

N° 03/2016-D/MT

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

UNIVERSITÉ DES SCIENCES ET TECHNOLOGIE DE HOUARI
BOUMEDIENNE
FACULTÉ DE MATHÉMATIQUES



THÈSE

Présentée pour l'obtention du grade de DOCTEUR EN SCIENCES

EN RECHERCHE OPÉRATIONNELLE
OPTION : MATHÉMATIQUES DE GESTION

Par : MAHDI Sara

Sujet :

L'optimisation globale dans un ensemble
Pareto optimal discret

Soutenue publiquement le 03/04/2016 devant le jury composé de :

Mr M. ABBAS	Professeur	à l'U.S.T.H.B	Président
Mr D. CHAABANE	Professeur	à l'U.S.T.H.B	Directeur de thèse
Mr M. AIDÈNE	Professeur	à l'U.M.M.T.O	Examineur
Mr M. AÏDER	Professeur	à l'U.S.T.H.B	Examineur
Mme F. BELLAHCENE	Maitre de conférences/A	à l'U.M.M.T.O	Examinatrice
Mr S. BOUROUBI	Professeur	à l'U.S.T.H.B	Examineur
Mr M. MOULAI	Professeur	à l'U.S.T.H.B	Examineur
Mr R. E. STEUER	Professeur	à l'U. Georgia.USA	Examineur

Remerciements.

Mes premiers remerciements s'adressent à mon directeur de thèse, Professeur CHAABANE Djamel de m'avoir encadré et orienté tout au long des années de doctorat, qu'il trouve ici l'expression de ma profonde gratitude et reconnaissance.

Mes vifs remerciements vont également aux membres du jury pour avoir accepté de participer à ce jury et d'avoir accepté de juger ce travail.

J'exprime ainsi toute ma reconnaissance à monsieur ABBAS Moncef, Professeur à l'USTHB, qui m'a honoré d'accepter d'examiner mon travail et de présider le jury de soutenance.

Mes vifs remerciements vont également à messieurs AÏDER Meziane, BOUROUBI Sadek, MOULAÏ Mustapha, professeurs à l'USTHB, madame BELLAHCENE Fatima et monsieur AIDÈNE Mohamed, professeurs à l'UMMTO, qui m'ont fait l'honneur d'accepter de juger ce travail en tant qu'examineurs.

Que Monsieur STEUER Ralph E, Professeur à l'université de Georgia USA, trouve ici mes vifs remerciements et reconnaissances d'avoir accepté d'examiner ce travail et d'être dans le jury de cette thèse.

Je remercie très chaleureusement et sincèrement Daoud Malika qui m'a guidée et encouragée tout le temps. Pour son aide, ses précieux conseils et sa disponibilité qui m'ont permis d'avancer dans la réalisation de ce travail.

Je ne pourrais pas oublier mes amis qui m'ont aidé durant cette période. Je leur adresse toute ma reconnaissance pour leur soutien et leur sincère amitié.

Mes remerciements vont enfin à mes chers parents qui se sont toujours intéressés à mes travaux et qui m'ont toujours poussé à persévérer, je remercie aussi mes frères et mes sœurs pour leur soutien et leur encouragement.

Dédicaces.

*A mes très chers parents,
A mes frères et mes soeurs,
A toute ma famille,
A tous mes amis.*

Table des matières

liste des figures	iv
1 Introduction à l'optimisation multi-objectifs	5
1.1 Introduction	5
1.2 L'abécédaire de l'optimisation multi-objectifs	6
1.3 Dominance et efficacité	7
1.3.1 Détection géométrique des solutions Pareto optimales	13
1.3.2 Points particuliers	14
1.4 Illustration des définitions :	15
1.5 Caractérisation des solutions efficaces	16
1.6 Fonctions scalarisantes	17
1.6.1 Caractérisation à l'aide de poids	17
1.6.2 Caractérisation à l'aide de points cibles	18
1.7 Problématique	18
1.8 Classification des méthodes de résolution	19
1.9 Approches de résolution	20
1.9.1 Approches transformant le problème multi-objectifs en un problème mono-objectif "Les approches scalaires"	21
1.9.2 Approches Non Pareto	25
1.9.3 Approches Pareto :	27
1.10 Conclusion	28
2 L'optimisation multi-objectifs discrète	29
2.1 Introduction	29
2.2 La programmation linéaire mono-objectif	30
2.2.1 Complexité	31
2.2.2 Résultats fondamentaux de la programmation linéaire	31
2.2.3 Algorithme du simplexe (Dantzig, G.B 1947)[22]	32
2.2.4 Méthodes de Résolutions des Problèmes (<i>ILP</i>)	35
2.3 La programmation linéaire multi-objectifs en nombres entiers	38
2.3.1 Formulation mathématique d'un problème MOILP	38
2.3.2 Solutions supportées/non supportées	39

2.4	Résolution des problèmes (<i>MOILP</i>)	40
2.4.1	Méthode de Steuer et Choo [67]	40
2.4.2	La méthode de Klein & Hannan [47]	41
2.4.3	La méthode de Sylva & Crema [68]	42
2.4.4	Méthode de R. Gupta & R. Malhotra [39]	44
2.4.5	Méthode de M. Abbas & M. Moulaï [3]	48
2.4.6	Méthode de M. Abbas & D. Chaabane [1]	49
2.5	Conclusion	51
3	L'optimisation d'un critère linéaire sur un ensemble efficace	52
3.1	Introduction	52
3.2	Notations et résultats théoriques	54
3.3	Résolution du problème (P_E) dans le cas continu	56
3.3.1	Méthode de Yamamoto [78]	56
3.3.2	Méthode de Benson [7]	58
3.4	Résolution du problème (P_E) dans le cas discret	60
3.4.1	Résultats fondamentaux	61
3.4.2	Méthode de Jesus [46]	62
3.4.3	Méthode de Chaabane et al.[16]	65
3.5	Conclusion	69
4	La programmation linéaire fractionnaire	70
4.1	Introduction	70
4.2	Programmation linéaire fractionnaire mono-objectif	72
4.2.1	Formulation mathématique	72
4.2.2	Résultats et définitions	72
4.2.3	Résolution géométrique d'un problème linéaire frac- tionnaire	74
4.2.4	Résolution d'un programme linéaire fractionnaire	76
4.2.5	La résolution directe	76
4.2.6	La résolution par paramétrisation	80
4.2.7	La résolution d'un programme équivalent	81
4.3	Programmation linéaire fractionnaire multi-objectifs	82
4.3.1	Formulation du problème	82
4.3.2	Résolution du problème	83
4.4	Conclusion	83
5	L'optimisation d'un critère linéaire fractionnaire sur un en- semble efficace discret	85
5.1	Introduction	85
5.2	Résultats de base	87

5.2.1	Test d'efficacité	88
5.2.2	Arête incidente	88
5.3	Description de la méthode	88
5.4	Illustration numérique	91
5.5	Expérimentations et résultats	95
5.6	Conclusion	97
6	L'optimisation d'un critère linéaire fractionnaire sur l'ensemble efficace d'un problème (<i>MOILFP</i>).	98
6.1	Introduction	98
6.2	Résolution du problème (<i>FP_E</i>)	100
6.2.1	La méthode de O. Zerdani et M. Moulai [74]	100
6.2.2	Illustration numérique	103
6.3	Une nouvelle approche de résolution	107
6.3.1	Résultats de base	108
6.3.2	Description de la méthode	109
6.3.3	Illustration numérique	112
6.3.4	Expérimentations et résultats	118
7	Conclusion et perspectives	120

Table des figures

1.1	L'espace de décisions et l'espace des objectifs d'un problème bi-objectifs (deux fonctions objectifs).	7
1.2	l'optimalité locale au sens de Pareto	9
1.3	Structure de front de Pareto	10
1.4	Ensemble convexe (à gauche) et non convexe (à droite)	11
1.5	Espace des décisions	16
1.6	Espace des critères	16
1.7	l'approche d'agrégation	22
1.8	Interprétation graphique de l'approche ϵ – <i>contrainte</i>	23
1.9	Interprétation graphique de l'approche Goal programming. . . .	24
1.10	Principe de l'algorithme V.E.G.A.	25
2.1	L'arborescence de Branch and Bound.	37
2.2	Solutions supportées et non supportées	40
4.1	Le domaine réalisable	76
5.1	Le domaine d'admissibilité D	91
5.2	Le domaine réduit D_1	93
5.3	les solutions efficaces de (P)	95
6.1	Le domaine d'admissibilité D	112
6.2	Le domaine réduit D_1	113
6.3	Les solutions efficaces du problème (P)	114
6.4	Le domaine d'admissibilité D	114
6.5	Le domaine réduit D_1	115
6.6	Les solutions efficaces du problème (P)	116
6.7	Le domaine d'admissibilité D	116
6.8	Le domaine réduit D_1	117
6.9	L'ensemble des solutions efficaces	118

Introduction Générale

La majorité des problèmes d'optimisation issus du monde réel sont de nature multi-objectifs et exigent la considération simultanée de plusieurs critères généralement conflictuels. Chaque critère correspond à une fonction objectif à optimiser (maximiser ou minimiser); qui relève de la modélisation du problème traité, tels problèmes d'optimisation sont réputés pour être particulièrement difficiles à résoudre que leurs équivalents mono-objectif. La difficulté réside dans l'absence d'une relation d'ordre total entre les solutions admissibles. Une solution peut être meilleure qu'une autre sur certains objectifs et moins bonne sur les autres. Donc il n'existe généralement pas une solution qui satisfait simultanément l'ensemble des objectifs. C'est pourquoi le concept de solution optimale devient moins pertinent dans l'optimisation multi-objectifs. Dans ce cas, on essaye de déterminer un ensemble de solutions de compromis entre les différents objectifs à optimiser. L'étude de compromis a donné lieu à la définition des solutions optimales au sens de Pareto; une solution est dite *Pareto optimale* si l'amélioration à l'égard d'une fonction objectif entraîne une détérioration d'une autre fonction objectif, l'union de ces dernières solutions forme l'ensemble Pareto optimal; l'image de cet ensemble dans l'espace des objectifs (l'espace des critères) est appelé *front de Pareto*. L'optimisation multi-objectifs s'intéresse aux particularités liées à l'existence de ces solutions optimales multiples et aux méthodes de résolution dédiées à ce type de problèmes, qui sont souvent NP-difficiles.

Ainsi, la résolution d'un problème d'optimisation multi-objectifs consiste à trouver les solutions *Pareto optimales* ou encore les solutions *efficaces* qui répondent au mieux aux préférences du décideur, les difficultés liées à la résolution de ces problèmes résultent du fait que

- Il n'existe pas d'ordre total entre les solutions admissibles,
- La structure du front de Pareto est variable d'un problème à l'autre,
- Le nombre des solutions Pareto optimales croît en fonction de la taille du problème, en particulier selon le nombre d'objectifs.

La résolution des problèmes multi-objectifs relève de deux disciplines assez différentes (même si des efforts sont faits pour essayer de diminuer l'écart existant entre ces disciplines). En effet, résoudre un problème multi-objectifs

peut être réalisé en deux phases :

- a. La recherche des solutions de meilleur compromis ou un sous-ensemble de celles-ci. C'est la phase de l'optimisation multi-objectifs.
- b. Le choix de la solution à retenir. C'est la phase de la décision multi-objectifs, où le décideur doit extraire, parmi l'ensemble des solutions de compromis, celle (s) qu'il utilisera.

Il peut arriver dans la programmation multi-objectifs que l'ensemble des solutions efficaces, qui est une partie de l'ensemble des solutions admissibles, soit très vaste et parfois infini comme dans le cas continu. Dans une telle situation, il est souvent impossible d'énumérer toutes les solutions efficaces et la prise de décision devient une tâche très difficile, ce qui motive le retour vers le décideur dans le but d'exprimer ses préférences comme une fonction à optimiser dans cet ensemble. Ceci nous mène vers la recherche d'une solution efficace optimisant le critère des préférences du décideur sur l'ensemble des solutions efficaces.

Dans cette thèse, nous nous intéressons à l'optimisation d'un critère, appelé critère principal, sur l'ensemble des solutions efficaces d'un problème de programmation mathématique multi-objectifs à variables discrètes (entières), l'intérêt de telle classe de problèmes multi-objectifs résulte du fait que dans la majorité des applications concrètes de la programmation mathématique, la présence des variables discrètes est inévitable. Le problème en variables continues est difficile et il devient plus difficile en variables discrètes. Cependant, nous accentuons sur la qualité des solutions trouvées lors de développement de nos méthodes.

Le noyau de notre travail est principalement focalisé sur la programmation fractionnaire, l'intérêt porté à ce sujet tient à la variété de problèmes d'optimisation rencontrés dans plusieurs applications réelles (économie, ingénierie,...etc.) pour lesquels on considère l'optimisation d'un rapport de deux fonctions.

Dans ce contexte, le travail présenté s'articule autour de deux volets :

- Le premier volet consiste à optimiser un critère linéaire fractionnaire sur l'ensemble des solutions efficaces d'un problème de programmation linéaire multi-objectifs en nombres entiers (*MOILP* : Multiple Objective Integer Linear Programming). Ce travail a fait l'objet d'une publication dans la revue *International RAIRO Operations Research*, et de deux communications internationales :
 - 2nd International Symposium on Operational Research 2011, ISOR11.
 - International Conference on Applied Mathematical Optimization and Modelling 2014, (APMOD Conference 2014).

- Le deuxième volet, comme une généralisation du premier problème, nous abordons le problème d’optimisation d’un critère linéaire fractionnaire sur l’ensemble efficace d’un problème de programmation linéaire fractionnaire multi-objectifs en nombres entiers (*MOILFP* : Multiple Objective Integer Linear Fractional Programming problem). Ce travail a fait l’objet d’une communication dans la conférence internationale : 11th International Conferences on Multiple Objective Programming and Goal Programming (MOPGP 2015), Tlemcen, Algeria.

La thèse présentée est formée de six chapitres :

- Le premier chapitre a pour objectif de présenter le contexte de l’optimisation multi-objectifs qui sera le cadre de travail de cette thèse. Nous introduisons les notions fondamentales et les résultats théoriques liés à l’optimisation multi-objectifs tels que la dominance, la surface de compromis ainsi que les principales approches de résolution.
- Le deuxième chapitre est consacré à l’optimisation multi-objectifs linéaires en nombres entiers, nous exposons en premier lieu un aperçu sur les notions de base de la programmation linéaire en nombre entiers, nous présenterons par la suite les résultats principaux de la programmation linéaire multi-objectifs en nombres entiers ainsi que quelques méthodes existantes dans la littérature.
- Le troisième chapitre est consacré à l’optimisation d’un critère linéaire sur un ensemble efficace, nous étudions le problème dans le cas continu et le cas discret en exposant quelques méthodes de résolution pour les deux cas.
- Le quatrième chapitre s’attache à la programmation linéaire fractionnaire, qui représente le noyau de notre étude, nous présentons dans une première partie les résultats de base de la programmation linéaire fractionnaire mono-objectif ainsi que les principales approches de résolution. Ensuite, dans une deuxième partie, nous présentons la forme générale d’un problème de programmation linéaire fractionnaire multi-objectifs ainsi que quelques résultats liés à ce problème.
- Le cinquième chapitre expose notre première contribution qui consiste à l’élaboration d’une méthode exacte pour résoudre le problème de l’optimisation d’un critère linéaire fractionnaire sur l’ensemble efficace d’un problème (*MOILP*).
- Le dernier chapitre, est consacré à l’optimisation d’un critère linéaire fractionnaire sur l’ensemble efficace d’un problème de programmation linéaire fractionnaire multi-objectifs en nombres entiers (*MOILFP*), nous présentons en premier lieu la seule méthode existante dans la littérature, à notre connaissance, pour résoudre le problème. Ensuite, nous aborderons notre deuxième contribution en mettant au point une

nouvelle méthode exacte pour résoudre le problème dans l'espace des critères.

- Enfin, une conclusion résume les principaux résultats obtenus dans ce travail et quelques perspectives pour les futurs axes de recherche.

1

Introduction à l'optimisation multi-objectifs

Sommaire

1.1	Introduction	5
1.2	L'abécédaire de l'optimisation multi-objectifs . .	6
1.3	Dominance et efficacité	7
1.4	Illustration des définitions :	15
1.5	Caractérisation des solutions efficaces	16
1.6	Fonctions scalarisantes	17
1.7	Problématique	18
1.8	Classification des méthodes de résolution	19
1.9	Approches de résolution	20
1.10	Conclusion	28

1.1 Introduction

L'optimisation multi-objectifs est un domaine fondamental de l'aide à la décision multicritère, auquel de nombreux milieux scientifiques et industriels doivent faire face. Au cours des dernières années, un grand nombre de travaux, à la fois théoriques et appliqués, ont été publiés dans ce domaine.

La résolution d'un problème d'optimisation multi-objectifs consiste à trouver les solutions de meilleur compromis entre les critères à optimiser, connues par les solutions Pareto optimales, qui correspondent au mieux aux préférences du décideur. L'une des questions les plus difficiles est donc liée à l'identification de ces solutions ou d'une approximation de celles-ci pour des problèmes complexes.

Ce chapitre présente le contexte de l'optimisation multi-objectifs qui sera le cadre de travail de cette thèse. Nous introduisons les principales définitions telles que la dominance, la surface de compromis ainsi que les principales approches de résolution.

1.2 L'abécédaire de l'optimisation multi-objectifs

L'optimisation multi-objectifs ou multicritère consiste à optimiser (maximiser ou minimiser) un vecteur dont chacune de ses composantes est une fonction de \mathbb{R}^n appelée un objectif ou un critère, un problème d'optimisation multi-objectifs (*MOP*) ("Multiple Objective Optimization Problem") peut être défini comme suit :

$$(MOP) \begin{cases} \text{"Opt"} & F(x) = (f_1(x), f_2(x), \dots, f_p(x)) \\ \text{s.c.} & x \in S = \{x \in \mathbb{R}^n | g_i(x) \leq 0, i = 1, 2, \dots, m\} \end{cases} \quad (1.1)$$

où p est le nombre d'objectifs ($p \geq 2$), S est l'ensemble des solutions admissibles dans **l'espace de décisions** \mathbb{R}^n et $x = (x_1, x_2, \dots, x_n) \in S$ est un vecteur représentant les variables de décisions. $F(x) = (f_1(x), f_2(x), \dots, f_p(x))$ est le vecteur des objectifs à optimiser.

Quand les objectifs f_p et les fonctions g_j sont linéaires, on obtient un problème de programmation linéaire multi-objectifs (**MOLP** : Multiple Objective Linear Programming) défini par

$$(MOLP) \begin{cases} \text{"Opt"} & z_k = c^k x; \quad k = 1, \dots, p \\ \text{s.c.} & x \in S \end{cases} \quad (1.2)$$

où $c^k \in \mathbb{R}^{1 \times n}$ pour $k = 1, \dots, p$ et l'ensemble S est déterminé par des contraintes linéaires $S = \{x \in \mathbb{R}^n | Ax \leq b, x \geq 0\}$; $A \in \mathbb{R}^{(m \times n)}$; $x \in \mathbb{R}^{n \times 1}$; $b \in \mathbb{R}^{m \times 1}$; $m, n \in \mathbb{N}^*$.

En ajoutant la contrainte d'intégrité des variables de décisions, on obtient un problème de programmation linéaire multi-objectifs en nombre entiers (**MOILP** : Multiple Objective Integer Linear Programming) défini par :

$$(MOILP) \begin{cases} \text{"Opt"} & z_k = c^k x; \quad k = 1, \dots, p \\ \text{s.c.} & x \in D \end{cases} \quad (1.3)$$

où $D = S \cap \mathbb{Z}^n$, \mathbb{Z} étant l'ensemble des nombres entiers relatifs.

À chaque solution $x \in S$ est associée une image $z \in Z$ dite vecteur objectif définie par l'application suivante :

$$\begin{aligned} F : S &\longrightarrow Z \\ x &\longmapsto z = F(x) = (z_1, z_2, \dots, z_p). \end{aligned}$$

où $Z = F(S) = \{z \in \mathbb{R}^p \mid z = F(x), x \in S\}$ représente l'ensemble des points réalisables dans l'**espace des objectifs** \mathbb{R}^p (ou également l'espace des critères).

Autrement dit, Z est la projection de S sur l'espace des objectifs (voir la Figure 1.1).

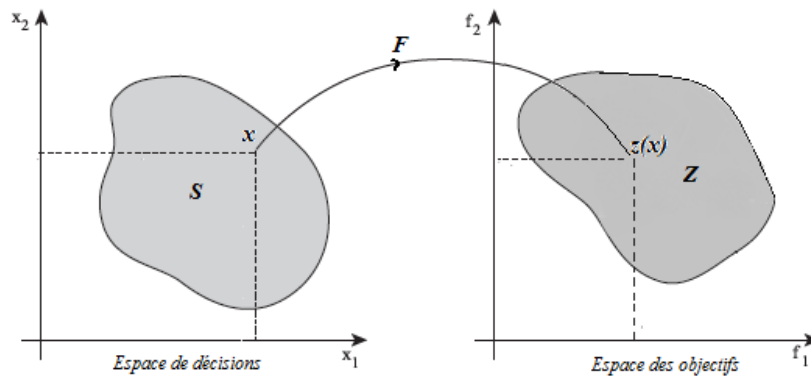


FIGURE 1.1 – L'espace de décisions et l'espace des objectifs d'un problème bi-objectifs (deux fonctions objectifs).

1.3 Dominance et efficacité

Dans l'optimisation multi-objectifs, le décideur évalue généralement une solution par rapport à chaque critère, et se positionne donc naturellement dans l'espace des objectifs. Néanmoins, contrairement au problème mono-objectif où il existe un ordre total parmi les solutions réalisables, dans le contexte multi-objectifs, en raison de la nature conflictuelle des objectifs, il n'y a généralement pas une solution réalisable qui optimise tous les objectifs à la fois. Ainsi, une relation d'ordre partiel est généralement définie, la plus connue et utilisée est la relation de dominance au sens de Pareto, définie par un équilibre tel qu'on ne peut pas améliorer un objectif sans détériorer au moins un des autres objectifs (Vilfredo Pareto).

Sans perte de généralité, nous supposons par la suite que tous les objectifs sont à maximiser et nous définissons le problème (*MOLP*) comme suit

$$(MOLP) \begin{cases} \text{“max” } z_k = c^k x; & k = 1, \dots, p \\ \text{s.c. } & x \in S. \end{cases} \quad (1.4)$$

Définition 1.1 (relation de dominance)

Soit “>” la relation d’ordre partiel définie sur \mathbb{R}^p par :

$\forall z, z' \in \mathbb{R}^p$, $z > z'$ si et seulement si :

- z est au moins aussi bon que z' sur tous les objectifs, i.e :
 $\forall i \in \{1, \dots, p\}, z_i \geq z'_i$;
- et z est strictement meilleur que z' sur au moins un objectif, i.e :
 $\exists i \in \{1, \dots, p\}, z_i > z'_i$;

Propriété de la relation de dominance

La relation binaire “>” :

- n’est pas réflexive; une solution ne se domine pas elle même;
- n’est pas symétrique; on n’a jamais $z > z'$ et $z' > z$;
- est transitive; car $z > z'$ et $z' > z''$ implique $z > z''$.

Définition 1.2

- $\forall z^1, z^2 \in Z$, z^1 domine z^2 si et seulement si $z^1 > z^2$.
- Un vecteur objectif $z \in Z$ est dit *non dominé*, si est seulement si, il n’existe pas $z' \in Z$ tel que $z' > z$.
- $\tilde{x} \in S$ est une solution efficace ou optimale au sens de Pareto, si et seulement si, il n’existe pas de $x \in S$ tel que $z_i(x) \geq z_i(\tilde{x}), \forall i = 1, \dots, p$ avec au moins une inégalité stricte (c-à-d. $\nexists x \in S : z(x) > z(\tilde{x})$). Le vecteur objectif correspondant $z(x)$ est dit solution non dominée.

Définition 1.3

- Soient $z, z' \in Z$, z domine fortement z' ssi : $z_i > z'_i, \forall i \in \{1, \dots, p\}$, cette relation sera notée $z \gg z'$.
- Un vecteur objectif $z \in Z$ est dit *faiblement non dominé* s’il n’existe pas de $z' \in Z$ tel que $z' \gg z$.
- Une solution $\tilde{x} \in S$ est dite *faiblement efficace* s’il n’existe pas de $x \in S$ tel que $z(x) \gg z(\tilde{x})$.
- Une solution $\tilde{x} \in S$ est dite fortement efficace si et seulement s’il n’existe pas une autre solution $x \in S$ tel que $x \neq \tilde{x}$ et $z_i(x) \geq z_i(\tilde{x}), \forall i = 1, 2, \dots, p$. Le vecteur objectif $z(\tilde{x})$ est dit fortement non dominé.

Définition 1.4 (ϵ -Dominance)

Un vecteur objectif $z \in Z$ ϵ -domine un vecteur objectif $z' \in Z$, avec $\epsilon > 1$, si et seulement si $z_i \geq \epsilon \cdot z'_i, \forall i \in \{1, \dots, p\}$, cette relation sera notée $z \geq_\epsilon z'$.

On définit l'optimalité locale et l'optimalité globale au sens de Pareto comme suit :

Définition 1.5 Optimalité locale au sens de Pareto

Une solution $x \in S$ est *optimale localement au sens de Pareto* s'il existe un réel $\delta > 0$ tel qu'il n'y ait pas une solution $x' \in S$ qui domine x avec $x' \in S \cap B(x, \delta)$ où $B(x, \delta)$ représente une boule de centre x et de rayon δ .

D'une manière équivalente, une solution x est optimale localement au sens de Pareto si elle est optimale au sens de Pareto sur une restriction de l'ensemble S .

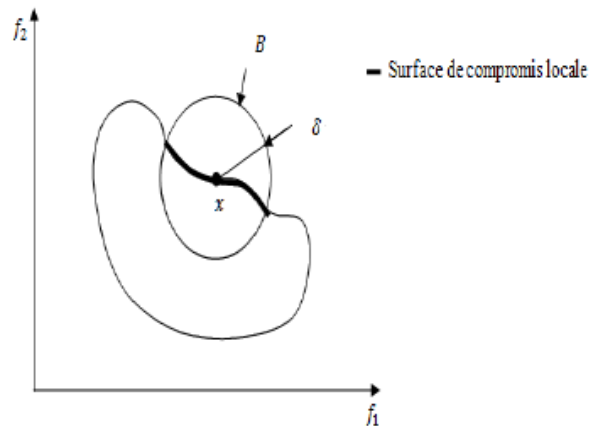


FIGURE 1.2 – l'optimalité locale au sens de Pareto

Définition 1.6 Optimalité globale au sens de Pareto

Une solution $x \in S$ est *optimale globalement au sens de Pareto* (ou optimale au sens de Pareto) s'il n'existe pas de $x' \in S$ telle que $Z(x')$ domine $Z(x)$.

L'ensemble des vecteurs non-dominé est appelé *front de Pareto* ou la surface de compromis et l'ensemble des solutions de S dont l'ensemble image représente le front de Pareto est appelé *l'ensemble Pareto optimal* ou *l'ensemble efficace*.

Nous pouvons définir le front de Pareto de la manière suivante :

Définition 1.7 Soit Z l'image dans l'espace des objectifs de l'ensemble réalisable S . Le front de Pareto E de Z est défini comme suit :

$$E = \{z \in Z \mid \nexists y \in Z, y > z\}.$$

Les figures ci-dessous présentent pour un problème à deux objectifs les quatre fronts de Pareto en fonction du désir de l'utilisateur de minimiser ou maximiser les objectifs.

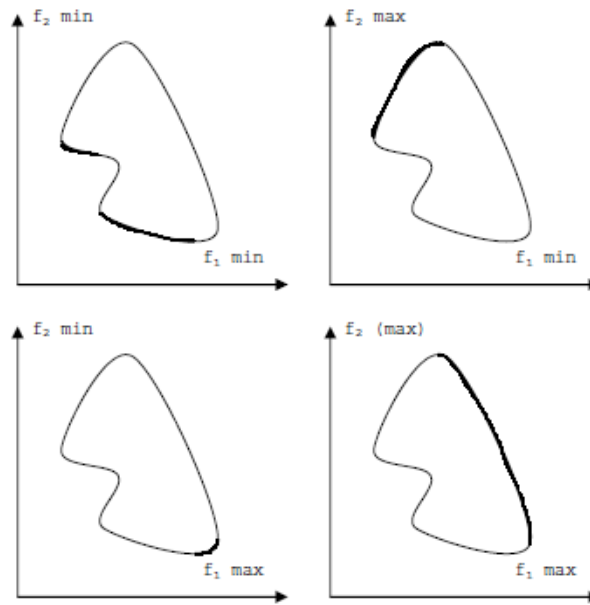


FIGURE 1.3 – Structure de front de Pareto

Définition 1.8 *Ensemble Convexe*

Un ensemble S est dit convexe lorsque pour tout x et y de S , le segment $[x, y]$ est inclus dans S , c-à-d :

$$\forall x, y \in S; \forall \lambda \in [0, 1] : \lambda x + (1 - \lambda)y \in S.$$

- Soit une famille $\{S_i\}_{i=1\dots k}$ d'ensembles convexes et $S = \bigcap_{i=1}^k S_i$. Alors S est convexe.

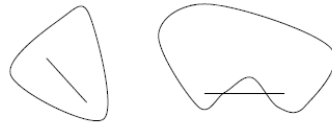


FIGURE 1.4 – Ensemble convexe (à gauche) et non convexe (à droite)

- Soit Λ l'ensemble de tous les vecteurs $\lambda = (\lambda_i)_{i=1,\dots,p}$ définis par :

$$\Lambda = \left\{ \lambda \in \mathbb{R}^p \mid \sum_{i=1}^p \lambda_i = 1, \lambda_i \geq 0, i = 1, \dots, p \right\}.$$

Pour $\lambda \in \Lambda$, on définit le problème (P_λ) par

$$(P_\lambda) \begin{cases} \text{“max”} & \sum_{i=1}^p \lambda_i c^i(x) \\ \text{s.c.} & x \in S. \end{cases} \quad (1.5)$$

Théorème 1.1 (Geoffrion, 1968)[34]

Soit $\hat{x} \in S$, \hat{x} est une solution efficace si et seulement si \hat{x} est une solution optimale du problème paramétrique (P_λ) pour un certain vecteur $\lambda \geq 0$.

Théorème 1.2 (Iserman, 1974) [45]

- a. $\hat{x} \in S$ une solution optimale du problème paramétrique (P_λ)
 - Si $\lambda \geq 0$ alors \hat{x} est faiblement efficace.
 - Si $\lambda > 0$ alors \hat{x} est efficace.
 - Si $\lambda \geq 0$ et \hat{x} est l'unique solution optimale de (P_λ) alors \hat{x} est fortement efficace.
- b. En outre, si S est convexe alors :
 - Si \hat{x} est efficace alors il existe $\lambda > 0$ tel que \hat{x} est une solution optimale de (P_λ) .
 - Si \hat{x} est faiblement efficace alors il existe $\lambda \geq 0$ tel que \hat{x} est une solution optimale de (P_λ) .

Théorème 1.3 [45] Une solution $\hat{x} \in S$ est efficace si et seulement s'il existe $\lambda \in \mathbb{R}^p, \lambda > 0$ tel que

$$\lambda z(\hat{x}) \geq \lambda z(x) \text{ pour tout } x \in S$$

Définition 1.9 *Point extrême*

On dit que x est un point extrême (ou sommet) d'un convexe non vide S si

$$x = \lambda y + (1 - \lambda)z, \text{ pour } y, z \in S \text{ et } 0 < \lambda < 1 \text{ alors } x = y = z.$$

En d'autres termes, x n'est à l'intérieur d'aucun segment de S .

Définition 1.10 *Combinaison linéaire convexe*

Un vecteur y est une combinaison linéaire convexe des points $\{x_1, \dots, x_p\}$ s'il existe des coefficients réels $\lambda_i, i = 1 \dots p$, tels que :

$$y = \sum_{i=1}^p \lambda_i x_i, \text{ avec } \lambda_i \geq 0, \forall i = 1 \dots p, \text{ et } \sum_{i=1}^p \lambda_i = 1.$$

Définition 1.11 *Envelope convexe*

L'enveloppe convexe d'un ensemble $S \subset \mathbb{R}^n$ est l'ensemble des points de \mathbb{R}^n qui s'écrivent comme combinaisons convexes des points de S . On la note par :

$$\text{conv}(S) = \left\{ x \in \mathbb{R}^n \mid x = \sum_{i=1}^p \lambda_i x_i, x_i \in S, \lambda_i \geq 0, i = 1 \dots p, \text{ et } \sum_{i=1}^p \lambda_i = 1 \right\}.$$

Autrement dit, c'est le plus petit ensemble convexe contenant S .

- S est convexe si et seulement si $S = \text{conv}(S)$.

Définition 1.12

- L'ensemble $\{x \in \mathbb{R}^n \mid a^t x = b\}$ représente un hyperplan de \mathbb{R}^n .
- L'ensemble $\{x \in \mathbb{R}^n \mid a^t x \leq b\}$ représente un demi-espace fermé de \mathbb{R}^n dont l'hyperplan correspondant constitue *la frontière*.

Définition 1.13

- Un polyèdre P est l'intersection d'un nombre fini de demi-espaces fermés et/ou d'hyperplans. Un polyèdre est un ensemble convexe fermé.
- Un polyèdre P est dit borné, s'il existe une valeur β finie et positive telle que

$$|x_j| \leq \beta, \quad \forall j = 1 \dots n, \quad \forall x \in P.$$

- Un polyèdre borné et non vide est dit polytope.

1.3.1 Détection géométrique des solutions Pareto optimales

Pour tester l'efficacité d'un point $x^* \in D$, Steuer [66] a introduit le concept d'ensemble dominant qui est principalement basée sur la notion du cône.

Définition 1.14 (Cône) Soit le vecteur $V \subset \mathbb{R}^n$, $V \neq \emptyset$, V est un cône si et seulement si $\alpha v \in V$ pour tout scalaire $\alpha \geq 0$ et tout $v \in V$.

- Le vecteur d'origine de \mathbb{R}^n ($0_{\mathbb{R}^n}$) est contenu dans chaque cône.
- Un cône convexe est un cône qui est également un ensemble convexe.

Définition 1.15 (cône polaire) Soit $V \subset \mathbb{R}^n$ un cône. Le cône polaire non négatif de V (noté V^\geq) est le cône convexe $V^\geq = \{y \in \mathbb{R}^n \mid y^t v \geq 0, \forall v \in V\}$.

Ce qui signifie que tous les vecteurs de V^\geq font un angle inférieur ou égal à 90° avec chaque vecteur de V .

Définition 1.16 (Vecteurs générateurs d'un cône) Soit un ensemble de vecteurs $\{v^1, v^2, \dots, v^p\}$ de \mathbb{R}^n et l'ensemble V tel que :

$$V = \{v \in \mathbb{R}^n \mid v = \sum_{i=1}^p \alpha_i v^i, \alpha_i \geq 0\},$$

V est l'ensemble de toutes les combinaisons linéaires à coefficients non négatifs des v^i , $i = 1, \dots, p$ et est le cône convexe engendré par l'ensemble $\{v^1, v^2, \dots, v^p\}$. Les vecteurs v^i , $i = 1, p$ sont appelés les générateurs de V .

- La dimension d'un cône $V \subset \mathbb{R}^n$ est donnée par le nombre de vecteurs linéairement indépendants dans V .

Définition 1.17 (cône semi-polaire positif) Soit $V \subset \mathbb{R}^n$ un cône convexe généré par $\{v^1, v^2, \dots, v^p\}$. Alors, le cône semi-polaire positif (noté $V^>$) est le cône convexe

$$V^> = \{y \in \mathbb{R}^n \mid y^t v^i \geq 0, \text{ pour tout } i \text{ et } y^t v^i > 0 \text{ pour au moins un } i\} \cup \{0_{\mathbb{R}^n}\}.$$

Définition 1.18 (Ensemble dominant)

Soit $x^* \in D$ et $C^>$ le cône semi-polaire positif généré par les gradients des p fonctions objectifs i.e.

$$C^> = \{y \in \mathbb{R}^n \mid c^i y \geq 0, \text{ pour tout } i \text{ et } c^i y > 0 \text{ pour au moins un } i\} \cup \{0_{\mathbb{R}^n}\}$$

On définit l'ensemble dominant de x^* noté ED_{x^*} par :

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

C'est-à-dire

$$ED_{x^*} = \{x \in \mathbb{R}^n \mid x = x^* + y, y \in C^>\}.$$

L'ensemble dominant ED_{x^*} contient tous les points dont les vecteurs critères dominant le vecteur critère de x^* . Notons que la somme des ensembles x^* et $C^>$ effectue une translation du cône semi-polaire positif de l'origine vers le point en question. Le théorème suivant montre l'importance de cet ensemble dans la détection des solutions efficaces :

Théorème 1.4 [66] *Soit ED_{x^*} l'ensemble dominant en $x^* \in D$. Alors x^* est efficace si et seulement si : $ED_{x^*} \cap D = \{x^*\}$.*

Preuve

\Rightarrow / Supposons que $ED_{x^*} \cap D \neq \{x^*\}$. Alors, il existe $\bar{x} \in ED_{x^*} \cap D$, et $\bar{x} \neq x^*$. Puisque $\bar{x} \in ED_{x^*}$, alors $\bar{x} = x^* + y$ où $y \in C^>$. Comme $Cy \geq 0$, $Cy \neq 0$ alors $C\bar{x} \geq Cx^*$, $C\bar{x} \neq Cx^*$. Ceci contredit le fait que x^* est efficace. Alors si x^* est efficace, $ED_{x^*} \cap D = \{x^*\}$.

