

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

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

Remarque 3.2

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

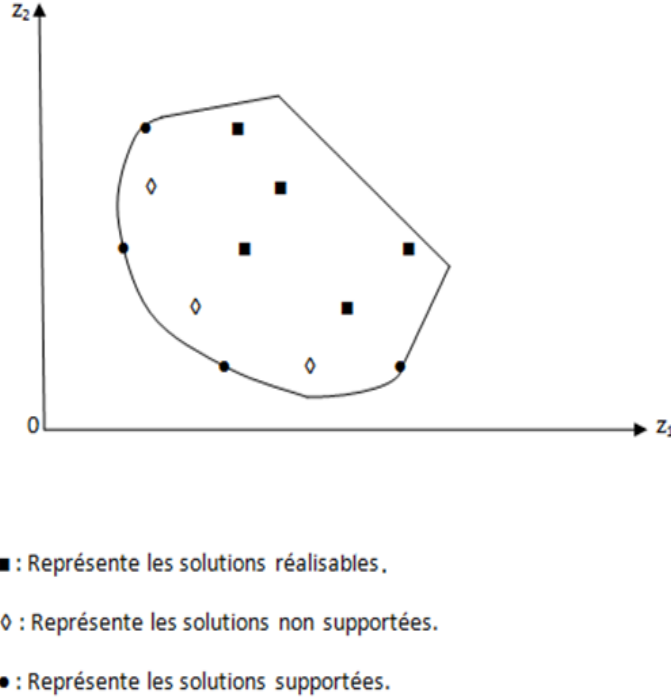


FIGURE 3.1 – Illustration des solutions supportées et solutions non supportées.

3.4 Détection graphique des solutions efficaces

Dans un problème de programmation linéaire multi-objectifs en nombres entiers (MOILP), pour tester graphiquement l'efficacité d'une solution, Ralph Steuer en 1985 [26] a introduit le concept d'ensemble dominant qui est fondé principalement sur la notion de cône.

3.4.1 Définition du Cône

Soit $V \in \mathbb{R}^n$, $V \neq 0_{\mathbb{R}^n}$, V est un cône si et seulement si :

$$\forall x \in V \text{ et pour tout } \alpha \geq 0, \alpha \cdot x \in V \quad (3.2)$$

Remarque 3.3

- Le vecteur origine $0_{\mathbb{R}^n}$ est contenu dans chaque cône.
- Tous les cônes sont non bornés.

3.4.2 Définition du cône positif et du cône négatif

- Un cône négatif est défini par :

$C^- = \{x \in \mathbb{R}^n | F(x) \in \mathbb{R}^k \text{ avec } F(x) \leq 0\}$ (Pour un problème multi-objectifs à minimiser).

- Un cône positif est défini par :

$C^+ = \{x \in \mathbb{R}^n | F(x) \in \mathbb{R}^k \text{ avec } F(x) \geq 0\}$ (Pour un problème multi-objectifs à maximiser).

Théorème du contact [12]

Une solution $x \in \mathbb{R}^n$ est efficace pour un problème multi-objectifs si et seulement si :

- $(C^- + x) \cap D = \{x\}$ (Pour un problème multi-objectifs à minimiser).
- $(C^+ + x) \cap D = \{x\}$ (Pour un problème multi-objectifs à maximiser).

Ce théorème fournit un test permettant de détecter les solutions efficaces graphiquement, c'est à dire :

- Si $D_x \cap D = \{x\}$ alors x est efficace pour le problème multi-objectifs.
- Si $D_x \cap D \neq \{x\}$ alors x n'est pas efficace pour le problème multi-objectifs.

Où : $D_x = C^- + x$ ou $D_x = C^+ + x$ est appelé “**Ensemble Dominant**”.

3.4.3 Exemple illustratif

Soit à déterminer graphiquement l'ensemble des solutions efficaces du programme linéaire (P) à deux objectifs et à deux variables bornées entières suivant :

$$(P) \begin{cases} \max & Z(x) = (z_1 = x_1; z_2 = -x_1 + 3x_2) \\ (x_1, x_2) \in D = \{(x_1, x_2) \in \mathbb{Z}_+^n / x_1 + 2x_2 \leq 7; x_1 \leq 5; x_2 \leq 3\} \end{cases}$$

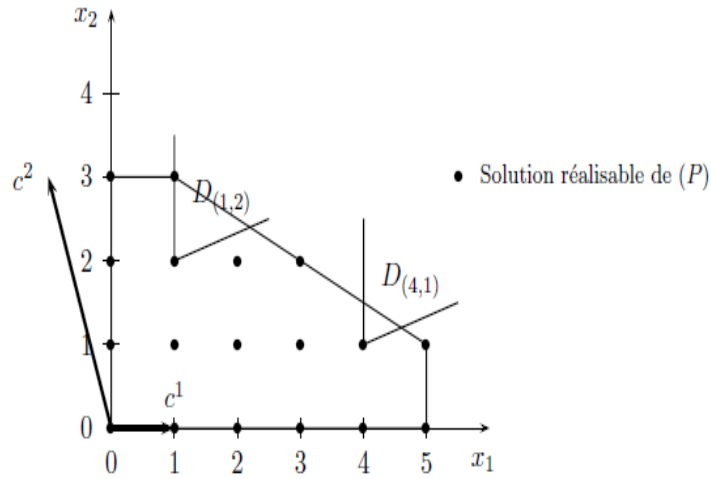


FIGURE 3.2 – Détection graphique des solutions efficaces de (P).

La FIGURE 3.2 nous montre les points réalisables de (P), ainsi que les vecteurs gradients c^1 et c^2 des fonctions objectifs z_1 et z_2 respectivement. Pour chaque point $(x_1, x_2)^t$ de D , $D_{(x_1, x_2)}$ représente l'ensemble de dominance en ce point.

Comme le montre la FIGURE 3.2, la solution réalisable $(1, 2)^t$ n'est pas efficace pour le problème (P) car $D_{(1,2)} \cap D = \{(1, 2)^t, (1, 3)^t\} \neq \{(1, 2)^t\}$. Par contre, la solution $(4, 1)^t$ est efficace puisque $D_{(4,1)} \cap D = \{(4, 1)^t\}$.

En suivant ce raisonnement, on obtient l'ensemble des solutions efficaces de (P) représentées par des carrés blancs et noirs dans la FIGURE 3.3 suivante.

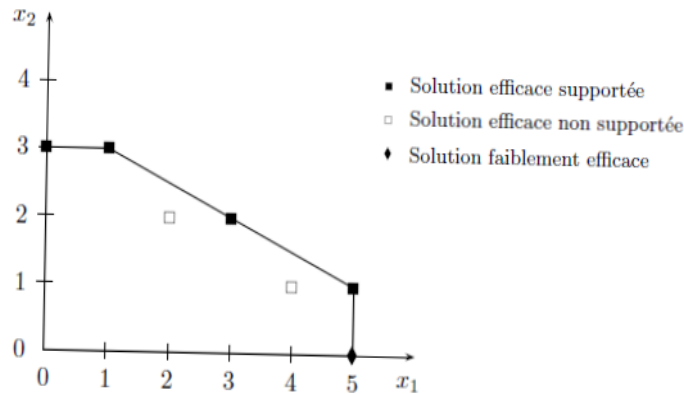


FIGURE 3.3 – Représentation des différents types de solutions dans l'espace de décision.

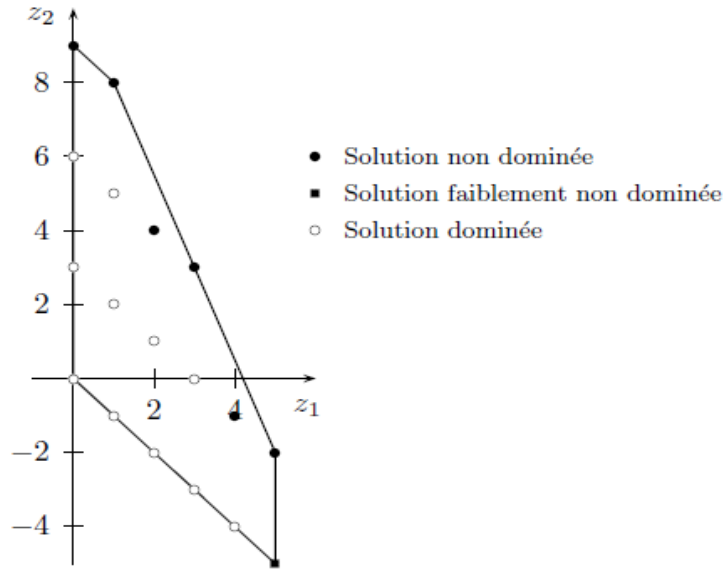


FIGURE 3.4 – Représentation des différents types de solutions dans l'espace des critères.

Dans la FIGURE 3.4, on a une illustration des différents types de solutions (dominées, faiblement non dominées et non dominées). Il est évident que la solution $(5, -5)^t$ est faiblement non dominée puisqu'il n'existe pas de solution $z = (z_1, z_2)$ qui appartient à l'ensemble des points réalisables dans l'espace des critères telle que $z_1 > 5$.

3.5 Quelques méthodes de résolution des problèmes (MOILP)

La difficulté particulière dans les problèmes (MOILP) est la caractérisation de l'ensemble des solutions efficaces non supportées. De ce fait, nombreuses stratégies et applications sont proposées, qui consiste à caractériser l'ensemble des solutions efficaces tant supportées que non supportées.

Dans cette partie, nous présentons en détail trois méthodes de résolution :

Méthode de A.Crema et J.Sylva [17], Méthode de M.Moulai et M.Abbas [22] et la Méthode de F.Z.Ouail, M.E-A.Chergui et M.Moulai [7].

3.5.1 Méthode de A.Crema et J.Sylva [17]

La méthode développée par A.Crema et J.Sylva est une variante de celle de Klein et Hannan [4]. Son principe repose sur la résolution d'une succession de programmes linéaires en nombres entiers optimisant à chaque étape une combinaison positive des critères.

Un ensemble de contraintes est rajouté à chaque fois assurant la détection d'une nouvelle solution efficace. A la fin, la méthode fournit l'ensemble de toutes les solutions non dominées du problème de programmation linéaire en nombres entiers à objectifs multiples.

Proposition 1 (A.Crema et J.Sylva [17])

Soient $x^1, x^2, \dots, x^l$ des solutions efficaces du problème (MOILP). Posons :

$D_s = \{x \in \mathbb{Z}^n / C^k x \leq C^k x^s, \forall s = \overline{1, l}\}$. Si x^* est une solution efficace pour le problème :

$$(P_l) \begin{cases} \max & C^k x \\ \text{s.c.} & x \in (D \setminus \bigcup_{s=1}^l D_s) \end{cases} \quad (3.3)$$

Alors x^* est une solution efficace pour le problème (MOILP). De plus, si le problème (P_l) est irréalisable, alors $\{(C^k x^s)_{s=1}^l\}$ est l'ensemble de toutes les solutions non dominées du problème (MOILP).

