

N° d'ORDRE :03/2016-D/MT

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie Houari Boumediène
Faculté des Mathématiques



THESE

Présenté pour l'obtention du grade de Docteur en sciences

En : MATHÉMATIQUES

Spécialité : Recherche Opérationnelle : Mathématique de Gestion

Par : BRAHMI Boualem

Thème

Contribution dans l'optimisation non convexe

Soutenue publiquement le : 06/10/2016 devant le jury composé de :

BOUROUBI	Sadek	Professeur	U.S.T.H.B	Alger	Président
CHAABANE	Djamal	Professeur	U.S.T.H.B	Alger	Directeur de These
ABBAS	Moncef	Professeur	U.S.T.H.B	Alger	Examineur
AIDENE	Mohamed	Professeur	UMMTO	Tizi-Ouzou	Examineur
AIDER	Méziane	Professeur	U.S.T.H.B	Alger	Examineur
BELAHCENE	Fatima	MCA	UMMTO	Tizi-Ouzou	Examinatrice
MOULAI	Mustapha	Professeur	U.S.T.H.B	Alger	Examineur

Remerciements

Un très grand merci à mon directeur de thèse, Prof. CHAABANE Djamel, professeur à l'USTHB-Alger, pour m'avoir fortement encouragé à poursuivre dans cette voie et choisi pour ce sujet de thèse.

Merci à toutes les personnes avec qui j'ai travaillé avec un immense plaisir, Prof. Lakhdar Saïs, professeur à l'université d'Artois-Lens, Amine Belabed, doctorant à la faculté des science de Lens, et tous ceux avec qui j'ai pu être en contact dans le laboratoire CRIL de Lens-France. Grâce à eux, j'ai aussi pu rencontrer et travailler avec des chercheurs de différents pays, ce qui est un atout et une chance exceptionnelle, et je leur en suis particulièrement reconnaissant.

Je tiens bien évidemment à exprimer ma reconnaissance aux membres du jury :Prof. Moncef ABBAS, Prof. Moustafa MOULAI et Prof. Méziane AIDER, professeurs à l'USTHB-Alger, ainsi le Prof. AIDENE Mohamed et D. BELEHCEN Fatima, professeurs à l'UMMTO qui m'ont fait l'honneur d'accepter d'évaluer ce travail, ainsi Prof. BOUROUBI Sadek président du jury, pour sa confiance et son estime.

C'est d'une manière indubitable et profondément sincère que je dis un grand merci à mes parents et ma famille pour leur soutien, la fierté et la confiance qu'ils me portent, même si la recherche n'est pas leur milieu et qu'ils ne comprennent pas toujours tout de ce que je fais de mes journées.

Un immense MERCI, tout à fait particulier, à ma femme, je suis conscient que cette thèse ne serait pas ce qu'elle est sans sa patience et soutien.

Je n'oublierai pas de remercier mes amis et mes collègues qui m'ont toujours encouragé et soutenu. Que tous ceux qui ont de près ou de loin contribué à l'aboutissement de ce travail trouvent ici l'expression de mes vifs remerciements.

Boualem BRAHMI.

A toute ma famille.

☞ *Mes parents ;*

☞ *Mes frères et sœurs ;*

☞ *Ma femme ;*

☞ *Mon fils Sam.*

Je dédie ce travail.

Table des figures

1.1	Espace convexe (à gauche) et non convexe (à droite)	16
1.2	Représentation de l'espace des décisions et l'espaces objectifs correspondant.	24
1.3	Illustration des différentes définitions	25
1.4	Représentation des différents types de solutions en bi-objectif	31
1.5	Exemple de l'importance des solutions non supportées	31
1.6	Région des solutions efficaces non supportées.	32
1.7	Espace de décision	34
1.8	Espace des critères	34
1.9	Solutions supportées et non supportées	34
2.1	Schéma de résolution	38
2.2	classification d'un point de vue concepteur	39
2.3	Arbre de "Branch and Bound"	50
2.4	Représentation graphiques des espaces de décision et des critères de l'exemple 3.	56
2.5	Solution optimale de (P_0)	57
2.6	Solution optimale de (P_1)	58
2.7	Solution optimale de (P_2) , x^3 est la solution optimale.	59
2.8	Ensemble des solutions efficaces/non dominés de l'exemple 3.	60
3.1	Le domaine d'admissibilité D	87
3.2	La région réduite D^1	88
3.3	La région réduite D^2	89
3.4	La région réduite D^3	90
4.1	Génération des solution efficaces par un méthode basée sur la distances : (a) cas continu, (b) cas discret	92

TABLE DES FIGURES

4.2	Dominance faible et programme de la norme de Tchebychev	96
4.3	Ensemble des solutions efficeces/nondominés de l'exemple 5.	105
4.4	Iteration 2.	106
4.5	Liste des solutions efficaces/non-dominées de l'exemple 5	107

Liste des tableaux

2.1	Tableau optimal associé à la base B.	48
2.2	Ensemble des solutions efficaces/non dominées de l'exemple 3	60
4.1	Ensemble des solutions efficaces/nondominées de l'exemple 5	105
4.2	Résultats obtenus en chaque itération de l'exemple 5	107
4.3	Résultats obtenus pour : $10 \leq n \leq 100$ et $10 \leq m \leq 100$	109
4.4	Résultats obtenus pour : $100 \leq n \leq 220$ et $100 \leq m \leq 200$	110

Table des matières

Liste des figures	2
Liste des tableaux	3
Table des matières	3
Introduction générale	8
1 Optimisation : Concepts de base	12
1.1 Introduction	12
1.2 Introduction sur les problèmes d'optimisation	13
1.2.1 Problème d'optimisation	13
1.2.2 Face à un problème d'optimisation	14
1.3 Optimisation mono-objectif	15
1.3.1 Concepts préliminaires	15
1.3.1.1 Ensembles et fonctions	15
1.3.1.2 Normes et distances	17
1.3.1.3 Boules, sphères	17
1.3.1.4 Rappels sur les normes vectorielles	18
1.3.1.5 Minima locaux et globaux	18
1.3.2 Programmation linéaire	19
1.3.2.1 Introduction	19
1.3.3 Définition et différentes forme d'un PL	19
1.3.3.1 Espaces des solutions	21
1.3.3.2 Dualité	22
1.3.4 Programmation linéaire en nombres entiers	22
1.4 Optimisation multi-objectif	23
1.4.1 Définitions	23
1.4.2 Points particuliers	24

1.4.3	Propriétés	25
1.4.4	Caractérisation des solutions efficaces continues	27
1.4.4.1	Paramètres de préférences	27
1.4.4.2	Fonctions scalarisantes	27
1.5	Résultats théoriques sur la caractérisation	28
1.6	Optimisation linéaire multi-objectifs	28
1.6.1	Quelques résultats de base	29
1.6.2	Détection graphique de l'efficacité	32
1.7	Conclusion	34
2	Méthodes de résolution des programmes multi-objectifs	36
2.1	Approches de résolution multi-objectifs	36
2.1.1	Classification d'un point de vue décideur	37
2.1.2	Classification point de vue concepteur	38
2.1.2.1	Approches non Pareto	39
2.1.2.2	Approches Pareto	41
2.2	Résolution d'un programme linéaire	42
2.2.1	Résultats de base	42
2.2.1.1	La méthode du simplexe	42
2.3	Méthodes de résolution d'un programme linéaire en nombres entiers	47
2.4	Méthodes exactes pour la résolution des problèmes MOLP	51
2.4.1	Simplexe multi-objectifs	51
2.5	Méthodes exactes pour la résolution des problèmes MOILP	52
2.5.1	Méthode de D. Klein et E. Hannan :	52
2.5.2	Méthode de Sylva et Crema (2004)	53
2.5.2.1	Exemple didactique :	56
2.6	Conclusion	60
3	Optimisation d'une fonction linéaire sur l'ensemble efficace	61
3.1	Optimisation d'une fonction linéaire sur l'ensemble efficace continu	61
3.1.1	Introduction	61
3.1.2	Notations et résultats de base	62
3.1.3	Algorithmes de recherche des sommets adjacents	65
3.1.4	La méthode de Yamamoto, 2002[76]	65
3.1.5	Algorithmes de recherche des sommets non adjacents	66
3.1.6	Algorithme basé sur le dual	68

3.1.7	Algorithme de recherche de faces efficaces	70
3.1.8	Algorithme de la bisection	71
3.1.9	Autres méthodes	72
3.2	Optimisation d'une fonction linéaire sur l'ensemble efficace discret	74
3.2.1	Introduction	74
3.2.2	Notations et notions préliminaires	74
3.2.3	Test d'efficacité	75
3.2.4	Méthodes de résolution	76
3.2.4.1	Méthode de Abbas et Chaabane[1]	76
3.2.4.2	L'algorithme de la méthode	78
3.2.4.3	Méthode de Jorge,2009 [42]	80
3.2.4.4	L'algorithme de la méthode	80
3.2.4.5	Illustration numérique	81
3.2.4.6	Méthode de Chaabane et Pirlot, 2010 [17]	84
3.2.4.7	L'algorithme de la méthode	84
3.2.4.8	Illustration numérique :	86
3.3	Conclusion	90
4	La norme augmentée de Tchebychev pour optimiser une fonction Li-	
	néaire sur l'Ensemble des Solutions Efficaces Discrètes	91
4.1	Introduction	91
4.2	Quelques méthodes de résolution multicritère basée sur la distance avec un objectif de référence	92
4.3	Norme de Tchebychev et résultats préliminaires	93
4.3.1	La méthode de Steuer et Choo, 1983[62]	96
4.3.2	L'algorithme de Ralphs et al., 2007 [55]	98
4.4	Optimisation d'un critère linéaire sur l'ensemble efficace discret basée sur la norme augmentée de Tchebychev	99
4.4.1	Description générale de l'algorithme	101
4.4.2	l'algorithme de la méthode	103
4.4.3	Exemple d'illustration	104
4.5	Évaluation numérique	106
4.5.1	Implémentation et instances	106
4.5.2	Interprétation des résultats	110
4.6	Conclusion	111

TABLE DES MATIÈRES

Conclusion générale et perspectives	112
Bibliographie	114

Introduction générale

Cette thèse s'inscrit dans le domaine de l'optimisation non convexe, en particulier de la résolution de problème d'optimisation d'une fonction sur l'ensemble des solutions efficaces d'un programme linéaire multi-objectifs en nombres entiers.

Beaucoup de problèmes d'optimisation issus du monde réel sont complexes à résoudre. De plus, les problèmes d'optimisation rencontrés en pratique sont rarement mono-objectif, et requièrent souvent la prise en compte de plusieurs critères conflictuels, notamment en logistiques, qui souvent complexe et de grande de taille. D'autant que la complexité des problèmes multi-objectifs augmente en fonction de la taille du problème à résoudre, aussi bien en termes d'espaces de recherche qu'en termes de nombre de fonctions objectifs à optimiser. En effet, contrairement au cas mono-objectif, il n'existe pas une solution optimale unique, mais un ensemble de solutions efficaces, dites Pareto optimales. Une solution est Pareto optimale si l'amélioration à l'égard d'une fonction objectif entraîne invariablement une détérioration relativement à une autre fonction objectif. Ces solutions représentent les compromis entre les différentes fonctions objectifs à optimiser. L'union des solutions *Pareto optimales* (*solution efficace*) forme l'ensemble *Pareto optimal* (ensemble efficace), et l'image de cet ensemble dans l'espace des objectifs est appelé le *Front de Pareto*. Ainsi, l'optimisation multi-objectifs s'intéresse aux particularités liées à l'existence de ces solutions optimales multiples, et aux méthodes de résolutions dédiées à ce type de problèmes, bien souvent NP-difficiles.

La résolution d'un problème multi-objectifs consiste à trouver une solution Pareto optimale qui répondant au mieux aux préférences du décideur. L'optimisation multi-objectifs est donc étroitement liée au domaine de la décision multicritère. Les approches de résolution initiales consistaient à réduire le problème original en un problème mono-objectif. Néanmoins, depuis plusieurs années, une des questions fondamentales de l'optimisation multi-objectif s'apparente à l'identification de l'ensemble Pareto optimal. Vu la dimension de l'ensemble Pareto optimal, qui est souvent grand et de temps en temps infini.

De ce fait les problèmes multi-objectifs aussi bien académiques que industriels sont NP-difficile, et ne peuvent pas donc résolus en un temps de calcul raisonnable. Il s'avère donc nécessaire de considérer d'autres alternatives, parmi elles celles de la prise de décision multicritères. Dans des applications pratiques de prise de décision à critères multiples, les décideurs doivent souvent choisir un certain point préféré de l'ensemble des solutions efficaces ; ceci nous conduit à trouver les solutions efficaces en décrivant la structure de l'ensemble efficaces. Comme, dans beaucoup de cas, les critères sont en conflit, les décideurs essayent alors d'optimiser un certain critère de compromis sur l'ensemble efficace, ce qui mène à rechercher une méthode qui optimise une fonction linéaire sur un ensemble de solutions efficaces. Pour cela, l'optimisation d'une fonction sur l'ensemble de solutions efficaces d'un problème multiobjectifs à connu un intérêt grandissant au cours des quatre dernières décennies.

Benson (1992)[7] a prouvé que dans quelques problèmes de modélisation impliquant des objectifs multiples, les modèles d'optimisation sur l'ensemble des solutions efficaces sont les réalistes et appropriés que les programmes¹ linéaires multi-objectifs plus habituels (MOLP)². En effet, dans certaines situations, les décideurs n'ont pas besoin de tout l'ensemble des solutions efficaces d'un problème multi-objectifs mais uniquement de solutions efficaces qui réalisent l'optimum d'un objectif différent des objectifs déjà fixés. Ceci nous mène vers la recherche de la solution optimale d'un critère, appelé critère principal, sur *l'ensemble de solutions efficaces* du problème multi-objectifs. L'intérêt porté sur ce type de problème est d'éviter d'énumérer toutes les solutions efficaces du problème multi-objectifs, mais seulement un sous ensemble de solutions efficaces sera fourni.

On ne peut nier l'intérêt et l'avantage acquis par cette fameuse alternative néanmoins elle possède une difficulté, qui réside dans la non convexité de l'ensemble des solutions efficaces. Cela dit, cette alternative appartient à la classe des problèmes d'optimisation non convexe ou bien d'optimisation globale.

Généralement, le développement des modèles théoriques et des techniques traitant *des problèmes d'optimisation* peuvent être partagé en deux classes à savoir, la classe de problèmes convexes, et la classe de problèmes non convexes. Cette classification a poussé Rokhafollar à dire "*the great watershed in optimization isn't between linearity or non linearity but between convexity and non convexity*", dont elle a connu une accélération spectaculaire, particulièrement après la deuxième guerre mondiale.

1. Notons que le mot "programme" n'a pas ici le sens usuel (élaboration d'un programme pour ordinateur) ; il est utilisé car un ensemble de valeurs des variables satisfaisant les contraintes d'un problème de programmation mathématique est parfois appelé un programme.

2. MOLP :Multi Objective Linear Programming

Dans ce travail, nous portons un intérêt pour le cas où le domaine de réalisabilité³ est non convexe et où les fonctions à optimiser ainsi que toutes les contraintes du problème multi-objectifs sont linéaires. Cet intérêt est dicté par l'aspect applicatif important de la programmation mathématique où dans bien des cas la présence des variables entières est inévitable. En effet, que l'on s'intéresse à l'optimisation d'un système de production, au design de réseaux de télécommunication, à la bio-informatique ou encore à l'extraction de connaissances, on sera confrontés à des problèmes d'optimisation discrète. On aura alors affaire à un problème de programmation multi-objectifs linéaire en nombres entiers. Les applications qui ont été faites de la programmation linéaire multi-objectif ces dernières années sont très variées. Cela tient bien sûr au fait que le modèle est relativement général et permet de traiter une quantité de problèmes de gestion scientifique. Mais cela est dû aussi d'une part au fait que la programmation linéaire multi-objectif faisait appel à la programmation linéaire mono-objectif dont on connaît des algorithmes extrêmement efficaces pour les résoudre à l'occurrence la méthode du simplexe. Cette dernière développée par G. B. Dantzig (vers 1947) a conduit à plusieurs algorithmes généraux qui permettent de résoudre aisément des problèmes de tailles considérables.

Organisation de manuscrit

Ce manuscrit est organisé en quatre chapitres :

Un chapitre étudie les concepts de base de *l'optimisation*, en donnant les principales définitions et propriétés liées à l'optimisation mono-objectif et multi-objectifs. Les propriétés données dans ce chapitre seront utilisées constamment dans les chapitres ultérieurs.

Un aperçu de définitions, propriétés et la classe générale des méthodes dédiées à la programmation multi-objectif particulièrement à la programmation linéaire multi-objectif seront exposés dans le deuxième chapitre. Ce dernier s'inscrit dans la continuité du précédent en s'intéressant cette fois-ci aux méthodes exactes de résolution des problèmes d'optimisation multi-objectifs, particulièrement dans le cas linéaire et en se basant plus sur le cas où les variables de décisions sont entières, nous détaillons les méthodes jugées intéressantes pour notre contribution, à savoir celles qui sont basées sur la théorie de la norme de Tchebychev et celles basées sur la réduction progressive du domaine de recherche. Vu la variété théorique et applicative de l'optimisation non convexe, notre travail sera restreint seulement à l'optimisation d'une fonction linéaire sur un ensemble de solutions efficaces d'un problème linéaire multi-objectif en nombre entiers⁴. Nous consac-

3. qui représente l'ensemble de solutions efficaces discrète

4. Le problème d'optimisation sur l'ensemble des solutions efficaces appartient à la classe d'optimisation

crons le troisième chapitre à la présentation d'un état d'art des méthodes de résolution de problèmes d'optimisation d'une fonction linéaire sur l'ensemble des solutions efficaces des deux cas continu et discret.

Dans le dernier chapitre, nous proposons un algorithme exact pour la résolution du problème d'optimisation d'une fonction linéaire sur l'ensemble de solutions efficaces d'un programme linéaire multi-objectifs en nombres entiers. Une évaluation numérique de l'algorithme proposé en l'implémentant sous l'environnement Matlab et utilisant CPLEX 12.2 Library à été présentée. Ce travail a fait l'objet d'une première publication dans la revue "International Transaction in Operational Research (ITOR)", et une autre soumission dans journal RAIRO-Operations Research (2015).

Enfin, une brève conclusion résume les principaux acquis de cette thèse et ouvre des perspectives de recherche prolongeant le travail accompli.

1

Optimisation : Concepts de base

Le premier chapitre, est un chapitre introductif à l'optimisation. Il contient un rappel des notions fondamentales ainsi les principales définitions et propriétés de l'optimisation mono-objectif et celles de l'optimisation multi-objectifs. Une présentation d'une variétés de problèmes réels, traités dans la littérature en optimisation, a été présentée.

1.1 Introduction

Le désir humain de perfection trouve son expression dans la théorie de l'optimisation. Elle étudie comment décrire et atteindre ce qui est meilleur, une fois que l'on connaît comment mesurer et modifier ce qui est bon et ce qui est mauvais. "... La théorie de l'optimisation comprend l'étude quantitative des optimum et les méthodes pour les trouver"[71].

À partir de la citation ci-dessus, on comprend la motivation d'un processus d'optimisation. En fait, l'optimisation est une branche des mathématiques, cherchant à analyser et à résoudre analytiquement ou numériquement, les problèmes qui consistent à déterminer le meilleur élément d'un ensemble, au sens d'un critère quantitatif donné. Ce mot vient du latin optimum qui signifie le meilleur.

1.2 Introduction sur les problèmes d'optimisation

De nombreux secteurs de l'industrie sont concernés par les problèmes d'optimisation. En effet, que l'on s'intéresse à l'optimisation d'un système de production, au traitement d'images, à la conception de systèmes, au design de réseaux de télécommunications ou à la bio-informatique nous pouvons être confrontés à des problèmes d'optimisation. Plusieurs problèmes ont été traités dans différents domaines :

- Design de systèmes dans les sciences d'ingénieurs (mécanique, aéronautique, chimie,... etc.) : ailes d'avions, moteurs d'automobiles [34] ;
- Ordonnancement et affectation : ordonnancement en productions, localisation d'usines, planification de trajectoires de robots mobiles [33],... etc.
- Agronomie : programme de production agricole,... etc.
- Transport : gestion de containers, design de réseaux de transport [32], tracé autoroutier, etc.
- Environnement : gestion de la qualité de l'air, distribution de l'eau, déchet ménagers,... etc.
- Télécommunications : design d'antennes, affectation de fréquences, radiotéléphonie mobile,... etc.

1.2.1 Problème d'optimisation

Un problème d'optimisation est défini par :

1. Un *espace* de recherche (de *décision*) : ensemble de solutions ou de configurations constitué des différentes valeurs prises par les variables de décision.
2. Une ou plusieurs fonction(s) dite *objectif(s)*, à optimiser (minimiser ou maximiser).
3. Un ensemble de *contraintes* à respecter.

Dans la plupart des problèmes, l'espace d'état (décision) est fini ou dénombrable. Les variables du problème peuvent être de nature diverse (réelle, entier, booléenne,... etc.) et exprimer des données qualitatives ou quantitatives. La *fonction objectif* représente le but à atteindre pour le décideur. L'*ensemble de contrainte* définit des conditions sur l'espace d'état que les variables doivent satisfaire. Ces contraintes sont souvent des contraintes d'inégalité ou d'égalité et permettent en général de limiter l'espace de recherche (solutions réalisables). La résolution d'un problème d'optimisation consiste à trouver un point de l'espace de recherche qui satisfait au mieux la fonction objectif. Le résultat est appelé *valeur optimale* ou *optimum*. Néanmoins en raison de la taille des problèmes réels,

la résolution optimale s'est souvent montrée impossible dans un temps raisonnable. Cette impossibilité technique impose la résolution approchée du problème, qui consiste à trouver une solution de bonne qualité (la plus proche possible de l'optimum). Il est vital pour déterminer si une solution est meilleure qu'une autre, que le problème introduise un critère de comparaison (*une relation d'ordre*). La plupart des problèmes d'optimisation appartiennent à la classe des problèmes *NP-difficile* où il n'existe pas d'algorithme qui fournit la solution optimale en temps polynomial en fonction de la taille du problème et le nombre d'objectifs à optimiser. Dans la littérature il existe des problèmes académiques utilisés comme des benchmarks : sac à dos, voyageur de commerce, Flowshop,... et des problèmes réels (applications industrielles) : télécommunications, transport, environnement,...etc.

1.2.2 Face à un problème d'optimisation

Dans un processus de résolution des problèmes d'optimisations reflétant une situation réelle, trois étapes à considérer :

- Élaborer un modèle (mathématique) : l'expression de l'objectif à optimiser et les contraintes à respecter (Modélisation).
- Développer un algorithme de résolution (Résolution).
- Évaluer la qualité des solutions produites (Analyse).

Un problème d'optimisation mathématique peut être formulé comme suit :

$$\max_{x \in X} \text{ ou } \min_{x \in X} f(x) \tag{1.1}$$

où X est l'ensemble de solutions réalisables qui représente une partie de \mathbb{R}^n qui décrit l'ensemble des contraintes, généralement $X = \{x | g_i(x) \leq 0, i = 1..m\}$ tel que $g_i(x) : \mathbb{R}^n \rightarrow \mathbb{R}$, m : le nombre de contraintes et f est la fonction à optimiser.

Un problème d'optimisation est caractérisé par :

- Le domaine des variables de décision X : soit continu et on parle alors de *problème continu*, soit discret et on parle donc de *problème combinatoire* ;
- La nature de la fonction objectif f à optimiser et les contraintes ($g_i, i = 1..m$) : soit linéaires et on parle alors de *problème linéaire*, soit non linéaires et on parle donc de *problème non linéaire* ;
- Le nombre de fonctions objectifs à optimiser : soit une fonction scalaire et on parle alors de *problème mono-objectif* qui sera discuté dans la section suivante, soit une fonction vectorielle et on parle donc de *problème multi-objectifs*, que nous aborderons dans la section 1.4 ;

- La présence ou non des contraintes : on parle de *problème sans contrainte* ou avec *contrainte*
 - Sa taille : *problème de petite* ou de *grande taille* ;
 - L’environnement : *problème dynamique* (la fonction objectif change dans le temps).
- et si en plus
- Le domaine des variables ainsi la (les) fonction(s) à optimiser soient convexe on parle de problème convexe ;
 - Le domaine des variables ou la (les) fonction(s) à optimiser soient non convexe on parle alors du problème non convexe, qui fera l’objet de notre étude.

1.3 Optimisation mono-objectif

Le problème général d’optimisation mono-objectif consiste à optimiser (minimiser ou maximiser) une fonction objectif en tenant compte d’un certain nombre de contraintes auxquelles le problème est assujéti.

1.3.1 Concepts préliminaires

Nous allons ici rappeler brièvement les définitions de quelques notions importantes auxquelles nous ferons appel par la suite, ainsi que quelques propriétés.

1.3.1.1 Ensembles et fonctions

La notion de convexité joue un rôle très important dans la théorie classique de l’optimisation. Elle est un outil indispensable pour la recherche des conditions à la fois nécessaires et suffisantes d’optimalité. En effet, la majorité des algorithmes proposés dans le cas de l’optimisation convexe sont efficaces.

Définition 1 (Ensembles convexes) Soit l’ensemble $\mathcal{X} \subseteq \mathbb{R}^n$. \mathcal{X} est convexe si

$$\lambda x + (1 - \lambda)y \in \mathcal{X}, \quad \forall x, y \in \mathcal{X}; \forall \lambda \in [0; 1].$$

Définition 2 Soient les p vecteurs $x^i \in \mathbb{R}^n$, $i = 1, 2, \dots, p$. L’enveloppe convexe de \mathcal{X} , notée $\text{conv}(\mathcal{X})$, est l’ensemble des combinaisons convexes finies d’éléments de \mathcal{X} , c’est-à-dire $\text{conv}(\mathcal{X}) = \sum_{i=1}^p \lambda_i x^i \mid \lambda_i \in \mathbb{R}^+, x^i \in \mathcal{X}, \forall i = 1, 2, \dots, p$ et $\sum_{i=1}^p \lambda_i = 1$.

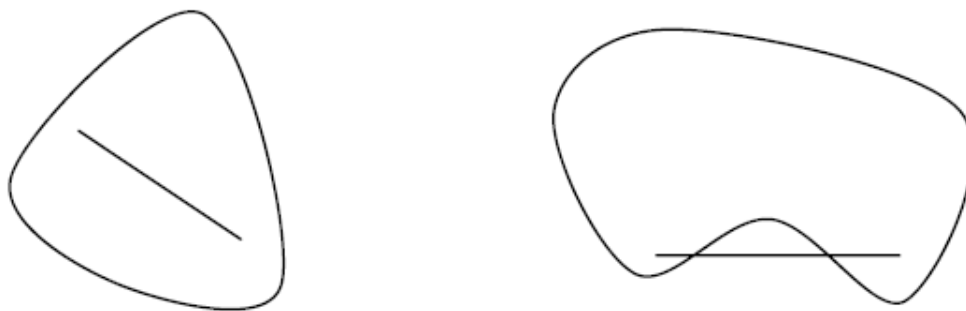


FIGURE 1.1 – Espace convexe (à gauche) et non convexe (à droite)

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

Définition 3 Soit X un ensemble convexe de \mathbb{R}^n . On définit la variété linéaire engendrée par X comme étant combinaison affine $\text{aff}(X) = \sum_{i=1}^n \lambda_i x_i \mid \lambda_i \in \mathbb{R}, x_i \in X, \forall i = 1, 2, \dots, n$ et $\sum \lambda_i = 1$.

Définition 4 On appelle intérieur relatif d'un ensemble convexe X , son intérieur dans $\text{aff}(X)$, muni de la topologie induite de celle de \mathbb{R}^n . On le note $\text{int}(X) = \{x \in X \mid \text{il existe } r > 0 \text{ tel que } \mathbf{B}(x, r) \cap \text{aff}(X) \subset X\}$ où $\mathbf{B}(x, r) = \{y : \|x - y\| \leq r\}$ est la boule de centre x et de rayon r .

Propriété 1 Si X_1, \dots, X_r sont des ensembles convexes, alors l'intersection $\bigcap_{i=1}^r X_i$ est convexe.

Définition 5 Soit $f : X \rightarrow \mathbb{R}$ une fonction définie sur un ensemble convexe X de \mathbb{R}^n . On appelle domaine de f l'ensemble

$$\text{dom}(f) = \{x \in X \mid f(x) < +\infty\}.$$

Définition 6 (Fonction convexe) Soit l'ensemble convexe $X \subseteq \mathbb{R}^n$. Une fonction $f : X \mapsto \mathbb{R}^n$ est convexe si

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

Une fonction f est concave si $-f$ est convexe. Une fonction f est strictement convexe si l'inégalité ci-dessus est stricte pour tout $x, y \in X$ tels que $x \neq y$ et tout $\lambda \in]0; 1[$.

Définition 7 (Fonction quasi-convexe) On dit qu'une fonction $f(x)$ est quasi-convexe sur un intervalle $I(I \subseteq \mathbb{R}^n)$ si et seulement si :

$$\forall (x_1, x_2) \in I^2, \forall \lambda \in [0, 1], f(\lambda x_1 + (1 - \lambda)x_2) \leq \max\{f(x_1), f(x_2)\}$$

On dit qu'une fonction f est quasi-concave sur l'intervalle I si et seulement si $-f$ est quasi-convexe.

1.3.1.2 Normes et distances

Définition 8 On appelle norme dans \mathbb{R}^q toute application de \mathbb{R}^q dans \mathbb{R}^+ , notée $x \mapsto \|x\|$, et vérifiant les propriétés suivantes :

1. Pour tout $x \in \mathbb{R}^q$, $\|x\| \geq 0$ et $\|x\| = 0 \Rightarrow x = 0$;
2. Pour tout $x \in \mathbb{R}^q$ et tout $y \in \mathbb{R}^q$, $\|x + y\| \leq \|x\| + \|y\|$;
3. Pour tout $x \in \mathbb{R}^q$ et tout $\lambda \in \mathbb{R}$, $\|\lambda x\| \leq |\lambda| \cdot \|x\|$.

On utilise généralement trois normes dites usuelles définies respectivement pour $x = (x^1, \dots, x^q)$ au moyen des formules $\|x\| = ((x^1)^2 + \dots + (x^q)^2)^{\frac{1}{2}}$ (norme euclidienne), $\|x\| = \max(|x^1|, \dots, |x^q|)$, $\|x\| = |x^1| + \dots + |x^q|$.

Dans la pratique on est amené à travailler avec la **structure affine** de \mathbb{R}^q . Disons, pour simplifier, à considérer les éléments de \mathbb{R}^q comme des **points** sur lesquels opèrent des vecteurs. On obtient alors la notion de distance :

Définition 9 Soit $x \mapsto \|x\|$ une norme dans \mathbb{R}^q , a et b deux points de \mathbb{R}^q . On appelle distance de a à b déduite de la norme, et on note $d(a, b)$, la norme du vecteur d'origine a et d'extrémité b , soit $d(a, b) = \|b - a\| = \|a - b\|$.

La propriété de la fonction distance la plus intuitive, et la plus utilisée, est l'inégalité triangulaire :

$$\forall (a, b, c) \in \mathbb{R}^q \times \mathbb{R}^q \times \mathbb{R}^q, \quad d(a, c) \leq d(a, b) + d(b, c).$$

1.3.1.3 Boules, sphères

Définition 10 $x \mapsto \|x\|$ une norme dans \mathbb{R}^q . Soit a un point de \mathbb{R}^q et r un nombre réel strictement positif. on appelle boule ouverte (resp. boule fermée, resp. sphère) de centre a et de rayon r , et le note $B(a, r)$ (resp. $B^I(a, r)$, resp. $S(a, r)$), l'ensemble des points $x \in \mathbb{R}^q$ tels que $d(a, x) = \|x - a\| < r$ (resp. $d(a, x) = \|x - a\| \leq r$, resp. $d(a, x) = \|x - a\| = r$)

1.3.1.4 Rappels sur les normes vectorielles

Une norme vectorielle est une fonction notée $\|\cdot\|$ définie sur un espace vectoriel dans \mathbb{R}^+ et satisfaisant aux trois axiomes suivants :

1. $\|x\| \geq 0$ pour tout x et $\|x\| = 0 \Leftrightarrow x = 0$
2. $\|ax\| = |a|\|x\|$ pour tout x et scalaire a
3. $\|x + y\| \leq \|x\| + \|y\|$ pour tout x, y .

Exemple 1 1. dans \mathbb{R}^n muni du produit scalaire $x^T y, \|x\| = (x^T x)^{\frac{1}{2}} = \sqrt{\sum_{i=1}^n x_i^2}$ est une norme. C'est la norme euclidienne, les plus couramment employée.

2. $\|x\|_\infty = \max_{1 \leq i \leq n} \{|x_i|\}$ est la norme du max (ou la norme de Tchebychev, ou la norme ∞)

3. $\|x\|_1 = \sum_{i=1}^n |x_i|$ est la norme l_1

4. les trois normes précédentes sont des cas particuliers des normes $l_p : \|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}$

1.3.1.5 Minima locaux et globaux

Soit l'ensemble $X \subseteq \mathbb{R}^n$ et une fonction $f : X \mapsto \mathbb{R}$ non linéaire de classe C^1 . Considérons le problème :

$$\min_{x \in X} f(x). \quad (1.2)$$

Les minima locaux et globaux de f sur X sont définis de la manière suivante :

Définition 11 (minimum local). Intuitivement, un vecteur $x^* \in X$ est un minimum local de f sur X s'il a un coût plus faible que celui de ses voisins. Formellement, x^* est un minimum local de f sur X si

$$\exists \epsilon > 0 \text{ tel que } f(x^*) \leq f(x), \quad \forall x \in X \text{ avec } \|x - x^*\| < \epsilon.$$

où $\|v\|$ désigne la norme du vecteur v .

Le minimum local est strict si

$$f(x^*) < f(x) \quad \forall x \in X \text{ et } x \neq x^* \text{ avec } \|x - x^*\| < \epsilon.$$

Définition 12 (minimum global). Un vecteur $x^* \in X$ est un minimum global de f sur X s'il a un coût plus faible que celui de tous les autres vecteurs dans X . Formellement, x^* est un minimum global de f sur X si

$$f(x^*) \leq f(x), \quad \forall x \in X$$

Le minimum global est strict si

$$f(x^*) < f(x), \quad \forall x \in X \setminus \{x^*\}.$$

Les maxima locaux et globaux sont définis de manière similaire. Notons que x^* est un maximum local (respectivement global) de la fonction f sur l'ensemble X si x^* est un minimum local (respectivement global) de $-f$ sur X . Il découle de cette observation que tout problème de maximisation peut être réduit immédiatement à un problème de minimisation (et inversement) en multipliant la fonction objectif par -1 .

Remarque 1 *dans le cas d'une fonction objectif convexe, il n'y a pas de distinction entre minimum local et global : tout minimum local est également global, comme l'établit le théorème suivant.*

Théorème 1 [4] *Soit $f : X \subseteq \mathbb{R}^n \mapsto \mathbb{R}$ une fonction convexe définie sur un ensemble convexe X . Alors, tout minimum local de f sur X est également un minimum global. Si f est strictement convexe, alors il existe au plus un minimum global de f .*

Puisque notre travail sera consacré au problème d'optimisation linéaire, alors dans la section suivante nous allons étudier la Programmation Linéaire (PL).

