

République Algérienne Démocratique et populaire
Ministère de l'enseignement Supérieur et de la Recherche Scientifique

Université M'hamed Bougara de BOUMERDES

Faculté des Sciences

Département de Mathématiques



Mémoire présenté pour l'obtention du diplôme de Master en Recherche
Opérationnelle
Option : Recherche Opérationnelle, Optimisation et Management Stratégique

Réalisé par :
M^{lle} CHAID Oumayma Zahra
M^{lle} GUEZGOUZ Saadia

*Optimisation linéaire sur l'ensemble
efficient d'un problème multi-objectif
linéaire discrêt*

Soutenu publiquement le 30/06/2019 devant le jury :

- M ^r BEZOUÏ Madani	President	M.A.A	U.M.B.B
- M ^{me} DRICI Wassila	Promotrice	M.A.A	U.M.B.B
- M ^{me} KHODJA Karima	Examinatrice	M.A.A	U.M.B.B

Année universitaire : 2018/2019

Table des matières

Introduction général	4
1 Etat de l'art sur l'optimisation multi-objectifs	6
1.1 Introduction	6
1.2 Définition	6
1.3 Concepts de base et définitions	7
1.3.1 Convexité	7
1.3.2 Points particuliers	8
1.4 Dominance et optimalité	9
1.4.1 Surface de compromis	11
1.4.2 Front de pareto	12
1.4.3 Structure de front de pareto	12
1.4.4 Dominance de Geoffrion	13
1.5 Choix de la méthode d'aide a la décision	14
1.6 Approches de résolution d'un problème multi-objectif (MOP)	15
1.6.1 Approche mono-objectif	15
1.6.2 Approche multi-objectif	17
1.7 Conclusion	18
2 La Programmation multi-objectif linéaire en nombres entiers	19
2.1 Introduction	19
2.2 Définition	19
2.3 Formulation d'un problème de multi-objectif en nombres entiers <i>MOILP</i> .	20
2.4 Méthode de résolution d'un problème MOILP	20
2.4.1 Méthode de D.Klein et E.Hannan	21

2.4.2	Méthode A.Sylva et J.Crema	22
2.5	Conclusion	24
3	L'optimisation d'un critère sur un ensemble efficient	25
3.1	Introduction	25
3.2	Méthodes de résolution d'un problème d'optimisation multi-objectif sur l'ensemble efficace dans le cas discret	26
3.2.1	Définitions et notations	26
3.2.2	Méthode de Jorge [28]	28
3.2.3	Description de la méthode	28
3.2.4	Formulation de l'algorithme	29
3.2.5	Exemple	31
3.2.6	Méthode de Chaabane et Pirlot [9]	36
3.2.7	Description de la méthode	36
3.2.8	Exemple	40
3.2.9	La Méthode Chergui et Ouail [12]	50
3.2.10	Exemple	53
3.3	Conclusion	60
4	Implémentation et résultats	61
4.1	Introduction	61
4.2	Le logiciel Matlab	61
4.2.1	Utilité du Matlab	62
4.3	Choix du langage	62
4.3.1	Programation avec Matlab	63
4.4	Résultats et discussion	63
4.5	Conclusion	64
	Conclusion général	67

Table des figures

1.1	Ensemble convexe et non convexe	7
1.2	Illustration des points Nadir et Idéal	9
1.3	Pareto local	10
1.4	Théorème de contact	11
1.5	Surface de compromis	11
1.6	Illustration Front de pareto	12
1.7	Différents types de solutions	13
3.1	Présentation de l'ensemble de décision du MOILP	31
3.2	Les solutions efficaces du MOILP	32
3.3	Résolution graphique de problème relaxé	33
3.4	Résolution graphique de T_1	35
3.5	Ensemble de décision du MOILP	40
3.6	Résolution graphique par la méthode des cône (Chaabane)	41
3.7	Résolution graphique du problème relaxé (Chaabane)	42
3.8	Résolution graphique du MOILP	54
3.9	Solution efficace du MOILP	54
3.10	Résolution graphique du problème relaxé	55
3.11	Arborécence des nœuds relatif à l'exemple	57
3.12	Arborécence des nœuds relatif à l'exemple de Jorge	60

Liste des Algorithmes

1	Algorithme de Klein et Hannan	21
2	Algorithme de Sylva et Crema	23
3	Algorithme de Jorge	29
4	Algorithme de Chaabane et Pirlot	38
5	Algorithme de Chergui et Ouail	52

Remerciment

Nous remercions Dieu tout puissant clément et miséricordieux de nous avoir soigné et aidé.

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

*Nous remercions les membres de jury qui ont accepté de juger ce travail et d'y apporter leur caution : monsieur **Bezoui Madani** et madame **Khodja Karima**.*

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

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

Dédicas

Je remercie le Dieu de m'avoir donné la force d'accomplir ce travail pour aller plus loin.

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

Je le dédie à mon frère, mes soeurs et mes cousines, je les remercie pour leurs encouragements et leurs aides ainsi que toute ma grande famille.

*À l'homme qui m'a choisi pour me donner son nom à jamais, mon fiancé, **Ibrahim**.*

*À ma binôme **Oumayma**.*

À tous mes amis sans citer les noms, merci boucoups pour votre encouragement.

À tous ceux qui m'aiment et ceux que j'aime.

Saadia

Dédicas

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

Je dédie ce travail à :

*MA TRÈS CHÈRE MÈRE : **CHABANI Fatiha** avec tous mes sentiments de respect, d'amour, de gratitude et de reconnaissance pour tous les sacrifices déployés pour m'élever dignement et assurer mon éducation dans les meilleures conditions ,à l'âme de chère papa .*

*Mon chère frère **Mouhamed** , ça jolie femme et ça petite adorable **Aicha**.*

*Tous membre de la famille **CHAID** et **CHABANI**.*

Tous mes amis sans citer leurs noms.

*Ma binôme **Saadia**.*

Tous les étudiants de la recherche opérationnelle de l'université UMBB qui sont en chaumage.

Peuple algérien qui se bat pour sa liberté.

Tous qui aime oumayma et tous que oumayma aime.