Proposition 2 (A.Crema et J.Sylva [17])

Soient $x^1, x^2, \dots, x^l$ des solutions efficaces du problème (MOILP). Posons :

$D_s = \{x \in \mathbb{Z}^n / C^k x \leq C^k x^s, \forall s = \overline{1, l}\}$. Si x^* est une solution optimale pour le problème mono-objectif :

$$(P_l^\lambda) \begin{cases} \max & \sum_{k=1}^p \lambda_k C^k x \\ \text{s.c.} & x \in (D \setminus \bigcup_{s=1}^l D_s) \end{cases} \quad (3.4)$$

Pour certaines valeurs du vecteur $\lambda \in \mathbb{R}^p$, $\lambda > 0$, alors x^* est une solution efficace pour le problème (MOILP).

Théorème 3.1 (A.Crema et J.Sylva [17])

Si x^* est une solution optimale du problème mono-objectif :

$$(P_\lambda) \begin{cases} \max & \sum_{k=1}^p \lambda_k C^k x \\ \text{s.c.} & x \in S \end{cases} \quad (3.5)$$

Pour un certain $\lambda \in \Lambda = \{\lambda \in \mathbb{R}^p : \sum_{k=1}^p \lambda_k = 1\}$, alors, x^* est une solution efficace du problème multi-objectifs.

Algorithme de la méthode

Étape 1.

Après avoir fixé le vecteur poids $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_p)$ à des valeurs strictement positives, la première étape de l'algorithme consiste en la résolution du problème :

$$(P_1) : \max \left\{ \sum_{k=1}^p \lambda_k C^k x \mid x \in D \right\} \quad (3.6)$$

Deux cas se présentent :

Cas 1.

Si (P_1) n'admet pas de solutions, alors le problème (MOILP) l'est aussi.

Cas 2.

Sinon, une solution x^1 est trouvée et elle est efficace (d'après le théorème 3.1)

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

Après k étapes du processus :

Cas 1.

Si (P_l) est irréalisable, alors l'algorithme prend fin.

Cas 2.

Sinon, une nouvelle solution efficace, soit x^l , est trouvée et le nouveau problème (P_{l+1}) est défini à partir de (P_l) en lui éliminant toutes les solutions vérifiant $C^k x \leq C^k x^l, \forall k = \overline{1, p}$.

Ceci peut être traduit par le rajout de contraintes suivantes :

$$\begin{cases} C^k x \geq (C^k x^l + 1)y_k^l - M_k(1 - y_k^l), & k = \overline{1, p}. \\ \sum_{k=1}^p y_k^l \geq 1, y_k^l \in \{0, 1\}, & k = \overline{1, p}. \end{cases} \quad (3.7)$$

Où $-M_k$ est la borne inférieure pour les valeurs réalisables de la $k^{ème}$ fonction objectif.

Étape générale ($l+1$)

Résoudre le problème (P_{l+1})

$$(P_{l+1}) \left\{ \begin{array}{l} \max \quad \sum_{k=1}^p \lambda_k C^k x \\ \text{s.c.} \quad x \in D \\ C^k x \geq (C^k x^s + 1)y_k^s - M_k(1 - y_k^s) \\ \sum_{k=1}^p y_k^s \geq 1, y_k^s \in \{0, 1\} \text{ pour } k = \overline{1, p}, s = \overline{1, l} \end{array} \right. \quad (3.8)$$

Étape finale n :

La procédure continue jusqu'à ce que le problème (P_n) devient irréalisable.

A la fin, on obtient l'ensemble de toutes les solutions efficaces ou seulement une partie qui intéresse le décideur.

3.5.2 Méthode de M.Abbas et M.Moulai [22]

Cette technique est basée sur la méthode des coupes, elle est appropriée à la résolution d'un problème (MOILP), la première itération consiste à déterminer une solution optimale entière réalisable pour le problème relâché et ceci en considérant un seul objectif.

La seconde itération nous permet de trouver toutes les solutions entières qui alternent la solution entière optimale obtenue ci-dessus, si elle existe, et obtenir la première solution entière efficace du problème principal. Dans la prochaine étape, nous introduisons une coupe afin de calculer les autres solutions entières efficaces restantes et ceci en utilisant la méthode duale du simplexe. En arrivant à l'étape finale, l'algorithme aura exploité toutes les solutions entières efficaces.

Soit le problème (MOILP) suivant :

$$(MOILP) \left\{ \begin{array}{l} \max Z_1(x) = c^1 x \\ \max Z_2(x) = c^2 x \\ \cdot \\ \cdot \\ \cdot \\ \max Z_r(x) = c^r x \\ \text{s.c.} \quad x \in S = \{x \in \mathbb{R}^n / Ax \leq b, x \geq 0\} \\ x \text{ entier} \end{array} \right. \quad (3.10)$$

Nous supposons que S est un polyèdre borné non vide de $\mathbb{R}^n$.

Soit (P_1) le problème linéaire uni-critère suivant :

$$(P_1) \begin{cases} \max & Z_1(x) = c^1 x \\ \text{s.c.} & x \in S \\ & x \text{ entier} \end{cases} \quad (3.11)$$

Notons qu'à la place du problème (P_1) on pourra considérer d'une manière analogue le problème (P_1) avec un objectif Z_i pour un indice quelconque $i = \overline{2, k}$.

Introduisons alors les notations suivantes :

$$S_1 = \{x \in \mathbb{R}^{n_1} / A_1 x \leq b_1, A_1 \in \mathbb{R}^{m_1 \times n_1}, b_1 \in \mathbb{R}^{m_1}, x \geq 0\}$$

S_1 est la région courante tronquée de S obtenue après de successives coupes de Gomory.

$x_1^1 = (x_{1,j}^1)$ est la solution entière optimale sous Z_1^1 obtenue sur S_1 .

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

Z_i^1 est la valeur de Z_i , $i = \overline{1, k}$ correspondant à la solution x_1^1 .

B_1^1 est une base de S_1 .

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

$I_1 = \{i : a_{1,j}^1 \in B_1^1\}$ indices de base de B_1^1 .

$N_1 = \{j : a_{1,j}^1 \notin B_1^1\}$ indices hors base de B_1^1 .

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

$$Z_1(x_1^1) = c^1 x_1^1.$$

$\Gamma_1 = \{j : j \in N_1 \text{ et } Z_{1,j}^1 - c_j^1 = 0\}$ où $Z_{1,j}^1 = c_{B_1^1}^1 y_{1,j}^1$, c_j^1 est la $j^{\text{ème}}$ composante du vecteur c^1 et $c_{B_1^1}^1$ est le vecteur coût des variables de base associées à B_1^1 dans le vecteur c^1 .

Pour $k \geq 2$,

$$S_k = \{x \in \mathbb{R}^{n_k} = A_k x \leq b_k; A_k \in \mathbb{R}^{m_k \times n_k}; b_k \in \mathbb{R}^{m_k}; x \geq 0\}.$$

S_k est la région courante tronquée de S obtenues après l'application de la coupe

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

où $j_{k-1} \in \Gamma_{k-1}$ et d'éventuelles successives coupes de Gomory.

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

$x_k^q = \{x_{k,j}^q\}$ sont les (P_{k-1}) solution entière, lorsqu'elles existes alternatives à x_k^1 .

où p_k est un nombre entier et $q \in \{2, \dots, p^k\}$.

B_k^1 est une base de de S_k .

$a_{k,j}^1 \in \mathbb{R}^{m_k \times 1}$ sont les vecteurs activités de $x_{k,j}^1$ sous S_k .

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

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

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

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

$Z_i(x_k^1) = c^i x_k^1$.

$\Gamma_k = \{j : j \in N_k \text{ et } Z_{k,j}^1 - C_j^1 = 0\}$ où $Z_{k,j}^1 = c_{B_k^1}^1 y_{k,j}^1$; c_j^1 est la $j^{\text{ème}}$ correspondante du vecteur c^1 et $c_{B_k^1}^1$ est le vecteur coût des variables de base associées à B_k^1 dans le vecteur c^1 .

Ainsi, Γ_k est l'ensemble correspondant à la solution x_k^1 .

Définition 3.1

Une arête $E_{k,j}$ incidente à la solution x_k^1 est définie par l'ensemble :

$$E_{k,j} = \left\{ x = x_i \in \mathbb{R}^n \left| \begin{array}{l} x_i = x_i^k - \theta_{jk} y_{k,ijk}, i \in I_k \\ x_{jk} = \theta_{jk} \\ x_j = 0 \text{ pour tout } j \in N_k \setminus \{j_k\} \end{array} \right. \right\} \quad (3.12)$$

Où :

$$0 \leq \theta_{jk} \leq \left\{ \min_{i \in I_k} \frac{x_{k,i}^1}{y_{k,ijk}^1} ; y_{k,ijk} \geq 0 \right\} \quad (3.13)$$

θ_{jk} est un entier positif et $\theta_{jk} y_{k,ijk}$ sont des entiers pour tout $i \in I_k$, si de telles valeurs existent.

Résultats préliminaires

Puisque la résolution d'un problème (MOILP) consiste à déterminer l'ensemble des solutions efficaces entières, alors on parcourt tous les points entiers de l'ensemble S en utilisant la technique des coupes

Remarque 3.5

La coupe $\sum_{j \in N_k \setminus \{j_k\}} x_j \geq 1$, tronquant l'arête E_{jk} relative à $j_k \in \Gamma_k$, est une forme modifiée de la coupe de Dantzig, alors que la coupe de Dantzig $\sum_{j \in N_j} x_j \geq 1$ pour laquelle $\Gamma_k = \emptyset$ tronque un point.

Algorithme de la méthode

Étape 1.

Résoudre le problème (P_1) et retrouver la solution optimale entière x_1^1 sur S_1

Construire Γ_1 .

Étape 2.

1/ Si $\Gamma_1 = \emptyset$, x_1^1 est l'unique solution optimale du problème sur S_1 et Z_1^1 est la valeur optimale de Z_1 , calculer les valeurs Z_i^1 de Z_i données par x_1^1 , $i = \overline{2, r}$

— Enregistrer le premier r-uplet efficace comme étant $(Z_1^1, Z_2^1, \dots, Z_r^1)$ pour construire l'ensemble efficace Eff_0 .

— Tronquer le point x_1^1 par la coupe de Dantzig.

En appliquant la méthode duale du simplexe et, si nécessaire, des coupes successives de Gomory, on obtient une solution réalisable entière $x_2^1 = (x_{2,j}^1)$ dans la région tronquée S_2 .

— Lire le r-uplet correspondant $(Z_1, Z_2, \dots, Z_r)$ et ajouter à Eff_0 s'il n'est pas dominé par l'un des r-uplets efficaces identifiés auparavant pour obtenir Eff_1 .