1.3.2 Programmation linéaire

1.3.2.1 Introduction

La programmation linéaire peut se définir comme un outil mathématique qui permet d'analyser divers types de situations dans laquelle nous retrouvons une fonction objectif que l'on désire optimiser (max, min), et qui est exprimée en fonction des variables. Ces variables appelées variables de décision (dont on veut déterminer les valeurs optimales) sont soumises à des restrictions imposées par les ressources de la situation que l'on veut analyser. Ces restrictions prennent la formes d'équations ou bien d'inéquations.

1.3.3 Définition et différentes forme d'un PL

Un PL est un problème qui consiste à optimiser (max ou min) une fonction linéaire à plusieurs variables soumises à des contraintes linéaires (inégalité ou égalité linéaire)

Présentation d'un PL

Un PL est présenté sous la forme max (ou min) selon l'objectif

$$Z = c_1x_1 + \dots + c_nx_n \quad (1.3)$$

soumises aux contraintes :

$$\begin{array}{rcl} a_{11}x_{11} + \dots + a_{1n}x_{1n} & (\leq, =, \geq) & b_1 \\ a_{21}x_{21} + \dots + a_{2n}x_{2n} & (\leq, =, \geq) & b_2 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ a_{m1}x_{m1} + \dots + a_{mn}x_{mn} & (\leq, =, \geq) & b_m \end{array} \quad (1.4)$$

et aux contraintes de non négativité

$$x_j \geq 0, \quad \forall j = 1, \dots, n \quad (1.5)$$

- Remarque 2**
1. Les $(a_{ij})_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}}$ représentent la quantité de la ressource i requise par unité x_j
 2. Les $(b_i)_{1 \leq i \leq m}$ représentent les quantités des diverses ressources disponibles.
 3. Les éléments (a_{ij}) , (b_i) et (c_j) sont des quantités connus du modèle (paramètres du modèle)

Dans la suite on suppose que la fonction objectif d'un programme linéaire est à maximiser. On appelle forme standard d'un PL la forme suivante :

$$\begin{array}{rcl} \max & z & = C_1x_1 + C_2x_2 + \dots + C_nx_n \\ & \sum_{j=1}^n a_{ij}x_j & = b_i, \quad b_i \geq 0, \quad i = 1..m, \quad x_j \geq 0 \end{array}$$

ou encore

Problème 1

$$\begin{array}{rcl} \max & Z & = C^t x \\ & Ax & = b \\ & x & \geq 0 \end{array}$$

Propriété 2 Tout programme linéaire peut se mettre sous forme standard ou sous forme canonique, à travers les transformations suivantes

1. Si la fonction objectif est du type $\min Z$, on résout le PL en remplaçant ($\min Z = \max(-Z)$).
2. Si un $b_i \leq 0$, on multiplie la contrainte i par -1 .
3. Si la contrainte i est de la forme ($\geq b_i$) on lui soustrait une quantité $e_i \geq 0$ (e_i : variable d'écart).
4. Si la contrainte i est de la forme ($\leq b_i$) on lui ajoute $e_i \geq 0$.

1.3.3.1 Espaces des solutions

On considère le PL précédent, on appelle

Solution du PL : c'est tout vecteur $x \in \mathbb{R}^n$ satisfait les contraintes 1.4

Solution réalisable : On appelle solution réalisable tout vecteur $x \in \mathbb{R}^n$ qui satisfait les contraintes 1.4 et les contraintes de non-négativité 1.5.

Solution optimale : On appelle solution optimale, toute solution réalisable qui optimise l'objectif.

solution de base : On appelle solution de base composé de $n - m$ variables, toute solution satisfaisante (1.4) et telle qu'elle soit composée de $(n - m)$ variables nulles et m variables constitue une base.

Solution de base réalisable : On appelle solution de base réalisable toute solution de base satisfaisant les contraintes de non-négativité (1.5).

Point extrême : Un point de l'ensemble \mathcal{X} convexe est dit point extrême si x ne peut pas s'exprimer comme combinaison linéaire convexe des éléments de \mathcal{X} .

Hyperplan : Un hyperplan dans \mathcal{R}^n est l'ensemble de la forme

$$H = \{x \in \mathbb{R}^n \mid \alpha_1 x_1 + \dots + \alpha_n x_n = \beta\}, (\alpha_i, \beta) \in \mathbb{R} \times \mathbb{R}$$

Demi espace fermé : l'ensemble

$$H = \{x \in \mathbb{R}^n, \alpha_1 x_1 + \dots + \alpha_n x_n \geq \beta\}$$

est appelé demi-espace fermé

Polyèdre : L'intersection de demi-espaces fermés est dit polyèdre fermé

Ensemble borné : Un ensemble $X \subset \mathbb{R}^n$ est dit borné si $\forall x \in X \quad \|x\| \leq a$ (a est fini)

Nous décrivons les conditions d'optimalités, les propriétés de la solution optimale et la méthode du simplexe dans le chapitre 2 section 2.6

1.3.3.2 Dualité

La dualité est l'un des concepts les plus importants en programmation linéaire, en effet à tout modèle de PL (qu'on appelle programme primal) qui lui correspond un autre modèle de PL appelé dual.

L'étude de ce concept lié à la dualité permet :

- En résolvant le programme primal d'obtenir également du tableau optimal la solution optimale du dual(et inversement).
- D'améliorer éventuellement le processus itératif de résolution, en résolvant l'un ou l'autre des problèmes, selon la structure du primal ou du dual
- La théorie de la dualité permet d'établir d'autres algorithmes de résolution comme l'algorithme dual du simplexe
- D'un point de vue pratique la dualité est fondamentale dans l'analyse de sensibilité de la solution optimale et de la post optimale.

D'une manière générale on écrit :

Forme standard :

<u>Primal</u>	<u>Dual</u>
$max Z = Cx$	$min \omega = ub$
$Ax = b$	$u \in \mathbb{R}^m$
$x \geq 0$	$uA \geq C$

Forme canonique :

<u>Primal</u>	<u>Dual</u>
$max Z = Cx$	$min \omega = ub$
$Ax \leq b$	$u \geq 0$
$x \geq 0$	$uA \geq C$

La procédure de l'algorithme dual simplexe, est décrite en détail dans le chapitre 5 de la référence [65].

1.3.4 Programmation linéaire en nombres entiers

Dans de très nombreuses applications, les variables considérées ne peuvent prendre que des valeurs entières, donnant ainsi lieu à la programmation linéaire en variables entières ou variables mixtes si deux types de variables, continues et discrètes coexistent dans la modélisation. La nature de ces problèmes est alors radicalement différente car une solution

optimale est bien difficile à caractériser.

En considérant un programme linéaire pour lequel on ajoute une contrainte d'intégrité qui restreint les valeurs des variables aux seuls nombres entiers, on obtient un problème de programmation linéaire en variables entières (PLNE ou ILP : Integer Linear Programming). Un tel problème se modélise donc comme suit :

$$\begin{aligned} \max Z &= Cx \\ Ax &= b \\ x &\in \mathbb{N} \end{aligned}$$

avec A d'ordre $m \times n$, $rg(A) = m$

À cause de ce changement de nature des variables, l'ensemble admissible X n'est plus convexe et les programmes linéaires en variables entières nécessitent donc d'autres méthodes de résolution, que celles des programmes linéaires en variables continues que nous allons voir dans la deuxième partie du chapitre suivant section 2.3.

1.4 Optimisation multi-objectif

De nombreux problèmes d'optimisation issus du monde réel, notamment dans le domaine de la logistique, doivent faire face à beaucoup de difficultés. En effet, ils sont souvent caractérisés par des espaces de recherche vastes et complexes, de multiples fonctions objectifs contradictoires, ce qui implique l'optimisation multi-objectifs.

L'optimisation multi-objectifs cherche à optimiser plusieurs composants d'un vecteur de fonctions coût. Contrairement à l'optimisation mono-objectif, la solution d'un Problème multi-objectifs (PMO) n'est pas une solution unique, mais un ensemble de solutions, connu comme l'ensemble des solutions Pareto Optimales (PO)¹. Toute solution de cet ensemble est optimale dans le sens qu'aucune amélioration ne peut être faite sur un composant du vecteur sans dégradation d'au moins un autre composant du vecteur.

1.4.1 Définitions

Un problème d'optimisation multi-objectifs ne considère plus une unique fonction à optimiser mais repose sur p fonctions objectifs, c'est-à-dire un vecteur de fonctions objectif. On a donc le modèle :

1. Cet ensemble est appelé aussi frontière Pareto.

Problème 2

$$\begin{aligned} \text{"optimiser"} \quad & Z(x) = (z_1(x), z_2(x), \dots, z_p(x)) \\ \text{t.q} \quad & x \in X \end{aligned} \tag{1.6}$$

Où $p \geq 2$ est le nombre de fonctions objectifs,

- $x = (x_1, \dots, x_n)$ est le vecteur représentant les variables de décision (ou l'action), X représente l'ensemble des solutions réalisables associés à des contraintes d'égalité, d'inégalité et des bornes explicites (espace de décision);
- $Z(x) = (z_1(x), z_2(x), \dots, z_p(x))$ est le vecteur des critères à optimiser et p le nombre de critères.

Définition 13 On appelle espace des décisions $X \subset \mathbb{R}^n$ et espaces des objectifs $Y = \{z(x) : x \in X\} \subset \mathbb{R}^p$. Y , l'image de X par le vecteurs des objectifs, est appelé ensemble réalisable. les vecteurs de \mathbb{R}^n et Y sont respectivement appelés point et point réalisable.

Remarque 3 Dans la suite de ce chapitre, nous considérons, sans perte de généralité, que toutes les fonctions objectifs sont à maximiser.

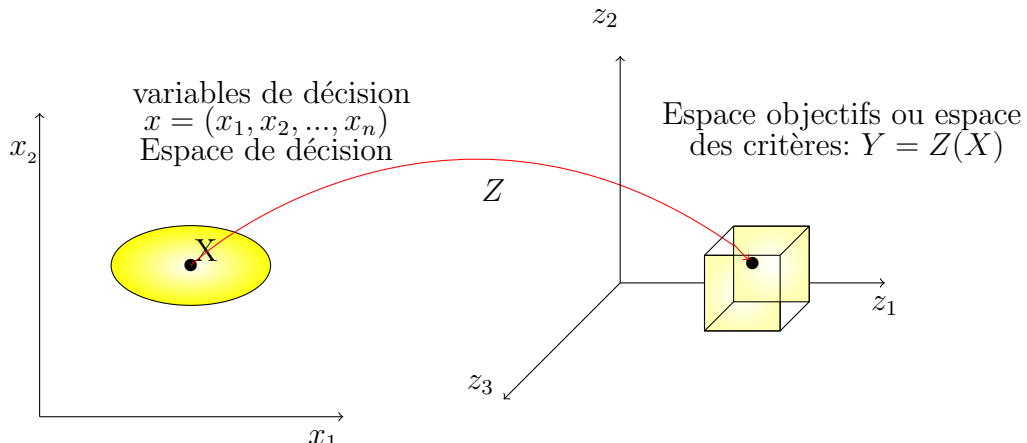


FIGURE 1.2 – Représentation de l'espace des décisions et l'espaces objectifs correspondant.

1.4.2 Points particuliers

En vue d'avoir certains points de références permettant de discuter de l'intérêt des solutions trouvées, des points particuliers ont été définis dans l'espace objectif. Ces points peuvent représenter des solutions réalisables ou non.

1. Tout d'abord, le **point idéal** z^{Id} est le point qui a comme valeur pour chaque objectif la valeur optimale de l'objectif considéré.

$$z^{Id} \text{ tel que } \forall i \in [1..n], Z_i^{Id} = \max_{x \in X} Z_i(x)$$

Ce point ne correspond pas à une solution réalisable car si c'était le cas, cela sous-entendrait que les objectifs ne sont pas contradictoires et qu'une solution optimisant un objectif, optimise simultanément tous les autres, ce qui ramènerait le problème à un problème ayant une seule solution Pareto optimale.

2. De ce point idéal peut être défini le **point utopique** z^U de la façon suivante :

$$z^U = z^{Id} + \epsilon U$$

ou $\epsilon > 0$ et U est le vecteur unitaire ($U = (1, \dots, 1) \in \mathbb{R}^n$). Il est clair, de par sa définition, que ce point n'est pas réalisable.

3. Enfin le **point Nadir** qui est défini en bi-objectif par :

$$z^N \text{ tel que } \forall i \in \{1, 2\}, Z_i(z^N) = \min_{x \in X / Z_j(x) = Z_j(z^{Id})} Z_i(x) \text{ avec } j \neq i$$

Cela revient donc à affecter pour chaque objectif du point Nadir la meilleure valeur possible parmi les solutions optimisant l'autre objectif.

Une visualisation de l'ensemble de ces définitions est donnée sur la figure 1.3

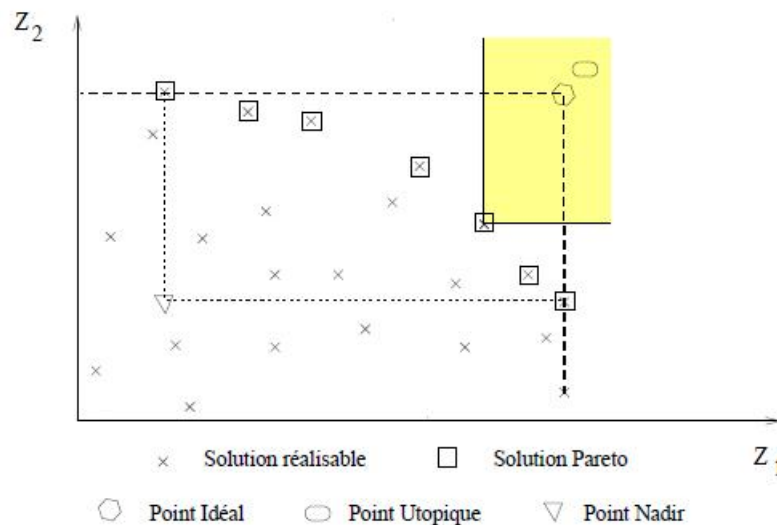


FIGURE 1.3 – Illustration des différentes définitions

1.4.3 Propriétés

En l'absence d'une décision optimale dans le problème multicritère (1.6) qui optimise toutes les fonctions objectifs, il est nécessaire pour identifier les meilleurs compromis de définir une relation d'ordre entre ces éléments. Dans le cas des problèmes d'optimisation

multicritère, ces relations d'ordre sont appelées relations de dominance. Pour définir clairement et formellement ces notions d'optimalité, les relations usuelles $=, <, \leq$ sont étendues aux cas des vecteurs.

Notations :

Contrairement aux problèmes mono-objectifs, où les relations usuelles $<, \leq, \dots$ suffisent pour comparer entre des points, elles sont insuffisantes pour comparer des vecteurs issus de problèmes multicritère. Nous adopterons donc ces notations permettant de prendre en compte tous les cas de figures rencontrés lors de la comparaison de deux vecteurs.

Pour deux vecteurs $x = (x_1, x_2, \dots, x_n)$ et $y = (y_1, y_2, \dots, y_n)$ de \mathbb{R}^n , on note :

$$\begin{aligned} y \geq x &\iff y_i \geq x_i \quad \forall i \in \{1, \dots, n\}; \\ y \geq x &\iff y_i \geq x_i \quad \forall i \in \{1, \dots, n\}, \text{ et } y \neq x \\ y > x &\iff y_i > x_i \quad \forall i \in \{1, \dots, n\}. \end{aligned}$$

Efficacité et non dominance Les fonctions objectifs sont supposées conflictuelles. De ce fait, il n'existe pas de solution admissible optimisant tous les objectifs simultanément. Contrairement au cas des problèmes mono-objectif là où la notion d'optimalité est bien définie dans l'optimisation multi-objectif cette notion ne s'applique donc plus. Il est alors nécessaire de définir un contexte de résolution afin de donner un sens à "min" ou "max".

Définition 14 (Dominance) Soient deux points $y^1, y^2 \in \mathbb{R}^p$. On dit que

- y^1 domine faiblement y^2 si $y_i^1 \geq y_i^2, \forall i = 1, \dots, p$. On écrit alors $y^1 \geq y^2$.
- y^1 domine strictement (fortement) y^2 si $y_i^1 > y_i^2, \forall i = 1 \dots p$. On écrit alors $y^1 > y^2$.
- y^1 domine y^2 si $y^1 \geq y^2$, et $y^1 \neq y^2$ On écrit alors $y^1 \geq y^2$.

Définition 15 (Solution efficace) Soit $x^* \in X$

- x^* est dite efficace s'il n'existe aucune autre solution admissible $x \in X$ telle que $z(x) \geq z(x^*)$. On dit alors que $y^* = z(x^*)$ est un point non dominé.
- x^* est dite faiblement efficace s'il n'existe aucune autre solution admissible $x \in X$ telle que $z(x) > z(x^*)$. On dit alors que $y^* = z(x^*)$ est un point faiblement non dominé.

Définition 16 (Ensembles efficace et non dominé) L'ensemble des solutions efficaces est noté X_E et l'ensemble des points non dominés est Y_N .

Y_N peut alternativement être défini par $Y_N := \{y \in Y : (y - \mathbb{R}^p) \cap Y = \{y\}\}$.

Définition 17 (Équivalence) Deux solutions admissibles x et $x' \in X$ sont dites équivalentes si $z(x) = z(x')$.

1.4.4 Caractérisation des solutions efficaces continues

Il peut arriver dans la programmation mathématique multicritère 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. Les méthodes pour aider un décideur à faire son choix parmi les solutions efficaces peuvent être classées par catégorie selon la façon dont le décideur articule ou incorpore ses préférences. Chaque modèle de préférence est généralement explicité en fixant un certain nombre de paramètres. Citons brièvement les paramètres qui sont fréquemment utilisés.

1.4.4.1 Paramètres de préférences

- Un vecteur $(\lambda_1, \dots, \lambda_p)$ (les λ_k étant appelés coefficients d'importance des p fonctions critères ou paramètres de préférence).
- Des vecteurs de performances ayant des significations particulières, comme le point de référence qui est défini par des niveaux d'aspiration (valeurs souhaitables) sur chaque critère et le point de réserve qui est défini par des niveaux de réservation (valeurs non souhaitables) sur chaque critère.

Nous présentons dans la prochaine section quelques types de fonctions scalarisantes.

1.4.4.2 Fonctions scalarisantes

Étant donné un ensemble de paramètres de préférence $\Lambda = \{(\lambda_1, \dots, \lambda_p) \in \mathbb{R}^p\}$ (les λ_k , étant appelés coefficients d'importance des p fonctions critères), nous définissons une fonction croissante des critères et agrégeons les valeurs des critères pour chaque solution :

$$\begin{aligned} s : \mathbb{R}^p \times \Lambda &\longrightarrow \mathbb{R} \\ (z, \lambda) &\longmapsto s(z, \lambda) \end{aligned}$$

où $\lambda \in \Lambda \subset \mathbb{R}^p$ est le vecteur de paramètres choisi.

Les fonctions d'agrégation les plus employées sont décrites ci-dessous.

Caractérisation à l'aide de poids – La somme pondérée (largement utilisée en optimisation linéaire multi-objectifs où le problème (MOLP) (MOLP :Multiple

Objective Linear Programming) peut se ramener à un problème de programmation paramétrique dans le cas linéaire),

$$\triangleright s_1(Z, \lambda) = \sum_{i=1}^p \lambda_i z_k; \sum_{i=1}^p \lambda_i = 1, \lambda_i \geq 0 \forall i \in \{1, \dots, p\}$$

$$\triangleright s_2(Z, \lambda) = \sum_{i=1}^p \lambda_i |z_i - \bar{z}_i|$$

\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.

1.5 Résultats théoriques sur la caractérisation

Nous présentons quelques caractérisations qui permettent de tester l'efficacité d'une solution réalisable d'un problème multi-objectifs.

Théorème 2 [27] et [72] *Soit $x^* \in X$ un vecteur décision donné et \hat{x} un vecteur quelconque de X . Le vecteur x^* est Pareto optimal pour le problème multi-objectifs (1.6) si et seulement si x^* est une solution optimale du problème auxiliaire suivant*

$$\begin{cases} \max \sum_{i=1}^p z_i(x) \\ \text{s.c. } x \in X \text{ avec } z_i(x) \leq z_i(\hat{x}) \quad \forall i \in \{1, \dots, p\} \end{cases}$$

Théorème 3 [9] *Soit $x^* \in X$ une solution réalisable arbitraire donnée et soit le problème mono-critère suivant :*

$$\begin{cases} \max \Theta = \sum_{i=1}^p \epsilon_i \\ \text{s.c. } x \in X \text{ et } z_i(x) + \epsilon_i = z_i(x^*) \text{ avec } \epsilon_i \geq 0 \quad \forall i \in \{1, \dots, p\} \end{cases} \quad (1.7)$$

Le vecteur x^ est Pareto optimal pour le problème (1.6) si et seulement si la valeur optimale de la fonction objectif Θ est nulle dans le problème (1.7). Si pour une solution optimale \hat{x} du problème (1.7), la valeur de Θ est non nulle alors \hat{x} est Pareto optimal.*

1.6 Optimisation linéaire multi-objectifs

- Problème de programmation linéaire multi-objectifs est un problème de programmation mathématique multi-objectifs avec : les objectifs Z_k et les fonctions g_i sont linéaires .

Problème 3

$$(MOLP) \begin{cases} \text{"optimiser"} & Z_k = C_k x, \quad k = 1, \dots, p \\ \text{t.q} & x \in X \end{cases} \quad (1.8)$$

Où $C \in \mathbb{R}^{1 \cdot n}$, $X = \{x \in \mathbb{R}^n / Ax \leq b, x \geq 0\}$, $A \in \mathbb{R}^{m \cdot n}$ et $b \in \mathbb{R}^{m \cdot 1}$

- Dans ce travail, nous nous intéressons au problème de programmation linéaire multi-objectifs en nombres entiers (MOILP : **M**ultiple **O**bjective **I**nteger **L**inear **P**rogramming) défini par :

Problème 4 *Un problème de programmation linéaire multi-objectifs en nombres entiers peut être formulé comme suit :*

$$(MOILP) \begin{cases} \text{"optimiser"} & Z_k = C_k x, \quad k = 1, \dots, p \\ \text{t.q} & x \in D \end{cases} \quad (1.9)$$

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

Définition 18 (Face) *Le sous-ensemble non vide F du polyèdre X est une face s'il existe un polyèdre L tel que $F = L \cap X$ et X est tout entier situé dans un demi-espace fermé délimité par L mais non inclus à L .*

Définition 19 (p-facette) *Soit F un sous-ensemble du polyèdre X et L un hyperplan de support de X . Alors F est dit une "p-facette" de X si et seulement si $L \cap X = F$ et F de dimension égale à p . Les points extrêmes sont des faces de dimension 0, les arêtes de X sont des faces de dimension 1.*

Définition 20 (Face Efficace) . *Une face F de X est dite face efficace si tout $x \in F$ est efficace.*

1.6.1 Quelques résultats de base

Considérons le problème (MOLP). L'ensemble X_E représente son ensemble de solutions efficaces et $X \setminus X_E$ l'ensemble des solutions non efficaces.

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

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

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

$$(P_\lambda) \begin{cases} \max \sum_{i=1}^p \lambda_i Z_i(x) \\ \text{tq} \quad Z = (Z_1(x), \dots, Z_p(x))' \in \psi(S) \end{cases} \quad (1.10)$$

qui consiste en la recherche d'un élément $\tilde{Z} \in \psi(S)$ tel que $\lambda\tilde{Z} \geq \lambda Z$ pour tout $Z \in \psi(S)$.

Classification des solutions efficaces et des points non dominés

Une méthode bien connue pour calculer des solutions efficaces consiste à résoudre des problèmes dont la fonction objectif est une somme pondérée des p objectifs initiaux. Citons l'un des théorèmes de base dans la recherche des solutions efficaces.

Théorème 4 [37] *Étant donné le problème (P_λ) , alors \tilde{Z} est un vecteur non dominé si et seulement si \tilde{Z} est une solution optimale du problème paramétrique (P_λ) .*

Définition 21 *Les vecteurs $\lambda \in \mathbb{R}^p$ sont appelés poids.*

Les problèmes (MOLP) et (MOILP) sont de nature différente, la différence est due au fait que l'ensemble X est convexe par contre l'ensemble D n'est plus convexe, pour cette raison le principe de Geoffrion qui permet de caractériser l'ensemble des solutions efficaces par la résolution du problème paramétrique (P_λ) n'est pas valable pour les problèmes (MOILP), car certaines solutions efficaces peuvent ne pas être obtenues pour aucun $\lambda > 0$.

En effet cette méthode de calcul, populaire grâce à sa simplicité de mise en œuvre, a conduit à la distinction des solutions efficaces et des points non dominés en deux catégories :

- les points non dominés supportés, se trouvant sur la frontière de l'enveloppe convexe de Y_N . Les ensembles de points non dominés supportés et des solutions efficaces supportées correspondantes sont notés Y_{SN} et X_{SE} . Ces points sont obtenus par la résolution de sommes pondérées.
- les points non dominés non supportés, constitués des points de Y_N situés à l'intérieur de son enveloppe convexe. Les ensembles de points non dominés non supportés et des solutions efficaces non supportées correspondantes sont notés Y_{NN} et X_{NE} .

Par ailleurs, on peut encore distinguer deux types de solutions parmi X_{SE} :

- les solutions efficaces supportées extrêmes dont l'image se situe sur un sommet de $conv(Y)$. Ces solutions forment X_{SE1} et $Y_{SN1} = z(X_{SE1})$ est l'ensemble des points non dominés supportés extrêmes ;
- les solutions efficaces supportées non extrêmes dont l'image n'est pas sur un sommet de $conv(Y)$. Ces solutions forment X_{SE2} et $Y_{SN2} = z(X_{SE2})$ est l'ensemble des points non dominés supportés non extrêmes.

Une représentation graphique des solutions supportées et non supportées est donnée dans la figure 1.4 La principale difficulté réside dans la détermination des solutions non supportées (c'est-à-dire celles qui ne sont pas sur la "frontière efficace" mais sont situées dans

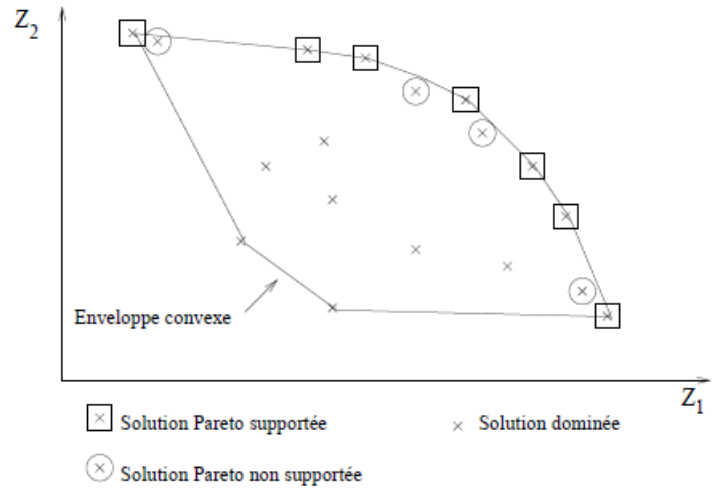


FIGURE 1.4 – Représentation des différents types de solutions en bi-objectif

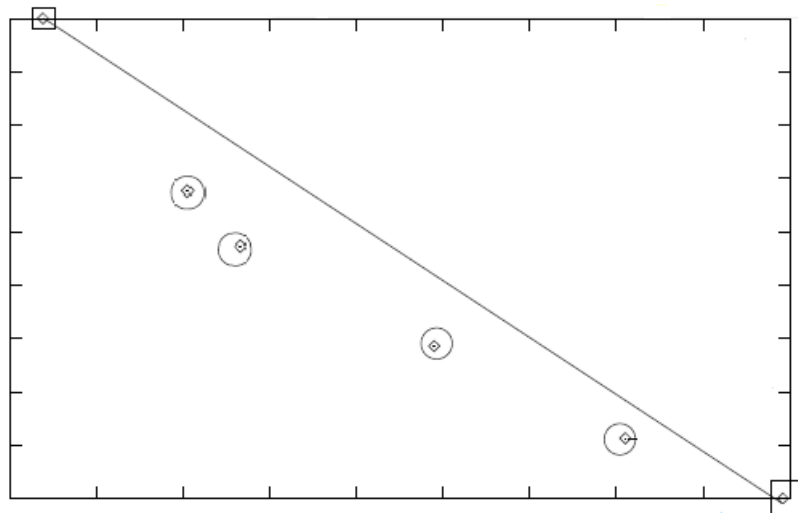


FIGURE 1.5 – Exemple de l'importance des solutions non supportées

la région colorée en noir dans la figure 1.6) puisqu'on ne dispose pas de méthode algorithmiquement simple pour les calculer, comme c'est le cas pour les solutions supportées.

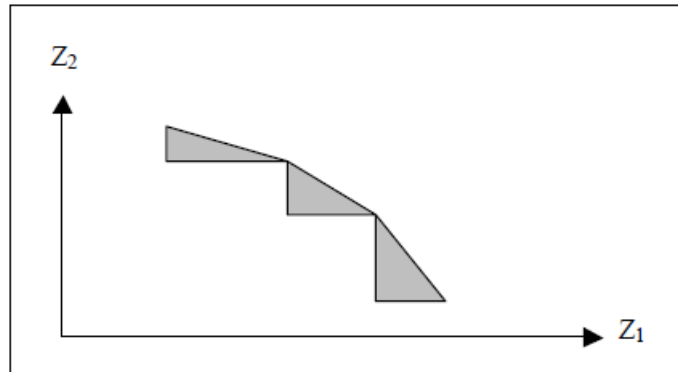


FIGURE 1.6 – Région des solutions efficaces non supportées.

1.6.2 Détection graphique de l'efficacité

Soit le problème de programmation linéaire multi-objectifs en nombres entiers (MOILP). Pour tester l'efficacité en un point $x^* \in D$, Steuer, 1986 [61] a introduit le concept d'ensemble dominant qui est principalement basée sur la notion du cône.

Définition 22 (Cône) Soit $V \in \mathbb{R}^n$, $V \neq \emptyset$, 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 $0 \in \mathbb{R}^n$ appartient en chaque Cône.

Définition 23 (Cône polaire) Soit $V \in \mathbb{R}^n$ un Cône. Le cône polaire non négatif de V (noté V^\geq) est le cône convexe $V^\geq = \{y \in \mathbb{R}^n \mid y^t v \geq 0, \forall v \in V\}$. C'est-à-dire, tous les vecteurs de V^\geq font un angle inférieur ou égal à 90° avec chaque vecteur de V .

Définition 24 (Cône semi-polaire positif) Soit V 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 25 (Générateurs) Considérons v_1, v_2, \dots, v_p , un ensemble de p vecteurs de \mathbb{R}^n et l'ensemble V tel que :

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

V est l'ensemble de toutes les combinaisons coniques 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 .

Définition 26 (Ensemble dominant)

$x^* \in D$ et $C^>$ le cône polaire semi positif du cône C généré par les gradients des p fonctions objectifs i.e. : $V^> = \{y \in \mathbb{R}^n \mid c^i y \geq 0, \text{ pour tout } i \text{ et } c^i y > 0 \text{ pour au moins un } i\} \cup \{0_{\mathbb{R}^n}\}$

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

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

C'est à dire :

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

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

Théorème 5 [61] Soit ED_{x^*} l'ensemble dominant en $x^* \in D$. Alors x^* est efficace si et seulement si : $ED_{x^*} \cap D = \{x^*\}$

Exemple 2 Considérons le programme linéaire bi-objectifs en nombres entiers suivant :

$$(MOILP) \left\{ \begin{array}{l} \max \quad (x_1, -x_1 + x_2) \\ t.q \quad x_1 + 2x_2 \leq 10 \\ \quad \quad x_1 + x_2 \leq 6 \quad x_1, x_2 \in \mathbb{Z}_+ \\ \quad \quad x_1 \leq 4 \end{array} \right.$$

- Les solutions efficaces du programme (P) sont $\{(4, 2), (3, 3), (2, 4), (1, 4), (0, 5)\}$. Par exemple 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 $ED_{x^*} \cap D \neq x^*$

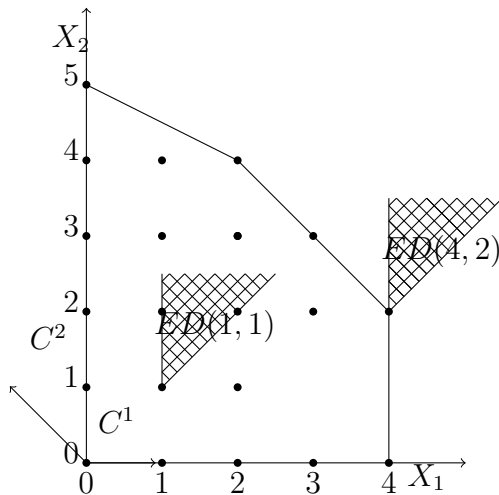


FIGURE 1.7 – Espace de décision

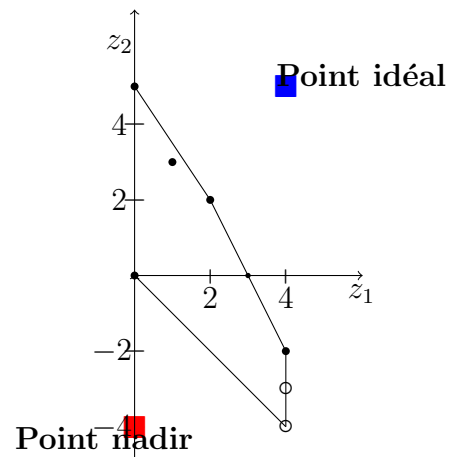


FIGURE 1.8 – Espace des critères

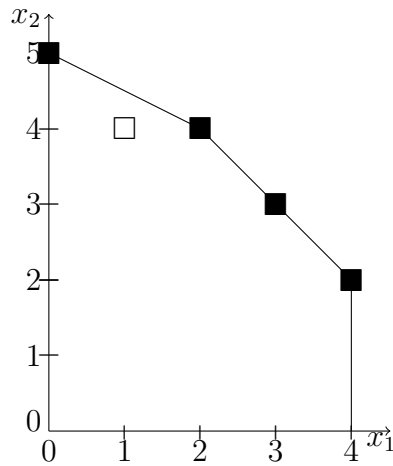


FIGURE 1.9 – Solutions supportées et non supportées

- Deux solutions faiblement non dominées sont détectées, c'est les points $(4,-4)$ et $(4,-3)$.
- Les solutions efficaces supportées sont : $\{(4,2), (3,3), (2,4), (0,5)\}$. Une solution efficace non supportée est détectée, c'est le point $(1,4)$.
- Le point idéal est le point : $z^{id} = (4,5)$.
- Le point utopique $Z^U = Z^{id} + (1,1) = (5,6)$.
- Le point nadir est le point de coordonnées : $z^N = (0,-4)$.

1.7 Conclusion

Dans ce chapitre, les aspects fondamentaux concernant l'optimisation ont été présentés et illustrés par des exemples, à savoir l'optimisation mono-objectif et l'optimisation multi-

1.7. CONCLUSION

objectifs, et plus particulièrement la programmation linéaires multi-objectifs en variables continues ou discrètes.