Oumayma

Introduction générale

La Recherche Opérationnelle a porté l'appellation **Operationnal research** (en Anglais) et **Operation research** (en Américain). Le mot **operation** est à comprendre dans le sens d'activité militaire. Son application était donc avant tout militaire.

La recherche opérationnelle est née pendant la Seconde Guerre mondiale des efforts conjugués d'éminents mathématiciens (dont Neumann, Dantzig, Blackett) à qui ils leur était demandé de fournir des techniques d'optimisation des ressources militaires. Le premier succès de cette approche a été obtenue en 1940 par Patrick Blackett qui résolut un problème d'implantation optimale de radars de surveillance. La dénomination est restée par la suite, même si le domaine militaire n'est plus le principal champ d'application de cette discipline, le mot **opérationnelle** prenant alors plutôt le sens d'effectif.

Depuis une dizaine d'années le champ d'application de la Recherche opérationnelle s'est élargi à des domaines comme l'économie, la finance, le marketing et la planification d'entreprise. Plus récemment, la Recherche opérationnelle a été utilisée pour la gestion des systèmes de santé et d'éducation.

Des problèmes dans d'autres domaines d'application se sont également multipliés, citons par exemple les domaines de :

- La programmation linéaire classique ou programmation linéaire en nombres entiers.
- La théorie des graphes (plus courts chemins, flots, arbres, colorations, recouvrements...).
- Les modèles probabilistes (chaînes de Markov, phénomènes aléatoires, files d'attente...).

L'une des principales missions pour laquelle la recherche opérationnelle s'est vouée est d'aider à la prise de décision et à la gestion.

Les modèles traditionnels développés dans le cadre des méthodes quantitatives de gestion considéraient en général un critère unique, pour lesquels il existe des solutions optimales. Les algorithmes mis au point consistent alors à définir un moyen d'atteindre, le plus rapidement possible, une telle solution. Cependant, dans de nombreux cas, cette modélisation des problèmes ne traduit pas exactement la réalité à appréhender.

De nombreux problèmes du monde réel impliquent des objectifs multiples. En raison du conflit entre les objectifs, trouver une solution réalisable qui optimise simultanément tous les objectifs est généralement impossible, ainsi, le problème revient à générer un ensemble de solutions de bon compromis, c'est-à-dire des solutions dans lesquelles il est

impossible d'améliorer la valeur d'un objectif sans dégrader la valeur d'au moins un autre objectif, on parle dans ce cas de problème d'optimisation multi-objectif (ou multicritère) qui définit un ensemble de solutions acceptables assurant un compromis entre les objectifs considérés, appelées solutions de compromis ou solutions efficaces.

Dans plusieurs situations pratiques, il est nécessaire d'utiliser des variables discrètes dans la modélisation des problèmes. En effet, dans la majorité des applications concrètes de programmation mathématique, la présence des variables entière est inévitable, on parle alors d'optimisation discrète ou programmation en nombres entiers. De ce fait, la programmation mathématique multi-objectif en nombre entiers représente le cadre général de notre travail dans ce mémoire.

En outre, certains chercheurs soutiennent (voir par exemple Jorge (2009)[28]) que présenter de nombreuses solutions efficaces à un décideur peut, plus que tout, confondre la décision et peut rendre presque impossible la sélection d'une solution privilégiée. Une approche qui atténue ce problème est de trouver la solution la plus souhaitable parmi les solutions efficaces (de préférence sans les énumérer toutes). Cette approche est connue sous le nom d'optimisation sur l'ensemble efficient. De nombreux chercheurs ont abordé le problème d'optimisation d'une fonction linéaire sur l'ensemble efficient d'un problème multi-objectif [41][4][5], et la plupart de ces travaux traite le cas de l'optimisation d'une fonction linéaire par rapport à l'ensemble efficace d'un problème multi-objectif linéaire en nombre entiers.

La première méthode pour l'optimisation d'un critère linéaire sur l'ensemble efficace discret fut proposée en 1992 par Nguyen [33], cette méthode consiste à calculer une borne supérieure de la valeur optimale de la fonction objectif. En 2006, un algorithme évitant l'énumération explicite de tous les points efficaces dans l'espace des variables de décision est proposé par Abbas et al. [3], où différents types de coupes sont imposées de telle manière que l'amélioration de la valeur optimale de la fonction objectif soit garantie à chaque itération. En 2008, Jorge [28] développe un algorithme basé sur l'analyse d'un ordre approprié de problèmes linéaires en nombres entiers pour éliminer successivement les solutions moins bonnes sur le critère principal. En 2010, Chaabane et Pirlot [9] proposèrent une méthode de résolution dans l'espace des critères dans laquelle la valeur de la fonction objectif principal est améliorée en optimisant une somme pondérée des critères à chaque itération.

Récemment, Chaabane et Merbek en 2012[10] présentent des variations mineures de l'algorithme de Jorge. L'algorithme diffère principalement par la technique utilisée pour déterminer si une solution est efficace (test d'efficacité). Malheureusement, dans ces deux dernières méthodes, aucun calcul la comparaison avec l'algorithme de Jorge n'est incluse. Par conséquent, il n'est pas possible de déterminer si les changements entraînent à l'amélioration des performances.

L'objectif de ce mémoire consiste à étudier l'optimisation d'un critère linéaire sur l'ensemble efficient d'un problème multi-objectif en variables entières. On présente quelques

méthode existantes dans la littérature et on propose une nouvelle méthode pour résoudre ce type de problème.

Ce mémoire de master est organisé de la façon suivante :

- ✠ Chapitre 1 dresse l'état de l'art d'optimisation multi-objectif , nous nous sommes focalisé plus particulièrement sur l'optimisation linéaire en nombres entiers. Nous présentons aussi de principales approches de résolution proposées dans la littérature.
- ✠ Chapitre 2 se focalise sur la programmation linéaire multi-objectif en variables entière MOILP et sur quelques travaux récents existant dans la littérature.
- ✠ Chapitre 3 aborde la problématique de l'optimisation d'un critère sur l'ensemble efficient d'un problème multi-objectif linéaire discret et nous détaillons quelques méthodes de résolution pour le cas d'un critère linéaire.
- ✠ Chapitre 4 dresse une contribution qui consiste à optimiser un critère linéaire sur l'espace de critère d'un problème multi-objectif (MOILP).
- ✠ Finalement, nous clôturons ce mémoire par une conclusion générale et quelques perspectives de recherche.