2/ Si $\Gamma_1 \neq \emptyset$, on choisit un indice quelconque $j \in \Gamma_1$ et calculer le nombre

$$\theta_{j_1} \leq \min_{i \in I_1} \{x_{1,i}^1 / y_{1,i,j_1}^1; y_{1,i,j_1}^1 > 0\}$$

correspondant à x_1^1 .

i) Si $\theta_{j_1} \geq 1$, déterminer toutes les solutions entières réalisables x_1^q , $q = \overline{2, r}$ alternatives à x_1^1 le long de l'arête E_{j_1} . Chaque solution apporte une amélioration au r-uplet potentiellement non dominé de la forme $(Z_1, Z_2, \dots, Z_r)$.

Eff_0 est l'ensemble des r-uplets potentiellement non dominés à l'étape (2).

Tronquer l'arête E_{j_1} par la coupe de Dantzing améliorée suivante :

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

Et ceci en appliquant la méthode duale du simplexe et de successives coupes de Gomory, pour obtenir une solution réalisable entière $x_2^1 = (x_{2,j}^1)$ dans la région tronquée S_2 .

Lire le r-uplet correspondant $(Z_1, Z_2, \dots, Z_r)$ et l'ajouter à Eff_0 s'il n'est pas dominé par l'un des r-uplets efficaces précédemment identifiés pour obtenir Eff_1 .

ii) Si $\theta_{j_1} < 1$ pour tous les indices $j_1 \in \Gamma_1$, choisir un indice $j_1 \in \Gamma_1$ et appliquer la coupe :

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

En utilisant la méthode duale du simplexe et de successives coupes de Gomory, on obtient alors une solution réalisable entière $x_2^1 = (x_{2,j}^1)$ dans S_2 .

Lire le r-uplet correspondant $(Z_1, Z_2, \dots, Z_r)$ et le rajouter à l'ensemble Eff_0 s'il n'est pas dominé par l'un des r-uplets efficaces précédemment déterminés pour obtenir Eff_1 .

Étape 3.

- Choisir un indice $j_2 \in I_2$ et déterminer toutes les solutions entières réalisables x_2^q , $q = \overline{2, p_2}$ alternatives à x_2^1 , lorsqu'elles existent sur E_{j_2} .

Lire les r-uplets obtenus, augmenter l'ensemble Eff_1 , par les nouveaux r-uplets non dominés pour construire Eff_2 . Donc Eff_2 est l'ensemble de tous les potentiels r-uplets efficaces à l'étape (3).

- Tronquer l'arête E_{j_2} en appliquant la coupe

$$\sum_{j \in N_2 \setminus \{j_2\}} x_j \geq 1$$

et chercher la solution optimale entière dans la région tronquée courante pour passer à l'étape (4) de la procédure.

Étape k.

- Choisir $j_{k-1} \in I_{k-1}$ et explorer l'arête correspondante $E_{j_{k-1}}$ pour d'éventuelles solutions entières réalisables x_{k-1}^q , $q = \overline{2, p_{k-1}}$ alternatives à x_{k-1}^1 . Augmenter Eff_{k-2} par les nouveaux r-uplets non dominés pour construire Eff_{k-1} .
- L'arête $E_{j_{k-1}}$ est tronquée par la coupe.

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

En utilisant la méthode duale du simplexe et d'éventuelles successives coupes de Gomory. La solution optimale entière obtenue sur la région tronquée S_k sera soit une solution réalisable de (P_1) alternative à x_{k-1}^1 ou la prochaine meilleure solution x_k^1 ou un point non entier. Ceci marque le début de l'étape (k+1).

Étape finale

La procédure se termine quand le pivot n'est pas réalisable, indiquant que la région courante contient des solutions réalisables non entières et que tous les points efficaces ont été déterminés.

Théorème 3.2

La procédure décrite ci-dessus pour la génération de tous les points efficaces converge en un nombre fini d'étapes.

Exemple illustratif

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

$$(P) \left\{ \begin{array}{l} \max \quad Z_1 = C^1 x = x_1 + 2x_2 \\ \max \quad Z_2 = C^2 x = 3x_1 - 2x_2 \\ \max \quad Z_3 = C^3 x = -x_1 + 2x_2 \\ s.c. \quad x_1 + x_2 \leq 7 \\ \quad \quad 2x_1 \leq 11 \\ \quad \quad 2x_2 \leq 7 \\ \quad \quad x_1, x_2 \geq 0 \\ \quad \quad x_1, x_2 \in \mathbb{Z}_+ \end{array} \right.$$

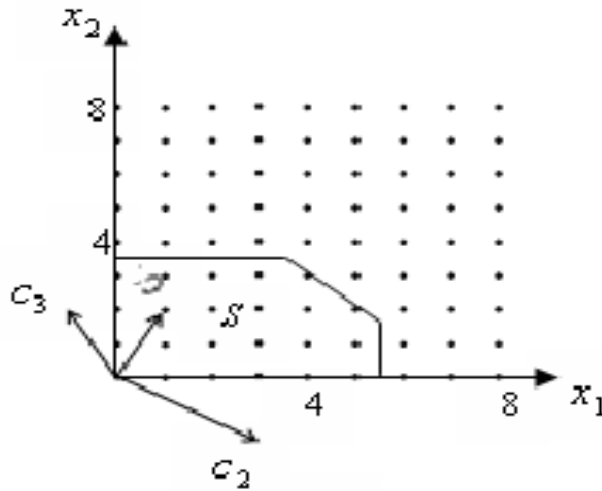


FIGURE 3.5 – Graphe de l'exemple.

Étape (1)

Résoudre le problème relaxé (P_1) suivant :

$$(P_1) \left\{ \begin{array}{l} \max \quad Z_1 = C^1 x = x_1 + 2x_2 \\ s.c. \quad x_1 + x_2 \leq 7 \\ \quad \quad 2x_1 \leq 11 \\ \quad \quad 2x_2 \leq 7 \\ \quad \quad x_1, x_2 \geq 0 \\ \quad \quad x_1, x_2 \in \mathbb{Z}_+ \end{array} \right.$$

La solution de base réalisable optimale du problème (P_1) est donnée par la TABLE suivante :

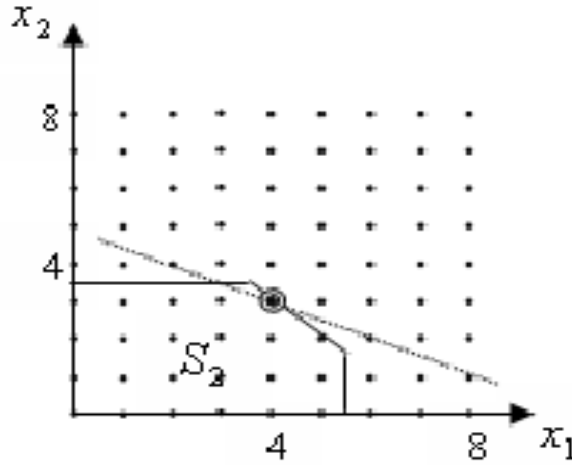
base	x_1	x_2	x_3	x_4	x_5	x_6	b
x_1	1	0	1	0	0	-1	4
x_4	0	0	-2	0	0	2	3
x_2	0	1	0	0	0	1	3
x_5	0	0	0	1	1	-2	1
$z_j^1 - c_j^1$	0	0	1	0	0	1	10
$z_j^2 - c_j^2$	0	0	3	0	0	-5	6
$z_j^3 - c_j^3$	0	0	-1	0	0	3	2

TABLE 3.1 –

La solution optimale de (P_1) est $x_1^1 = (4, 3)^t$, et elle produit le premier triplet efficace $(10, 6, 2)^t$. On a donc $Eff_0 = \{x_1^1 = (4, 3)^t\}$.

Étape (2)

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

FIGURE 3.6 – La troncature du point x_1^1 .

En utilisant l'algorithme dual du simplexe on obtient une solution réalisable $x_2^1 = (3, 3)^t$ donnée par la TABLE 3.2 optimale suivante :

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

TABLE 3.2 –

Le triplet $(9, 3, 3)^t$ qui correspond à $x_2^1 = (3, 3)^t$ est non dominé, par conséquent :

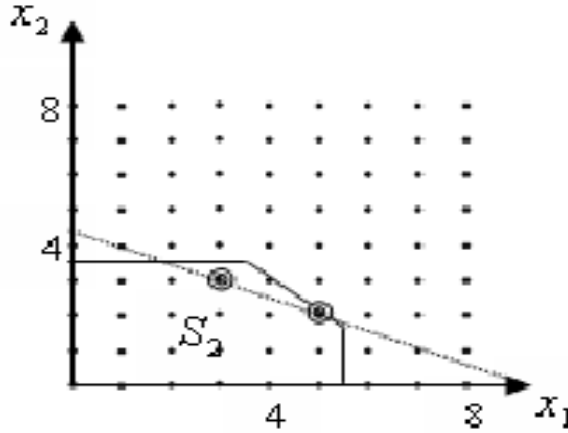
$$Eff_1 = \{x_1^1 = (4, 3)^t, x_2^1 = (3, 3)^t\}.$$

Étape (3)

$\Gamma_2 \neq \emptyset$, x_2^1 n'est pas unique prendre, $j_2 = 6$, on a $\phi_{j_2} \leq \min \left\{ \frac{5}{4}, \frac{3}{1}, \frac{1}{1} \right\} = 1$. La solution alternative $x_2^2 = (5, 2)^t$ donne le triplet non dominé $(9, 11, -1)^t$. Donc on augmente

$$Eff_2 = Eff_1 \cup \{x_2^2 = (5, 2)^t\}.$$

L'arête E_6 est tronquée par la coupe $x_7 \geq 1$, $\implies -x_7 + x_8 = -1$, comme le présente la FIGURE suivante :

FIGURE 3.7 – La troncature de l'arête E_6 .

En appliquant l'algorithme dual du simplexe, on obtient une solution entière réalisable $x_3^1 = (2, 3)$, le triplet $(8, 0, 4)^t$ correspondant à la solution x_3^1 est non dominée donc :

$$Eff_3 = \{Eff_2 \cup x_3^1 = (2, 3)\}.$$

Étape (4)

$\Gamma_3 \neq \emptyset$, x_3^1 n'est pas unique, prendre $j_2 = 6$, on a $\phi_{j_3} \leq \min \left\{ \frac{7}{4}, \frac{3}{1}, \frac{2}{1} \right\} = \frac{7}{4}$. La solution alternative $x_3^2 = (4, 2)^t$ donne le triplet non dominé $(8, 8, 0)^t$, donc on augmente

$$Eff_4 = Eff_3 \cup \{x_3^2 = (4, 2)^t\}$$

De la même manière on obtient :

$Eff_9 = \{(4, 3)^t, (3, 3)^t, (5, 2)^t, (2, 3)^t, (4, 2)^t, (1, 3)^t, (5, 1)^t, (0, 3)^t, (5, 0)^t\}$ est une solution entière réalisable, $x_{10}^1 = (1, 0)^t$, le triplet $(1, 3, -1)^t$ correspondant à la solution x_{10}^1 est dominé. Donc $Eff_9 = Eff_{10}$.