2

Méthodes de résolution des programmes multi-objectifs

Après avoir présenté, dans le chapitre précédant, les principales propriétés et définitions liées aux problèmes d'optimisation mono-objectif et multi-objectifs dans le cas continu et discret, nous consacrons ce chapitre à la présentation des différentes méthodes exactes de résolution de ces problèmes. Nous nous étendons plus particulièrement sur les méthodes dédiées aux problèmes multi-objectifs en nombres entiers.

2.1 Approches de résolution multi-objectifs

La résolution de problèmes multi-objectifs relève de deux disciplines assez différentes. En effet, résoudre un problème multi-objectifs peut être divisé en deux phases :

1. la recherche des solutions de meilleur compromis : c'est la phase d'optimisation multi-objectifs.
2. le choix de la solution à retenir : c'est la tâche du décideur qui, parmi l'ensemble des solutions de compromis, doit extraire celle(s) qu'il utilisera. On parle alors ici de décision multi-objectifs et cela fait appel à *la théorie de la décision*.

Dans les différents travaux, nous rencontrons deux classifications différentes des approches de résolution de problèmes multi-objectifs. Le premier classement adopte un point de vue décideur, les approches sont classées en fonction de l'usage que l'on désire en faire. Le deuxième classement adopte un point de vue concepteur, les approches sont triées de leurs façons de traiter les fonctions objectifs.

2.1.1 Classification d'un point de vue décideur

La résolution d'un problème multi-objectifs menant à la détermination d'un ensemble de solutions Pareto, il est nécessaire de faire intervenir l'humain (décideur), pour le choix final de la solution à garder.

Ainsi, avant de se lancer dans la résolution d'un problème multi-objectifs, il faut se poser la question du type de méthode d'optimisation à utiliser. En effet, on peut répartir les méthodes de résolution de problèmes multi-objectifs en trois familles, et cela en fonction du moment où intervient le décideur. Ainsi nous pouvons trouver les familles suivantes :

- Les méthodes d'optimisation **a priori** : dans ce cas, le compromis que l'on désire faire entre les objectifs a été défini avant l'exécution de la méthode. Ainsi une seule exécution permettra d'obtenir la solution recherchée. Cette approche est donc rapide, mais il faut cependant prendre en compte le temps de modélisation du compromis et la possibilité pour le décideur de ne pas être satisfait de la solution trouvée et de relancer la recherche avec un autre compromis.
- Les méthodes d'optimisation **progressives** : ici, le décideur intervient dans le processus de recherche de solutions en répondant à différentes questions afin d'orienter la recherche. Cette approche permet donc de bien prendre en compte les préférences du décideur, mais nécessite sa présence tout au long du processus de recherche.
- Les méthodes d'optimisation **a posteriori** : dans cette troisième famille de méthodes, on cherche à fournir au décideur un ensemble de bonnes solutions bien réparties. Il peut ensuite, au regard de l'ensemble des solutions, sélectionner celle qui lui semble la plus appropriée. Ainsi, il n'est plus nécessaire de modéliser les préférences du décideur (ce qui peut s'avérer être très difficile), mais il faut en contre-partie fournir un ensemble de solutions bien réparties, ce qui peut également être difficile et requérir un temps de calcul important (mais ne nécessite pas la présence du décideur).

Donc il existe deux types de comportement. Le premier est de ramener un problème multi-objectifs à un problème mono critère au risque d'enlever toute signification au problème. Le second comportement est d'apporter des réponses au problème en prenant en compte

l'ensemble mono-objectif. La différence entre ces deux communautés s'exprime dans le schéma présenté en figure 2.1. Soit le décideur intervient dès le début de la définition du problème, en exprimant sa préférence, afin de transformer un problème multi-objectifs en un problème mono critère. Soit le décideur effectue son choix dans l'ensemble des solutions proposées par le solveur multi-objectifs.

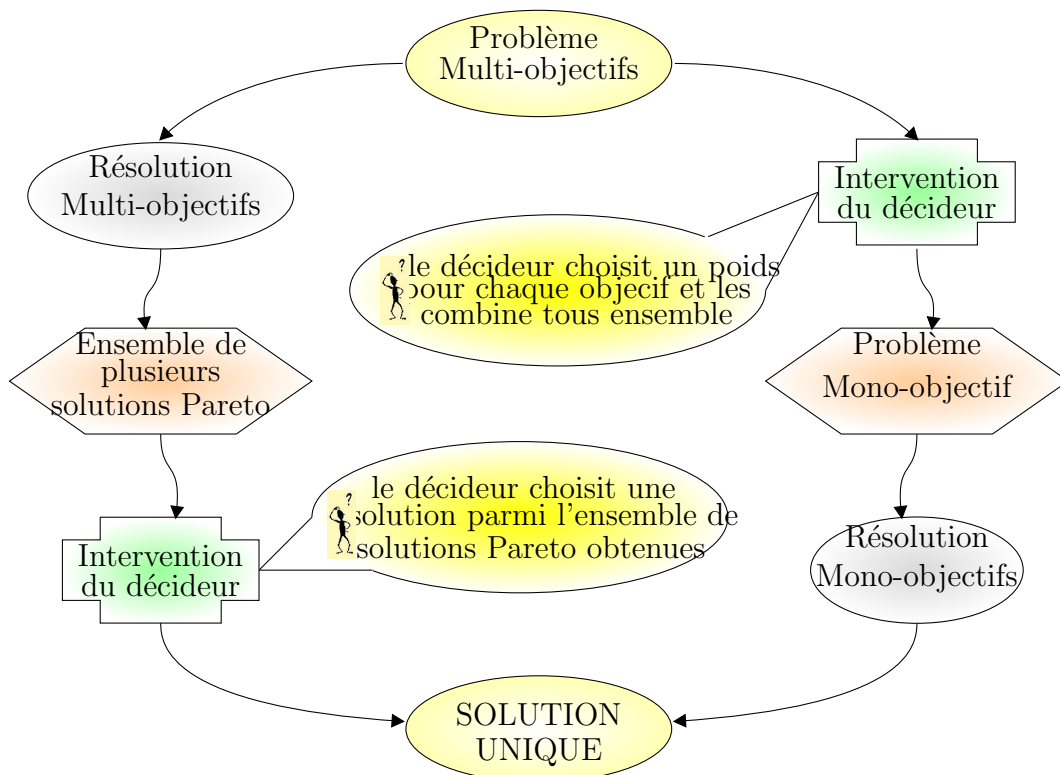


FIGURE 2.1 – Schéma de résolution

2.1.2 Classification point de vue concepteur

Ce classement adopte un point de vue plus théorique articulé autour des notions d'agrégation et d'optimum Pareto. Les approches utilisées pour la résolution de problèmes multi-objectifs peuvent être classées en deux catégories [6] : les approches non Pareto et les approches Pareto (figure 2.2).

1. Les approches non Pareto ne traitent pas le problème comme un véritable problème multi-objectifs. Elles cherchent à ramener le problème initial à un ou plusieurs problèmes mono-objectif.

2. Les approches Pareto ne transforment pas les objectifs du problème, ceux-ci sont traités sans aucune distinction pendant la résolution.

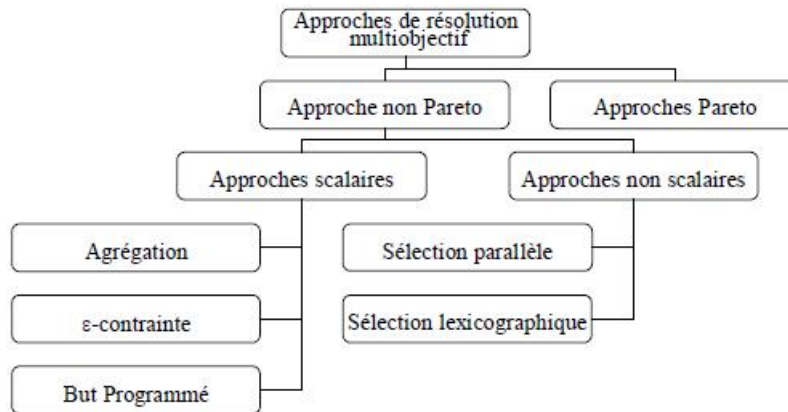


FIGURE 2.2 – classification d’un point de vue concepteur

2.1.2.1 Approches non Pareto

Les approches non Pareto sont classées en deux catégories : les approches scalaires, qui transforment le problème multi-objectifs en problème mono-objectif et les approches non scalaires, qui gardent l’approche multi-objectifs, mais en traitant séparément chacun des objectifs.

▷ **Les approches scalaires** : Ces approches sont de type a priori, à l’origine, les problèmes multi-objectifs étaient transformés en problèmes mono-objectif. Plusieurs approches différentes ont été mises au point pour transformer les problèmes multi-objectifs en problèmes mono-objectif : les approches agrégées, programmation par but, et les approches e-contraintes,...etc.

a) Approche d’agrégation : c’est l’une des premières approches utilisée pour résoudre les problèmes multi-objectifs. Elle consiste à transformer un problème multi-objectifs en un problème mono-objectif, en définissant une fonction objectif unique comme étant la somme pondérée des différentes fonctions objectifs du problème initial.

Fonctions scalarisantes : l’ensemble des méthodes agrégées repose sur l’axiome suivant : tout décideur essaye inconsciemment de maximiser une fonction d’utilité U

$$U = U(z_1, z_2, \dots, z_k)$$

où $U : \mathbb{R}^p \times \Lambda \rightarrow \mathbb{R}$, $\Lambda = \{\lambda \in \mathbb{R}^p : \sum_{k=1}^p \lambda_k = 1, \lambda_k > 0\}$ qui agrège les valeurs des critères pour chaque solution ($\Lambda \subset \mathbb{R}^p$ est l’ensemble des paramètres choisi). Il est possible de dé-

finir une fonction croissante dite fonction scalarisante, qui agrège les valeurs des critères pour chaque solution : $U(Z, \lambda) : \mathbb{R}^p \times \Lambda \rightarrow \mathbb{R}$

Les fonctions d'utilités les plus employées sont :

- Somme pondérée des objectifs (particulièrement utilisée dans le cas linéaire) :

$$U_1(Z, \lambda) = \sum_{k=1}^p \lambda_k Z_k, \lambda \in \Lambda,$$

ou bien,

$$U_2(Z, \lambda) = \sum_{k=1}^p \lambda_k |Z_k - Z_k^{id}|, \lambda \in \Lambda.$$

Où Z_k^{id} est la k^{me} composante du point idéal

- Norme L_p pondérée :

$$U_3(Z, \lambda) = \left[\sum_{k=1}^p \lambda_k |Z_k - Z_k^{id}|^p \right]^{\frac{1}{p}}, \lambda \in \Lambda, p \in \mathbb{Z}_+^*.$$

- Norme L_∞ pondérée de Tchebychev :

$$U_4(Z, \lambda) = \max_{1 \leq k \leq p} \left\{ \lambda_k |Z_k - Z_k^{id}| \right\}, \lambda \in \Lambda.$$

- Norme composée (Tchebychev pondérée augmentée) :

$$U_5(Z, \lambda) = \max_{1 \leq k \leq p} \{ \lambda_k |Z_k - Z_k^{id}| \} + \rho \sum_{k=1}^p \lambda_k |Z_k - Z_k^{id}|, \rho > 0.$$

b) Programmation par but : Dans les approches de ce type, le décideur doit définir des buts T_i ou références qu'il désire atteindre pour chaque objectif z_i . Ces valeurs sont introduites dans la formulation du problème, on le transformant en un problème mono-objectif. La nouvelle fonction objectif est modifiée de façon à minimiser les écarts entre les résultats et les buts à atteindre.

c) Approches ϵ -contraintes : Dans cette approche, le problème consiste à optimiser une seule fonction objectif z_k sujette à des contraintes sur les autres fonctions objectif (Convertir $(p - 1)$ des p objectifs du problème en contraintes).

Conclusion sur Les approches scalaires : Ces différentes approches de résolution 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 à un seul objectif. Mais ces méthodes ont aussi des

inconvenients. Certaines ne peuvent traiter complètement des problèmes non convexes et sont donc très sensibles à la forme du front Pareto. Les autres, bien que pouvant traiter les problèmes non convexes, restent quand même sensibles à la forme du front Pareto. Un autre inconvénient important est qu'il faille relancer plusieurs fois les algorithmes de résolution avec des valeurs différentes 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 référence.

▷ **Les approches non scalaires non Pareto** : ces approches sont de type a posteriori qui ne transforment pas le problème multi-objectifs en un problème mono-objectif, mais utilisent des opérateurs qui traitent séparément les différents objectifs, elles n'utilisent non plus la notion de dominance Pareto : sélection parallèle, sélection lexicographique.

b) Sélection lexicographique Cette approche, proposée par Fouran, 1985 [31], elles classent 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 en intégrant les valeurs obtenues comme contraintes pour la résolution sur des objectifs moins prioritaire et ainsi de suite. La solution obtenue à l'étape k sera la solution du problème. Le risque essentiel de cette méthode est la grande importance attribuée aux objectifs classés en premier. La meilleure solution trouvée pour l'objectif le plus important va faire converger l'algorithme vers une zone restreinte de l'espace d'état et enfermer les points dans une niche.

2.1.2.2 Approches Pareto

Les approches Pareto utilisent directement la notion de dominance dans la sélection des solutions générées. Le principal avantage de ces approches, c'est l'optimisation simultanée d'objectifs contradictoires.

Maintenant que les caractéristiques ainsi les approches de résolution principales de l'optimisation multi-objectifs ont été abordées, la suite de ce chapitre présente quelques méthodes exactes de résolution des problèmes d'optimisation linéaire mono-objectif et multi-objectifs.

2.2 Résolution d'un programme linéaire

La programmation linéaire uni-critère reste une source d'inspiration fondamentale pour le développement de méthodes multi-objectifs, nous allons présenter ici quelques caractéristiques de la solution optimale d'un PL, ainsi que la méthode du simplexe.

2.2.1 Résultats de base

Théorème 6 [57] *l'ensemble des solutions réalisables d'un PL détermine dans l'espace des décisions un ensemble appelé domaine réalisable qui est*

- (i) *soit un ensemble vide ;*
- (ii) *soit un ensemble polyédrique non vide ;*
- (iii) *soit un polyèdre convexe.*

Théorème 7 [57] *S'il existe au moins une solution optimale finie du PL, alors il existe une solution optimale qui est un point extrême*

2.2.1.1 La méthode du simplexe

L'algorithme du simplexe a été développé par George DANTZIG en 1947 et permet de résoudre un programme linéaire sous forme standard.

Le principe de l'algorithme est de démarrer à partir d'une solution de base réalisable, puis se déplacer d'un sommet du convexe vers un autre sommet, le déplacement permet un changement de base et chaque itération de l'algorithme améliore la valeur de la fonction objectif.

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

L'importance de cet algorithme réside dans le fait qu'il permet :

- de détecter si la solution optimale est unique
- s'il existe des solutions optimales multiples
- de détecter qu'il n'existe pas de valeurs finies pour la fonction objectif

Concepts de base

2.2. RÉSOLUTION D'UN PROGRAMME LINÉAIRE

On considère le programme linéaire sous forme standard

$$\begin{aligned} \max Z &= Cx \\ Ax &= b \\ x &\geq 0 \end{aligned}$$

avec A d'ordre $m \times n$, $\text{rang}(A) = m$.

Pour caractériser un sommet et reconnaître son optimalité, il est nécessaire d'utiliser le concept de solution de base pour le système $Ax = b$. On s'impose que $\text{rang}(A) = m$ pour garantir que les équations ne soient pas redondantes.

Notations :

I : ensemble des indices des variables de base

J : ensemble des indices des variables hors base

x_B : variables de base

x_{HB} : variables hors base

$A_B = (a_i)_{i \in I}$: la matrice formée des m vecteurs de base

$A_{HB} = (a_j)_{j \in J}$: la matrice des coefficients des variables hors base

$C_B = (c_j)_{j \in I}$: les coefficients économiques des variables de base

$C_{HB} = (c_j)_{j \in J}$: les coefficients économiques des variables hors base

$A = [A_B, A_{HB}]$, $x = [x_B, x_{HB}]$.

Le programme linéaire devient :

$$\begin{aligned} \max Z &= C_B x_B + C_{HB} x_{HB} \\ A_B x_B + A_{HB} x_{HB} &= b \\ x_B \geq 0, x_{HB} &\geq 0 \end{aligned}$$

x_B est une solution de base $\Rightarrow x_{HB} = 0 \Rightarrow A_B x_B = b$

A_B matrice inversible alors $x_B = A_B^{-1}b$

La solution de base correspondant à la base B du système $Ax = b$ est

$$x_B = A_B^{-1}b, \quad x_{HB} = 0$$

On obtient une solution de base réalisable lorsque $x_B \geq 0$, $x_B \neq 0_{\mathbb{R}^m}$

Résultat fondamental :

Toute solution de base réalisable du système $Ax = b$ est un sommet du polyèdre convexe :

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

Le polyèdre convexe X , possède un nombre fini de points extrêmes.

Théorème 8 [57] Soit X un polyèdre convexe fermé et borné $X = \{x \in \mathbb{R}^n, Ax = b, x \geq 0\}$, ayant N points extrêmes s_1, s_2, \dots, s_N , alors tout points dans X peut être écrit sous la forme d'une combinaison linéaire convexe des points $(s_i)_{1 \leq i \leq n}$, c'est à dire

$$\forall x \in X, x = \sum_{i=1}^N \lambda_i s_i, \quad \sum_{i=1}^N \lambda_i = 1, \quad \lambda_i \geq 0, \forall i = 1, \dots, N$$

Théorème 9 [57]

La fonction objectif d'un programme linéaire atteint son optimum en un point extrême de l'ensemble des solutions réalisables X . De plus si la fonction Z atteint son optimum en plusieurs points extrêmes elle prend, cette même valeur maximale en tout point combinaison linéaire convexe de ses points extrêmes.

On considère le PL sous forme standard

$$\begin{aligned} \max \quad & Z = Cx \\ & Ax = b \\ & x \geq 0 \end{aligned}$$

On peut construire à partir de A deux matrices A_B et A_{HB} pour obtenir une solution de base on annule $(n - m)$ variables.

A partir d'une solution de base réalisable on peut obtenir une autre solution de base réalisable qui améliore la valeur de Z en transformant une variable hors base (variable entrante) et au même temps une variable de base actuelle en une variable hors base (variable sortante) en effectuant un changement de base, dans le cas ou il n'est pas possible d'améliorer la valeur de Z la solution courante est optimale.

Remarque 4 1. Dans chaque changement de base le nombre de variables de base est constant égal à $m = \text{rang}(A)$.

2. Pour appliquer cette méthode il est impératif d'avoir une solution de base de départ.

Représentation de la méthode du simplexe par les tableaux

Pour simplifier et mieux organiser les calculs , on utilise une représentation des différents systèmes par des tableaux du simplexe.

on considère PL

$$(P1) \left\{ \begin{aligned} \max \quad & Z = Cx \\ & Ax = b \\ & x \geq 0 \end{aligned} \right.$$

2.2. RÉSOLUTION D'UN PROGRAMME LINÉAIRE

en transforme le problème (P1) sous forme standard

$$(P1) \Rightarrow \begin{cases} \max & Z = \sum_{j=1}^n c_j x_j \\ & \sum_{j=1}^n a_{ij} x_j = b_i \quad i = 1..m \\ & x_j \geq 0 \quad j = 1..n \end{cases}$$

On suppose que les variables de base sont x_1, \dots, x_m et les variables hors base x_{m+1}, \dots, x_n

On pose :

$$P_0 = \begin{pmatrix} b_1 \\ \cdot \\ \cdot \\ \cdot \\ b_m \end{pmatrix}, \quad P_j = \begin{pmatrix} a_{1j} \\ \cdot \\ \cdot \\ \cdot \\ a_{mj} \end{pmatrix} \quad j = 1..n$$

$$Z_0 = \sum_{i=1}^m c_i x_i \quad \text{et} \quad Z_j = \sum_{i=1}^m a_{ij} c_i$$

ou a_{ij} représente le coefficient de la $j^{\text{ème}}$ variable dans la $i^{\text{ème}}$ contrainte

On donne le premier tableau du simplexe

Base	C_B	P_0	P_1	P_m	P_{m+1}	.	.	.	P_n
x_1	c_1	b_1	1	0	.	.	0	.	0	$a_{1,m+1}$.	.	.	$a_{1,n}$
.	.	.	0	1
.
.	1	.	0
.
.
.	1	0
x_m	c_m	b_m	0	0	0	.	.	.	0	1	$a_{m,m+1}$.	.	$a_{m,n}$
Z_j			C_1	C_m	z_{m+1}	.	.	.	z_n
∇_j			0	0	$C_{m+1} - z_{m+1}$.	.	.	$C_n - z_n$

Les itérations du simplexe représente le passage d'un tableau à un autre tableau avec les critère suivants :

a) Critère d'entrée : La variable hors base x_j candidate à entrer dans la base est celle qui possède le coefficient économique le plus élevé dans la fonction objectif c'est à dire

$$c_r - z_r = \max_{j \in J} \{ \nabla_j = c_j - z_j > 0 \}$$

b) **Critère de sortie** : La variable de base x_s devait quitter la base est celle qui vérifie

$$\frac{b_s}{a_{sr}} = \min\left\{\frac{b_i}{a_{ir}}, a_{ir} > 0, i \in I\right\}.$$

Le changement de base correspond à un renouvellement de tableau soient r et s les indices des variables entrantes et sortantes, a_{ij} l'élément de la $i^{\text{ème}}$ ligne et $j^{\text{ème}}$ colonne du tableau k , $a_{sr}^{(k)}$ l'élément pivot du tableau k

On aura : ligne pivot

$$a_{sj}^{(k+1)} = a_{sj}^{(k)} / a_{sr}^{(k)}.$$

et autre lignes

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - a_{sj}^{(k)} \frac{a_{ir}^{(k)}}{a_{sr}^{(k)}}.$$

Énoncé de l'algorithme du simplexe

Les étapes de l'algorithme du simplexe se résume comme suit :

1. Mettre le PL sous forme standard ;
2. Recherche d'une solution de base réalisable de départ ;
3. Construire le premier tableau du simplexe ;
4. Tester $\nabla_j \leq 0 \quad \forall j \in \{1..n\}$, si oui la solution courante est optimale aller à 8 sinon aller à 5 ;
5. Soit $\nabla_r = \max_{j \in J} \{\nabla_j \mid \nabla_j > 0\}$, la variable x_r entre en base
-tester si $a_{ir} \leq 0 \forall i = 1..n$, si oui, le programme linéaire ne possède pas de solution optimale finie(modèle non borné) aller à 8 sinon aller à 6
6. On détermine la variable sortante x_s par $\frac{b_s}{a_{sr}} = \min\left\{\frac{b_i}{a_{ir}}, a_{ir} > 0, i \in I\right\}$
ou a_{sr} est le pivot aller à 7 ;
7. Obtenir le nouveau tableau du simplexe, on divise la ligne pivot par le pivot, on obtient les autres lignes du tableau en utilisant

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - a_{sj}^{(k)} \frac{a_{ir}^{(k)}}{a_{sr}^{(k)}}, i \neq s, j \in 1, \dots, n + 1;$$

8. Terminer.

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

Les méthodes de résolution de problème ILP¹ à la fois très différentes et plus complexes que celles de PL en variables continues qui cependant souvent utilisées comme sous-programmes.

Dans l'évolution de la programmation linéaire en variables entières ou mixtes, il existe trois étapes successives :

- les méthodes de coupure furent les premières à être proposées, suite principalement aux travaux de R.E. Gomory en 1958 ;
- cette première approche fut ensuite supplantée au début des années 60 par celle du "Branch and Bound" ou procédure par séparation et évaluation. La notion de relaxation d'un problème joue un rôle essentiel dans ce type méthode ;
- l'approche des méthodes de coupure connut vingt ans plus tard une seconde jeunesse dans le cadre de la théorie polyédrale avec la notion d'inégalités valides.

Ce domaine est bien sûr traité avec plus de détail dans les ouvrage spécialisés, en particulier dans [75, 60] et [49].

Soit à résoudre le programme linéaire en nombres entiers suivant :

$$(ILP) \begin{array}{ll} \max & Z = Cx \\ \text{t.q} & x \in D, D = X \cap \mathbb{N}^n \end{array} \quad (2.1)$$

Dont le problème relaxé est : $(LP) \text{Max}\{Z = Cx \mid x \in X\}$.

Définition 27 (Inégalité valide pour un problème (ILP)) Une inégalité $\alpha x \geq \alpha_0$ ou $\alpha x \leq \alpha_0$ avec α vecteur ligne ($1 \times n$) et α_0 une constante, est dite inégalité valide pour D si elle est vérifiée par tout point de D .

Une telle inégalité est donc une contraintes qui définit un demi-espace fermé contenant entièrement D .

Coupe fractionnaire de Gomory [38]

1. ILP :Integer Linear Program

2.3. MÉTHODES DE RÉOLUTION D'UN PROGRAMME LINÉAIRE EN NOMBRES ENTIERS

Définition 28 Soit α un scalaire quelconque, on désigne par :

- $\lfloor \alpha \rfloor$ le plus grand entier inférieur ou égal à α ;
- $\lceil \alpha \rceil$ le plus petit entier supérieur ou égal à α ;
- $\langle \alpha \rangle = \alpha - \lfloor \alpha \rfloor$.

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

Génération de la coupe de Gomory :

Soit le tableau optimal issu de la résolution de (LP) par la méthode du simplexe où la solution optimale est supposée non entière (voir tab2.1). Sans perte de généralités, on suppose qu'à l'optimum, il y a un total de $(m + n)$ variables où m représente le nombre de variables de base notées x_i , ($i = 1, m$) et n représente le nombre de variables hors base notées y_j , ($j = 1, n$).

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

TABLE 2.1 – Tableau optimal associé à la base B.

Où :

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

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

Dans cette méthode, le programme linéaire (LP) est résolu en première étape en utilisant la méthode du simplexe. A l'optimum on a les conditions :

1. $\hat{c} \leq 0$
2. $\hat{b} \geq 0$

qui sont vérifiées. Si de plus :

3. \hat{b} est entier, alors la solution optimale de (LP) est aussi optimale pour (ILP). Sinon, une ou plus des variables de base soumises à l'intégrité sont à valeurs fractionnaires. L'idée principale de l'algorithme de Gomory [38] est de maintenir les conditions (1) et (2) satisfaites et de rajouter des contraintes dites *coupes de Gomory* [38] une par une jusqu'à ce que la troisième condition soit vérifiée. A partir du tableau optimal, choisir une variable

2.3. MÉTHODES DE RÉOLUTION D'UN PROGRAMME LINÉAIRE EN NOMBRES ENTIERS

de base dont la valeur est fractionnaire, soit x_i . La i^{me} équation du tableau est donnée par :

$$x_i + \sum_{j=1}^n \hat{a}_{ij} y_j = \hat{b}_i. \quad (2.2)$$

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

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

$$x_i + \sum_{j=1}^n \langle \hat{a}_{ij} \rangle y_j + \sum_{j=1}^n [\hat{a}_{ij}] y_j = \langle \hat{b}_i \rangle + [\hat{b}_i]$$

Puisque $0 \leq \langle \hat{a}_{ij} \rangle < 1$, ceci implique :

$$x_i + \sum_{j=1}^n [\hat{a}_{ij}] y_j \leq \langle \hat{b}_i \rangle + [\hat{b}_i]$$

Or le membre gauche est entier et $0 \leq \langle \hat{b}_i \rangle < 1$. Par conséquent cette dernière équation implique :

$$x_i + \sum_{j=1}^n [\hat{a}_{ij}] y_j \leq [\hat{b}_i]$$

et en soustrayant cette dernière inéquation de (2.2), on obtient :

$$\sum_{j=1}^n \langle \hat{a}_{ij} \rangle y_j \geq \langle \hat{b}_i \rangle. \quad (2.3)$$

qui est bien valide. De plus (2.3) n'est pas satisfaite par la solution optimale de (LP), donc (2.3) est bien une coupe. En ajoutant une variable d'écart x_s à (2.3), on obtient la coupe de Gomory définie par

$$x_s - \sum_{j=1}^n \langle \hat{a}_{ij} \rangle y_j = -\langle \hat{b}_i \rangle. \quad (2.4)$$

Une fois la coupe est générée, les coefficients de (2.4) sont insérés dans une nouvelle ligne du tableau (2.1).

Pour ce nouveau programme, la condition (1) est toujours satisfaite mais pas la condition (2). On effectue donc une ou plusieurs itérations de l'algorithme dual du simplexe,

2.3. MÉTHODES DE RÉOLUTION D'UN PROGRAMME LINÉAIRE EN NOMBRES ENTIERS

jusqu'à ce que la condition (2) soit satisfaite. Si \hat{b} est entier, la solution courante est optimale pour (ILP), sinon on recommence le processus. Après un nombre fini d'itérations, ou bien on obtient une solution optimale entière, ou bien le problème devient impossible.

Méthode "Branch and Bound"

Le principe de cette méthode est de découper (branch) l'espace initial de recherche en domaines de plus en plus restreints afin d'isoler l'optimum global (principe de séparation). L'algorithme de recherche forme ainsi un arbre (voir la figure 2.3) dont chaque nœud représente une partie de l'espace. Ensuite chaque nœud est évalué de façon à déterminer sa borne (bound) inférieure² en fonction d'un critère d'évaluation. Si cette borne n'est pas meilleure que la solution courante alors la recherche sur ce nœud est stoppée, sinon la séparation continue.

L'efficacité de cette technique dépend essentiellement du choix des critères de séparation et d'évaluation. Un bon choix permettra à l'algorithme de réduire l'arbre de recherche en évitant la construction de certaines branches. Dans le pire des cas, un mauvais choix peut amener l'algorithme à explorer tout l'espace de recherche. L'utilisation de cette méthode reste limitée à des espaces de petites dimensions.

Cette méthode est essentiellement utilisée sur des problèmes discrets. Couplée avec une méthode d'analyse d'intervalle [48] elle peut être également utilisée dans le cadre de problèmes continus avec l'avantage de ne pas exiger la continuité de la fonction et d'éviter que la propagation d'erreurs numériques empêche la validation d'une solution.

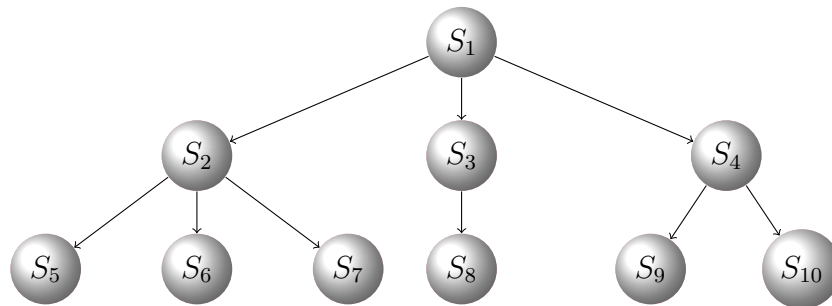


FIGURE 2.3 – Arbre de "Branch and Bound"

2. Inférieure dans le cas d'une minimisation, supérieure dans le cas d'une maximisation.

2.4 Méthodes exactes pour la résolution des problèmes MOLP

De nombreuses extensions de l'algorithme du simplexe ont été proposées pour résoudre les problèmes multi-objectifs parmi lesquelles on cite : Philip, 1972[51] ; Zeleny, 1974 [77] ; Gal, 1977 [36] ; Isermann, 1977 [41] ; Philip, 1977 [52] ; Eker et Kouada, 1975 [27] ; Steuer, 1986 [61] ; Armand et Malivert, 1991 [5] et Trijbsch et al 1975 [69]. Tous ces algorithmes ont pour principe d'énumérer toutes les solutions de base efficaces. Un de leur principaux défauts provient de la nécessité d'énumérer plusieurs fois une même solution si le problème est dégénéré, ce qui est un phénomène courant. D'autres méthodes s'appuient sur le fait que l'ensemble réalisable Y est souvent bien plus petit que l'ensemble admissible X en terme de dimensions et accomplissent donc leur recherche dans l'espace des objectifs citons par exemple Dauer et Fosnaugh, 1995 [23] ; Dauer et saleh, 1990 [20] ; Dauer, 1993 [22] ; Dauer et Gallagher, 1996 [21] et Benson, 1998 [10, 11, 12]. En se basant sur un espace plus petit (en terme de dimensions), ces méthodes ne sont pas affectées par les problèmes de dégénérescence ou d'énumération de solutions équivalentes.

2.4.1 Simplexe multi-objectifs

Une première méthode de résolution des MOLP consiste en une extension de l'algorithme du simplexe servant à résoudre les problèmes mono-objectifs.

Elle repose sur des notions d'adjacence entre les solutions. Soit $G = (V, A)$ le graphe d'adjacence d'un MOLP dont les sommets sont les solutions de base efficaces et les arêtes représentent les pivots permettant de passer d'une solution de base à une autre. Isermann, 1977 [41] a montré que ce graphe est connexe, ce qui signifie qu'il est possible de passer d'une solution efficace à toute autre solution efficace en n'effectuant que des pivots passant par d'autres solutions efficaces.

Cette méthode de simplexe multi-objectifs est décrite dans [26] et permet d'obtenir toutes les solutions efficaces extrêmes et, de façon implicite, toutes les faces efficaces de \mathbb{R}^n et toutes les faces non dominées de \mathbb{R}^p .

2.5 Méthodes exactes pour la résolution des problèmes MOILP

Sur les dernières décennies, plusieurs méthodes ont été développées pour résoudre des problèmes MOILP (MOILP défini dans la page 29), dans le sens de trouver l'ensemble des solutions efficaces entières, ou au moins une partie considérable, comme certaines méthodes nécessitent la présence du décideur (DM) (les méthodes interactives), dont elles génèrent un sous-ensemble de solutions efficaces et d'autres méthodes sans l'intervention du décideur énumèrent toutes les solutions efficaces. En général les approches de résolution peuvent être classées comme exactes ou heuristiques relativement au concept utilisé. Pour les concepts employés dans les algorithmes exacts sont d'une nature diverse, y compris la programmation dynamique (Villarreal et Karwan, 1981 [70]), "Branch and Bound" (Marcotte et Soland, 1986 [47]; Ramesh et al., 1989 [56]), méthodes séquentielles (Klein et Hannan, 1982 [45]; Sylva et Crema, 2004, 2007 [63, 64]), directions de référence (Karaivanova et al., 1995 [43]); les normes pondérées (Eswaran et al., 1995 [28]; Steuer et Choo, 1983 [62]); La somme pondérée avec l'ajout des contraintes (Chalmet et al., 1981 [19]; Ferreira et al., 1994 [30]); et la programmation en 0-1 (Bitran, 1977 [13]). Un état de l'art de méthodes interactives publiées jusqu'à 2007 est présenté dans [2].

De la même façon que dans le contexte mono-objectif, les programmes multi-objectifs en variables entières ou mixtes nécessitent d'énumérer des solutions admissibles. Le cadre de cette thèse étant les programmes non convexes, nous précisons dès maintenant notre propos aux problèmes dont les variables sont discrètes. Dans la suite de ce chapitre, nous nous étendons plus particulièrement sur les deux méthodes de résolution à savoir l'algorithme multi-objectifs de Sylva et Crima et la méthode D. Klein et E. Hannan, car ces deux méthodes et la méthode de Steuer et Choo (méthode basée sur un point de référence), qui sera présentée dans le dernier chapitre, sont centrales dans la proposition algorithmique de la thèse.