Notation

- E : l'ensemble des solutions non dominés du problème (P).
- $S = \{x \in \mathbb{R}^n / Ax \leq b\}$.
- R^n : l'ensemble des vecteurs colonne d'ordre n .
- $R_+^n : \{x \in R^n; x \geq 0\}$.
- $R_{++}^n : \{x \in R^n; x > 0\}$.
- $Z_S = \{z_j(x)/x_i \in S, i = \overline{1, n}, j = \overline{1, p}\}$.
- Z_p : La valeur objectif du $p^{\text{ème}}$ critère.
- Eff : L'ensemble des solutions efficaces.
- $D = S \cap \mathbb{Z}$: L'ensemble des solutions entières.
- MOP : Porblème d'optimisation multi-objectif.
- $MOILP$: Problème d'optimisation multi-objectifs en nombres entiers.

1

Etat de l'art sur l'optimisation multi-objectifs

1.1 Introduction

Dans un problème mono-objectif, il existe une seule fonction objectif bien définie à optimiser (maximiser ou minimiser) afin d'aboutir à une solution optimale. Cependant, dans certains cas, le recours à une seule fonction objectif ne permet pas d'appréhender l'ensemble des points de vue ou aptitude d'un système, ce qui peut conduire à étendre le problème d'optimisation au cadre multi-objectif.

En effet, dans la plupart des problèmes d'optimisation plusieurs objectifs doivent être optimisés simultanément pour aboutir à une solution satisfaisante.

1.2 Définition

L'optimisation multi-objectif (appelée aussi optimisation multi-critère) est une branche de l'optimisation mathématique traitant spécifiquement des problèmes d'optimisation ayant plusieurs fonctions objectifs.

Les premiers travaux menés sur les problèmes multi-objectifs furent réalisés au 19^{me} siècle sur des études en économie par Edgeworth et généralisés par Pareto.

Un problème d'optimisation avec objectifs multiples (*MOP*) peut être représenté par le programme suivant :

$$(MOP) \begin{cases} \text{maximiser} & Z(x) = (z_1(x), z_2(x), \dots, z_p(x)) \\ \text{t.q} & Ax \leq b \\ & x \in R^n \text{ et } p \geq 2 \end{cases}$$

- A : Matrice d'ordre $n \times m$.
- b : Vecteur d'ordre m.
- p : Représente le nombre de critère.

1.3 Concepts de base et définitions

Définition 1.3.1 Cône

Soit $V \subset \mathbb{R}^n$ $V \neq \emptyset$. V un cône si $\alpha v \in V$, pour tous scalaire $\alpha > 0$ et tous $v \in V$.

Définition 1.3.2 Cône positif

Un cône positif est défini dans $\mathbb{R}^p$ par $C^+ = \{x/z(x) \in \mathbb{R}^p \text{ et } z(x) \geq 0\}$.

1.3.1 Convexité

Définition 1.3.3 Ensemble convexe

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

Définition 1.3.4 Fonction convexe

Soit Z une fonction de $\mathbb{R}$ dans $\mathbb{R}$, définie sur un intervalle I contenant au moins deux points. On dit que Z est convexe sur I si et seulement si :

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

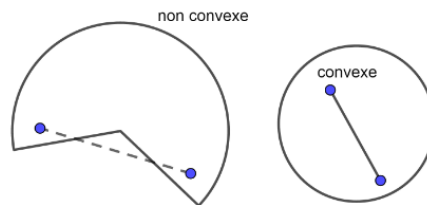


FIGURE 1.1: Ensemble convexe et non convexe

Définition 1.3.5 *Enveloppe convexe*

L'enveloppe convexe d'un sous-ensemble S de $\mathbb{R}^n$ est l'intersection de tous les sous-ensembles convexes de $\mathbb{R}^n$ contenant S . L'enveloppe convexe de S est par conséquent le plus petit sous-ensemble convexe de $\mathbb{R}^n$ qui contient S .

1.3.2 Points particuliers**Définition 1.3.6** *Point idéal*

Le point idéal est le vecteur défini par :

$$I = (\max_{x \in S} Z_1(x), \dots, \max_{x \in S} Z_p(x)) \in \mathbb{R}^p$$

NB : le point idéal en général $I \notin S$.

Définition 1.3.7 *Point anti-idéal*

Le point anti-idéal est le vecteur défini par :

$$A = (\min_{x \in S} Z_1(x), \dots, \min_{x \in S} Z_p(x)) \in \mathbb{R}^n$$

NB : le point anti-idéal en général $A \notin S$.

Définition 1.3.8 *Point nadir*

Le point nadir est le point de coordonnées :

$$n_i = \min_{j=1, \dots, p} Z_{ij} \quad , i = \overline{1, p}$$

Z_{ij} appartient représente la matrice des gains suivante :

$$\begin{pmatrix} \overline{Z_1} & Z_{12} & \dots & Z_{1p} \\ Z_{21} & \overline{Z_2} & \dots & Z_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ Z_{p1} & Z_{p2} & \dots & \overline{Z_{pp}} \end{pmatrix}$$

Où $\overline{Z_i} = \max_{x \in S} Z_i(x)$

Les coordonnées de ce point correspondent aux pires valeurs efficaces obtenues par chaque fonction objectif, lorsque l'on restreint l'espace des solutions à la surface de compromis.

Remarque 1.3.1 1. Le point idéal est utilisé dans plusieurs méthodes d'optimisation comme point de référence.

2. Le point nadir, lui, sert à restreindre l'espace de recherche ; il est utilisé dans certaines méthodes d'optimisation interactives.