\Leftarrow / Supposons que $ED_{x^*} \cap D = \{x^*\}$ et x^* n'est pas efficace, ceci implique qu'il existe \bar{x} tel que le vecteur objectif de \bar{x} domine le vecteur objectif de x^* avec $\bar{x} \notin D$, donc le vecteur objectif de x^* est non dominé dans D , et par conséquent, x^* est efficace. □

1.3.2 Points particuliers

En vue d'avoir certains points de références permettant de discuter l'intérêt des solutions trouvées, des points particuliers ont été définis dans l'espace des objectifs, ces points peuvent représenter des solutions réalisables ou non. Nous présentons ici les points les plus utilisés, le point idéal, le point anti-idéal, la matrice des gains et le point nadir.

Définition 1.19 (Le point idéal)

Le point idéal $z^* = (z_1^*, z_2^*, \dots, z_p^*)$ est le vecteur qui maximise chaque fonction objectif individuellement, i.e :

$$z_i^* = \max_{x \in S} z_i(x) \text{ pour } i \in \{1, \dots, p\}.$$

Le point idéal est généralement une solution utopique, dans le sens où il n'appartient pas à l'espace des objectifs réalisable, mais il est souvent employé dans les méthodes d'optimisation comme un point de référence exprimant le but que le décideur veut atteindre pour chaque objectif, par exemple dans la normalisation des objectifs.

Définition 1.20 (Point anti-idéal) Le point anti-idéal \underline{z} est le point de \mathbb{R}^p de coordonnées

$$\underline{z}_i = \min_{x \in S} z_i(x) \text{ pour } i \in \{1, \dots, p\}.$$

Définition 1.21 (Matrice des gains) Soit \hat{x}^j une solution optimale du critère z_j . La matrice $(p \times p)$ formée des éléments de $z_{kj} = z_k(\hat{x}^j)$ est dite matrice des gains (“payoff matrix”).

$$\begin{pmatrix} z_1^* & \cdots & z_{1j} & \cdots & z_{1p} \\ \cdot & & \cdot & & \cdot \\ \cdot & & \cdot & & \cdot \\ \cdot & & \cdot & & \cdot \\ z_{k1} & \cdots & z_{kj} & \cdots & z_{kp} \\ \cdot & & \cdot & & \cdot \\ \cdot & & \cdot & & \cdot \\ \cdot & & \cdot & & \cdot \\ z_{p1} & \cdots & z_{pj} & \cdots & z_p^* \end{pmatrix} .$$

Les coordonnées du point idéal apparaissent sur la diagonale de cette matrice. Lorsqu’un critère j possède plusieurs solutions optimales, la colonne j de la matrice des gains dépendra de la solution \hat{x}^j choisie (la matrice des gains est univoquement déterminée si, pour tout critère j , la solution \hat{x}^j est unique).

Définition 1.22 (Point nadir) Le point η de \mathbb{R}^p de coordonnées

$$\eta_k = \min_{j=1, \dots, p} (z_{kj}) \quad k = 1, \dots, p;$$

où z_{kj} est un élément de la matrice des gains, est dit point nadir. η_k est la plus petite valeur sur la ligne k de la matrice des gains.

Le point nadir peut être défini par le minimum de chaque fonction objectif dans le front de Pareto E et non pas dans tout l’espace réalisable i.e :

$$\eta_i = \min_{x \in E} z_i(x) \quad \text{pour } i \in \{1, \dots, p\}.$$

Ce vecteur sert à restreindre l’espace de recherche ; il est utilisé dans certaines méthodes d’optimisation interactives.

1.4 Illustration des définitions :

Considérons le problème bi-objectifs suivant :

$$(P) \begin{cases} \max & z_1(x) = x_1 \\ \max & z_2(x) = -x_1 + x_2 \\ \text{s.c.} & x_1 + 2x_2 \leq 10 \\ & x_1 + x_2 \leq 6 \quad x_1, x_2 \in \mathbb{Z}_+ \\ & x_1 \leq 4 \end{cases}$$

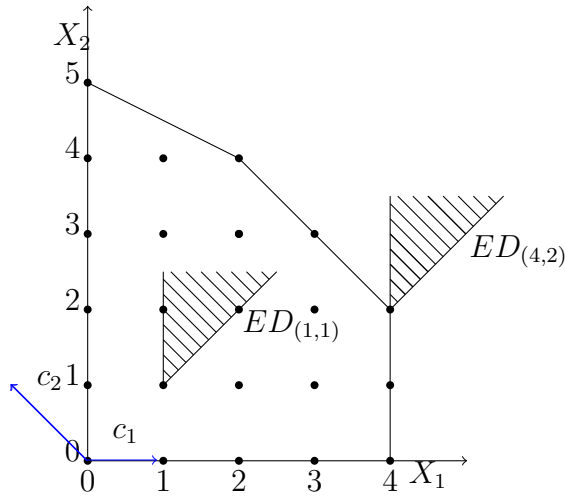


FIGURE 1.5 – Espace des décisions

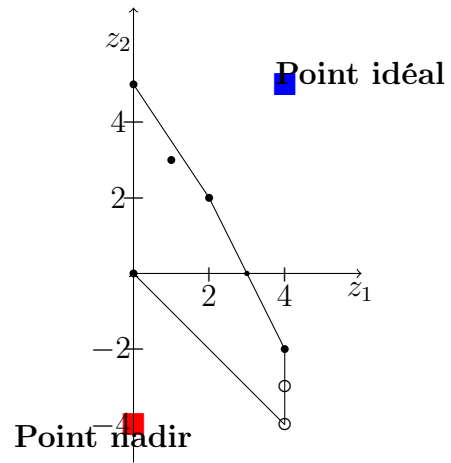


FIGURE 1.6 – Espace des critères

- Les solutions efficaces du problème (P) sont $\{(4, 2), (3, 3), (2, 4), (1, 4), (0, 5)\}$.
- Le point $x^* = (4, 2)$ est efficace car $ED_{x^*} \cap D = x^*$, tandis que le point $x' = (1, 1)$ n'est pas efficace car $ED_{x'} \cap D \neq x'$.
- Deux vecteurs objectif faiblement non dominés sont détectées, elles sont les points $(4, -4)$ et $(4, -3)$.
- Le point idéal est le point : $(4, 5)$.
- La matrice des gains ici est unique : $\begin{pmatrix} 4 & 0 \\ -4 & 5 \end{pmatrix}$
- Le point nadir est le point de coordonnées : $\eta = (0, -4)$

1.5 Caractérisation des solutions efficaces

Il peut arriver dans la programmation mathématique multi-objectifs que l'ensemble des solutions efficaces, qui est une partie de l'ensemble des solutions admissibles, soit très vaste et parfois infini comme dans le cas continu. Dans une telle situation, il est souvent impossible d'énumérer toutes les solutions efficaces et même lorsque c'est possible, il est nécessaire d'aider le décideur à faire son choix parmi les solutions efficaces. La sélection d'une solution efficace spécifique comme candidat de meilleur compromis nécessite donc une certaine connaissance de la structure de préférence du décideur. Cette information est obtenue directement ou indirectement et peut parfois se traduire en terme de *paramètres de préférences*, citons brièvement les paramètres qui sont fréquemment utilisés.

★ *Le vecteur de poids* : $(\lambda_1, \dots, \lambda_p)$ ou λ_k reflète l'importance relative de chaque critère k , $k = (1, \dots, p)$.

Notons que ces paramètres peuvent également être des points cibles tels que :

★ *Le point de référence* qui est défini par des niveaux d'aspiration (valeurs souhaitables) sur chaque critère.

★ *Le point de réservation* qui est défini par des niveaux de réservation (valeurs non souhaitables) sur chaque critère.

1.6 Fonctions scalarisantes

Étant donné un ensemble de paramètres de préférence $\Lambda = \{(\lambda_1, \dots, \lambda_p) \in \mathbb{R}^p\}$, il est possible de définir une fonction croissante dite *fonction scalarisante*, qui agrège les valeurs des critères pour chaque solution :

$$S(Z, \lambda) : \mathbb{R}^p \times \Lambda \rightarrow \mathbb{R},$$

où $\lambda \in \Lambda \subset \mathbb{R}^p$ est le vecteur de paramètres choisi. Nous citons dans la suite les fonctions d'agrégation les plus employées.

1.6.1 Caractérisation à l'aide de poids

- La somme pondérée : est largement utilisée dans l'optimisation linéaire multi-objectifs où le problème (*MOLP*) peut se ramener à un problème de programmation paramétrique

$$S_1(z, \lambda) = \sum_{k=1}^p \lambda_k z_k, \text{ avec } \sum_{k=1}^p \lambda_k = 1, \lambda_k > 0, \forall k \in \{1, \dots, p\},$$

$$S_2(z, \lambda) = \sum_{k=1}^p \lambda_k |z_k - \bar{z}_k|,$$

où \bar{z}_k est la $k^{\text{ème}}$ composante du point idéal.

$S_2(z, \lambda)$ mesure la déviation qui sépare l'évaluation des propositions, qui sont généralement des points efficaces ou faiblement efficaces, du point d'aspiration. Cette déviation peut être mesurée par d'autres normes parmi lesquelles citons :

- Norme L_q pondérée :

$$S_3(z, \lambda) = \left[\sum_{k=1}^p \lambda_k |z_k - \bar{z}_k|^q \right]^{\frac{1}{q}}, q \in \mathbb{Z}_+^*.$$

- Norme L_∞ de Tchebycheff pondérée :

$$S_4(z, \lambda) = \max_{1 \leq k \leq p} \{\lambda_k |z_k - \bar{z}_k|\}.$$

- Norme composée (Tchebycheff pondérée augmentée) :

$$S_5(z, \lambda) = \max_{1 \leq k \leq p} \{ \lambda_k |z_k - \bar{z}_k| + \rho \sum_{k=1}^p \lambda_k |z_k - \bar{z}_k| \}; \rho > 0.$$

1.6.2 Caractérisation à l'aide de points cibles

- Niveaux d'aspiration : qui peut être obtenue en minimisant la fonction

$$S_6(z, \lambda) = \left[\sum_{k=1}^p \lambda_k |z_k - \hat{z}_k|^q \right]^{\frac{1}{q}}, q \in \mathbb{Z}_+^*,$$

où $(\hat{z}_1, \dots, \hat{z}_p) \in Z$ est un point cible dont on souhaite s'approcher autant que possible.

- Niveaux de réservation : qui peut être obtenue en maximisant la fonction

$$S_7(z, \lambda) = \left[\sum_{k=1}^p \lambda_k |z_k - z'_k|^q \right]^{\frac{1}{q}},$$

et des contraintes sur les critères $z_k \geq z'_k$, où z'_k représente une valeur dont on souhaite s'écarter le plus possible.

1.7 Problématique

La difficulté principale d'un problème multi-objectifs est qu'il n'existe pas de définition de la solution optimale, le décideur peut simplement exprimer le fait qu'une solution est préférable à une autre mais il n'existe pas une solution meilleure que toutes les autres.

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

Pour répondre à ce problème, deux approches sont adoptées, la première est de ramener un problème multi-objectifs à un problème mono-objectif au risque d'enlever toute signification au problème. La deuxième approche est de tenter d'apporter des réponses au problème en prenant en compte l'ensemble des critères. Dans ces deux approches, soit le décideur intervient dès le début de la définition du problème, en exprimant ses préférences, afin de transformer le problème multi-objectifs en un problème mono-objectif, soit

il effectue son choix dans l'ensemble des solutions proposées par le solveur multi-objectifs.

Le principal objectif d'un solveur multi-objectifs est donc de rendre les décisions plus faciles et moins subjectives en proposant un sous-ensemble représentatif de l'espace des objectifs.

1.8 Classification des méthodes de résolution

Dans les différentes publications, nous rencontrons deux classifications différentes des méthodes de résolution de problèmes multi-objectifs. Le premier classement adopte un point de vue décideur, les méthodes sont classées en fonction de l'usage que l'on désire en faire. Le deuxième classement est plus théorique, plus conceptuel, les méthodes sont triées en fonction de leur façon de traiter les fonctions objectifs.

- a. Dans le premier classement, on distingue trois schémas possibles selon l'intervention du décideur. Soit le décideur intervient dès le début de la définition du problème, en exprimant ses préférences, afin de transformer un problème multi-objectifs en un problème mono-objectif. Soit le décideur effectue son choix dans l'ensemble des solutions proposées par le solveur multi-objectifs :
 - **Les méthodes a priori :(décideur \rightarrow recherche)**
Elles consistent à combiner les différentes fonctions objectifs en une fonction d'utilité suivant les préférences du décideur. Dans ce cas, le décideur est supposé connaître a priori le poids de chaque objectif afin de les mélanger dans une fonction unique. Cela revient à résoudre un problème mono-objectif. Cependant dans la plupart des cas, le décideur ne peut pas exprimer clairement sa fonction d'utilité, soit par manque d'expérience ou d'informations, soit parce que les différents objectifs sont généralement non comparables.
 - **Les méthodes a posteriori :(recherche \rightarrow décideur)**
Dans ces méthodes, le décideur prend sa décision d'après un ensemble de solutions fourni par la méthode d'optimisation. Dans ce cas, la qualité de la décision 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, ce qui peut être difficile et requérir un temps de calcul important.
 - **Les méthodes interactives :(recherche \leftrightarrow décideur)**
Dans ce cas, les processus de décision et d'optimisation sont alternés. Le décideur intervient de manière à modifier certaines variables ou contraintes afin de diriger le processus d'optimisation. Le décideur

modifie ainsi interactivement le compromis entre ses préférences et les résultats obtenus. Cette approche permet donc de bien prendre en compte les préférences du décideur, mais nécessite sa présence tout au long du processus de recherche.

- b. La deuxième classification adopte un point de vue plus théorique articulé autour des notions d'agrégation et de Pareto optimalité, les approches de cette classification peuvent être divisées en trois catégories :
- **Approches transformant le problème multi-objectifs en un ou plusieurs problème(s) mono-objectif(s)** : ces approches transforment le problème initial afin de se ramener à la résolution de un ou plusieurs problèmes mono-objectif. En général, ces méthodes nécessitent une bonne connaissance du problème et ne fournissent qu'une seule solution.
 - **Approches Pareto** : ces approches utilisent directement la notion de dominance au sens de Pareto dans la sélection des solutions générées. Cette idée a été initialement introduite par Goldberg [35] pour résoudre les problèmes proposés par Schaffer (Schaffer, 1985).
 - **Approches Non Pareto** : ces approches ne transforment pas le problème d'origine. Elles effectuent leur recherche en traitant indépendamment chacun des objectifs, l'exemple le plus classique est l'algorithme VEGA (Vector Evaluated Genetic Algorithm). Ces méthodes ont souvent du mal à trouver les solutions de compromis puisqu'elles se focalisent sur les portions extrêmes du front de Pareto. Nous pouvons classer dans cette catégorie les méthodes lexicographiques qui donnent un ordre de priorité sur les objectifs à traiter.

1.9 Approches de résolution

Un grand nombre d'approches existent pour résoudre les problèmes multi-objectifs. Certaines utilisent des connaissances du problème pour fixer des préférences sur les critères pour contourner l'aspect multi-objectifs du problème. D'autres mettent tous les critères au même niveau d'importance, mais là aussi il existe plusieurs façons de réaliser une telle opération. Plusieurs ouvrages ou articles de synthèse ont été rédigés, des états de l'art plus complets peuvent être consultés notamment dans [Ulungu and Teghem, 1994a; Miettinen, 1999; Ehrgott, 2000; Ehrgott and Gandibleux, 2000; Deb, 2001; Collette et Siarry, 2002]. Nous décrirons dans cette section les principales approches de résolution en commentant leurs avantages et leurs inconvénients.

1.9.1 Approches transformant le problème multi-objectifs en un problème mono-objectif "Les approches scalaires"

A l'origine, les problèmes multi-objectifs étaient transformés en problèmes mono-objectifs, plusieurs approches ont été mises au point pour transformer les problèmes multi-objectifs en problèmes mono-objectifs en optimisant une fonction d'utilité. Parmi ces méthodes, nous citons la méthode d'agrégation, la méthode ϵ -contrainte, et la méthode de programmation par but.

a. La méthode d'agrégation

C'est l'une des premières méthodes utilisée pour la génération des solutions Pareto optimales. Elle consiste à transformer le problème multi-objectifs en un problème mono-objectif en définissant une fonction objectif unique $F(x)$ comme étant la somme pondérée de toutes les fonctions objectifs, en affectant à chacun d'eux un coefficient de poids qui représente l'importance relative que le décideur attribue à l'objectif :

$$F(x) = \sum_{i=1}^p \lambda_i f_i(x),$$

où $\lambda_i \in [0, 1]$, $\sum_{i=1}^p \lambda_i = 1$.

La figure 1.7 illustre le fonctionnement de la méthode. Fixer un vecteur poids revient à trouver un hyperplan dans l'espace des objectifs (une droite pour un problème bi-objectifs). La solution Pareto optimale est le point où l'hyperplan possède une tangente commune avec l'espace réalisable (le point x dans la figure 1.7(a)). Donc, pour une agrégation donnée, il n'existe généralement qu'une seule solution Pareto optimale. En faisant varier le vecteur λ , il est possible de trouver d'autres points Pareto optimaux, mais tous ces points se trouveront sur les parties convexes de la surface de compromis.

Cette approche a l'avantage de pouvoir réutiliser tous les algorithmes classiques dédiés aux problèmes d'optimisation à un seul objectif.

Cependant cette approche a aussi deux inconvénients importants. Le premier est dû au fait que pour avoir un ensemble de points bien répartis sur le front Pareto, les différentes valeurs λ_i doivent être choisies judicieusement. Il est donc nécessaire d'avoir une bonne connaissance du problème. Le deuxième inconvénient provient du fait que cette méthode ne permet pas de calculer intégralement la surface de compromis lorsque celle-ci n'est pas convexe.

La figure 1.7(b) illustre ce cas où seulement deux solutions Pareto optimales peuvent être trouvées, sont les points y et z et il n'existe pas de valeur possible pour λ permettant de trouver le point x . En effet, cette approche ne permet pas d'approcher la totalité du front Pareto lorsque celui-ci est non convexe.

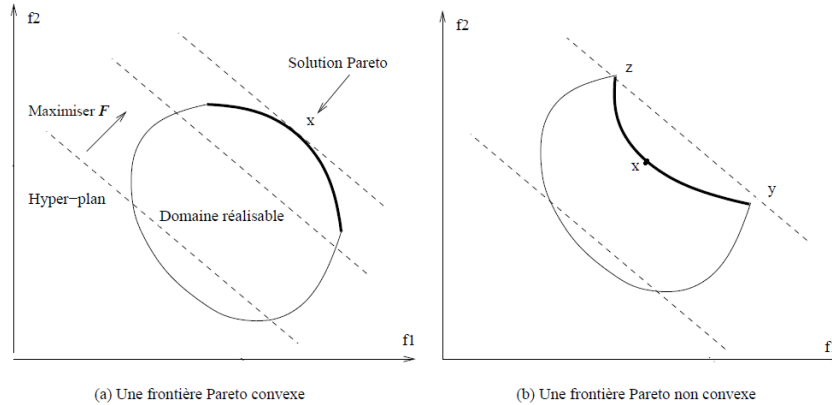


FIGURE 1.7 – l'approche d'agrégation

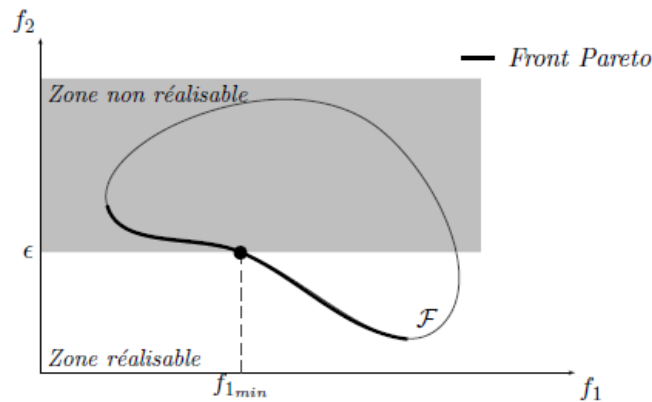
b. L'approche ϵ -contraintes

Dans cette approche, le problème consiste à optimiser une seule fonction objectif f_k sujette à des contraintes sur les autres objectifs (Convertir $p - 1$ des p objectifs du problème en contraintes). En général, l'objectif choisi est celui que le décideur souhaite optimiser en priorité

$$\begin{cases} \min_{x \in S} f_k(x), \\ f_i(x) \leq \epsilon_i, \quad i \in \{1, \dots, p\}, i \neq k; \end{cases}$$

Où $\epsilon = (\epsilon_1, \dots, \epsilon_{k-1}, \epsilon_{k+1}, \dots, \epsilon_p)$.

L'approche ϵ -contraintes doit aussi être appliquée plusieurs fois en faisant varier le vecteur ϵ pour trouver un ensemble de points Pareto optimaux. Cette approche a l'avantage de ne pas être trompée par les problèmes non convexes. Ainsi la figure 1.8 illustre un problème de minimisation à deux objectifs, le cas où un point $(\epsilon; f_{1_{min}})$ de la partie non convexe est trouvé. La figure 1.8 montre aussi comment cette approche procède. En transformant des fonctions objectifs en contraintes, elle diminue la zone réalisable par paliers. Ensuite, le processus d'optimisation trouve le point optimal sur l'objectif restant.

FIGURE 1.8 – Interprétation graphique de l'approche ϵ – *contrainte*.

L'inconvénient de cette approche réside dans le fait qu'il faut lancer un grand nombre de fois le processus de résolution. De plus, pour obtenir des points intéressants et bien répartis sur la surface de compromis, le vecteur ϵ doit être choisi judicieusement. Il est clair qu'une bonne connaissance du problème a priori est requise.

c. Programmation par but "Goal programming"

Cette méthode a été initialement conçue par Charnes et Cooper (1961) dans le cas linéaire ; elle a été prolongée par des travaux d'Ijiri (1965) et d'Ignizio (1999) dans le cas non linéaire. Dans cette approche, le décideur doit définir des buts B_i qu'il désire atteindre pour chaque objectif f_i . Ces valeurs sont introduites dans la formulation d'un problème mono-objectif dont l'objectif est de minimiser les écarts relatifs par rapport aux buts à atteindre.

$$\min_{x \in S} \sum_{i=1}^p |B_i - f_i(x)|.$$

Différentes approches sont envisageables, comme celles du min-max [Coello, 1998], ou du but à atteindre. Ces approches, bien qu'elles travaillent par agrégation des objectifs, elles permettent de générer les solutions Pareto optimales qui se trouvent dans les parties non convexe de la surface des compromis.

Il existe plusieurs manières de caractériser la distance entre un point de référence (le but) et un autre, notamment à l'aide des normes.

Une norme est définie de la manière suivante :

$$L_r = \left[\sum_{i=1}^p |B_i - f_i(x)|^r \right]^{\frac{1}{r}} \text{ avec } r \geq 1.$$

Les principales normes utilisées sont :

$$L_1 = \sum_{i=1}^p |B_i - f_i(x)|.$$

$$L_\infty = \max_{i \in \{1, \dots, p\}} (B_i - f_i(x)).$$

Cette dernière est utilisée dans l'approche Min-Max appelée aussi approche de Tchebychev [Miettinen, 1999] :

$$\begin{cases} \text{Minimiser} & \max_{i \in \{1, \dots, p\}} (B_i - f_i(x)) \\ \text{s.c.} & x \in S; \end{cases}$$

Dans cette approche, le point de référence joue un rôle fondamental, s'il est mal choisi, la recherche peut s'avérer être très laborieuse.

La figure 1.9 illustre, en dimension 2, le cas d'une recherche avec un but B fixé. Il est clair que l'approche permet de traiter les problèmes non convexes à condition que le point de référence soit choisi judicieusement. Les méthodes de résolution implémentant cette approche utilisent souvent le point idéal comme point de référence. Ce point idéal évolue donc en fonction de la recherche. En effet, plus la surface de compromis courante trouvée par la méthode se rapproche du front Pareto optimal, plus le point optimal se rapprochera du point idéal du problème.

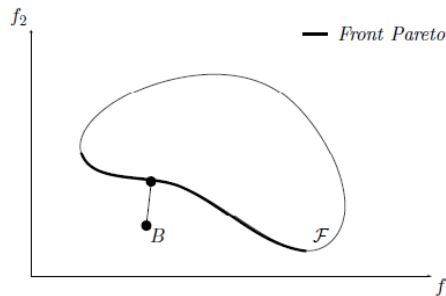


FIGURE 1.9 – Interprétation graphique de l'approche Goal programming.

1.9.2 Approches Non Pareto

Ces approches traitent séparément les différents objectifs, elles sont sensibles au paysage du front Pareto (convexité, continuité,...). Elles ne sont efficaces et faciles à implémenter que pour les problèmes (*MOLP*) avec un nombre réduit d'objectifs. Les approches les plus connues dans cette catégorie sont l'algorithme VEGA (Vector Evaluated Genetic Algorithm) et la méthode lexicographique qui exige un ordre de priorité sur les objectifs à traiter.

a. **Vector Evaluated Genetic Algorithm (V.E.G.A) :**

En 1985 Schaffer propose une extension d'un algorithme génétique simple pour la résolution d'un problème multi-objectifs. Cette méthode est appelée Vector Evaluated Genetic Algorithm (*V.E.G.A*). L'idée de l'algorithme est simple ; on considère une population de N individus. Ces N individus sont répartis en p groupes (p étant le nombre de fonctions objectif de notre problème) de $\frac{N}{p}$ individus (avec N multiple de p). A chaque groupe, on associe une fonction objectif particulière. Cette fonction permet de déterminer l'efficacité d'un individu au sein du groupe. Ensuite, les individus sont mélangés et les croisements sont opérés en tenant compte l'efficacité de chaque individu.

La figure 1.10 représente les différentes séquences de fonctionnement de la méthode. Sur cette figure, nous avons représenté les différents types de populations que l'on traite au cours du déroulement de la méthode V.E.G.A. (soit un ensemble d'individus, soit des groupes d'individus).

La description des différentes étapes de la méthode :

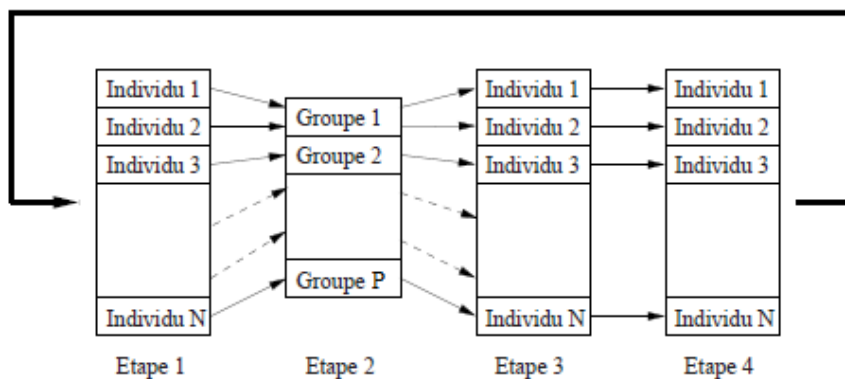


FIGURE 1.10 – Principe de l'algorithme V.E.G.A.

- Etape 1 : Itération i . initialisation d'une population de taille N .

- Etape 2 : Création de p groupes (sous-populations) chacun est composé de $\frac{N}{p}$ individus.
- Etape 3 : Calcul des efficacités, mélange des individus.
- Etape 4 : on applique l'algorithme génétique classique (croisement-mutation-sélection) puis on passe à l'itération suivante $i + 1$.

L'inconvénient de cette méthode est d'obtenir, en fin d'optimisation, une population constituée d'individus moyens dans tous les objectifs. Une telle population ne permet pas d'obtenir une surface de compromis bien dessinée. En effet, la population va se concentrer autour d'un "point" moyen. De plus, il a été montré que cette méthode est équivalente à la méthode de pondération des fonctions objectif (Coello et al. 2007). Donc, elle ne permet pas de trouver des solutions qui se trouveraient dans une concavité.

b. La méthode lexicographique :

Cette approche est proposée par Fourman (1985), elle classe les objectifs en fonction d'un ordre d'importance proposé par le décideur. Ensuite, l'optimum est obtenu en optimisant tout d'abord la fonction objectif la plus importante puis la deuxième et ainsi de suite, en intégrant les valeurs obtenues comme contraintes dans la résolution des objectifs moins prioritaire. La procédure est répétée jusqu'à ce que tous les objectifs soient traités et la solution obtenue à l'étape p sera la solution du problème.

Cette méthode procède en p étapes :

- Étape 1 :

$$\begin{cases} \text{Maximiser} & f_1(x) \\ \text{s.c.} & x \in S; \end{cases}$$

On note f_1^* la solution de ce problème. Ensuite, on transforme la première fonction objectif en contrainte d'égalité puis on maximise la seconde fonction objectif et on résout le problème suivant :

- Étape 2 :

$$\begin{cases} \text{Maximiser} & f_2(x) \\ \text{s.c.} & f_1(x) = f_1^*; \\ & x \in S. \end{cases}$$

On répète cette démarche jusqu'à la fonction objectif p et on aura dans cette étape :

- Étape p :

$$\begin{cases} \text{Maximiser} & f_p(x) \\ \text{s.c.} & f_1(x) = f_1^*; f_2(x) = f_2^*; \dots; f_{p-1}(x) = f_{p-1}^*; \\ & x \in S. \end{cases}$$

L'inconvénient majeur de cette méthode est qu'elle requiert un choix de la séquence des objectifs à optimiser. Ce choix est largement arbitraire, un peu

comme celui des coefficients de pondération, dans la méthode de pondération des fonctions objectif. Deux ordonnancements différents des fonctions objectif n'aboutissent généralement pas à la même solution.

1.9.3 Approches Pareto :

Les approches Pareto utilisent directement la notion de dominance dans la sélection des solutions générées, cette idée a été introduite initialement dans les AGs(les Algorithmes Genétiques) par Goldberg (Goldberg, 1989). Le principal avantage de ces approches est qu'elles sont capables de générer des solutions Pareto optimales dans les portions concaves de la frontière Pareto. Les AGs ont été largement utilisés pour la résolution des problèmes multi-objectifs, étant donné qu'ils travaillent sur une population de solutions. Deux objectifs doivent être pris en compte dans la résolution du problème :

- a. Converger vers la frontière Pareto : la plupart des travaux de recherche sur l'application des AGs aux problèmes multi-objectifs se sont concentrés sur l'étape de sélection. Dans cette étape, *des méthodes de ranking* sont appliquées dont le rôle est d'établir un ordre (rank) entre les individus. Cet ordre dépend de la notion de dominance et donc directement de l'optimalité Pareto. Les méthodes de ranking permettent de converger vers les solutions Pareto optimales ;
- b. Trouver des solutions diversifiées dans la frontière Pareto : les méthodes de maintien de la diversité, par la formation de niches écologiques¹ et d'espèces, peuvent être particulièrement utiles pour stabiliser des sous-populations multiples le long de la frontière Pareto.

La première génération de ces approches est présentée par les algorithmes suivants :

- M.O.G.A (Multi-Objective Genetic Algorithm).
- N.S.G.A (Non dominated Sorting Genetic Algorithm)
- N.P.G.A (Niche-Pareto Genetic Algorithm).

Après environ dix années de succès de ces algorithmes, une deuxième génération des approches Pareto est considérée maintenant comme l'état de l'art de l'optimisation multi-objectifs. Ce sont en général des approches élitistes manipulant une population secondaire externe. Parmi les approches les plus représentatives de cette génération on peut citer :

- SPEA et SPEA 2 (Strength Pareto Evolutionary Algorithm)
- NSGA-II (Non dominated Sorting Genetic Algorithm)
- PAES (The Pareto Archived Evolution Strategy)

1. Analogie avec les niches écologiques : ensemble d'individus situés dans un espace restreint.

- PESA et PESA II (The Pareto Envelope-based Selection Algorithm)
- MOMGA et MOMGA II (Multi-objective Messy Genetic Algorithm)
- MicroGA pour l'Optimisation multi-objectifs.

1.10 Conclusion

Nous avons présenté dans ce chapitre, les concepts fondamentaux de l'optimisation multi-objectifs tel que la notion de dominance ; la structure de la surface de compromis et la définition des solutions Pareto optimale. Ensuite, une classification des méthodes de résolution est présentée selon l'intervention du décideur dans le processus de résolution. Les approches agrégatives ainsi que les approches non Pareto semblent peu efficaces, elles transforment un problème d'optimisation multi-objectifs en un ou plusieurs problèmes à un seul objectif. Que ce soit sous la forme d'une somme pondérée, ou sous la forme d'une distance à un but, cette transformation permet d'utiliser facilement les méthodes d'optimisation issues de l'optimisation mono-objectif. Cependant, ces méthodes ont aussi des inconvénients. Certaines ne peuvent traiter complètement les problèmes non convexes et sont donc très sensibles à la forme du front de Pareto (concavité, non uniformité). Un autre inconvénient important est qu'il faut relancer plusieurs fois les algorithmes de résolution avec de différentes valeurs pour certains paramètres (vecteur de poids par exemple) pour obtenir plusieurs points distincts de la surface de compromis. Ces méthodes nécessitent aussi souvent une bonne connaissance du problème a priori, notamment pour fixer les vecteurs de poids ou les points de références. Par contre les méthodes Pareto issues des méta-heuristiques pour l'optimisation multi-objectifs semblent bien se prêter à ce type de problèmes, elles ont prouvées leurs efficacité pour un grand nombre de problèmes même ceux ayant des fronts de Pareto non convexes ou non uniformes.

2

L'optimisation multi-objectifs discrète

Sommaire

2.1	Introduction	29
2.2	La programmation linéaire mono-objectif	30
2.3	La programmation linéaire multi-objectifs en nombres entiers	38
2.4	Résolution des problèmes (<i>MOILP</i>)	40
2.5	Conclusion	51

2.1 Introduction

Une grande variété de problèmes d'applications réelles nécessite la présence des variables de décision en nombres entiers (ex; variables binaires : problèmes d'optimisations combinatoire, affectation, sac à doc, etc.) et en présence du plusieurs objectifs, la modélisation multi-objectifs discrète s'impose, cette dernière est de nature complètement différente de celle des problèmes multi-objectifs continus et le traitement nécessite naturellement plus d'attention et de rigueur quand on procède à la résolution.

Dans ce chapitre nous présentons en premier lieu un aperçu sur les notions de base de la programmation linéaire en nombre entiers, nous présenterons par la suite les principaux résultats de la programmation linéaire multi-objectifs

en nombres entiers ainsi que quelques méthodes de recherche des solutions efficaces discrètes.

2.2 La programmation linéaire mono-objectif

La programmation linéaire mono-objectif est la source fondamentale de la programmation linéaire multi-objectifs, la forme générale d'un problème de programmation linéaire peut être donnée par :

$$(PL) \begin{cases} \max \text{ (ou min)} & z = cx \\ \text{s.c.} & x \in S, \end{cases} \quad (2.1)$$

où $S = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$; $A \in \mathbb{R}^{(m \times n)}$, $c \in \mathbb{R}^{1 \times n}$, et $b \in \mathbb{R}^{m \times 1}$.

Dans le cas où certaines variables sont entières, on a alors un problème de programmation linéaire mixte (*MILP*) qui s'écrit comme suit :

$$(MILP) \begin{cases} \max & (cx + hy) \\ \text{s.c.} & Ax + Gy = b \\ & x \geq 0 \\ & y \in \mathbb{N}. \end{cases} \quad (2.2)$$

Si toutes les variables sont entières, on aura un problème de programmation linéaire en nombres entiers (*ILP*) donné par :

$$(ILP) \begin{cases} \max & z = cx \\ \text{s.c.} & x \in D \end{cases} \quad (2.3)$$

où $D = S \cap \mathbb{Z}^n$, \mathbb{Z} est l'ensemble des nombres entiers relatifs.

Si toutes les variables sont restreintes à être égales à 0 ou 1, on a un problème de programmation binaire écrit comme suit :

$$(LPBIL) \begin{cases} \max & z = cx \\ \text{s.c.} & Ax = b \\ & x \in \{0, 1\}^n \end{cases} \quad (2.4)$$

2.2.1 Complexité

Contrairement à la programmation linéaire en variables continues où on s'intéresse seulement aux solutions sommets du polyèdre, les solutions optimales du problème (*ILP*) peuvent se trouver à "l'intérieur" de l'enveloppe et par conséquent la recherche d'une solution optimale est souvent NP-difficile; le problème d'existence d'une solution $x \in D$ peut être NP-complet : on n'espère pas, en général, qu'il existe un algorithme en temps polynomial pour ces problèmes.