2.5.1 Méthode de D. Klein et E. Hannan :

La technique proposée par Klein et Hannan [45] peut être utilisée aussi bien pour identifier l'ensemble de toutes les solutions efficaces que pour en caractériser une partie seulement. Elle consiste à résoudre progressivement une séquence de programmes linéaires mono-objectif en nombres entiers avec des contraintes ajoutées à chaque étape. Les contraintes supplémentaires éliminent les solutions efficaces déjà trouvées, et font en

sorte que les nouvelles solutions générées soient efficaces.

Algorithm 1: L'algorithme de D. Klein et E. Hannan

Étape1 : Résoudre le problème (P_1) défini comme suit (L'indice i est pris arbitrairement dans $\{1, \dots, p\}$) :

$$(P_1) : \text{Max}\{Z^i = c^i x : x \in D\}$$

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

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

Étape générale j : A l'étape j , soit le problème (P_j) à résoudre, il est défini comme suit :

$$(P_j) \begin{cases} \text{max} & Z_i = C^i x \\ \text{t.q} & x \in D \\ & \bigwedge_{k=1}^r (\bigvee_{s=1, s \neq i}^p (C^s x \geq C^i \tilde{x}_k + \epsilon^s)) \end{cases} \quad (2.5)$$

Avec $0 < \epsilon^s \leq 1$ et \tilde{x}_k ($k=1, \dots, r$) les points efficaces obtenus aux étapes $1, \dots, j-1$.

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

Étape finale : La procédure s'arrête lorsque le problème (P_j) est irréalisable.

2.5.2 Méthode de Sylva et Crema (2004)

L'algorithme de Sylva et Crema génère tout les vecteurs non dominés, initialement l'algorithme sélectionne un vecteur poids positive $\lambda > 0$. l'algorithme s'arrête si le problème $(MOILP_\lambda)$ (défini dans la proposition 1) est irréalisable c'est-à-dire le problème

n'a aucune solution efficace. Sinon le modèle $(MOILP_\lambda)$ est mis à jour par l'ajout de p variables binaires et $(p + 1)$ contraintes qui réduit le domaine réalisable par la solution non dominée obtenue par $(MOILP_\lambda)$. Si on a l solutions non dominées, alors on résout le $(MOILP_{\lambda^{(l)}})$ pour trouver $(l + 1)^{\text{ème}}$ solution non dominée.

Proposition 1 [63] Soient x^1, x^2, \dots, x^l des solutions efficaces du problème MOILP. Posons $D_s = \{x \in \mathbb{Z}^n | C^k x \leq C^k x^s, \forall s = 1, \dots, l\}$. Si x^* est une solution optimale pour le problème mono objectif :

$$(MOILP_{\lambda^{(l)}}) \begin{cases} \text{Max} & \sum_{k=1}^p \lambda_k C^k x \\ \text{tq} & x \in (D \cup_{s=1}^l D_s) \end{cases}$$

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

Proposition 2 [63] Soient x^1, x^2, \dots, x^l des solutions efficaces du problème MOILP. Posons $D_s = \{x \in \mathbb{Z}^n | C^k x \leq C^k x^s, \forall s = 1, \dots, l\}$. Si x^* solution efficace pour le problème :

$$(P_l) \begin{cases} \text{Max} & C^k x \\ \text{tq} & x \in (D \cup_{s=1}^l D_s) \end{cases} \quad (2.6)$$

alors x^* est une solution efficace pour le problème MOILP. De plus, si le problème (2.6) est irréalisable, alors $\{(C^k x^s)_{s=1}^l\}$ est l'ensemble de toutes les solutions non dominées du problème MOILP.

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

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

$$(P_{(l+1)}) \begin{cases} \text{Max} & \sum_{k=1}^p \lambda_k C^k x \\ \text{tq} & x \in D \\ & C^k x \geq (C^k x^s + f_k) y_k^s - M_k (1 - y_k^s) \\ & \sum_{k=1}^p y_k^s \geq 1 \quad y_k^s \in \{0, 1\} \text{ pour } k = 1, \dots, p, s = 1, \dots, l \end{cases}$$

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

Algorithm 2: Algorithme de la méthode de Sylva et Crema

Étape1 : Après avoir fixé le vecteur poids $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_p)$ à des valeurs strictement positives et $l = 1$, la première étape de l'algorithme consiste en la résolution du problème :

$$(P_1) : \text{Max} \left\{ \sum_{k=1}^p \lambda_k C^k x \mid x \in D \right\}.$$

Deux cas se présentent :

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

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

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

Après k étapes du processus :

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

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

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

Où $-M_k$ est la borne inférieure pour les valeurs réalisables de la k^{me} fonction objectif (Par exemple, dans le cas où la matrice des critères C est positive, M_k peut être fixée à 0 pour tout k).

Étape générale $(l + 1)$: Résoudre le problème (P_{l+1})

$$(P_{l+1}) \begin{cases} \text{Max} & \sum_{k=1}^p \lambda_k C^k x \\ \text{tq} & x \in D \\ & C^k x \geq (C^k x^s + 1)y_k^s - M_k(1 - y_k^s) \\ & \sum_{k=1}^p y_k^s \geq 1 \quad y_k^s \in \{0, 1\} \text{ pour } k = 1, \dots, p, s = 1, \dots, l \end{cases}$$

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

2.5.2.1 Exemple didactique :

Nous illustrons le fonctionnement de l'algorithme sur un exemple numérique.

Exemple 3 Soit le problème linéaire en nombres entiers à deux objectifs (BOILP) suivant :

$$(P) \left\{ \begin{array}{l} \max \quad z_1 = x_1 + 3x_2 \\ \max \quad z_2 = -3x_1 - x_2 \\ \left\{ \begin{array}{l} -2x_1 + 5x_2 \leq 23 \\ 4x_1 + x_2 \leq 31 \\ x_1 - x_2 \leq 4 \\ -x_1 - 3x_2 \leq -8 \\ -3x_1 - x_2 \leq -8 \\ x_1, x_2 \in \mathbb{Z}_+^* \end{array} \right. \end{array} \right. \quad (2.7)$$

l'espaces de décision et celui des critères sont représentés dans la figure 2.4 On pose $-M_1 =$

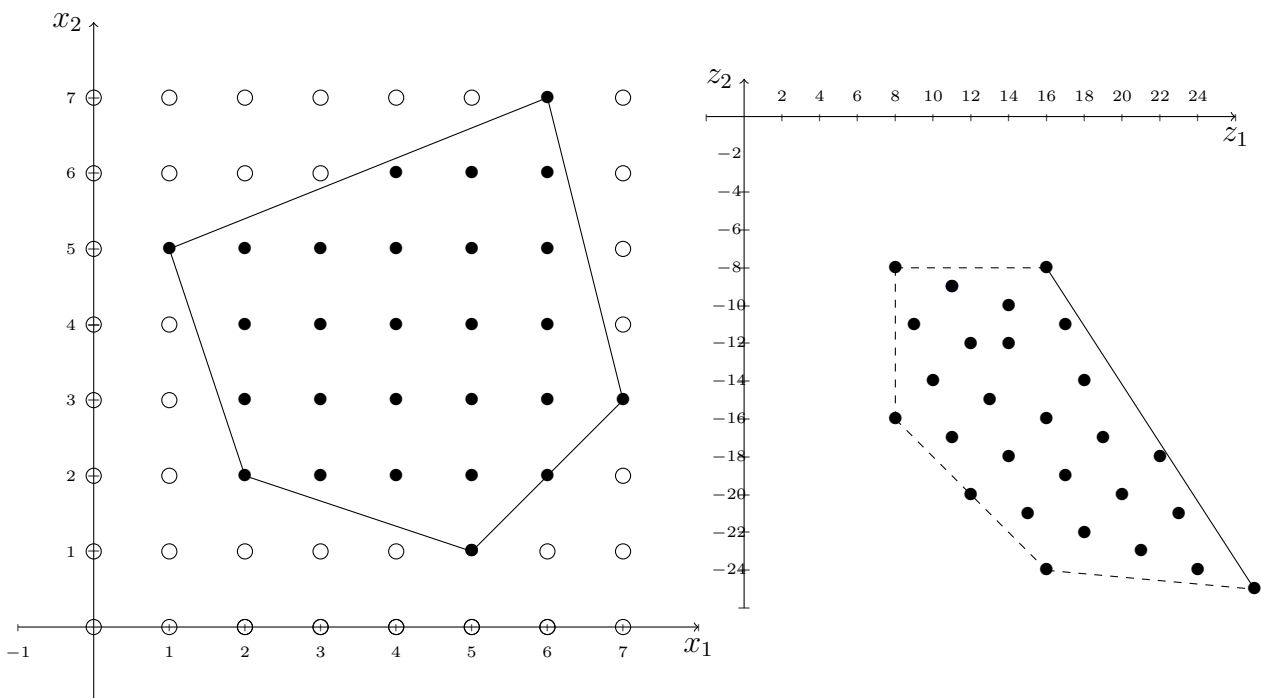


FIGURE 2.4 – Représentation graphique des espaces de décision et des critères de l'exemple 3.

$0, -M_2 = -28$, les bornes inférieures des fonctions objectifs $Z_1, f_k = 1 \forall k$ et $\lambda = (1, 1)'$. La fonction scalarisante est donnée par : $z_\lambda = \lambda' Cx = 1(x_1 + 3x_2) + 1(-3x_1 - x_2) = -2x_1 + 2x_2$

soit

$$(P_0) \left\{ \begin{array}{l} \max \quad z_\lambda = -2x_1 + 2x_2 \\ \left\{ \begin{array}{l} -2x_1 + 5x_2 \leq 23 \\ 4x_1 + x_2 \leq 31 \\ x_1 - x_2 \leq 4 \\ -x_1 - 3x_2 \leq -8 \\ -3x_1 - x_2 \leq -8 \\ x_1, x_2 \in \mathbb{Z}_+^* \end{array} \right. \end{array} \right. \quad (2.8)$$

On résout le problème (P_0) et on obtient $x^1 = (1, 5)'$ de valeur $Z_\lambda(P_0) = 8$. Donc x^1 est une solution efficace du problème (P) correspondant au vecteur non dominé $(16, -8)$ (voir le point encerclé, qui correspond à la solution efficace, dans la figure 2.5) Le deuxième

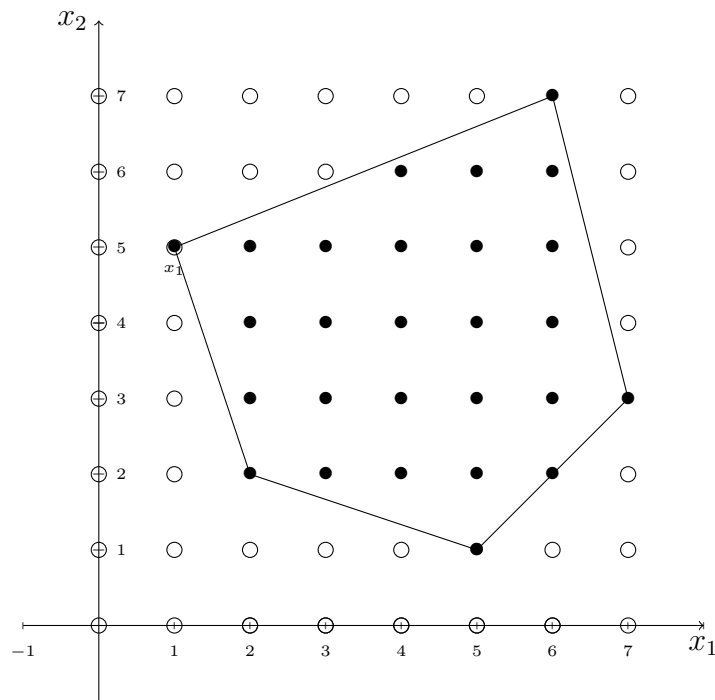


FIGURE 2.5 – Solution optimale de (P_0) .

problème à résoudre est :

$$(P_1) \left\{ \begin{array}{l} \max \quad z_\lambda = -2x_1 + 2x_2 \\ \left\{ \begin{array}{l} (x_1, x_2) \in D_0 \\ x_1 + 3x_2 \geq (16 + 1)y_1^1 - 0(1 - y_1^1) \\ -3x_1 - x_2 \geq (-8 + 1)y_2^1 - 28(1 - y_2^1) \\ y_1^1 + y_2^1 \geq 1 \\ y_1^1, y_2^1 \in \{0, 1\} \end{array} \right. \end{array} \right. \quad (2.9)$$

La résolution de (P_1) donne $x_1 = 2, x_2 = 5, y_1^1 = 1, y_2^1 = 0$ avec valeur optimale $Z_\lambda(P_1) = 6$. Donc, $x^2 = (2, 5)'$ est un autre point efficaces correspondant au couple non dominé $(17, -11)'$, (le point x^2 encerclée voir la figure 2.7) Le troisième problème à résoudre est :

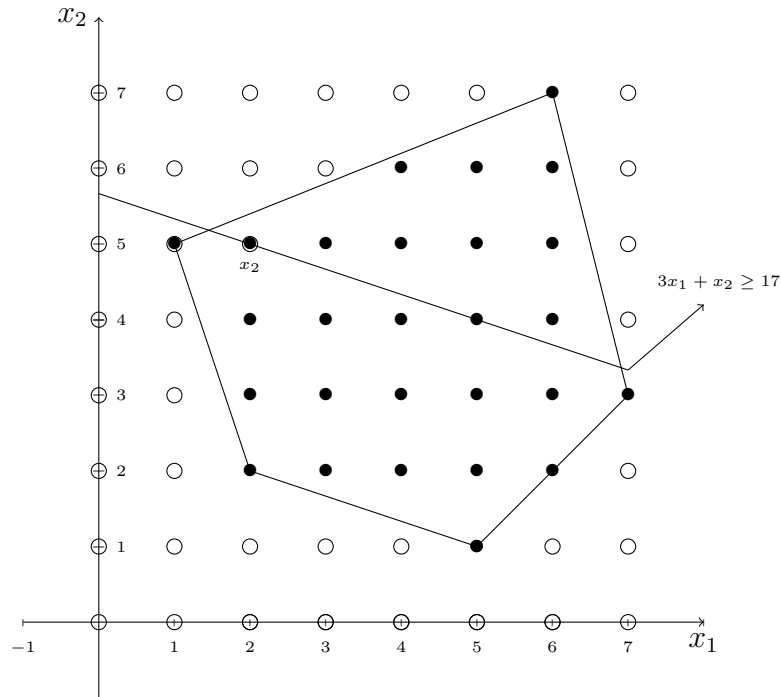


FIGURE 2.6 – Solution optimale de (P_1) .

$$(P_2) \left\{ \begin{array}{l} \max \quad z_\lambda = -2x_1 + 2x_2 \\ D_2 \left\{ \begin{array}{l} (x_1, x_2) \in D_1 \\ x_1 + 3x_2 \geq (17 + 1)y_1^2 - 0(1 - y_1^2) \\ -3x_1 - x_2 \geq (-11 + 1)y_2^2 - 28(1 - y_2^2) \\ y_1^2 + y_2^2 \geq 1 \\ y_1^2, y_2^2 \in \{0, 1\} \end{array} \right. \end{array} \right. \quad (2.10)$$

Une solution optimale de (P_2) est $x^3 = (4, 6)'$, $y_1^1 = y_1^2 = 1$ et $y_2^1 = y_2^2 = 0$ correspondant à $Z_\lambda(P_2) = 4$ et le couple non dominé $(22, -18)'$

On ajoute les cinq contraintes au domaine D_2 et on résout le problème :

$$(P_3) \left\{ \begin{array}{l} \max \quad z_\lambda = -2x_1 + 2x_2 \\ D_3 \left\{ \begin{array}{l} (x_1, x_2) \in D_2 \\ x_1 + 3x_2 \geq (22 + 1)y_1^3 - 0(1 - y_1^3) \\ -3x_1 - x_2 \geq (-18 + 1)y_2^3 - 28(1 - y_2^3) \\ y_1^3 + y_2^3 \geq 1 \\ y_1^3, y_2^3 \in \{0, 1\} \end{array} \right. \end{array} \right. \quad (2.11)$$

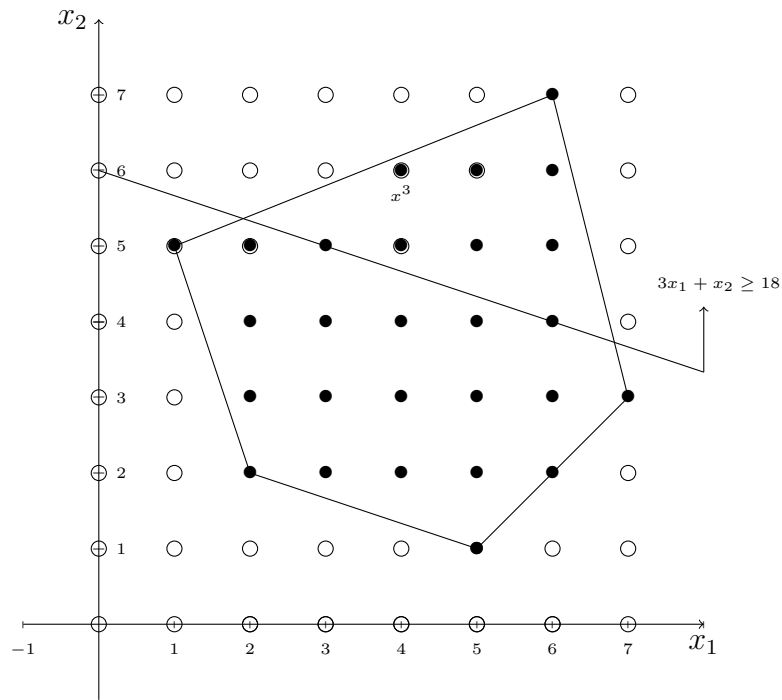


FIGURE 2.7 – Solution optimale de (P_2) , x^3 est la solution optimale.

La solution optimale de (P_3) est $x^4 = (3, 5)$, $y_1^1 = y_1^2 = y_1^3 = 0$ et $y_2^1 = y_2^2 = y_2^3 = 1$ avec $Z_\lambda(P_3) = 4$. Le couple non dominé est $(18, -14)'$. On définit le cinquième problème comme suit :

$$(P_4) \left\{ \begin{array}{l} \max \quad z_\lambda = -2x_1 + 2x_2 \\ D_4 \left\{ \begin{array}{l} (x_1, x_2) \in D_3 \\ x_1 + 3x_2 \geq (18 + 1)y_1^4 - 0(1 - y_1^4) \\ -3x_1 - x_2 \geq (-14 + 1)y_2^4 - 28(1 - y_2^4) \\ y_1^4 + y_2^4 \geq 1 \\ y_1^4, y_2^4 \in \{0, 1\} \end{array} \right. \end{array} \right. \quad (2.12)$$

La solution optimale de (P_4) est $x^5 = (6, 7)'$ correspondant à $Z_\lambda(P_5) = 2$ et le couple non dominé $(27, -25)'$

Le processus continue de la même manière jusqu'à ce que le domaine d'admissibilité devienne vide. L'algorithme s'arrête et donne l'ensemble des solutions efficaces ainsi l'ensemble des solutions non dominés qui leurs correspondants. Ces ensembles sont donnés dans le tableau 2.2. la représentation graphique des points efficaces (supportés et non supportés) ainsi leurs images dans l'espace des critères est donnée dans la figure 4.3

Solutions efficaces	(1,5)	(2,5)	(3,5)	(4,5)	(4,6)	(5,6)	(6,7)	(6,6)
Solutions non dominées	(16,-8)	(17,-11)	(18,-14)	(19,-17)	(22,-18)	(23,-21)	(27,-25)	(24,-24)

TABLE 2.2 – Ensemble des solutions efficaces/non dominées de l'exemple 3

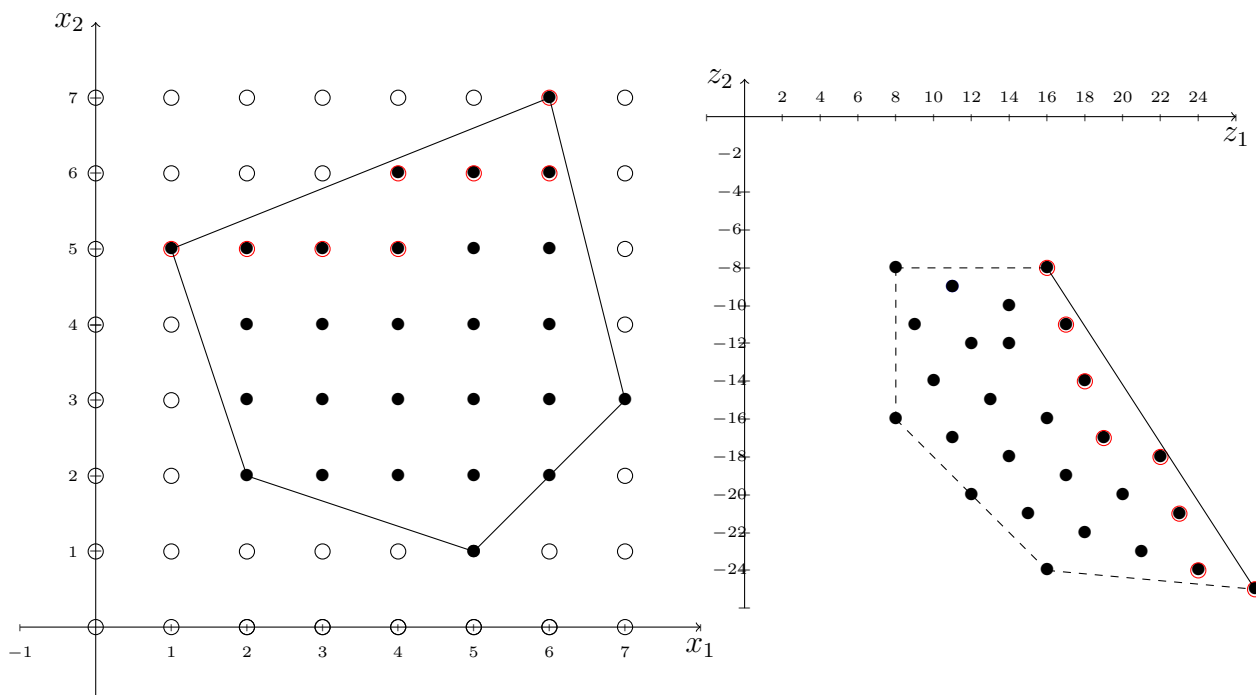


FIGURE 2.8 – Ensemble des solutions efficaces/non dominées de l'exemple 3.

2.6 Conclusion

Dans ce chapitre et en premier temps, nous avons présenté les principales approches de résolution de problèmes multi-objectifs. Puis quelques méthodes exactes de résolution de problème MOILP ont été présentées, à savoir la méthode de D. Klein et E. Hannan(1982) et la méthode de Sylva et Crima (2004)[63], ces dernières utilisent, dans leur processus de résolution, une technique de coupe (d'ajout des contraintes) pour éliminer les solutions dominées ainsi d'éviter les solutions non dominées déjà trouvées. Dans le chapitre suivant, nous allons présentés un état de l'art des méthodes de résolution de problème d'optimisation d'une fonction linéaire sur l'ensemble de solutions efficaces continu et discret du MOLP.

3

Optimisation d'une fonction linéaire sur l'ensemble efficace

Dans ce chapitre, nous présentons quelques résultats de base sur l'optimisation d'un critère linéaire sur l'ensemble des solutions efficaces d'un problème linéaire multi-objectifs en variables continues et discrètes. Quelques algorithmes connus dans le cas continu et discret sont alors décrits, [76] pour le cas continu, celui de Abbas et Chaabane, 2006 [1] et celui de Jorge, 2009 [42] et Chaabane et Pirlot, 2010 [17] pour le cas discret.

3.1 Optimisation d'une fonction linéaire sur l'ensemble efficace continu

3.1.1 Introduction

Depuis une trentaine d'années, le domaine de l'optimisation multi-objectifs connaît une évolution substantielle. Cette évolution a impliqué un développement de plusieurs techniques, ce qui est perçu comme une richesse incontestable de ce domaine. Les problèmes de programmation mathématique à objectifs *multiples* , consiste à optimiser simultanément

3.1. OPTIMISATION D'UNE FONCTION LINÉAIRE SUR L'ENSEMBLE EFFICACE CONTINU

plusieurs critères non comparables sur un ensemble de solutions réalisables non vide. le concept de points efficaces noté X_E joue un rôle important dans l'analyse et la résolution de ces problèmes.

Dans certaines situations, les décideurs n'ont pas besoin de l'ensemble des solutions efficaces d'un problème de programmation multi-objectifs mais uniquement de solutions efficaces qui réalisent l'optimum d'un objectif différent des objectifs déjà fixés. Ceci nous mène vers la recherche de la solution optimale d'un critère sur *l'ensemble de solutions efficaces* du problème multi-objectifs.

Rappelons que un problème linéaire multi-objectifs peut être formulé comme suit

$$\max_{x \in X} c_i^T x \quad i = 1, \dots, p \quad (3.1)$$

avec $c_i \in \mathbb{R}^n \setminus \{0\}$ pour $i = 1, \dots, p$,

$$X = \{x \in \mathbb{R}^n : Ax \leq b, x \geq 0\}. \quad (3.2)$$

A est une matrice de dimension $m \times n$ et $b \in \mathbb{R}^m$

On note par C la $n \times p$ matrice, et X l'ensemble réalisable de problème (3.1), Notons par X_E l'ensemble des solutions efficaces du problème 3.1.

L'une des plus importante et intéressante approche en optimisation multicritère est considérée comme suit :

$$\max\{f(x) : x \in X_E\}, \quad (3.3)$$

avec f une fonction réelle. Sous les hypothèse que X est non vide, alors X_E est non vide, et supposons que $f(x)$ est une fonction linéaire qui peut être défini comme suit $f(x) = \phi = d^t x$. Le problème (3.3) devient

$$(\mathbb{P}_E) : \max\{\phi = d^t x : x \in X_E\},$$

Notons que l'ensemble des *solutions efficaces* X_E est en général non convexe. Le problème (\mathbb{P}_E) rentre alors dans le cadre de l'optimisation global (ou la programmation non convexe).

3.1.2 Notations et résultats de base

- \mathbb{R}^p désigne l'ensemble des vecteurs colonnes réels d'ordre p

3.1. OPTIMISATION D'UNE FONCTION LINÉAIRE SUR L'ENSEMBLE EFFICACE CONTINU

- $\mathbb{R}_+^p = \{x \in \mathbb{R}^p; x \geq 0\}$, $\mathbb{R}_{++}^p = \{x \in \mathbb{R}^p; x > 0\}$;
- \mathbb{R}_p désigne l'ensemble des vecteurs lignes réels d'ordre p
- $\mathbb{R}_{p+} = \{x \in \mathbb{R}_p, x \geq 0\}$, $\mathbb{R}_{p++} = \{x \in \mathbb{R}_p, x > 0\}$;
- e est le vecteur ligne de composantes égales à 1 ;
- $\mathbf{1}$ est le vecteur colonne de composantes égales à 1 ;
- X_v désigne l'ensemble des sommets ou point extrêmes du polyèdre X ;
- X_w : l'ensemble des solutions faiblement efficaces de (MOLP) ;
- \mathcal{Y}_w : l'ensemble des points faiblement non dominés de (MOLP) ;

Définition 29 – Rappelons que l'ensemble $Y = CX = \{y \in \mathbb{R}^p; y = Cx \mid x \in X\}$ est appelé l'image de X ;

- L'ensemble $Y^{\leq} = Y + \mathbb{R}_-^p = \{y \in \mathbb{R}^p; y \leq Cx \mid x \in X\}$ est appelé partie inférieure de Y ;
- L'ensemble $Y^{<} = Y + \mathbb{R}_{--}^p = \{y \in \mathbb{R}^p; y < Cx \mid x \in X\}$ est appelé partie inférieure strictement de Y ;

Définition 30 Pour $\lambda \in \mathbb{R}_{p++}$ et $x \in X$, la fonction définie par $g_\lambda(x) = \max\{\lambda Cx' \mid x' \in X; Cx' \geq Cx\} - \lambda Cx$ est appelé "fonction lacune"

Remarque 5 Soit $x \in X$. $x \in X_E$ si et seulement si $g_\lambda(x) = 0$, et de plus un point x' qui résout $\max\{\lambda Cx' \mid x' \in X; Cx' \geq Cx\}$ appartient à X_E .

Lemme 1 Si $Cx = Cx'$ alors $g_\lambda(x) = g_\lambda(x')$.

Le théorème suivant rappelle quelques caractérisation bien connues de l'ensemble des solutions efficaces X_E du problème linéaire multi-objectifs (MOLP).

Théorème 10 [76]

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

Considérons le problème paramétrique ($P(\lambda)$) suivant :

$$(P(\lambda)) \{ \max \quad \lambda Cx \mid \text{s.c} \quad x \in X \} \tag{3.4}$$

3.1. OPTIMISATION D'UNE FONCTION LINÉAIRE SUR L'ENSEMBLE EFFICACE CONTINU

La deuxième propriété du théorème 10 montre que toute solution efficace de l'ensemble E est une solution optimale du problème paramétrique $(P\lambda)$ pour tout $\lambda \in \Lambda$ tel que :

$$\Lambda = \{\lambda \in \mathbb{R}_{p++} \text{ tel que } e\lambda = 1\}.$$

Soit $x = (x_B, x_N)$ une solution de base réalisable de X ; $A = [B, N]$ et $C = [C_B, C_N]$ sont respectivement des partitions de A et C qui correspondent respectivement aux parties basique et non basique de x .

Lemme 2 .

1. Soit $x = (x_B, x_N) = [B^{-1}b, 0]$ une solution de base réalisable de X , $x \in X_E$ si et seulement si $\lambda(C_N - C_N B^{-1}N) - \nu_B B^{-1}N \leq 0$ pour un certain $\lambda \in \Lambda$ et $\nu_B \in \mathbb{R}_{m+}$ avec $\nu_B x_B = 0$
2. Si $x = (x_B, x_N) = [B^{-1}b, 0]$ une solution de base réalisable de X non dégénérée la propriété (1) de théorème (10) est équivalente à $\lambda(C_N - C_N B^{-1}N) \leq 0$ pour un certain $\lambda \in \Lambda$.
3. Soient c^j et x^j deux colonnes respectivement de C_N et N correspondant à une variable hors base x_j . Si $\lambda(C_N - C_N B^{-1}N) \leq 0$ et $\lambda(c^j - C_N B^{-1}a^j) = 0$ pour un certain $\lambda \in \Lambda$. Alors l'arête obtenue par x_j est efficace.

Le problème (\mathbb{P}_E) de l'optimisation d'une fonction ϕ sur l'ensemble des solutions efficaces X_E de problème $(MOLP)$ a été étudié pour la première fois en 1972 par Philip [51], qui a décrit un algorithme basé sur le déplacement sur les sommets efficaces adjacents dans le cas où la fonction à optimiser est linéaire. Depuis, un bon nombre de chercheurs notamment dans le cas continu ont suivi cette voie. Plusieurs méthodes ont été développées où plusieurs formulations équivalentes au problème (\mathbb{P}_E) ont été proposées. Il existe dans la littérature une diversité d'algorithmes pour résoudre ce problème. Yamamoto, 2002 [76] a classifié ces méthodes en différentes catégories à savoir :

- les algorithmes de recherche de sommets adjacents ;
- les algorithmes de recherche de sommets non adjacents ;
- les algorithmes basés sur la méthode de "branch and bound" ;
- les algorithmes basés sur la méthode de relaxation Lagrangienne ;
- les algorithmes basés sur la méthode duale ;
- les algorithmes basés sur la bisection.

Nous présentons dans cette section quelques algorithmes, nous ne discutons pas leurs avantages et inconvénients parce que nous ne possédons pas assez d'expériences informatique pour les évaluer.

3.1.3 Algorithmes de recherche des sommets adjacents

Les algorithmes proposés par Philip, 1972 [51]; Ecker et Song, 1994 [25]; Fulop, 1994 [35] concernent le cas où la fonction objectif ϕ du problème (P_E) est linéaire. Ceux proposés par Bolintinenu, 1993 [14] et Yamamoto, 2002 [76] optimisent une fonction quasi-convexe sur l'ensemble X_E . Tous ces algorithmes sont basés sur deux techniques : se déplacer d'un sommet efficace à un sommet adjacent efficace par l'intermédiaire d'une arête efficace avec une grande valeur de la fonction ϕ , la deuxième technique consiste à isoler par des coupes les parties de X où la fonction ϕ prend des valeurs plus petites.

Nous allons présentés ici la méthode de Yamamoto, 2002 [76] pour optimiser une fonction ϕ quasi-convexe sur l'ensemble des solutions efficaces E de $(MOLP)$.

3.1.4 La méthode de Yamamoto, 2002[76]

Notons par $H = \{x \in \mathbb{R}^n \mid ax = \alpha\}$ l'hyperplan déterminé par les deux demi-espaces $H_+ = \{x \in \mathbb{R}^n \mid ax \geq \alpha\}$ et $H_- = \{x \in \mathbb{R}^n \mid ax \leq \alpha\}$; H_{++} et H_{--} sont respectivement leurs intérieures.

Soit X_v l'ensemble des points extrêmes (sommets) du polyèdre X . Pour $x, x' \in X_v$ $[x, x']$ dénote l'arête qui relie x et x' . Pour $x \in X_v \cap X_E$, on note par $N_E(x)$ l'ensemble des sommets liés à x par une arête efficace tel que : $N_E(x) = \{x' \mid x' \in X_v \cap X_E; [x, x'] \subset X_E\}$.

Lemme 3 *Si ϕ est quasi-convexe et $x \in X_v \cap X_E$. Supposons que $\{x' \mid x' \in N_E(x); \phi(x') > \phi(x)\} \neq \emptyset$ alors x est un maximum local pour le problème P_E .*

Ce lemme caractérise la solution optimale de P_E si elle existe. Il est utilisé dans l'algorithme ci-dessous pour exécuter le cycle secondaire (Boucle sur k).

L'algorithme de la méthode

Étape $\langle 0 \rangle$ (Initialisation)

- Poser $p = k = 0$, $X^0 = X$ et Chercher $x^0 \in X_v \cap X_E$.
- Si $N_E(x^0) \neq \emptyset$ alors x^0 est optimale pour le problème P_E . Sinon aller à l'étape $\langle p \rangle$.

Étape $\langle 1 \rangle$ (Boucle principale $\langle p \rangle$)

- $\langle p.1 \rangle$ Si $\{x \mid x \in N_E(x^p); \phi(x) > \phi(x^p)\} \neq \emptyset$ choisir x^{p+1} dans cet ensemble et poser $p = p + 1$ et aller à l'étape $\langle p.2 \rangle$.
- $\langle p.2 \rangle$ Sinon. Poser $L^p = \{x \mid \phi(x) \leq \phi(x^p)\}$ et aller à la boucle secondaire $\langle k \rangle$.