Étape (11)

$\Gamma_{10} \neq \emptyset$ donc, x_{10}^1 n'est pas unique, prendre $j_{10} = 18$, on a $\phi_{j_{10}} \leq \min \left\{ \frac{9}{4}, \frac{0}{1}, \frac{6}{1} \right\} = 0$. La solution x_{10}^1 ne possède pas de solution alternative après sa troncature, le dual du simplexe donne une solution entière réalisable $x_{11}^1 = (0, 0)^t$ donnée par la TABLE optimale suivante :

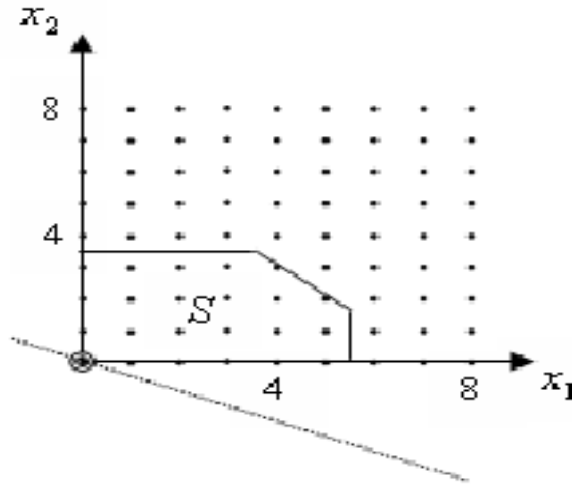
base	x_1	x_2	x_3	x_4	x_5	x_6	x_{18}	x_{19}	b
x_1	1	0	0	0	0	0	-2	1	0
x_4	0	0	0	1	0	0	4	-2	11
x_2	0	1	0	0	0	0	1	0	0
x_5	0	0	0	0	1	0	-2	0	7
x_3	0	0	1	0	0	0	1	-1	7
x_6	0	0	0	0	0	1	-1	0	3
$z_j^1 - c_j^1$	0	0	0	0	0	0	0	1	0
$z_j^2 - c_j^2$	0	0	0	0	0	0	-8	3	0
$z_j^3 - c_j^3$	0	0	0	0	0	0	4	-1	0

TABLE 3.3 –

Étape (12)

$\Gamma_{11} \neq \emptyset$, x_{11}^1 n'est pas unique, prendre $j_{11} = 18$, on a $\phi_{j_{11}} \leq \min \left\{ \frac{11}{4}, \frac{0}{1}, \frac{7}{1} \right\} = 0$.

La solution x_{11}^1 ne possède pas de solution alternative après la troncature de l'arête E_{18} , comme le présente la FIGURE 3.8 suivante :

FIGURE 3.8 – La troncature de l'arête E_{18} .

Le dual du simplexe donne une solution non réalisable donnée par la Table optimale suivante :

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

TABLE 3.4 –

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

$$Ef_{10} = \{(4, 3)^t, (3, 3)^t, (5, 2)^t, (2, 3)^t, (4, 2)^t, (1, 3)^t, (5, 1)^t, (0, 3)^t, (5, 0)^t\}$$

Et le front de Pareto est donné par :

$$FP = \{(10.6, 2)^t, (9, 3, 3)^t, (9, 11, -1)^t, (8, 0, 4)^t, (8, 8, 0)^t, (7, -3, 5)^t, (7, 13, -3)^t, (6, -6, 6)^t, (5, 15, -5)^t\}.$$

3.5.3 Méthode de F.Z.Ouail, M.E-A.Chergui et M.Moulai [7]

Cette méthode est basée sur le principe de Branch & Bound couplé à une nouvelle coupe efficace pour la recherche de solutions entières [7]. C'est une nouvelle approche exacte développée pour générer toutes les solutions efficaces du problème (MOILP) sans avoir à calculer toutes les solutions entières possibles.

Définitions et notations

Soit le problème (MOILP) suivant :

$$(\text{MOILP}) \left\{ \begin{array}{l} \max \quad Z_1(x) = c^1 x \\ \max \quad Z_2(x) = c^2 x \\ \cdot \\ \cdot \\ \cdot \\ \max \quad Z_r(x) = c^r x \\ s.c \quad x \in S = \{x \in \mathbb{R}^n / Ax \leq b, x \geq 0\} \\ x \text{ entier} \end{array} \right.$$

Où :

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

— On suppose que le domaine S est non vide et compact.

On définit aussi le programme paramétrique suivant :

$$(P_l) \left\{ \begin{array}{l} \max \quad Z(x) = \sum_{q=1}^r \lambda_q c^q x \\ x \in S_l \end{array} \right. \quad (3.14)$$

Avec $\lambda_q \geq 0, \forall q = \overline{1, r}$ dont les valeurs sont fixées à priori.

Soit x_l^* une solution entière de (PL) obtenue après le processus de branchement, on associe à x_l^* les notations et définitions suivantes :

- B_l : Ensemble des indices des variables de base.
- N_l : Ensemble des indices des variables hors base.
- H_l : Ensemble des directions possibles d'amélioration des critères.

$$H_l = \{j \in N_l \mid \exists q = \overline{1, r} \hat{c}_j^q > 0\} \cup \{j \in N_l \mid \hat{c}_j^q = 0, \forall q = \overline{1, r}\}$$

Avec $\hat{c}_j^q$ est la $j^{\text{ème}}$ composante du coût réduit du critère q .

- Contrainte supplémentaire appelée coupe efficace.

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

- Les ensembles suivants :

$$S_0 = S$$

$$S_{l+1} = \{x \in S_l \mid \sum_{j \in H_l} x_j \geq 1\}$$

$$T_{l+1} = \{x \in S_l \mid \sum_{j \in N_l \setminus H_l} x_j \geq 1\}$$

Définition 4.1

Une coupe est dite efficace pour le problème (MOILP), si son adjonction au domaine S supprime au moins une solution réalisable continue de S , mais ne supprime pas de solutions réalisables entières efficaces de S .

Notons que la définition de coupe efficace pour (MOILP) généralise la définition d'une coupe pour un problème (PLNE).

Principe de la méthode

L'idée de la méthode repose sur le balayage du domaine des solutions réalisables S afin de détecter des solutions entières potentiellement efficaces. Quand c'est possible, on peut éliminer des parties du domaine S qui ne contiennent pas de solutions entières efficaces par l'ajout de la contrainte supplémentaire appelée coupe efficace $\sum_{j \in H_l} x_j \geq 1$.

Le problème (P_0) est résolu avec la méthode du simplexe et le tableau utilisé est augmenté de k nouvelles lignes pour pouvoir évaluer les critères en même temps que le critère $Z(x)$ de (P_0) . Toutes les opérations sont identifiées dans des nœuds l et des branches dans un arbre. A chaque nœud l de l'arborescence, nous devons résoudre un programme (P_l) .

Le nœud l est sondé si le programme (P_l) correspondant n'est pas réalisable ou si $H_l = \emptyset$. Si la solution optimale x^* de (P_l) n'est pas entière, on passe alors au processus de séparation : soit x_j une coordonnée de x^* telle que $x_j = \alpha_j$ avec α_j est fractionnaire. On crée deux nouveaux nœuds à partir du nœud l qu'on séparera en deux grâce à la contrainte supplémentaire : $x_j \leq [\alpha_j]$ et $x_j \geq [\alpha_j] + 1$ où $[\alpha_j]$ est la partie entière de α_j .

De là, chaque branche correspondante définit un nouveau programme à résoudre.

Algorithme de la méthode

Etape 1. (Initialisation)

$S_0 := S$; $l := 0$ et $Eff := \emptyset$; (Eff est l'ensemble des solutions efficaces de (MOILP))

- Résoudre le programme linéaire (P_0) au nœud 0. Soit $\bar{x}_0$ la solution optimale obtenue.
- Si $\bar{x}_0$ n'est pas entière, aller à l'étape 2a. Sinon, aller à l'étape 2b.

Etape 2. (Étape générale)

Tant qu'il existe un nœud non sondé dans l'arborescence faire :

- Choisir le nœud l le plus récemment créé non encore sondé et résoudre le programme linéaire correspondant (P_l) .
- Si (P_l) est non réalisable, alors le nœud l est sondé.
- Sinon, soit $\bar{x}_l$ une solution optimale de (P_l) .

- Si $\bar{x}_l$ n'est pas entière, aller à l'étape 2a.
- Sinon, poser x_l^* la solution entière trouvée et aller à l'étape 2b.

Etape 2a. Soit $\bar{x}_{lj}$ une coordonnée fractionnaire de $\bar{x}_l$.

- Séparer le noeud l en deux nouveaux noeuds : ajouter la contrainte $x_j \leq \lfloor \bar{x}_{lj} \rfloor$ au premier noeud, la contrainte $x_j \geq \lfloor \bar{x}_{lj} \rfloor + 1$ au second noeud et aller à l'étape 2.

Etape 2b. Si Cx_l^* n'est pas dominé par Cy pour toute solution $y \in \text{Eff}$,

alors $\text{Eff} := \text{Eff} \cup \{x_l^*\}$.

- S'il existe $y \in \text{Eff}$ telle que Cy est dominé par Cx_l^* ,

alors $\text{Eff} := \text{Eff} \setminus \{y\} \cup \{x_l^*\}$.

- Déterminer les ensembles B_l , N_l et H_l .
- Si $H_l = \emptyset$ alors le noeud correspondant est sondé, aller à l'étape 2.
- Sinon
 - Rajouter la contrainte $\sum_{j \in H_l} x_j \geq 1$ pour obtenir l'ensemble S_{l+1} ,
 - Résoudre le programme linéaire obtenu (P_{l+1}) par la méthode duale du simplexe et poser $\bar{x}_{l+1}$ la solution optimale trouvée.
 - Si $\bar{x}_{l+1}$ est entière, poser x_{l+1}^* la solution entière trouvée et aller à l'étape 2b,
 - Sinon, aller à l'étape 2a.