2.2.2 Résultats fondamentaux de la programmation linéaire

La forme standard du problème (*PL*) est définie par :

$$(PL) \begin{cases} \max \text{ (ou min) } z = cx \\ \text{s.c. } Ax = b \\ x \geq 0. \end{cases} \quad (2.5)$$

Nous supposons que la matrice A est de rang m ($m \leq n$). Soit B une sous-matrice carrée constituée de m colonnes indépendantes de A . On associe à cette base les décompositions $A = [B|N]$, $x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$;

Définition 2.1 Une solution x est dite admissible si elle vérifie les contraintes

$$\begin{cases} Ax = b \\ x \geq 0 \end{cases} \quad (2.6)$$

Définition 2.2 On appelle *variables de base*, les m variables de \mathbb{R}^n correspondant au système de m vecteurs libres définis par les contraintes linéaires. Les $(n - m)$ variables restantes sont appelées *variables hors-base*. Une solution est dite de base si elle est constituée de variables de base.

Définition 2.3 Toute solution de base vérifiant l'équation (2.6) est une solution de base admissible.

Notons x_B les variables de base correspondant aux m variables positives ou nulles et x_N les variables hors-base correspondant aux $n - m$ variables nulles. I l'ensemble des indices de base et J l'ensemble des indices hors-base.

Définition 2.4 Une solution de base $x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$ est dégénérée si x_B a au moins une composante nulle.

En mettant en évidence la décomposition de la matrice A et du vecteur x , le problème (PL) peut être réécrit sous la forme :

$$(P_B) \begin{cases} \max \text{ (ou min) } & z = c_B x_B + c_N x_N \\ \text{s.c.} & Bx_B + Nx_N = b \\ & x_B, x_N \geq 0 \end{cases} \quad (2.7)$$

D'après (P_B) on a :

$$Bx_B + Nx_N = b \Rightarrow x_B = B^{-1}b - B^{-1}Nx_N.$$

En substituant cette expression dans la fonction objectif, on trouve :

$$z = c_B(B^{-1}b - B^{-1}Nx_N) + c_N x_N$$

et par conséquent :

$$z - c_B B^{-1}b = (c_N - c_B B^{-1}N)x_N.$$

Comme x_B est admissible, on obtient :

$$\begin{cases} \max \text{ (ou min) } & z - c_B B^{-1}b = (c_N - c_B B^{-1}N)x_N \\ \text{s.c.} & B^{-1}b - B^{-1}Nx_N \geq 0 \\ & x_N \geq 0 \end{cases} \quad (2.8)$$

Une condition nécessaire et suffisante en l'absence de dégénérescence, pour que x_B soit une solution de base optimale est que $(c_N - c_B B^{-1}N) \leq 0$ pour un problème de maximisation et $(c_N - c_B B^{-1}N) \geq 0$ pour un problème de minimisation.

2.2.3 Algorithme du simplexe (Dantzig. G.B 1947)[22]

L'algorithme du simplexe est introduit en 1947 par G.B. Dantzig [22], il décrit un moyen intelligent de se déplacer d'une solution de base admissible à une autre améliorant la valeur de la fonction objectif, jusqu'à trouver une solution optimale en un nombre fini d'étapes. Depuis, les chercheurs ne cessent de développer et d'implémenter différentes techniques en utilisant toujours cet algorithme.

Nous supposons dans un premier temps connaître une base réalisable B , alors l'écriture canonique du problème par rapport à la base B est donnée par le tableau suivant :

$\begin{array}{cccc} 1 & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{array}$	$\bar{A}^N = (A^B)^{-1}A^N$	$\bar{b} = (A^B)^{-1}b$
$\begin{array}{cccc} 0 & \cdots & 0 & 0 \end{array}$	$\bar{c}_N = c_N - c_B(A^B)^{-1}A^N$	$z - c_B(A^B)^{-1}b$

TABLE 2.1 – Ecriture canonique de (PL) par rapport à B

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

TABLE 2.2 – Tableau simplexe optimal associé à la base B

On considère un problème de maximisation, les différentes étapes de l'algorithme sont résumées dans la présentation suivante :

Algorithm 1: Algorithme du simplexe (Dantzig [22])

Entrées: A, b, c : les paramètres : contraintes et fonction objectif

Sorties: X_{opt} : Solution optimale du problème et du Tableau 2.2

Phase Initiale

Début

On établit le tableau du simplexe initial en passant à l'écriture canonique par rapport à la base B (voir Tableau 2.1)

Les contraintes s'écrivent : $\bar{A}X = \bar{b}$; $\bar{A} = B^{-1}A$, $\bar{b} = B^{-1}b$.

$\bar{c}_N \leftarrow c_N - c_B B^{-1}N$

SI $\bar{c}_N \leq 0$ **ALORS**

la solution est optimale et l'algorithme prend fin.

SINON

Considérons l'indice j tel que : $(\bar{c}_N)_j = \max_{k \in J} (\bar{c}_N)_k \mid (\bar{c}_N)_k > 0$

SI $a_{ij} < 0, \forall i \in \{1, \dots, m\}$ **ALORS**

il n'y a pas de solution optimale finie et la valeur de la fonction n'est pas bornée. $z \rightarrow \infty$;

SINON

La variable x_j rentre dans la base; l'indice ℓ de la variable qui sort de la base est donné par :

$$\frac{\bar{b}_\ell}{\bar{a}_{\ell k}} = \min_{i \in I} \left\{ \frac{\bar{b}_i}{\bar{a}_{ik}} \right\}, \bar{a}_{ik} > 0;$$

Faire, $\bar{I} \leftarrow I \setminus \ell \cup j$

Un nouveau tableau de simplexe est établi en appliquant les formules suivantes :

$$a_{ik}^* \leftarrow a_{ij} - \frac{a_{ik}a_{lk}}{a_{lk}}, \quad i \neq k, \quad j \neq l;$$

$$a_{kj}^* \leftarrow \frac{a_{lj}}{a_{lk}}, \quad j \neq k;$$

$$a_{il}^* \leftarrow -\frac{\bar{a}_{ik}}{a_{lk}}, \quad i \neq k;$$

$$a_{lk}^* \leftarrow \frac{1}{a_{lk}}$$

$$c_j = c_j - c_\ell \times \frac{a_{\ell j}}{a_{\ell \ell}}, \quad j = 1, \dots, n$$

Retour:

Arrêt: L'algorithme prend fin quand la solution est optimale ou l'opération pivot est impossible. Voir le tableau optimal (2.2) à une itération k

2.2.4 Méthodes de Résolutions des Problèmes (*ILP*)

Dans la littérature, il existe deux techniques pour résoudre les problèmes (*ILP*), la méthode des coupes et la méthode de séparation et évaluation "Branch and Bound". Les deux sections suivantes présentent le détail de ces deux approches.

1. Les méthodes des coupes : Les méthodes des coupes sont des algorithmes exacts pour les problèmes (*ILP*). Ces méthodes résolvent une séquence de relaxations du problème (*ILP*), les solutions obtenues sont progressivement améliorées pour donner une meilleure approximation de la solution optimale. Pour les grandes instances, le problème (*ILP*) ne peut pas être résolu à l'optimum, les algorithmes de coupes produisent alors des solutions relativement proches d'une solution optimale en un temps d'exécution raisonnable. Nous décrivons dans cette section deux types de coupes les plus utilisées.

a. **La coupe de Dantzig**[22]

Cette coupe est utilisée lorsque la solution optimale du problème (*PL*) n'est pas entière, dans ce cas la coupe de Dantzig est ajoutée comme une contrainte au problème (*PL*) afin d'obtenir une nouvelle solution optimale entière. Cette coupe est non profonde, ce qui ralentit la procédure de résolution. Malgré cela elle est utilisée dans les méthodes exactes d'optimisation multi-objectifs pour réduire le domaine d'admissibilité.

La coupe est formulée par :

$$\sum_{j \in J} x_j \geq 1,$$

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

b. **La coupe fractionnaire de Gomory**[36]

L'idée principale de cette méthode est d'ajouter des contraintes linéaires qui n'excluent aucune solution entière réalisable, une par une jusqu'à ce que la solution optimale de la relaxation soit entière. Dans une première étape, on résout le programme relaxé (*LP*), on cherche une solution de base optimale en utilisant la méthode du simplexe, si elle existe, on choisit une variable de base non entière et on génère une inégalité sur la contrainte associée à cette variable afin de couper la région de faisabilité courante.

Étant donnée une base optimale B du problème relaxé (*LP*), le tableau optimal correspondant est donné par le tableau 2.2, où : $\pi = c_B B^{-1}$: est dit vecteur multiplicateur relatif à la base B .

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

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

La i ème ligne du tableau optimal est donnée par :

$$x_j + \sum_{j \in N} \bar{a}_{ij} x_j = \bar{b}_i, \quad (2.9)$$

où

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

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

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

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

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

$$\sum_{j \in N} [\bar{a}_{ij}] x_j \leq \sum_{j \in N} \bar{a}_{ij} x_j.$$

De l'équation (2.9) on a :

$$x_j + \sum_{j \in N} [\bar{a}_{ij}] x_j \leq \bar{b}_i.$$

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

$$x_j + \sum_{j \in N} [\bar{a}_{ij}] x_j \leq [\bar{b}_i]. \quad (2.10)$$

En soustrayant (2.9) de (2.10) on obtient :

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

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

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

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

2. Méthode de séparation et évaluation "Branch & Bound"

La méthode de séparation et évaluation "Branch and Bound" a été développée par Land et Doig (1960), spécialement élaborée pour des problèmes en variables discrètes (*ILP*), le principe de cette méthode consiste à subdiviser l'ensemble S (l'ensemble de solutions admissibles) $S = \{x \in \mathbb{R} | Ax \leq b\}$, en un nombre fini de sous-ensembles S_i ; généralement on prend

$$S = \bigcup_i S_i \text{ avec } S_i \cap S_j = \emptyset, \forall i \neq j.$$

Ces subdivisions successives sont représentées à l'aide d'une arborescence de racine S_0 et des "noeuds" S_i présentant les sous-ensembles de solutions effectués. Le déroulement de la méthode est constitué princi-

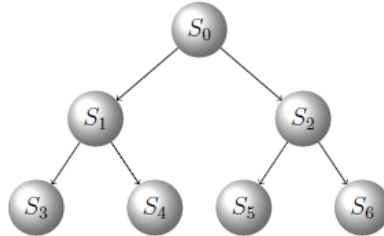


FIGURE 2.1 – L'arborescence de Branch and Bound.

palement de trois procédures :

- (a) **Procédure de séparation** : La racine S_0 de l'arborescence représente la solution initiale du problème (*ILP*) en relaxant la contrainte d'intégrité, si sa solution est entière, la méthode s'arrête. Sinon, on choisit une variable x_j non entière, soit la variable ayant la plus grande partie fractionnaire, on divise le problème en deux sous-problèmes en ajoutant la contrainte $x_j \leq \lfloor x_j \rfloor$ pour l'un et la contrainte $x_j \geq \lfloor x_j \rfloor + 1$ pour l'autre, tel que $\lfloor x_j \rfloor$ est la partie entière de la variable choisie pour le branchement. Ainsi, en résolvant les deux sous-problèmes représentés par les noeuds S_1 et S_2 et en prenant la meilleure solution trouvée, ce principe de

séparation peut être appliqué de manière récursive à chacun des sous-ensembles de solutions obtenus.

- (b) **Procédure d'évaluation** : l'évaluation d'un noeud de l'arborescence a pour but de déterminer l'optimum (une borne supérieure pour un problème de maximisation ou inférieure pour un problème de minimisation) de l'ensemble des solutions réalisables associé au noeud en question. Ou de prouver que cet ensemble ne contient pas de solution intéressante pour la résolution du problème initial. La solution optimale du sous-problème associé à un noeud donné est appelée solution partielle.
- (c) **Procédure Stérilisation** : Le but de cette procédure est d'éviter l'examen de tous les noeuds de l'arborescence. Dans le cas où la borne supérieure de la solution optimale du sous-problème traité est inférieure à la borne supérieure globale (ie. la meilleure solution trouvée jusqu'à présent), on est certain que toute solution réalisable de ce sous-problème ne sera pas meilleure que l'optimum global courant, il est donc inutile d'effectuer la séparation de son ensemble de solutions. On peut également arrêter la recherche dans un noeud lorsque le sous-problème qui lui est associé est non réalisable.

Cette méthode peut également être appliquée aux problèmes avec variables binaires (0 – 1) et aux problèmes à variables mixtes (*MILP*).

2.3 La programmation linéaire multi-objectifs en nombres entiers

2.3.1 Formulation mathématique d'un problème MOILP

Un problème de programmation linéaire multi-objectifs en nombre entiers (*MOILP*) est constitué d'un espace de décisions discret non convexe défini par un ensemble de contraintes linéaires sur lequel plusieurs critères souvent conflictuels sont optimisés.

Mathématiquement, ce problème peut être formulé par :

$$(MOILP) \begin{cases} \text{“max”} & Z(x) = Cx \\ t.q & x \in D. \end{cases}$$

Où $D = S \cap \mathbb{Z}$, \mathbb{Z} étant l'ensemble des nombres entiers relatifs, $S = \{x \in \mathbb{R}^n | Ax \leq 0\}$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, C est une matrice de dimension $p \times n$ d'éléments réels avec des vecteurs lignes $c^i \in \mathbb{R}^n$, $i = 1, 2, \dots, p$.

Notons par IE l'ensemble de toutes les solutions efficaces et par $\mathcal{Z}(IE)$ l'ensemble de tous les points non dominés du problème ($MOILP$). Notons que les notions de bases concernant, la dominance, les points caractéristiques (Points idéal, point nadir, etc.) présentées dans le chapitre précédent restent valables pour les problèmes ($MOILP$).

2.3.2 Solutions supportées/non supportées

Sur le front Pareto d'un problème ($MOILP$), deux types de solutions peuvent être différenciées : les solutions supportées (supported efficient solutions) notée SE et les solutions non supportées (non-supported efficient solutions) notée NSE . Les premières sont celles situées sur l'enveloppe convexe de l'ensemble des solutions réalisables et peuvent être trouvées par la résolution du programme mathématique suivant :

$$(P_\lambda) \left\{ \begin{array}{l} \max \sum_{i=1}^p \lambda_i c^i x \\ t.q \end{array} \quad x \in S. \right.$$

Avec $\lambda_i \geq 0$ pour $i = 1, \dots, p$ et $\sum_{i=1}^p \lambda_i = 1$;

l'ensemble de ces solutions peut être généré par la résolution de (P_λ) pour différentes valeurs du vecteur de poids λ .

Théorème 2.1 (*Théorème de Geoffrion*) *Étant donnée le problème (P_λ) , alors la solution x^* est optimale au sens de Pareto si et seulement si x^* est une solution optimale du problème paramétrique (P_λ) .*

L'obtention des solutions supportées est relativement aisée. Néanmoins, il existe en général d'autres solutions qui, bien qu'efficaces, ne peuvent être obtenues par la résolution d'un programme (P_λ) . En effet, ces solutions dites *non supportées*, sont situées à l'intérieur de l'enveloppe convexe de $\mathcal{Z}(IE)$. Il existe plusieurs approches permettant de générer ces solutions non supportées comme par exemple les approches de programmation par but fondées sur l'utilisation d'une norme de Tchebycheff. Cependant, ces approches modifient la structure du problème combinatoire qui perd ainsi ses éventuelles propriétés remarquables. L'obtention des solutions non supportées est donc en général encore plus difficile.

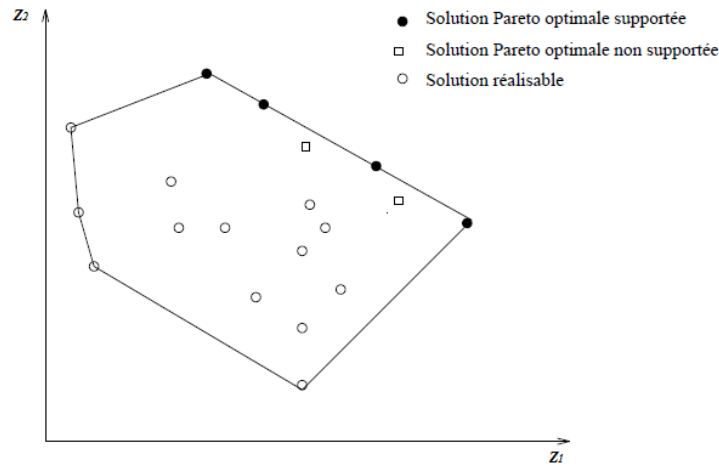


FIGURE 2.2 – Solutions supportées et non supportées

2.4 Résolution des problèmes (*MOILP*)

Plusieurs chercheurs, citons en particulier, Steuer et Choo (1986), Klein et Hannan (1982), Crema et Sylva (2003, 2004), Gupta et Malhotra (1992), Abbas et Moulaï (1999), Abbas et Chaabane (2002), motivés par de nombreuses stratégies (applications), se sont intéressés à caractériser totalement ou partiellement l'ensemble des solutions efficaces du problème (*MOILP*). Dans cette partie, quelques méthodes de résolution des problèmes (*MOILP*) sont exposées, quelques-unes nécessitent la présence du décideur (méthodes interactives), et d'autres donnent l'ensemble des solutions efficaces sans intervention de ce dernier.

2.4.1 Méthode de Steuer et Choo [67]

Cette méthode est interactive, elle peut être utilisée pour tout problème de programmation multi-objectifs linéaire ou non avec éventuellement des variables entières. Soit le problème

$$(P) \begin{cases} \text{“max”} & F = H(x) \\ \text{s.c.} & x \in D \end{cases}$$

Où $H(x)$ est un vecteur de fonction quelconque (linéaire ou non) et D est non convexe.

La méthode de résolution se décompose en trois étapes :

- Étape 1 : Consiste à calculer un point cible, par exemple le point idéal. La recherche des solutions efficaces se fait par quadrillage de l'ensemble D au moyen d'un ensemble diversifié de valeurs du vecteur paramètres λ . Pour chaque valeur de λ , la distance de Tchebychev calculée par rapport au point cible est minimisée, ceci donne un sous-ensemble de solutions efficaces supportées. Soit x^1 la solution efficace choisie par le décideur dans ce sous-ensemble.
- Étape 2 : Déterminer ensuite un deuxième ensemble de valeurs du vecteur λ de façon à quadriller le voisinage élargi de x^1 . Une nouvelle solution efficace, soit x^2 , est désignée par le décideur comme dans la première étape.
- Étape 3 : Continuer la procédure mais en focalisant le paramètre λ dans le voisinage restreint du nouveau compromis jusqu'à l'arrêt du processus de décisions.

2.4.2 La méthode de Klein & Hannan [47]

La technique présentée par D. Klein & E. Hannan est une méthode interactive, pour générer séquentiellement un sous-ensemble efficace ou l'ensemble de toutes les solutions efficaces du problème (*MOILP*). Elle consiste à résoudre progressivement des programmes linéaires mono-objectif (*ILP*) avec des contraintes ajoutées à chaque itération. La méthode génère des solutions efficaces de façon que l'utilisateur ne soit pas obligé de déterminer entièrement l'ensemble des solutions efficaces s'il ne s'intéresse qu'à quelques solutions. Une brève présentation de cette méthode est résumée par l'algorithme ci-dessous.

L'algorithme de la méthode :

Étape 1 : Choisir arbitrairement un critère $i, i \in \{1, 2, \dots, p\}$ du problème (*MOILP*) et résoudre le problème mono-objectif suivant :

$$(P_0) \begin{cases} \max & Z_i = c^i x \\ \text{s.c.} & x \in D. \end{cases}$$

Si la solution de (P_0) est unique, alors elle est efficace et elle est l'unique élément dans la liste initiale des solutions efficaces IE_0 , sinon soit $\zeta(P_0)$ l'ensemble des solutions optimales de (P_0) , par comparaison deux à deux des vecteurs critères associés, retenir celles qui sont non dominées pour construire IE_0 l'ensemble des solutions efficaces correspondant à $\zeta(P_0)$.

Étape j : ($j \geq 1$) résoudre le problème (P_j) défini par :

$$(P_j) \begin{cases} \max & Z_i = c^i x \\ \text{t.q} & x \in D. \\ & \bigwedge_{k=1}^r \left(\bigvee_{s=1, s \neq i}^p (c^s x \geq c^s \tilde{x}_k + \epsilon^s) \right). \end{cases}$$

Où $0 < \epsilon^s \leq 1$ et $\tilde{x}_k, k=\overline{1,r}$ sont les solutions efficaces obtenues dans les itérations $0, 1, \dots, j-1$.

Si $\epsilon^s < 1$, la méthode génère un sous-ensemble de l'ensemble des solutions efficaces, et si $\epsilon^s = 1$, la procédure donne toutes les solutions efficaces.

Les contraintes supplémentaires ajoutées à chaque itération assurent que les solutions optimales du problème (P_j) , si elles existent, seront meilleures que toutes les solutions efficaces $\{\tilde{x}_k, k=\overline{1,r}\}$ sur au moins un critère $s \neq i$.

La liste des solutions efficaces obtenue à l'étape j est

$$IE_j = \bigcup_{k=0}^{j-1} IE_k.$$

La procédure s'arrête lorsque le problème (P_j) devient irréalisable. Il est clair que la procédure est finie étant donné qu'on élimine au moins une solution à chaque étape et qu'il existe un nombre fini (D est borné) de solutions admissibles.

2.4.3 La méthode de Sylva & Crema [68]

Cette méthode est une variante de celle de Klein & Hannan présentée précédemment. Son principe repose sur la résolution d'une succession de programmes linéaires en nombres entiers optimisant à chaque étape une combinaison positive des critères. Un ensemble de contraintes est rajouté à chaque fois assurant la détection d'une nouvelle solution efficace.

L'algorithme génère aussi des sous-ensembles de solutions efficaces qui peuvent être utiles dans la construction des méthodes interactives pour des problèmes concrets de grande dimension.

Proposition 2.1 Soient x^1, x^2, \dots, x^k des solutions efficaces du problème (MOILP). Posons $D_j = \{x \in \mathbb{Z}^n \mid c^i x \leq c^i x^j, \forall i = \overline{1,p}\}, j = \overline{1,k}$. Si x^* est efficace pour le problème multi-objectifs :

$$(P_k) \begin{cases} \text{“max”} & Cx \\ x \in & \left(D - \bigcup_{j=1}^k D_j \right) \end{cases}$$

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

Corollaire 1 ([68]) Soient x^1, x^2, \dots, x^k des solutions efficaces du problème (*MOILP*) et $D_j = \{x \in \mathbb{Z}^n \mid c^i x \leq c^i x^j, \forall i = \overline{1, p}\}$ ($j = \overline{1, k}$). Si x^* est une solution optimale pour le problème mono-objectif :

$$(P_k^\lambda) \begin{cases} \text{“max”} & \sum_{i=1}^p \lambda_i c^i x \\ x \in & \left(D - \bigcup_{j=1}^k D_j \right) \end{cases}$$

pour certaines valeurs du vecteur $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_p)$, $\lambda > 0$, alors x^* est une solution efficace pour le problème (*MOILP*).

L’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 résoudre le problème (P_1^λ) :

$$(P_1^\lambda) \equiv \max \left\{ \sum_{i=1}^p \lambda_i c^i x \mid x \in D \right\}.$$

Deux cas se présentent :

Cas 1. Si (P_1^λ) est irréalisable, alors (*MOILP*) l’est aussi.

Cas 2. Sinon, une solution x^1 est trouvée et elle est efficace en vertu du corollaire précédent.

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

Après k étapes du processus :

Cas 1. Si (P_k^λ) est irréalisable, alors l’algorithme prend fin.

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

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

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

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

$$(P_{k+1}^\lambda) \left\{ \begin{array}{l} \max \sum_{i=1}^p \lambda_i c^i x \\ x \in D \\ c^i x \geq (c^i x^j + 1)y_i^j - M_i(1 - y_i^j), i = \overline{1, p}, j = \overline{1, k} \\ \sum_{i=1}^p y_i^j \geq 1, y_i^j \in \{0, 1\}, i = \overline{1, p}, j = \overline{1, k} \end{array} \right.$$

Remarque 1 Pour les problèmes de grandes tailles, l'énumération de toutes les solutions non dominées devient très coûteuse en terme de temps de calcul. Dans ce cas, l'intérêt se porte sur une partie seulement des solutions efficaces, cette partie peut être obtenue en intégrant à la procédure une étape d'interaction avec le décideur. Cette étape a pour objectif d'éliminer des solutions efficaces que le décideur juge insatisfaisantes. Dans ce cas le problème (P_{k+1}^λ) devient :

$$(P_{k+1}^\lambda) \left\{ \begin{array}{l} \max \sum_{i=1}^p \lambda_i c^i x \\ x \in D \\ c^i x \geq (c^i x^j + f_i)y_i^j - M_i(1 - y_i^j) \quad i = \overline{1, p}, \quad j = \overline{1, k} \\ \sum_{i=1}^p y_i^j \geq 1 \quad y_i^j \in \{0, 1\} \quad i = \overline{1, p} \quad j = \overline{1, k} \end{array} \right.$$

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

Remarque 2 Le nombre de problèmes (P^λ) devant être résolu est donné par le nombre des solutions non dominées plus un dernier problème irréalisable. En effet, cette méthode est fortement liée au nombre de critères et le nombre de solutions non dominées du problème étudié. Car, généralement, plus le nombre de critères est grand, plus le cône engendré est restreint, ce qui fait augmenter le nombre de solutions non dominées du problème entraînant le traitement d'un nombre considérable de problèmes (P^λ) .

2.4.4 Méthode de R. Gupta & R. Malhotra [39]

Cette méthode a été proposée par R. Gupta & R. Malhotra [39], ayant pour but de trouver toutes les solutions efficaces du problème *MOILP*. Mais

il s'est avéré qu'une erreur au niveau du test d'arrêt empêche dans certains cas l'algorithme de donner tous les points efficaces du problème étudié. Un contre exemple a été présenté par M. Moulay [54] ainsi que par D. Chaabane [15].

Nous présentons dans ce qui suit, quelques notations et résultats théoriques utilisés dans la méthode :

Considérons le problème initial, correspondant à l'itération $k = 1$:

$$(P_1) \begin{cases} \max & Z_1 = c^1 x; \\ \text{s.c.} & x \in D, \end{cases}$$

où $D = \mathbb{Z}^n \cap S = \{x \in \mathbb{R}^n \mid Ax = 0, x \geq 0\}$.

Notons qu'à la place du problème (P_1) , on pourra considérer d'une manière analogue le problème $(P_i)_{i=\overline{2,p}}$ qui maximise $c^i x$ sur D .

On définit pour $k \geq 2$:

- $S_k = \{x \in \mathbb{R}^{n_k} \mid A_k x = b_k, x \geq 0\}$ comme étant la région tronquée courante obtenue après avoir appliqué la coupe :

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

, et éventuellement des coupes successives de Gomory, avec j_{k-1} un indice hors base quelconque.

- $x^k = (x_{k,j})$: la $k^{\text{ème}}$ solution optimale entière du problème (P_1) obtenue dans S_k .
- B_k : est la base associée à x^k .
- $a_{k,j}$: est le vecteur d'activité de x^k .
- $I_k = \{j \mid a_{k,j} \in B_k\}$: l'ensemble des indices des variables de base.
- $N_k = \{j \mid a_{k,j} \notin B_k\}$: l'ensemble des indices des variables hors base.
- $y_{k,j} = (B_k)^{-1} a_{k,j}$.
- $\Gamma_k = \{j \in N_k \mid Z_{k,j}^1 - c_j^1 > 0 \text{ et } Z_{k,j}^i - c_j^i < 0 \text{ pour au moins un } i \in \{2, \dots, p\}\}$;

où $Z_{k,j}^i = c_{B_k}^i y_{k,j}$ et c_j^i est la j^{me} composante du vecteur c^i et $c_{B_k}^i$ est le vecteur des coefficients coûts des variables de base associées à B_k du vecteur c^i .

Définition 2.5 Une arête E_{j_k} , $j_k \in N_k$ incidente à x^k est définie comme étant l'ensemble :

$$E_{j_k} = \left\{ X = (x_i) \in \mathbb{R}^{n_k} \left| \begin{array}{l} x_i = x_i^k - \theta_{j_k} y_{k,i j_k}, i \in B_k; \\ x_{j_k} = \theta_{j_k}; \\ x_l = 0, \forall l \in N_k \setminus \{j_k\} \end{array} \right. \right\}$$

où : $0 \leq \theta_{j_k} \leq \theta = \min_{i \in B_k} \left\{ \frac{x_i^k}{y_{k,ij_k}} \mid y_{k,ij_k} > 0 \right\}$.

Les points entiers se trouvant sur l'arête E_{j_k} sont identifiés de telle sorte que θ_{j_k} soit entier et $\theta_{j_k} \times y_{k,ij_k}$ entier $\forall i \in B_k$.

Les résultats suivants suggèrent une coupe qui peut être vue comme une généralisation de la coupe classique de Dantzig ; son avantage est de tronquer toute une arête au lieu d'une seule solution.

Théorème 2.2 *Toutes les solutions entières réalisables du problème (P_1) alternatives à x^1 sur l'arête E_{j_1} , $j_1 \in \Gamma_1$ de la région S (ou la région tronquée S_k) et émanant de x^1 , appartiennent au demi espace ouvert :*

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

Théorème 2.3 *Une solution entière réalisable du problème (P_1) qui n'est pas sur l'arête E_{j_1} , $j_1 \in \Gamma_1$ à travers x^1 , appartient au demi espace fermé :*

$$\sum_{j \in N_1 \setminus \{j_1\}} x_j \geq 1. \quad (2.11)$$

Corollaire 2 *Une solution entière réalisable du problème (P_1) qui n'est pas sur l'arête E_{j_k} , $j_k \in \Gamma_k$, $k \geq 2$ à travers x^k dans la région tronquée S_k , appartient au demi espace fermé :*

$$\sum_{j \in N_k \setminus \{j_k\}} x_j \geq 1. \quad (2.12)$$

L'algorithme de la méthode : Notons par $SND_0(P)$ l'ensemble des solutions non dominées du problème (P) générées à l'étape 1, et par $SND(P)$ l'ensemble des solutions potentiellement non dominées générées aux étapes k , $k \geq 2$.

Étape 1. Résoudre le problème (P_1) .

Cas 1. Si la solution optimale de (P_1) est unique, soit x^1 , alors le vecteur critère $(z_1^1, z_1^2, \dots, z_1^p)$ qui lui est associé est enregistré dans $SND_0(P)$ comme étant le premier p -uplet non dominé.

Cas 2. Si la solution trouvée n'est pas unique, alors déterminer toutes les solutions qui lui sont alternatives, et par comparaison deux à deux des vecteurs critères associés, éliminer ceux qui sont dominés pour construire l'ensemble $SND_0(P)$ des premiers vecteurs non

dominées.

$(z_1^1, z_1^2, \dots, z_1^p)$ est choisit comme étant le premier p -uplet non dominé pris dans l'ordre lexicographique (i.e. le vecteur qui a la plus grande valeur de z^2 , en cas d'égalité choisir celui qui a la plus grande valeur de z^3 , et ainsi de suite), soit x^1 la solution correspondante.

Étape 2. Construire l'ensemble Γ_1 , choisir un indice quelconque $j_1 \in \Gamma_1$ et trouver le rapport minimum θ de l'opération pivot.

Cas 1. Si $\theta < 1$, alors choisir un autre $j_1 \in \Gamma_1$. Car aucune solution entière réalisable ne peut être obtenue sur l'arête E_{j_1}

Cas 2. Si $\theta \geq 1$, déterminer toutes les solutions entières se trouvant sur l'arête E_{j_1} , évaluer les p critères en chacune d'elles et éliminer ceux qui sont dominés pour construire l'ensemble $SND(P)$ des solutions potentiellement non dominées générées à l'étape 2.

éliminer l'arête E_{j_1} par la coupe : $\sum_{j \in N_1 \setminus \{j_1\}} x_j \geq 1$.

Cas 3. Si pour tout $j_1 \in \Gamma_1$ on a $\theta < 1$ alors choisir arbitrairement un $j_1 \in \Gamma_1$ et appliquer la coupe $\sum_{j \in N_1 \setminus \{j_1\}} x_j \geq 1$.

Utiliser la méthode duale du simplexe et des coupes successives de Gomory si nécessaire pour obtenir une solution entière x^2 dans la région tronquée S_2 et mettre à jour $SND(P)$.

Étape k. ($k \geq 3$) Choisir un indice $j_{k-1} \in \Gamma_{k-1}$, déterminer toutes les solutions entières se trouvant sur l'arête $E_{j_{k-1}}$, lorsqu'elles existent, évaluer tous les critères sur chacune des solutions trouvées et éliminer les p -uplets dominés et mettre à jour l'ensemble $SND(P)$.

Tronquer l'arête $E_{j_{k-1}}$ par la coupe : $\sum_{j \in N_{k-1} \setminus \{j_{k-1}\}} x_j \geq 1$.

et chercher de nouveau une solution entière dans la région tronquée S_k , soit x^k . Mettre à jour l'ensemble $SND(P)$.

Étape finale n+1. L'algorithme prend fin dans l'un des cas suivants :

- a. $\Gamma_n = \emptyset$ et $Z_{nj}^1 - c_j^1 > 0, \forall j \in N_n$.
- b. $\Gamma_n \neq \emptyset$ mais pour tout $j_n \in N_n$, les solutions entières se trouvant sur l'arête E_{j_n} ne sont pas efficaces.

L'ensemble courant des solutions non dominées $SND(P)$ union $SND_0(P)$ enregistré à l'étape 1, fournit l'ensemble de toutes les solutions non dominées du problème.

2.4.5 Méthode de M. Abbas & M. Moulaï [3]

Cette méthode, dénommée "MODILIM", a été proposée par M. Abbas & M. Moulaï [3] pour déterminer toutes les solutions efficaces du problème (*MOILP*). Elle peut être vue comme une alternative à celle de Gupta & Malhotra [39], où les auteurs ont proposé un autre test d'arrêt permettant à l'algorithme de fournir toutes les solutions efficaces.

Notations :

- $SND(P)$: l'ensemble des solutions potentiellement non dominées générées aux étapes k , $k \geq 1$ et nb_{jk} , le nombre de solutions entières sur l'arête E_{jk} y compris x^k .
- $\Omega_k = \{j \in N_k \mid Z_{k,j}^1 - c_j^1 = 0\}$.

Développement de la méthode

Étape 1. Résoudre le problème (P_1) et trouver la solution optimale entière x^1 sur S_1 et construire l'ensemble Ω_1 .

Étape 2. Tester l'ensemble Ω_1 .

- Cas 1.** Si $\Omega_1 = \emptyset$, x^1 est l'unique solution optimale sur S_1 . Soit $(z_1^1, z_1^2, \dots, z_1^p)$ le vecteur critère correspondant, il est enregistré dans $SND(P)$ comme étant le premier p -uplet non dominé.
Tronquer le point x^1 par la coupe de Dantzig :

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

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

- Cas 2.** Si $\Omega_1 \neq \emptyset$, choisir un indice quelconque $j_1 \in \Omega_1$ et calculer le nombre θ de l'opération pivot.

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

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

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

(b) Si pour tout $j_1 \in \Omega_1$, on a $\theta < 1$, alors choisir un indice quelconque $j_1 \in \Omega_1$ et appliquer la coupe :
$$\sum_{j \in N_1 \setminus \{j_1\}} x_j \geq 1.$$

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

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

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

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

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

2.4.6 Méthode de M. Abbas & D. Chaabane [1]

Cette méthode dénommée "SEEVD", est une forme modifiée de la méthode de Gupta & Malhotra [39] où le test d'arrêt a été modifié afin de produire toutes les solutions efficaces du problème de programmation linéaire multi-objectifs en nombres entiers. Le principe de la méthode est que dans la première étape, une solution initiale est déterminée en optimisant un des critères, cette solution est le premier élément de la liste des solutions efficaces, puis dans la deuxième étape une séquence de coupes est appliquée après avoir exploré les arêtes incidentes à cette solution selon une direction bien choisie définie par un ensemble d'indices hors-base, ceci nous permet de générer une nouvelle solution efficace ajoutée à la liste précédente de solutions efficaces et de réduire le domaine de recherche jusqu'à ce qu'il devient vide. Dans ce cas, la procédure s'arrête avec une liste finale qui contient toutes les solutions efficaces du problème traité.

La technique étant présentée en détail dans l'algorithme suivant :

Algorithme

Étape 1. Résoudre le problème (P_1) . Soit x^1 la solution optimale et $(z_1^1, z_1^2, \dots, z_1^p)$ le vecteur critère correspondant.

Cas 1. Si $\Omega_1 = \{j \in N_1 \mid Z_{1,j}^1 - c_j^1 = 0\} = \emptyset$, alors la solution optimale est unique. Enregistrer le premier p -uplet non dominé $(z_1^1, z_1^2, \dots, z_1^p)$ dans $SND(P)$ et aller à l'étape 2.

Cas 2. Si $\Omega_1 \neq \emptyset$, alors la solution optimale peut ne pas être unique. Pour chaque $j \in \Omega_1$ calculer θ .

(a) Si pour tout $j \in \Omega_1$, on a $\theta < 1$, alors il n'y a pas de solutions alternatives à x^1 le long des arêtes E_j , $j \in \Omega_1$. $SND(P) = \{(z_1^1, z_1^2, \dots, z_1^p)\}$ et aller à l'étape 2.

(b) Sinon, tant qu'il existe au moins un $j \in \Omega_1$ tel que $\theta \geq 1$ faire :

- Explorer l'arête E_j
- Évaluer en chaque solution entière trouvée les p critères,
- Mettre à jour $SND(P)$.

Choisir arbitrairement un $j \in \Omega_1$, initialiser k à 1 et aller à l'étape (2.2).