Étape $\langle 2 \rangle$ (Boucle secondaire $\langle k \rangle$)

3.1. OPTIMISATION D'UNE FONCTION LINÉAIRE SUR L'ENSEMBLE EFFICACE CONTINU

- $\langle k.1 \rangle$ Chercher $v^k \in \operatorname{argmax} \{ \phi(x) \mid x \in X^k \}$. Si $\phi(x^p) \geq \phi(v^k) - \epsilon$, pour une certaine tolérance fixée $\epsilon > 0$ arrêter avec x^p comme une v -approximation de la solution optimale du problème P_E . Sinon, aller à la sous-étape $\langle k.2 \rangle$.
- $\langle k.2 \rangle$ Chercher un hyperplan H^k de L^p tel que $L^p \subseteq H^k_+$ et $v^k \in H^k_-$.
- $\langle k.3 \rangle$ Si elle existe une arête efficace $[\mu', \mu'']$ tel que $[\mu', \mu''] \cap H^k \neq \emptyset$ et $\max \{ \phi(\mu'), \phi(\mu'') \} > \phi(x^p)$ poser $x^{p+1} = \mu$ avec $\phi(\mu) \equiv \max \{ \phi(\mu'), \phi(\mu'') \}$ et $\mu \in \{ \mu', \mu'' \}$ et poser $p = p + 1$ et aller la boucle principale $\langle p \rangle$. Sinon, aller à aller à la sous-étape $\langle k.4 \rangle$.
- $\langle k.4 \rangle$ Poser $X^{k+1} = X^k \cap H^k_+$, $k = k + 1$ et aller à la boucle secondaire $\langle k \rangle$

L'algorithme génère une séquence de sommets efficaces x^0, x^1, \dots et les polytopes X^0, X^1, \dots tels que $\phi(x^0) < \phi(x^1), \dots$ et $X = X^0 \supseteq X^1 \supseteq \dots$

3.1.5 Algorithmes de recherche des sommets non adjacents

Les algorithmes qui recherchent les sommets adjacents exploitent une étape pour énumérer les sommets efficaces de l'ensemble de décision avec une dimension inférieure. Dans cette section, nous mettons l'accent sur un algorithme de recherche de sommets non adjacents proposé par Benson, 1992 [7] qui évite l'énumération des sommets adjacents.

Nous supposons que $\phi = d^t x$, $d \in \mathbb{R}_n$ ϕ est linéaire et que le problème ($MOLP$) possède $k + 1$ solutions efficaces notées x^0, x^1, \dots, x^k . On pose $\alpha = \max \{ dx^j \mid j = 0, 1, \dots, k \}$, (P^k) dénote le problème défini ci-après pour chercher le couple $(x, \lambda) \in \mathbb{R}^n \times \mathbb{R}_p$ qui vérifie :

$$(P^k) \begin{cases} \lambda Cx & \geq \lambda Cx^j & j = 0, 1, \dots, k \\ x & \in X, \lambda \in \Lambda \\ dx & \geq dx^j & j = 0, 1, \dots, k \end{cases}$$

Lemme 4 Soient $x^0, x^1, \dots, x^k \in X_E$, si (P^k) ne possède pas de solutions alors $x^* \in \operatorname{argmax} \{ dx^j \mid j = 0, 1, \dots, k \}$ est une solution optimale du problème (\mathbb{P}_E).

Remarque 6 Si $(\bar{x}, \bar{\lambda}) \in X \times \Lambda$ satisfait les contraintes $\lambda Cx \geq \lambda Cx^j \quad j = 0, 1, \dots, k$ du problème (P^k), on dit alors \bar{x} est une solution efficace de l'enveloppe convexe x^0, x^1, \dots, x^k . Dans ce sens, le problème (P^k) est équivalent au problème (\mathbb{P}_E).

L'algorithme de la méthode

- $\langle 0 \rangle$ (Initialisation)
Poser $k = 0$; trouver une solution efficace x^0 du problème ($MOLP$) et aller à l'étape $\langle k \rangle$.
- $\langle k \rangle$ (Itération sur k)

3.1. OPTIMISATION D'UNE FONCTION LINÉAIRE SUR L'ENSEMBLE EFFICACE CONTINU

- $\langle k.1 \rangle$ Trouver une solution $(x, \lambda) \in \mathbb{R}^n \times \mathbb{R}_p$ qui vérifie le problème (P^k) . Si elle n'existe pas de solutions $x^* \in \operatorname{argmax} \{dx^j \mid j = 0, 1, \dots, k\}$ est une solution optimale de problème (P_E) . Autrement poser $(\bar{x}^{k+1}, \bar{\lambda}^{k+1})$ la solution trouvée.
- $\langle k.2 \rangle$ résoudre le programme linéaire "Test" suivant

$$(Test^k) \begin{cases} \max & eCx \\ \text{S.c} & x \in X \\ & Cx \geq C\bar{x}^{k+1} \end{cases}$$

Soit \hat{x} une solution de $(Test^k)$. Si $eC\hat{x} = eC\bar{x}^{k+1}$ aller à l'étape $\langle k.3 \rangle$. Sinon aller à l'étape $\langle k.5 \rangle$.

- $\langle k.3 \rangle$ Si \bar{x}^{k+1} est un sommet de X poser $x^{k+1} = \bar{x}^{k+1}$ et $k = k + 1$ et aller à l'étape $\langle k \rangle$. Sinon aller à l'étape $\langle k.4 \rangle$.
- $\langle k.4 \rangle$ Soit F une face de X tel que \bar{x}^{k+1} appartient à l'intérieur de la face F et résoudre le problème linéaire "Face" suivant :

$$(Face^k) \begin{cases} \max & dx \\ \text{S.c} & x \in F \end{cases}$$

Soit x^{k+1} une solution de $(Face^k)$. Poser $k = k + 1$ et aller à l'étape $\langle k \rangle$.

- $\langle k.5 \rangle$ Résoudre le problème $P(\bar{\lambda}^{k+1})$ défini dans (3.4), poser x^{k+1} sa solution optimale, $k = k + 1$ et aller à l'étape $\langle k \rangle$.

□

Concernant l'itération $\langle k.4 \rangle$, il existe plusieurs techniques pour trouver une face. Benson a proposé d'utiliser les variables du dual tel que :

$$F = \{x \mid x \in X, (e + \mu)Cx = v\}.$$

où μ est la variable dual optimale qui satisfait la contrainte $Cx \geq C\bar{x}^{k+1}$ du problème $(Test^k)$ et $v = \max \{(e + \mu)Cx \mid x \in X\}$.

Lemme 5 .

1. $x^k \in X_v \cap X_E$ pour tout $k = 0, 1, \dots$
2. $x^{k+1} \notin \{x^k \mid k = 0, 1, \dots\}$.

Ce lemme montre que toute solution efficace x^k est un sommet de X et que l'algorithme génère une séquence distincte de sommets de l'ensemble X .

3.1.6 Algorithme basé sur le dual

La dualité non convexe est un sujet prometteur dans l'optimisation globale. Dans cette section, nous allons décrire brièvement un algorithme pour résoudre le problème (\mathbb{P}_E) basé sur le concept de la dualité proposé par Tatch et al., 1996 [66].

Soit

$$C^{\leq} = \{z \mid z \in \mathbb{R}^n : Cz \leq 0; c^i z < 0 \text{ pour certains } i = 1, 2, \dots, p\}.$$

Lemme 6 [66] $X_E = X \setminus (X + C^{\leq})$.

Remarquons de ce lemme que l'ensemble X_E des solution efficaces de $(MOLP)$ peut être exprimé comme différence de deux ensembles convexes. Par conséquent le problème (\mathbb{P}_E) peut se reformuler comme suit :

$$(\mathbb{P}_E) \begin{cases} \max & \phi(x) \\ \text{S.c} & x \in X \setminus (X + C^{\leq}). \end{cases}$$

Notons par I_p la matrice $p \times p$ ayant tous ces élément égale à 1. Pour tout paramètre positif s , on définit les trois $p \times p$ matrice C_s , C_s^{\leq} et X_s tel que

$$\begin{aligned} C_s &= (I + sI_p)C; \\ C_s^{\leq} &= \{z \mid C_s z \leq 0\}; \\ X_s &= X \setminus \text{int}(X + C_s^{\leq}). \end{aligned}$$

Avec I désigne la matrice d'identité de dimension $p \times p$, et $\text{int}(X + C_s^{\leq})$ représente l'enveloppe intérieure de $X + C_s^{\leq}$.

Lemme 7 [76] $X_s = \{x \mid x \in X; \exists \lambda \in \mathbb{R}_{p+} : \lambda C_s x \geq \lambda C_s x', \quad \forall x' \in X\}$.

Remarquons que X_s est aussi une différence de deux ensembles convexes.

Lemme 8 [76] *Il existe $\hat{s} > 0$ suffisamment petit tel que $X_s = X_E$ si $0 < s < \hat{s}$.*

Considérons le vecteur ligne $v \in \mathbb{R}_n$ soit

$$\xi(v) = \sup\{\phi(x) \mid x \in X; vx \geq 1\}.$$

Où $\xi(v) = -\infty$ si $\sup\{\phi(x) \mid x \in X; vx \geq 1\} = \emptyset$. Soit H une partie de \mathbb{R}^n , l'ensemble $H^0 = \{v \mid v \in \mathbb{R}^n; vx \leq 1; x \in H\}$ est appelé l'ensemble polaire de H .

Nous supposons que $\text{int}C_s^{\leq} \neq \emptyset$ et ϕ est une fonction concave. Thach et al., 1996 [66] considèrent le problème dual de (\mathbb{P}_E) .

$$(\mathcal{D}_s) \begin{cases} \max & \xi(v) \\ \text{S.c} & v \in X(X + C_s^{\leq})^0 \end{cases}$$

3.1. OPTIMISATION D'UNE FONCTION LINÉAIRE SUR L'ENSEMBLE EFFICACE CONTINU

Théorème 11 (Dualité)[76]

Si $\xi(\mathcal{D}_s)$ est la valeur optimale de la fonction objectif du dual (\mathcal{D}_s) alors $\phi(\mathbb{P}_E) = \xi(\mathcal{D}_s)$

Théorème 12 [76] Le problème dual (\mathcal{D}_s) de (P_E) est équivalent au problème $(\Xi(\gamma))$ tel que

$$(\Xi(\gamma)) \begin{cases} \max & \Xi(\gamma) \\ \text{S.c} & \gamma \in \Gamma. \end{cases}$$

Où $\Gamma = \left\{ \gamma \mid \gamma \in \mathbb{R}_{p+} : \sup_{x \in X} \gamma C_s x \leq 1 \right\}$ et $(X + C_s^{\leq})^0 = \{\gamma C_s \mid \gamma \in \Gamma\}$. Les auteurs de cet algorithme exploite l'approche dual décrite dans les deux théorèmes précédents et proposent la methode suivante :

⟨0⟩ (Initialisation)

Construire le polyèdre $\Gamma^0 \subseteq \Gamma$, le sommet de Γ^0 est facile à énumérer, poser

$k = 0$ aller à l'étape ⟨ k ⟩

⟨ k ⟩ (itération sur k)

– ⟨ $k.1$ ⟩ Résoudre le problème relaxé suivant :

$$\begin{cases} \max & \Xi(\gamma) \\ \text{S.c} & \gamma \in \Gamma^k \end{cases}$$

pour avoir la solution γ^k

– ⟨ $k.2$ ⟩ Résoudre le programme linéaire suivant :

$$\begin{cases} \max & \gamma^k C_s x \\ \text{S.c} & x \in X \end{cases}$$

pour avoir la solution x^k qui est un sommet de Γ^k et poser sa valeur optimale

$$\sigma^k = \gamma^k C_s x^k$$

– ⟨ $k.3$ ⟩ Si $\sigma^k \leq 1$ "qui signifie que $\gamma^k \in \Gamma$ et γ^k est une solution de $\max\{\Xi(\gamma) \mid \gamma^k \in \Gamma^k\}$ " résoudre le problème $\max\{\phi(x) \mid x \in X; \gamma^k C_s x \geq 1\}$. Poser x^* sa solution si elle existe. Arrêter x^* est une solution optimale du problème P_E

– ⟨ $k.4$ ⟩ Si $\sigma^k > 1$ signifie que $\gamma^k \notin \Gamma$. Réduire γ^k à $\gamma^{k+1} = \gamma^k \cap \{\gamma \mid \gamma C_s x^k \leq 1\}$.

Poser $k = k + 1$ et aller à l'étape ⟨ $k.4$ ⟩

□

Puisque Γ est un polyèdre fini, défini par un nombre fini de contraintes, l'algorithme converge vers la solution optimale de (\mathbb{P}_E) et génère une séquence de sommets distincts de X .

3.1.7 Algorithme de recherche de faces efficaces

Dans cette section, nous décrivons un algorithme pour résoudre le problème (\mathbb{P}_E) proposé par Sayin, 2000 [58] basé sur l'énumération des faces efficaces de X .

Soit $x \in \mathbb{R}^n$, $x \in X$, notons par $I(x) = \{i \mid i \in \{1, 2, \dots, n\}; x_i = 0\}$ (i-e l'ensemble des indices où x est égal à 0). Pour $I \subseteq \{1, 2, \dots, n\}$ soit $F(I) = \{x \mid x \in X; x_i = 0 \text{ pour } i \in I\}$. Alors l'ensemble des solutions efficaces X_E peut être décomposé comme suit :

$$X_E = \bigcup_{I \subset \{1, 2, \dots, n\}} (X_E \cap F(I)).$$

Par conséquent, le problème (\mathbb{P}_E) se réduit en une famille de problèmes $(\mathbb{P}_E(I))$ tels que :

$$(\mathbb{P}_E(I)) \begin{cases} \max & \phi(x) \\ \text{S.c} & x \in X_E \cap F(I) \end{cases}$$

Puisque $X_E \cap F(I) \subseteq X \cap F(I) = F(I)$ le problème relaxé de $(\mathbb{P}_E(I))$ dénoté par $(\overline{\mathcal{P}}(I))$ est défini par :

$$(\overline{\mathcal{P}}(I)) \begin{cases} \max & \phi(x) \\ \text{S.c} & x \in F(I) \end{cases}$$

Le principe général de la méthode en question consiste à résoudre une liste finie de problème $(\overline{\mathcal{P}}(I))$. Considérons une solution efficace x^* initiale, au début, la liste se constitue uniquement du problème $(\overline{\mathcal{P}}(\emptyset))$ identique à (\mathbb{P}_E) puisque $F(\emptyset) = X$. Choisir un problème $(\mathbb{P}_E(I))$ et résoudre son problème relaxé $(\overline{\mathcal{P}}(I))$, deux cas sont envisageables :

1. $(\overline{\mathcal{P}}(I))$ est irréalisable, le problème $(\mathbb{P}_E(I))$ sera examiné et retiré de la liste.
2. $(\overline{\mathcal{P}}(I))$ possède une solution optimale x .

(a) $\phi(x) < \phi(x^*)$ le problème $(\mathbb{P}_E(I))$ sera examiné et retiré de la liste.

(b) $\phi(x) > \phi(x^*)$ on distingue deux cas :

(i) $x \in X_E$ le problème $(\mathbb{P}_E(I))$ sera examiné et retiré de la liste.

(ii) $x \notin X_E$ le problème $(\mathbb{P}_E(I))$ sera examiné et retiré de la liste et pour tout indice $k \in \{1, 2, \dots, n\} \setminus I$ le problème $(\mathbb{P}_E(I \cup k))$ est ajouté à la liste.

La question posée dans l'exécution de l'algorithme serait la gestion de la liste des programmes $(\mathbb{P}_E(I))$ comme elle est le cas dans les méthodes "Branch and bound". En particulier, lorsque n est très grand, la liste des problèmes à traiter se développe rapidement et devient très large pour la sauvegarder dans la mémoire. Pour cette raison, l'expérience informatique de cette méthode est limitée dans la taille du problème [76].

3.1.8 Algorithme de la bisection

Phong et Tuyen, 2000 [53] proposent un algorithme basé sur la bisection pour résoudre le problème (\mathbb{P}_E) où la fonction à optimiser $\phi = d^t x$ est linéaire. L'idée de base est d'utiliser le principe de la bisection pour localiser la valeur optimale $\phi(\mathbb{P}_E)$. En effet, l'algorithme débute avec l'intervalle initial $[\ell_0, \mu_0]$ qui contient la valeur $\phi(\mathbb{P}_E)$ puis résoudre le problème $(\mathcal{P}(\alpha)) \equiv \{\text{trouver } x \in X_E \text{ tel que } d^t x \geq \alpha\}$ avec $\alpha = \frac{(\ell_k + \mu_k)}{2}$ et réduire ainsi l'intervalle $[\ell_k, \mu_k]$ à $[\alpha, \mu_k]$ où $(\mathcal{P}(\alpha))$ possède une solution ou bien à $[\ell_k, \alpha]$ si $(\mathcal{P}(\alpha))$ est irréalisable. Ainsi, l'algorithme génère une solution comme ε -approximation de la solution optimale du problème (\mathbb{P}_E) .

Pour $\lambda \in \Lambda$ soit $\sigma(\lambda)$ la valeur de la solution optimale du problème paramétrique défini dans (3.4) tel que

$$\sigma(\lambda) = \max\{\lambda Cx \mid x \in X\}.$$

et

$$\tau_\alpha(\lambda) = \max\{\lambda Cx \mid x \in X; d^t x \geq \alpha\}.$$

Lemme 9 [53]

- (1) $\tau_\alpha(\lambda) = \max\{\lambda Cv \mid v \text{ est un sommet efficace de } X \cap \{x \mid dx \geq \alpha\}\}$.
- (2) $\tau_\alpha(\lambda) \leq \sigma(\lambda)$ pour tout $\lambda \in \mathbb{R}_p$.
- (3) $\tau_\alpha(\cdot)$ est une fonction convexe sur Λ .
- (4) $\tau_\alpha(\lambda)$ est une fonction non croissante en $\alpha \in \mathbb{R}$.

Théorème 13 [53]

- (a) $X_E \cap \{x \mid d^t x \geq \alpha\} \neq \emptyset$ si et seulement si $\tau_\alpha(\lambda) = \sigma(\lambda)$ pour un certain $\lambda \in \Lambda$.
- (b) $\tau_\alpha(\lambda) = \sigma(\lambda)$ pour un certain $\lambda \in \Lambda$ si et seulement s'il existe $(\bar{\lambda}, \bar{\mu})$ de epi σ tel que $\bar{\mu} = \sigma(\bar{\lambda})$.

Où epi σ dénote l'épigraphe de la fonction σ tel que :

$$\text{epi } \sigma = \{(\lambda, \mu) \mid (\lambda, \mu) \in \Lambda \times \mathbb{R} : \sigma(\lambda) \leq \mu\}.$$

Ce théorème montre l'existence d'une solution optimale pour le problème (\mathbb{P}_E) .

Notons par ω l'ensemble non vide des sommets de $X_E \cap X_v$, X_v étant l'ensemble des points extrêmes de X tels que :

$$\sigma_\omega(\lambda) = \max\{\lambda Cv \mid v \in \omega\}.$$

3.1. OPTIMISATION D'UNE FONCTION LINÉAIRE SUR L'ENSEMBLE EFFICACE CONTINU

Alors pour tout $\lambda \in \Lambda$ on a : $\sigma_\omega(\lambda) \leq \sigma(\lambda)$.

La technique principale utilisée dans l'algorithme est de générer l'ensemble des sommets de l'ensemble $\text{epi } \sigma_\omega \cup \bar{v}$. Notons que σ_ω est représenté par un nombre fini de contraintes d'inégalité et chacune correspond à un sommet de ω tel que :

$$\text{epi } \sigma_\omega = \{(\lambda, \mu) \mid (\lambda, \mu) \in \Lambda \times \mathbb{R} : (\mu - \Lambda C v) \geq 0 \quad v \in \omega\}.$$

Nous supposons que l'ensemble des sommets de l'ensemble σ_ω est connu. Il s'agit de trouver un sommet \bar{v} de X par la maximisation de $\bar{\lambda} C x$ sur l'ensemble X , ce sommet ajoutera une contrainte d'inégalité $\mu - \bar{\lambda} C \bar{v} \geq 0$ ce qui éliminera le sommet $(\bar{\lambda}, \bar{\mu})$ de $\text{epi } \sigma_\omega$. Pour générer l'ensemble de sommets de $\text{epi } \sigma_\omega \cup \bar{v}$, l'algorithme produit uniquement les sommets de $\text{epi } \sigma_\omega \cap \{(\lambda, \mu) \mid \mu - \lambda C \bar{v} = 0\}$.

Pour une tolérance $\varepsilon > 0$. Après un nombre k de bisections on obtient l'intervalle $[\ell_k, \mu_k]$ tel que (P_{μ_k}) ne possède pas de solution, tant que (P_{ℓ_k}) possède une solution avec $\bar{\lambda} \in \Lambda$ sur laquelle σ coïncide avec τ_{ℓ_k} . On résout le problème $\max\{\bar{\lambda} C x \mid x \in X : dx \geq \ell_k\}$ pour avoir une solution x^* considérée comme ε -approximation de la solution optimale du problème (P_E) , i-e $x^* \in X_E$ et $dx^* \geq dx - \varepsilon$ pour tout $x \in E$.

Les auteurs illustrent le fonctionnement de la méthode par un exemple bi-critère ($p = 2$) avec 3 variable ($n = 3$) et 4 contraintes ($m = 4$), la méthode requiert 11 itérations pour $\varepsilon = 0.01$.

3.1.9 Autres méthodes

Dans des contextes des méthodes basées sur la recherche de sommets adjacents, Ecker et Song, 1994 [25] ont développé un algorithme dans le cas où la fonction ϕ à optimiser est linéaire en utilisant une technique de pivotage. En effet, étant donné un point efficace x' , cette technique consiste à trouver les arêtes efficaces adjacentes à x' . s'il n'y a pas d'arêtes efficaces et x' conduit à l'amélioration de la valeur de $\phi(\mathbb{P}_E)$ alors x' est une solution optimale locale pour (\mathbb{P}_E) . La technique de pivotage utilisée permet de surmonter ce problème d'optimum local. Afin de réduire le domaine réalisable X , les auteurs ont proposé de résoudre un problème réduit \overline{MOLP} tel que

$$(MOLP) \begin{cases} \max Cx \\ \text{s.c} \quad x \in \bar{X} \end{cases}$$

où $\bar{X} = \{x \in X \mid dx \geq dx'\}$ avec x' est une solution efficace déjà trouvée. A chaque itération de l'algorithme en question, une solution x est testée pour l'efficacité en résolvant un programme linéaire simple qui a été développé par Ecker et Kouada, 1978 [24]. Un

exemple illustratif de la méthode se trouve dans le chapitre 4 de la référence [16].

White, 1996 [73] considère le problème (P_E) avec une fonction objectif ϕ linéaire et propose plusieurs formulations équivalentes au problème (P_E) . Dauer [23] considère le même problème avec ϕ quasi-convexe et ils se sont intéressés à résoudre une forme équivalente au problème basée sur la relaxation Lagrangienne en combinant la fonction objectif ϕ avec un multiplicateur Lagrangienne et la fonction lacune $g(x)$ telle qu'elle est définie dans la définitions 30 page 63.

Benson et Lee, 1996 [8] proposent un algorithme pour optimiser une fonction ϕ semi-continu sur l'ensemble des solution efficaces d'un problème bi-critère. Thoai, 2000[67] considère le cas où $\phi(x) = \varphi(\phi(x))$ et développe un algorithme approximatif.

Horst et Thoai, 1999 [39]; et Thoai, 2001 [68] ont proposé une variante de "Branch and Bound" pour le problème (\mathcal{P}_{E_w}) (i-e l'optimisation sur l'ensemble des solutions faiblement efficaces) où la fonction objectif ϕ est supposée concave. Les auteurs ont proposé une formulation équivalente a ce problème, l'idée de base de cet algorithme est de décomposer l'ensemble réalisable dans l'espace des critères en un nombre fini de cônes.

Horst et al., 2007 [40] ont proposé une forme équivalente au problème (P_E) qui peut être représenté par une contrainte convexe renversée de la forme $g(z) \leq 0$, où g est une fonction concave. Cette fonction est considérée comme prolongation de la fonction objectif ϕ où différentes formes des problème linéaire multi-critères ont été discuté, y compris le problème de flot maximum comme exemple.

3.2 Optimisation d'une fonction linéaire sur l'ensemble efficaces discret

3.2.1 Introduction

Comme nous l'avons constatés de la section précédente, le problème de l'optimisation d'une fonction linéaire sur l'ensemble des solutions efficaces du problème (*MOILP*) a suscité une intention remarquable. Néanmoins, la majorité des méthodes proposées sont basées sur la transformation du problème (\mathbb{P}_E) en formulations équivalentes qui peut modifier sa propre structure. Malheureusement, le cas discret n'a pas reçu une telle importance, à nos connaissances il n'y a que trois méthodes exactes qu'ont été proposés à savoir : la méthode de Abbas et Chaabane, 2006 [1] ; la méthode de Jorge, 2009 [42] et celle de Chaabane et Pirlot, 2010 [17]. Cette dernière est une combinaison des deux premières méthodes. Dans ce chapitre, nous les présentons ainsi que quelques résultats de base et notions préliminaires pour justifier leur fonctionnement.

3.2.2 Notations et notions préliminaires

Rappelons qu'un problème (*MOILP*) peut être formulé par :

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

avec $\mathcal{D} = X \cap \mathbb{Z}$, \mathbb{Z} étant l'ensemble des nombres entiers relatifs. X est supposé borné et convexe $X = \{x \in \mathbb{R}^n \mid Ax \leq b, \}$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, C est une matrice de dimension $p \times n$ d'éléments entiers, et ses vecteurs lignes $c^i \in \mathbb{R}^n$, $i = 1, 2, \dots, p$.

Le problème de l'optimisation d'une fonction linéaire (\mathbb{P}_E) sur l'ensemble des solutions efficaces E de (*MOILP*) est :

$$(\mathcal{P}_E) \begin{cases} \max & \phi = d^t x \\ \text{s.c} & x \in X_E \end{cases} \quad (3.6)$$

Le problème relaxé de (\mathbb{P}_E) est (\mathcal{P}_R) défini par :

$$(\mathcal{P}_R) \begin{cases} \max & \phi = d^t x \\ \text{s.c} & x \in D \end{cases} \quad (3.7)$$

Soit le problème mono-critère ($\mathcal{P}_i(\mathcal{D})$), $i \in \{1, \dots, p\}$

$$(\mathcal{P}_i(\mathcal{D})) \begin{cases} \max & c^i x \\ \text{s.c} & x \in \mathcal{D} \end{cases} \quad (3.8)$$

Le problème $(P_i(\mathcal{D}))$ peut avoir plusieurs solutions optimales, nous rappelons la notion de solution alternative dans la définition suivante.

Définition 31 Soit x^* une solution optimale du problème $(P_i(\mathcal{D}))$, une solution réalisable $\bar{x} \in \mathcal{D}$ est dite solution alternative à x^* si $C^i x^* = C^i \bar{x}$.

Considérons le problème paramétrique (\mathcal{P}_λ) :

$$(\mathcal{P}_\lambda) \begin{cases} \max & z_\lambda = \sum_{i=1}^p \lambda_i c^i x, \quad i = 1, 2, \dots, p \\ \text{s.c} & x \in \mathcal{D} \end{cases} \quad (3.9)$$

Où λ est un vecteur de

$$\Lambda = \left\{ \lambda \in \mathbb{R}^p, \sum_{i=1}^p \lambda_i = 1, \lambda_i \geq 0, i = 1, 2, \dots, p \right\}$$

Théorème 14 Si \hat{x} est une solution optimale du problème paramétrique (P_λ) pour un certain $\lambda > 0$, alors \hat{x} est efficace pour le problème (MOILP).

La réciproque de ce théorème n'est pas vraie, puisque l'espace de décisions \mathcal{D} n'est pas convexe. Sur le front Pareto, deux types de solutions peuvent être différenciées : les solutions supportées et les solutions non supportées. Les premières sont celles situées sur l'enveloppe convexe de l'ensemble des solutions et peuvent donc être énumérées à l'aide d'une agrégation linéaire des objectifs en résolvant le problème paramétrique $(\mathcal{P}(\lambda))$. Par contre, les solutions non supportées sont réparties à l'extérieur du convexe du front Pareto et ne peuvent être caractérisées pour aucun $\lambda > 0$ (Voir l'exemple introduit par Bowman, 1976 [15]).

3.2.3 Test d'efficacité

Dans les deux algorithmes décrits dans cette section, à chaque itération, on est appelé à tester l'efficacité d'une solution réalisable x par la résolution d'un problème de programmation linéaire $(\mathcal{P}(x))$ développé par Ecker et Kouada, 1978 [24] dont les paramètres restent inchangés, sauf le vecteur second membre des contraintes, qu'il faut remplacer par le vecteur critère de la solution réalisable en question. Son principe est simple est décrit dans le théorème suivant.

Théorème 15 [24] Soit x^* une solution réalisable de \mathcal{D} , x^* est une solution efficace du (MOILP) si et seulement si la valeur optimale de la fonction objectif Θ^* est nulle dans

3.2. OPTIMISATION D'UNE FONCTION LINÉAIRE SUR L'ENSEMBLE EFFICACES DISCRET

le programme linéaire suivant :

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

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 $\exists x \in \mathcal{D}$ 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 $x \in \mathcal{D}$ 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 $\Theta^* = 0$.

□

3.2.4 Méthodes de résolution

3.2.4.1 Méthode de Abbas et Chaabane[1]

Les auteurs ont développé une méthode exacte pour résoudre le problème (\mathbb{P}_E) dans l'espace des critères, dont la fonction objectif est donnée par une somme pondérée des critères du (MOILP). Le principe de la méthode est le suivant : Dans une première étape la condition nécessaire du théorème de Geoffrion est utilisée pour déterminer une solution efficace initiale en résolvant le problème paramétrique (\mathcal{P}_λ), en chaque itération k , la region d'admissibilité est réduite en ajoutant des contraintes qui se traduit par deux types de coupes éliminant les solutions dominées par la solution efficace courante, pour éliminer toutes les solutions moins bonnes sur le critère principal en résolvant un problème paramétrique (\mathcal{P}_λ^k).

Notations et résultats préliminaires

Dans cette méthode nous adopterons les notations suivantes :

- z_1, z_2, \dots, z_p dénotent les critères initiaux du (MOILP)
- x_{opt} est la solution optimale du problème (\mathbb{P}_E).
- ϕ_{opt} est la valeur optimale de ϕ .
- $z_i(x_{opt})$ est l'évaluation du critère i en la solution x_{opt} .
- $H^0 = \mathcal{D} = S \cap \mathbb{Z}^n$.

Pour $k \geq 1$ on a :

3.2. OPTIMISATION D'UNE FONCTION LINÉAIRE SUR L'ENSEMBLE EFFICACES DISCRET

- I_k : l'ensemble des indices de base du tableau optimal de l'itération k .
- N_k : l'ensemble des indices hors-base du tableau optimale de l'itération k .
- D_k : la région tronquée à l'itération k .
- E_{j_k} : l'arête incidente à x_j à l'étape k .

Soit le problème (\mathcal{P}_λ^k) défini par :

$$(\mathcal{P}_\lambda^k) \begin{cases} \max & z_\lambda = \sum_{k=1}^p \lambda_k z_k(x), \quad k = 1, 2, \dots, p \\ \text{s.c} & x \in \mathcal{D}^k \end{cases}$$

Où $\lambda > 0$ et \mathcal{D}^k est le domaine réduit défini par :

$$\mathcal{D}^k = \left(\mathcal{D}^k \setminus \bigcup_{r=0}^{k-1} \Delta_r \right) \cap \mathcal{D}_{x_{opt}}.$$

où $\mathcal{D}_{x_{opt}} = \{x \mid x \in \mathcal{D}; dx \geq \phi_{opt} + 1\}$ et $\Delta_r = \{x \mid x \in \mathbb{Z}; Cx \leq Cx_r\}$ avec x^0, x^1, \dots, x^{k-1} des solutions efficaces obtenues en résolvant les problèmes $(\mathcal{P}_\lambda^0), (\mathcal{P}_\lambda^1), \dots, (\mathcal{P}_\lambda^{k-1})$. Ceci peut être traduit mathématiquement par les contraintes additionnelles suivantes :

$$H^k = H^{k-1} \cap \left\{ \begin{array}{l} x \in D \mid z_i(x) \geq z_i(x_{opt} + 1)y_i^k - M_i(1 - y_i^k), \quad \forall i = 1, 2, \dots, p \\ \sum_{i=1}^p y_i^k \geq 1, y_i^k \in \{0, 1\}, \quad \forall i = 1, 2, \dots, p \end{array} \right\}$$

Où $-M_i$ est la borne inférieure de la i ème fonction objectif dans le domaine \mathcal{D} et à chaque critère z_i nous associons une variable binaire y_i^k définie par :

$$y_i^k = \begin{cases} 1 & \text{si le critère } z_i \text{ est strictement amélioré par rapport à } z_i(x_{opt}); \\ 0 & \text{sinon.} \end{cases}$$

Coupes de type I

L'algorithme proposé par les auteurs est basé principalement sur l'exploration des arêtes incidentes à une solution trouvée et l'utilisation de coupes éliminant une arête contenant des solutions admissibles au lieu d'un seul point admissible.

Proposition 3 [1] Soit x_k une solution optimale du problème uni-critère $(\mathcal{P}_1(\mathcal{D}_k))$. Supposons que $j_k \in N_k$. Une arête E_{j_k} incidente à la solution x_k est défini par l'ensemble

$$E_{j_k} = \left\{ \begin{array}{l} | x_i = x_{k,j} - \theta_{j_k} y_{k,i,j_k}, i \in I_k \\ x_i \in \mathcal{D}_k \mid x_{j_k} = \theta_{j_k} \\ | x_\alpha = \forall \alpha \in N_k \setminus \{j_k\} \end{array} \right\}.$$

Où \mathcal{D}_k est la région tronquée à l'itération k , $x_{k,i}$ est la i ème composante de la solution x_k , y_{k,i,j_k} est l'élément de la ligne i de la colonne j_k dans le tableau optimale de x_k et θ_{j_k} est un entier positif tel que $0 < \theta_{j_k} \leq \min_{i \in I_k} \left\{ \frac{x_{k,i}}{y_{k,i,j_k}}, y_{k,i,j_k} > 0 \right\}$ et $\theta_{j_k} \times y_{k,i,j_k}$ est un vecteur entier $\forall i \in I_k$ si de tels vecteurs existent.

Théorème 16 [1] Une solution réalisable entière du problème $(\mathcal{P}_1(\mathcal{D}_k))$ qui n'est pas sur l'arête E_{j_k} et incident à x_k de la région tronquée \mathcal{D}_k , se situe dans le demi espace fermé

$$\sum_{j \in N_k \setminus j_k} x_j \geq 1.$$

Ce théorème montre que la coupe de type *I* peut être considérée comme une généralisation de la coupe de Dantzig. L'avantage d'utiliser cette coupe est de tronquer toutes les solutions réalisables entières du problème $(\mathcal{P}_1(\mathcal{D}_k))$ qui se trouvent sur l'arête E_{j_k} issue de la solution optimale x_k , tandis que, la coupe de Dantzig ne tronque qu'un seul point du domaine d'admissibilité à savoir le point x_k .

Coupes de type II

Après avoir trouvé une nouvelle solution efficace de $(MOILP)$, qui améliore la meilleure valeur de ϕ obtenue jusqu'à présent notée ϕ_{opt} , nous imposons une coupe dite coupe de type *II* pour chercher une nouvelle solution efficace dont la valeur ϕ est meilleure ou égale à ϕ_{opt} .

$$d^t x \geq \phi_{opt}$$

3.2.4.2 L'algorithme de la méthode

Étape 1

- Résoudre le problème relaxé (\mathcal{P}_R) défini dans (3.7) et soit x^* une solution optimale de ce problème.
- Si x^* est efficace, terminer x^* est optimale pour le problème (\mathbb{P}_E)
- Sinon, aller à l'étape (2)

Étape 2

- Poser $\phi_{opt} = -\infty$ et résoudre un problème uni-critère $(P_i(\mathcal{D}))$, $i \in \{1, \dots, p\}$, par exemple, prendre $i = 1$ est poser sa solution x^1 .