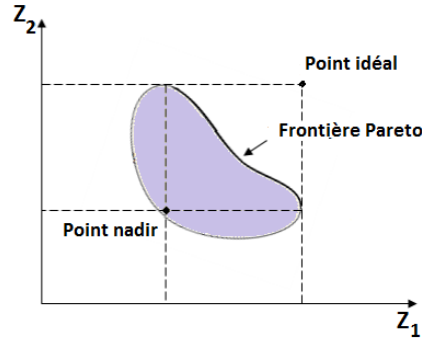


FIGURE 1.2: Illustration des points Nadir et Idéal

1.4 Dominance et optimalité

La solution d'un problème multi-objectif est une multitude de points de $\mathbb{R}^n$, il est vital pour identifier ces meilleurs compromis de définir une relation d'ordre entre ces éléments. Dans le cas des problèmes d'optimisation multi-objectifs, ces relations d'ordre sont appelées relations de dominance. Plusieurs relations de dominance ont déjà été présentées : la A- dominance , la dominance au sens de Geoffrion [24], la cône-dominance , . . . Mais la plus célèbre et la plus utilisée c'est la dominance au sens de Pareto.

Définition 1.4.1 Soient $Z, Z' \in Z_s$, on dit que Z domine Z' si :

$$Z_i \geq Z'_i \quad \forall i = \overline{1, p} \quad \text{et} \quad \exists i \in \{1, \dots, p\} \quad \text{tel que} \quad Z_i > Z'_i$$

et on note $Z \succ Z'$. Pour chaque paire de solution Z et Z' y il existe une des relation suivantes :

- Z domine Z' .
- Z' est dominée par Z .
- Z et Z' sont équivalentes au sens de dominance (solution de pareto équivalentes).

Définition 1.4.2 Dominance forte

Soient deux vecteurs critères $Z, Z' \in Z_s$. On dit que Z domine fortement Z' si et seulement si $Z \succ Z'$.

Si Z domine fortement Z' , alors Z est meilleur que Z' sur tous les critères.

Définition 1.4.3 Efficacité

Soit $x^* \in S$, on dit que x^* est efficace s'il $\nexists x \in S$ tel que :

$$Z(x) \text{ domine } Z(x^*).$$

Autrement dit : x^* est efficace si et seulement s'il $\nexists x \in S$ tel que :

$$Z_i(x) \geq Z_i(x^*) \quad \forall i = \overline{1, p}, \quad \exists i = \{1, p\} \quad Z_i(x) > Z_i(x^*), \quad \text{tel que} :$$

$$x \neq x^*$$

Définition 1.4.4 *Efficacité faible*

Soit $x^* \in S$, on dit que x^* est faiblement efficace s'il $\nexists x \in S$ tel que :
 $x \neq x^*$ et $Z(x) \geq Z(x^*)$.

Définition 1.4.5 *Efficacité forte*

Soit $x^* \in S$, on dit que x^* est fortement efficace s'il $\nexists x \in S$ tel que :
 $x \neq x^*$ et $Z(x) > Z(x^*)$

Remarque 1.4.1

1. Un point est efficace si son image par Z est un vecteur critère non dominé. Le terme efficacité est aussi connu sous Pareto optimalité ou non infériorité.
2. À partir d'un point efficace, il est impossible d'augmenter la valeur d'un des critères sans diminuer la valeur d'au moins un autre critère.
3. Une solution est faiblement efficace si son vecteur critère n'est pas fortement dominé.
4. Une solution x est fortement efficace s'il n'existe pas une autre solution telle que le vecteur critère, qui lui est associé, soit aussi bon que celui de x .

Optimalité locale au sens de Pareto

Un vecteur $x \in \mathbb{R}^n$ est optimal localement au sens de Pareto s'il existe un réel $\delta > 0$ tel qu'il n'y ait pas de vecteur x' qui domine le vecteur x avec $x \in \mathbb{R}^n \cap \mathbf{B}(x, \delta)$, où $\mathbf{B}(x, \delta)$ représente une boule de centre x et de rayon δ .

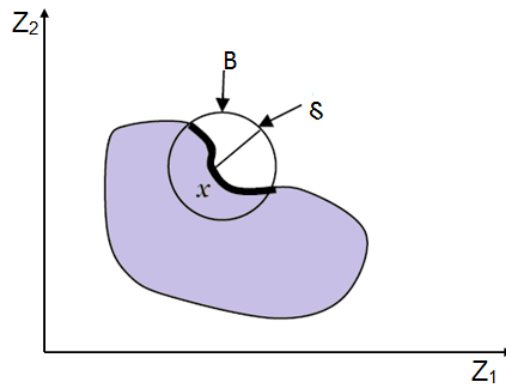


FIGURE 1.3: Pareto local

Optimalité globale au sens de Pareto

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

Remarque 1.4.2 On dit aussi qu'un vecteur x est optimal au sens de Pareto pour un problème d'optimisation multi-objectifs donné si : $(c^- + x) \cap S = x \Rightarrow$ Le théorème de contact. Où S désigne l'espace de solutions réalisables.

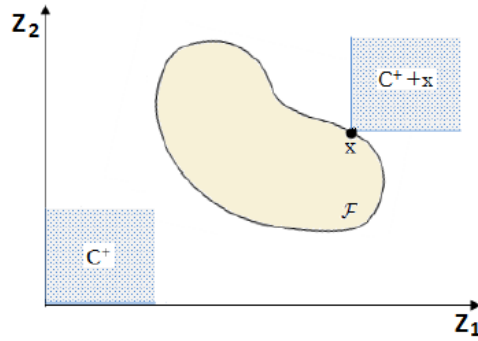


FIGURE 1.4: Théorème de contact

1.4.1 Surface de compromis

Les solutions obtenues lors du classement basé sur le principe de dominance de Pareto forme ce que l'on appelle la surface de compromis (ou Pareto Front).

Soit un problème à deux fonctions objectifs Z_1 et Z_2 à minimiser sous les contraintes $g(x) \leq 0$ et $h(x) = 0$.

- On appelle S l'ensemble de valeurs du couple $(Z_1(x), Z_2(x))$ quand x respecte les contraintes $g(x)$ et $h(x)$.
- On appelle P la surface de compromis.

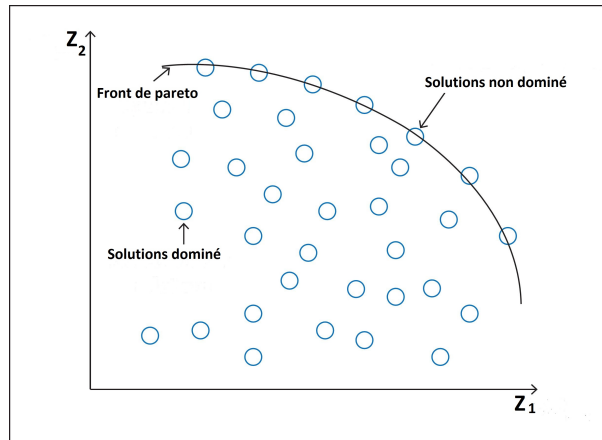


FIGURE 1.5: Surface de compromis

Les solutions optimales au sens de pareto où solutions non dominé, forment le front de pareto.