Étape 2. $k = 1$;

2.1 Construire l'ensemble $\Psi_k = \{j \in N_k \mid Z_{k,j}^1 - c_j^1 > 0 \text{ et } Z_{k,j}^i - c_j^i > 0 \text{ pour au moins un } i, i = \overline{2, p}\}$.

Cas 1. Si $\Psi_k = \emptyset$, aller à l'étape (2.2) (la coupe devient une coupe de Dantzig : $\sum_{j_k \in N_k} x_j \geq 1$).

Cas 2. Sinon, mettre $\psi = \Psi_k$ et aller à (a).

- (a) Choisir un indice $j_k \in \psi$ et calculer le nombre θ .
- Si $\theta < 1$, il n'y a aucune solution entière sur l'arête E_{j_k} , faire $\psi = \psi \setminus \{j_k\}$. Si $\psi = \emptyset$, choisir un $j_k \in \Psi_k$ et aller à l'étape (2.2), sinon aller à (a).
 - Sinon, déterminer les solutions entières sur E_{j_k} , évaluer en chacune d'elles les r critères et mettre à jour l'ensemble $SND(P)$. Aller à l'étape (2.2).

2.2 Utiliser la coupe $\sum_{j \in N_k \setminus \{j_k\}} x_j \geq 1$ pour réduire le domaine de recherche et par application des méthodes duale du simplexe et les coupes de Gomory si nécessaire, on obtient x^{k+1} comme étant la solution optimale du problème augmenté. Mettre à jour $SND(P)$,

$k = k + 1$ et aller à l'étape (2.1).

Étape 3. La procédure prend fin quand l'opération pivot est impossible, le problème est devenu irréalisable dans la nouvelle région tronquée et la liste finale $SND(P)$ représente l'ensemble de toutes les solutions non dominées.

2.5 Conclusion

Dans le cadre de notre travail nous nous intéressons aux problèmes d'optimisation multi-objectifs en nombres entiers, Nous avons présenté d'une part la structure générale des problèmes ($MOILP$) et quelques résultats théoriques liés à ces problèmes. D'autre part, nous avons énuméré quelques méthodes de résolution de la programmation multi-objectifs discrète.

Le chapitre suivant est dédié à l'optimisation d'un critère sur l'ensemble des solutions efficaces du problème multi-objectifs. Dans certaines situations pratiques, les décideurs n'ont pas besoin de tout l'ensemble des solutions efficaces mais uniquement des solutions efficaces réalisant l'optimum d'un objectif différent des objectifs déjà fixés.

3

L'optimisation d'un critère linéaire sur un ensemble efficace

Sommaire

3.1	Introduction	52
3.2	Notations et résultats théoriques	54
3.3	Résolution du problème (P_E) dans le cas continu .	56
3.4	Résolution du problème (P_E) dans le cas discret .	60
3.5	Conclusion	69

3.1 Introduction

Dans certaines situations pratiques, l'énumération de toutes les solutions efficaces d'un problème multi-objectifs n'est pas toujours recommandée, car il peut s'avérer que cet ensemble soit de cardinalité importante, et il devient difficile pour le décideur de choisir le meilleur compromis en terme de ses préférences. Afin d'éviter ces situations, il semble très utile de considérer l'optimisation d'un critère qui exprime les préférences du décideur sur l'ensemble des solutions efficaces comme un moyen efficace pour guider le décideur vers une meilleure direction vers ses préférences. Ce problème appartient à la classe des problèmes d'optimisation globale, sa difficulté est principalement due à la structure de son ensemble réalisable (l'ensemble des

solutions efficaces) qui peut ne pas être convexe, ainsi que le type de la fonction à optimiser.

L'importance et la motivation de ce problème ont été discutées en détail dans la littérature (voir par exemple [44, 5, 64, 56]. En particulier, Benson [5] prouve que dans quelques problèmes de modélisation impliquant des objectifs multiples, les modèles d'optimisation sur l'ensemble des solutions efficaces sont plus réalistes et appropriés que les programmes linéaires multi-objectifs habituels (*MOLP*). En outre, la solution de ces problèmes d'optimisation, avec l'ensemble efficace défini implicitement, évite les difficultés informatiques d'énumérer tous les points extrêmes efficaces. L'exemple de planification de la production donné par Benson [58] illustre bien ces points. Durant les dernières soixante années, les chercheurs se sont intéressés à l'optimisation d'une fonction (linéaire ou non linéaire) sur l'ensemble des solutions efficaces d'un problème de programmation linéaire multi-objectifs (*MOLP*). Ce problème a été étudié pour la première fois par Philip en 1972, qui a proposé un algorithme basé sur le déplacement sur les sommets efficaces adjacents dans le cas où le critère à optimiser est linéaire. Depuis, notamment dans le cas continu, plusieurs chercheurs, citons à titre d'exemple, Benson [6],[7],[8], Yamamoto [78], Eker et Song [25], Sayin [60], motivés par de nombreuses applications [5],[33], ont suivi cette voie.

Pour le cas discret, la première tentative d'étudier le problème a été réalisée par N.C. Nguyen (1992) [56] où seulement une borne supérieure de la valeur optimale du critère principal a été donnée. La première méthode proposée pour l'optimisation sur l'ensemble efficace d'un problème (*MOILP*) en évitant l'énumération explicite de tous les points efficaces est celle de M. Abbas & D. Chaabane [2], où différents types de coupes sont imposées de telle manière que l'amélioration de la valeur optimale du critère principal à chaque itération soit garantie, suivie par l'algorithme de J. M. Jorge [46] qui est basé sur l'analyse d'un ordre approprié des problèmes linéaires en nombres entiers pour éliminer successivement les solutions moins bonnes sur le critère principal. Récemment, en 2012, Chaabane et all.[16] proposent une méthode de résolution dans l'espace des critères dans laquelle la norme pondérée de Tchebychev est optimisée progressivement pour rapprocher du point idéal et améliorer la valeur du critère principal. D'autre part, le domaine de faisabilité est réduit en éliminant les solutions réalisables moins bonnes dans le sens de Pareto.

Le problème d'optimisation d'un critère sur l'ensemble des solutions efficaces d'un problème (*MOLP*) peut être formulé d'une manière générale par :

$$(P_E) \begin{cases} \max & \phi(x) \\ \text{s.c.} & x \in E \end{cases} \quad (3.1)$$

Où E représente l'ensemble des solutions efficaces du problème multi-objectifs suivant

$$(MOLP) \begin{cases} \text{“max” } & Cx \\ \text{s.c.} & x \in S \end{cases} \quad (3.2)$$

Où $S = \{x \in \mathbb{R}^n | Ax \leq b; x \geq 0\}$; $A \in \mathbb{R}^{m \times n}$; $x \in \mathbb{R}^n, b \in \mathbb{R}^m, C \in \mathbb{R}^{p \times n}$.

Le problème relaxé de (3.1) est défini par :

$$(P_R) \begin{cases} \max & \phi(x) \\ \text{s.c.} & x \in S \end{cases} \quad (3.3)$$

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

3.2 Notations et résultats théoriques

Dans cette section, nous présentons les notations nécessaires à la description des algorithmes présentés dans ce chapitre.

- \mathbb{R}^p l'ensemble des vecteurs colonnes réels d'ordre p ;
- $\mathbb{R}_+^p = \{x \in \mathbb{R}^p | x \geq 0\}$; $\mathbb{R}_{++}^p = \{x \in \mathbb{R}^p | x > 0\}$;
- \mathbb{R}_p désigne l'ensemble des vecteurs lignes réels d'ordre p .
- $\mathbb{R}_{p+} = \{x \in \mathbb{R}_p | x \geq 0\}$; $\mathbb{R}_{p++} = \{x \in \mathbb{R}_p | x > 0\}$;
- \mathbb{R}_-^p et \mathbb{R}_{--}^p sont définies de la même manière que \mathbb{R}_+^p et \mathbb{R}_{++}^p , e désigne un vecteur dont toutes les composantes sont égales à 1.
- Y_E est l'ensemble des solutions non dominées, Y_w est l'ensemble des solutions faiblement non dominées.

Définition 3.1

- a. $Y = \{y | y \in \mathbb{R}^p, y = Cx \text{ pour tout } x \in S\}$ est appelé l'ensemble des résultats (outcome set).
- b. $Y^{\leq} = \{y | y \in \mathbb{R}^p, y \leq Cx \text{ pour tout } x \in S\}$ est appelé l'ensemble des résultats inférieurs (lower outcome set).
- c. $Y^{<} = \{y | y \in \mathbb{R}^p, y < Cx \text{ pour tout } x \in S\}$ est appelé l'ensemble des résultats strictement inférieurs (strictly lower outcome set).

Définition 3.2

- Un point $y \in Y$ est dit solution non dominée du problème (MOLP) si $Y \cap (y + \mathbb{R}_+^p) = \{y\}$.

- Un point $y \in Y$ est dit faiblement non dominé si $Y \cap (y + \mathbb{R}_{++}^p) = \emptyset$.
- L'ensemble de toutes les solutions non dominées sera noté par Y_E .

Lemme 3.1 $E = \{x \in S / Cx \in Y_E\}$.

Définition 3.3 Pour $\lambda \in \mathbb{R}_{p++}$ et $x \in S$ la fonction

$$g_\lambda(x) = \max \{ \lambda Cx' \mid x' \in S; Cx' \geq Cx \} - \lambda Cx$$

$g_\lambda(x)$ est dite fonction lacune ("gap function"), $x \in E$ si et seulement si $g_\lambda(x) = 0$.

Quand $\lambda = e = (1, \dots, 1) \in \mathbb{R}_p$ alors g_λ est notée par g .

Dans le théorème suivant nous citons quelques caractérisations de l'ensemble des solutions efficaces E du problème (MOLP) (Voir [78],[66]).

Théorème 3.1 [78] *L'ensemble des solutions efficaces du problème (MOLP) est tel que*

$$\begin{aligned} E &= \{x \mid x \in S, \exists \lambda \in \mathbb{R}_{p++} \text{ tel que } \lambda Cx \geq \lambda Cx' \forall x' \in S\}. \\ &= \{x \mid x \in S, \nexists x' \in \mathbb{R}^n \text{ tel que } Cx' \geq 0, Cx' \neq 0, Ax' = 0, x'_i \geq 0 \text{ pour } i \text{ avec } x_i = 0\}. \\ &= \{x \mid x \in S, \exists (\lambda, \mu, \nu) \in \mathbb{R}_{++}^n \times \mathbb{R}^m \times \mathbb{R}_+^n \text{ tel que } \lambda C - \mu A + \nu = 0; \nu x = 0\}. \\ &= \{x \mid x \in S, \exists (\lambda, \mu) \in \mathbb{R}_{++}^n \times \mathbb{R}^m \text{ tel que } \lambda C - \mu A \leq 0, \lambda Cx - \mu b = 0\}. \\ &= \{x \mid x \in S; g_\lambda(x) = 0\}. \end{aligned}$$

Définition 3.4 Soit S un polyèdre de \mathbb{R}^n . $F \subset S$ est une face de S si :

- a. F est convexe.
- b. Pour tout $x \neq y$ de S tels que le segment ouvert $]x, y[$ rencontre F alors x et y sont dans F .

Remarque 3 Si les points x et x' se trouvent sur la même face de S alors $x \in E$ si et seulement si $x' \in E$.

Dans le cas continu, les méthodes utilisées pour chercher l'optimum sur l'ensemble des solutions efficaces E , utilisent la propriété de connexité de E . Celle-ci est établie par le théorème suivant :

Théorème 3.2 [66] *L'ensemble E des solutions efficaces d'un problème linéaire multi-objectifs continu est convexe si deux points extrêmes quelconques dans E sont reliés par un chemin formé d'arêtes efficaces (une arête efficace est une arête de S contenue dans E).*

3.3 Résolution du problème (P_E) dans le cas continu

Les algorithmes existants pour résoudre ce problème peuvent être classés en plusieurs groupes selon leur principe de fonctionnement, citons :

- les algorithmes de recherche de sommet adjacent
- les algorithmes de recherche de sommets non adjacents
- les algorithmes basés sur la méthode de séparation et évaluation
- les algorithmes basés sur la méthode de relaxation Lagrangienne
- les algorithmes basés sur la méthode duale et de bisection

Les algorithmes proposés par Philip [57], Ecker et Song [25] et Fülöp [33] concernent le cas où $\phi(x)$ est linéaire. Par contre, ceux proposés par Bolintineanu [10] et Yamamoto, Y. [78] optimisent une fonction quasi-convexe sur E . Ces auteurs se sont basés principalement sur deux techniques :

- Se déplacer d'un sommet efficace à un sommet voisin efficace avec une plus grande valeur de la fonction objectif par l'intermédiaire d'une arête efficace ;
- Isoler par des coupes la partie de S où la fonction objectif linéaire $\phi(x)$ prend la plus grande valeur possible.

3.3.1 Méthode de Yamamoto [78]

Cette méthode est basée sur la recherche des sommets adjacents où la fonction ϕ est quasi-convexe, nous rappelons la définition d'une fonction quasi-convexe dans la définition suivante :

Définition 3.5 Une fonction f est quasi-convexe sur S si :

$$\forall x, y \in S \text{ et } \forall \lambda \in [0, 1] \text{ on a } f(\lambda x + (1 - \lambda)y) \leq \max\{f(x), f(y)\};$$

f est quasi-convexe sur S est ainsi équivalente à :

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

C'est une propriété assez large, il y a "beaucoup" de fonctions quasi-convexes en économie et la plupart des fonctions d'utilité "classiques" le sont dans ce domaine.

Soit S_v l'ensemble des points extrêmes du polyèdre S . Pour $x, x' \in S_v$, $[x, x']$ représente l'arête reliant x et x' . Pour $x \in S_v \cap E$, on note $N_E(x)$ l'ensemble des sommets efficaces liés à x par une arête efficace tel que :

$$N_E(x) = \{x' | x' \in S_v \cap E; [x, x'] \subset E\}.$$

Lemme 3.2 Soit $x \in S_v \cap E$ et supposons que $\{x' | x' \in N_E(x), \phi(x') > \phi(x)\} = \emptyset$ alors x est un maximum local pour le problème (3.1).

Ce lemme caractérise la solution optimale locale du problème (3.1) quand elle existe. Il est utilisé dans l'algorithme de Yamamoto pour brancher vers la deuxième boucle (Boucle sur k).

Algorithme de Yamamoto

Soit $H = \{x \in \mathbb{R}^n | ax = \alpha\}$ l'hyperplan déterminé par les semi-plans et $H_+ = \{x \in \mathbb{R}^n | ax \geq \alpha\}$, $H_- = \{x \in \mathbb{R}^n | ax \leq \alpha\}$; H_{++} et H_{--} sont leurs intérieurs respectivement.

Algorithm 2: Algorithme de Yoshitsugu Yamamoto ([78])

Phase Initiale Poser $k = \ell = 0$, $s^0 = S$, trouver $x^0 \in S_v \cap E$.
SI $N_E(x^0) = \emptyset$ **ALORS**
 | x^0 est une solution optimale du problème (P_E)
SINON
 | aller à l'étape suivante.
Étape1(Boucle (ℓ))
SI $\{x | x \in N_E(x^\ell); \phi(x) > \phi(x^\ell)\} = \emptyset$ **ALORS**
 | choisir $x^{\ell+1}$ de cet ensemble.
 | Faire $\ell = \ell + 1$ et aller à l'étape suivante.
SINON
 | Soit $S^\ell = \{x | \phi(x) \leq \phi(x^\ell)\}$ et aller à l'étape 2.
Étape2 Boucle (k)
 Trouver $v^k \in \arg \max \{\phi(x) | x \in S^k\}$.
SI $\phi(x^\ell) \geq \phi(v^k) - \epsilon$, pour $\epsilon > 0$ **ALORS**
 | arrêter avec x^ℓ comme une ϵ - approximation de la solution optimale du problème (P_E)
SINON
 | Trouver un hyperplan support H^k de S^ℓ tel que $S^\ell \subseteq H_+^k$ et $v^k \in H_{--}^k$.
 | **SI** il existe une arête efficace $[u', u'']$ telle que $[u', u''] \cap H^k \neq \emptyset$ et $\max \{\phi(u'), \phi(u'')\} > \phi(x^\ell)$ **ALORS**
 | Soit $x^{\ell+1}$ celui des u' et u'' donnant la valeur de la fonction objectif la plus grande.
 | Faire $\ell = \ell + 1$ et aller à l'Étape 1,(boucle (ℓ))
 | **SINON**
 | Faire $S^{k+1} = S^k \cap H_+^k$, $k = k + 1$ et aller à (boucle(k)).
Arrêt: L'algorithme génère une séquence de sommets efficaces x^0, x^1, \dots et les polytopes S^0, S^1, \dots tels que $\phi(x^0) < \phi(x^1) < \dots$ et $S = S^0 \supseteq S^1 \supseteq \dots$

3.3.2 Méthode de Benson [7]

Cette méthode est basée sur la recherche de sommet non adjacent où la fonction objectif $\phi(x) = dx$ est supposée linéaire ainsi que l'existence de $(k + 1)$ solutions efficaces $x^0, x^1, \dots, x^k \in E$ et soit $\alpha^k = \max\{dx^j | j \in \{0, \dots, k\}\}$; Le problème (P^k) suivant joue un rôle fondamental dans l'algorithme pour la recherche d'un point $(x, \lambda) \in \mathbb{R}^n \times \mathbb{R}_p$ satisfaisant :

$$(P^k) \begin{cases} \lambda Cx \geq \lambda Cx^j \text{ pour } j \in \{0, \dots, k\}; \\ x \in X; \\ \lambda \in \Lambda; \\ dx > \alpha^k. \end{cases} \quad (3.4)$$

Où $\Lambda = \{\lambda_i : \sum_{i=1}^k \lambda_i = 1 \text{ et } \lambda_i \geq 0, \forall i = \{1, \dots, k\}\}$.

Remarque 4 Si $(\bar{x}, \bar{\lambda}) \in X \times \Lambda$ vérifie les contraintes $\lambda Cx \geq \lambda Cx^j$ pour $j \in \{0, \dots, k\}$ du problème (P^k) alors \bar{x} est une solution efficace de l'enveloppe convexe de x^0, x^1, \dots, x^k et \bar{x} lui-même.

Lemme 3.3 Supposons que $x^0, x^1, \dots, x^k \in E$ et que le problème (P^k) n'admet pas de solution alors $x^* \in \operatorname{argmax}\{dx^j | j \in \{0, \dots, k\}\}$ est une solution optimale du problème initial (P_E)

Remarque 5

- a. Pour montrer que \bar{x}^{k+1} est un sommet de S , il suffit de vérifier l'indépendance linéaire des colonnes de A correspondant aux composantes positives de \bar{x}^{k+1} ;
- b. Une manière de déterminer une face F de S est de considérer par exemple $F = \{x \in S | x_j = 0 \text{ pour } j \text{ avec } \bar{x}_j^{k+1} = 0\}$;
- c. L'algorithme génère une suite de sommets distincts de E et fournit une solution optimale de (P_E) en un nombre fini d'itérations.

Algorithm 3: Algorithme de Benson ([7])

Phase Initiale

Trouver un sommet efficace x^0 , poser $k = 0$ et aller à l'étape (k)

Étape k

(k_1) : Trouver une solution $(x, \lambda) \in \mathbb{R}^n \times \mathbb{R}_p$ du problème (P^k)

SI (P^k) n'admet pas de solution **ALORS**

| $x^* \in \operatorname{argmax}\{dx^j | j \in \{0, \dots, k\}\}$ est une solution optimale de (P_E)

SINON

| soit $(\bar{x}^{k+1}, \bar{\lambda}^{k+1})$ la solution trouvée.

(k_2) : Résoudre le programme linéaire suivant

$$(Test^k) : \{\max eCx | Cx \geq C\bar{x}^{k+1}, x \in S\}$$

Soit \hat{x} une solution de ($Test^k$)

SI $eC\hat{x} = eC\bar{x}^{k+1}$ **ALORS**

| aller à (k_3)

SINON

| aller à (k_5)

(k_3) : **SI** \bar{x}^{k+1} est un sommet de S **ALORS**

| poser $x^{k+1} = \bar{x}^{k+1}$; $k = k + 1$ et aller à l'étape k

SINON

| aller à (k_4)

(k_4) : Soit F une face de S dont son intérieur contient \bar{x}^{k+1} ; résoudre le programme linéaire ($Face^k$) suivant

$$(Face^k) : \{\max dx | x \in F\}$$

dont la solution est un point extrême \bar{x}^{k+1} , poser $k = k + 1$ et aller à l'étape (k).

(k_5) : Résoudre le programme $SC(\bar{\lambda}^{k+1})$ suivant dont la solution optimale est x^{k+1}

$$SC(\bar{\lambda}^{k+1}) \{\max \bar{\lambda}^{k+1} Cx | x \in S\}$$

Poser $k = k + 1$ aller à l'étape(k).

3.4 Résolution du problème (P_E) dans le cas discret

Dans cette section nous nous intéressons à l'optimisation d'une fonction linéaire $\phi(x)$ sur l'ensemble des solutions efficaces d'un problème $(MOILP)$ ("Multiple Objective Integer Linear Programming") qui se formule par :

$$(MOILP) \begin{cases} \text{“max” } Cx \\ \text{s.c. } x \in D \end{cases} \quad (3.5)$$

Où : $D = S \cap \mathbb{Z}^n$, \mathbb{Z} étant l'ensemble des nombres entiers relatifs. S est borné et convexe $S = \{x \in \mathbb{R}^n | Ax \leq b, x \geq 0\}$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, C est une matrice de dimension $p \times n$ d'éléments réels, et ses vecteurs lignes $c^k \in \mathbb{R}^n$, $k = 1, \dots, p$.

Le problème principal est donné par :

$$(P_{IE}) \begin{cases} \max \phi(x) = dx \\ \text{s.c. } x \in IE \end{cases} \quad (3.6)$$

IE est l'ensemble de toutes les solutions efficaces entières du problème (3.5), d est un vecteur ligne de dimension n qui a pour $j^{\text{ème}}$ composante le nombre entier d_j .

Soit le problème relaxé :

$$(P_{IR}) \begin{cases} \max \phi(x) = dx \\ \text{s.c. } x \in D \end{cases} \quad (3.7)$$

Le problème (P_{IE}) est un problème d'aide à la décision. Ce problème est lié à deux classes de problèmes : les problèmes de l'optimisation linéaire multi-objectifs en nombres entiers dont l'étude est en plein essor et les problèmes de l'optimisation unicritère discrète qui constitue une extension assez large de la programmation linéaire. Compte tenu de cette caractérisation, Il n'est pas surprenant que peu de travaux aient été réalisés. En fait, dans la littérature, on ne trouve que peu d'articles dédiés au problème (P_{IE}) , citons [56, 2, 46, 16, 17]. Tous ces articles traitent le cas (linéaire-linéaire) c'est-à-dire, optimiser un critère linéaire sur un ensemble efficace d'un problème multi-objectifs linéaire.

Naïvement, on peut résoudre le problème (P_{IE}) en formant la liste de toutes les solutions efficaces IE et en optimisant $\phi(x) = dx$ sur cette liste. Cette approche n'est pas appropriée pour des raisons d'ordre pratique liées à la difficulté de déterminer IE qui peut être un ensemble de taille exponentielle en le nombre de variables. Les sources de motivation pour étudier

de tel problème vise à éviter cependant l'énumération explicite de toutes les solutions efficaces. On peut se demander pourquoi optimiser sur l'ensemble des solutions Pareto-optimales alors qu'il suffit d'optimiser sur la frontière efficace, l'ensemble des points non dominés dans l'espace des critères. Ceci est possible si la fonction à optimiser peut être exprimée en fonction des critères initiaux (variables dans l'espace des critères), mais la difficulté s'installe lorsque cette fonction est exprimée en fonction des variables de décision.

Le problème en variables continues est difficile à traiter ; il devient plus difficile en variables discrètes. Les difficultés particulières rencontrées dans sa résolution sont dues à :

- L'optimisation sur un ensemble qui n'est pas connu a priori ;
- Le cadre non convexe et discret du domaine de décision D ;
- Sur le front de Pareto non convexe deux types de solutions peuvent être différenciées : les solutions supportées et les solutions non supportées. Nous présentons dans cette section deux méthodes de résolution, la première est celle de J. M. Jorge (2008) car notre contribution dans le chapitre suivant est inspirée de cet algorithme. Nous détaillerons par la suite la méthode la plus récente pour résoudre le problème, celle de D .Chaabane et al.(2012)[16].

3.4.1 Résultats fondamentaux

Théorème 3.3 (Isermann 1974)[45] *Soit x^* un point arbitraire réalisable de S , x^* est une solution efficace du problème (MOLP) si et seulement si la valeur optimale de la fonction objectif Θ^* est nulle dans le programme linéaire suivant :*

$$P(x^*) \begin{cases} \max \Theta = \sum_{i=1}^p \psi_i \\ \text{s.c.} & c^i x - \psi_i = c^i x^* \\ & x \in S \\ & \psi_i \in \mathbb{R}^+, \forall i = 1, \dots, p \end{cases} \quad (3.8)$$

Le problème $P(x^*)$ est souvent utilisé pour tester l'efficacité d'une solution réalisable donnée. Le théorème suivant, montre que $P(x^*)$ peut être aussi utilisé pour générer une solution efficace même si la solution x^* ne l'est pas.

Théorème 3.4 (Ecker et Kouada 1978)[26] *Si $P(x^*)$ possède une valeur maximale finie non nulle atteinte en un point réalisable \tilde{x} alors \tilde{x} est efficace.*

Théorème 3.5 [26] *Si $P(x^*)$ n'admet pas une solution optimale finie, alors l'ensemble, E , des solutions efficaces du problème (MOLP) est vide.*

Ces trois théorèmes ont été énoncés dans le cas continu, ils restent valables pour le cas où les variables de décision sont discrètes. Dans les algorithmes qui seront présentés dans le reste du document, on est appelé à tester l'efficacité d'une solution réalisable x^* par la résolution du problème $P(x^*)$, appelé souvent "**Test d'efficacité**" tel qu'il est décrit dans le théorème 3.3 dont les paramètres restent inchangés, sauf le domaine de décision qu'il faut remplacer par le domaine discret D .

Théorème 3.6 [15] *Soit x^* une solution réalisable de D , $x^* \in IE$ si et seulement si la fonction objectif Θ est nulle dans le problème de programmation linéaire mixte suivant :*

$$(P(x^*)) \left\{ \begin{array}{l} \max \Theta = \sum_{i=1}^p \psi_i \\ \text{s.c. } c^i x - \psi_i = c^i x^* \\ x \in D \\ \psi_i \in \mathbb{R}^+, \forall i = 1, \dots, p \end{array} \right. \quad (3.9)$$

Preuve

(\Rightarrow) Par l'absurde, soit x^* est solution efficace de $(MOILP)$, on suppose que $\Theta^* \neq 0$, alors $\exists i \in \{1, 2, \dots, p\}$ tel que $\psi_i > 0$ et $\exists x \in D$ tel que $c^i x > c^i x^*$ donc Cx domine Cx^* ce qui est contradictoire avec l'hypothèse que x^* est efficace.

(\Leftarrow) Soit maintenant $\Theta^* = 0$, on suppose que x^* n'est pas efficace, alors il existe $x \in D$ tel que $Cx \geq Cx^*$ et $Cx \neq Cx^*$ alors $\exists i \in \{1, 2, \dots, p\}$ tel que $c^i x - c^i x^* > 0$ donc $\psi_i > 0$, ce qui contredit le fait $\Theta^* = 0$. □

3.4.2 Méthode de Jesus [46]

L'algorithme proposé consiste à résoudre le problème (P_{IE}) dans l'espace des critères et produire une solution optimale globale sans devoir énumérer l'ensemble de toutes les solutions efficaces IE , la procédure commence à résoudre le problème relaxé (P_{IR}) , la solution optimale est testée pour l'efficacité. Évidemment, seulement dans un nombre réduit de cas spéciaux la solution optimale de (P_{IR}) fournit une solution optimale de (P_{IE}) . Ainsi, si ce n'était pas le cas, une nouvelle solution efficace, soit \hat{x}^1 , qui domine la solution optimale de (P_{IR}) est alors générée par le test d'efficacité.

Dans l'espace des critères, plusieurs solutions efficaces peuvent avoir le même vecteur critères, pour celà le problème (T_i) est résolu pour optimiser le critère principal sur toutes les solutions équivalentes à \hat{x}^1 .

$$(T_1) : \max \left\{ dx \mid Cx = C\hat{x}^1, x \in D \right\}, \quad (3.10)$$

Ensuite, dans chaque itération, le problème (R_l) est résolu pour optimiser le critère principal sur le domaine restreint par des contraintes en nombres entiers qui sont incluses progressivement pour éliminer les solutions dominées par la solution efficace courante, afin de fournir une solution non dominée par les solutions détectées antérieurement jusqu'à ce qu'une solution optimale soit finalement trouvée.

$$(R_l) : \max \left\{ dx \mid x \in D - \cup_{s=1}^l D_s \right\}, \quad (3.11)$$

où $D_s = \{x \in \mathbb{Z}^n \mid Cx \leq C\tilde{x}^s\}$ avec $\tilde{x}^1, \tilde{x}^2, \dots, \tilde{x}^l$ sont des solutions efficaces obtenues jusqu'à l'itération l .

L'algorithme de la Méthode

Algorithm 4: Algorithme de Jesus([46])

Entrées:

- ↓ $A_{(m \times n)}$: matrice des contraintes.
- ↓ $b_{(m \times 1)}$: vecteur second membre.
- ↓ $d_{(1 \times n)}$: vecteur du critère principal.
- ↑ $c_{(p \times n)}$: matrice des critères.

Sorties:

- ↑ x_{opt} : Solution optimale du problème (P_{IE}).
- ↑ ϕ_{opt} : valeur optimale du critère principal ϕ .

Initialisation

Poser $\phi_{inf} = -\infty$, $\phi_{sup} = +\infty$, $l = 1$, et résoudre le problème relaxé (P_{IR})

SI (P_{IR}) *n'est pas réalisable* **ALORS**

 | Terminer, le problème (P_{IE}) n'a pas de solution.

SINON

 | soit x^l solution optimale de (P_{IR}).

Étape 1 : Tester l'efficacité de x^l

SI x^l *est efficace* **ALORS**

 | l'algorithme prend fin et $x_{opt} = x^l$, $\phi_{opt} = dx^l$

SINON

 | poser $\phi_{sup} = dx^l$ et aller à l'étape 2.

Étape 2 : Soit $\hat{x}^l \in IE$ une solution optimale du test d'efficacité, résoudre le problème (T_l), soit \tilde{x}^l une solution optimale

SI $d\tilde{x}^l > \phi_{inf}$ **ALORS**

 | poser $\phi_{inf} = d\tilde{x}^l$ et $x_{opt} = \tilde{x}^l$. Aller à l'étape 3.

SINON

 | $\phi_{inf} = \phi_{sup}$. Terminer, x_{opt} est la solution optimale de (P_{IE}).

Étape 3 : Résoudre le problème (R_l)

SI (R_l) *n'est pas réalisable* **ALORS**

 | Terminer, x_{opt} est une solution optimale de (P_{IE}).

SINON

 | soit x^{l+1} solution optimale de (R_l)

SI $dx^{l+1} \leq \phi_{inf}$ **ALORS**

 | Terminer, x_{opt} est une solution optimale de (P_{IE}).

SINON

 | poser $l = l + 1$ et aller à l'étape 1.

3.4.3 Méthode de Chaabane et al.[16]

La méthode proposée consiste à choisir une solution non dominée la plus proche possible du point idéal améliorant le critère principal d'une itération à une autre. Une fois une solution efficace est mise à jour, le domaine de faisabilité est réduit en éliminant les solutions réalisables moins bonnes dans le sens de Pareto. La norme pondérée augmentée de Tchebychev est utilisée pour minimiser la distance entre la solution réalisable courante et le point idéal. Le résultat de Bowman (3.14) est appliqué afin d'éviter une minimisation d'une fonction non linéaire (la norme de Tchebychev) par la résolution d'un programme linéaire équivalent qui semble comme une alternative au test d'efficacité, sa résolution nous assure l'obtention d'une solution efficace à partir d'une solution réalisable.

La théorie de la norme de Tchebychev

Dans cette section nous rappelons brièvement quelques notions et résultats que les auteurs ont utilisés pour développer leur méthode.

On considère l'ensemble Δ , l'ensemble des poids défini par :

$$\Delta = \{\beta \in \mathbb{R}^p; 0 \leq \beta_i \leq 1, \sum_{i=1}^p \beta_i = 1\}.$$

La norme pondérée de Tchebychev dans \mathbb{R}^p ce n'est que la norme infinie (l_∞) définie par :

$$\|z_1, \dots, z_p\|_\infty^\beta = \max_{i=1, \dots, p} \{\beta_i |z_i|\}.$$

Définition 3.6 La norme pondérée de Tchebychev d'un vecteur z peut être calculée en considérant le point idéal z^* comme origine, elle est donnée par :

$$\|z^* - z\|_\infty^\beta = \max_{i=1, \dots, p} \{\beta_i |z_i^* - z_i|\}.$$

Définition 3.7 La norme augmentée de Tchebychev est donnée par l'expression suivante :

$$\|z^* - z\|_\infty^\beta = \max_{i=1, \dots, p} \{\beta_i (|z_i^* - z_i|) + \rho \sum_{i=1}^p (|z_i^* - z_i|)\},$$

où : ρ est un réel positif très petit. Cette augmentation de la norme sert à éviter le problème de solutions faiblement efficaces.

La norme de Tchebychev est souvent utilisée pour calculer la distance pondérée entre n'importe quel point de Z est le point d'aspiration z^* . En effet, il s'agit de trouver les points réalisables dont la distance est la plus petite du point idéal z^* , pour se faire on doit résoudre les problèmes suivants :

$$\min_{z \in Z} \{\|z^* - z\|_\infty^\beta\}. \quad (3.12)$$

Le problème équivalent à (3.12) appelé programme de la norme pondérée de Tchebychev "weighted Tchebychev program"(WTP), noté $P(\beta)$, proposé par Bowman [11], est donné par :

$$(P(\beta)) \begin{cases} \min \omega \\ \omega \geq \beta_i(z_i^* - z_i); \quad 1 \leq i \leq p \\ x \in D, \omega \geq 0, \end{cases} \quad (3.13)$$

où :

$$z_i = c^i x, \text{ et } \beta_i = \frac{1}{z_i^* - z_i} \left[\sum_{i=1}^p \frac{1}{z_i^* - z_i} \right]^{-1}, \quad \forall 1 \leq i \leq p.$$

Pour ne pas diviser par zéro un point utopique "utopian point" est utilisé comme un point d'aspiration et défini par l'expression suivante :

$$z^{**} = z^* + 1.$$

Ainsi, le problème $(P(\beta))$ est défini par :

$$(P(\beta)) \begin{cases} \min \omega \\ \omega \geq \beta_i(z_i^{**} - z_i); \quad 1 \leq i \leq p \\ x \in D, \omega \geq 0, \end{cases} \quad (3.14)$$

et

$$\beta_i = \frac{1}{z_i^{**} - z_i} \left[\sum_{i=1}^p \frac{1}{z_i^{**} - z_i} \right]^{-1}, \quad \forall 1 \leq i \leq p.$$

Les théorèmes suivants présentent quelque conditions pour caractériser une solution non dominée.

Théorème 3.7 [67] *Si Z est fini et soit $M = \{z \in Z | (x, z, \omega) \text{ une solution optimale de } P(\beta) \text{ pour un certain vecteur } \beta \in \Delta\}$, alors il existe $\bar{z} \in M$ tel que \bar{z} est un point Pareto.*

Théorème 3.8 [11] *Si x^* est une solution efficace alors x^* est une solution optimale de $P(\beta)$ pour un certain $\beta \in \Delta$,*

Théorème 3.9 [11] *Si l'ensemble des solutions efficaces du problème (3.5) est fortement efficace alors toute solution optimale du programme (3.14) est une solution efficace de problème (3.5) pour un certain vecteur $\beta \in \Delta$.*

Le programme linéaire sous la norme augmentée de Tchebychev est défini par :

$$(P_\rho(\beta)) \begin{cases} \min \omega + \rho \sum_{i=1}^p (|z_i^{**} - z_i|) \\ \omega \geq \beta_i(z_i^{**} - z_i); \quad 1 \leq i \leq p \\ x \in D, \omega \geq 0. \end{cases} \quad (3.15)$$

Description de la Méthode

La méthode proposée consiste à trouver une solution efficace optimale du problème principal, sans passer explicitement par toutes les solutions efficaces du problème (3.5). L'algorithme s'articule autour de deux parties :

En première partie, un vecteur non dominé est caractérisé par la résolution du problème (3.15) pour une valeur de $\rho > 0$ suffisamment petite. En deuxième partie, la réduction progressive du domaine d'admissibilité est effectuée en ajoutant des contraintes issues de la définition de solution "Pareto Optimale".

Initialement, on détermine le point utopique z^{**} , puis on résout le problème relaxé (3.7). Une solution optimale est obtenue et une borne supérieure du critère principale est mise à jour pour des valeurs très petites de ρ . On résout le problème (3.15) afin de trouver un vecteur non dominé \bar{z} le plus proche possible du vecteur utopique z^{**} .

Comme dans l'espace des critères plusieurs solutions efficaces peuvent avoir le même vecteur \bar{z} (Solution efficace équivalente), donc un autre programme est résolu pour choisir la meilleure solution équivalente sur le critère principal. Une nouvelle solution efficace par conséquent est générée et ajoutée à la liste courante, le critère principal ϕ est évalué et la région d'admissibilité courante est réduite par les contraintes (3.16).