Exemple illustratif

Considérons l'exemple suivant dans $\mathbb{R}^2$:

$$(P) \begin{cases} \max & z_1(x) = 3x_1 - x_2 \\ \max & z_2(x) = -x_1 + x_2 \\ & 4x_1 + 3x_2 \leq 20 \\ & x_1 - x_2 \leq 3 \\ & x_2 \leq 4 \\ & x \in \mathbb{R}_+^2, x \text{ entier} \end{cases}$$

$$\text{Eff} = \emptyset$$

Pour $l = 0$, Construisons le programme (P_0) :

On fixe à priori les valeurs de λ_q : $\lambda_1 = \lambda_2 = 1$

$$(P_0) \begin{cases} \max & Z = \sum_{q=1}^2 \lambda_q c^q x = 2x_1 \\ & x \in S_0 = S \end{cases}$$

Le programme (P_0) est résolu en effectuant la somme des z_i , $i = \overline{1,2}$ sans contrainte d'intégrité des variables. La solution optimale $(\frac{29}{7}, 7, \frac{8}{7})^t$ est obtenue dans la TABLE du simplexe suivante :

B	b	x_3	x_4
x_2	$\frac{8}{7}$	$\frac{1}{7}$	$-\frac{4}{7}$
x_1	$\frac{29}{7}$	$\frac{1}{7}$	$\frac{3}{7}$
x_5	$\frac{20}{7}$	$-\frac{1}{7}$	$\frac{4}{7}$
$-z^1$	$-\frac{79}{7}$	$-\frac{2}{7}$	$-\frac{13}{7}$
$-z^2$	3	0	1

TABLE 3.5 –

La séparation est déclenchée avec la création de deux nœuds. Le nœud 1 est sondé car en rajoutant la contrainte $x_1 \geq 5$; le problème est non réalisable. Au nœud 2, la contrainte $x_1 \geq 4$ est rajoutée au TABLE 3.5 et la solution entière optimale $(4, 1)^t$ est obtenue dans la TABLE 3.6 suivante, après avoir utilisé le dual du simplexe :

B	b	x_4	x_6
x_2	1	-1	1
x_1	4	0	1
x_5	3	1	-1
x_3	1	3	-7
$-z^1$	-11	-1	-2
$-z^2$	3	1	0

TABLE 3.6 –

$Eff = \{(4, 1)^t\}$, $N_2 = \{4, 6\}$ et $H_2 = \{4\}$.

La coupe efficace $x_4 \geq 1$ est alors ajoutée et la solution optimale $(\frac{26}{7}, \frac{12}{7})^t$ est donnée dans la TABLE 3.7 suivante :

B	b	x_3	x_7
x_2	$\frac{12}{7}$	$\frac{1}{7}$	$-\frac{4}{7}$
x_1	$\frac{26}{7}$	$\frac{1}{7}$	$\frac{3}{7}$
x_5	$\frac{16}{7}$	$-\frac{1}{7}$	$\frac{4}{7}$
x_6	$\frac{2}{7}$	$-\frac{1}{7}$	$-\frac{3}{7}$
x_4	1	0	-1
$-z^1$	$-\frac{66}{7}$	$-\frac{2}{7}$	$-\frac{13}{7}$
$-z^2$	2	0	1

TABLE 3.7 –

Séparation en deux nouveaux nœuds car la solution n'est pas entière. La contrainte $x_2 \leq 1$ est rajoutée à la TABLE 3.7 au nœud 3 et le nœud 4 est créé après adjonction de la contrainte $x_2 \geq 2$ à la TABLE 3.7. La solution optimale $(3, 1)^t$ est obtenue au nœud 3.

B	b	x_7	x_8
x_2	1	0	7
x_1	3	1	7
x_5	3	0	-1
x_6	1	-1	-1
x_4	1	-1	0
x_3	5	4	-7
z^1	-8	-3	-2
z^2	2	1	0

TABLE 3.8 –

$$Eff = \{(4, 1)^t, (3, 1)^t\}, N_3 = \{7, 8\} \text{ et } H_3 = \{7\}$$

On rajoute la coupe efficace $x_7 \geq 1$ et la solution entière optimale $(2, 1)^t$ est obtenue dans la TABLE 3.9 suivante :

B	b	x_8	x_9
x_2	1	7	0
x_1	2	7	1
x_5	3	-1	0
x_6	2	-1	-1
x_4	2	0	-1
x_3	9	-7	-4
x_7	1	0	-1
z^1	-5	-2	-3
z^2	1	0	1

TABLE 3.9 –

$Eff = \{(4, 1)^t, (3, 1)^t, (2, 1)^t\}$, $N_4 = \{8, 9\}$ et $H_4 = \{9\}$.

On rajoute la coupe efficace $x_9 \geq 1$, ... etc

L'exploration en profondeur à partir du noeud 3, donne la dernière TABLE 3.10 suivante :

B	b	x_8	x_{12}
x_2	1	7	0
x_1	-1	7	1
x_5	3	-1	0
x_6	5	-1	-1
x_4	5	0	-1
x_3	21	-7	-4
x_7	4	0	-1
x_9	3	0	-1
x_{10}	2	0	-1
x_{11}	1	0	-1
$-z^1$	4	-2	-3
$-z^2$	-2	0	1

TABLE 3.10 –

Et le nœud correspondant est sondé car le dual n'est pas réalisable.

$Eff = \{(4, 1)^t, (3, 1)^t, (2, 1)^t\}$

Au nœud 4, la contrainte $x_2 \geq 2$ est rajoutée à la TABLE 3.7. La solution optimale non entière $(\frac{7}{2}, 2)^t$ est obtenue, les nœuds 5 et 6 sont créés, le nœud 5 est sondé compte tenu de la contrainte $x_1 \geq 4$.

Au nœud 6, la contrainte $x_1 \leq 3$ est rajoutée et la solution entière optimale $(3, 2)^t$ est obtenue.

$Eff = \{(4, 1)^t, (3, 1)^t, (3, 2)^t\}$, $N_6 = \{8; 9\}$ et $H_6 = N_6$.

La contrainte $x_8 \geq 1$ est rajoutée. En procédant de cette manière, dans la TABLE 3.11 on obtient la solution entière optimale $(2, 4)^t$ au dernier nœud 8 :

B	b	x_{11}	x_{12}
x_2	4	0	-1
x_1	2	1	0
x_5	0	0	1
x_6	2	-1	0
x_4	5	-1	-1
x_7	4	-1	-1
x_9	1	-1	0
x_8	2	0	-1
x_3	0	-4	3
x_{10}	1	0	-1
$-z^1$	-2	-3	-1
$-z^2$	-2	1	1

TABLE 3.11 –

$Eff = \{(4, 1)^t, (3, 1)^t, (3, 2)^t, (2, 4)^t\}$, $N_9 = \{11, 12\}$ et $H_9 = N_9$.

On rajoute la contrainte $x_{11} + x_{12} \geq 1$ et on obtient la dernière TABLE 3.12 :

B	b	x_3	x_{13}
x_2	5	0	-1
x_1	$\frac{5}{4}$	$\frac{1}{4}$	$\frac{3}{4}$
x_5	-1	0	1
x_6	$\frac{11}{4}$	$-\frac{1}{4}$	$-\frac{3}{4}$
x_4	$\frac{27}{4}$	$-\frac{1}{4}$	$-\frac{7}{4}$
x_7	$\frac{23}{4}$	$-\frac{1}{4}$	$-\frac{7}{4}$
x_9	$\frac{7}{4}$	$-\frac{1}{4}$	$-\frac{3}{4}$
x_8	3	0	-1
x_{11}	$\frac{3}{4}$	$-\frac{1}{4}$	$-\frac{3}{4}$
x_{10}	2	0	-1
x_{12}	1	0	-1
$-z^1$	$\frac{5}{4}$	$-\frac{3}{4}$	$-\frac{13}{4}$
$-z^2$	$-\frac{15}{4}$	$\frac{1}{4}$	$\frac{7}{4}$

TABLE 3.12 –

Le dual n'est pas réalisable, donc le nœud 8 est sondé.

Comme tous les nœuds de l'arborescence sont sondés alors, l'algorithme s'arrête et l'ensemble efficace complet est trouvé :

$Eff = \{(4, 1)^t, (3, 1)^t, (3, 2)^t, (2, 2)^t, (2, 3)^t, (2, 4)^t, (1, 4)^t, (0, 4)^t\}$.

3.6 Conclusion

A travers ce chapitre, nous avons pris connaissance des problèmes de programmation linéaire multi-objectifs en nombres entiers et la formule mathématique de ce type de problème. Ainsi, nous avons présenté les deux types de solutions efficaces qu'on retrouve dans les problèmes (MOILP). Puis, nous avons introduit dans ce chapitre les notions de bases nécessaires pour la détection graphique des solutions efficaces. En dernier, trois méthodes de résolution d'un problème (MOILP) ont été présentées, dont la dernière servira comme support pour la résolution de notre problème principal au chapitre quatre.

Chapitre 4

Optimisation d'une Fonction Linéaire sur l'Ensemble Efficace d'un Problème Multi-Objectifs Discret

4.1 Introduction

Souvent en pratique, l'énumération de tout l'ensemble des solutions efficaces d'un problème linéaire multi-objectifs n'est pas évident, dans le sens où la taille de l'ensemble des solutions efficaces est très vaste ce qui cause la saturation du décideur pour le choix de sa solution préférée.

Afin d'éviter ces situations, il est utile de considérer l'optimisation d'une fonction linéaire sur l'ensemble des solutions efficaces étant non vide. De là, la notion de points efficients joue un rôle important dans l'analyse et la résolution de ces problèmes. La difficulté principale de ce type de problèmes est due à la non convexité de l'ensemble des solutions efficaces. Par conséquent, le problème appartient à la classe des problèmes d'optimisation globale (programmation non convexe) [12].

Philip [16] en 1972 était le premier dans ce domaine, il a étudié ce problème dans le cas continu et a suggéré deux procédures pour sa résolution, décrivant dans un algorithme basé sur le déplacement sur les sommets efficaces adjacents.

Récemment, plusieurs chercheurs, citons en particulier : Benson [10], [11], [12], Isermann et Steuer [9], Yamamoto [33], Ecker et Song [14], Sayin [28], et autres, motivés par de nombreuses applications [10], [13], se sont intéressés au problème d'optimisation d'une fonction linéaire sur l'ensemble efficace d'un problème multi-objectifs linéaire.

Contrairement au cas continu qui a été étudié et développé au début par plusieurs auteurs, le cas discret a récemment vu un développement. En effet, ce n'est qu'en 1992 que la première méthode pour l'optimisation d'un critère linéaire sur l'ensemble efficace à va-

riables entières a vu le jour, proposé par Nguyen [23]. En 2006, une méthode proposée pour l'optimisation sur l'ensemble efficace d'un problème (MOILP) tout en évitant l'énumération explicite de tous les points efficaces dans l'espace des variables de décision, est proposée par M.Abbas et D.Chaabane [19].

4.2 Rappels et préliminaires

Rappelons la formulation du problème (MOILP) étudié au chapitre 3 :

$$(\text{MOILP}) \begin{cases} \max & Z_i(x) = c^i x, \ i = \overline{1, r} \\ \text{s.c.} & x \in D = S \cap \mathbb{Z}^n \end{cases} \quad (4.1)$$

Où :

- $r \geq 2$
- $S = \{x \in \mathbb{R}^n / A.x \leq b, x \geq 0\}$.
- A est une matrice de dimension $(m \times n)$.
- $b \in \mathbb{R}^m$.
- D est l'ensemble des solutions entières dans S .
- C est une matrice de dimension $(r \times n)$ formée par le vecteur $c^i, i = \overline{1, r}$.