2.1 Si $J_1 = \{j \in N_1 \mid Z_{1,j}^1 - C_j^1 = 0\} = \emptyset$ alors la solution optimale trouvée est unique et elle est efficace, poser $\phi_{opt} = dx^1$, $x_{opt} = x^1$ et aller à l'étape (3)

2.2 Si $J_1 \neq \emptyset$, alors la solution optimale x^1 peut ne pas être unique, tester son efficacité, si'elle n'est pas efficace, poser $\phi_{opt} = dx^1$, $x_{opt} = x^1$ et aller à l'étape (3)

Étape 3

- Poser $k=1$ et exécuter les sous étapes suivantes

3.1 Construire l'ensemble $\Gamma_k = \{j \in N_k \mid Z_{1,j}^k - C_j^1 x_j \geq 0 \text{ et } \phi_j^k - d_j \leq 0\}$

- Si $\Gamma_k = \emptyset$ ajouter la coupe de Dantzig $\sum_{j \in N_k} x_j \geq 1$ et aller à l'étape (3.2)

3.2. OPTIMISATION D'UNE FONCTION LINÉAIRE SUR L'ENSEMBLE EFFICACES DISCRET

– Sinon, soit $\gamma = \Gamma_k$ et aller à l'étape (a)

(a) Si $\gamma = \emptyset$, alors, soit $j_k \in \Gamma_k$, appliquer la coupe de type I $\sum_{j \in N_k \setminus \{j_k\}} \geq 1$ et aller à l'étape (3.2). Sinon, sélectionner $j_k \in \gamma$ et calculer $\theta_{j_k}^0$ la partie entière de

$$\min_{i \in I_k} \left\{ \frac{x_{k,i}}{z_{k,i,j_k}} \mid z_{k,i,j_k} > 0 \right\}.$$

– Si $\theta_{j_k}^0$ alors il n'y a aucune solution réalisable entière sur l'arête E_{j_k} , poser $\gamma := \gamma \setminus \{j_k\}$ et aller à l'étape (a);

– Sinon, si $\theta_{j_k}^0 > 1$ aller à l'étape (b).

(b) Si x^k est efficace et $dx^k \geq \phi_{opt}$ calculer β_k tel que

$$\beta_k = (d_{j_k} - \sum_{i \in I_k} d_i z_{k,i,j_k}).$$

– Si $\beta_k \neq 0$ aller à l'étape (C);

– Sinon, poser $\gamma := \gamma \setminus \{j_k\}$ et aller à l'étape (a);

– Si x^k n'est pas efficace ou $dx^k < \phi_{opt}$ aller à l'étape (c) (L'arête E_{j_k} doit être explorée quelle que soit la valeur de β_k).

(c) Explorer l'arête E_{j_k} , en cherchant des solutions réalisables de $(P(\mathcal{D}_k))$ correspondant à θ et tester pour l'efficacité à partir de $\theta = \theta_{j_k}^0$ jusqu'à $\theta = 1$ (tel que $\theta \in \mathbb{Z}_+^*$). Dès qu'une solution efficace \bar{x}^k vérifie $d\bar{x}^k > \phi_{opt}$ est trouvée pour une valeur de θ , remplacer x_{opt} par \bar{x}^k et ϕ_{opt} par $d\bar{x}^k >$ et aller à la sous-étape (3.2), s'il n'y a aucune solution efficace entière sur l'arête E_{j_k} , poser $\gamma := \gamma \setminus \{j_k\}$ et aller à l'étape (a).

3.2 Soit $k = k + 1$. La nouvelle région tronquée \mathcal{D}_k est obtenue comme sous-ensemble de \mathcal{D}_{k-1} an appliquant la coupe de type II ($dx \geq \phi_{opt}$) puis en utilisant la méthode dual de simplex et les coupes Gomory autant de fois nécessaire pour trouver une nouvelle solution optimale x^k , poser $x_{opt} = x^k$ et $\phi_{opt} = dx^k$ et aller à la sous-étape (3.1).

3.3 Soit $k = k + 1$. La nouvelle région tronquée \mathcal{D}_k est obtenue comme sous-ensemble de \mathcal{D}_{k-1} (ou \mathcal{D} si $k = 1$) an appliquant la coupe de Dantzig ou la coupe de type I puis en utilisant la méthode dual de simplex et les coupes Gomory autant de fois nécessaire pour trouver une nouvelle solution optimale x^k , $\phi^k = dx^k$

– Si x^k est efficace et $\phi_{opt} < dx^k$ poser $x_{opt} = x^k$ et $\phi_{opt} = dx^k$ et aller à la sous-étape (3.1);

– Sinon, aller à l'étape (3.1) sans mettre rien à jour.

Étape terminale La procédure prend fin, ou bien à la première étape si la solution x^0 est efficace, ou lorsque l'opération de pivotage indique que la région courante ne contient aucun point entier réalisable. La solution optimale est alors x_{opt} et sa valeur sur le critère ϕ est ϕ_{opt} .

Pour plus d'éclaircissement le lecteur peut consulté des exemples illustratifs de la méthode dans chapitre 5 page 123 de la référence [16] et dans [1].

3.2.4.3 Méthode de Jorge,2009 [42]

L'algorithme proposé consiste à produire une solution optimale globale de (\mathbb{P}_E) sans devoir énumérer l'ensemble de toutes les solutions efficaces E_X , la procédure commence à résoudre le problème relaxé (\mathcal{P}_R) , sa solution est testée pour l'efficacité, évidemment, seulement dans un nombre limité de cas spéciaux la solution optimale de (\mathcal{P}_R) fournit une solution optimale de (\mathbb{P}_E) . Ainsi, si ce n'était pas le cas, une solution efficace qui domine la solution optimale de (\mathcal{P}_R) est alors générée par le programme linéaire de test d'efficacité définit dans le théorème(15). Par suite, dans chaque itération, le critère principal est optimisé sur le domaine restreint $\mathcal{D} - \mathcal{D}_s$ en incluant progressivement des contraintes 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 et efficace soit finalement trouvée.

3.2.4.4 L'algorithme de la méthode

Étape 0 (Initialisation)

- Poser $\phi_{inf} = -\infty$, $\phi_{sup} = +\infty$ et $l = 1$
- Résoudre le PL relaxé (\mathcal{P}_R)
 - Si (\mathcal{P}_R) est irréalisable, terminer le PL (\mathbb{P}_E) est aussi irréalisable;
 - Sinon, soit x^l une solution optimale de (\mathcal{P}_R)

Étape 1

- Si x^l est efficace. Terminer $x_{opt} = x^l$ et $\phi_{opt} = dx^l$
- Sinon, poser $\phi_{sup} = dx^l$ et aller à l'étape (2)

Étape 2

Soit \hat{x}^l une solution optimale du test d'efficacité dont le vecteur critère domine celui de x^l . Dans l'espace des critères, plusieurs solution peuvent avoir le même vecteur critère que \hat{x}^l , pour cela le problème $(\mathcal{T}_l) \equiv \max\{dx^l \mid f(x) = f(\hat{x}^l), x \in \mathcal{D}\}$ est résolu pour trouver

3.2. OPTIMISATION D'UNE FONCTION LINÉAIRE SUR L'ENSEMBLE EFFICACES DISCRET

les solutions équivalentes à \hat{x}^l .

Soit \bar{x}^l une solution optimale de (\mathcal{T}_l)

- Si $d\bar{x}^l > \phi_{inf}$, poser $\phi_{inf} = d\bar{x}^l$ et $x_{opt} = \bar{x}^l$
- $\phi_{inf} = \phi_{sup}$, terminer, x_{opt} est optimale pour (\mathbb{P}_E)

Étape 3

Résoudre le problème $(\mathcal{P}_l) \equiv \max\{dx \mid x \in \mathcal{D} - \bigcup_{s=1}^l \mathcal{D}_s\}$ où $\mathcal{D}_s = \{x \in \mathbb{Z}^n \mid f(x) \leq f(\bar{x}^s)\}$ avec $\bar{x}^1, \bar{x}^2, \dots, \bar{x}^l$ sont respectivement les solutions de optimales de $(\mathcal{T}_1), (\mathcal{T}_2), \dots, (\mathcal{T}_l)$.

- Si (\mathcal{P}_l) n'est pas réalisable, terminer x_{opt} est optimale pour (\mathbb{P}_E)
- Sinon, soit x^{l+1} une solution optimale de (\mathcal{P}_l)
 - Si $d\bar{x}^{l+1} \leq \phi_{inf}$. terminer x_{opt} est optimale pour (\mathbb{P}_E)
 - Sinon, poser $l = l + 1$ et aller à l'étape (1)

3.2.4.5 Illustration numérique

Considérant l'exemple à deux objectifs suivant :

$$(P(\mathcal{D})) \left\{ \begin{array}{l} \max f_1(x) = z_1 = x_1 - 2x_2 \\ \max f_2(x) = z_2 = -x_1 + 4x_2 \\ -2x_1 + 4x_2 \leq 0 \\ x_1 \leq 3 \\ x_2 \leq 2 \\ x_1, x_2 \in \mathbb{Z}_+^* \end{array} \right.$$

On considère le problème principale :

$$(P_E) \left\{ \begin{array}{l} \max \phi = -x_1 - 2x_2 \\ x \in X_E \end{array} \right.$$

L'ensemble des solutions efficaces est : $E = \{(2; 0), (3; 0), (2; 1), (3; 1), (1; 2), (2; 2), (3; 2)\}$

les solutions $(2; 1)$ et $(2; 0)$ sont non-supportées.

les bornes inférieures des deux fonctions objectifs sont : $-M_1 = -3$ et $-M_2 = -3$

Itération 1

Étape 0 : $\phi_{inf} = -\infty$, $\phi_{sup} = +\infty$, $l = 1$

Résoudre le PL relaxé $(P_R) \equiv \max\{-x_1 - 2x_2, \quad x \in \mathcal{D}\}$

$x^1 = (0; 0)$ sa solution optimale et $f(x^1) = (0; 0)$ son vecteur critère.

3.2. OPTIMISATION D'UNE FONCTION LINÉAIRE SUR L'ENSEMBLE EFFICACES DISCRET

Étape 1 : tester l'efficacité de x^1 en résolvant le problème

$$P(x^1) \left\{ \begin{array}{l} \max \quad \Theta = \psi_1 + \psi_2 \\ \text{s.c} \quad (x_1, x_2) \in \mathcal{D} \\ x_1 - 2x_2 - \psi_1 = 0 \\ -x_1 + 4x_2 - \psi_2 = 0 \end{array} \right.$$

$\Theta \neq 0$ donc $x^1 = (0; 0)$ n'est pas efficace, on pose $\phi_{sup} = (-1, -2)(0; 0) = 0$

Étape 2 :

- $\hat{x}^1 = (2; 1)$ est la solution efficace donnée par le test d'efficacité $P(x^1)$ et $f(\hat{x}^1) = (0; 2)$
- Soit à résoudre le problème (T_1)

$$(T_1) \left\{ \begin{array}{l} \max \quad \phi = -x_1 - 2x_2 \\ \text{s.c} \quad (x_1, x_2) \in \mathcal{D} \\ x_1 - 2x_2 = 0 \\ -x_1 + 4x_2 = 2 \end{array} \right.$$

- La solution optimale de (T_1) est $\bar{x}^1 = \hat{x}^1 = (2; 1)$
- $\phi(\bar{x}^1) = -4 > \phi_{inf} = -\infty \Rightarrow \phi_{inf} = -4$ et $x_{opt} = \bar{x}^1 = (2; 1)$
- $\phi_{inf} \neq \phi_{sup}$, aller à l'étape (3).

Étape 3 : Résoudre le problème (P_1) défini par :

$$(P_1) \left\{ \begin{array}{l} \max \quad \phi = -x_1 - 2x_2 \\ x_1, x_2 \in \mathcal{D} \\ f_i(x) \geq (f(x_{opt}) + 1)y_i^1 - M_i(1 - y_i^1), i = 1, 2 \\ \sum_{i=1}^2 y_i^1 \geq 1 \\ y_i^1 \in \{0, 1\}, i = 1, 2 \end{array} \right.$$

$$(P_1) \left\{ \begin{array}{l} \max \quad \phi = -x_1 - 2x_2 \\ x_1, x_2 \in \mathcal{D} \\ x_1 - 2x_2 \geq y_1^1 - 3(1 - y_1^1) \\ -x_1 + 4x_2 \geq 3y_2^1 - 3(1 - y_2^1) \\ y_1^1 + y_2^1 \geq 1 \\ y_1^1, y_2^1 \in \{0, 1\} \end{array} \right.$$

3.2. OPTIMISATION D'UNE FONCTION LINÉAIRE SUR L'ENSEMBLE EFFICACES DISCRET

- $x^2 = (1; 0)$ est la solution optimale de (P_1) , $f(x^2) = (1; -1)$
- comme $\phi(x^2) = -1 > \phi_{inf}$, aller à l'étape (1)

Itération 2 :

Étape 1 :

- La solution $x^2 = (1; 0)$ n'est pas efficace car la valeur de la fonction objectif $\Theta^* \neq 0$ de programme de test d'efficacité $P(x^2)$.
- $\phi_{sup} = -1$

Étape 2 :

- La solution efficace donné par le programme de test d'efficacité est $\hat{x}^2 = (3; 1)$ et $f(\hat{x}^2) = (1; 1)$
- On résout le problème (T_2)

$$(T_2) \equiv \max\{\phi = -x_1 - 2x_2 \mid x_1 - 2x_2 = 1; -x_1 + 4x_2 = 1; \quad x_1, x_2 \in \mathcal{D}\}$$

- la solution optimale de (T_2) est $\bar{x}^2 = \hat{x}^2 = (3; 1)$
- $\phi(\bar{x}^2) = -5 < \phi_{inf} = -4$

Étape 3 :

- Résoudre le problème (P_2)

$$(P_2) \left\{ \begin{array}{l} \max \quad \phi = -x_1 - 2x_2 \\ \quad \quad \quad x_1, x_2 \in \mathcal{D} \\ \quad \quad \quad x_1 - 2x_2 \geq y_1^1 - 3(1 - y_1^1) \\ \quad \quad \quad -x_1 + 4x_2 \geq 3y_2^1 - 3(1 - y_2^1) \\ \quad \quad \quad x_1 - 2x_2 \geq 2y_1^2 - 3(1 - y_1^2) \\ \quad \quad \quad -x_1 + 4x_2 \geq 2y_2^2 - 3(1 - y_2^2) \\ \quad \quad \quad y_1^1 + y_2^1 \geq 1 \\ \quad \quad \quad y_1^2 + y_2^2 \geq 1 \\ \quad \quad \quad y_1^1, y_2^1, y_1^2, y_2^2 \in \{0, 1\} \end{array} \right.$$

- La solution optimale de (P_2) est $x^3 = (2; 0)$ et $y = (1; 0; 1; 0)$, $f(x^3) = (2; -2)$
- $\phi(x^3) = -2 > \phi_{inf} = -4$, $l = 3$

Itération 3 :

Étape 1 :

La solution $x^3 = (2; 0)$ est efficace puisque la valeur optimale de la fonction objectif Θ de programme de test d'efficacité $(P(x^3))$ est nulle.

L'algorithme prend fin, $x_{opt} = (2; 0)$ est optimale pour le problème principal considéré et la valeur optimale du critère est $\phi_{opt} = -2$

3.2.4.6 Méthode de Chaabane et Pirlot, 2010 [17]

La méthode proposée par les auteurs est une combinaison entre les principes utilisés dans les deux algorithmes présentés précédemment, le principe de la méthode est le suivant :

Dans une première étape, on cherche une solution efficace initiale, on part de la solution optimale du problème relaxé (P_R), si elle existe, cette solution est testée pour l'efficacité, pour voir si par hasard, on peut résoudre le problème (P_E) juste en résolvant son problème relaxé, autrement, le problème de test d'efficacité fournit une solution efficace que l'on note \hat{x} . Cette dernière est utilisée pour optimiser le critère principal sur les solutions admissibles ayant un même vecteur critères en résolvant le problème (T_l)(défini à l'étape 2 de l'algorithme de la section suivante), soit \bar{x} la solution optimale de ce problème, est considérée comme une première solution efficace, on initialise alors $X_{opt} = \bar{x}$ et $\phi_{opt} = d\bar{x}$. Ensuite, dans chaque itération, le domaine de recherche est réduit progressivement par des contraintes éliminant toutes les solutions dominées par la solution \bar{x} en résolvant le problème (P) la solution optimale trouvée, soit x^l , produit une valeur maximale du critère ϕ dans le domaine réduit, si elle est efficace, la procédure se termine avec x^l une solution optimale du problème, sinon, en utilisant le tableau optimal correspondant, une exploration des arêtes incidentes à cette solution est réalisée pour chercher une solution entière alternative efficace. Si une telle solution existe, elle est la solution optimale de (P_E), sinon, l'algorithme va poursuivre par une alternance des étapes de réduction et d'exploration jusqu'à ce qu'une solution améliore la valeur de ϕ soit trouvée où le domaine de recherche devient vide.

Dans la suite, nous donnons une description formelle de l'algorithme.

3.2.4.7 L'algorithme de la méthode

Étape 0 : Résoudre le problème relaxé (P_R).

- Si (P_R) est irréalisable, alors la procédure prend fin, le problème principal (P_E) n'a pas de solution.
- Sinon, soit x^0 une solution optimale de (P_R).

Étape 1 : Poser $l = 0$, $\phi_{opt} = -\infty$, $H^0 = D$.

- Si x^0 est efficace ; terminer, x^0 est une solution optimale de (P_E).
- Sinon, soit \hat{x}^0 solution efficace dont le vecteur critères correspond domine celui de x^0 (solution optimale du test d'efficacité de x^0).

Étape 2 : Résoudre le problème (T_l) défini par

$$(T_l) \begin{cases} \max & \phi(x) \\ \text{t.q} & x \in D. \\ & Cx = C\hat{x}^l \end{cases}$$

Soit \bar{x}^l une solution optimale de (T_l) .

- Si $\phi(\bar{x}^l) > \phi_{opt}$; poser $X_{opt} = \bar{x}^l, \phi_{opt} = d\bar{x}^l$ et aller à l'étape 3.

Étape 3 : Poser $l = l + 1$ et résoudre le problème (P_l) défini par

$$(P_l) \begin{cases} \max & \phi(x) \\ \text{t.q} & x \in H^l. \end{cases}$$

$$H^l = H^{l-1} \cap \left\{ \begin{array}{l} x \in D | Z_i(x) \geq (Z_i(X_{opt}) + 1)y_i^l - M_i(1 - y_i^l) \\ \text{pour } i = 1, 2, \dots, p. \\ \sum_{i=1}^p y_i^l \geq 1, y_i^l \in \{0, 1\}. \end{array} \right\}$$

- Si $H^l = \emptyset$; terminer, X_{opt} est la solution optimale de (P_E) .

- Sinon; soit x^l solution optimale de (P_l) .

- Si $\phi(x^l) \leq \phi_{opt}$; terminer, X_{opt} est la solution optimale de (P_E) .

- Sinon,

▷ Si x^l est efficace; terminer $X_{opt} = x^l, \phi_{opt} = dx^l$.

▷ Sinon, soit \hat{x}^l solution efficace domine x^l (solution optimale du teste d'efficacité de x^l); aller à l'étape 4.

Étape 4 : Construire $J_l = \{j \in N_l | \phi_j - d_j = 0\}$

Où N_l est l'ensemble des indice hors base dans le tableau optimal correspondant à x^l .

(4.1) Si $J_l = \emptyset$, aller à l'étape 2.

(4.2) Sinon, chercher une solution entière efficace x^l alternative à x^l en explorant les arêtes incidentes. choisir $j_l \in J_l$ et calculer $\theta_{j_l}^0 = \lfloor \min_{i \in I_l} \{ \frac{x_{l,i}}{y_{l,i j_l}}, y_{l,i j_l} > 0 \} \rfloor$.

- Si $\theta_{j_l}^0 = 0$, aucune solution entière ne peut être existe sur l'arête E_{j_l} , poser $J_l = J_l \setminus \{j_l\}$ et aller à (4-1).

- Sinon, chercher une solution entière efficace sur l'arête E_{j_l} correspondant à $\theta_{j_l}^0$ commençant de $\theta = 1$ jusqu'à $\theta = \theta_{j_l}^0$ (θ est un entier positif) et tester pour l'efficacité.

- Si $\exists x^l$ solution entière efficace alternative à x^l ; terminer,

$X_{opt} = x^l, \phi_{opt} = dx^l$.

- Sinon, poser $J_l = J_l \setminus \{j_l\}$ aller à l'étape (4.1).

3.2. OPTIMISATION D'UNE FONCTION LINÉAIRE SUR L'ENSEMBLE EFFICACES DISCRET

Propriété 3 Soit x^l une solution alternative à x^l la solution optimale du problème (P_l) avec $\phi(x^l) > \max_{s \in \{1, \dots, l\}} \{\phi(\bar{x}^s)\}$.

Si $x^l \in E$ alors x^l est une solution optimale de (P_E) .

Propriété 4 L'algorithme converge en un nombre fini d'itérations avec une solution optimale du problème (P_E) .

Preuve :

Par l'hypothèse que la région d'admissibilité S est bornée (il y a un nombre limité de solutions entières) et D est supposé non vide, on aura au moins une solution entière; dans chaque itération de l'algorithme, où bien une solution efficace améliore la valeur de ϕ est trouvée où bien le domaine de recherche est réduit progressivement jusqu'à ce qu'il devient vide.

3.2.4.8 Illustration numérique :

Nous illustrons le fonctionnement de l'algorithme sur un exemple didactique.

Exemple 4 Considérons l'exemple proposé par Philip, 1972 [51]

$$(P(D)) \left\{ \begin{array}{l} \max \quad Z_1 = 2x_1 - x_2 \\ \max \quad Z_2 = -x_1 + 2x_2 \\ t.q. \quad x_1 + 2x_2 \leq 8, \\ \quad \quad x_1 \leq 5, \\ \quad \quad x_2 \leq 7, \\ \quad \quad x_1 + x_2 \leq 10, \\ \quad \quad x_1, x_2 \in \mathbb{N}^+ \end{array} \right.$$

Le problème principal est donné par

$$(P_E) \left\{ \begin{array}{l} \max \quad \phi = -x_1 - 3x_2 \\ t.q. \quad x_1, x_2 \in E. \end{array} \right.$$

Étape 0 : Résoudre le problème relaxé (P_R)

$$(P_R) \left\{ \begin{array}{l} \max \quad \phi = -x_1 - 3x_2 \\ t.q. \quad x_1 + 2x_2 \leq 8, \\ \quad \quad x_1 \leq 5, \\ \quad \quad x_2 \leq 7, \\ \quad \quad x_1 + x_2 \leq 10, \\ \quad \quad x_1, x_2 \in \mathbb{N}^+ \end{array} \right.$$

3.2. OPTIMISATION D'UNE FONCTION LINÉAIRE SUR L'ENSEMBLE EFFICACES DISCRET

La solution optimale de (P_R) est $x^0 = (0, 0)$, les bornes inférieures des deux fonctions objectifs sont $-M_1 = -7, -M_2 = -5$ respectivement. **Étape 1** : $l = 0, \phi_{opt} = -\infty$,

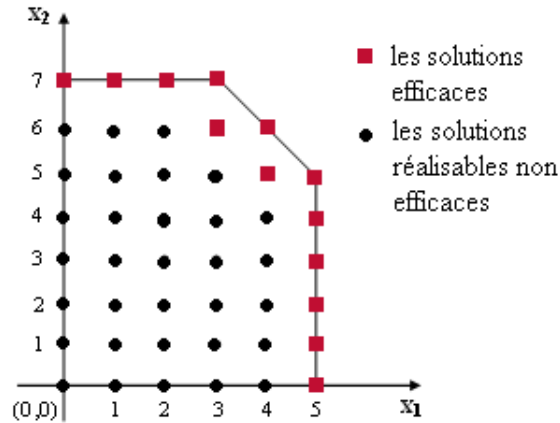


FIGURE 3.1 – Le domaine d'admissibilité D

$H^0 = D$, la solution x^0 n'est pas efficace, $\hat{x}^0 = (5, 5)$ est la solution obtenue en résolvant le problème de test d'efficacité $P(x^0)$ dont le vecteur critères correspondant est $Z(\hat{x}^0) = (5, 5)$.

Étape 2 : Résoudre le problème (T_0)

$$(T_0) \begin{cases} \max & \phi = -x_1 - 3x_2 \\ t.q. & (x_1, x_2) \in D. \\ & 2x_1 - x_2 = 5 \\ & -x_1 + 2x_2 = 5 \end{cases}$$

La solution optimale est $\bar{x}^0 = (5, 5)$, $\phi(\bar{x}^0) = -20 > \phi_{opt}$, faire alors $X_{opt} = (5, 5)$, $\phi_{opt} = -20$.

Étape 3 : $l = l + 1 = 1$, résoudre le problème (P_1)

$$(P_1) \begin{cases} \max & \phi = -x_1 - 3x_2 \\ t.q. & (x_1, x_2) \in D \\ & 2x_1 - x_2 \geq 6y_1^1 - 7(1 - y_1^1) & (1) \\ & -x_1 + 2x_2 \geq 6y_2^1 - 5(1 - y_2^1) & (2) \\ & y_1^1 + y_2^1 \geq 1 \\ & (x_1, x_2) \in \mathbb{N}^+, (y_1^1, y_2^1) \in \{0, 1\}. \end{cases}$$

La solution optimale de ce problème est $x^1 = (3, 0)$, $\phi(x^1) = -3$, $Z(x^1) = (6, -3)$ et $y = (1, 0)$.

Comme $\phi(x^1) > \phi_{opt}$ tester l'efficacité cette solution en résolvant le problème

3.2. OPTIMISATION D'UNE FONCTION LINÉAIRE SUR L'ENSEMBLE EFFICACES DISCRET

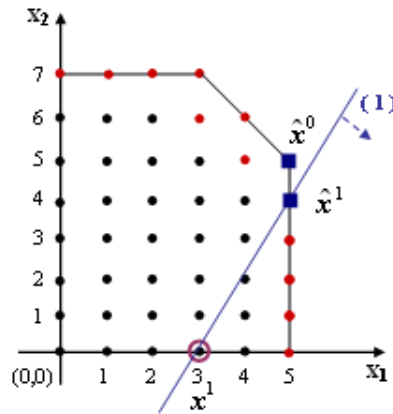


FIGURE 3.2 – La région réduite D^1

$$(P(x^1)) \left\{ \begin{array}{l} \max \quad \Theta = \psi_1 + \psi_2 \\ t.q. \quad x_1 \leq 5, \\ \quad \quad x_2 \leq 7, \\ \quad \quad x_1 + x_2 \leq 10, \\ \quad \quad 2x_1 - x_2 - \psi_1 = 6, \\ \quad \quad -x_1 + 2x_2 - \psi_2 = -3, \\ \quad \quad (x_1, x_2) \in \mathbb{N}^+, \psi_i \in \mathbb{R}^+, i = 1, 2. \end{array} \right.$$

On obtient $\Theta^* \neq 0$ et $\hat{x}^1 = (5, 4)$ solution optimale de $(P(x^1))$ dont le vecteur critères correspondant $Z(\hat{x}^1) = (6, 3)$.

Étape 4 : l'ensemble $J_1 = \{j \in N_1 | c_j - d_j = 0\} = \emptyset$ (en utilisant le tableau optimale de x^1), aller alors l'étape 2.

Itération 2

Étape 2 : Résoudre le problème (T_1)

$$(T_1) \left\{ \begin{array}{l} \max \quad \phi = -x_1 - 3x_2 \\ t.q. \quad (x_1, x_2) \in D. \\ \quad \quad 2x_1 - x_2 = 6 \\ \quad \quad -x_1 + 2x_2 = 3 \end{array} \right.$$

$\bar{x}^1 = (5, 4)$, $\phi(\bar{x}^1) = -17 > \phi_{opt}$, faire $X_{opt} = (5, 4)$, $\phi_{opt} = -17$.

Étape 3 : $l = l + 1 = 2$, résoudre le problème (P_2)

3.2. OPTIMISATION D'UNE FONCTION LINÉAIRE SUR L'ENSEMBLE EFFICACES DISCRET

$$(P_2) \left\{ \begin{array}{l} \max \quad \phi = -x_1 - 3x_2 \\ t.q. \quad (x_1, x_2) \in D^1 \\ \quad \quad 2x_1 - x_2 \geq 7y_1^2 - 7(1 - y_1^2) \quad (3) \\ \quad \quad -x_1 + 2x_2 \geq 4y_2^2 - 5(1 - y_2^2) \quad (4) \\ \quad \quad y_1^2 + y_2^2 \geq 1 \\ \quad \quad (x_1, x_2) \in \mathbb{N}^+, (y_1^2, y_2^2) \in \{0, 1\}. \end{array} \right.$$

La solution optimale est $x^2 = (4, 0)$ qui n'est pas efficace avec $\phi(x^2) = -4$, $Z(x^2) = (8, -4)$ et $y = (1, 0, 1, 0)$. Soit $\hat{x}^2 = (5, 2)$, $Z(\hat{x}^2) = (8, -1)$ solution optimale du test d'efficacité de x^2 .

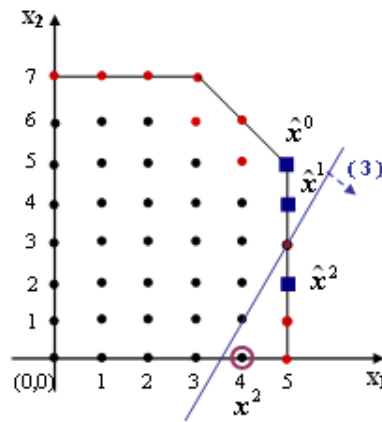


FIGURE 3.3 – La région réduite D^2

Étape 4 : l'ensemble $J_2 = \{j \in N_2 | c_j - d_j = 0\} = \emptyset$, aller alors l'étape 2.

Itération 3

Étape 2 : Résoudre le problème (T_2)

$$(T_2) \left\{ \begin{array}{l} \max \quad \phi = -x_1 - 3x_2 \\ t.q. \quad (x_1, x_2) \in D. \\ \quad \quad 2x_1 - x_2 = 8 \\ \quad \quad -x_1 + 2x_2 = -1 \end{array} \right.$$

$\bar{x}^2 = (5, 2)$, $\phi(\bar{x}^2) = -11 > \phi_{opt}$, faire alors $X_{opt} = (5, 2)$, $\phi_{opt} = -11$.

Étape 3 $l = l + 1 = 3$, résoudre le problème (P_3)

$$(P_3) \left\{ \begin{array}{l} \max \quad \phi = -x_1 - 3x_2 \\ t.q. \quad (x_1, x_2) \in D^2 \\ \quad \quad 2x_1 - x_2 \geq 9y_1^3 - 7(1 - y_1^3) \quad (5) \\ \quad \quad -x_1 + 2x_2 \geq -5(1 - y_2^3) \quad (6) \\ \quad \quad y_1^3 + y_2^3 \geq 1 \\ \quad \quad (x_1, x_2) \in \mathbb{N}^+, (y_1^3, y_2^3) \in \{0, 1\}. \end{array} \right.$$

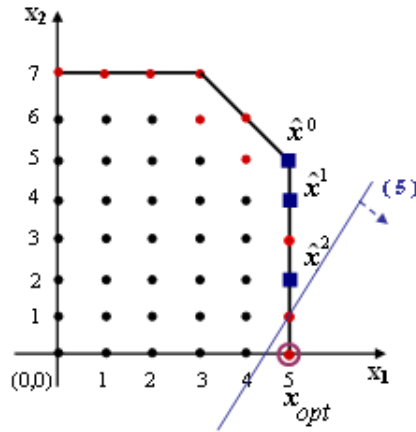


FIGURE 3.4 – La région réduite D^3

$x^3 = (5, 0)$, $\phi(x^3) = -5 > \phi_{opt}$, après le test d'efficacité de x^3 on trouve qu'elle est efficace, la procédure prend fin avec $X_{opt} = (5, 0)$ et $\phi_{opt} = -5$.

L'ensemble de toutes les solutions efficaces de ce problème est

$$E_X = \{(0, 7), (1, 7), (2, 7), (3, 7), (3, 6), (4, 6), (4, 5), (5, 5), (5, 4), (5, 3), (5, 2), (5, 1), (5, 0)\}.$$

Tandis que l'algorithme proposé par les auteurs optimise le critère principal ϕ sans devoir passer par toutes ces solutions mais seulement par $E_{X_1} = \{(5, 5), (5, 4), (5, 2), (5, 0)\}$.

3.3 Conclusion

Dans ce chapitre, nous avons présenté un état de l'art de quelques méthodes de résolution du problème d'optimisation d'une fonction linéaire sur l'ensemble de solutions efficaces des problèmes *MOLP* et *MOILP*. Les problèmes sont très difficiles vu la non convexité du domaine où on optimise la fonction linéaire ϕ . Malheureusement nous n'avons pas assez d'expériences informatiques pour les évalués et de faire une étude comparative.

4

La norme augmentée de Tchebychev pour optimiser une fonction Linéaire sur l'Ensemble des Solutions Efficaces Discrètes

4.1 Introduction

Dans ce chapitre, nous proposons une méthode exact pour la résolution du problème d'optimisation d'une fonction linéaire sur l'ensemble de solutions efficaces discrètes d'un problème linéaire multi-objectifs, qui présente la contribution principale de notre travail [18]. Dans cette méthode, deux phases importantes sont à considérer : la phase d'ajout des contraintes de coupes pour la réduction progressif du domaine réalisable, a fin de trouver une solution optimale relatif a ce domaine réduit et la phase de génération d'une solution efficace/non dominée qui domine la solution optimale trouvée déjà en première phase. La solution efficace/non dominée peut être générée par la résolution du programme de la norme augmentée de Tchebychev. Avant de présenter notre algorithme tout d'abord nous aborderons dans la section suivante quelques concepts et méthodes liés à la norme de Tchebychev pour la résolution de problèmes multi-objectifs.

4.2 Quelques méthodes de résolution multicritère basée sur la distance avec un objectif de référence

Rappelons qu'un problème linéaire multi-objectifs en nombre entier peut être formulé comme suit

$$(P) \quad \max\{Cx, \quad x \in D\} \quad (4.1)$$

où $D = S \cap \mathbb{Z}$ avec $S = \{x \in \mathbb{R}^n \mid Ax \leq b; x \geq 0\}$ est un polyèdre borné et non vide; $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{Z}^{m \times 1}$, $C = (c^i)_{i \in \{1, \dots, p\}} \in \mathbb{Z}^{p \times n}$ et $p \geq 2$.