1.4.2 Front de pareto

Définition 1.4.6 *Le front de pareto est l'ensemble de tous les points efficace.*

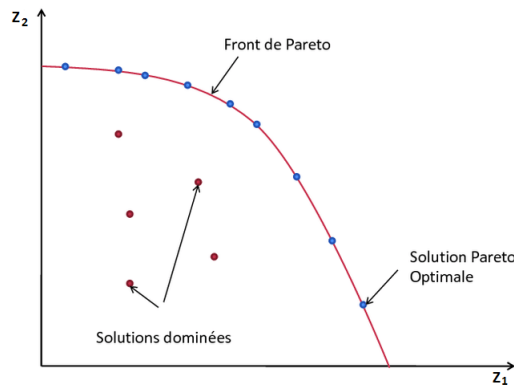


FIGURE 1.6: Illustration Front de pareto

1.4.3 Structure de front de pareto

L'objectif des problèmes d'optimisations multi-objectifs est de fournir aux décideurs un ensemble (le plus complet possible) de solutions Pareto, afin qu'ils puissent ensuite choisir les solutions qui les intéressent le plus.

Une question se pose donc sur la nature de ces solutions Pareto et la nécessité de les obtenir toutes. Une étude de la frontière Pareto doit donc être réalisée.

Lorsque deux solutions ont exactement les mêmes valeurs pour l'ensemble des objectifs, elles sont équivalentes dans l'espace objectif, mais peuvent correspondre à deux solutions différentes dans l'espace décisionnel. Une question importante est de savoir s'il est intéressant de garder ces deux différentes solutions.

Front minimal

Lors de la résolution d'un problème comportant énormément de solutions Pareto, il est peut être préférable de privilégier une bonne approximation de l'ensemble de la frontière

et donc favoriser la diversité (du côté objectif) des solutions retenues.

Front maximal complet

Au contraire, lorsque la frontière Pareto comporte peu de solutions, afin d'avoir une bonne représentation de l'ensemble des solutions non dominées, il sera intéressant de rechercher les solutions de même valeur.

Solutions supportées / non supportées

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 (voir figure 1.7) et peuvent donc être trouvées à l'aide d'une agrégation linéaire des objectifs [24].

Elles sont donc plus simples à obtenir que les solutions non supportées qui elles, sont plus difficile à trouver.

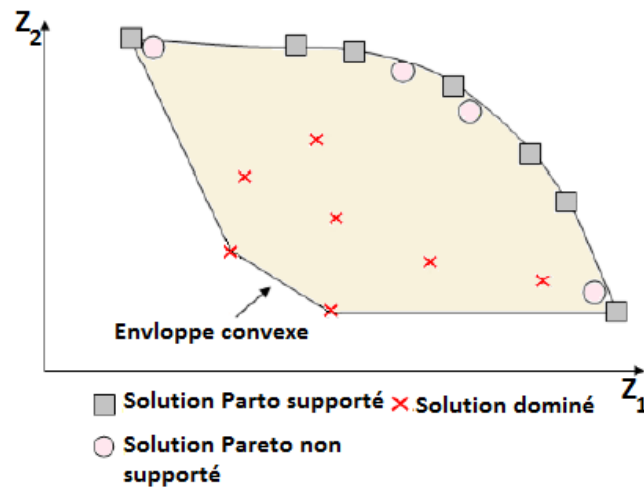


FIGURE 1.7: Différents types de solutions

1.4.4 Dominance de Geoffrion

La dominance au sens de **Geoffrion**[24][22] [32] est très importante dans le monde d'optimisation multi-objectif les solutions obtenu sont appelées solutions pareto optimales propre.

Définition 1.4.7 *Dominance au sens de Geoffrion [24] une solution $x^* \in S$ est dite de pareto optimale propre si :*

- x^* est pareto optimale.
- Il existe un nombre $M > 0 \forall i$ et $\forall x \in S$ vérifiant $Z_i(x) > Z_i(x^*)$, il existe un indice j tel que :

$$Z_j(x^*) > Z_j(x) \quad \text{et} \quad \frac{Z_i(x^*) - Z_i(x)}{Z_j(x) - Z_j(x^*)} \geq M \quad (1.1)$$

1.5 Choix de la méthode d'aide a la décision

Pour résoudre un problème multi-objectif, on a déterminer un ensemble de solution pareto, il est nécessaire de faire intervenir l'humain à travers un décideur, pour le choix final de la solution à garder.

Avant de commencer la résolution il faut choisir la méthode de résolution à utiliser. Ces méthodes peuvent être classer en trois familles [18] :

- 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. La solution rechercher est obtenu en une seule exécution, donc c'est une méthode rapide mais il faut prendre en considération la possibilité de non satisfaction de décideur de résultat trouvé donc on a à relancer la recherche avec un autre compromis.
- Les méthodes d'optimisation **progressives** :
dans ce cas, 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. au regard de l'ensemble des solutions, il peut sélectionner celle qui lui semble la plus appropriée. Ainsi, il n'est plus nécessaire de modéliser les préférences du décideur, 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.

1.6 Approches de résolution d'un problème multi-objectif (MOP)

Un grand nombre d'approches existent pour résoudre les problèmes multi-objectif. Certaines utilisent des connaissances du problème pour fixer des préférences sur les critères et ainsi contourner l'aspect multicritère du problème. D'autres mettent tous les critères au même niveau d'importance, mais là aussi il existe plusieurs façons de réaliser une telle opération. Des ouvrages ou articles de synthèse ont été rédigés, des états de l'art plus complets peuvent être consultés notamment dans (Ulungu et al.) [40], Miettinen [32], Ehrgott [22], (Ehrgott et al.) [21], Deb [17], (Collette et al.) [13]. On distingue deux approches de résolution d'un problème d'optimisation multi-objectif, peuvent être distinguées dans la littérature voir Ehrgott [22], (Roy et al.) [35] et Roy [36] :

- **Approche mono-objectif** : consiste à ramener le problème à un problème d'optimisation mono-objectif, au risque d'enlever toute signification au problème.
- **Approche multi-objectif** : consiste à proposer des solutions en tenant compte de l'ensemble des critères.