On note par Eff l'ensemble de toutes les solutions efficaces entières du problème (MOILP).

Le problème de l'optimisation d'une fonction linéaire sur l'ensemble des solutions efficaces Eff d'un problème (MOILP) est écrit sous la forme suivante :

$$(\text{OI}) \begin{cases} \max & \varphi(x) = dx \\ \text{s.c.} & x \in Eff \end{cases} \quad (4.2)$$

Où :

- d est un vecteur ligne de dimension n de composantes entières.
- La fonction $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}$ est une fonction linéaire continue.

La fonction φ est en général différente des critères du problème (MOILP) mais elle peut être prise comme une combinaison linéaire de ces derniers.

Formules et notions

On considère le problème $(OI)_l$ de programmation linéaire suivant à l'étape l , $l \geq 0$, de la méthode proposée par la suite :

$$(OI)_l \begin{cases} \max & \varphi(x) \\ \text{s.c.} & x \in S_l \end{cases} \quad (4.3)$$

On note par x_{opt} la solution optimale efficace de (OI) trouvée jusqu'à l'étape l et φ_{opt} correspond à la valeur optimale de la fonction objectif.

$S_0 = S$ et S_{l+1} est obtenue de S_l en ajoutant des coupes de Type I et de Type II décrites ci-dessous. Soit x_l^* la première solution entière obtenue en résolvant $(OI)_l$ en utilisant le processus de branchement étudié dans la méthode de *Branch & Bound*.

Par la suite, nous présentons quelques formules et notations nécessaires pour la présentation de la méthode de résolution :

- On note par $B_l(N_l)$ l'ensemble des indices de variables de base (de variables hors base respectivement) de x_l^* .
- On note par $\bar{c}_j^i$ la $j^{\text{ème}}$ composante du vecteur de coût réduit c^i pour tout $i = \overline{1, r}$ dans le dernier tableau simplexe .
- on fait appel au processus de coupes pour résoudre le problème (OI), pour cela deux types de coupes ont été construites. En effet, on construit la coupe efficace notée **coupe de Type I**, définie comme suit :

$$\sum_{j \in H_l} x_j \geq 1 \quad (4.4)$$

où l'ensemble H_l est défini par :

$$H_l = \{j \in N_l / \exists i = \overline{1, r}; \bar{c}_j^i > 0\} \cup \{j \in N_l / \bar{c}_j^i = 0, \forall i = \overline{1, r}\} \quad (4.5)$$

Le principe de cette coupe repose sur le balayage du domaine des solutions réalisables en éliminant les solutions non efficaces sans avoir à les énumérer.

D'autre part, **la coupe de type II** est construite selon l'inégalité suivante :

$$\varphi(x) \geq \varphi_{opt} \quad (4.6)$$

Cette coupe permet de supprimer des points qui nous intéressent peu en terme d'optimalité.

- Définissons les deux ensembles suivants au nœud l

De type I (dans une solution entière) :

$$S_{l+1}^1 = \left\{ x \in S_l / \sum_{j \in H_l} x_j \geq 1 \right\} \quad (4.7)$$

De type II (dans une solution non entière) :

$$S_{l+1}^2 = \{x \in S_l / \varphi(x) \geq \varphi_{opt}\} \quad (4.8)$$

- Ainsi, Considérons l'ensemble suivant :

$$S_{l+1} = S_{l+1}^1 \cup S_{l+1}^2 \quad (4.9)$$

- Eff_l est l'ensemble des solutions potentiellement efficaces de (MOILP) obtenue jusqu'à l'étape l . Les solutions qui appartiennent à Eff_l sont des solutions entières réalisables telle qu'aucune d'entre elles ne dominent l'autre dans l'espace des critères. L'ensemble Eff_l est mis à jour, en testant à chaque fois une solution entière x_l^* atteinte, si $Z(x_l^*)$ est dominé ou pas. Si $Z(x_l^*)$ n'est pas dominé par aucun vecteur $Z(x)$, $x \in Eff_{l-1}$, alors $Eff_l = Eff_{l-1} \cup \{x_l^*\}$ et enlever aussi les solutions de Eff_l dont le vecteur critère est dominé par $Z(x_l^*)$ ou par $Z(y_l)$, y_l solution du programme $(P_{x_l^*})$ donné ci-dessous.

4.3 Test d'efficacité

Dans la procédure proposée, on est appelé à tester l'efficacité des solutions réalisables. Nous avons jugé utile de montrer que le résultat déjà appliqué pour le cas des variables de décision continues, est vrai aussi pour le cas des variables discrètes. Son utilisation est très simple, à chaque fois qu'on veut tester l'efficacité d'une solution donnée, on résout un problème de programmation linéaire inchangé dans tous ces paramètres, sauf le vecteur second membre des contraintes, qu'il faut remplacer par la solution en question. Le théorème suivant est utilisé dans les différentes étapes de l'algorithme pour tester l'efficacité d'une solution réalisable donnée du problème mixte comme il est rapporté dans les articles [20], [15], [23].

Théorème Isermann

Soit x^* un élément arbitraire de la région D. x^* est une solution efficace si et seulement si la valeur optimale de la fonction objectif Θ est nulle dans le problème de programmation linéaire mixte suivant :

$$(P_{x_l^*}) \begin{cases} \max & \Theta = \sum_{i=1}^r \psi_i \\ \text{t.q} & Cx - I\psi = Cx^* \\ & x \in D \\ & \psi_i \in \mathbb{R}^+, \forall i = \overline{1, r} \end{cases} \quad (4.10)$$

Où :

- C est une matrice $r \times n$, dont la $i^{\text{ème}}$ ligne correspond à c^i , $i = \overline{1, r}$.
- I est la matrice d'identité ($r \times r$).
- $\psi = (\psi_i)_{i=\overline{1, r}}$.

Preuve

Soit $(x^*) \in \text{Eff}$, comme $\psi_i \geq 0$ pour tout $i \in \mathcal{J} = \{1, \dots, r\}$, alors $Cx \geq Cx^*$ dans (P_{x^*}) . Supposons que $\max \Theta \neq 0$, alors il existe $i \in \mathcal{J}$ tel que $\psi_i > 0$, $I\psi \geq 0$ et $I\psi \neq 0$ ce qui implique $Cx > Cx^*$. Ainsi il existe un $x \in D$ tel que $Cx \geq Cx^*$ et $Cx \neq Cx^*$. Ce qui est en contradiction avec l'hypothèse selon laquelle x^* est efficace.

Soit $\max \Theta = 0$, supposons que x^* est non efficace. Il existe un point $x \in D$ tel que $Cx \geq Cx^*$ et $Cx \neq Cx^*$. Par conséquent, il existe alors x tel que $Cx - Cx^* \geq 0_{\mathbb{R}^n}$ et $Cx - Cx^* \neq 0_{\mathbb{R}^n}$, d'où $I\psi > 0$, par conséquent il existe i tel que $\psi_i > 0$ ce qui est en contradiction avec l'hypothèse $\max \Theta = 0$.

Ce théorème peut être aussi utilisé pour générer un point efficace même si le point x^* ne l'est pas. En effet, dans le cas où la valeur optimale de la fonction objectif Θ est non nulle, alors toute solution optimale y de $(P_{x_l^*})$ est efficace.

4.4 Méthode de résolution (Méthode de F.Z.Ouail, M.E-A.Chergui et M.Moulai) [21]

Le cas discret n'a pas été aussi bien traité que le cas continu dans la résolution des problèmes d'optimisation d'une fonction linéaire sur l'ensemble efficace à variables entières d'un problème (MOILP). En effet, le plus grand challenge pour ce type de problème est sa classification, il est classé parmi les problèmes NP-Difficiles, ainsi que le peu d'articles dédiés à ces problèmes.

Cette partie sera entièrement consacrée à la résolution du programme (OI) par la méthode de F.Z.Ouail, M.E-A.Chergui et M.Moulai [21]. La description de cette méthode est comme suit :

4.4.1 Principe de la méthode

Comme nous l'avons cité précédemment au chapitre 3, la méthode de Ouail, Chergui et Moulai [36] est basée sur la technique de Branch & Bound, elle est renforcée par des coupes efficaces et des tests d'intégrité, ce qui permet la recherche de la solution optimale. L'algorithme proposé génère une solution au problème principal à résoudre (OI) en un nombre fini d'itérations sans avoir à énumérer l'ensemble des solutions efficaces Eff . On commence par résoudre le programme $(OI)_l$ en utilisant la méthode du simplexe (ou éventuellement la méthode du dual du simplexe). Si la solution obtenue n'est pas entière, le test d'intégrité est déclenché. Lorsqu'on atteint une solution x_l^* entière, on ajoute de nouvelles coupes au dernier tableau simplexe, ceci permet de réduire considérablement la zone de recherche des solutions.

Le nœud l est sondé si : la valeur optimale de φ_{opt} obtenue jusqu'à l'étape l est supérieure ou égale à la valeur de φ au nœud l .

4.4.2 Formulation de l'algorithme

L'algorithme de la méthode de F.Z.Ouail, M.E-A.Chergui et M.Moulai [21] est présenté comme suit :

Étape 1. (Initialisation)

Poser $l := 0$; $S_0 := S$; la valeur optimale de la fonction objective $\varphi_{opt} = -\infty$;

Eff_l est l'ensemble des solutions efficaces potentielles du problème (MOILP) ;

On pose $Eff_l := \emptyset$.

Étape 2. (Étape principale)

Tant qu'il n'y a pas un nœud saturé dans l'arborescence, faire :

- Résoudre le programme $(OI)_l$ en utilisant la méthode du simplexe (ou éventuellement la méthode du dual du simplexe) et aller à l'étape 3.1.

Étape 3. (Tests)

Étape 3.1. (Test de réalisation)

- Si le programme $(OI)_l$ n'est pas réalisable, alors on arrête et le nœud l est sondé.
- Sinon, soit x_l^* une solution optimale de $(OI)_l$,
 - Si $\varphi_{opt} \geq \varphi(x_l^*)$, alors le nœud l est sondé.
 - Sinon, aller à l'étape 3.2.

Étape 3.2. (Test d'intégrité)

- Si x_l^* est une solution entière, alors mettre à jour l'ensemble des solutions efficaces Eff_l et aller à l'**étape 3.3**.
- Sinon, aller à l'**étape 4**.

Étape 3.3. (Test d'efficacité)

- Si x_l^* n'est pas incluse dans Eff_l , alors x_l^* n'est pas efficace et aller à l'**étape 5**.
- Résoudre $(P_{x_l^*})$.
 - Si x_l^* est efficace alors mettre à jour $\varphi_{opt} = \varphi(x_l^*)$ et $x_{opt} = x_l^*$. Le nœud l est sondé, aucune amélioration de φ n'est possible.
 - Sinon, soit y_l la solution de $(P_{x_l^*})$, y_l entière efficace. Mettre à jour si nécessaire φ_{opt} , x_{opt} et ainsi que l'ensemble Eff_l . Aller à l'**étape 5**.