Comme nous l'avons mentionné, il existe plusieurs méthodes qui permettent de trouver la frontière de Pareto. Nous avons choisi la méthode basée sur la distance avec un objectif de référence, ce dernier correspond dans notre travail au point idéal ou bien point utopique, l'avantage principal de cette méthode c'est qu'elle génère l'ensemble des solutions efficaces même si la frontière efficace (ou Paréto) n'est pas convexe (comme le montre la figure 4.1). La distance la plus communément utilisée par cette méthode est la distance de

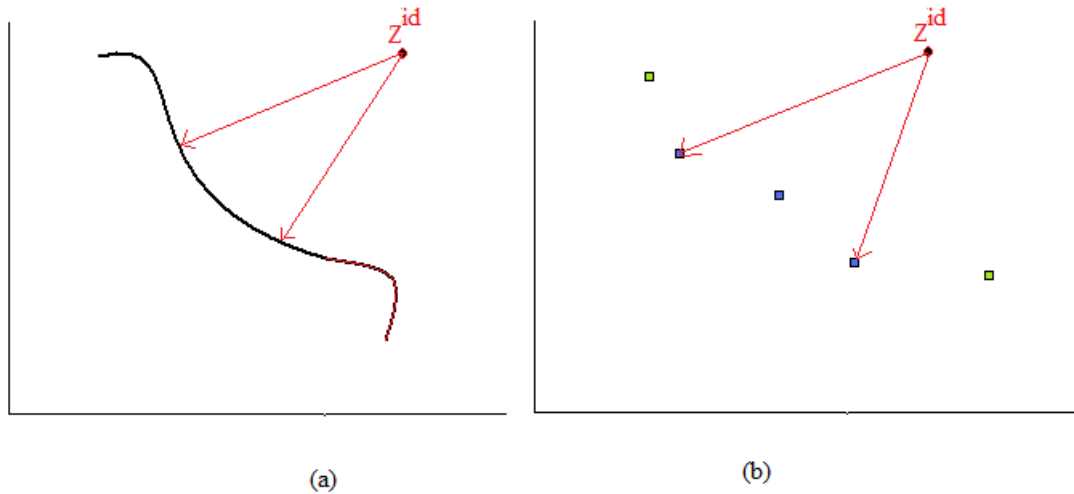


FIGURE 4.1 – Génération des solutions efficaces par une méthode basée sur les distances : (a) cas continu, (b) cas discret

Tchebychev. Cette distance minimise, sur tous les objectifs, la différence entre la valeur de l'objectif et la valeur du point de référence. Cette fonction de distance peut être pondérée et les poids relatifs à chacun des objectifs peuvent être variés pour énumérer l'ensemble des solutions efficaces. Puisque notre objectif est d'éviter d'énumérer tout l'ensemble des solutions efficaces, alors l'utilisation de cette méthode est restreinte seulement pour la

caractérisation d'une solution efficace. Cette méthode peut également utiliser la distance euclidienne.

4.3 Norme de Tchebychev et résultats préliminaires

La théorie de Tchebychev est originalement fondée par Bowman, 1976 [15] et ces propriétés sont bien exploitées par plusieurs auteurs, telle que l'utilisation de la fonction scalarisante, de normes de Tchebychev, par les méthodes de la programmation multi-objectifs avec point de référence (voir par exemple dans Wierzbicki, 1980 [74]) et aussi par une classe d'algorithmes interactives pour l'optimisation multi-objectifs (voir par exemple dans Steuer et Choo, 1983 [62] et la version la plus récente à notre connaissance de Steuer [46]), une autre catégorie particulière dédiée pour les problèmes bi-objectifs en nombre entier et non linéaire, on trouve les algorithmes de Eswaran et al., 1989 [29]; Ralphs et al., 2006 [54], et autres méthodes comme celles de Alves et Climaco, 2000 [3]; Karaivanova et al., 1977 [44]; Neumayer et Schweigert, 1994 [50]; et celle de Schandl et al., [59].

Bowman, 1976 [15] prouve que la scalarisation de normes de Tchebychev est la plus appropriée pour la génération de l'ensemble des solutions non dominées d'un problème multi-objectifs, particulièrement celles qui sont non supportées.

L'ensemble de solutions non dominées dans l'espace de critères produit certaines informations pour le *MOILP*, si les fonctions objectifs sont bornées sur la région réalisable. Les bornes supérieures de l'ensemble des solutions non dominées sont données par le point idéal $Z^{id} \in \mathbb{R}^p$.

Rappelons que le vecteur strictement supérieur au vecteur idéal est appelé vecteur utopique noté par Z^U .

Soit l'espace des vecteurs de préférences noté δ (appelé aussi vecteur poids) telque :

$$\Delta = \left\{ \beta \in \mathbb{R}^p \mid 0 < \beta_i < 1, \sum_{i=1}^p \beta_i = 1 \right\}.$$

La norme pondérée de Tchebychev relativement a un point de référence, le point idéal est donnée par :

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

La norme pondérée de Tchebychev dans \mathbb{R}^p relativement à un point de référence, le point utopique, est définie comme suit

$$\|z^U - z\|_{\beta} = \max_{i=1, \dots, p} \{ \beta_i |z_i^U - z_i| \}. \quad (4.2)$$

La norme augmentée de Tchebychev est définie comme suit :

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

La norme augmentée de Tchebychev relativement a un point de référence, le point utopique, est définie comme suit :

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

Avec $\rho > 0$ et suffisamment petit.

Ces deux normes caractérisent la distance entre n'importe quel point $z \in \mathcal{Z}$ Soit à résoudre le problème suivant :

$$(MOILP) \equiv \max \{ c^i x, \quad x \in \mathcal{D}, \quad i = 1, 2, \dots, p \};$$

qui équivalent à

$$(MOILP) \equiv \max \{ z^i, \quad z \in Y, \quad i = 1, 2, \dots, p \}.$$

Étant donné un point non dominé \bar{z} , pour déterminer un point non dominé dans le voisinage de \bar{z} et plus proche du point idéal, par la résolution des programmes linéaire mixtes appelés programmes de la norme pondérée de Tchebychev (WNP) suivant :

(a) Programme de la norme pondérée de Tchebychev :

$$P(\beta) \left\{ \begin{array}{l} \text{Min } \omega \\ \omega \geq \beta_i (y_i^{id} - y_i), 1 \leq i \leq p, \beta \in \Delta; \\ y \in \mathcal{Z}; \\ \omega \geq 0. \end{array} \right.$$

(b) Programme de la norme augmentée de Tchebychev :

$$P_\rho(\beta) \left\{ \begin{array}{l} \min \quad \omega + \rho \sum_{i=1}^p [y_i^{id} - y_i] \\ \omega \geq \beta_i (y_i^{id} - y_i), 1 \leq i \leq p; \\ y_i = c^i x; \\ x \in D; \\ \omega \geq 0. \end{array} \right. \quad (4.3)$$

Pour un $\rho > 0$ et suffisamment petit.

tels que

$$\beta_i = \begin{cases} \frac{1}{y_i^{id} - y_i} \left[\sum_{i=1}^p \frac{1}{y_i^{id} - y_i} \right]^{-1}, & \text{si } y_i^{id} \neq y_i \forall i; \\ 1, & \text{si } y_i^{id} = y_i; \\ 0, & \text{si } y_i^{id} \neq y_i, \exists j, y_j^{id} = y_j; \end{cases} \quad (4.4)$$

Les résultats suivants fournissent quelques conditions sur la caractérisations de la solution non dominée.

Théorème 17 [61] *Soit Y est fini et*

$$M = \{y \in Y \mid (x, y, \omega) \text{ est la solution minimale du } P(\beta) \text{ pour un certain } \beta \in \Delta, \}$$

alors, il existe un $\hat{y} \in M$ tel que $\hat{y} \in Y_N$.

Théorème 18 [15] *Soit $y = C\hat{x}, \hat{x} \in D$ est une solution non dominée du problème MOILP si elle est optimale pour $P(\beta)$, pour un certain β .*

Théorème 19 [15] *Si Y est uniformément non dominant alors chaque solution du $P(\beta)$ est non dominée.*

Dans plusieurs problèmes pratiques les solutions faiblement non dominées sont indésirables. Pour surmonter une telle situation nous utiliserons dans notre algorithme la norme augmentée de Tchebychev définie en 4.3.

Théoriquement, le terme $\rho \sum_{i=1}^p (y^U - \bar{y}_i)$ correspond à la destination souhaitable ou acceptable pour avoir un compromis. Ce concept reflète le taux de changement dans les valeurs des fonctions objectifs concernant l'augmentation d'une fonction objectif qui se produit lorsque la valeur d'une autre fonction objectif diminue. En ce sens, le déplacement d'un vecteur non dominé à un autre nécessite ce changement, ce mouvement indique une "légère inclinaison" de quelque valeur des fonctions objectifs. Les valeurs exactes de ρ qui fonctionne correctement dépend de la taille relative des valeurs optimales de la fonction objectif et ne peut pas être calculée a priori. Le principal objectif d'introduire le programme Tchebychev pondérée augmentée est d'éviter d'obtenir une solution faiblement non dominée. Une représentation visuelle de $(P(\beta))$ pour une valeur particulière de β est montré dans (fig.4.2). Le rectangle $(ABCD)$ est appelé isoquant et y est dit sommet de l'isoquant. $(ABCD)$ représente les lignes de niveau optimal de la norme de Tchebychev. y^1 et y^2 se trouvent sur les lignes de niveau optimal de la norme de Tchebychev. Les solutions y^1, y^2 et y^3 sont optimales pour $P(\beta)$ (pour certaines valeurs de β), mais y^1 est une solution faiblement non dominée (non supportée). y^2 et y^3 sont non dominées (supportées).

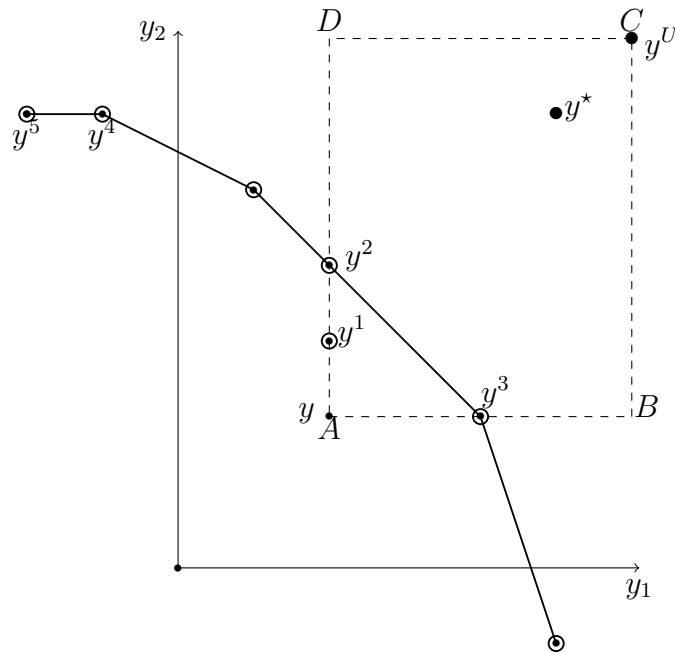


FIGURE 4.2 – Dominance faible et programme de la norme de Tchebychev

Remarque 7 Dans l’algorithme proposé nous utiliserons la norme augmentée de Tchebychev avec point un de référence, qui est le point utopique au lieu de point idéal et cela pour éviter la division par zéro dans la formule (4.4).

Dans la section suivante nous allons présentés deux méthodes basées sur la norme de Tchebychev, la première dédiée à la résolution du problème multi-objectifs en nombre entier qui est La méthode de Steuer et Choo, 1983 [62] et la deuxième est la méthode de Ralphs et al.,2007 [55] pour la résolution des problèmes bi-critères.

4.3.1 La méthode de Steuer et Choo, 1983[62]

La procédure proposée consiste à échantillonner un sous ensemble de solutions efficaces en déterminant les points non dominés associés les plus proches du point idéal z^{id} selon un échantillon de poids de la norme de Tchebychev générés aléatoirement. A chaque itération, une technique de filtrage est utilisée pour avoir des valeurs de poids dispersées de petit et de plus petit sous-ensembles de points non dominés sont déterminés suffisamment bons jusqu’à l’arrête du processus de décisions. Le décideur intervient pour donner sa solution préférée, puis son voisinage est exploré pour déterminer un sous-ensemble de points non dominés qui sont adjacents et plus proches du point idéal.

Les auteurs ont proposé de résoudre les programmes linéaire mixtes appelés programmes de la norme pondérée de Tchebychev décrite dans la section précédente l’algorithme uti-

lise trois paramètres (t,s,r) dont leurs valeurs peuvent influencer sa convergence tel que t =nombre d'itération maximale et s = la taille de l'échantillon à développer dans chaque itération et r est le facteur de convergence dont le but est de réduire séquentiellement l'espace de vecteur de poids Δ . Les auteurs ont discuté vivement leurs importance et proposent l'algorithme suivant :

L'algorithme de la méthode

- ⟨1⟩ Trouver le point idéal z^{id} .
- ⟨2⟩ Soit $\bar{\Delta}^0 = \left\{ \beta \in \mathbb{R}^p, 0 \leq \beta_i \leq 1, \sum_{i=1}^p \beta_i = 1 \right\}$.
- ⟨3⟩ Poser $k = 0$ et donner les valeurs des paramètres t, s, r .
- ⟨4⟩ $k = k + 1$.
- ⟨5⟩ Générer aléatoirement $100 \times s$ vecteurs de poids dans $\bar{\Delta}^k$.
- ⟨6⟩ Filtrer ces vecteurs pour avoir $2 \times s$ vecteurs dispersés (représentatif).
- ⟨7⟩ Pour chacun de ces vecteurs résoudre le programme de la norme augmentée de Tchebychev $P_a(\bar{\beta})$.
- ⟨8⟩ Filtrer les vecteurs obtenus pour avoir un échantillon de s vecteurs.
- ⟨9⟩ Soit z^k le vecteur jugé préféré par le décideur dans cet échantillon.
- ⟨10⟩ Calculer le vecteur β^k correspondant à z^k qui définit le point isoquant de la norme augmenté de Tchebychev.
- ⟨11⟩ Si $k < t$ aller à l'étape ⟨12⟩, si $k = t$ aller à l'étape ⟨14⟩.
- ⟨12⟩ Pour le vecteur β^k calculer dans l'étape ⟨10⟩ calculer

$$\bar{\Delta}^{(k+1)} = \left\{ \beta \in \mathbb{R}^p, \beta_i \in [\ell_i, \mu_i], \sum_{i=1}^p \beta_i = 1 \right\};$$

où

$$[\ell_i, \mu_i] = \begin{cases} [0, r^k], & \text{si } \beta_i^{(k)} - \frac{1}{2}r^k \leq 0; \\ [1 - r^k, 1], & \text{si } \beta_i^{(k)} + \frac{1}{2}r^k \geq 1; \\ [\beta_i^{(k)} - \frac{1}{2}r^k, \beta_i^{(k)} + \frac{1}{2}r^k], & \text{sinon.} \end{cases}$$

- ⟨13⟩ Aller à l'étape ⟨4⟩
- ⟨14⟩ Calculer l'image inverse du vecteur(s) critère choisi (s) par le décideur.
Arrêter.

4.3.2 L'algorithme de Ralphs et al., 2007 [55]

Ralphs et al., 2007 [55] ont proposé un algorithme paramétrique pour énumérer l'ensemble des solutions efficaces du problème linéaire bi-critère en nombres entiers basé sur la scolarisation de Tchebychev (Norme de Tchebychev). L'algorithme se focalise sur l'idée de Eswaran et al., 1989 [29] qui ont développé un algorithme pour le même problème mais pour le cas non linéaire. Le principe général de l'algorithme est simple. Étant donné deux solutions non dominé z^p, z^q , la procédure génère une autre solution (si elle existe) non dominé dans l'intervalle $[z^p, z^q]$. Cet intervalle est retiré de la liste des intervalles à explorer.

Un problème linéaire bi-critère en nombre entier est formulé comme suit :

$$(BIP) \equiv \max\{[f_1(x), f_2(x)], \quad x \in \mathcal{D} \subset \mathbb{Z}^n\}; \quad (4.5)$$

qui équivalent à

$$(BIP) \equiv \max\{[z_1, z_2], \quad z \in \mathcal{Z}\}; \quad (4.6)$$

où $z_i = f_i, i=1,2$. Et \mathcal{Z} représente l'image de \mathcal{D} selon f . et \mathcal{Z}_E dénote l'image de E . Et $|\mathcal{Z}_E| = |E| = N$.

La norme pondérée de Tchebychev devient alors :

$$\|z^{id} - z\|_\beta = \max\{\beta|z_1^{id} - z_1|, (1 - \beta)|z_2^{id} - z_2|\}.$$

Cette norme est utilisée pour calculer la distance minimale entre n'importe que point $z \in \mathcal{Z}$ et le point idéal z^{id} , d'après Bowman, 1976 [15] cela revient à résoudre le problème :

$$\min_{z \in \mathcal{Z}} \{\|z^{id} - z\|_\beta\}. \quad (4.7)$$

D'après Bowman, 1976 [15], le problème (4.7) est equivalent au problème suivant :

$$P(\beta) \equiv \min_{z \in \mathcal{Z}} \{\omega \mid \omega \geq \beta(z_1^{id} - z_1); \omega \geq (1 - \beta)(z_2^{id} - z_2); \quad 0 \leq \beta \leq 1\}. \quad (4.8)$$

(a) Pour tout point non dominé z^p on a :

$$\beta_p = \frac{z_2^{id} - z_2^p}{z_1^{id} - z_1^p + z_2^{id} - z_2^p}. \quad (4.9)$$

(b) Pour chaque paire (z^p, z^q) de point non dominé on a :

$$\beta_{p,q} = \frac{z_2^{id} - z_2^q}{z_1^{id} - z_1^p + z_2^{id} - z_2^q}. \quad (4.10)$$

Théorème 20 [15] *Si les solutions non dominées sont ordonnées dans cet ordre*

$$z_1^1 < z_1^2 < \dots < z_1^N \quad \text{et} \quad z_2^1 > z_2^2 > \dots > z_2^N \quad \text{alors} \quad \beta_1 > \beta_{1,2} > \beta_{2,3} > \beta_{3,4} > \dots > \beta_{N-1,N} > \beta_N.$$

Selon cet ordre les deux points z^p et z^{p+1} sont adjacents dans la liste $\mathcal{Z}_E = \{z^1, z^2, \dots, z^N\}$.

L'algorithme de la méthode

⟨0⟩ (Initialisation) : Résoudre le problème $(P\beta)$ pour $\beta = 0$ et $\beta = 1$ pour trouver respectivement les deux points non dominés z^1 et z^N ($z_1^1 > z_1^N$ et $z_2^1 < z_2^N$) et déterminer le point idéal $z^{id} = (z_1^1, z_2^N)$. Poser $I = \{[z^1, z^N]\}$ et $S = \{(x^1, z^1), (x^N, z^N)\}$

Itération : Tant que $I \neq \emptyset$ faire

⟨1⟩ Retirer $[z^1, z^N]$ de I .

⟨2⟩ Calculer $\beta_{p,q}$ selon l'équation (4.10), et résoudre le problème $P(\beta_{p,q})$ défini dans (4.8). Si sa solution est z^p ou z^q alors z^p et z^q sont adjacentes dans la liste $\mathcal{Z}_E = \{z^1, z^2, \dots, z^N\}$.

⟨3⟩ Sinon, une nouvelle solution z^r non dominé est trouvé. $S = S \cup \{(x^r, z^r)\}$ et ajouter les deux intervalles $[z^p, z^r]$ $[z^r, z^N]$ à I .

4.4 Optimisation d'un critère linéaire sur l'ensemble efficace discret basée sur la norme augmentée de Tchebychev

Dans cette section nous allons décrire l'algorithme de la méthode proposée, qui résout le problème d'optimisation d'un critère linéaire sur l'ensemble de solutions efficaces discrètes d'un problème linéaire multi-objectifs, en utilisant la norme augmentée de Tchebychev.

Rappelons que le problème de l'optimisation d'une fonction linéaire \mathcal{P}_E sur l'ensemble des solutions efficaces X_E de $(MOILP)$ est :

$$(\mathcal{P}_E) \begin{cases} \max & \phi = d^t x \\ \text{s.c} & x \in X_E \end{cases} \quad (4.11)$$

Le problème relaxé de (\mathcal{P}_E) est (\mathcal{P}_R) défini par :

$$(\mathcal{P}_R) \begin{cases} \max & \phi = d^t x \\ \text{s.c} & x \in X \end{cases} \quad (4.12)$$

4.4. OPTIMISATION D'UN CRITÈRE LINÉAIRE SUR L'ENSEMBLE EFFICACE DISCRET BASÉE SUR LA NORME AUGMENTÉE DE TCHEBYCHEV

Proposition 4 Soit $\beta \in \Delta$, pour un $\rho > 0$ et suffisamment petit, alors chaque solution optimale du $P_\rho(\beta)$ est non dominée pour le (MOILP).

Preuve Soit \hat{y} une solution optimale du $P_\rho(\beta)$ et supposons que \hat{y} est dominée, nous supposons qu'il existe $y \in Y$ tel que $y \geq \hat{y}$.

$$y \geq \hat{y} (y_i \geq \hat{y}_i, \forall i \in \{1 \dots p\}), \text{ et } \exists k \in \{1 \dots p\} \text{ tel que } y_k > \hat{y}_k.$$

Pour un $\rho \geq 0$, nous avons

$$\rho \sum_{i=1}^p (y_i^{id} - y_i) < \rho \sum_{i=1}^p (y_i^{id} - \hat{y}_i)$$

et

$$\max_{i \in \{1 \dots p\}} (y_i^{id} - y_i) \leq \max_{i \in \{1 \dots p\}} (y_i^{id} - \hat{y}_i)$$

donc

$$\|y^{id} - y\| + \rho \sum_{i=1}^p (y_i^{id} - y_i) < \|y^{id} - \hat{y}\| + \rho \sum_{i=1}^p (y_i^{id} - \hat{y}_i)$$

ce qui est contradictoire avec \hat{y} est optimale pour le problème (4.3) \square .

Proposition 5 Soit $\hat{y} \in Y$ et $\hat{\beta} \in \Delta$, si \hat{y} est non dominée alors \hat{y} est la solution optimale unique du $P_\rho(\hat{\beta})$, pour un certain $\rho > 0$ est suffisamment petit.

Preuve : Soit \hat{y} une solution non dominée du (MOILP) et supposons qu'il existe une autre solution optimale \bar{y} pour le problème $P_\rho(\hat{\beta})$

on aura alors

$$\bar{\omega} + \rho \sum_{i=1}^p (y_i^{id} - \bar{y}_i) < \hat{\omega} + \rho \sum_{i=1}^p (y_i^{id} - \hat{y}_i) \quad (4.13)$$

et aussi

$$\bar{\omega} > \hat{\omega} \text{ (puisque } \hat{y} \text{ est non dominé pour plus de détail voir [61])} \quad (4.14)$$

de 4.13 et 4.14 on a

$$\sum_{i=1}^p (\bar{y}_i - \hat{y}_i) > 0 \implies \exists k \in \{1 \dots p\} | \bar{y}_k > \hat{y}_k$$

contradiction avec le fait que \hat{y} est non dominée. \square

4.4.1 Description générale de l'algorithme

L'algorithme proposé produit une solution optimale globale du problème P_E et cela sans l'énumération de toutes les solutions efficaces du problème multi-objectifs, mais seulement un sous ensemble de solutions efficaces sera fourni. Notre technique s'articule autour de deux idées essentielle ; nous caractérisons la solution non dominée par la résolution de programme de la norme augmentée de Tchebychev, la deuxième consiste à réduire progressivement le domaine d'admissibilité par l'ajout des contraintes.

Initialement, la procédure détermine le vecteur utopique Z^u , puis le problème relaxé P_R est résolu, une solution optimale x est obtenue, dont la borne supérieure du critère principal est mis à jour, et son image dans l'espace des critères $z = Cx$ est considérée comme le sommet de référence de l'isoquant $Z < Z^u$. Pour une petite valeur $\rho > 0$, le programme de la norme augmentée de Tchebychev $P_\rho(\beta)$ est résolu afin de trouver un vecteur non dominé \hat{Z} le plus proche au point utopique, dans la direction déterminée par Z^u et Z . Dans l'espace des décision plusieurs solutions efficaces peuvent avoir la même image \hat{Z} dans l'espace des critères, un autre programme est exigé pour trouver une solution efficace équivalente (alternative), si elle existe, améliorant le critère principal. Alors une nouvelle solution efficace est générée et ajoutée à la liste courante après avoir évaluée la borne inférieure du critère principal. Des nouvelles contraintes, pas seulement évite de régénérer la solution efficace courante, mais aussi élimine un sous ensemble de solutions dominées, sont imposées. L'algorithme s'arrête lorsque l'espace réalisable devient vide ou bien la borne inférieure coïncide avec la borne supérieure.

Supposons, que tout les composantes de la matrice des critères sont entiers. A chaque itération k , la région réalisable D est réduite graduellement par l'élimination de toute les solutions dominées par $C\hat{x}^k$, cette technique d'ajout des contraintes été utilisée la première fois par Klein et Hannan, 1982 [45] et aussi par Sylva et Crema, 2004 [63]. Cette élimination peut être produite par la résolution du problème suivant :

$$(P_R^k) \equiv \max\{dx, x \in D - \bigcup_{s=1}^k D_s\}.$$

Lorsque $Z(P)$ n'est pas uniformément dominante (l'image de D dans l'espace des critères contenant au moins une solution faiblement non dominée), le programme de la norme de Tchebychev n'est pas approprié pour notre approche, cependant, la norme augmentée de Tchebychev est l'alternative (voir [61]). L'utilisation de cette norme, nous permet de caractériser les solutions non dominées, et à surmonter le cas délicat des solutions non supportées (voir [15]).

4.4.2 l'algorithme de la méthode

Algorithm 3: Optimisation d'une fonction linéaire sur un ensemble efficace discret

Input

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

Output

- ↑ x_{opt} : solution optimale du problème (P_E).
- ↑ ϕ_{opt} : la valeur optimale du critère principal ϕ

Initialization

- Pour $i = 1, \dots, p$, résoudre $z_i^* = \max\{c^i x, x \in D\}$; avec $z_i^U = z_i^* + 1$ et poser la borne inférieure $M_i := \min\{c^i x, x \in D\}$.
- $\phi_{sup} := +\infty$ and $\phi_{inf} := -\infty$: les bornes superieure et inferieure de la fonction ϕ ;
- $k = 1, E_1 := \emptyset; \bar{D} := D$
- *end* := *false*

while *end* = *false* **do**

Résoudre ($P_R^k \equiv \max\{d^t x, x \in \bar{D}\}$);

if (P_R^k est irréalisable ou $\phi_{inf} \geq \phi_{sup}$ **then** x_{opt} est la solution optimale de (P_E); *end* := *true* : **Terminer**

else

soit x^k la solution optimale du (P_R^k).

Soit $\phi_{sup} = d^t x^k$, calculer le vecteur préférence β^k de $z^k = Cx^k$

soit (\hat{x}^k, \hat{z}^k) la solution optimale du $P_\rho(\beta^k)$

if $d\hat{x}^k = \phi_{sup}$ **then** $x_{opt} = \hat{x}^k, \phi_{opt} = \phi_{sup}$; *end* := *true* **Terminer**

else

Solve $Q(\hat{z}^k) \equiv \max\{dx \mid x \in D, Cx = \hat{z}^k\}$

Soit \bar{x}^k la solution optimale $Q(\hat{z}^k)$;

if $d^t \bar{x}^k > \phi_{inf}$ **then** $x_{opt} := \bar{x}^k, \phi_{inf} := d\bar{x}^k$ and $\phi_{opt} := \phi_{inf}$; soit

$E_{k+1} = E_k \cup \{\bar{x}^k\}, k = k + 1$ and $\bar{D} := D \setminus \cup_{s=1}^{k-1} D_s$;

$D_s = \{x \in \mathbb{Z}^n \mid Cx \leq C\bar{x}^s; \bar{x}^s \in E_{k-1}\}$

else

if $\phi_{inf} \geq \phi_{sup}$ **then** x_{opt} est la solution optimale du (P_E) et ϕ_{opt} la valeur optimale de ϕ ; *end* := *true* : **Terminer**

else

$x_{opt} := \bar{x}^k, \phi_{inf} := d\bar{x}^k$ and $\phi_{opt} := \phi_{inf}$; soit $E_{k+1} = E_k \cup \{\bar{x}^k\}$,

$k = k + 1$ and $\bar{D} := D \setminus \cup_{s=1}^{k-1} D_s$;

$D_s = \{x \in \mathbb{Z}^n \mid Cx \leq C\bar{x}^s; \bar{x}^s \in E_{k-1}\}$

4.4. OPTIMISATION D'UN CRITÈRE LINÉAIRE SUR L'ENSEMBLE EFFICACE DISCRET BASÉE SUR LA NORME AUGMENTÉE DE TCHEBYCHEV

Proposition 6 *L'algorithme présenté précédemment converge en un nombre fini d'itérations .*

Preuve. Puisque D est un ensemble fini, le nombre de solutions efficaces $|X_E|$ est fini. A chaque itération de l'algorithme une nouvelle solution efficace est générée et la region d'admissibilité est reduite jusqu'à que le domaine devient vide. Donc la procedure converge vers une solution optimale du problème (P_E) en un nombre fini d'itérations.

4.4.3 Exemple d'illustration

Nous reprenons l'exemple traité dans le chapitre 2 section 2.5.2.1 :

Exemple 5 *Soit le problème MOILP suivant*

$$(MOILP) \left\{ \begin{array}{l} \max \quad z_1 = x_1 + 3x_2 \\ \max \quad z_2 = -3x_1 - x_2 \\ \left\{ \begin{array}{l} -2x_1 + 5x_2 \leq 23 \\ 4x_1 + x_2 \leq 31 \\ x_1 - x_2 \leq 4 \\ -x_1 - 3x_2 \leq -8 \\ -3x_1 - x_2 \leq -8 \\ x_1, x_2 \in \mathbb{Z}_+^* \end{array} \right. \end{array} \right. \quad (4.15)$$

et le problème principal

$$(P_E) \equiv \max\{\phi = x_1 - 4x_2 \mid (x_1 \ x_2)' \in X_E\}.$$

Le problème relaxé (P_R) est donné par

$$(P_R) \equiv \max\{\phi = x_1 - 4x_2 \mid (x_1 \ x_2)' \in D\}.$$

L'ensemble des solutions efficaces du problème $(MOILP)$ est donné dans le tableau 4.1, ainsi en figure(Fig.4.3), dont 6 solutions efficaces non supportées parmi 8 solutions efficaces.

ρ est fixé à 0.004.

Step 0 Initialisation.

- Point ideal $z^* = (27 \ -8)'$; $z^U = (28 \ -7)'$; les bornes inférieures des fonction objectifs sont $M_1 = 0, M_2 = -25$;

4.4. OPTIMISATION D'UN CRITÈRE LINÉAIRE SUR L'ENSEMBLE EFFICACE DISCRET BASÉE SUR LA NORME AUGMENTÉE DE TCHEBYCHEV

Solutions efficaces	(1,5)	(2,5)	(3,5)	(4,5)	(4,6)	(5,6)	(6,7)	(6,6)
Solutions non dominées	(16,-8)	(17,-11)	(18,-14)	(19,-17)	(22,-18)	(23,-21)	(27,-25)	(24,-24)

TABLE 4.1 – Ensemble des solutions efficaces/nondominées de l'exemple 5

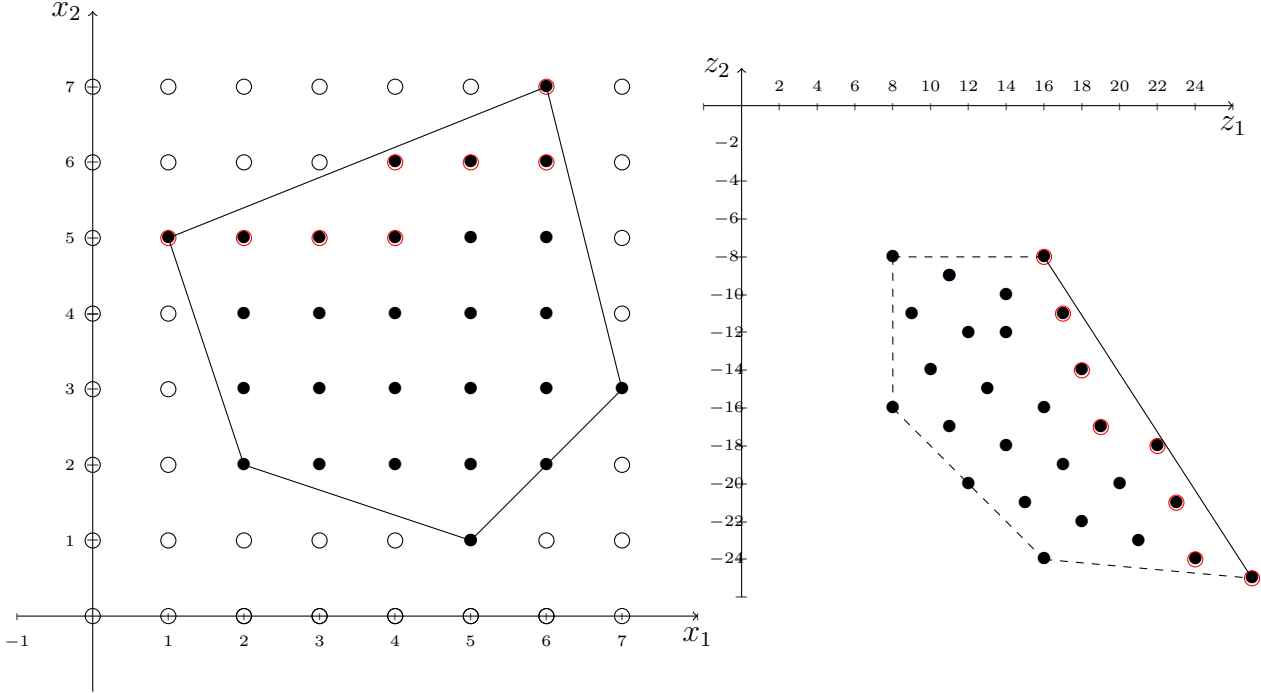


FIGURE 4.3 – Ensemble des solutions efficaces/nondominées de l'exemple 5.

– $\phi_{inf} := -\infty$, $\phi_{sup} := +\infty$, $k = 1$, $X_{E_1} := \emptyset$ et $end := false$.

L'itération 1

On résout le problème relaxé $P_R^1 \equiv \max\{\phi | x \in D\}$

La solution optimale est $x^1 = (5 \ 1)'$. Soit $z^1 = Cx^1 = (8 \ -16)'$, son image dans l'espace des critères. $\phi_{sup} = dx^1 = 1$. $\phi_{inf} \not\leq \phi_{sup}$. Nous calculons le vecteur de préférence β^1 associé à z^1 , $\beta^1 = (0.3103, 0.6897)$.

On résout le programme Tchebychev généralisé $P_\rho(\beta^1)$ défini comme suit

$$P_\rho(\beta^1) \left\{ \begin{array}{l} \min \quad \omega - 0.004(-2x_1 + 2x_2) \\ \omega \geq 0.3103(28 - x_1 - 3x_2) \\ \omega \geq 0.6897(-7 + 3x_1 + x_2) \\ x_1, x_2 \in D \\ \omega \geq 0 \end{array} \right. \quad (4.16)$$

4.5. ÉVALUATION NUMÉRIQUE

La solution $\hat{z}^1 = (17; -11)$ est un point non dominé avec une distance pondérée de Tchebychev minimal, nous obtenons $\hat{x}^1 = (2; 5)$ et $\phi(\hat{x}^1) = -18 \neq \phi_{sup}$. Nous résolvons $Q(\hat{z}^1) \equiv \max\{dx, Cx = \hat{z}^1, x \in D\}$, dont le but d'avoir une solution alternative améliorant le critère principal. $\bar{x}^1 = (2; 5) = \hat{x}^1$; $\phi_{inf} = -18 \not\leq \phi_{sup} = 1$ et $X_{E_2} = \{\bar{x}^1\}$.