L'algorithme prend fin quand la région d'admissibilité courante devient vide ou la borne inférieure du critère principal coïncide avec la borne supérieure.

Le domaine réduit $D - \bigcup_{s=1}^k D_s$ est :

$$D - \bigcup_{s=1}^k D_s = \left\{ \begin{array}{l} c^i x \geq (c^i x^s + 1)y_i^s + M_i(1 - y_i^s) \\ \sum_{i=1}^p y_i^s = 1 \\ y_i^s \in \{0, 1\}, i = 1, \dots, p, s = 1, \dots, k \\ x \in D. \end{array} \right\} \quad (3.16)$$

où M_i est la borne inférieure du $i^{\text{ème}}$ critère, $\{x^s; s = 1, \dots, k\}$ sont les solutions obtenues à la $k^{\text{ème}}$ itération.

L'algorithme de la Méthode

Algorithm 5: L'algorithme de Chaabane et al.[16]

Entrées:

- ↓ $A_{(m \times n)}$: matrice des contraintes.
- ↓ $b_{(m \times 1)}$: vecteur second membre.
- ↓ $d_{(1 \times n)}$: vecteur du critère principal.
- ↑ $c_{(p \times n)}$: matrice des critères.

Sorties:

- ↑ x_{opt} : Solution optimale du problème (P_{IE})
- ↑ ϕ_{opt} : valeur optimale du critère principal ϕ

Initialisation

pour $i = 1, \dots, p$ **faire**

- résoudre $z_i^* = \max\{c^i x; x \in D\}; z^{**} = z^* + 1;$
- $M_i = \min\{c^i x; x \in D\}$

$\phi_{sup} \leftarrow +\infty; \phi_{inf} \leftarrow -\infty; E_1 \leftarrow \emptyset; k = 1; \bar{D} \leftarrow D; fin \leftarrow faux.$

tant que $fin \leftarrow faux$ **faire**

résoudre le problème (P_{IR}^k) = $\max\{dx, x \in \bar{D}\}$

SI (P_{IR}^k) est irréalisable où $\phi_{inf} \geq \phi_{sup}$ **ALORS**

- x_{opt} est une solution optimale de (P_{IE}); $fin \leftarrow vrai;$
- l'algorithme prend fin

SINON

Soit x^k solution optimale du problème (P_{IR}^k);

Soit $\phi_{sup} = dx^k$; calculer le vecteur poids β^k de $z^k = Cx^k$;

Soit (\hat{x}^k, \hat{z}^k) une solution optimale du problème $P_\rho(\beta^k)$

SI $d\hat{x} = \phi_{sup}$ **ALORS**

- $x_{opt} \leftarrow \hat{x}^k; \phi_{opt} \leftarrow \phi_{sup}; fin \leftarrow vrai;$ l'algorithme prend fin

SINON

résoudre le problème $Q(\hat{z}^k) \equiv \max\{dx \mid x \in D; Cx = \hat{z}^k\};$

Soit \bar{x}^k une solution optimale du problème $Q(\hat{z}^k)$

SI $d\bar{x}^k > \phi_{inf}$ **ALORS**

- $x_{opt} \leftarrow \bar{x}^k; \phi_{inf} = d\bar{x}^k; \phi_{opt} \leftarrow \phi_{inf};$ soit
- $E_{k+1} \leftarrow E_k \cup \{\bar{x}^k\},$
- $k \leftarrow k + 1$ et $\bar{D} \leftarrow D \setminus \cup_{s=1}^{k-1} D_s$

SINON

SI $\phi_{inf} \geq \phi_{sup}$ **ALORS**

- x_{opt} une solution optimale de (P_{IE}) et ϕ_{opt} une valeur optimale de ϕ ; $fin \leftarrow vrai;$ l'algorithme prend fin

SINON

- $x_{opt} \leftarrow \bar{x}^k; \phi_{inf} = d\bar{x}^k; \phi_{opt} \leftarrow \phi_{inf};$ soit
- $E_{k+1} \leftarrow E_k \cup \{\bar{x}^k\},$
- $k \leftarrow k + 1$ et $\bar{D} \leftarrow D \setminus \cup_{s=1}^{k-1} D_s$

3.5 Conclusion

Dans ce chapitre, nous avons présenté quelques méthodes de résolution du problème (P_E) dans le cas continu et le cas discret. Le problème est très difficile à résoudre vu la non convexité de son domaine réalisable. Mathématiquement, ce problème est classé parmi les problèmes difficiles de l'optimisation globale ou non convexe.

Dans le chapitre suivant, nous abordons la programmation linéaire fractionnaire, particulièrement la programmation hyperbolique en nombres entiers qui fait l'objet de nos préoccupations dans cette étude.

4

La programmation linéaire fractionnaire

Sommaire

4.1	Introduction	70
4.2	Programmation linéaire fractionnaire mono-objectif	72
4.3	Programmation linéaire fractionnaire multi-objectifs	82
4.4	Conclusion	83

4.1 Introduction

Le terme programmation fractionnaire est utilisé pour désigner un type de problèmes d'optimisation où la fonction objectif est un quotient $f(x)/g(x)$, soumis à un ensemble de contraintes. L'intérêt porté à ce type de problèmes tient à la variété de problèmes d'optimisation rencontrés en ingénierie et en économie pour lesquels on considère l'optimisation d'un rapport de deux fonctions. Les programmes fractionnaires linéaires ou non linéaire, en variables réelles, en variables entières ou en variables binaires, apparaissent dans plusieurs domaines de la recherche opérationnelle tels que les bases de données, l'optimisation combinatoire, la programmation stochastique, etc. De nombreuses applications des programmes fractionnaires ont été décrites dans la littérature (voir [31, 32, 42, 58, 61, 62, 13]).

Les problèmes mathématiques d'optimisation avec une fonction objectif qui est un rapport de deux fonctions linéaires ont beaucoup d'applications : dans les do-

maines des finances (planification d'une entreprise, gestion de feuille de solde bancaire), de transport maritime, de ressources en eau, de santé, etc. En effet, dans telles situations, il est souvent question d'optimiser un rapport de deux fonctions tel que dette/capitaux propres, rendement/employé, coût effectif/coût standard, bénéfice/coût, inventaire/ventes, risque des actifs/capital, coût/patient, docteur/patient, etc, sujet à des contraintes. En outre, si les contraintes sont linéaires, nous obtenons un problème de programmation linéaire fractionnaire. Dans cette thèse, nous nous intéressons à cette catégorie des problèmes de la programmation fractionnaire.

Un des premiers programmes fractionnaires est un modèle d'équilibre pour une économie en expansion, présenté par Von Neumann en 1937 [75]. Le modèle détermine le taux de croissance d'une économie comme maximum du plus petit de plusieurs rapports sortie-entrée. Cependant, une étude systématique de la programmation fractionnaire a commencé beaucoup plus tard. En 1962, Charnes et Cooper ont publié leur papier classique dans lequel ils prouvent qu'un programme linéaire fractionnaire peut être réduit à un programme linéaire en utilisant un changement de variable non-linéaire approprié [19]. Séparément, Martos prouva en 1964 [52] que des programmes fractionnaires linéaires peuvent être résolus avec une procédure de parcours de sommets adjacents du domaine des solutions réalisables juste comme des programmes linéaires avec la méthode du Simplexe. Il a identifié des propriétés généralisées de convexité des rapports linéaires permettant une telle prolongation de la technique de programmation linéaire. Depuis, différentes approches ont été proposées dans la littérature pour résoudre les deux problèmes de programmation linéaire fractionnaire en variables réelles et en variables discrètes.

Notons que la programmation fractionnaire linéaire peut être aussi étendue aux problèmes de la programmation fractionnaire linéaire multi-objectifs où plusieurs rapports sont pris en considération. Le problème de la programmation multi-objectifs linéaire fractionnaire (MOLFP : Multiple Objective Linear Fractional Programming) est l'un des modèles les plus populaires utilisés dans la prise de décision à critères multiples. De nombreuses études et applications ont été rapportées dans la littérature, pour un aperçu de ces études et applications, voir par exemple [4, 21, 48, 59, 18, 14].

Contrairement à la programmation linéaire multi-objectifs (*MOLP*), Steuer [66] montre que l'ensemble des solutions efficaces d'un problème (*MOLFP*) n'est pas nécessairement fermé, certains des points intérieurs de l'ensemble des solutions réalisables peuvent être efficaces alors que d'autres ne le sont pas et les sommets efficaces ne sont pas tous reliés par des arêtes efficaces. Il devient difficile de générer l'ensemble des solutions efficaces. Dans ce chapitre nous présentons dans une première partie les résultats de base de la programmation linéaire fractionnaire mono-objectif ainsi que les principales approches de résolution. Ensuite, dans une deuxième partie, nous présentons la forme générale d'un problème de programmation linéaire fractionnaire multi-objectifs ainsi que quelques résultats liés à ce problème.

4.2 Programmation linéaire fractionnaire mono-objectif

4.2.1 Formulation mathématique

Un programme fractionnaire consiste à optimiser un objectif mis sous la forme d'un rapport de deux fonctions linéaires ou non linéaires, soumis à un ensemble de contraintes. Un problème de programmation fractionnaire est donc de la forme suivante :

$$(P_F) \begin{cases} \max & \frac{f(x)}{h(x)} \\ \text{s.c.} & x \in S, \end{cases} \quad (4.1)$$

où $S = \{x \in \mathbb{R}^n | g_i(x) \leq 0 \text{ pour } i \in \{1, \dots, m\}\}$; f, h et $g_i, i \in \{1, \dots, m\}$ des fonctions réelles définies sur \mathbb{R}^n dans \mathbb{R} , avec h ne s'annulant pas sur le sous-ensemble S de \mathbb{R}^n . En vérifiant les hypothèses classiques suivantes :

- Les fonctions f, h et $g_i, i \in \{1, \dots, m\}$ sont continues sur \mathbb{R}^n ;
- S est un domaine non vide et borné de \mathbb{R}^n ;
- $h(x) > 0, \forall x \in S$.

Lorsque f, h et $g_i, i \in \{1, \dots, m\}$ sont des fonctions affines et $x \in \mathbb{R}_+^n$ alors on a un problème de programmation linéaire fractionnaire mono-objectif ou encore un programme hyperbolique qui s'écrit sous la forme suivante :

$$(P_{LF}) \begin{cases} \max & z(x) = \frac{cx + \alpha}{dx + \beta} \\ & x \in S, \end{cases} \quad (4.2)$$

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

Il est clair que si d est un vecteur nul avec $\beta \neq 0$ alors, (P_{LF}) n'est autre qu'un problème de programmation linéaire. Si de plus, les variables sont astreintes à ne prendre que des valeurs entières ($x \in \mathbb{Z}^n$), on parle dans ce cas de problème de programmation linéaire fractionnaire en nombre entiers ou encore programme hyperbolique discret, noté (P_{ILF}) , qui fait l'objet de nos préoccupations dans la suite de ce travail.

4.2.2 Résultats et définitions

Fonctions convexes/fonctions convexes généralisées

Définition 4.1 On dit qu'une fonction $f : S \rightarrow \mathbb{R}$, $S \subseteq \mathbb{R}^n$ un ensemble convexe et non vide, est convexe si :

$$\forall x, y \in S, \forall \lambda \in [0, 1] \text{ on a : } f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y). \quad (4.3)$$

f est dite strictement convexe sur S si :

$$\forall x, y \in S, x \neq y, \forall \lambda \in]0, 1[, f(\lambda x + (1 - \lambda)y) < \lambda f(x) + (1 - \lambda)f(y). \quad (4.4)$$

Définition 4.2 On dit qu'une fonction $f : S \rightarrow \mathbb{R}$, $S \subseteq \mathbb{R}^n$ un ensemble convexe et non vide, est concave si :

$$\forall x, y \in S, \forall \lambda \in [0, 1] \text{ on a : } f(\lambda x + (1 - \lambda)y) \geq \lambda f(x) + (1 - \lambda)f(y). \quad (4.5)$$

f est dite strictement concave sur S si :

$$\forall x, y \in S, x \neq y, \forall \lambda \in]0, 1[, f(\lambda x + (1 - \lambda)y) > \lambda f(x) + (1 - \lambda)f(y). \quad (4.6)$$

Autrement dit, la fonction f est dite concave (resp. strictement concave) sur S si et seulement si $-f$ est convexe (resp. strictement convexe) sur S .

Définition 4.3

- Soit $f : S \subset \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction convexe sur un ensemble convexe et non vide S alors f est continue sur l'intérieur de S ($int(S)$).
- Toute combinaison linéaire à coefficients positifs de fonctions convexes est une fonction convexe.

Définition 4.4 f est dite quasiconcave sur S si

$$\forall x, y \in S, \forall \lambda \in [0, 1], f(\lambda x + (1 - \lambda)y) \geq \min\{f(x), f(y)\}. \quad (4.7)$$

Définition 4.5 Soit S un ensemble non vide de \mathbb{R}^n et $f : S \rightarrow \mathbb{R}$. f est dite différentiable en $\bar{x} \in int(S)$ s'il existe un vecteur gradient noté $\nabla f(\bar{x})$ et une fonction $\alpha : \mathbb{R}^n \rightarrow \mathbb{R}$ tels que :

$$f(x) = f(\bar{x}) + \nabla f(\bar{x})(x - \bar{x}) + \|x - \bar{x}\| \alpha(\bar{x}; x - \bar{x}), \forall x \in S.$$

où $\lim_{x \rightarrow \bar{x}} \alpha(\bar{x}; x - \bar{x}) = 0$ et $\nabla f(\bar{x}) = (\frac{\delta f(\bar{x})}{\delta x_1}, \dots, \frac{\delta f(\bar{x})}{\delta x_n})^t$.

Définition 4.6 Soit f différentiable sur S , f est dite pseudoconvexe sur S si

$$\forall x, y \in S, (x - y)^t \nabla f(x) \geq 0 \Rightarrow f(y) \geq f(x). \quad (4.8)$$

Définition 4.7 Soit f différentiable sur S , f est dite pseudoconcave sur S si

$$\forall x, y \in S, (x - y)^t \nabla f(x) \leq 0 \Rightarrow f(y) \leq f(x). \quad (4.9)$$

Proposition 4.1 Si f est convexe alors f est quasiconvexe. Si de plus, f est différentiable alors, f est pseudoconvexe.

Proposition 4.2 Si f est concave alors f est quasiconcave. Si de plus, f est différentiable alors, f est pseudoconcave.

Remarque 6 Les réciproques des deux propositions sont fausses en général.

Lemme 4.1 La fonction objectif f du problème (P_{LF}) est à la fois pseudoconvexe et pseudoconcave sur S .

Théorème 4.1 Soit x^* solution optimale du problème (P_{LF}) , alors x^* est un point extrême de S .

Les programmes linéaires fractionnaires partagent quelques propriétés importantes avec les programmes linéaires :

- a. f étant quasiconvexe sur S , tout minimum local est un minimum global ;
- b. Si une solution x vérifie les conditions suivantes d'optimalité de Kuhn-Tucker, elle est optimale pour (P_{LF}) :

$$(KT) \begin{cases} \nabla f^t(x) + v^t A - u^t = 0 \\ u^t x = 0 \\ u \geq 0, v \in \mathbb{R}. \end{cases} \quad (4.10)$$

Ceci est dû au fait que f est pseudoconvexe et les contraintes étant linéaires, elles sont quasiconvexes aussi. (Les conditions (KT) sont nécessaires et suffisantes pour une solutions optimale de (P_{LF})).

- c. Toute solution optimale est atteinte en un point extrême du polyèdre convexe S (théorème 4.1).

L'extention de l'algorithme du Simplexe au cas des programmes linéaires fractionnaires est due à Martos [52] et Swarup [70].

4.2.3 Résolution géométrique d'un problème linéaire fractionnaire

Selon Steuer [66], les programmes linéaires fractionnaires présentent l'intérêt particulier d'avoir des courbes de niveaux linéaires de leurs fonctions objectif.

Pour illustrer cette propriété, considérons une k -courbe niveau quelconque de la fonction objectif :

$$k = \frac{cx + \alpha}{dx + \beta}$$

Après simplification, nous obtenons : $cx + \alpha = k(dx + \beta)$.

Ce qui donne : $(c - kd)x = k\beta - \alpha$, qui est une expression linéaire de la k -courbe niveau de la fonction objectif.

Puisque k est quelconque, on constate que chaque courbe niveau du critère linéaire fractionnaire est linéaire sur S pourvu que le dénominateur ne soit pas nul sur S . Donc, si un programme linéaire fractionnaire (P_{LF}) possède une solution optimale, au moins un point extrême de S est optimal.

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

On appelle *ensemble rotation*, l'ensemble de tous les points d'intersection entre la 0-courbe niveau du numérateur et la 0-courbe niveau du dénominateur.

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

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

Exemple illustratif

Considérons le problème suivant :

$$(P_{LF}) \begin{cases} \max & z(x) = \frac{x_1+x_2-1}{5x_1+x_2-1} \\ s.c. & -2x_1 + x_2 \leq 0, \\ & 6x_1 + x_2 \leq 21, \\ & -2x_1 + 4x_2 \leq 6, \\ & x_1, x_2 \in \mathbb{N} \end{cases}$$

La courbe de niveau k est l'ensemble CN définit par :

$$CN = \left\{ (x_1, x_2) \mid (1 - 5k)x_1 + (1 - k)x_2 = 1 - k \right\}$$

Donc pour :

- ▷ $k = 0$, on a $x_1 + x_2 = 1$: courbe de niveau 0.
- ▷ $k = 1$, on a $x_1 = 0$: courbe de niveau 1.

Les lignes discontinues représentent les courbes de niveau 0 du numérateur et du dénominateur dont l'intersection est le point rotation $r = (0, 1)$. La flèche circulaire représente le gradient de la fonction linéaire fractionnaire, elle indique le sens et l'angle avec lequel se déplacent les courbes de niveaux. Tandis que c et d représentent respectivement les gradient des courbes de niveau 0 du numérateur et du dénominateur.

Le point extrême $x = (0, 0)$ de valeur optimale $z^* = 1$, qui correspond à la solution optimale de (P_{LF}) , est l'intersection du domaine D avec la courbe de niveau 1 en faisant déplacer la courbe de niveau 0 autour du point r suivant le sens de rotation trigonométrique.

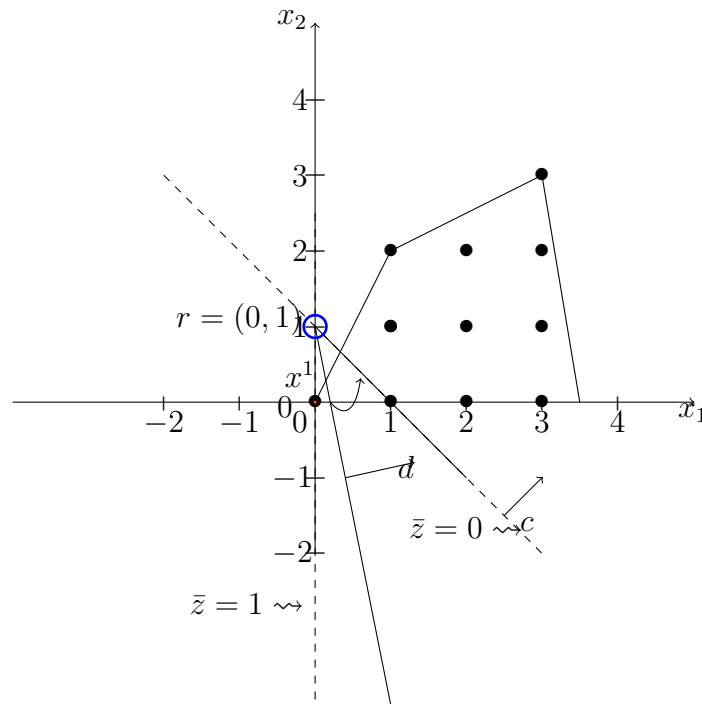


FIGURE 4.1 – Le domaine réalisable

4.2.4 Résolution d'un programme linéaire fractionnaire

Il existe dans la littérature, trois grandes stratégies de résolution d'un programme fractionnaire : la résolution directe, la résolution par paramétrisation et la résolution d'un programme équivalent à objectif simplifié.

4.2.5 La résolution directe

Dans cette stratégie, le programme fractionnaire est traité sous sa forme originale, c'est-à-dire sans modifier ni l'objectif ni l'ensemble des contraintes. Cette approche est utilisée pour résoudre les programmes hyperboliques tant qu'en variables continues qu'en variables entières ([9], [12], [37], [38]).

Méthode de Cambini et Martein [12]

On considère le programme linéaire fractionnaire en variables continues (P_{LF}) :

$$(P_{LF}) \begin{cases} \max & z = \frac{cx + \alpha}{dx + \beta} \\ x \in S. \end{cases}$$

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

La méthode est basée sur le concept de solution niveau optimale donné dans la définition suivante :

Définition 4.8 Une solution réalisable x^* est une *solution niveau optimale* pour le problème (P_{LF}) si x^* est une solution optimale du problème $P(x^*)$:

$$P(x^*) \begin{cases} \max & (cx + \alpha) \\ x \in & S \\ dx = & dx^* \end{cases}$$

Si de plus, x^* est un point extrême de S , x^* est dit solution niveau optimale de base.

L'algorithme génère une séquence finie $x^k, k = \overline{1, l}$ de solutions niveau optimales dont la première est trouvée de la façon suivante :

Résoudre le programme linéaire $(P_0) : \{\min dx + \beta \mid x \in S\}$, soit x^0 une solution optimale (car sa fonction objectif est bornée).

- Si x^0 est unique, alors elle est une solution niveau optimale de base, sinon résoudre le programme linéaire $P(x^0)$.
- Si $P(x^0)$ n'admet pas de solutions, alors la valeur de la fonction objectif est infinie ; sinon une solution optimale x^1 de $P(x^0)$ est aussi une solution niveau optimale de base pour (P_{LF}) .

Théorème 4.2 [12] *Le point x^k est une solution optimale du problème (P_{LF}) si et seulement si le vecteur gradient réduit $\hat{\gamma} = \hat{\beta}\hat{c} - \hat{\alpha}\hat{d}$ est tel que $\hat{\gamma}_j \leq 0$ pour tout indice hors base $j \in N_k$.*

Algorithm 6: Algorithme de Cambini et al.

Étape 1 Trouver la solution optimale niveau x^1 .

SI une telle solution n'existe pas **ALORS**

| $\sup \left\{ \frac{cx + \alpha}{dx + \beta} \mid x \in S \right\} = +\infty$, terminer.

SINON

| poser $k = 1$ et aller à l'étape 2.

Étape 2

SI $J = \{j \in N \mid \widehat{d}_j > 0\} = \emptyset$ **ALORS**

| Terminer, x^k est une solution optimale du problème (P_{LF})

SINON

| soit s tel que $\frac{\widehat{c}_s}{\widehat{d}_s} = \max_{j \in J} \left(\frac{\widehat{c}_j}{\widehat{d}_j} \right)$ **SI** $\widehat{\gamma}_s > 0$ **ALORS**

| aller à l'étape 3

SINON

| x^k est une solution optimale de (P_{LF}) .

Étape 3 La variable hors base x_s entre dans la base au moyen d'une opération pivot

SI Une telle opération n'est pas possible **ALORS**

| Terminer, $\sup \left\{ \frac{cx + \alpha}{dx + \beta} \mid x \in S \right\} = \frac{\widehat{c}_s}{\widehat{d}_s}$.

SINON

| poser $k = k + 1$ et aller à l'étape 2

Méthode de D. Granot and F. Granot [37]

Cette méthode est utilisée pour résoudre le problème de programmation linéaire fractionnaire en nombres entiers (P_{ILF}).

$$(P_{ILF}) \begin{cases} \max & z = \frac{cx + \alpha}{dx + \beta} \\ x \in D \end{cases}$$

où $D = \mathbb{Z}^n \cap S$, $S = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$.

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

$$\begin{cases} \max & \frac{\sum_{j \in N} \hat{c}_j x_j + \hat{\alpha}}{\sum_{j \in N} \hat{d}_j x_j + \hat{\beta}} \\ x_B + \hat{A}^N x_N = \hat{b} \\ x_B, x_N \geq 0, \text{ entiers} \end{cases}$$

où B est l'ensemble des indices de base et N est l'ensemble des indices hors-base. Dans cette méthode, la structure originale des contraintes est maintenue et les itérations sont réalisées dans un tableau du simplexe augmenté par trois lignes $(m+1)$, $(m+2)$ et $(m+3)$ correspondants respectivement aux vecteurs numérateur, dénominateur et au vecteur gradient réduit :

$$\hat{\gamma}_j = \hat{\beta} \hat{c}_j - \hat{\alpha} \hat{d}_j, \quad \forall j \in N. \quad (4.11)$$

A chaque itération de l'algorithme, les $(m+2)$ premières lignes sont modifiées à travers les opérations ordinaires de pivot, tandis que la dernière ligne est modifiée selon (4.11).

- Si $\hat{\gamma}_j \leq 0; \forall j \in N$, alors la solution courante est une solution optimale pour le problème (P_{ILF}).
- Sinon, il existe un indice s , $s \in N$ pour lequel $\hat{\gamma}_s > 0$. Soit θ_r tel que :

$$\theta_r = \min \left\{ \frac{\hat{b}_i}{\hat{A}_i^s} \mid \hat{A}_i^s > 0 \right\}$$

alors la ligne r sert comme une ligne pivot pour générer une coupe de Gomory de la forme :

$$v + \sum_{j \in N} \left[\frac{\hat{A}_{rj}}{\hat{A}_r^s} \right] x_j = \left[\frac{\hat{b}_i}{\hat{A}_r^s} \right], \quad v \geq 0. \quad (4.12)$$

Afin de résoudre (P_{ILF}), la coupe (4.12) est rajoutée au tableau du simplexe pour servir comme ligne pivot, avec la colonne s comme colonne pivot. Sachant que la valeur du pivot dans ce cas est de 1, les nouveaux coefficients obtenus après l'exécution des opérations de pivot sont tous des entiers.

4.2.6 La résolution par paramétrisation

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

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

Autour de cette approche, plusieurs algorithmes ont été proposés, nous citons entre autres : l'algorithme de Isbell et Marlow [43] en 1956 pour les programmes hyperboliques, l'algorithme généralisé de Dinkelbach [24] en 1967 pour les programmes fractionnaires non linéaires et l'algorithme de Seshan et Tikekar [63] pour les problème de programmation linéaire fractionnaire en nombres entiers.

L'algorithme de Dinkelbach [24]

Considérons le problème de programmation fractionnaire suivant :

$$(PF) \begin{cases} \max & F(x) = \frac{f(x)}{h(x)} \\ & x \in S. \end{cases}$$

L'ensemble S est supposé non vide, compact dans \mathbb{R}^n , les fonctions $f(x), h(x)$ sont des fonctions continues à valeurs réelles dans S , $h(x) > 0, \forall x \in S$.

Le problème paramétré associé $Q(\lambda)$ consiste à simplifier l'objectif en combinant linéairement le numérateur et le dénominateur par l'intermédiaire d'un paramètre réel λ .

$$Q(\lambda) \begin{cases} \max & v(\lambda) = f(x) - \lambda h(x) \\ & x \in S. \end{cases}$$

Le résultat fondamental liant le problème (PF) au problème $Q(\lambda)$ est donné par le théorème suivant

Théorème 4.3 *Soit $y \in S$, y est une solution optimale du problème (PF) si et seulement si y est une solution optimale du problème $Q(\lambda^*)$ avec $\lambda^* = F(y)$.*

Proposition 4.3 *La fonction $v(\lambda)$ est continue, strictement décroissante, convexe. $v(\lambda) > 0$ et $v(\lambda)$ tend vers $-\infty$ quand λ tend vers $+\infty$. Si de plus, le programme est hyperbolique, alors v est linéaire par morceaux.*

En particulier, l'équation $v(\lambda) = 0$ admet une solution unique λ^* , plus précisément :

Proposition 4.4

- a. $v(\lambda) = 0 \Leftrightarrow \lambda = \lambda^*$;
- b. $v(\lambda) > 0 \Leftrightarrow \lambda < \lambda^*$
- c. $v(\lambda) < 0 \Leftrightarrow \lambda > \lambda^*$

Algorithm 7: Algorithme de Dinkelbach [24]

Étape 1 Soit $x_1 \in S$ une solution réalisable initiale.

Poser $\lambda_1 = F(x_1)$, $k = 1$ et aller à l'étape 2.

Étape 2 Résoudre le problème $Q(\lambda_k)$, soit x_{k+1} une solution optimale, aller à l'étape 3.

Étape 3

SI $v(\lambda_k) = 0$ **ALORS**

 | Terminer, x_k une solution optimale de (PF)

SINON

 | poser $\lambda_{k+1} = F(x_{k+1})$, $k = k + 1$ et aller à l'étape 2

4.2.7 La résolution d'un programme équivalent

Un changement de variables permet de simplifier aussi l'objectif mais en faisant augmenter le nombre de variables et de contraintes. Cette transformation a été proposée par Charnes and Cooper [19] pour transformer un problème hyperbolique en un problème linéaire équivalent :

$$(P_{LF}) \begin{cases} \max & \frac{cx + \alpha}{dx + \beta} \\ Ax \leq b \\ x \geq 0 \end{cases} \Leftrightarrow (P_{eq}) \begin{cases} \max & cy + \alpha z \\ Ay - bz \leq 0 \\ dy + \beta z = 1 \\ y \geq 0, z \geq 0 \end{cases}$$

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

Proposition 4.5 (Charnes & Cooper [19]) Si (y^*, z^*) est une solution optimale de (P_{eq}) , alors $z^* > 0$ et $x^* = \frac{y^*}{z^*}$ est une solution optimale de (P_{LF}) .

Cette transformation a pour but d'appliquer les algorithmes standards tels que l'algorithme du Simplexe [19]. Pour les programmes fractionnaires en variables entières, Granot et al. [37] proposent une méthode de génération de coupes (de type Gomory) appliquée au programme linéaire (P_{eq}). D'autre part, dans le cas d'un programme hyperbolique à n variables bivalentes, Williams [76] propose une transformation spécifique en un programme linéaire équivalent en variables mixtes dont la taille croit de $n + 1$ variables continues et de $3n$ contraintes. Ce nouveau programme est résolu par un algorithme de type "branch & bound".

4.3 Programmation linéaire fractionnaire multi-objectifs

4.3.1 Formulation du problème

Le problème de la programmation linéaire fractionnaire multi-objectifs (*MOLFP* : Multiple Objective Linear Fractional Programming problem) est l'un des modèles les plus populaires utilisés dans la prise de décision à critères multiples, la formule générale de ce problème peut être donnée comme suit :

$$(MOLFP) \begin{cases} \text{“max”} & Z_i(x) = \frac{c^i x + \alpha^i}{d^i x + \beta^i} \quad i = \overline{1, p} \\ x \in S. \end{cases}$$

où : p est le nombre d'objectifs ($p \geq 2$), $S = \{x \in \mathbb{R}^n | Ax \leq b, x \geq 0\}$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, c^i et d^i sont des vecteurs de $\mathbb{R}^{1 \times n}$, α^i et β^i sont des réels $\forall i = \overline{1, p}$.

Nous supposons que l'ensemble S est non vide et compact dans \mathbb{R}^n et que pour tout $i = \overline{1, p}$, on a $d^i x + \beta^i > 0, \forall x \in S$.

Dans de nombreuses situations réelles modélisables par la programmation mathématique, les variables intervenant dans la modélisation sont soumises à être totalement en nombres entiers, on parle alors de problème de programmation multi-objectifs linéaire fractionnaire en nombres entiers (*MOILFP* : Multiple Objective Integer Linear Fractional Programming problem) défini comme suit

$$(MOILFP) \begin{cases} \text{“max”} & Z_i(x) = \frac{c^i x + \alpha^i}{d^i x + \beta^i} \quad i = \overline{1, p}, \\ x \in D, \end{cases}$$

où $D = \mathbb{Z} \cap S$, \mathbb{Z} étant l'ensemble des nombres entiers relatifs.

Comme pour les problèmes de programmation linéaire multi-objectifs, la résolution des problèmes de programmation linéaire fractionnaire multi-objectifs est de déterminer toutes les solutions qui sont efficaces au sens de la définition suivante :

Définition 4.9 Un point $x \in S$ est appelé *solution efficace*, ou solution *Pareto-optimale*, pour le problème (*MOLFP*), s'il n'existe pas un autre point $y \in S$ tel

que $Z_i(y) \geq Z_i(x), \forall i \in \{1, \dots, p\}$ et $Z_i(y) \neq Z_i(x)$ pour au moins un $i \in \{1, \dots, p\}$, et le vecteur $Z(x) = (Z_i(x))_{i=1, \dots, p}$ est dit *solution non dominée*.

4.3.2 Résolution du problème

Une recherche bibliographique nous a permis de constater que le problème de la programmation multi-objectifs linéaire fractionnaire (*MOLFP*) a été largement étudié par plusieurs auteurs. De nombreuses études et applications ont été rapportées dans la littérature, nous citons entre autre; Kornbluth et Steur [48] proposent un algorithme qui génère l'ensemble des solutions faiblement efficaces au moyen d'un algorithme basé sur la méthode du Simplexe. Une nouvelle technique pour optimiser la somme pondérée des fonctions objectifs linéaires fractionnaires est proposée par Costa dans [21], mais cette technique ne génère qu'une seule solution non-dominée pour le problème (*MOLFP*) associée à un vecteur de poids donné. Cependant, le problème multi-objectifs linéaire fractionnaire à variables entières (*MOILFP*) n'a pas reçu autant d'attention que le problème (*MOLFP*). On ne trouve que très peu de méthodes dédiées au problème (*MOILFP*), voir par exemple; Abbas et Moulaï dans [4] proposent une généralisation de leur méthode dans le cas d'un problème (*MOILP*) [3].

Dans [40], Gupta et Malhotra proposent une méthode qui généralise la méthode décrite par les même auteurs dans [39] au cas linéaire fractionnaire, mais la méthode a une défaillance dans le critère d'arrêt, elle s'arrête sans avoir généré toutes les solutions efficaces du problème, une amélioration de cette méthode est proposée par Moulaï dans [54]. Saad et Hughes ont décrit une méthode pour un problème bi-objectifs dans [59]. Chegrui et Moulaï dans [18] proposent une nouvelle méthode pour générer toutes les solutions efficaces du problème (*MOILFP*) en utilisant la méthode de Cambini et Martein [12] pour déterminer une solution optimale du problème relaxé (*MOLFP*), puis une solution entière est trouvée par l'application de processus de séparation de la méthode branch & bound. En outre, une coupe efficace est construite pour éliminer non seulement des solution non entières du domaine, mais aussi des solutions entières qui ne sont pas efficaces. Ainsi la méthode permet d'éviter l'énumération de toutes les solutions entières possibles du problème.

4.4 Conclusion

Dans ce chapitre, nous avons présenté d'une part, les résultats de base de la programmation linéaire fractionnaire mono-objectif ainsi que les principales approches de résolution d'un problème de programmation linéaire fractionnaire à variables continues et particulièrement à variables entières qui représente le noyau de notre thèse. D'autre part, nous avons exposé la formule générale d'un problème linéaire fractionnaire multi-objectifs suivi par un aperçu des méthodes de résolution existantes dans la littérature.

Dans le chapitre suivant, nous abordons notre première contribution dans le domaine multi-objectifs. Un algorithme exact, pour optimiser un critère linéaire fractionnaire sur l'ensemble de solutions efficaces d'un problème multi-objectifs en nombres entiers (*MOILP*) est présenté. Ce travail a fait l'objet d'une publication dans la revue International RAIRO Operations Research [49].

5

L'optimisation d'un critère linéaire fractionnaire sur un ensemble efficace discret

Sommaire

5.1	Introduction	85
5.2	Résultats de base	87
5.3	Description de la méthode	88
5.4	Illustration numérique	91
5.5	Expérimentations et résultats	95
5.6	Conclusion	97

5.1 Introduction

L'optimisation d'un critère qui exprime les préférences du décideur sur l'ensemble des solutions efficaces est l'un des sujets de recherche importants dans la programmation multi-objectifs, et il devient un moyen utile pour distinguer les préférences des décideurs parmi les nombreuses solutions efficaces. Ce problème est classé comme un problème d'optimisation globale ou non convexe et sa difficulté principale est due à la non-convexité de l'ensemble efficace, qui n'est pas connu explicitement.

Les problèmes d'optimisation avec une fonction objectif fractionnaire apparaissent dans plusieurs domaines et applications tels que, le domaine de finance (rendement/employé ; coût effectif/coût standard ; bénéfice/coût ;etc). Le domaine de l'économie (rendement/risque ; rendement/investissement ; inventaire/ventes ;...). Le domaine de la santé comme la planification dans un hôpital (coût/patient ; infirmière/patient ; docteur/patient ;...). Par conséquent, il est très intéressant d'avoir un outil puissant pour optimiser de tels objectifs sur un ensemble efficace, ce qui a justifié notre intérêt à étudier ce problème.