Étape 4. (Branchement)

Choisir x_j une coordonnée de x_l^* tel qu'on pose $x_j = \alpha_j$ est un nombre fractionnaire. Séparer le nœud l en deux nouveaux nœuds et obtenir deux sous programmes de $(OI)_l$ en ajoutant : la contrainte $x_j \leq [\alpha_j]$ pour obtenir le programme $(OI)_{l_1}$, et la contrainte $x_j \geq [\alpha_j] + 1$ pour obtenir le programme $(OI)_{l_2}$, où $l_1 > l+1$, $l_2 > l+1$, $l_1 \neq l_2$ et $[\alpha_j]$ est la partie entière de x_j , aller à l'**étape 2**. On ajoute la coupe $\varphi(x) \geq \varphi_{opt}$.

Étape 5. (Coupe efficace)

Construire l'ensemble H_l .

- Si $H_l = \emptyset$, alors le nœud l est sondé puisqu'il n'y aura pas de solution efficace par la suite.
- Sinon, construire l'ensemble S_{l+1}^1 (en ajoutant la coupe de Type I), aller à l'**étape 2**.

4.5 Résultats fondamentaux

Dans le but de justifier les différentes étapes de l'algorithme cité précédemment, une sélection de résultats théoriques ont été établis, ainsi pour mieux comprendre le déroulement de la méthode.

On note par D_l l'ensemble $D_l = S_l \cap \mathbb{Z}^n$.

Théorème 4.1

On suppose qu' à l'obtention de la solution entière x_l^* l'ensemble $H_l \neq \emptyset$.

Si $x \neq x_l^*$ est une solution optimale pour le programme (OI) dans le domaine S_l ,

alors $x \in S_{l+1}$.

Preuve :

Soit $x \neq x_l^*$ une solution entière dans le domaine S_l tel que $x \notin S_{l+1}$,

alors $x \notin S_{l+1}^1$ et $x \notin S_{l+1}^2$.

- Si $x \notin S_{l+1}^1$, alors $x \in \left\{ x \in S_l \mid \sum_{j \in N_l \setminus H_l} x_j \geq 1 \right\}$

Par conséquent, les coordonnées de x vérifient les inégalités suivantes :

$$\begin{aligned} & \circ \sum_{j \in H_l} x_j < 1 \\ & \circ \sum_{j \in N_l \setminus H_l} x_j \geq 1 \end{aligned}$$

De plus, $x_j = 0$ pour tout $j \in H_l$, et $x_j \geq 1$ pour au moins un indice $j \in N_l \setminus H_l$.

En utilisant le tableau simplexe de la solution x_l^* , les égalités suivantes sont par tous les critères $i = \overline{1, r}$.

$$\begin{aligned} c^i x &= c^i x_l^* + \sum_{j \in N_l} \hat{c}_j^i x_j \\ \implies c^i x &= c^i x_l^* + \sum_{j \in H_l} \hat{c}_j^i x_j + \sum_{j \in N_l \setminus H_l} \hat{c}_j^i x_j \\ \implies c^i x &= c^i x_l^* + \sum_{j \in N_l \setminus H_l} \hat{c}_j^i x_j \end{aligned}$$

Ainsi, on obtient $c^i x \leq c^i x_l^*$ pour tous les critères $i = \overline{1, r}$, avec $c^i x < c^i x_l^*$ pour au moins un critère où $\hat{c}_j^i \leq 0$ pour tout $j \in N_l \setminus H_l$.

Au final, on conclut que la solution x n'est pas une solution efficace, de là toutes les solutions entières efficaces appartiennent au domaine S_{l+1}^1 (*)

- Si $x \in S_{l+1}^2$, alors x n'est pas optimal, et ceci représente une contradiction (**)

D'après (*) et (**), on conclut que $x \in S_{l+1}$.

Théorème 4.2

Soit x_l^* la solution entière actuelle du programme $(OI)_l$, et si x_l^* est efficace pour le programme (P) , alors x_l^* est une solution optimale pour le programme (OI) sur D_l .

Preuve :

On suppose que x_l^* n'est pas optimale pour le programme (OI). Donc $\exists x \in D_l, x \neq x_l^*$ telle que $\varphi > \varphi_{opt}$ introduit par le théorème 4.1 . Comme, x_l^* est une solution efficace alors $\varphi_{opt} \geq \varphi(x_l^*)$, ainsi $\varphi(x) \geq \varphi(x_l^*)$. D'autre part, dans le tableau simplexe actuel, l'expression de φ peut être écrite sous la forme suivante :

$$\begin{aligned}\varphi(x) &= \varphi(x_l^*) + \sum_{j \in N_l} \hat{\varphi}_j x_j \geq \varphi(x_l^*) \\ \implies \varphi(x_l^*) + \sum_{j \in N_l} \hat{\varphi}_j x_j &\geq \varphi(x_l^*) \\ \implies \sum_{j \in N_l} \hat{\varphi}_j x_j &> 0.\end{aligned}$$

Ceci contredit le fait que $\hat{\varphi}_j \leq 0, \forall j \in N_l$.

Proposition 4.1

Si $H_l = \emptyset$, alors $\forall x \in D_{l+1}$, x n'est pas efficace.

Preuve :

$H_l = \emptyset$, alors $\forall i = \overline{1, r}, \forall j \in N_l$, on a $\hat{c}_j^i \leq 0$ et $\exists i_0 = \overline{1, r}$ tel que $\hat{c}_j^{i_0} < 0, \forall j \in N_l$. Donc x_l^* domine tous les points x du domaine D_l où $x \neq x_l^*$.

Proposition 4.2

Si $\varphi_{opt} \geq \varphi(x_l^*)$, alors $\exists x \in D_l$ telle que $\varphi(x) > \varphi_{opt}$.

Preuve :

Il est évident que toutes les solutions x pour lesquelles $\varphi(x) < \varphi_{opt}$ ne sont pas intéressantes même si elles sont efficaces et qu'elles donnent une meilleure valeur pour φ .

Théorème 4.3

L'algorithme prend fin après un nombre fini d'itérations et renvoie la solution optimale du programme (OI).

Preuve :

L'ensemble $S = \{x \in \mathbb{R}^n / Ax \leq b, x \geq 0\}$, est l'ensemble des solutions réalisables du programme (MOILP). Étant compact, l'ensemble S contient un nombre fini de solutions entières.

A chaque étape l de l'algorithme, si une solution entière x_l^* est obtenue, on procède à son élimination aussi bien que le sous-ensemble des solutions entières non intéressantes en appliquant le théorème 4.1 par l'ajout des coupes. D'autre part, quatre tests de saturation sont utilisés dans l'algorithme, et cela sans la perte de la solution optimale du programme (OI). Premièrement, lorsque l'ensemble H_l est vide alors la solution correspondante x_l^* est un point idéal et le nœud correspondant peut être taillé (abandonné) car aucun critère ne peut être améliorer. Deuxièmement, si à l'étape l , la solution entière correspondante x_l^* est efficace, alors le nœud correspondant est saturé car x_l^* est optimale pour le programme (OI) sur l'ensemble D_l . Troisièmement, si φ_{opt} (la valeur de la meilleure solution efficace trouvée pour (OI)) est supérieur à celle de la solution optimale sur l'ensemble D_l , alors le nœud l est aussi saturé. Finalement, le cas trivial est quand le domaine réduit devient irréalisable. Par conséquent, l'algorithme converge vers une solution optimale après un nombre fini d'étapes.

4.6 Exemple illustratif

Considérons le problème d'optimisation d'une fonction linéaire sur l'ensemble efficace entier suivant :

$$(OI) \begin{cases} \text{Max } (-x_1 - 2x_2) \\ x \in Eff \end{cases}$$

Où Eff est l'ensemble des solutions efficaces du problème lineaire multi-objectifs entier suivant :

$$(P) \begin{cases} \text{Max } (z_1 = x_1 - 2x_2; z_2 = -x_1 + 4x_2) \\ \text{s.c. } -2x_1 + x_2 \leq 0 \\ x_1 \leq 3 \\ x_2 \leq 2 \\ x_1, x_2 \geq 0 \text{ entiers} \end{cases}$$

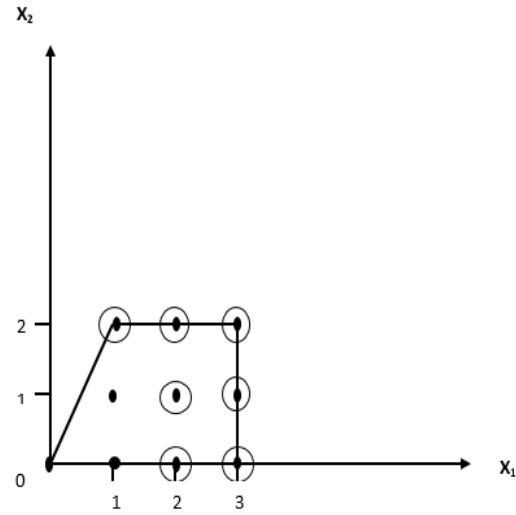


FIGURE 4.1 – Graphe de l'exemple.

Étape 1.

On pose $\varphi_{opt} = -\infty$ et $l = 0$, après avoir résolu le programme $(OI)_0$, la solution optimale obtenue est $x_0^* = (0, 0)^t$, elle n'est pas efficace car en résolvant le problème $(P_{x_0^*})$ la valeur optimale de sa fonction objectif n'est pas nulle. La solution optimale de $(P_{x_0^*})$ est $(3, 1)^t$, elle est efficace.

Mise à jour ; $\varphi_{opt} = -5$ et $x_{opt} = (3, 1)^t$.

$H_0 = \{1, 2\} \neq \emptyset$.

	x_1	x_2	b_i
x_3	-2	1	0
x_4	1	0	3
x_5	0	1	2
$-\varphi$	-1	-2	0
$-c^1$	1	-2	0
$-c^2$	-1	4	0

TABLE 4.1 –

La coupe efficace $x_1 + x_2 \geq 1$ est rajoutée, et en utilisant la méthode duale du simplexe on obtient la TABLE suivante :

	x_6	x_2	b_i
x_3	-2	3	2
x_4	1	-1	2
x_5	0	1	2
x_1	-1	1	1
$-\varphi$	-1	-1	-1
$-c^1$	1	-3	1
$-c^2$	-1	5	-1

TABLE 4.2 –

La solution $x_1^* = (1, 0)^t$ est obtenue mais elle n'est pas efficace (d'après le test d'efficacité), après la résolution de problème $(P_{x_1^*})$ on obtient une solution entière $(3, 1)^t$ qui est efficace.