1.6.1 Approche mono-objectif

Les objectifs du problème d'optimisation sont transformés en une seule fonction objectif. Dans ce cas, le décideur est supposé quantifier a priori l'importance de chaque critère afin de construire une fonction unique.

Théorie de l'utilité multi-attributs

Si l'on interroge le décideur sur ses préférences, ses réponses seront en accord avec une certaine fonction U que l'on ne connaît pas. Le problème est donc d'essayer d'estimer cette fonction. En théorie de l'utilité, on définit une fonction de préférence comme suit :

Définition 1.6.1 *Étant donné un ordre de préférence noté " $\succ$ " sur l'ensemble Z_s , une fonction U à valeurs réelles vérifiant $U(Z_1) > U(Z_2)$, $Z_1 \succ Z_2$; est appelée fonction de préférence ou d'utilité.*

Définition 1.6.2 *La fonction $Z_i : \mathbb{R}^n \rightarrow \mathbb{R}$ est dite fortement (strongly) décroissante si :*

$$\forall x^1, x^2 \in \mathbb{R}^n ; \text{ si } x_j^1 \leq x_j^2 ; \forall j \in \{1, \dots, n\} \text{ et } \exists l \in \{1, \dots, n\} / x_l^1 < x_l^2 \text{ alors } Z_i(x^1) > Z_i(x^2).$$

Théorème 1.6.1 *Soit $U : \mathbb{R}^k \rightarrow \mathbb{R}$ une fonction d'utilité associée au problème (P) et supposons que U est fortement décroissante. Si x° est une solution du problème $\max_{x \in E} U(Z(x))$ alors x° est une solution Pareto optimale du problème (P) .*

Les modèles les plus couramment utilisés pour la fonction d'utilité U sont *le modèle additif* et *le modèle multiplicatif* :

$$U(Z(x)) = \sum_{i=1}^k U_i(Z_i(x)) \text{ et } U(Z(x)) = \prod_{i=1}^k U_i(Z_i(x))$$

où les fonctions U_i sont strictement croissantes et à valeurs réelles. Elles servent uniquement à transformer les critères initiaux Z_i de manière à ce qu'ils s'expriment tous suivant la même échelle.

Méthode des sommes pondérées

Cette méthode de résolution d'un problème d'optimisation multi-objectif est la plus évidente et largement utilisée en pratique. Il s'agit d'associer à chaque fonction objectif un coefficient de pondération et à faire la somme des fonctions objectifs pondérées pour obtenir une nouvelle et unique fonction objectif.

Le problème multicritère (P) se transforme alors de la manière suivante :

$$\begin{cases} \text{minimiser } \sum_{i=1}^p \lambda_i Z_i(x) \\ \text{s.c} \\ x \in E \text{ et } \lambda_i \in \Lambda = \{\lambda_i : \sum_{i=1}^k \lambda_i = 1 \text{ et } \lambda_i \geq 0, \forall i = \overline{1, p}\} \end{cases} .$$

Cette méthode est très efficace du point de vue algorithmique, mais son principal inconvénient est qu'elle ne permet pas de trouver les solutions enfermées dans des concavités.

Méthode dite "but programmé ou Goal programming"

Cette méthode, qui relève d'une théorie très avancée dans le domaine des problèmes multicritère, a été initialement conçue par (Charnes et Cooper) [11] dans le cas linéaire. L'idée générale de la méthode est d'établir un but à atteindre pour chaque critère. Généralement, le point qui satisfait tous les buts n'est pas réalisable, la solution préférée serait donc celle qui se rapproche le plus possible de ces buts.

Soit $Z^* = (Z_1^*; \dots; Z_k^*)$ le vecteur de référence ou but fixé par le décideur relativement à tous les critères, le problème revient à considérer la relation suivante :

$$\begin{cases} \text{minimiser } \left(\sum_{i=1}^k |Z_i(x) - Z_i^*|^p \right)^{1/p} \\ \text{s.c} \\ x \in E \end{cases}$$

où $p \geq 1$, et Z_i^* est le vecteur de référence (but) ou le vecteur idéal.

La méthodologie du Goal programming repose sur les points suivants :

- fixer les valeurs des cibles Z_i^* que l'on désire atteindre sur chaque critère .

- définir des déviations positives d_i^+ et négatives d_i^- relativement à ces buts.
- minimiser la somme pondérée de ces déviations relativement à une norme $\|\cdot\|_p$ définie dans $\mathbb{R}^n$.

La formulation mathématique du Goal programming est la suivante :

$$\begin{cases} \min_{x \in E} \left(\sum_{i=1}^k (d_i^+ + d_i^-)^p \right)^{1/p} \\ s.c \\ Z_i(x) + d_i^- - d_i^+ = Z_i^* & ; \\ d_i^- \geq 0, d_i^+ \geq 0, i \in \{1, \dots, k\} & . \end{cases}$$

d_i^+ et d_i^- sont respectivement appelés la sur-réalisation et la sous-réalisation du i^{me} critère.

La méthode ε – contrainte

Dans cette méthode, on n'optimise qu'un seul objectif jugé important par le décideur, les autres sont transformés en contraintes d'inégalités par rapport à un vecteur seuil ε . Le principe de la méthode ε –contrainte ou méthode de compromis est intéressant lorsque l'on cherche à énumérer toutes les solutions d'un front Pareto. L'inconvénient principal de cette méthode est qu'elle nécessite une résolution mono-objectif pour chacune des solutions du front. Lorsque ce nombre est élevé, cela peut être vu comme une limite, d'autant plus lorsque la méthode de résolution mono-objectif est coûteuse.

1.6.2 Approche multi-objectif

Approche a posteriorie

L'expression des préférences et la pondération des objectifs dans ces approches se fait sur la base de détermine un ensemble de solutions réalisable et la sélection de la meilleure solution ne se fait qu'à la fin processus.