Comme nous l'avons déjà souligné dans le chapitre 3, le problème est largement étudié par de nombreux auteurs dans le cas d'un critère linéaire (à variables réelles ou entières). Cependant, le cas non linéaire, particulièrement, le cas d'un critère linéaire fractionnaire (un rapport de deux fonctions affines), n'a pas reçu autant d'attention. Ainsi, nous proposons un algorithme exact pour optimiser une fonction linéaire fractionnaire sur l'ensemble efficace d'un problème de programmation linéaire multi-objectifs en nombres entiers (*MOILP*), sans avoir à énumérer l'ensemble de toutes les solutions efficaces. L'algorithme est basé sur une simple technique de pivotage, en rassemblant la méthode de Cambini et Martein [12] par la méthode de Branch & bound [77] pour trouver des solutions entières. À chaque itération, le domaine d'admissibilité est réduit en ajoutant des contraintes, connues dans la littérature par "Corners constraints" [47], pour éliminer non seulement les solutions efficaces déjà parcourues mais pour éliminer aussi les régions dominées par ces solutions. Ensuite, la solution optimale courante est évaluée par le vecteur gradient réduit et une nouvelle direction qui améliore la valeur de la fonction objectif est définie.

Rappelons le problème (*MOILP*) suivant :

$$(P) \begin{cases} \text{“max”} & Cx \\ \text{s.c.} & x \in D. \end{cases}$$

où $D = S \cap \mathbb{Z}^n$, \mathbb{Z} étant l'ensemble des nombres entiers relatifs, $S = \{x \in \mathbb{R}^n | Ax \leq b, x \geq 0\}$, $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{Z}^{m \times 1}$, $C \in \mathbb{Z}^{p \times n}$ avec $(c^i)_{i \in \{1, \dots, p\}}$ sont des vecteurs lignes, $p \geq 2$.

On suppose que D est non vide et S est un polyèdre convexe et borné. L'ensemble des solutions efficaces entières du problème (P) est noté $E(P)$.

Le problème principal à étudier est défini par :

$$(FP_E) \quad \max \left\{ \varphi(x) = \frac{cx + \alpha}{dx + \beta}, x \in E(P) \right\}. \quad (5.1)$$

Le problème relaxé est donné par :

$$(FP_R) \quad \max \left\{ \varphi(x) = \frac{cx + \alpha}{dx + \beta}, x \in D \right\}. \quad (5.2)$$

où $c, d \in \mathbb{Z}^n$, α, β sont des scalaires et $dx + \beta > 0$ sur D .

Un point $x \in D$ est dit *solution efficace* pour le problème (P), si et seulement si,

il n'existe pas un autre point $\bar{x} \in D$ tel que $c^i(\bar{x}) \geq c^i(x)$ pour tout $i \in \{1, 2, \dots, p\}$ et $c^i(\bar{x}) > c^i(x)$ pour au moins un i . Le vecteur critère $Z(x) = Cx$ est dit *solution non-dominée*. Autrement, x est non efficace et le vecteur critère correspondant $Z(x)$ est dominé.

On utilisera tout au long de ce chapitre les notations suivantes :

Considérons le problème mono-objectif linéaire fractionnaire suivant :

$$(FP_k) \begin{cases} \max & \varphi(x) = \frac{cx+\alpha}{dx+\beta} \\ \text{s.c.} & x \in D_k \end{cases} \quad (5.3)$$

Où

- D_k est la région courante à l'itération k ;
- x^k est une solution optimale entière de (FP_k) obtenue dans D_k ;
- $a_{k,j}$ est le vecteur d'activité de x_j^k correspondant à la région courante D_k ;
- B_k est la base associée à la solution x^k ;
- $I_k = \{i \mid a_{k,i} \text{ est une colonne de } B_k\}$ l'ensemble des indices correspondants aux variables de base ;
- $N_k = \{j \mid a_{k,j} \text{ n'est pas une colonne de } B_k\}$ l'ensemble des indices correspondants aux variables hors-base ;
- $y_{k,j} = (y_{k,ij}) = (B_k)^{-1}a_{k,j}$.
- x_{opt} est une solution optimale de (FPE) .
- $\varphi_{opt} = \varphi(x_{opt})$ est la valeur optimale du critère principal $\varphi(x)$.

Les définitions et les résultats de base utilisés dans ce chapitre sont présentés dans la section suivante, la description de la méthode ainsi que l'algorithme détaillé sont présentés dans la section 3. Dans la section 4, une illustration numérique est incluse pour expliquer l'algorithme proposé, les résultats expérimentaux sont présentés dans la section 5. La section 6 conclut le chapitre.

5.2 Résultats de base

L'approche adoptée pour résoudre le problème (FP_k) à chaque itération k , est la méthode de Cambini et Martein [12], qui est principalement basée sur l'évaluation du vecteur gradient réduit $\hat{\gamma}$ défini par

$$\hat{\gamma} = \hat{\beta}\hat{c} - \hat{\alpha}\hat{d}$$

où $\hat{c}, \hat{d}, \hat{\alpha}$ et $\hat{\beta}$ sont les valeurs mises à jour de c, d, α et β respectivement. Notons que, afin de trouver une solution optimale entière de (FP_k) , une intégration du processus de la méthode Branch & bound est éventuelle.

Le théorème suivant nous permet de trouver une solution optimale de (FP_k) :

Théorème 5.1 [52] *Le point $\hat{x} \in S$ est une solution optimale du problème (FP_k) si et seulement si $\hat{\gamma}_j \leq 0$ pour tout $j \in N_k$.*

Remarque 7 Rappelons qu'une condition suffisante pour l'unicité de la solution optimale \hat{x} est que l'ensemble $J_k = \{j \in N_k \mid \hat{\gamma}_j = 0\}$ est vide. Autrement, il existe une autre solution $\tilde{x} \in D_k$ tel que $\varphi(\tilde{x}) = \varphi(\hat{x})$ et nous définissons \tilde{x} comme une solution alternative à \hat{x} .

5.2.1 Test d'efficacité

Dans la procédure proposée, on est appelé à tester l'efficacité des solutions réalisables en utilisant le résultat, décrit dans le chapitre 3, suivant

Théorème 5.2 Soit x^* une solution réalisable, $x^* \in E(P)$ si est seulement si la fonction objectif Θ est nulle dans le problème de programmation linéaire mixte suivant :

$$T_{eff}(x^*) \begin{cases} \max \Theta = \sum_{i=1}^p \psi_i \\ s.c. \quad Cx - I\psi = Cx^* \\ x \in D \\ \psi = (\psi_i)_{i=1, \dots, p}, \psi_i \in \mathbb{R}^+, \forall i \end{cases} \quad (5.4)$$

5.2.2 Arête incidente

L'algorithme proposé est basé principalement sur l'exploration des arêtes incidentes à une solution trouvée

Définition 5.1 Une arête E_{j_k} , $j_k \in N_k$ incidente à x^k est définie comme étant l'ensemble :

$$E_{j_k} = \left\{ X = (x_i) \in \mathbb{R}^{n_k} \left| \begin{array}{l} x_i = x_i^k - \theta_{j_k} y_{k,ij_k}, i \in I_k; \\ x_{j_k} = \theta_{j_k}; \\ x_l = 0, \forall l \in N_k \setminus \{j_k\} \end{array} \right. \right\} \quad (5.5)$$

où : $0 < \theta_{j_k} \leq \theta = \min_{i \in I_k} \left\{ \frac{x_i^k}{y_{k,ij_k}} \mid y_{k,ij_k} > 0 \right\}$.

Les points entiers se trouvant sur l'arête E_{j_k} sont identifiées de telle sorte que θ_{j_k} soit entier et $\theta_{j_k} \times y_{k,ij_k}$ entier $\forall i \in I_k$.

Remarque 8 Notons que (5.5) nous permet de déterminer les solutions réalisables alternatives quand la solution optimale obtenue en résolvant le problème (FP_k) n'est pas unique ($J_k \neq \emptyset$).

5.3 Description de la méthode

La procédure commence par résoudre le problème relaxé (FPR) . Évidemment, si (FPR) est irréalisable; le problème (FPE) est également irréalisable. Si ce n'est

pas le cas, la solution optimale de (FP_R) , notée x^0 , est testée pour l'efficacité afin d'obtenir une première solution efficace \hat{x}^0 . Ceci est fait en résolvant le problème $T_{eff}(x^0)$ (voir l'Eq. 5.4). Dans l'espace des critères plusieurs solutions efficaces peuvent avoir le même vecteur critères (Solutions efficaces équivalentes). Pour choisir la meilleure solution équivalente sur le critère principal, le problème (FT_k) est résolu.

$$(FT_k) : \max \left\{ \varphi(x) = \frac{cx + \alpha}{dx + \beta} \mid Cx = C\hat{x}^k, x \in D \right\}. \quad (5.6)$$

Soit \bar{x}^k une solution optimale de (FT_k) , les valeurs x_{opt}, φ_{opt} sont mises à jour. Faire $x_{opt} = \bar{x}^k, \varphi_{opt} = \varphi(\bar{x}^k)$.

Ensuite, le domaine d'admissibilité est réduit progressivement en ajoutant des contraintes éliminant successivement toutes les solutions dominées par la solution efficace courante \bar{x}^k en résolvant le problème (FP_k) :

$$(FP_k) : \max \left\{ \varphi(x) = \frac{cx + \alpha}{dx + \beta} \mid x \in D_k = D \setminus \bigcup_{s=0}^{k-1} D_s \right\}, \quad (5.7)$$

où $D_s = \{x \in \mathbb{Z}^n \mid Cx \leq C\bar{x}^s\}$ et $\bar{x}^0, \bar{x}^2, \dots, \bar{x}^s$ sont les solutions efficaces obtenues jusqu'à l'itération k .

Le domaine réduit D_k peut être défini par les contraintes suivantes :

$$D_k = D_{k-1} \cap \left\{ x \in D \mid \begin{array}{l} c^i(x) \geq (c^i(\bar{x}^k) + 1)y_i^k - M_i(1 - y_i^k) \quad (*) \\ \sum_{i=1}^p y_i^k \geq 1, y_i^k \in \{0, 1\}, i = \overline{1, p} \quad (**) \end{array} \right\}$$

où $D_0 = D$ et $-M_i$ est la borne inférieure de la i^{me} fonction objectif dans D . Notons que, quand $y_i^k = 0$, la contrainte $(*)$ n'est pas restrictive et lorsque $y_i^k = 1$, la i^{me} fonction objectif est strictement améliorée. La contrainte $(**)$ veut dire qu'au moins un des critères est amélioré.

Soit x^k une solution optimale de (FP_k) , si elle est efficace, la procédure se termine avec x^k comme une solution optimale du problème principal (FP_E) .

Autrement, une procédure d'exploration est appliquée sur les arêtes incidentes à x^k en utilisant le gradient réduit $\hat{\gamma}_j$ de la fonction objectif, cherchant une solution alternative efficace qui améliore la fonction φ . Si une telle solution ne peut être trouvée, le processus continue à réduire le domaine d'admissibilité et améliorer la valeur de φ jusqu'à ce qu'une solution optimale soit obtenue ou la région réduite devient vide.

Algorithm 8: L'optimisation sur un ensemble efficace

Entrées:

- ↓ $A_{(m \times n)}$: matrice des contraintes ;
- ↓ $b_{(m \times 1)}$: vecteur second membre ;
- ↓ $c_{(1 \times n)}, \alpha$: vecteur de numérateur du critère principal ;
- ↓ $d_{(1 \times n)}, \beta$: vecteur de dénominateur du critère principal ;
- ↓ $C_{(p \times n)}$: matrice des critères ;

Sorties:

↑ $x_{opt}, \uparrow \varphi_{opt}$: solution optimale, valeur optimale du critère φ .

Phase Initiale $\varphi_{opt} = -\infty, k = 0$.

pour $i = 1, \dots, p$ **faire**

└ calculer $-M_i = \{\min c^i x \mid x \in D\}$.

(Étape 1.) Résoudre (FP_R) .

SI (FP_R) est irréalisable **ALORS**

└ Terminer, (FP_E) est irréalisable.

SINON

└ Soit x^0 une solution optimale de (FP_R)

(Étape 2.) Tester l'efficacité de x^0

SI $x^0 \in E(P)$ **ALORS**

└ Terminer, $x_{opt} = x^0, \varphi_{opt} = \varphi(x^0)$.

SINON

└ Soit \hat{x}^0 une solution optimale de $T_{eff}(x^0)$.

(Étape 3.) Résoudre (FT_k) , soit \bar{x}^k une solution optimale ;

SI $\varphi(\bar{x}^k) > \varphi_{opt}$ **ALORS** $x_{opt} = \bar{x}^k, \varphi_{opt} = \varphi(\bar{x}^k)$.

(Étape 4.) $k := k + 1$, résoudre le problème (FP_k) .

SI (FP_k) est irréalisable **ALORS**

└ Terminer, x_{opt} est une solution optimale de (FP_E) .

SINON

┌ Soit x^k une solution optimale de (FP_k)

┌ **SI** $\varphi(x^k) \leq \varphi_{opt}$ **ALORS**

└ Terminer, x_{opt} une solution optimale de (FP_E) .

┌ **SINON**

└ Tester l'efficacité de x^k

└ **SI** $x^k \in E(P)$ **ALORS**

└ Terminer, $x_{opt} = x^k, \varphi_{opt} = \varphi(x^k)$

└ **SINON**

└└ Soit \hat{x}^k une solution optimale de $T_{eff}(x^k)$

(Étape 5.) Soit $J_k = \{j_k \in N_k \mid \hat{\gamma}_{j_k} = 0\}$.

SI $J_k = \emptyset$ **ALORS**

└ Aller à l'étape 3.

SINON

┌ **tant que** $J_k \neq \emptyset$ **faire**

└ choisir $j_k \in J_k$; explorer l'arête E_{j_k} correspondante à θ_{j_k} en

└ commençant de $\theta_{j_k} = \min_{i \in I_k} \left\{ \frac{x_i^k}{y_{k,ij_k}} ; y_{k,ij_k} > 0 \right\}$ jusqu'à $\theta_{j_k} = 1$.

└ **SI** une solution alternative efficace existe, soit \tilde{x} , **ALORS**

└ Terminer, $x_{opt} = \tilde{x}, \varphi_{opt} = \varphi(\tilde{x})$

└ **SINON**

└└ $J_k = J_k \setminus \{j_k\}$

Proposition 5.1 L'algorithme proposé converge dans un nombre fini d'itérations.

Preuve Comme la région d'admissibilité D est supposée non vide et bornée, D possède un nombre limité de solutions réalisables entières. Ainsi, le nombre de solutions efficaces $|E(P)|$ est fini. Dans chaque itération l'algorithme génère une nouvelle solution efficace pour améliorer la valeur de φ et le domaine se réduit progressivement jusqu'à ce qu'il devient vide. Ainsi, la procédure converge vers une solution optimale dans un nombre fini d'itérations.

5.4 Illustration numérique

Dans cet exemple didactique, nous exposons les différentes étapes de l'algorithme. Considérons le problème (MOILP) suivant :

$$(P) \left\{ \begin{array}{l} \max \quad Z_1 = x_1 - 3x_2 \\ \max \quad Z_2 = x_1 + 3x_2 \\ D \quad \left\{ \begin{array}{l} x_1 + 2x_2 \leq 8 \\ 2x_1 + x_2 \leq 7 \\ x_1 - x_2 \leq 2 \\ x_1, x_2 \in \mathbb{N} \end{array} \right. \end{array} \right.$$

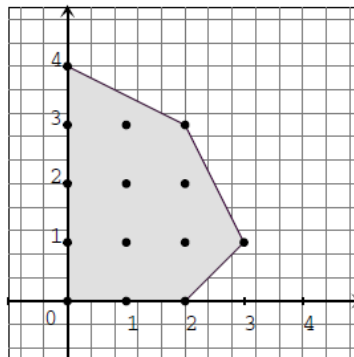


FIGURE 5.1 – Le domaine d'admissibilité D

Le problème principal (FP_E) est défini par

$$(FP_E) \left\{ \begin{array}{l} \max \quad \varphi(x) = \frac{-5x_1 - x_2 - 1}{4x_1 + x_2 + 1} \\ s.t. \quad x_1, x_2 \in E(P). \end{array} \right.$$

Le problème relaxé est

$$(FP_R) \left\{ \begin{array}{l} \max \quad \varphi(x) = \frac{-5x_1 - x_2 - 1}{4x_1 + x_2 + 1} \\ D \quad \left\{ \begin{array}{l} x_1 + 2x_2 \leq 8 \\ 2x_1 + x_2 \leq 7 \\ x_1 - x_2 \leq 2 \\ x_1, x_2 \in \mathbb{N} \end{array} \right. \end{array} \right.$$

Step 0. Poser $\varphi_{opt} = -\infty, k := 0, D_0 = D$, la solution optimale du problème (FP_R) est $x^0 = (0, 0)$ et $\varphi(x^0) = -1$.

B	x_1	x_2	x_3	x_4	x_5	x_B
x_3	1	2	1	0	0	8
x_4	2	1	0	1	0	7
x_5	1	-1	0	0	1	2
\hat{c}	-5	-1	0	0	0	-1
\hat{d}	4	1	0	0	0	1
$\hat{\gamma}$	-1	0	0	0	0	-1

TABLE 5.1 – Le tableau optimal correspondant à x^0 .

Step 1. Pour tester l'efficacité de x^0 , nous résolvons le problème $T_{eff}(x^0)$

$$T_{eff}(x^0) \left\{ \begin{array}{l} \max \quad \Theta = \psi_1 + \psi_2 \\ s.t. \quad x_1 + 2x_2 \leq 8 \\ \quad \quad 2x_1 + x_2 \leq 7 \\ \quad \quad x_1 - x_2 \leq 2 \\ \quad \quad x_1 - 3x_2 - \psi_1 = 0 \\ \quad \quad x_1 + 3x_2 - \psi_2 = 0 \\ \quad \quad x_1, x_2 \in \mathbb{N}, \psi_{i,i=\overline{1,2}} \geq 0 \end{array} \right.$$

Nous obtenons $\Theta^* \neq 0$, $x^0 \notin E(P)$ et la solution optimale de $T_{eff}(x^0)$ est $\hat{x}^0 = (3, 1)$, $Z(\hat{x}^0) = (0, 6)$.

Step 2. Nous résolvons le problème (FT_0)

$$(FT_0) \left\{ \begin{array}{l} \max \quad \varphi(x) = \frac{-5x_1 - x_2 - 1}{4x_1 + x_2 + 1} \\ s.t. \quad x_1 + 2x_2 \leq 8 \\ \quad \quad 2x_1 + x_2 \leq 7 \\ \quad \quad x_1 - x_2 \leq 2 \\ \quad \quad x_1 - 3x_2 = 0 \\ \quad \quad x_1 + 3x_2 = 6 \\ \quad \quad x_1, x_2 \in \mathbb{N} \end{array} \right.$$

Une solution optimale de (FT_0) est $\bar{x}^0 = (3, 1)$ et $\varphi(\bar{x}^0) = -1.21 > \varphi_{opt}$; nous initialisons $x_{opt} = (3, 1)$ et $\varphi_{opt} = -1.21$.

Step 3. Poser $k := k + 1 = 1, -M = (-12, 0)$ et résoudre (FP_1)

$$(FP_1) \left\{ \begin{array}{l} \max \quad \varphi(x) = \frac{-5x_1 - x_2 - 1}{4x_1 + x_2 + 1} \\ D_1 \quad \left\{ \begin{array}{l} x_1 + 2x_2 \leq 8 \\ 2x_1 + x_2 \leq 7 \\ x_1 - x_2 \leq 2 \\ -x_1 + 3x_2 + 13y_1^1 \leq 12 \quad (1) \\ -x_1 - 3x_2 + 7y_2^1 \leq 0 \quad (2) \\ y_1^1 + y_2^1 \geq 1 \\ (y_1^1, y_2^1) \in \{0, 1\}^2. \end{array} \right. \end{array} \right. \quad (1)$$

Soit $x^1 = (0, 3)$ une solution optimale avec $Z(x^1) = (-9, 9)$ et $\varphi(x^1) = -1$.

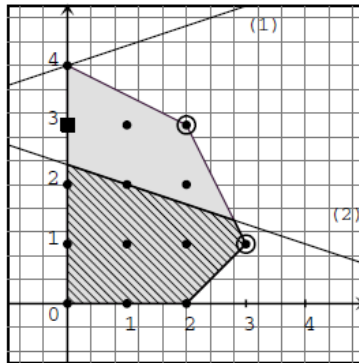


FIGURE 5.2 – Le domaine réduit D_1

Step 4.

Comme $\varphi(x^1) > \varphi_{opt}$, nous testons l'efficacité de cette solution en résolvant le problème $T_{eff}(x^1)$

$$T_{eff}(x^1) \left\{ \begin{array}{l} \max \quad \Theta = \psi_1 + \psi_2 \\ s.t. \quad x_1 + 2x_2 \leq 8 \\ \quad \quad 2x_1 + x_2 \leq 7 \\ \quad \quad x_1 - x_2 \leq 2 \\ \quad \quad x_1 - 3x_2 - \psi_1 = -9 \\ \quad \quad x_1 + 3x_2 - \psi_2 = 9 \\ \quad \quad x_1, x_2 \in \mathbb{N}, \psi_{i,i=\overline{1,2}} \geq 0 \end{array} \right.$$

Nous obtenons $x^1 \notin E(P)$ et $\hat{x}^1 = (2, 3)$ une solution optimale de $T_{eff}(x^1)$ avec $Z(\hat{x}^1) = (-7, 11)$.

Step 5. En utilisant le tableau optimale de x^1 , nous obtenons :

$$B = \{5, 6, 7, 4, 2, 3, 11, 12, 8, 13, 9\}; N = \{1, 10, 14, 15\}; \\ \hat{\gamma}_N = \{-4, 0, 0, 0\}; J_1 = \{j \in N / \hat{\gamma}_j = 0\} = \{10, 14, 15\};$$

B	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_B
x_5	1	0	0	0	1	0	0	0	0	0	0	0	0	0	2	2
x_6	2	0	0	0	0	1	0	0	0	0	0	0	0	0	1	4
x_7	1	0	0	0	0	0	1	0	0	0	0	0	0	0	-1	5
x_4	0	0	0	1	0	0	0	0	0	-1	0	0	0	-1	0	1
x_2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	-1	3
x_3	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0
x_{11}	0	0	0	0	0	0	0	0	0	0	1	0	0	-1	0	1
x_{12}	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0
x_8	-1	0	0	0	0	0	0	1	0	0	0	0	0	-13	3	3
x_{13}	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	2
x_9	-1	0	0	0	0	0	0	0	1	7	0	0	0	7	-3	2
\hat{c}	-5	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-4
\hat{d}	4	0	0	0	0	0	0	0	0	0	0	0	0	0	1	4
$\hat{\gamma}$	-4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1

TABLE 5.2 – Le tableau optimale de x^1 .

Notons que les variables x_3 et x_4 correspondent aux variables y_1^1 et y_2^1 respectivement. Comme $J_1 \neq \emptyset$, les arêtes incident à x^1 sont explorées (Voir Eq.(6.6)).

Nous trouvons la seule solution entière sur l'arête E_{15} défini par

$$E_{15} = \left\{ \begin{array}{ll} x_{15} = \theta_{15} = \min\{\frac{2}{2}; \frac{4}{1}; \frac{3}{3}\} & = 1; \\ x_5 = 2 - 1(2) & = 0; \\ x_6 = 4 - 1(1) & = 3; \\ x_7 = 5 - 1(-1) & = 6; \\ x_4 = 1 - 1(0) & = 1; \\ x_2 = 3 - 1(-1) & = 4; \\ x_3 = 0 - 1(0) & = 0; \\ x_{11} = 1 - 1(0) & = 1; \\ x_{12} = 0 - 1(0) & = 0; \\ x_8 = 3 - 1(3) & = 0; \\ x_{13} = 2 - 1(-1) & = 3; \\ x_9 = 2 - 1(-3) & = 5; \\ x_1 = x_{10} = x_{14} & = 0. \end{array} \right.$$

$\tilde{x}^1 = (0, 4)$ est efficace, la procédure se termine avec $x_{opt} = (0, 4)$ et $\varphi_{opt} = -1$.

L'ensemble de toutes les solutions efficaces du problème (P) est $E(P) = \{(2, 0), (3, 1), (2, 2), (2, 3), (0, 4)\}$. Tandis que, l'algorithme proposé optimise le critère principal φ sans devoir passer par toutes ces solutions mais seulement par $\{(3, 1), (2, 3), (0, 4)\}$.

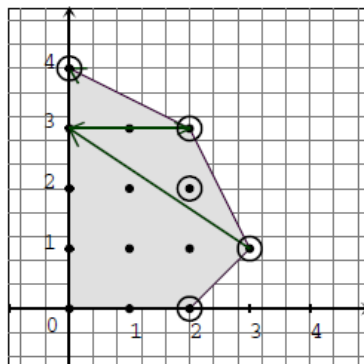


FIGURE 5.3 – les solutions efficaces de (P)

5.5 Expérimentations et résultats

Afin de tester la performance de l'algorithme proposé, l'implantation numérique a été réalisé sous l'environnement *MATLAB* et exécuter sur un PC Intel(R)Core(TM) i3 CPU 2.13 GHZ, la performance de l'algorithme est évaluée en utilisant 360 instances générées aléatoirement selon une distribution uniforme discrète : $A \in U([1, 30])$, $b \in U([25, 100])$ et $C \in U([-15, 15])$. Les vecteurs c, d , et la constante α sont générés de la même distribution que C . La constante β est générée telle que $dx + \beta > 0$.

Les instances traitées ont été regroupées en 36 catégories selon le nombre de variables, de contraintes et de fonctions objectifs. Le nombre de fonctions objectifs p a été fixé à 3, 5 et 8 et pour chaque catégorie 10 exécutions indépendantes ont été effectuées.

Les résultats rapportés dans le tableau (5.3) – le temps médian(en secondes), le nombre médian d'itérations requises, les valeurs minimums et maximums de chaque mesure – montre que l'algorithme proposé, pour des problème de petites et relativement moyennes dimensions, fonctionne efficacement en terme de nombre d'itérations ($\#iter$) et de temps d'exécution (cpu(sec)). Dans tous ces problèmes, le nombre de solutions efficaces produites (qui est égal au nombre d'itérations effectuées) est relativement petit par rapport aux tailles des instances traitées. En ce qui concerne les problèmes de plus grandes dimensions, la résolution de tels problèmes devient difficile due à plusieurs facteurs, tels que la nature multi-objectifs des problèmes, la nature discrète du domaine de recherche ainsi que le type du critère principal (une fonction linéaire fractionnaire).

p $n \times m$	$\underline{p = 3}$		$\underline{p = 5}$		$\underline{p = 8}$	
	cpu(sec)	# iter	cpu(sec)	# iter	cpu(sec)	# iter
5×5	0.39 [0.02; 1.56]	3.5 [1; 6]	0.6 [0.03; 23.5]	3 [1; 8]	0.67 [0.13; 38]	3 [1; 7]
10×5	1.09 [0.08; 15.64]	3.5 [1; 7]	2.26 [0.08; 34.54]	3.5 [1; 9]	3.25 [0.13; 75.35]	3 [1; 10]
15×5	3.21 [0.14; 96.88]	4.5 [1; 11]	5.18 [0.15; 48.69]	3.5 [1; 6]	8.56 [0.18; 91.29]	4 [1; 8]
20×5	4.39 [0.11; 39]	3.5 [1; 6]	6.14 [1.35; 129.6]	3 [1; 7]	8.02 [0.28; 141]	4 [1; 8]
20×10	6.56 [0.23; 101]	3.5 [1; 6]	7.42 [0.27; 148]	4 [1; 10]	14.5 [0.35; 165]	4.5 [1; 7]
30×10	18.04 [0.27; 153]	3.5 [2; 8]	18.11 [0.51; 213]	4 [1; 8]	20.5 [3.48; 195]	3.5 [1; 6]
35×15	22.27 [1.51; 296]	4 [2; 7]	25.79 [1.50; 312]	4 [1; 8]	24.69 [2.57; 325.27]	3.5 [1; 7]
40×15	32.27 [3.07; 387]	3.5 [1; 6]	38.42 [11.89; 398.36]	3.5 [2; 7]	48.93 [7.32; 430]	4.5 [1; 8]
50×15	75 [8; 370]	4 [2; 7]	84.47 [3.14; 714.88]	4.5 [1; 7]	69.75 [11.19; 516.59]	4.5 [2; 6]
60×20	97.35 [12.75; 500.25]	3.5 [2; 6]	88.7 [12.37; 506]	3.5 [2; 6]	102 [14.3; 1014]	3.5 [2; 8]
70×20	93.5 [23.35; 892]	3.5 [2; 6]	100 [29; 1355]	3 [2; 7]	125 [33.7; 1804]	3 [2; 7]
80×20	119 [29; 725]	3 [2; 5]	131 [32.53; 1874]	4 [2; 7]	126.3 [15.45; 1488]	3.5 [1; 6]

TABLE 5.3 – Résultats des instances générées aléatoirement.

5.6 Conclusion

Dans ce chapitre, nous avons présenté une nouvelle méthode exacte qui optimise une fonction linéaire fractionnaire sur l'ensemble efficace d'un problème (*MOILP*), les difficultés rencontrées sont principalement dues à la non-convexité de l'ensemble efficace. La complexité algorithmique des problèmes de programmation linéaires en nombres entiers, est bien connue NP-difficile. Néanmoins, la qualité de la solution demeure notre intérêt principal.

La méthode proposée résout le problème en évitant l'énumération explicite de toutes les solutions efficaces, rassemblant les coupes de Sylva et Crema pour éliminer les solutions efficaces précédemment trouvées, avec un processus d'exploration des arêtes incidentes en utilisant le vecteur gradient réduit de la solution optimale courante, afin de trouver une solution efficace alternative qui améliore la valeur de la fonction objectif.

L'algorithme a été programmé dans l'environnement (*MATLAB*) et examiné sur des instances aléatoirement générées selon une distribution uniforme discrète. Comme tous les algorithmes exacts, connus dans la littérature par "cutting plane algorithms", notre algorithme génère une solution optimale exacte pour des dimensions relativement moyennes dans un temps d'exécution (*CPU*) raisonnable. Pour des dimensions plus élevées, nous suggérons, comme perspective de recherche, une coopération d'une telle technique exacte avec des méta-heuristiques pour prendre en compte le temps d'exécution et la qualité des solutions. Dans le prochain chapitre, nous abordons notre deuxième contribution dans ce domaine, l'optimisation d'un critère linéaire fractionnaire sur l'ensemble efficace d'un problème de programmation linéaire fractionnaire multi-objectifs en nombres entiers (*MOILFP*).

6

L'optimisation d'un critère linéaire fractionnaire sur l'ensemble efficace d'un problème (*MOILFP*).

Sommaire

6.1	Introduction	98
6.2	Résolution du problème (FP_E)	100
6.3	Une nouvelle approche de résolution	107

6.1 Introduction

Dans ce chapitre, nous nous intéressons à l'optimisation d'une fonction linéaire fractionnaire, notée φ , sur l'ensemble efficace d'un problème de programmation linéaire fractionnaire multi-objectifs en nombres entiers (*MOILFP*). À notre connaissance le problème n'a pas vu beaucoup de développements, ce qui nous a motivé d'avantage à l'étudier en mettant au point, une nouvelle méthode exacte pour résoudre le problème dans l'espace des critères.

Dans la littérature, on ne trouve que très peu de méthodes dédiées au cas continu, le problème a été traité seulement par Muu et Tuyen [55, 71], Hackman et Passy [41]. Pour le cas discret, la seule méthode proposée pour résoudre le problème est celle de Zerdani et Moulaï en 2011 [74], dans laquelle, un algorithme exact est présenté

basé principalement sur le processus de séparation et évaluation de la méthode "Branch & bound" couplé à une nouvelle coupe efficace pour réduire le domaine des solutions réalisables.

Ainsi, nous proposons une nouvelle méthode exacte pour générer une solution optimale globale du problème en évitant l'énumération explicite de tout l'ensemble efficace, l'algorithme proposé est basé sur deux techniques :

- Afin de réduire la région d'admissibilité, nous proposons des coupes linéaires en nombres entiers, qui sont inspirées des coupes de Sylva et Crema [68], pour éliminer les solutions indésirables.
- Pour examiner l'efficacité d'une solution réalisable du problème (MOILFP), nous proposons une formulation linéaire équivalente au test d'efficacité d'Ecker et Kouada [26] où les critères fractionnaires sont linéarisés.

Rappelons que la formule générale d'un problème de programmation linéaire fractionnaire multi-objectifs en nombres entiers (MOILFP) est donnée par

$$(P) \begin{cases} \text{"max"} & Z_i(x) = \frac{N_i(x)}{D_i(x)}, i \in \{1, \dots, p\}, p \geq 2 \\ \text{s.c} & x \in D \end{cases} \quad (6.1)$$

où $D = S \cap \mathbb{Z}^n, S = \{x \in \mathbb{R}^n | Ax \leq b, x \geq 0\}, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^{m \times 1}, N_i(x) = c_i x + \alpha_i, D_i(x) = d_i x + \beta_i, \alpha_i, \beta_i$ sont des scalaires, $c_i, d_i \in \mathbb{R}^n, \forall i \in \{1, \dots, p\}$.

Dans toute la suite de ce chapitre, nous supposons que l'ensemble D est non vide, S est un polyèdre borné de \mathbb{R}^n et $d_i x + \beta_i > 0, \forall i \in \{1, \dots, p\}$. L'ensemble de toutes les solutions efficaces du problème (P) est noté par (EF), la solution optimale et la valeur optimale du critère principal sont notées x_{opt}, φ_{opt} respectivement.

Le problème principal que nous étudions est formulé par :

$$(FPE) \begin{cases} \max & \varphi(x) = \frac{N(x)}{D(x)} \\ \text{s.c} & x \in (EF) \end{cases} \quad (6.2)$$

où $\frac{N(x)}{D(x)} = \frac{cx + \alpha}{dx + \beta}; c, d \in \mathbb{R}^n, \alpha, \beta \in \mathbb{R}$ et $dx + \beta > 0$.

Nous définissons le problème relaxé par

$$(FPR) \begin{cases} \max & \varphi(x) = \frac{N(x)}{D(x)} \\ \text{s.c} & x \in D. \end{cases} \quad (6.3)$$

Comme tous les problèmes multi-objectifs, la solution du problème (P) est de déterminer les solutions efficaces ou Pareto-optimales dans le sens de la définition suivante :

Définition 6.1 Un point $x \in D$ est appelé **solution efficace** pour le problème (P), si et seulement si, il n'existe pas un autre point $\bar{x} \in D$ tel que $Z(\bar{x}) \geq Z(x)$ et $Z(\bar{x}) \neq Z(x)$. le vecteur $Z(x) = (Z_1(x), Z_2(x), \dots, Z_p(x))$ est dit **solution non-dominée**. Autrement, x est non efficace et le vecteur $Z(x)$ est dominée.

6.2 Résolution du problème (FP_E)

Nous présentons dans cette section en premier lieu, la seule méthode qui existe dans la littérature pour résoudre le problème (FP_E), la méthode de O. Zerdani et M. Moulai [73] suivie par notre nouvelle méthode de résolution. Pour les deux méthodes, nous présentons l'algorithme de la méthode avec une illustration numérique.

6.2.1 La méthode de O. Zerdani et M. Moulai [74]

L'approche adoptée dans cette méthode pour générer la solution optimale du problème (FP_E) est basée sur la résolution du problème mono-objectif linéaire fractionnaire (P_l) à chaque étape l de l'algorithme.

$$(P_l) \begin{cases} \max & Z_1(x) = \frac{c_1x + \alpha_1}{d_1x + \beta_1} \\ \text{s.c} & x \in S_l \end{cases} \quad (6.4)$$

Avec $S_0 = S$, notons qu'à la place de Z_1 , on peut similairement considérer le problème (P_l) avec une autre fonction objectif $Z_i, i \in \{2, \dots, p\}$.

- Une coupe est dite efficace pour le problème (P) si son adjonction au domaine S supprime au moins une solution réalisable continue de S mais ne supprime pas des solutions réalisables entières efficaces de S .
- Soit x_l^* la première solution entière obtenue après la résolution du problème (P_l) dans la première itération ($l = 0$) en utilisant la méthode de Cambini et Martein [12].
- $B_l (N_l)$ est l'ensemble des indices des variables de base (hors base) de x_l^* respectivement.
- Soit $\bar{\gamma}_j^i$ la j^{me} composante du vecteur gradient réduit $\bar{\gamma}^i$ défini comme suit

$$\bar{\gamma}^i = \bar{\beta}^i \cdot \bar{c}^i - \bar{\alpha}^i \cdot \bar{d}^i, \quad \forall i \in \{1, \dots, p\}, \quad (6.5)$$

où $\bar{c}^i, \bar{d}^i, \bar{\alpha}^i$ et $\bar{\beta}^i$ sont les valeurs mises à jour de c^i, d^i, α^i et β^i respectivement.

Théorème 6.1 [52] *Le point x_l^* est une solution optimale du problème (P_l) si et seulement si $\hat{\gamma}^i \leq 0$ pour tout $j \in N_l$.*

La direction d'amélioration de chaque critère $Z_i, i \in \{1, \dots, p\}$ et du critère principal φ est déterminée à l'aide de leurs vecteurs gradient. Le format original des fonctions objectifs fractionnaires $Z_i, i \in \{1, \dots, p\}$ du problème (P) ainsi que le critère principal φ et la structure d'origine des contraintes sont maintenus. Les itérations sont effectuées dans un tableau du simplexe augmenté de $(m + 3p + 3)$ lignes. Les m premières lignes correspondent aux contraintes d'origine, les lignes $m + 3(i - 1) + 1$ et $m + 3(i - 1) + 2$ correspondent au numérateur et dénominateur des fonctions objectifs $Z_i, i \in \{1, \dots, p\}$ respectivement, la ligne $m + 3i$ correspond au vecteur $\hat{\gamma}_l^i$ à l'étape l et les trois dernières lignes correspondent au numérateur,

dénominateur et $\widehat{\gamma}_l$ du critère principal φ à l'étape l respectivement.