$H_1 = \{6, 2\} \neq \emptyset$, on applique alors la coupe efficace $x_2 + x_6 \geq 1$ et la méthode duale du simplexe donne :

	x_7	x_3	b_i
x_6	$-\frac{3}{5}$	$-\frac{1}{5}$	$\frac{1}{5}$
x_4	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{13}{5}$
x_5	$\frac{2}{5}$	$-\frac{1}{5}$	$\frac{6}{5}$
x_1	$-\frac{1}{5}$	$-\frac{2}{5}$	$\frac{2}{5}$
x_2	$-\frac{2}{5}$	$\frac{1}{5}$	$\frac{4}{5}$
$-\varphi$	-1	0	-2
$-c^1$	$-\frac{3}{5}$	$\frac{4}{5}$	$-\frac{6}{5}$
$-c^2$	$\frac{7}{5}$	$-\frac{6}{5}$	$\frac{14}{5}$

TABLE 4.3 –

La solution optimale obtenue est $x_2^* = (\frac{2}{5}, \frac{4}{5})^t$ et elle n'est pas entière, donc on applique la technique de *Branch and Bound* :

Soit x_1 la coordonnée fractionnaire de x_2^* , on construit les deux contraintes suivantes : $x_1 \leq [\frac{2}{5}] = 0$ et $x_1 \geq [\frac{2}{5}] + 1 = 1$ avec $[\frac{2}{5}]$ la partie entière de $\frac{2}{5}$.

- Pour $x_1 \leq 0$, le programme $(OI)_1$ est irréalisable, car :

D'après le précédant tableau optimal, on a :

$$\frac{2}{5} = x_1 - \frac{2}{5}x_3 - \frac{1}{5}x_7$$

$$\Rightarrow x_1 = \frac{2}{5} + \frac{2}{5}x_3 + \frac{1}{5}x_7$$

On remplace la valeur de x_1 dans la contrainte :

$$x_1 \leq 0 \Rightarrow \frac{2}{5} + \frac{2}{5}x_3 + \frac{1}{5}x_7 \leq 0 \text{ ce qui impossible.}$$

- Pour $x_1 \geq 1$ la solution optimale trouvée $x_3^* = (1, \frac{1}{2})^t$ qui est non entière.

	x_8	x_3	b_i
x_6	$-\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$
x_4	1	0	2
x_5	$-\frac{1}{2}$	$\frac{1}{2}$	$\frac{3}{2}$
x_1	-1	0	1
x_2	$\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$
x_3	$-\frac{5}{2}$	$\frac{1}{2}$	$\frac{3}{2}$
$-\varphi$	0	-1	-2
$-c^1$	2	-1	0
$-c^2$	-3	2	1

TABLE 4.4 –

Deux sous-programmes créés (processus de branchement) :

- Pour $x_2 \leq 0$, on obtient la TABLE suivante :

	x_7	x_9	b_i
x_6	-1	-1	1
x_4	1	2	1
x_5	0	-1	2
x_1	-1	-2	2
x_2	0	1	0
x_3	-2	-5	4
x_8	-1	-2	1
$-\varphi$	-1	0	-2
$-c^1$	1	4	2
$-c^2$	-1	-6	-2

TABLE 4.5 –

La solution optimale obtenue est $x_4^* = (2, 0)^t$ qui est entière et efficace (test d'efficacité), ainsi $\varphi_{opt} = -2$, $x_{opt} = (2, 0)^t$

- Pour $x_2 \geq 1$ on a :

	x_9	x_8	b_i
x_6	-1	-1	1
x_4	0	1	2
x_5	1	0	1
x_1	0	-1	1
x_2	-1	0	1
x_3	1	-2	1
x_7	2	-1	1
$-\varphi$	-2	-1	-3
$-c^1$	-2	1	-1
$-c^2$	4	-1	3

TABLE 4.6 –

La solution optimale est $x_5^* = (1, 1)^t$ et ce nœud est saturé car $\varphi_{opt} = -2$, $\varphi(x_5^*) = -3$ donc $\varphi_{opt} \geq \varphi(x_5^*)$, ainsi la solution optimale pour (OI) est $x_{opt} = (2, 0)^t$ et $\varphi_{opt} = -2$.

4.7 Conclusion

Dans ce chapitre, notre étude est portée sur une nouvelle méthode exacte qui optimise une fonction linéaire sur l'ensemble efficaces à variables entières d'un problème (MOILP), combinant le principe de Branch & Bound avec deux types de coupe. La première consiste à éviter la recherche des solutions dans des domaines qui ne contiennent pas des solutions efficaces, la deuxième consiste à éliminer les domaines qui ne contiennent pas les solutions optimales.

Par ailleurs, cette méthode a l'avantage d'être appliquer même si les coefficients de la fonction objectif sont réelles.

Nous avons d'abord introduit quelques rappels jugés nécessaires pour présenter la formulation mathématique du problème principal à résoudre. Ensuite, nous avons présenté l'algorithme de la méthode ainsi que les résultats fondamentaux. En dernier, un exemple numérique illustre les différentes étapes de la méthode.

Conclusion générale

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

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

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

- Le premier volet consiste à rappeler la notion d'optimisation multi-objectifs linéaire à variables discrets (MOILP).
- Le deuxième volet consiste à résoudre un problème d'optimisation où les contraintes représentent l'ensemble des solutions efficaces du problème (MOILP) du premier volet.

Bibliographie

- [1] A.Berro, “Optimisation Multi-objectifs et Stratégie d’Evolution en Environnement Dynamique”, Thèse de doctorat, université des sciences sociale de Toulouse, 2001.
- [2] A.Channe and W.Cooper, “Management Models and Industrial Application of Linear Programming”, Vol 1, John Wiley, New-York, 1961.
- [3] D.Moumita and P.K.De, “A note on Fuzzy Multiobjective Linear Fractional Programming Problem”, International journal of Computers Science and Network, p.568-572, 2014.
- [4] D.Klein and E.Hannan, “An Algorithm for the Multiple Objective Integer Linear Programming Problem”, European Journal of Operation Research, 9, 378-385, 1982.
- [5] F.Kebaili, H.Mouloud, “Optimisation de le Gestion de Portefeuille d’Actifs Financiers, memoire de fin d’étude”, université de USTHB, 2008.
- [6] F.Y.Edgeworth, “Mathematical Physics”, Keagan.P, Londres, Angleterre, 1881.
- [7] F.Z.Ouail, M.E-A.Chergui and M.Mouali, “Solving the Multiple Objective Integer Linear Programming Problem, Modeling, Computation and Optimization in Information Systems and Management Sciences Communications in Computer and Information Science”, Volume 14, Part 1, p.69-76, 2008.
- [8] G.B.Dantzing, “On a Linear Programming Combinatorial Approach to the Traveling Salesman Problem”, Operations Research, 7, p.58-66, 1959.
- [9] H.Isermann and R.E.Steuer, Computational Experience Concerning Payoff Tables and Minimum Criterion Values over the Efficient Set, European Journal of Operational Research 33, 91-97, 1987.
- [10] H.P.Benson, “Optimization over the Efficient Set”, Journal of Mathematical Analysis and Applications 98, 562-580, 1984.
- [11] R.E.Steuer, “Multiple Criteria Optimization : Theory, Computation and Application”, Wiley series in Probability and Mathematical Statistics, New-York, John Wiley & Sons, Inc. XXII, p.546, 1986.

- [12] H.P. “ A finite Non Adjacent Extreme Point Search Algorithm Over the efficient Set “ Journal of Optimization Theory and Application “ Vol. 73-64, 1992.
- [13] J.A.Fulop, “ Lutting Plane Algorithm for Linear Opt Over the Efficient Set”, Lecture Notes in Economics an Mathematical Systems, Springer- Verlag, Berlin, 374-385,1994.
- [14] J.G.Ecker and .J.H.Song, Optimization a Linear Function over an Efficient Set, Journal of Optimization Theory and Applications, Volume 83, n°3, 541-563, 1994.
- [15] J.M.Jorge, ” An algorithm for Optimizing a linear Function Over an Integer Efficient Set “European Journal of Opérational Research, 10. 1016, 2008.
- [16] J.Philip, “Algorithms for the Vector Maximization Problem”, Mathematical Programming 2, p.207-229, 1972.
- [17] J.Sylva and A.Crema, “A Method of Finding the Set of Non-dominated Vectors for Multiple Objective Integer Linear Programs”, European Journal of Perational Research, 158, p.46-55, 2004
- [18] K.Miettinen, “Non Linear Multiobjective Optimization”, Kluwer, Boston, 1999.
- [19] M.Abbas and D.Chabane, ” An Algorithm for solving Multiple Objective Inetegr Linear Programming Problem, RAIRO Opérations Research 36, 351-364, 2006.
- [20] M.Abbas et D.Chaabane, “ Optimizing a linear function Over an integer efficient set”, European Journal of Opérational Research 174, 2006 , pp. 1140-1161.
- [21] M.E-A. Chergui, M.Moulai et F.Z. Ouail, “ An Exact Method for Optimizing a Linear Function Over an Integer efficient Set “, Volume 16, 2017.
- [22] M.Moulai and M.Abbas, “Integer Linear Fractional Programming with Multiple Objective”, Journal of the Italian Operation Research Society, Vol 1_103-104, p.15-38, 2002.
- [23] N.C.Nguyen, “ An Algorithm for Opti a Linear Function Over the Integer Efficient Set”, Konrad-Zuse-Zentrum for informations technik Berlin, 1992.
- [24] Pareto Vilfredo,”cours d'économie politique”,Vol.1 et 2, Rouge Lausanne, 1896.
- [25] R.Benagad, S.Hamadi, “Optimisation des Coûts Carburants Air Algerie”, Thèse ingénieur, université USTHB, 2008.
- [26] R.E.Steuer, “Multiple Criteria Optimization : Theory, Computationand Application”, Wiley series in Probability an Mathematical Statistics, New-York, John Wiley & Sons, Inc. XXII, p.546, 1986.
- [27] R.E.Gomory, “Outline of an Algorithm for Integer Solutions to Linear Programs”, Bulletin of the AMS 64, p.275-278, 1958.

- [28] S. Sayin , “Opti Over the Efficient Set Using a Topdown Search of faces”, *Opérational Research* 48, 65-72, 2000.
- [29] V.Chankong and Y.Y.Haimes, “Multiobjective Decision Making : Theory and Methodology”, North Holland Series in System Science and Engeneering, p.406, 1983.
- [30] Y.Collette, P.Siarry, “Optimisation Multi-objectifs”, édition Eyrolles 61, 2002.
- [31] Y.Y.Haimes, “Modeling and Control of the Pollution of Water Resources Systems via Multilevel Approch”, New-York, 1971.
- [32] Y.Y.Haimes, L.S.Lasdon and D.A.Wismer, “Formulation of the Problems of Integrated System Identification and System Optimization”, 1986.
- [33] Y.Yamamoto, Optimization over the Efficient Set : Overview, *Journal of Global optimization*, volume 22, n°1-4, 285-317, 2002.