La méthode hybride

Il est souvent possible d'exploiter plusieurs méthodes pour en former une nouvelle. Dans ce cas, nous obtenons une "**méthode hybride**".

La méthode de Corley en 1980[14] est la plus connue. On utilisant la méthode de pondération des fonctions objectifs et la méthode du compromis. On part du problème multi-objectif (P) et on le transforme de la manière suivante :

$$\left\{ \begin{array}{l} \text{minimiser } \left(\sum_{i=1}^k \lambda_i * Z_i(x) \right) \\ \text{s.c} \\ Z_j(x) \leq \varepsilon_j \quad \text{pour } j \in \{1, \dots, k\}; \\ x \in E. \end{array} \right.$$

En faisant varier le vecteur ε dans ce problème paramétrique, on génère ainsi l'ensemble de toutes les solutions Pareto optimales.

1.7 Conclusion

Dans ce chapitre, nous avons essayé de rassembler un état de l'art sur les problèmes d'optimisation multi-objectifs, tout en montrant la manière de définir un problème pareil, ses contraintes, ses objectifs ainsi que les méthodes utilisées pour le résoudre tout en respectant le concept de Compromis et les frontières de Pareto. Nous avons parlé sur les méthodes utilisées dans le domaine de l'optimisation multi-objectifs qui sont capables d'offrir un ensemble de solutions dites optimales pour un seul problème en établissant un compromis entre les différents critères. Puis une classification de ces méthodes de résolution a été introduite tout en essayant de présenter plusieurs approches utilisées pour aborder les problèmes de l'optimisation multi-objectif.

Dans ce qui suit, on va cerner le cadre de notre étude à les problèmes d'optimisation multi-objectif dans le cas linéaire et à variables discrète.

2

La Programmation multi-objectif linéaire en nombres entiers

2.1 Introduction

La programmation linéaire est l'une des plus importantes techniques d'optimisation utilisées en recherche opérationnelle. Ses développements théoriques ont été suggérés et accélérés par un grand nombre d'applications pratiques, dans le domaine de l'économie, de la gestion et autres.

En pratique, très souvent, la présence de variables discrètes (ou entières) est inévitable dans la modélisation en optimisation et ces variables modifient considérablement la structure mathématique du problème. Il en résulte que des méthodes spécifiques à cette situation sont nécessaires. Ce chapitre a pour objectif principal de présenter le contexte de l'optimisation linéaire multi-objectif discrète et de présenter quelques méthodes exactes permettant de caractériser totalement ou partiellement l'ensemble de solutions efficaces du problème.

2.2 Définition

Un problème d'optimisation est un problème mathématique où, étant donné un ensemble de variables et des contraintes sur ces variables, on doit trouver une assignation qui maximise (ou minimise) une certaine fonction de coût. On parle de problème linéaire en

nombres entiers lorsque les contraintes et la fonction de coût sont combinaisons linéaires des variables et le problème est à nombres entiers si ces variables ne sont autorisées qu'à prendre des valeurs dans l'ensemble des entiers.

La contrainte qui force les variables à prendre des valeurs entières est appelée contrainte d'intégralité. Lorsque l'on gomme cette contrainte, on parle d'un problème relaxé ou de relaxation continue, et l'on a alors affaire à un problème d'optimisation linéaire continu. Le rapport entre l'optimal dans la version relaxée et dans la version entière est souvent appelé *integrality gap*.

2.3 Formulation d'un problème de multi-objectif en nombres entiers *MOILP*

Un problème de programmation linéaire multi-objectif (*MOILP*) est un problème d'optimisation consistant à maximiser (ou minimiser) des fonctions linéaires dite fonctions objectifs sous contraintes linéaires exprimées sous forme d'équations ou d'inéquations dans lequel les variables sont entières.

Un problème *MOILP* est défini comme suit :

$$(MOILP) \left\{ \begin{array}{ll} \text{optimiser} & Z_i(x) = c^i x \quad . \\ s.c & \\ Ax \leq b & ; \\ x \in D, i = \overline{1, p} & . \end{array} \right.$$

avec :

- n : nombre de variables ;
- m : nombre de contraintes ;
- A : matrice réelle des contraintes, d'ordre $n \times m$;
- $c^i = (c_1^i, c_2^i, \dots, c_n^i)$: vecteur coûts (profits) d'ordre n du $i^{\text{ème}}$ fonction objectif ;
- b : vecteur colonne d'ordre m des seconds membres ;
- $x = (x_1, x_2, \dots, x_n)$: vecteur de variables de décision d'ordre n .
- $D = S \cap \mathbb{R}^p$ est l'ensemble des solutions entière du problème *MOILP* .

2.4 Méthode de résolution d'un problème MOILP

Il existe plusieurs méthodes exactes de résolution des problèmes de programmation linéaire multi-objectif en nombres entiers ([26] [39] [1] [2]), qui permettent l'obtention d'au moins une solution efficace du problème à résoudre.

Nous présentons dans la section suivante quelques méthodes de résolution de ce type de problème.

2.4.1 Méthode de D.Klein et E.Hannan

La technique proposée par D.Klein et E.Hannan [29] 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 unicritère 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.

Algorithme 1 : Algorithme de Klein et Hannan

Étape 1 :

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

$$(P_1) : \max\{Z_k = c^k x, x \in D, k \in \{1, \dots, p\}\}$$

si Soit x_1 la solution optimale de (P_1) , est unique **alors**

x_1 efficace pour (P_1) .

sinon

déterminer toutes les solutions alternatives à x_1 et par comparaison deux à deux des vecteurs critères associés, garder uniquement celles qui sont efficaces pour construire l'ensemble $\text{Eff}(P_1)$ des solutions efficaces générées à l'étape 1.