À chaque étape de l'algorithme, toutes les lignes sont modifiées à travers les opérations ordinaires de pivotage à l'exception des lignes $m + 3i, i \in \{1, \dots, p\}$ et la ligne $m + 3p + 3$ qui sont modifiées en utilisant la relation (6.5).

La méthode utilise ces informations pour construire une coupe en mesure de supprimer des solutions entières qui ne sont pas efficaces pour le problème (P) .

Avant de déterminer l'expression mathématique de cette coupe efficace, il faut d'abord définir les ensembles suivants en x_l^* ,

$$H_l = H_l^1 \cup H_l^2$$

où :

$$H_l^1 = \{j \in N_l \mid \exists i \in \{1, \dots, p\}; \bar{\gamma}_j^i > 0 \text{ et } \bar{\gamma}_j \geq 0\};$$

$$H_l^2 = \{j \in N_l \mid \bar{\gamma}_j^i = 0, \forall i \in \{1, \dots, p\} \text{ et } \bar{\gamma}_j \geq 0\};$$

$$S_{l+1} = \left\{ x \in S_l \mid \sum_{j \in H_l} x_j \geq 1 \text{ ou } \varphi(x) \geq \varphi_{opt} \right\}.$$

La coupe efficace $\sum_{j \in H_l} x_j \geq 1$ élimine les solutions entières non efficaces et la coupe

$\varphi(x) \geq \varphi_{opt}$ élimine les solutions qui sont strictement plus mauvaises que la solution optimale courante φ_{opt} .

Définition 6.2 Une arête $E_{j_l}, j_l \in N_l$ incidente à x_l^* est définie par l'ensemble :

$$E_{j_l} = \left\{ x = (x_i) \in \mathbb{S}_l \mid \begin{array}{l} x_i = x_{l,i}^* - \theta_{j_l} y_{l,i,j_l}, i \in B_l; \\ x_{j_l} = \theta_{j_l}; \\ x_i = 0, \forall i \in N_l \setminus \{j_l\} \end{array} \right\} \quad (6.6)$$

où $0 < \theta_{j_l} \leq \theta = \min_{i \in B_l} \left\{ \frac{x_{l,i}^*}{y_{l,i,j_l}} \mid y_{l,i,j_l} > 0 \right\}$.

Les points entiers se trouvant sur l'arête E_{j_l} sont identifiées de telle sorte que θ_{j_l} et $\theta_{j_l} \times y_{l,i,j_l}$ sont des entiers $\forall i \in B_l$ si de telles valeurs de θ_{j_l} existent.

Définition 6.3 Soit $f : D \subset \mathbb{R}^n \rightarrow \mathbb{R}$ et $\bar{x} \in D$. Alors

$L_{\geq} f(\bar{x}) = \{x \in D : f(x) \geq f(\bar{x})\}$ est appelé *ensemble niveau* (level set) de \bar{x} pour f .

$L_{=} f(\bar{x}) = \{x \in D : f(x) = f(\bar{x})\}$ est appelé *courbe niveau* (level curve) de \bar{x} pour f .

Test d'efficacité

Dans l'approche proposée, pour tester l'efficacité d'une solution réalisable trouvée, les auteurs ont utilisés le théorème suivant.

Théorème 6.2 [27] *La solution $\bar{x} \in D$ est Pareto optimale pour le problème (P)*

si et seulement si $\bigcap_{i=1}^{i=p} L_{\geq} Z_i(\bar{x}) = \bigcap_{i=1}^{i=p} L_{=} Z_i(\bar{x})$.

L'algorithme de la méthode

Étape (0) Initialisation : $\varphi_{opt} = -\infty$

Résoudre le problème relaxé (FP_R)

- Si (FP_R) est irréalisable Alors (FP_E) est irréalisable.
- Sinon, soit x^0 la solution optimale de (FP_R).

Étape (1) x^0 est tester pour l'efficacité en appliquant le théorème (6.2)

- Si $x^0 \in (EF)$ Alors $x_{opt} = x^0$, Terminer.
- Sinon, Aller à l'étape (2).

Étape (2) Étape Générale

Tant qu'il existe un noeud non encore sondé, résoudre le problème (P_l) en utilisant le dual du Simplexe (la méthode de Cambini et Martein est utilisée seulement pour résoudre le problème initial (P_l), pour $l = 0$).

Étape (2a) :

- Si (P_l) est irréalisable, alors le noeud correspondant est sondé. Aller à l'étape (2).
- Sinon, soit \tilde{x}_l solution optimale obtenue.
 - Si \tilde{x}_l n'est pas entière. Aller à (2b).
 - Sinon, tester l'efficacité de \tilde{x}_l .
 - Si \tilde{x}_l est efficace, alors $x_{opt} = \tilde{x}_l$, $\varphi_{opt} = \varphi(\tilde{x}_l)$. Aller à (2c).
 - Sinon, Aller à l'étape (2c).

Étape (2b) Processus de branchement.

Choisir une composante non entière x_j de \tilde{x}_l telle que $x_j = \alpha_j$ avec α_j un nombre fractionnaire et séparer le noeud actuel l en deux nouveaux noeuds $k, k \geq l + 1$ et $h, h \geq l + 1, k \neq h$.

Dans le tableau du simplexe courant, la contrainte $x_j \leq \lfloor \alpha_j \rfloor$ est rajoutée et un nouveau domaine est considéré au noeud k , et la contrainte $x_j \geq \lfloor \alpha_j \rfloor$ est rajoutée pour obtenir un autre domaine au noeud h , Aller à l'étape (2a).

Étape (2c) Déterminer les ensembles N_l, H_l^1, H_l^2 et H_l , construire la coupe efficace.

- Si $H_l = \emptyset$, alors le noeud correspondant est sondé. Aller à l'étape (2).
- Sinon, explorer toutes les arêtes $E_{j_l}, j_l \in H_l$ incidentes à \tilde{x}_l et générer toutes les solutions entières \tilde{x}^u

(cas 1) Si de telles solutions existent, alors calculer

$$\bar{x}_k^u = \operatorname{argmax} \left\{ \varphi(\bar{x}^u) \mid \bar{x}^u \in \left(\bigcup_{j_l \in H_l} E_{j_l} \right) \setminus \{ \bar{x}_1^u, \bar{x}_2^u, \dots, \bar{x}_{k-1}^u \} \right\},$$

où \bar{x}_k^u est la première meilleure solution entière efficace trouvée sur toutes les arêtes. Alors

$$S_{l+1} = \{ x \in S_l \mid \varphi(x) \geq \varphi(\bar{x}_k^u) \},$$

mettre à jour x_{opt}, φ_{opt} et aller à l'étape (2).

(cas 2) Si toutes les solutions entières générées sur les arêtes E_{j_i} ne sont pas efficaces, alors

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

et aller à l'étape (2).

(cas 3) Si aucune solution entière n'est détectée sur toutes les arêtes E_{j_i} , alors

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

et aller à l'étape (2).

6.2.2 Illustration numérique

Les auteurs ont illustrés l'utilisation de son algorithme par le problème suivant :

$$(P) \begin{cases} \max Z_1 = \frac{-x_1+4}{x_2+1} \\ \max Z_2 = \frac{x_1+1}{x_1+2} \\ \max Z_3 = \frac{x_1-1}{x_1+1} \\ x \in D = \begin{cases} 4x_1 + 3x_2 \leq 20 \\ x_1 - x_2 \leq 3 \\ x_2 \leq 4 \\ x_1, x_2 \in N \end{cases} \end{cases}$$

Le problème principal à résoudre est :

$$(FPE) \begin{cases} \max \varphi(x) = \frac{5x_1+3x_2-4}{2x_1+x_2+1} \\ s.c. \quad x_1, x_2 \in (EF). \end{cases}$$

Étape (0) : Initialisation, poser $\varphi_{opt} = -\infty$ et résoudre le problème relaxé (FPR)

$$(FPR) \begin{cases} \max \varphi(x) = \frac{5x_1+3x_2-4}{2x_1+x_2+1} \\ s.c. \quad x_1, x_2 \in D. \end{cases}$$

La solution optimale est $x^0 = (2, 4)'$ et $\varphi(x^0) = 2$.

Étape (1) : x^0 est tester pour l'efficacité en appliquant le théorème (6.2), on a :

$$\bigcap_{i=1}^3 L_{\geq} Z_i(2, 4) = \{(2, 0)'; (2, 1)'; (2, 2)'; (2, 3)'; (2, 4)'; (3, 0)'; (3, 1)'\} \neq \bigcap_{i=1}^3 L_{=} Z_i(2, 4) = \{(2, 4)'\}.$$

Ainsi x^0 n'est pas efficace, aller à l'étape (2).

Étape (2) : Résoudre le problème (P_0)

$$(P_0) \begin{cases} \max Z_1 = \frac{-x_1+4}{x_2+1} \\ s.c. \quad x \in S_0 = S. \end{cases}$$

Les résultats de la résolution du problème (P_0), en utilisant la procédure de (Cambini et Martein [12]), sont résumés dans le Tableau 1.

Tableau 1	RHS	x_1	x_2
x_3	20	4	3
x_4	3	1	-1
x_5	4	0	1
x_6	-39	-13	-10
$-c^1$	-4	-1	0
$-d^1$	-1	0	1
$\bar{\gamma}^1$		-1	-4
$-c^2$	-1	1	0
$-d^2$	-2	1	0
$\bar{\gamma}^2$		1	0
$-c^3$	1	1	0
$-d^3$	-1	1	0
$\bar{\gamma}^3$		2	0
$-p$	4	5	3
$-q$	-1	2	1
$\bar{\gamma}$		13	7

Tableau 2	RHS	x_2	x_6
x_3	8	-1/13	4/13
x_4	0	-23/13	1/13
x_5	4	1	0
x_1	3	10/13	-1/13
x_7	-1	0	-1
$-c^1$	-1	10/13	-1/13
$-d^1$	-1	1	0
$\bar{\gamma}^1$		-3/13	-1/13
$-c^2$	-4	-10/13	1/13
$-d^2$	-5	-10/13	1/13
$\bar{\gamma}^2$		-10/13	1/13
$-c^3$	-2	-10/13	1/13
$-d^3$	-4	-10/13	1/13
$\bar{\gamma}^3$		-20/13	2/13
$-p$	-11	-11/13	5/13
$-q$	-7	-7/13	2/13
$\bar{\gamma}$		0	1

La solution optimale $x_1 = (0, 0)'$ étant entière donc elle est testée pour l'efficacité, on obtient $\bigcap_{i=1}^3 L_{\geq} Z_i(0, 0) = \bigcap_{i=1}^3 L_{=} Z_i(0, 0) = \{(0, 0)'\}$.

Ainsi x_1 est une première solution efficace correspondant à $\varphi(x_1) = -4 > -\infty$, d'où $x_{opt} = (0, 0)'$ et $\varphi_{opt} = -4$, aller à l'étape (2c).

$N_1 = \{1, 2\}; H_1^1 = \{1\}; H_1^2 = \emptyset; H_1 = H_1^1 \cup H_1^2 = \{1\} \neq \emptyset$.

On explore l'arête E_1 et on calcule $\theta_1 = \min(\frac{20}{4}; \frac{3}{1}) = 3$. Ainsi, on génère $\bar{x}_1^u = (3, 0)'; \bar{x}_2^u = (2, 0)'; \bar{x}_3^u = (1, 0)'$ trois solutions entières sur l'arête E_1 et $\bar{x}_1^u = \operatorname{argmax}\{\varphi(\bar{x}_i^u) | \bar{x}_i^u \in E_1, i \in \{1, \dots, 3\}\}$ avec \bar{x}_1^u une solution efficace qui correspond à $\varphi(3, 0) = 11/7 > -4$, donc $x_{opt} = (3, 0)'$ et $\varphi_{opt} = 11/7$.

Rajouter la coupe $\varphi(x) \geq \varphi_{opt} = 11/7 \Leftrightarrow -13x_1 - 10x_2 + x_6 = -39$ au Tableau 1 et on applique le dual du simplexe. La solution optimale entière $x_2 = (3, 0)'$ du programme (P_1) est obtenue dans le Tableau 2 et elle est efficace, aller à l'étape (2c).

$N_2 = \{2, 6\}; H_2^1 = \{6\}; H_2^2 = \emptyset; H_2 = \{6\} \neq \emptyset$. Aucune solution entière n'est détectée sur l'arête E_6 puisque $\theta_6 = 0$.

La contrainte $x_6 \geq 1$ est ajoutée au Tableau 2 et après pivotage, la solution optimale obtenue $x = (70/23; 1/23)$ n'est pas entière (Voir Tableau 4). Aller à l'étape (2b).

Tableau 4	RHS	x_4	x_7
x_3	177/23	-1/23	7/23
x_2	1/23	-13/23	-1/23
x_5	91/23	13/23	1/23
x_1	70/23	10/23	-1/23
x_6	1	0	-1
$(N_1) \mapsto x_8$	-22/23	-13/23	-1/23
$-c^1$	-22/23	10/23	-1/23
$-d^1$	-24/23	13/23	1/23
$\bar{\gamma}^1$		-2/23	-2/23
$-c^2$	-93/23	-10/23	1/23
$-d^2$	-116/23	-10/23	1/23
$\bar{\gamma}^2$		-10/23	1/23
$-c^3$	-47/23	-10/23	1/23
$-d^3$	-93/23	-10/23	1/23
$\bar{\gamma}^3$		-20/23	2/23
$-p$	-261/23	-11/23	8/23
$-q$	-164/23	-7/23	3/23
$\bar{\gamma}$		1/23	1

Tableau 6	RHS	x_8	x_9
x_3	5	3	4
x_2	1	-1	0
x_5	3	1	0
x_1	3	0	-1
x_6	10	-10	-13
x_4	1	-1	1
x_7	9	-10	-13
x_{10}	-12	-11	-12
$-c^1$	-1	0	-1
$-d^1$	-2	1	0
$\bar{\gamma}^1$		-1	-2
$-c^2$	-4	0	1
$-d^2$	-5	0	1
$\bar{\gamma}^2$		0	1
$-c^3$	-2	0	1
$-d^3$	-4	0	1
$\bar{\gamma}^3$		0	2
$-p$	-14	3	5
$-q$	-8	1	2
$\bar{\gamma}$		10	12

La séparation est déclenchée avec la création de deux noeud,

$(N_0) : x_2 \leq 0 \Leftrightarrow \frac{13}{23}x_4 + \frac{1}{23}x_7 \leq \frac{-1}{23}$. Ceci est impossible et le noeud (N_0) est sondé.

$(N_1) : x_2 \geq 1 \Leftrightarrow \frac{13}{23}x_4 + \frac{1}{23}x_7 \geq \frac{23}{23}$.

Cette contrainte est rajoutée au Tableau 4 et on obtient le Tableau 5 avec une solution optimale non entière $x = (\frac{30}{13}; 1)$, aller à l'étape (2b).

Deux autre noeuds sont encore créés :

$(N_2) : x_1 \geq 3 \Leftrightarrow \frac{1}{13}x_7 - \frac{10}{13}x_8 \geq \frac{9}{13}$.

$(N_3) : x_1 \leq 2 \Leftrightarrow \frac{1}{13}x_7 - \frac{10}{13}x_8 \leq \frac{-4}{13}$.

La contrainte $x_1 \geq 3$ est rajoutée au Tableau 5 et on obtient le Tableau 6 avec une solution entière optimale $x_3 = (3, 1)'$ qui n'est pas efficace, aller à l'étape (2c).

$N_3 = \{8, 9\}; H_3 = \{9\} \neq \emptyset$. On explore l'arête E_9 et on calcule $\theta_9 = 1$, la solution entière correspondante sur E_9 est $\bar{x}_1^u = (4, 1)'$.

Puisque $\bigcap_{i=1}^3 L_{\geq} Z_i(4, 1) = \bigcap_{i=1}^3 L_{=} Z_i(4, 1) = \{(4, 1)'\}$, donc \bar{x}_1^u est efficace avec

$\varphi(\bar{x}_1^u) = 19/10 > \varphi_{opt} = 11/7$, d'où $x_{opt} = (4, 1)'$ et $\varphi_{opt} = 19/10$.

La contrainte $\varphi(x) \geq 19/10 \Leftrightarrow -11x_8 - 12x_9 \leq -12$ est rajoutée au Tableau 6.

En procédant de cette manière, on obtient le Tableau 9 et $x_4 = (4, 1)'$ est une solution optimale entière efficace avec $\varphi(x_4) = \varphi_{opt}$, aller à l'étape (2c).

$N_4 = \{10, 12\}; H_4^1 = \{12\}; H_4^2 = \{10\}; H_4 = \{10, 12\} \neq \emptyset$. Aucune solution entière n'est détectée sur l'arête E_{10} et E_{12} puisque $\theta_{10} = \theta_{12} = 0$.

La coupe efficace $x_{10} + x_{12} \geq 1$ est rajoutée au Tableau 9 et le Tableau 10 est obtenu avec un dual non réalisable, alors le noeud correspondant est sondé.

Tableau 10	RHS	x_{12}	x_{13}
x_3	8/11	5/11	3/11
x_2	12/11	13/11	-1/11
x_5	32/11	-13/11	1/11
x_1	4	-1	0
x_6	263/11	-13/11	-10/11
x_4	-1/11	24/11	1/11
x_7	252/11	-13/11	-10/11
x_8	1/11	13/11	-1/11
x_9	1	-1	0
x_{11}	10/11	-13/11	1/11
x_{10}	1	1	-1
$-c^1$	0	-1	0
$-d^1$	-23/11	-13/11	1/11
$\bar{\gamma}^1$		-2/11	0
$-c^2$	-5	1	0
$-d^2$	-6	1	0
$\bar{\gamma}^2$		1	0
$-c^3$	-3	1	0
$-d^3$	-5	1	0
$\bar{\gamma}^3$		2	0
$-p$	-212/11	16/11	3/11
$-q$	-111/11	9/11	1/11
$\bar{\gamma}$		-12/11	1

Tableau 6'	RHS	x_7	x_9
x_3	78/10	3/10	-1/10
x_2	14/10	-1/10	-13/10
x_5	26/10	1/10	13/10
x_1	2	0	1
x_6	1	-1	0
x_4	24/10	-1/10	-23/10
x_8	4/10	-1/10	-13/10
$(N_5) \mapsto x_{10}$	-6/10	-1/10	-13/10
$-c^1$	-2	0	1
$-d^1$	-24/10	1/10	13/10
$\bar{\gamma}^1$		-2/10	-2/10
$-c^2$	-3	0	-1
$-d^2$	-4	0	-1
$\bar{\gamma}^2$		0	-1
$-c^3$	-1	0	-1
$-d^3$	-3	0	-1
$\bar{\gamma}^3$		0	-2
$-p$	-102/10	3/10	-11/10
$-q$	-64/10	1/10	-7/10
$\bar{\gamma}$		9/10	1/10

La seconde contrainte ($x_1 \leq 2$) (correspondante au noeud (N_3)), est rajoutée au Tableau 5 et le Tableau (6') est obtenu. En procédant de cette manière, on obtient le Tableau (8') et la solution optimale entière obtenue est $x_5 = (2, 2)'$ qui n'est pas efficace, aller à l'étape (2c). $H_5 = \{11\} \neq \emptyset$ et $\theta_{11} = 0$.

La coupe efficace $x_{11} \geq 1$ est ajoutée au Tableau (8') et le Tableau (9') est obtenu avec un dual irréalisable indiquant que le noeud correspondant est sondé.

Tableau 9'	RHS	x_{10}	x_{12}
x_3	2	3	4
x_2	2	-1	0
x_5	2	1	0
x_1	3	0	-1
x_6	20	-10	13
x_4	2	-1	1
x_8	1	-1	0
x_9	-1	0	1
x_7	19	-10	-13
x_{11}	1	0	-1
$-c^1$	-1	0	-1
$-d^1$	-3	1	0
$\bar{\gamma}^1$		-1	-3
$-c^2$	-4	0	1
$-d^2$	-5	0	1
$\bar{\gamma}^2$		0	1
$-c^3$	-2	0	1
$-d^3$	-4	0	1
$\bar{\gamma}^3$		0	2
$-p$	-17	3	5
$-q$	-9	1	2
$\bar{\gamma}$		10	11

L'algorithme s'arrête puisque tous les noeuds créés sont sondés et la solution optimale du problème principal (FP_E) est alors $x_{opt} = (4, 1)'$ et $\varphi_{opt} = 19/10$.

6.3 Une nouvelle approche de résolution

Dans cette section, nous introduisons une nouvelle méthode exacte pour résoudre le problème (FP_E) dans l'espace des critères en évitant le passage systématique par toutes les solutions efficaces.

Nous commençons par présenter les résultats de base utilisés dans ce travail, suivi par la présentation détaillée de notre méthode. Ensuite, nous illustrons le fonctionnement de l'algorithme proposé sur trois exemples numériques et nous terminons par des résultats expérimentaux pour valider la performance de notre algorithme.

6.3.1 Résultats de base

Test d'efficacité

Nous rappelons la définition du Test d'efficacité proposé par Ecker et Kouada [26] dans le théorème suivant

Théorème 6.3 *Une solution réalisable $x^* \in D$, $x^* \in (EF)$ si est seulement si la fonction objectif Θ est nulle dans le problème de programmation linéaire mixte suivant :*

$$T(x^*) \begin{cases} \max & \Theta = \sum_{i=1}^p \psi_i \\ \text{s.c.} & Z_i(x) - \psi_i = Z_i(x^*), i = 1, \dots, p \\ & x \in D \\ & \psi_i \in \mathbb{R}^+; \forall i = \overline{1, p}. \end{cases} \quad (6.7)$$

L'application de ce théorème sur le problème (MOILFP) nous conduit à résoudre un problème de programmation non linéaire à variables mixtes. Pour éviter cette situation, nous proposons une formulation linéaire équivalente au problème $T(x^*)$ dans le théorème suivant :

Théorème 6.4 *Une solution réalisable $x^* \in D$, $x^* \in (EF)$ si est seulement si la fonction objectif Θ est nulle dans le problème de programmation linéaire mixte suivant :*

$$FT(x^*) \begin{cases} \max & \Theta = \sum_{i=1}^p \psi_i \\ \text{s.c.} & (c_i x + \alpha_i) - Z_i(x^*)(d_i x + \beta_i) = \psi_i, i = 1, \dots, p \\ & x \in D \\ & \psi_i \in \mathbb{R}^+; \forall i = \overline{1, p}. \end{cases} \quad (6.8)$$

Preuve

Soit $x^* \in (EF)$, comme $\psi_i \geq 0$ pour tout $i \in \{1, \dots, p\}$ on a alors :

$$\begin{aligned} Z_i(x) \geq Z_i(x^*), \forall i = \overline{1, p} &\Rightarrow \frac{c_i x + \alpha_i}{d_i x + \beta_i} \geq Z_i(x^*), \forall i = \overline{1, p}; \\ &\Rightarrow (c_i x + \alpha_i) \geq Z_i(x^*)(d_i x + \beta_i), \forall i = \overline{1, p}; \\ &\Rightarrow (c_i x + \alpha_i) - Z_i(x^*)(d_i x + \beta_i) \geq 0, \forall i = \overline{1, p}; \\ &\Rightarrow (c_i x + \alpha_i) - Z_i(x^*)(d_i x + \beta_i) - \psi_i = 0, \psi_i \geq 0, \forall i = \overline{1, p}; \\ &\Rightarrow (c_i x + \alpha_i) - Z_i(x^*)(d_i x + \beta_i) = \psi_i, \psi_i \geq 0, \forall i = \overline{1, p}; \end{aligned}$$

– Supposons que $\max \Theta = \theta^* \neq 0$ alors $\exists i \in \{1, \dots, p\}$ tel que $\psi_i > 0$;

$$\begin{aligned} &\Rightarrow (c_i x + \alpha_i) - Z_i(x^*)(d_i x + \beta_i) > 0; \\ &\Rightarrow (c_i x + \alpha_i) > Z_i(x^*)(d_i x + \beta_i); \\ &\Rightarrow \frac{c_i x + \alpha_i}{d_i x + \beta_i} > Z_i(x^*); \\ &\Rightarrow Z_i(x) > Z_i(x^*); \end{aligned}$$

Ainsi, il existe $x \in D$ tel que $Z(x) \geq Z(x^*)$ et $Z(x) \neq Z(x^*)$. Ce qui est en contradiction avec l'hypothèse selon laquelle x^* est efficace.

– Supposons que $\theta^* = 0$ et $x^* \notin (EF)$ alors $\exists x \in D$ tel que :

$$\begin{aligned} Z(x) \geq Z(x^*) \text{ et } Z(x) \neq Z(x^*) &\Rightarrow \exists i \in \{1, \dots, p\}, Z_i(x) = \frac{c_i x + \alpha_i}{d_i x + \beta_i} > Z_i(x^*); \\ &\Rightarrow (c_i x + \alpha_i) > Z_i(x^*)(d_i x + \beta_i); \\ &\Rightarrow (c_i x + \alpha_i) - Z_i(x^*)(d_i x + \beta_i) > 0; \\ &\Rightarrow \psi_i > 0. \end{aligned}$$

Ce qui est en contradiction avec l'hypothèse $\theta^* = 0$. □

Solution équivalente

Pour optimiser le critère principal sur les solutions efficaces équivalentes à une solution efficace trouvée, soit \hat{x} , le problème suivant est résolu dans chaque itération de l'algorithme.

$$Q(\hat{x}) \begin{cases} \max & \varphi(x) = \frac{px + \alpha}{qx + \beta} \\ \text{s.c.} & Z_i(x) = Z_i(\hat{x}), i \in \{1, \dots, p\}. \\ & x \in D. \end{cases}$$

On note par G l'ensemble des solutions efficaces générées par l'algorithme.

6.3.2 Description de la méthode

La méthode proposée est basée principalement sur deux nouvelles techniques, la réduction du domaine d'admissibilité en ajoutant progressivement des contraintes linéaires à variables mixtes, qui sont inspirées par les coupes de Sylva et Crema [68] en résolvant le problème (FP_l) à chaque itération l

$$(FP_l) : \left\{ \max \varphi(x) = \frac{cx + \alpha}{dx + \beta} \mid x \in D_l \right\};$$

où

$$D_l = D_{l-1} \cap \left\{ x \in D \left| \begin{array}{l} (c_i x + \alpha_i) \geq (\tilde{Z}_i y_i^l + M_i(1 - y_i^l))L_i, i = \overline{1, p}; \quad (*) \\ \sum_{i=1}^p y_i^l \geq 1 \quad (**) \\ y_i^l \in \{0, 1\}, i = \overline{1, p}. \end{array} \right. \right\}$$

Avec $\tilde{Z}_i = Z_i(\bar{x}^{l-1}) + 1; i = \overline{1, p}$ et $(\bar{x}^1, \bar{x}^2, \dots, \bar{x}^{l-1})$ sont les solutions efficaces obtenues jusqu'à l'itération l .

M_i et L_i sont la borne inférieure et le minimum du dénominateur du $i^{\text{ème}}$ critère respectivement.

De plus, afin de tester l'efficacité d'une solution réalisable trouvée, le test d'efficacité équivalent $FT(x^*)$ est utilisé en explorant la linéarisation des critères fractionnaires.

Dans chaque itération de l'algorithme, la borne supérieure du critère principal, notée φ_{sup} , est mise à jour par sa valeur en la solution optimale obtenue dans le domaine réduit et la borne inférieure, notée φ_{inf} , est mise à jour par la meilleure valeur de ce critère dans les solutions efficaces générées. L'algorithme procède ainsi et prend fin, lorsque le domaine d'admissibilité devient vide ou lorsqu'on ne peut pas améliorer la valeur du critère principal ou lorsque la borne inférieure coïncide avec la borne supérieure.

Démonstration de la contrainte (*)

$$\begin{aligned}
 (*) \implies & \begin{cases} \frac{c_i x + \alpha_i}{d_i x + \beta_i} \geq \tilde{Z}_i \implies (c_i x + \alpha_i) - \tilde{Z}_i(d_i x + \beta_i) \geq 0 & \dots\dots (1) \\ \text{ou} \\ \frac{c_i x + \alpha_i}{d_i x + \beta_i} \geq M_i \implies (c_i x + \alpha_i) - M_i(d_i x + \beta_i) \geq 0 & \dots\dots (2) \end{cases} \\
 \implies & [(c_i x + \alpha_i) - \tilde{Z}_i(d_i x + \beta_i)]y_i + [(c_i x + \alpha_i) - M_i(d_i x + \beta_i)](1 - y_i) \geq 0; \\
 \implies & (c_i x + \alpha_i) - \tilde{Z}_i y_i(d_i x + \beta_i) - M_i(1 - y_i)(d_i x + \beta_i) \geq 0; \\
 \implies & (c_i x + \alpha_i) \geq (\tilde{Z}_i y_i + M_i(1 - y_i))(d_i x + \beta_i); \\
 \implies & (c_i x + \alpha_i) \geq (\tilde{Z}_i y_i + M_i(1 - y_i))L_i
 \end{aligned}$$

où $L_i = \min\{(d_i x + \beta_i), x \in D\}$ et $M_i = \{\min Z_i(x) = \frac{c_i x + \alpha_i}{d_i x + \beta_i} \mid x \in D\}$. pour tout $i \in \{1, \dots, p\}$.

La contrainte (**) impose une amélioration d'au moins d'un critère.

Proposition 6.1 L'algorithme proposé converge dans un nombre fini d'itérations.

Preuve Comme la région d'admissibilité D est supposée non vide et borné, D possède un nombre limité de solutions réalisables entières. Ainsi, le nombre de solutions efficaces $|(EF)|$ est fini. Dans chaque itération, l'algorithme génère une nouvelle solution efficace améliorant la valeur de φ et le domaine se réduit progressivement jusqu'à ce qu'il devient vide. Ainsi, la procédure converge vers une solution optimale dans un nombre fini d'itérations. \square

Algorithm 9: Optimisation sur l'ensemble efficace d'un (MOILFP)

Entrées:

- ↓ $A_{(m \times n)}$: matrice des contraintes ;
- ↓ $b_{(m \times 1)}$: vecteur second membre ;
- ↓ $c_i(1 \times n), \alpha_i$: vecteur numérateur du critère i ;
- ↓ $d_i(1 \times n), \beta_i$: vecteur dénominateur du critère i ;
- ↓ $c_{(1 \times n)}, \alpha$: vecteur numérateur du critère principal ;
- ↓ $d_{(1 \times n)}, \beta$: vecteur dénominateur du critère principal ;

Sorties:

- ↑ x_{opt} : Solution optimale du problème (FP_E).
- ↑ φ_{opt} : valeur optimale du critère principal φ .

Initialisation

$\varphi_{inf} = -\infty, \varphi_{sup} = +\infty, l = 0, D_0 = D, G = \emptyset$

pour $i = 1, \dots, p$ **faire**

Calculer

- $M_i = \{\min Z_i(x) = \frac{c_i x + \alpha_i}{d_i x + \beta_i} \mid x \in D\}$.
- $L_i = \{\min(d_i x + \beta_i) \mid x \in D\}$.

(**Étape 1.**) Résoudre (FP_R) $\equiv \{\max \varphi(x) = \frac{cx + \alpha}{dx + \beta} \mid x \in D\}$.

SI (FP_R) est irréalisable **ALORS**

| **Terminer**, (FP_E) est irréalisable.

SINON

| soit x^0 une solution optimale de (FP_R), poser $\varphi_{sup} = \varphi(x^0)$.

(**Étape 2.**) Tester d'efficacité de x^0 (résoudre $FT(x^0)$)

SI $x^0 \in (EF)$ **ALORS**

| **Terminer**, $x_{opt} = x^0, \varphi_{opt} = \varphi(x^0)$.

SINON

| soit \hat{x}^0 une solution optimale de $FT(x^0)$.

(**Étape 3.**) Résoudre $Q(\hat{x}^l)$, Soit \bar{x}^l une solution optimale,

SI $\varphi(\bar{x}^l) > \varphi_{inf}$ **ALORS** poser $\varphi_{inf} = \varphi(\bar{x}^l), x_{opt} = \bar{x}^l, \varphi_{opt} = \varphi(\bar{x}^l)$ et $G = G \cup \{\bar{x}^l\}$.

SI $\varphi_{inf} = \varphi_{sup}$ **ALORS Terminer** x_{opt} est une solution optimale de (FP_E).

(**Étape 4.**) $l := l + 1$, résoudre le problème (FP_l)

SI (FP_l) est irréalisable **ALORS**

| **Terminer** x_{opt} est une solution optimale de (FP_E).

SINON

| soit x^l une solution optimale de (FP_l), poser $\varphi_{sup} = \varphi(x^l)$

| **SI** $\varphi(x^l) < \varphi_{opt}$ **ALORS**

| **Terminer** x_{opt} est une solution optimale de (FP_E).

| **SINON**

| Tester l'efficacité de x^l (résoudre $FT(x^l)$)

| **SI** $x^l \in (EF)$ **ALORS**

| **Terminer** $x_{opt} = x^l, \varphi_{opt} = \varphi(x^l)$

| **SINON**

| soit \hat{x}^l une solution optimale de $FT(x^l)$, aller à (Étape 3.)

6.3.3 Illustration numérique

Nous présentons le déroulement de notre algorithme sur trois exemples numériques, nous commençons par l'exemple présenté auparavant dans la méthode de Zerdani et Moulaï.

Exemple 1 : Rappelons que le problème (MOILFP) est défini par

$$(P) \begin{cases} \max & Z_1 = \frac{-x_1+4}{x_2+1} \\ \max & Z_2 = \frac{x_1+1}{x_1+2} \\ \max & Z_3 = \frac{x_1-1}{x_1+1} \\ D & \begin{cases} 4x_1 + 3x_2 \leq 20 \\ x_1 - x_2 \leq 3 \\ x_2 \leq 4 \\ x_1, x_2 \in \mathbb{N} \end{cases} \end{cases}$$

Le problème principal (FP_E) est

$$(FP_E) \begin{cases} \max & \varphi(x) = \frac{5x_1+3x_2-4}{2x_1+x_2+1} \\ s.c. & x_1, x_2 \in (EF). \end{cases}$$

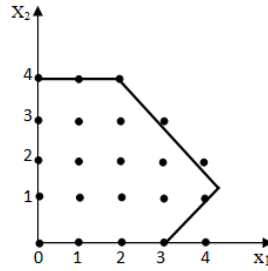


FIGURE 6.1 – Le domaine d'admissibilité D

Étape 1. : $M = (0; \frac{1}{2}; -1)$; $L = (1; 2; 1)$.

$$\varphi_{inf} = -\infty; \varphi_{sup} = +\infty; G = \emptyset, l = 0.$$

Étape 2. Résoudre le problème (FP_R) = {max $\varphi(x)$, $x \in D$ }.

$$x^0 = (2, 4), Z(x^0) = (\frac{2}{5}, \frac{3}{4}, \frac{1}{3}), \varphi(x^0) = 2 < \varphi_{sup} \text{ donc } \varphi_{sup} = 2.$$

Étape 3. Nous testons l'efficacité de x^0 par la résolution de $FT(x^0)$

$$FT(x^0) \begin{cases} \max & \Theta = \psi_1 + \psi_2 + \psi_3 \\ s.c. & x \in D \\ & (-x_1 + 4) - \frac{2}{5}(x_2 + 1) = \psi_1; \\ & (x_1 + 1) - \frac{3}{4}(x_1 + 2) = \psi_2; \\ & (x_1 - 1) - \frac{1}{3}(x_1 + 1) = \psi_3; \\ & x_1, x_2 \in \mathbb{N}, \psi_{i,i=\overline{1,3}} \geq 0 \end{cases}$$

$$\Theta^* \neq 0 \Rightarrow x^0 \notin (EF) \text{ et } \hat{x}^0 = (2, 0); Z(\hat{x}^0) = (2, \frac{3}{4}, \frac{1}{3}).$$

Étape 4. Résoudre le problème $Q(x^0)$

$$Q(x^0) \left\{ \begin{array}{l} \max \quad \varphi(x) = \frac{5x_1+3x_2-4}{2x_1+x_2+1} \\ \text{s.c.} \quad x \in D. \\ \quad \quad \frac{-x_1+4}{x_2+1} = 2 \\ \quad \quad \frac{x_1+1}{x_1+2} = \frac{3}{4} \\ \quad \quad \frac{x_1-1}{x_1+1} = \frac{1}{3} \end{array} \right.$$

$$\bar{x}^0 = (2, 0), \varphi(\bar{x}^0) = 1, 2 > \varphi_{inf} \Rightarrow \varphi_{inf} = 1, 2, G = \{(2, 0)\}.$$

Étape 5. Poser $l = l + 1 = 1$, résoudre le problème (FP_1)

$$(FP_1) \left\{ \begin{array}{l} \max \quad \varphi(x) = \frac{(5x_1+3x_2-4)}{(2x_1+x_2+1)} \\ \text{s.c.} \quad x \in D \\ \quad \quad (-x_1 + 4) \geq (3y_1^1) \times 1; \\ \quad \quad (x_1 + 1) \geq (\frac{7}{4}y_2^1 + \frac{1}{2}(1 - y_2^1)) \times 2; \\ \quad \quad (x_1 - 1) \geq (\frac{4}{3}y_3^1 - (1 - y_3^1)) \times 1; \\ \quad \quad y_1^1 + y_2^1 + y_3^1 \geq 1. \\ \quad \quad x_1, x_2 \in \mathbb{N}, (y_1^1, y_2^1, y_3^1) \in \{0, 1\}^3. \end{array} \right.$$

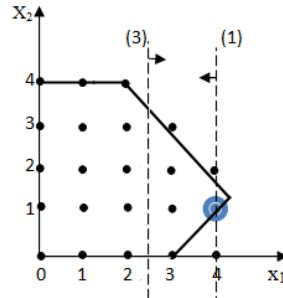


FIGURE 6.2 – Le domaine réduit D_1

$$x^1 = (4, 1), Y = (0.1.0) \text{ et } \varphi(x^1) = 1.9$$

Étape 6. x^1 est testée pour l'efficacité, on trouve $x^1 \in (EF)$ donc l'algorithme se termine avec $x_{opt} = (4, 1)$, $\varphi_{opt} = 1.9$ et $G = \{(2, 0); (4, 1)\}$.