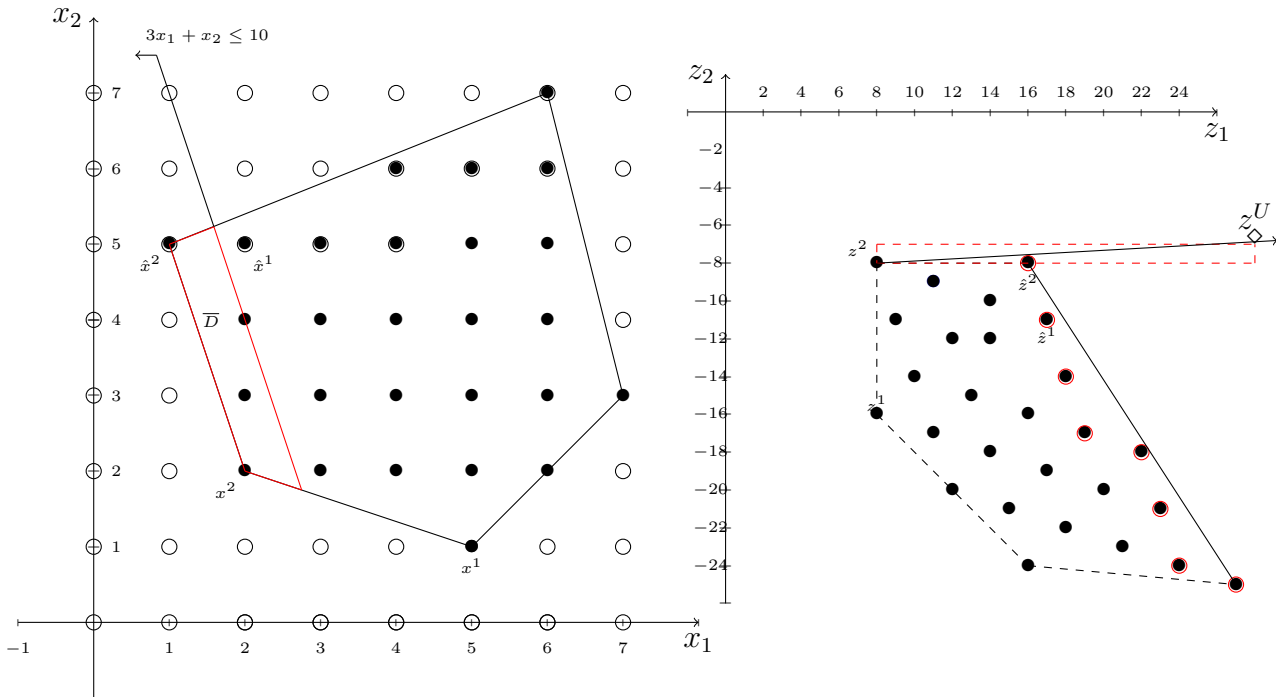


FIGURE 4.4 – Iteration 2.

Les résultats (pour $k \geq 2$) sont résumés dans le tableau (Table 4.2) et représentés dans la figure Fig.4.5. $(x_{opt}, \phi_{opt}) = ((4 \ 5)', -16)$ est la solution optimale obtenue à la 4^{me} itération.

4.5 Évaluation numérique

4.5.1 Implémentation et instances

La méthode proposée dans ce chapitre est évaluée sur un ensemble d'instances aléatoirement générées. L'algorithme est codé en environnement MATLAB et exécuté sur un PC HP Compaq, avec un Pentium(R) dual CPU à 1.4 GHz, 1 Go de RAM et Windows Vista. Nous utiliserons le solveur CPLEX 12.2 library pour la résolution des programmes linéaires en nombres entiers et les programmes linéaires mixtes. Les instructions principales de l'algorithme s'allongent dans la résolution de deux programmes linéaires spécifiques en

4.5. ÉVALUATION NUMÉRIQUE

iter k	(P_R)	Value of β_i^k	$(P_\rho(\beta^k))$	$(Q(\hat{z}^k))$	$\phi_{inf} \geq \phi_{sup}$ or (P_R) is unfeasible	OPT. SOL. x_{opt} ϕ_{opt}
	x^k z^k		\hat{x}^k \hat{z}^k	\bar{x}^k \bar{z}^k		
1	$(5 \ 1)'$	0.3103	$(2 \ 5)'$	$(2 \ 5)'$	False	$(2 \ 5)'$ -18
	$(8 \ -16)'$	0.6897	$(17 \ -11)'$	$(17 \ -11)'$		
2	$(2 \ 2)'$	0.0476	$(1 \ 5)'$	$(1 \ 5)'$	False	$(2 \ 5)'$ -18
	$(8 \ -8)'$	0.9524	$(16 \ -8)'$	$(16 \ -8)'$		
3	$(6 \ 4)'$	0.6000	$(4 \ 6)'$	$(4 \ 6)'$	False	$(2 \ 5)'$ -18
	$(18 \ -22)'$	0.4000	$(22 \ -18)'$	$(22 \ -18)'$		
4	$(4 \ 5)'$	0.5263	$(4 \ 5)'$	$(4 \ 5)'$	True	$(4 \ 5)'$ -16
	$(19 \ -17)'$	0.4737	$(19 \ -17)'$	$(19 \ -17)'$		

TABLE 4.2 – Résultats obtenus en chaque itération de l'exemple 5

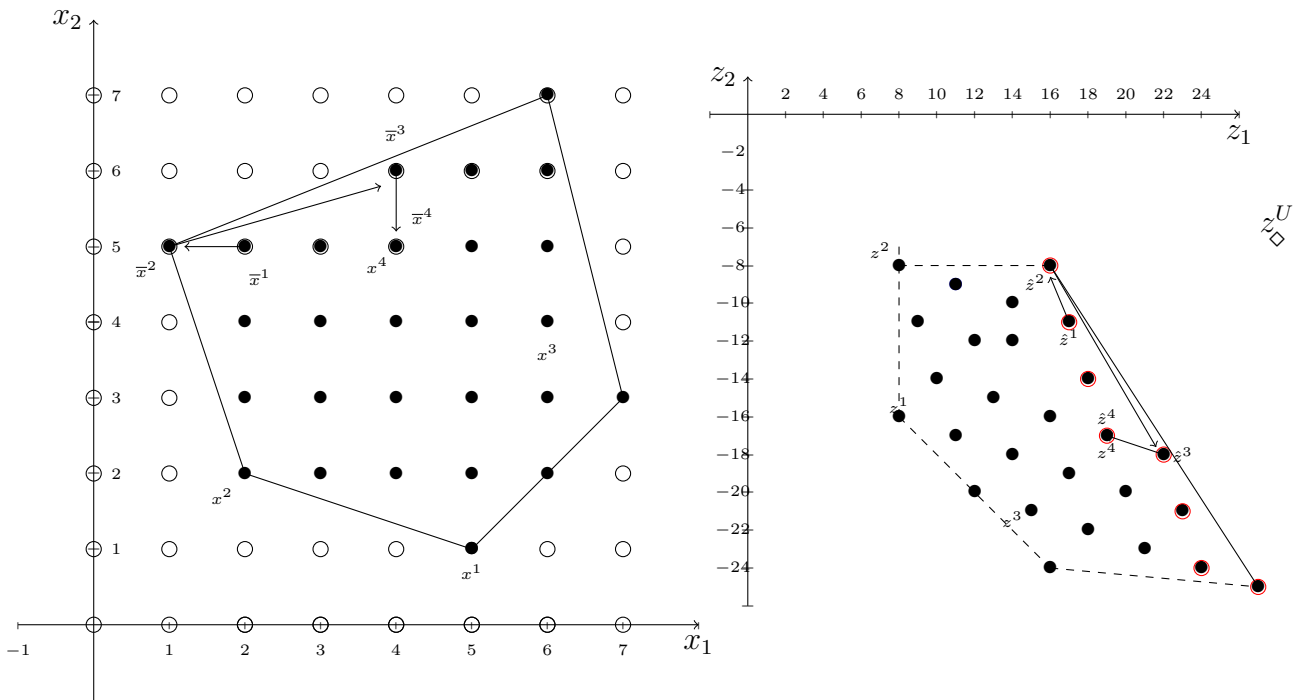


FIGURE 4.5 – Liste des solutions efficaces/non-dominées de l'exemple 5

nombre entier, le problème relaxé et le programme mixte de la norme augmentée de Tchebychev. L'algorithme a été testé avec des problèmes MOILP, aléatoirement générés selon des distribution uniforme, tel que les composantes des matrice A , C et le vecteur de seconde membre b se varies respectivement suivant les intervalles $[1, 30]$, $[-20, 20]$ et $[50, 150]$. Le vecteur d est généré de même que C . Pour éviter d'avoir un domaine de fai-

4.5. ÉVALUATION NUMÉRIQUE

sabilité vide, toutes les contraintes de chaque problème sont de type \leq . De plus, puisque tous les coefficients de la matrice A et le vecteur b sont positifs, alors la région réalisable est bornée. Le nombre de critères p prend les valeurs 3, 5 et 8. Les résultats obtenus sont représentés dans les tableaux 4.3 et 4.4, dont la moyenne, le minimum et le maximum, du temps d'exécution (CPU time en second) et de nombres d'itérations (Iter), sont obtenus pour dix instances aléatoirement générées.

4.5. ÉVALUATION NUMÉRIQUE

p	$p=3$		$p=5$		$p=8$		
	$m \times n$	CPU(Sec.)	Iter	CPU(Sec.)	Iter	CPU(Sec.)	Iter
10×10		0.38	2	0.45	1	0.59	2
		[0.26, 0.54]	[1, 3]	[0.36, 0.64]	[1, 3]	[0.54, 1.13]	[1, 5]
10×15		0.67	2.5	0.49	1	0.70	1
		[0.32, 1.76]	[1, 7]	[0.46, 1.07]	[1, 4]	[0.64, 0.82]	[1, 2]
15×15		0.36	1	0.49	1	0.63	1
		[0.28, 0.70]	[1, 4]	[0.42, 0.67]	[1, 3]	[0.05, 14.86]	[1, 5]
20×15		0.57	3.5	0.4715	1	0.67	1
		[0.28, 1.07]	[1, 5]	[0.39, 0.79]	[1, 3]	[0.60, 0.97]	[1, 1]
25×15		0.42	2	0.50	1	0.65	1
		[0.30, 0.56]	[1, 3]	[0.40, 0.71]	[1, 3]	[0.58, 1.10]	[1, 2]
25×25		0.48	2	0.63	1.5	0.94	1
		[0.31, 0.69]	[1, 3]	[0.46, 1.62]	[1, 5]	[0.71, 1.93]	[1, 8]
35×25		0.53	2	0.55	1	0.77	1
		[0.37, 1.31]	[2, 5]	[0.47, 1.06]	[1, 4]	[0.61, 4.15]	[2, 10]
35×30		0.66	3	0.58	1	0.92	1
		[0.40, 2.47]	[1, 6]	[0.47, 1.27]	[1, 3]	[0.70, 3.02]	[1, 13]
40×30		0.56	2	0.82	2	0.85	1
		[0.40, 0.85]	[1, 4]	[0.50, 2.35]	[2, 7]	[0.71, 2.38]	[1, 3]
40×40		0.88	3	0.92	2	1.1952	1.5
		[0.39, 2.64]	[1, 6]	[0.54, 1.80]	[1, 4]	[0.81, 2.60]	[1, 4]
50×50		0.91	3	0.86	2	1.12	1
		[0.60, 1.54]	[1, 5]	[0.56, 1.68]	[1, 5]	[0.95, 2.54]	[1, 3]
50×60		1.01	2.5	2.00	3.5	2.04	1
		[0.57, 5.38]	[1, 7]	[0.96, 3.52]	[1, 5]	[1.21, 3.61]	[1, 3]
60×60		1.07	2.5	2.13	2	1.85	1
		[0.49, 3.47]	[1, 7]	[0.70, 4.02]	[1, 5]	[1.04, 3.45]	[1, 3]
70×60		4.43	5.5	5.07	5	3.50	1.5
		[1.26, 11.41]	[1, 8]	[2.33, 8.67]	[2, 7]	[1.97, 8.53]	[1, 5]
70×80		2.71	3	2.83	3.5	2.83	1
		[1.26, 11.41]	[1, 8]	[2.33, 8.67]	[2, 7]	[1.97, 8.53]	[1, 5]
80×80		5.60	4.5	8.39	4.5	4.65	2
		[4.06, 16.25]	[3, 7]	[1.29, 29.82]	[1, 9]	[3.23, 9.10]	[1, 5]
80×90		1.40	1	2.33	1	3.29	1
		[0.93, 10.24]	[1, 5]	[1.79, 19.84]	[1, 7]	[2.79, 14.95]	[1, 5]
80×100		1.64	4	10.66	1	6.62	2
		[1.12, 107.30]	[1, 15]	[1.70, 167.83]	[2, 12]	[4.80, 17.61]	[1, 6]

TABLE 4.3 – Résultats obtenus pour : $10 \leq n \leq 100$ et $10 \leq m \leq 100$

4.5. ÉVALUATION NUMÉRIQUE

p	$p=3$		$p=5$		$p=8$	
	$m \times n$	CPU(Sec.)	Iter	CPU(Sec.)	Iter	CPU(Sec.)
100×100	8.67	4	9.33	3.5	1.45	1
	[4.21, 21.51]	[1, 7]	[1.37, 143.30]	[1, 13]	[1.33, 14.18]	[1, 10]
100×120	2.04	4	2.18	2.5	2.2659	1
	[1.00, 2.74]	[1, 5]	[1.34, 2.83]	[1, 3]	[1.89, 4.29]	[1, 4]
120×120	2.02	2.5	2.0787	2	2.43	1
	[1.00, 3.78]	[1, 6]	[1.45, 4.36]	[1,4]	[2.12, 40.11]	[1, 32]
130×140	3.2147	2	3.33	2	3.84	1
	[1.41, 6.56]	[1,5]	[1.99,12.58]	[1,5]	[2.71, 22.43]	[1, 11]
140×150	3.80	2	2.87	1	3.52	1
	[1.62,7.37]	[1, 5]	[2.24,18.22]	[1, 7]	[3.12,26.93]	[1, 3]
150×160	4.57	3.5	4.42	2	4.22	1
	[2.92, 9.03]	[1,5]	[2.76, 13.51]	[1, 5]	[3.35, 5.48]	[1, 2]
160×170	5.41	2.5	5.92	2	4.51	1
	[3.27, 11.09]	[2,6]	[3.31, 10.90]	[1, 5]	[4.20, 7.33]	[1, 2]
170×180	6.72	3	5.97	2	5.20	1
	[2.65, 15.30]	[1, 6]	[3.54, 9.90]	[1, 3]	[4.74, 12.36]	[1, 4]
180×200	5.20	2.5	9.04	3	6.28	1
	[2.97, 10.94]	[1, 5]	[4.28, 33.63]	[1, 7]	[5.83, 49.08]	[1, 14]
200×200	7.77	3	7.24	2	9.35	2
	[5.70, 11.68]	[2, 5]	[4.64, 10.24]	[1, 2]	[6.42, 12.58]	[1, 3]
200×220	9.66	3	8.99	2	10.54	2
	[4.41, 18.34]	[1, 5]	[5.67, 15.83]	[1, 4]	[7.62, 11.72]	[1, 2]

TABLE 4.4 – Résultats obtenus pour : $100 \leq n \leq 220$ et $100 \leq m \leq 200$

4.5.2 Interprétation des résultats

Les résultats obtenus sont encourageants, ce qui confirme l'efficacité de l'algorithme proposé en terme du temps d'exécution relativement à la taille considérable des instances traitées. Notons que l'exécution du programme de certaines instances n'aboutit pas aux résultats qui est probablement dû à la dégénérescence. Remarquons aussi que le temps d'exécution n'est pas forcément proportionnel à la taille de l'instance, et ça est dû à notre connaissance d'une part aux nombres de solutions efficaces visitées(générées) par l'algorithme et d'autre part par la structure de l'instance. Autrement dit, le nombre des

solutions efficaces générées n'est pas proportionnel au cardinal de l'ensemble des solutions efficaces d'une instance.

4.6 Conclusion

Dans ce chapitre, une nouvelle méthode pour le problème d'optimisation d'une fonction linéaire sur l'ensemble des solutions efficaces discret à été présentée. C'est une méthode exacte basée sur deux idées, la première consiste en premier lieu à la caractérisation de la solution non dominée, et en second à la recherche de la solution efficace qui améliore la fonction objectif principale. La deuxième idée est un processus d'ajout des coupes pour éliminer les solutions dominées trouvées précédemment, par conséquent, réduire la région admissible progressivement.

L'algorithme à été codé en utilisant l'environnement MATLAB avec l'intégration du CPLEX 12.2 library. Il est testé sur plusieurs problèmes aléatoirement générés. Les résultats obtenues sont très encourageants et montre que l'algorithme proposé est très efficace en termes de nombre de solutions efficaces générées (ne dépasse pas en valeur médiane 5) avec des problèmes d'une dimension considérables (200 contraintes, 220 variable et 8 objectifs).

Conclusion générale et perspectives

Dans ce travail, nous avons proposé un algorithme exact, permettant de résoudre un problème d'optimisation d'une fonction linéaire sur l'ensemble des solutions efficaces d'un programme multi-objectifs linéaire en nombres entiers.

Notre choix, de traiter un tel problème, est justifié d'une part, par l'intérêt grandissant que les problèmes multi-objectifs suscitent dans la vie actuelle où plusieurs objectifs concurrents doivent être satisfaits. La plupart du temps ces problèmes, particulièrement ceux qui sont de nature discrètes, sont NP-difficiles.

Et d'autre part, les techniques de résolution de tels problèmes évitent l'énumération de toutes les solutions efficaces du problème *MOILP*, car les décideurs ne sont pas souvent intéressés par toutes les solutions efficaces, mais seulement par celles qui optimisent la valeur d'un critère prédéterminé. Par conséquent les méthodes évitant d'explorer toutes les solutions efficaces sont très appréciées.

L'algorithme proposé opère en deux phases essentielles, la phase de caractérisation de solution efficaces/non dominée et la phase de réduction du domaine d'admissibilité.

Dans la littérature, il existe plusieurs caractérisations de la solution efficace, particulièrement dans le cas continu par exemple comme celles données dans [76] et celle du test d'efficacité. Ce dernier a été développé par Ecker et Kouada, 1978 [24] utilisé dans les trois premières méthodes [1, 42, 17] ayant été élaborées pour la résolution du problème en question, dans notre contribution qu'on peut qualifier de quatrième méthode, la scalarisation de la norme de Tchebychev a été employée, cette caractérisation, et ce d'après Bowman, 1976 [15], est la plus appropriée pour la génération des solutions efficaces/non dominées, particulièrement celle qui sont non supportées. La deuxième phase consiste en la réduction progressive du domaine réalisable où de recherche de la solution optimale, et cela par l'ajout des contraintes connu dans la littérature par "corner constraint". Ces dernières éliminent les solutions dominées ainsi que les solutions efficaces/non dominées générées déjà. Cette technique a été utilisée dans les méthodes de Jorge, 2009 [42] et de

Chaabane et Pirlot [17].

Nous avons implémenté la méthode proposée sous l'environnement Matlab, en utilisant CPLEX 12.2 Library pour résoudre les programmes linéaire en nombre entiers. A travers des expérimentations numériques, nous avons constaté l'efficacité de notre algorithme, même lorsque la taille du problème considéré est importante.

Ce travail a fait l'objet d'une publication dans la revue "International Transaction in Operation Research(2012) "et une autre soumission dans journal "RAIRO-Operations Research (2015)".

Parmi les travaux qui peuvent présenter des perspectives et que nous souhaitons aborder pour l'avenir, nous y trouvons par exemple :

- Résolution d'un problème d'optimisation d'une fonction non linéaire, sur l'ensemble des solutions efficaces discrètes, nous pensons plus particulièrement au cas où la fonction est quadratique en variables discrètes.
- Résolution du problèmes d'optimisation sur l'ensemble des solutions efficaces par des méta-heuristiques.

Bibliographie

- [1] M. Abbas and D. Chaabane. Optimizing a linear function over an integer efficient set. *European Journal of Operational Research*, 174 :1140–1161, 2006.
- [2] M. J. Alves and J. Climaco. A review of interactive methods for multiobjective integer and mixed-integer programming. *European Journal of Operational Research*, 180 :99–115, 2007.
- [3] M.J. Alves and J. Climaco. An interactive reference point approach for multiobjective mixed-integer programming using branch-and-bound. *European journal of operational research*, 124(3) :478–494, 2000.
- [4] Andreas Antoniou and Wu-Sheng Lu. *Practical Optimization Algorithms and Engineering Application*. Springer science+ business media, New York, 2007.
- [5] P. Armand and C. Malivert. Determination of the efficient set in multiobjective linear programming. *Journal of Optimization Theory and Applications*, 70(3) :467–489, 1991.
- [6] V. Barichard. *Approches hybrides pour les problèmes multiobjectifs*. PhD thesis, Ecole doctorale d’Angers, 2003.
- [7] H. P. Benson. A finite nonadjacent extreme-point search algorithm for optimization over the efficient set. *Journal of Optimization Theory and Applications*, 73 :47–64, 1992.
- [8] H. P. Benson and D. Lee. Outcome-based algorithm for optimizing over the efficient set of a bicriteria linear programming problem. *Journal of Optimization Theory and Applications*, 88 :77–105, 1996.
- [9] H.P. Benson. Existence of efficient solutions for vector maximization problems,. *Jornal of Optimization Theory and Applications*, 26(4) :569–580, 1978.

- [10] H.P. Benson. Further analysis of an outcome set-based algorithm for multiple-objective linear programming. *Journal of Optimization Theory and Applications*, 97(1) :1–10, 1998.
- [11] H.P. Benson. Hybrid approach for solving multiple-objective linear programs in outcome space. *Journal of Optimization Theory and Applications*, 98(1) :17–35, 1998.
- [12] H.P. Benson. An outer approximation algorithm for generating all efficient extreme points in the outcome set of a multiple objective linear programming problem. *Journal of Global Optimization*, 13 :1–34, 1998.
- [13] G. W. Bitran. Linear multiple objective programs with zero-one variables. *Mathematical Programming*, 13 :121–139, 1977.
- [14] S. Bolintineanu. Minimization of a quasi-concave function over an efficient set. *Mathematical Programming*, 61 :89–110, 1993.
- [15] V. J. Bowman. On the relationship of the tchebycheff norm and the efficient frontier of multiple-criteria objectives. In *Thiriez H. & Zionts S. (Eds)*, pages 76–85, 1976.
- [16] D. Chaabane. *Contribution a l'Optimisation Multicritère en Variables Discrettes*. PhD thesis, Faculté Polytechnique de Mons, 2006.
- [17] D. Chaabane and M. Pirlot. A method for optimizing over the integer efficient set. *Journal of Industrial and management Optimization*, 6(4) :811–823, 2010.
- [18] Djamel Chaabane, Boualem Brahmi, and Zoubir Ramdani. The augmented weighted tchebychev norm for optimizing a linear function over an integer efficient set of a multicriteria linear program. *Intl. Trans. in Op. Res*, 19 :531–545, 2012.
- [19] L. G. Chalmet, L. Lemondis, and D. J. Elzinga. An algorithm for the bi-criterion integer programming problem. *European Journal of Operational Research*, 25 :292–300, 1981.
- [20] J. Dauer and O. Saleh. Constructing the set of efficient objective values in multiple objective linear programs. *European Journal of Operational Research*, 46(3) :358–365, 1990.
- [21] J. P. Dauer and R. J. Gallagher. A combined constraint space, objective-space approach for determining high-dimensional maximal efficient faces of multiple objective linear programs. *European journal of operational research*, 88(2) :368–381, 1996.
- [22] J.P. Dauer. On degeneracy and collapsing in the construction of the set of objective values in a multiple objective linear program. *Annals of Operations Research*, 46(2) :279–292, 1993.

- [23] J.P Dauer and T.A Fosnaugh. Optimization over the efficient set. *Journal of Global Optimization*, 7 :261–277, 1995.
- [24] J.G. Ecker and I.A. Kouada. . finding all efficient extreme points for multiple objective linear programs. *Mathematical Programming*, 14 :249–261, 1978.
- [25] J.G. Ecker and J.H. Song. Optimizing a linear function over an efficient set. *Journal of Optimization Theory and Applications*, 83 :541–563, 1994.
- [26] Matthias Ehrgott. *Multicriteria Optimization*. Lecture Notes in Economics and Mathematical Systems : Springer Verlag, 2005.
- [27] J.G. Eker and I.A. Kouada. Finding all efficient point for linear multiple objective programs,. *Mathematical programming*, 8(8) :375–377, 1975.
- [28] P. K. Eswaran, A. Ravindran, and H. Moskowitz. Algorithms for nonlinear integer bicriterion problems. *Journal of Optimization Theory and Applications*, 2 :63, 2000.
- [29] P.K. Eswaran, A. Ravindran, and H. Moskowitz. Algorithms for nonlinear integer bicriterion problems,. *Journal of Optimization Theory and Applications*, 63,(2) :261–279, 1989.
- [30] C. Ferreira, J. Climaco, and J. Paix ao. The location-covering problem : A bicriterion interactive approach. *Investigacion Operativa*, 4 :119–139, 1994.
- [31] M.P. Fourman. Compaction of symbolic layout using genetic algorithm. *in proceedings of the first international conference on genetic algorithms (ICGA)*, 1985.
- [32] T. Friesz, G. Anandalingam, N. Mehta, K. Nam, S. Shah, and R. Tobin. The multiobjective equilibrium network design problem revisited : A simulated annealing approach. *EuropeanJournal of Operational Research*, 65 :44–57, 1993.
- [33] K. Fujimura. Path planning with multiple objectives. *IEEE Robotics and Automation Society Magazine*, 3 :33–38, 1996.
- [34] K. Fujita, N. Hirokawa, S. Akagi, S. Kimatura, and H. Yokohata. Multi-objective optimal design of automotive engine using genetic algorithm. *In Design Engineering Technical Conferences DETC 98, Atlanta, Georgia*, pages 1–11, 1998.
- [35] J. Fülöp. A cutting plane algorithm for linear optimization over the efficient set. *Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, Berlin*, 405 :374–385, 1994.
- [36] T. Gal. A general method for determining the set of all efficient solutions to a linear vector maximum problem. *European Journal of Operational Research*, 5 :307–322, 1977.

- [37] A.M. Geoffrion. Proper efficiency and the theory of vector maximization,. *Journal of Mathematical Analysis and Applications*, 22 :618–630, 1968.
- [38] R.E. Gomory. Outline of an algorithm for integer solutions to linear programs. , *Bulletin of the AMS*, 64 :275–278, 1958.
- [39] R. Horst and N.V. Thoai. Maximizing a concave function over the efficient or weakly efficient set. *European Journal of Operational Research*, 117 :239–252, 1999.
- [40] R. Horst, N.V. Thoai, Y. Yamamoto, and D. Zenke. On optimization over the efficient set in linear multicriteria programming. *Journal of Optimization and Theory Applications*, 134 :433–443, 2007.
- [41] H. Isermann. The enumeration of the set of all efficient solutions for a linear multiple objective program. *Operational Research Quarterly*, pages 711–725, 1977.
- [42] J.M. Jorge. An algorithm for optimizing a linear function over an integer efficient set. *European Journal of Operational Research*, 1(195) :98–103, 2009.
- [43] J. Karaivanova, P. Korhonen, S. Narula, J. Wallenius, and V. Vassilev. A reference direction approach to multiple objective integer linear programming. *European Journal of Operational Research*, 81 :176–187, 1995.
- [44] J.N. Karaivanova, S.C. Narula, and V.S. Vassilev. An interactive procedure for multiple objective integer linear programming problems. *Journal of Optimization Theory and Applications*, 68(3) :344–351., 1977.
- [45] D. Klein and E. Hannan. An algorithm for multiple objective integer linear programming problem. *European Journal of Operational Research*, 9 :378–385, 1982.
- [46] M. Luque, F. Ruiz, and R.E. Steuer. Modified interactive chebyshev algorithm (mica) for convex multiobjective programming. *European Journal of Operational Research*, 3 :557–564, 2010.
- [47] O. Marcotte and R.M. Soland. An interactive branch-and-bound algorithm for multiple criteria optimization. *Management Science*, 32 :1231–1240, 1986.
- [48] Frédéric Messine. *Méthodes d’Optimisation Globale basées sur l’Analyse d’Intervalle pour la Résolution de Problèmes avec Contraintes*. PhD thesis, Université Paul Sabatier Toulouse III,, 1997.
- [49] G.L. Nemhauser and L.A. Wolsey. Integer and combinatorial optimization. *Series in discrete mathematics and optimization. A Wiley-Interscience. Wiley*, 1988.
- [50] P. Neumayer and D. Schweigert. Three algorithms for bicriteria integer linear programs. *Special Issue : Decision Theory (Fehrenbach, 1992; Schwangau, 1993).OR Spektrum*, 16(4) :267–276., 1994.

- [51] J. Philip. Algorithms for the vector maximization problem. *Mathematical Programming*, 2 :207–229, 1972.
- [52] J. Philip. Vector maximization at a degenerate vertex. *Mathematical Programming*, 1(13) :357–359, 1977.
- [53] T.Q. Phong and T.Q. Tuyen. Bisection search algorithm for optimizing over the efficient set. *Vietman Journal of Mathematics*, 2000.
- [54] T.K. Ralphs, M.J. Saltzman, and M.M. Wiecek. An improved algorithm for solving biobjective integer programs. *Annals of Operational Research*, 147 :43–70, 2006.
- [55] T.K. Ralphs, M.J. Saltzman, and M.M. Wiecek. An improved algorithm for solving biobjective integer programs. *Ann Oper Res*, 147 :43–70, 2007.
- [56] R. Ramesh, M. H. Karwan, and S. Zionts. Preference structure representation using convex cones in multicriteria integer programming. *Management Science*, 35 :1092–1105, 1989.
- [57] S. Singiresu Rao. *Engineering Optimization Theory and Practice. third edition.* A Wiley-interscience publication John Wiley & Sons. New York, 1996.
- [58] S. Sayin. Optimizing over the efficient set using a top-down search of faces. *Operations Research*, 48 :65–72, 2000.
- [59] B. Schandl, K. Klamroth, and M.M. Wiecek. Norm-based approximation in bicriteria programming,. *Computational Optimization and Applications*, 20(1) :23–42, 2001.
- [60] A. Schrijver. *Theory of linear and integer programming.* A Wiley-interscience publication John Wiley & Sons., 1987.
- [61] R. E. Steuer. *Multiple Criteria Optimization :Theory, Computation and Application.* John Wiley, New York,, 1986.
- [62] R.E. Steuer and E.E. Choo. An interactive weighted tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 26 :326–344, 1983.
- [63] J. Sylva and A. Crema. A method for finding the set of non-dominated vectors for multiple objective integer linear programs. *European Journal of Operational Research*, 158 :46–55, 2004.
- [64] 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 Reserach*, 180 :1011–1027, 2007.
- [65] J. Teghem. *Programmation Linéaire.* Editions de l’Université de Bruxelles SMA, 2003.

- [66] P.T. Thach, H. Konno, and D. Yokota. Dual approach to nminimization on the set of paretooptimal solutions. *Journal of Optimization Theory and Applications*, 88 :689–707, 1996.
- [67] N.V. Thoai. A class of optimization problems over the efficient set of a multiple criteria nonlinear programming problem. *European Journal of Operational Research*, 122 :58–68, 2000.
- [68] N.V. Thoai. Conical algorithm in global optimization for optimizing over efficient sets. *Journal of Global Optimization*, 18 :321–336, 2001.
- [69] L. W. Trijbsch, Van Doorne A. G., and W. J. Selen. Simplified molp algorithm : the molp-s procedure. *Computers & operations research*, 18(8) :709–716, 1975.
- [70] B. Villarreal and M. H. Karwan. Multicriteria integer programming : A (hybrid) dynamic programming recursive approach. *Mathematical Programming*, 21 :204–223, 1981.
- [71] Phillips W. Beightler. *Foundations of optimization*. Englewood Cliffs, 1979.
- [72] R.E. Wendell and D.N. Lee. Efficiency in multiple objective optimization problems,. *Mathematical programming*, 12(3) :406–414, 1977.
- [73] D.J. White. The maximization of a function over the efficient set via a penalty function approach. *European Journal of Operational Research*, 94 :143–153, 1996.
- [74] A.P. Wierzbicki. The use of reference objectives in multiobjective optimization. In : Fandel, G. (ed.), *Multiple Criteria Decision Making, Theory and Application*. Springer-Verlag, Berlin, 5 :468–486, 1980.
- [75] L.A. Wolsey. Integer programming. *Series in discrete math ematics and optimization*. A Wiley-Interscience. Wiley, 1998.
- [76] Y. Yamamoto. Optimization over the efficient set : overview. *Journal of Global Optimization*, 22 :285–317, 2002.
- [77] M. Zeleny. Linear multiobjective programming. *Lecture Notes in Economics and Mathematical Systems*, 95, 1974.

Résumé : Le but principal de ce travail est l'étude d'une classe particulière de l'optimisation non convexe à l'occurrence le problème d'optimisation d'une fonction linéaire (appelée aussi critère principal) sur l'ensemble de solutions efficaces d'un problème linéaire multi-objectifs en nombres entiers(MOILP). Et pour la résolution du problème en question, un algorithme exact, basé sur la norme augmentée de Tchebychev et évite d'énumérer tout l'ensemble efficace, a été proposé. Dans cet algorithme, deux phases importantes sont à considérer : la phase d'ajout des contraintes de coupes pour la réduction progressive du domaine réalisable, afin de trouver une solution optimale relative à ce domaine réduit, et la phase de génération d'une solution efficace/non dominée qui domine la solution optimale déjà trouvée en première phase, et cela par la norme augmentée de Tchebychev. Une évaluation numérique de l'algorithme proposé en l'implémentons sous l'environnement Matlab utilisant CPLEX 12.2 Library a été présentée. Cette méthode développée est motivée par des résultats intéressants et encourageants. Ce travail a fait une première publication dans la revue "International Transactions in Operational Research(2012)" et une autre soumission dans le journal "RAIRO-Operations Research (2015)".

Mots clé : Non convexe, Optimisation multi-objectifs en nombres entiers, norme augmentée de Tchebychev.

Abstract : The principal aim of this work is the study of a particular class of non-convex optimization problem and more specially the case of optimizing a linear function (also called main criterion) over an integer efficient set of a multi-objective linear program. In this way, an exact algorithm is proposed and used to obtain the global optimal solution. We achieve this aim by combining two ideas : one consists of solving the augmented weighted Tchebychev program in the outcome space criteria to characterize non-dominated objective solutions, a second stage of optimization is required to find an efficient solution that improves the main objective. The second idea is a process of added cuts for eliminating the dominated solutions found previously, therefore reducing progressively the admissible region. The algorithm was coded using the MATLAB environment utilizing the CPLEX 12.2 library. the method developed is motivated by the interesting and encouraging results. This work was first published in the journal "International Transactions in Operational Research (2012)" and another submission in "RAIRO - Operations Research Journal (2015)"

KeyWords : Nonconvex ; integer programming ; multiple objective ; Tchebychev metrics.