Nous remarquons que notre méthode a résolu le problème en passant par deux point efficaces alors que la méthode de Zerdani et Moulaï a résolu le problème en visitant trois point efficaces.

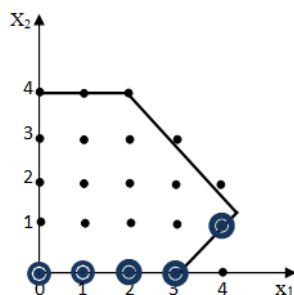


FIGURE 6.3 – Les solutions efficaces du problème (P)

Exemple 2 : Considérons le problème (MOILFP) suivant

$$(P) \begin{cases} \max & Z_1 = \frac{-x_1 - x_2}{x_1 + 2} \\ \max & Z_2 = \frac{x_1 - x_2 + 2}{x_1 + 1} \\ \max & Z_3 = x_1 + x_2 \\ D & \begin{cases} x_1 + x_2 \leq 7 \\ x_1 \leq 5 \\ x_2 \leq 3 \\ x_1, x_2 \in \mathbb{N} \end{cases} \end{cases}$$

Le problème principal (FP_E) est

$$(FP_E) \begin{cases} \max & \varphi(x) = \frac{6x_1 + 4x_2 - 5}{2x_1 + x_2 + 1} \\ s.c. & x_1, x_2 \in EF. \end{cases}$$

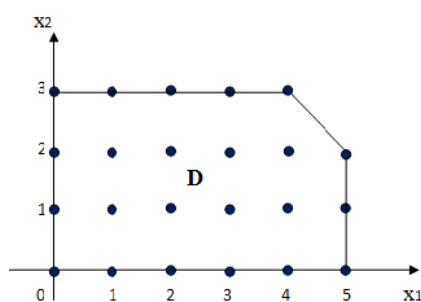


FIGURE 6.4 – Le domaine d'admissibilité D

Étape 1. : $M = (\frac{-3}{2}; -1; 0)$; $L = (2; 1; 1)$.

$$\varphi_{inf} = -\infty; \varphi_{sup} = +\infty; G = \emptyset, l = 0.$$

Étape 2. Résoudre le problème $(FP_R) = \{\max \varphi(x), x \in D\}$.

$$x^0 = (4, 3), Z(x^0) = (\frac{-7}{6}, \frac{3}{5}, 7) \text{ et } \varphi(x^0) = 2.58 < \varphi_{sup} \text{ donc } \varphi_{sup} = 2.58.$$

Étape 3. Nous testons l'efficacité de x^0 en résolvant le problème $FT(x^0)$

$$FT(x^0) \left\{ \begin{array}{l} \max \quad \Theta = \psi_1 + \psi_2 + \psi_3 \\ \text{s.c.} \quad x \in D \\ \quad \quad (-x_1 - x_2) - \left(\frac{-7}{6}\right)(x_1 + 2) = \psi_1; \\ \quad \quad (x_1 - x_2 + 2) - \frac{3}{5}(x_1 + 1) = \psi_2; \\ \quad \quad (x_1 + x_2) - 7 = \psi_3; \\ \quad \quad x_1, x_2 \in \mathbb{N}, \psi_{i,i=1,3} \geq 0 \end{array} \right.$$

$$\Theta^* \neq 0 \Rightarrow x^0 \notin (EF) \text{ et } \hat{x}^0 = (5, 2); Z(\hat{x}^0) = \left(-1, \frac{5}{6}, 7\right).$$

Étape 4. Résoudre le problème $Q(x^0)$

$$Q(x^0) \left\{ \begin{array}{l} \max \quad \varphi(x) = \frac{6x_1 + 4x_2 - 5}{2x_1 + x_2 + 1} \\ \text{s.c.} \quad x \in D. \\ \quad \quad \frac{-x_1 - x_2}{x_1 + 2} = -1; \\ \quad \quad \frac{x_1 - x_2 + 2}{x_1 + 1} = \frac{5}{6}; \\ \quad \quad x_1 + x_2 = 7; \end{array} \right.$$

$$\bar{x}^0 = (5, 2), \varphi(\bar{x}^0) = 2.54 > \varphi_{inf} \Rightarrow \varphi_{inf} = 2.54, G = \{(5, 2)\}.$$

Étape 5. Poser $l = l + 1 = 1$, résoudre le problème (FP_1)

$$(FP_1) \left\{ \begin{array}{l} \max \quad \varphi(x) = \frac{6x_1 + 4x_2 - 5}{2x_1 + x_2 + 1} \\ \text{s.c.} \quad x \in D \\ \quad \quad -x_1 - x_2 \geq \left[\frac{-3}{2}(1 - y_1^1)\right] \times 2; \\ \quad \quad x_1 - x_2 + 2 \geq \left[\frac{11}{6}y_2^1 - (1 - y_2^1)\right] \times 1; \\ \quad \quad x_1 + x_2 \geq 8y_3^1; \\ \quad \quad y_1^1 + y_2^1 + y_3^1 \geq 1. \\ \quad \quad x_1, x_2 \in \mathbb{N}, (y_1^1, y_2^1, y_3^1) \in \{0, 1\}^3. \end{array} \right.$$

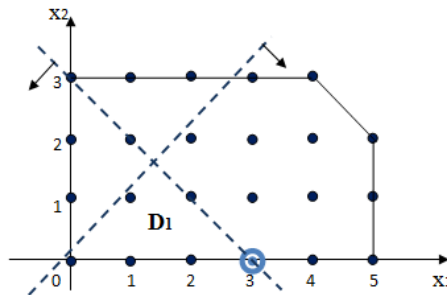


FIGURE 6.5 – Le domaine réduit D_1

$x^1 = (3, 0), Y = (0.1.0)$ et $\varphi(x^1) = 1.86 < \varphi_{inf}$ donc l'algorithme s'arrête avec $x_{opt} = (5, 2), \varphi_{opt} = 2.54$ et $G = \{(5, 2); (3, 0)\}$.

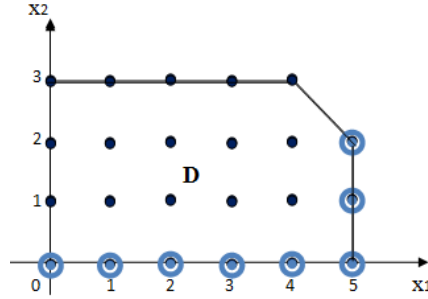


FIGURE 6.6 – Les solutions efficaces du problème (P)

Exemple 3 : Considérons le problème (MOILFP) proposé par Kornbluth et Steuer [48]

$$(P) \begin{cases} \max & Z_1 = \frac{x_1 - 4}{-x_2 + 3} \\ \max & Z_2 = \frac{-x_1 + 4}{x_2 + 1} \\ \max & Z_3 = -x_1 + x_2 \\ D & \begin{cases} -x_1 + 4x_2 \leq 0 \\ 2x_1 - x_2 \leq 8 \\ x_1, x_2 \in N \end{cases} \end{cases}$$

le problème principal (FP_E) est donné par

$$(FP_E) \begin{cases} \max & \varphi(x) = \frac{-5x_1 + 2x_2 - 5}{2x_1 - 2x_2 + 1} \\ s.c. & x_1, x_2 \in EF. \end{cases}$$

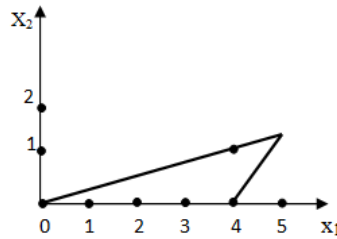


FIGURE 6.7 – Le domaine d'admissibilité D

Étape 1. : $M = (\frac{-4}{3}; 0; -4)$; $L = (2; 1; 1)$.

$$\varphi_{inf} = -\infty; \varphi_{sup} = +\infty; G = \emptyset, l = 0.$$

Étape 2. Résoudre le problème $(FP_R) = \{\max \varphi(x), x \in D\}$.

$$x^0 = (4, 0), \varphi(x^0) = -2.778, Z(x^0) = (0, 0, -4), \text{ poser } \varphi_{sup} = -2.778.$$

Étape 3. Tester l'efficacité de x^0 en résolvant le problème $FT(x^0)$

$$FT(x^0) \left\{ \begin{array}{l} \max \quad \Theta = \psi_1 + h_2 + \psi_3 \\ \text{s.c.} \quad x \in D \\ \quad (x_1 - 4) - 0 \times (-x_2 + 3) = \psi_1; \\ \quad (-x_1 + 4) - 0 \times (x_2 + 1) = \psi_2; \\ \quad (-x_1 + x_2) - (-4) = \psi_3; \\ \quad x_1, x_2 \in \mathbb{N}, \psi_{i, i=\overline{1,3}} \geq 0 \end{array} \right.$$

$\Theta^* \neq 0 \Rightarrow x^0 \notin (EF)$ et $\hat{x}^0 = (4, 1); Z(\hat{x}^0) = (0, 0, -3)$.

Étape 4. Résoudre le problème $Q(x^0)$

$$Q(x^0) \left\{ \begin{array}{l} \max \quad \varphi(x) = \frac{-5x_1 + 2x_2 - 5}{2x_1 - 2x_2 + 1} \\ \text{s.c.} \quad x \in D. \\ \quad \frac{x_1 - 4}{-x_2 + 3} = 0 \\ \quad \frac{-x_1 + 4}{x_2 + 1} = 0 \\ \quad -x_1 + x_2 = -3 \end{array} \right.$$

$\bar{x}^0 = (4, 1), \varphi(\bar{x}^0) = -3.285 > \varphi_{inf} \Rightarrow \varphi_{inf} = -3.285; x_{opt} = (4, 1);$
 $G = \{(4, 1)\}$.

Étape 5. $l = l + 1 = 1$ Résoudre le problème (FP_1)

$$(FP_1) \left\{ \begin{array}{l} \max \quad \varphi(x) = \frac{(-5x_1 + 2x_2 - 5)}{(2x_1 - 2x_2 + 1)} \\ \text{s.c.} \quad x \in D \\ \quad (x_1 - 4) \geq (y_1^1 - \frac{4}{3}(1 - y_1^1)) \times 2; \\ \quad (-x_1 + 4) \geq y_2^1; \\ \quad (-x_1 + x_2) \geq (-2y_3^1 - 4(1 - y_3^1)) \times 1; \\ \quad y_1^1 + y_2^1 + y_3^1 \geq 1. \\ \quad x_1, x_2 \in \mathbb{N}, (y_1^1, y_2^1, y_3^1) \in \{0, 1\}^3. \end{array} \right.$$

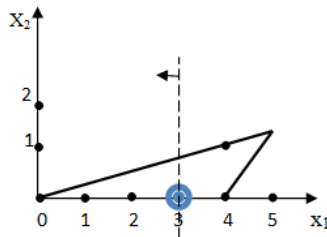


FIGURE 6.8 – Le domaine réduit D_1

$x^1 = (3, 0), Y = (0.1.0)$ et $\varphi(x^1) = -2.857 > \varphi_{inf}$.

Étape 6. Tester l'efficacité de x^1

$$FT(x^1) \begin{cases} \max & \Theta = \psi_1 + h_2 + \psi_3 \\ \text{s.c.} & x \in D \\ & (x_1 - 4) - \left(\frac{-1}{3}\right) \times (-x_2 + 3) = \psi_1; \\ & (-x_1 + 4) - 1 \times (x_2 + 1) = \psi_2; \\ & (-x_1 + x_2) - (-3) \times 1 = \psi_3; \\ & x_1, x_2 \in \mathbb{N}, \psi_{i,i=\overline{1,3}} \geq 0 \end{cases}$$

$\Theta^* \neq 0 \Rightarrow x^1 \notin (EF)$ donc l'algorithme s'arrête avec $x_{opt} = (3, 0)$ et $\varphi_{opt} = -2.857$ et $G = \{(4, 1); (3, 0)\}$.

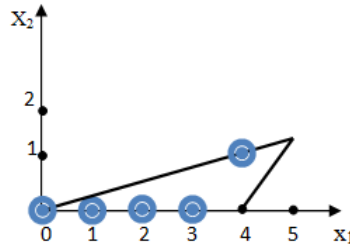


FIGURE 6.9 – L'ensemble des solutions efficaces

6.3.4 Expérimentations et résultats

L'algorithme proposé a été programmé sous l'environnement *MATLAB* et exécuté sur un PC Intel(R)Core(TM) i3 CPU 2.13 GHZ. Pour mesurer la performance de l'algorithme, au total de 270 problèmes ont été aléatoirement générés selon la procédure suivante : les composantes de la matrice A et le vecteur b sont générées selon les distributions uniformes discrètes $[1, 30]$ et $[50, 300]$ respectivement, Les vecteurs $c_i, d_i, i = \overline{1, p}$ des fonctions objectifs ainsi que les vecteurs c, d du critère principal sont générés selon la distribution uniforme discrète $[-10, 10]$. Les constantes α_i et α sont générées selon la même distribution

Pour assurer que tous les dénominateurs sont strictement positifs dans les domaines d'admissibilité, les constantes $\beta_i, i = \overline{1, p}$ et β sont générées telles que $d_i x + \beta_i > 0, i = \overline{1, p}$ et $d x + \beta > 0$ respectivement.

Les instances traitées ont été regroupées en 27 catégories selon le nombre de variables, de contraintes et de fonctions objectifs. Le nombre de fonctions objectifs p a été fixé à 2, 3 et 4.

Pour chaque instance, les résultats obtenus représentent la médiane du temps d'exécution et du nombre d'itération de dix exécutions indépendantes. En outre, les valeurs minimums et maximums de chaque mesure sont présentées en intervalles.

p $n \times m$	$p = 2$		$p = 3$		$p = 4$	
	cpu(sec)	# iter	cpu(sec)	# iter	cpu(sec)	# iter
3×3	0.134 [0.0118; 1.7255]	2 [1; 4]	0.179 [0.0173; 3.225]	2 [1; 4]	0.212 [0.0232; 7.4755]	2 [1; 3]
5×5	0.5929 [0.0867; 1.8831]	1.5 [1; 3]	1.0322 [0.049; 12.89]	1.5 [1; 3]	0.3419 [0.0259; 2.3347]	1 [1; 2]
10×5	12.1486 [2.4; 56]	2 [1; 4]	13.0297 [0.8452; 81.006]	1.5 [1; 4]	16 [0.3955; 178.217]	1 [1; 3]
15×5	17.344 [0.153; 150]	1 [1; 3]	17.776 [0.024; 715.32]	1 [1; 3]	19.146 [0.21; 107]	1 [1; 2]
15×10	35.025 [0.183; 127.85]	2 [1; 2]	34.5 [0.66; 3720]	1.5 [1; 4]	55 [0.46; 1250]	1 [1; 2]
20×10	55.89 [12; 532.06]	2 [1; 4]	38 [0.60; 486]	1 [1; 3]	32.2 [0.336; 675]	1 [1; 2]
25×10	147 [37.7; 6474]	1.5 [1; 4]	144 [46; 642]	1 [1; 2]	165 [19.44; 1045]	1 [1; 2]
25×15	183 [14.83; 6125]	1.5 [1; 4]	260 [21.08; 1712]	1 [1; 2]	220.45 [64; 2714]	1 [1; 2]
30×10	355.5 [105; 2487]	1 [1; 2]	428 [100; 3901]	1 [1; 3]	327 [87; 3270]	1 [1; 3]

TABLE 6.1 – Résultats des instances générées aléatoirement.

Nous constatons dans le tableau (6.1) que notre algorithme est efficace en terme du nombre des solutions efficaces générées au cours de l'exécution (égal au nombre d'itération effectuées) pour les problèmes de tailles réduites et de tailles moyennes. Concernant les problèmes de grandes tailles, nous remarquons que le temps d'exécution est significatif dans quelques instances traitées. En effet, les difficultés rencontrées dans la résolution de ces problème sont fortement liées à leurs dimensions, notamment le nombre des variables n et le nombre des objectifs p .

Dans l'étape initiale, le calcul des paramètres M_i et L_i pour $i \in \{1, \dots, p\}$ nécessite la résolution de p problèmes de programmation linéaire fractionnaire en nombre entiers et p problèmes de programmation linéaire en nombre entiers respectivement. De plus, à chaque itération réalisée, le problème relaxé s'accroît de $(p + 1)$ contraintes et de p variables binaires. Cependant, les coupes proposées ont montrées leurs efficacité dans la réduction du domaine d'admissibilité par le nombre réduit d'itérations effectuées même pour les problèmes de tailles relativement grandes.

7

Conclusion et perspectives

Les problèmes d'optimisation à objectifs multiples sont très variés et correspondent à des situations de décision très difficiles. Dans la résolution de ces problèmes, la notion d'optimalité disparaît au profit de la notion d'efficacité. Il s'agit de chercher un ensemble de solutions réalisant le meilleur compromis entre les critères considérés. L'ensemble de ces solutions peut avoir une cardinalité importante et l'énumération explicite de toutes ces solutions n'est pas toujours recommandée. Dans beaucoup de cas pratiques, le décideur se trouve face à un grand nombre de solutions efficaces dont la sélection de ses préférences s'avère difficile. L'optimisation d'un critère qui exprime les préférences du décideur sur l'ensemble de solutions efficaces est l'un des concepts importants et intéressants de la programmation multi-objectifs, et il devient un moyen fructueux pour aider le décideur à réaliser son choix et éviter une telle situation.

Le travail présenté dans cette thèse s'intéresse à l'étude du problème d'optimisation d'un critère non linéaire, particulièrement linéaire fractionnaire, sur un ensemble efficace discret. Dans ce contexte, nous avons présenté deux nouvelles méthodes exactes :

- La première méthode sert à résoudre le problème d'optimisation d'un critère linéaire fractionnaire sur l'ensemble efficace d'un problème de programmation linéaire multi-objectifs à variables discrètes (*MOILP*) sans avoir énumérer toutes les solutions efficaces, en combinant la réduction du domaine d'admissibilité avec un processus d'exploration des arêtes incidentes, en utilisant le vecteur gradient réduit du critère à optimiser. Nous avons aussi mené différents tests expérimentaux sur des instances générées aléatoirement pour évaluer la performance de l'algorithme proposé. Ce travail a fait l'objet d'une publication dans la revue International RAIRO

Operations Research [49].

- La deuxième méthode, comme une généralisation du premier problème, elle consiste à optimiser un critère linéaire fractionnaire sur l'ensemble efficace discret d'un problème multi-objectifs linéaire fractionnaire (*MOILFP*). La méthode proposée résout le problème dans l'espace des critères par la proposition d'un nouveau test d'efficacité pour les problèmes multi-objectifs linéaire fractionnaire, ainsi que l'utilisation de nouvelles coupes inspirées des coupes de Sylva et Crema pour réduire le domaine d'admissibilité progressivement.

L'algorithme proposé a été programmé et exécuté sous l'environnement *MATLAB*. Dans le but de tester la performance de l'algorithme, une série d'exécutions est réalisée sur des problèmes de différentes tailles aléatoirement générés. Les résultats obtenus témoignent que notre algorithme est efficace en terme du nombre de solutions efficaces parcourues. Vu l'aspect discret et le caractère multi-objectifs non linéaire (particulièrement fractionnaire) des problèmes, le temps d'exécution est significatif dans quelques instances de grandes tailles.

Ce travail ouvre des nombreuses perspectives de recherche, parmi lesquelles nous proposons les suivantes :

- En premier lieu, une étude comparative sur la méthode proposée et la méthode de Zerdani et Moulaï peut être considérée.
- Le problème d'optimisation d'un critère non linéaire, particulièrement fractionnaire, sur l'ensemble efficace d'un problème multi-objectifs stochastique est ouvert.
- L'application des méthodes proposées sur des problèmes réels peut être un axe de recherche pratique intéressant.
- L'intégration des méta-heuristiques dans l'optimisation d'un critère de compromis sur un ensemble efficace mérite d'être réalisé afin de traiter des problèmes de grandes tailles.

Bibliographie

- [1] M. Abbas and D. Chaabane. An algorithm for solving multiple objective integer linear programming problem. *RAIRO Operations Research*, Vol. 36, pp. 351-364, 2002.
- [2] M. Abbas and D. Chaabane. Optimizing a linear function over an integer efficient set. *European Journal of Operational Research*, Vol. 174, pp. 1140-1161, 2006.
- [3] M. Abbas and M. Moulai. Solving multiple objective integer linear programming. *Journal of the Italian Operations Research Society (Ricerca Operativa)*, Vol. 29, pp. 15-38, 1999.
- [4] M. Abbas and M. Moulai. Integer linear fractional programming with multiple objective. *Ricerca Operativa Journal of the Italian Operations Research Society*, Vol. 32, n° 103-104, pp. 15-38, 2002.
- [5] H.P. Benson. Optimization over the efficient set. *Journal of Mathematical Analysis and Applications*, Vol. 98, pp. 562-580, 1984.
- [6] H.P. Benson. An all-linear programming relaxation algorithm for optimizing over the efficient set. *Journal of Global Optimization*, Vol. 1, pp. 83-104, 1991.
- [7] H.P. Benson. A finite nonadjacent extreme-point search algorithm for optimization over the efficient set. *Journal of Optimization Theory and Applications*, Vol. 73, pp. 47-64, 1992.
- [8] H.P. Benson and S. Sayin. Optimization over the efficient set : Four special cases. *Journal of Optimization Theory and Applications*, Vol. 80, pp. 1-16, 1994.
- [9] G.R. Bitran and G. Novaes. Linear programming with a fractional objective function. *Operations Research*, Vol. 24, pp. 675-699, 1976.
- [10] S. Bolintineanu. Minimization of a quasiconcave function over an efficient set. *Mathematical Programming*, Vol. 61, pp. 89-110, 1993.
- [11] V.J. Bowman. On the relationship of the tchebytcheff norm and the efficient frontier of multiple-criteria objectives. In : Thiriez H.(ed.). *Multiple Criteria Decision Making*, Springer-Verlag, Berlin, pp. 248-258, 1976.
- [12] A. Cambini and L. Martein. A modified version of Martos's algorithm for the linear fractional problem. *Mathematics of Operations Research*, Vol. 53, pp. 33-44, 1986.

- [13] A. Cambini and L. Martein. Equivalence in linear fractional programming. *Optimization*, Vol. 23, pp. 41-51, 1992.
- [14] A. Cambini, L. Martein and I.M. Stancu-Minasian. A survey of bicriteria fractional problems. *Advanced Modeling and Optimization*, Vol. 1, n°1, pp. 9-46, 1999.
- [15] D. Chaabane. Contribution à l'optimisation multicritère en variables discrètes. PhD thesis, Polytechnique Faculty of Mons, Belgium, 2007.
- [16] D. Chaabane, B. Brahmi and Z. Ramdani. The augmented weighted tchebychev norm for optimizing a linear function over an integer efficient set of a multicriteria linear program. *Journal of International Transactions in Operational Research*, pp. 1-15, 2012.
- [17] D. Chaabane and M. Pirlot. A method for optimizing over the integer efficient set. *Journal of industrial and management optimization*, Vol. 6, n°4, pp. 811-823, 2010.
- [18] M.E.A. Chergui and M. Moulai. An exact method for a discrete multiobjective linear fractional optimization. *Journal of Applied Mathematics and Decision Sciences*, Vol. 2008, Article ID 760191, 12 pages, doi : 10.1155/2008/760191, 2008.
- [19] A. Charnes and W.W. Cooper. Programming with linear fractional functionals. *Naval Research Logistics Quarterly*, Vol. 9, pp. 181-186, 1962.
- [20] Y. Collette et P. Siarry. *Optimisation multiobjectif*. Editions Eyrolles, 2002.
- [21] J.P. Costa. Computing non-dominated solutions in MOLFP. *European Journal of Operational Research*, Vol. 181, n°3, pp. 1464-1475, 2007.
- [22] G.B. Dantzig. On a linear combinatorial approach to the travelling salesmen problem. *Journal of Operations Research*, Vol. 7, pp. 58-66, 1959.
- [23] K. Deb. *Multi-objective optimization using evolutionary algorithms*. New-York : John Wiley, 2001.
- [24] W. Dinkelbach. On nonlinear fractional programming. *Management Science*, Vol. 13, pp. 492-498, 1967.
- [25] J.G. Ecker and J.H. Song. Optimizing a linear function over an efficient set. *Journal of Optimization Theory and Applications*, Vol. 83, pp. 541-563, 1994.
- [26] J.G. Ecker and I.A. Kouada. Finding efficient points for multi-objective linear programs. *Mathematical Programming*, Vol. 8, pp. 375-377, 1975.
- [27] M. Ehrgott, H.W. Hamacher, K. Klamroth, S. Nickel, A. Schobel and M.M. Wiecek. A note on the equivalence of balance points and Pareto solutions in multiple-objective programming. *Journal of Optimization Theory and Applications*, Vol. 92, n°1, pp. 209-212, 1997.
- [28] M. Ehrgott and X. Gandibleux. A survey and annotated bibliography of multiobjective combinatorial optimization. *OR Spektrum*, Vol. 22, pp. 425-460, 2000.

- [29] M. Ehrgott. Multicriteria Optimization. Lecture Notes in Economics and Mathematical Systems, numéro 491. Springer Verlag, Berlin, Heidelberg, 2000.
- [30] P.K. Eswaran, A. Ravindran and H. Moskowitz. Algorithms for nonlinear integer bicriterion problems. Journal of Optimization Theory and Applications, Vol. 63, n°2, 1989.
- [31] J.E. Falk and S.W. Paloschay. Optimizing the sum of linear fractional functions. Advances in Global Optimisation, Kluwer Academic Publishers, pp. 221-258, 1992.
- [32] R.W. Freund and F. Jarre. Solving the sum-of-ratios problem by an interior-point method. Journal of Global Optimization, Vol. 19, pp. 83-102, 2001.
- [33] J. Fülöp. A cutting plane algorithm for linear optimization over the efficient set. Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, Berlin, Vol. 405, pp. 374-385, 1994.
- [34] A.M. Geoffrion. Proper efficiency and the theory of vector maximization. Journal of Mathematical Analysis and Applications, Vol. 22, pp. 618-630, 1968.
- [35] D.E. Golberg. Genetic algorithms in search, optimization and machine learning. Addison-Wesley Publishing Company. Reading, Massachussets, 1989.
- [36] R.E. Gomory. Outline of an algorithm for integer solutions to linear programs. Bulletin of the AMS, pp. 275-278, 1963.
- [37] D. Granot and F. Granot. On integer and mixed integer fractional programming problems. Annals of Discrete Mathematics, Vol. 1, pp. 221-231, 1977.
- [38] M. Grunspan. Fractional programming : A survey. Technical Report 50. Department of Industrial and Systems Engineering, University of Florida, 1971.
- [39] R. Gupta and R. Malhotra. Multi-criteria integer linear programming problem. Cahiers de CERO 34, pp. 51-68, 1992.
- [40] R. Gupta and R. Malhotra. Multi-criteria integer linear fractional programming problem. Optimization, Vol. 35, pp. 373-389, 1995.
- [41] S.T. Hackman and U. Passy. Maximizing a linear fractional function on a pareto efficient frontier. Journal of Optimization Theory and Applications, Vol. 113, n°1, pp. 83-103, 2002.
- [42] P. Hansen, M.V. Poggi De Aragao and C.C. Ribeiro. Hyperbolic 0-1 programming and information retrieval. Mathematical Programming, Vol. 52, pp. 255-263, 1991.
- [43] J.R. Isbell and W.H. Marlow. Attribution games. Naval Research Logistics Quarterly, Vol. 3, pp. 71-94, 1956.
- [44] H. Isermann and E. Steuer. Computational experience concerning payoff tables and minimum values over the efficient set. European Journal of Operational Research , Vol. 33, pp. 91-97, 1987.
- [45] H. Isermann. Proper efficiency and the linear vector maximization problem. Operations Research, Vol. 22, pp. 189-191, 1974.

- [46] J.M. Jorge. An algorithm for optimizing a linear function over an integer efficient set. *European Journal of Operational Research*, Vol. 195, pp. 98-103, 2009.
- [47] D. Klein and E. Hannan. An algorithm for multiple objective integer linear programming problem. *European Journal of Operational Research*, Vol. 9, pp. 378-385, 1982.
- [48] J.S.H. Kornbluth and R.E. Steuer. Multiple objective linear fractional programming. *Management Science*, Vol. 27, pp. 1024-1039, 1981.
- [49] S. Mahdi and D. Chaabane. A linear fractional optimization over an integer efficient set. *RAIRO Operations Research*, Vol. 49, pp. 265-278, DOI : 10.1051/ro/2014036, 2015.
- [50] S. Mahdi and D. Chaabane. An exact method for optimizing a linear fractional function over the integer efficient set. 2nd International Symposium on operational Research (ISOR 2011), Algiers, Algeria, 30 May-02 June, 2011.
- [51] S. Mahdi and D. Chaabane. Optimization over an efficient set with fractional criteria. 11th International Conferences on Multiple Objective Programming and Goal Programming (MOPGP 2015), Tlemcen, Algeria, 13-15 December, 2015.
- [52] B. Martos. Hyperbolic programming. *Naval Research Logistics Quarterly*, Vol. 11, pp. 135-155, 1964.
- [53] K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, 1999.
- [54] M. Moulai. *Optimisation multicritère fractionnaire linéaire en nombres entiers*. Thèse de doctorat. Université des Sciences et de la Technologie Houari Boumediene, Algérie, octobre 2002.
- [55] L.D. Muu and H.Q. Tuyen. Bilinear programming approach to optimization over the efficient set of a vector affine fractional problem. *Acta Mathematica Vietnamica*, Vol. 27, n°2, pp. 119-139, 2002.
- [56] N.C. Nguyen. An algorithm for optimizing a linear function over the integer efficient set. *Konrad-Zuse-Zentrum fur Informationstechnik Berlin*, Nov 1992.
- [57] J. Philip. Algorithms for the vector maximization problem. *Mathematical Programming*, Vol. 2, pp. 207-229, 1972.
- [58] T. Radzik. Fractional combinatorial optimization. *Handbook of Combinatorial Optimization*, Edited by Z.Z. Du and P. Pardalos, Kluwer Academic Publishers, pp. 429-478, 1998.
- [59] O.M. Saad and J.B. Hughes. Bicriterion integer linear fractional programs with parameters in the objective functions. *Journal of Information and optimization Sciences*, Vol. 19, n°1, pp. 97-108, 1998.
- [60] S. Sayin. Optimizing over the efficient set using a Top-Down search of faces. *Operations Research*, Vol. 48, pp. 65-72, 2000.

- [61] S. Schaible. Fractional programming. *Zeitschrift für Operations Research*, Vol. 27, pp. 39-54, 1983.
- [62] S. Schaible. Fractional programming. In R. Horst and P.M. Pardalos(eds.), *Handbook of Global Optimization*, Kluwer Academic Publishers, pp. 495-608, 1995.
- [63] C.R. Seshan and V.G. Tikekar. Algorithms for integer fractional programming. *Journal of the Indian Institute of Science*, Vol. 62(B), pp. 9-16, 1980.
- [64] M. Shigeno, I. Takahashi and Y. Yamamoto. Minimum maximal flow problem -an optimization over the efficient set-. *Journal of Global Optimization*, Vol. 25, pp. 425-443, 2003.
- [65] I.M. Stancu-Minasian. *Fractional programming : Theory, methods and applications*. Kluwer Academic Publishers, 1997.
- [66] R. Steuer. *Multiple criteria optimization : Theory, computation and applications*. John Wiley & Sons, New-York, 1985.
- [67] R.E. Steuer and E.U. Choo. An interactive weighted tchebychev procedure for multiple objective programming. *Mathematical Programming*, Vol. 26, n°3, pp. 326-344, 1983.
- [68] J. Sylva and A. Crema. A method for finding the set of non dominated vectors for multiple objective integer programming. *European Journal of Operational Research*, Vol. 158, pp. 46-55, 2004.
- [69] J. Sylva and A. Crema. A method for finding well-dispersed subsets of non-dominated vectors for multiple objective mixed integer linear programs. *European Journal of Operational Research*, Vol. 180, pp. 1011-1027, 2007.
- [70] K. Swarup. Linear fractional functional programming. *Operations Research*, Vol. 13, n°6, pp. 1029-1036, 1965.
- [71] H.Q. Tuyen and L.D. Muu. Biconvex programming approach to optimization over the weakly efficient set of a multiple objective affine fractional problem. *Operations Research Letters*, Vol. 28(2), pp. 81-92, 2001.
- [72] E.L. Ulungu and J. Teghem. Multi-objective combinatorial optimization : a survey. *Journal of Multi-Criteria Decision Analysis*, Vol. 3, pp. 83-104, 1994.
- [73] O. Zerdani. and M. Moulai. Optimization over an integer efficient set of a multiple objective linear fractional problem. *Applied Mathematical Sciences*, Vol. 5, n°50, pp. 2451-2466, 2011.
- [74] O. Zerdani. *L'optimisation non linéaire multiobjectif*. Thèse de doctorat. Université Mouloud Mammeri, TIZI-OUZOU, Algérie, 2013.
- [75] Von Neumann. J "Uber ein" ökonomisches Gleichungssystem und eine Verallgemeinerung des Brouwerschen Fixpuntsatzes. In K. Menger, editor, *Ergebnisse eines mathematischen Kolloquiums (8)*, Leipzig und Wien, pp. 73-83, 1937.

- [76] H.P. Williams. Experiments in the formulation of integer programming problems. *Mathematical Programming Studies*, Vol. 2, pp. 180-197,2009.
- [77] L.A. Wolsey. *Integer programming*. Wiley-Interscience publication, New York, 1998.
- [78] Y. Yamamoto. Optimization over the efficient set : Overview. *Journal of Global Optimization*, Vol. 22, pp. 285-317, 2002.

Résumé

Les problèmes d'optimisation à objectifs multiples sont très variés et correspondent à des situations de décision difficiles. Dans la résolution de ces problèmes, la notion d'optimalité disparaît face à la notion d'efficacité. Il s'agit de chercher un ensemble de solutions réalisant le meilleur compromis entre les critères considérés. L'ensemble de ces solutions peut avoir une cardinalité importante et la prise de décision devient une tâche très difficile, ce qui motive le retour vers le décideur pour exprimer ses préférences comme une fonction à optimiser dans cet ensemble. Dans cette thèse, nous nous intéressons à l'optimisation d'un critère linéaire fractionnaire sur l'ensemble efficace d'un problème multi-objectifs en nombre entiers. Dans ce contexte, deux nouvelles méthodes exactes sont présentées ; la première méthode sert à résoudre le problème d'optimisation d'un critère linéaire fractionnaire sur l'ensemble efficace d'un problème *MOILP*. Dans la deuxième méthode, comme une généralisation du premier problème, nous considérons le problème d'optimisation d'un critère linéaire fractionnaire sur l'ensemble efficace du problème *MOILFP*. La méthode proposée résout le problème dans l'espace des critères par la proposition d'un nouveau test d'efficacité pour les problèmes *MOILFP*, ainsi que l'utilisation de nouvelles coupes, inspirée des coupes de Sylva et de Crema pour réduire le domaine d'admissibilité progressivement.

Mots clés : Optimisation multi-objectifs, la programmation linéaire fractionnaire, Optimisation sur un ensemble efficace.

Abstract

The multiple objective optimization problems are very varied and correspond to difficult situations of decision. In the resolution of these problems, the notion of optimality disappears face the notion of efficiency. It is a question of seeking a set of the best compromise solutions between the considered criteria. The set of these solutions can have a significant cardinality and the decision-making becomes a very difficult task, which justifies the return to the decision maker to express his preferences as a function to be optimized over this set.

In this thesis, we are interested in the optimization of a linear fractional criterion on the efficient set of an integer multiobjective problem. In this context, two new exact methods are presented ; the first one solves the problem of optimization of a linear fractional criterion on the efficient set of a *MOILP* problem. In the second method, as a generalization of the first problem, we consider the problem of optimizing a linear fractional criterion on the efficient set of *MOILFP* problem. The proposed method solves the problem in the outcome space by the suggestion of a new efficiency test for the *MOILFP* problems, as well as the use of new cuts, inspired from Sylva and Crema cuts. These cuts enabled us to reduce progressively the admissibility domain.

Keywords Multiple objective optimization, linear fractional programming, Integer programming, efficient set.