Étape générale j :

A l'étape j , on résout le problème (P_j) qui est défini comme suit :

$$(P_j) \left\{ \begin{array}{l} \max \quad Z_i = c^i x ; \\ \text{s.c} \quad x \in D ; \\ \quad \bigcap_{l=1}^q (\bigcup_{i=1, i \neq s}^r c^i x \geq c^i y^l + f_i). \end{array} \right.$$

Avec $f_i \geq 1$ entier, y^l ($l = 1, \dots, q$) les points efficaces obtenus jusqu'à l'étape $j - 1$.

si $\text{Eff}(P_j)$ 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**

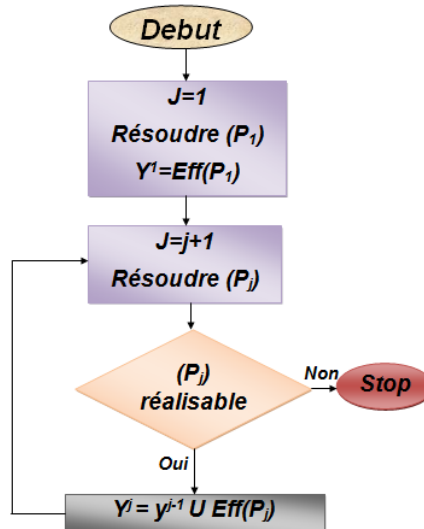
└ $Y^j = Y^{j-1} \cup \text{Eff}(Y^j)$ pour $j \geq 2$ avec $Y^1 = \text{Eff}(P_1)$

Étape finale n :

La procédure s'arrête lorsque le problème (P_n) est irréalisable.

Si à chaque étape j , $f_i = 1 ; \forall i = \overline{1, p} ; i \neq s$, la procédure trouve l'ensemble de toutes les solutions efficaces du problème (P) . Cependant, si pour certaines valeurs de i , $f_i > 1$ seulement un sous ensemble de solutions efficaces sera généré.

Organigramme de la méthode



2.4.2 Méthode A.Sylva et J.Crema

La méthode développée par Crema et Sylva [15] est une variante de celle de Klein et Hannan [29] étudiée précédemment. Son principe repose sur la résolution d'une succession de programmes linéaires en nombres entiers optimisant à chaque étape une combinaison positive des critères. Un ensemble de contraintes est rajouté à chaque fois assurant la détection d'une nouvelle solution efficace. A la fin, la méthode fournit l'ensemble de toutes les solutions non dominées du problème de programmation linéaire discrète à objectifs multiples.

Les auteurs considèrent que les données du problème *MOILP* sont entières, aussi bien la matrice des contraintes A , le vecteur second membre b que les coefficients c_j^i de tous les critères.

Algorithme 2 : Algorithme de Sylva et Crema**Étape 1**

Après avoir fixé le vecteur poids $(\lambda_1, \lambda_2, \dots, \lambda_r)$ à des valeurs strictement positives, la première étape consiste à résoudre le problème (P_1) défini comme suit :

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

si (P_1) , *n'admet pas solution* **alors**

alors (P) l'est aussi.

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 :

si (P_k) , *n'admet pas solution* **alors**

l'algorithme s'arrête.

sinon

└ une nouvelle solution efficace x_k est trouvée et le nouveau problème (P_{k+1}) est défini à partir de (P_k)

en éliminant toutes les solutions vérifiant $c^i x \leq c^i x^k, \forall i = \overline{1, p}$ ceci peut être traduit par le rajout de contraintes suivantes :

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

Où $-M_i$ est un minorant pour toute valeur réalisable de la i^{me} fonction objectif et $f_i \geq 1$ entier représente la plus petite augmentation possible du i^{me} critère.

Étape générale k :

À l'étape k , on résout le problème (P_k) qui est défini comme suit :

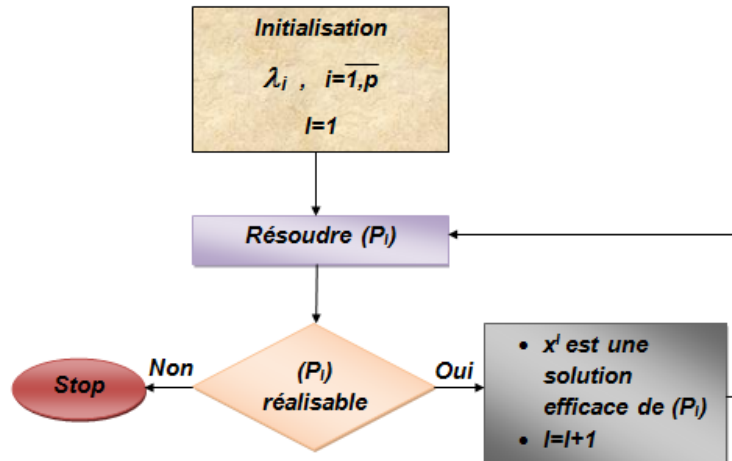
$$\begin{cases} \max \sum_{i=1}^p \lambda_i c^i x \\ \sum_{i=1}^p \lambda_i = 1 \\ s.c \\ x \in D; \\ c^i x \geq (c^i x^j - f_i) y_i^j - M_i(1 - y_i^j), & i = \overline{1, p}, \quad j = 1, \dots, k-1; \\ \sum_{i=1}^p y_i^j = 1, & y_i^j \geq 0, \quad i = \overline{1, p}, \quad j = 1, \dots, k-1; \\ y_i^j \in \{0, 1\} & i = \overline{1, p}, \quad j = 1, \dots, k-1. \end{cases}$$

Étape finale n :

La procédure s'arrête lorsque le problème (P_n) est irréalisable.

À l'étape n et une fois qu'on obtient toutes les solutions efficaces. En fait, à chaque étape une coupe est ajoutée d'une manière à optimiser au moins un des critères du coupe. Pour $f_i = 1, \forall i = \overline{1, p}$ la méthode génère toutes les solutions non dominées. Lorsque $f_i > 1$; elle prend une valeur permettant d'atteindre la valeur minimale souhaitée par le décideur pour le i^{me} critère. Dans ce cas, seul un sous-

Organigramme de la méthode :



2.5 Conclusion

Ce chapitre est consacré à l'introduction d'un problème multi-objectif en nombres entiers. Les définitions et les concepts de base sont présentés ainsi que les principaux résultats nécessaires à la résolution d'un problème MOILP. Nous avons décrit quelques méthodes exactes de résolution de *MOILP*.

3

L'optimisation d'un critère sur un ensemble efficient

3.1 Introduction

De nombreuses situations de prise de décision impliquent des critères multiples et non comparables. Souvent, dans de telles situations, chaque critère peut être modélisée comme une fonction objectif à optimiser sur un ensemble de solution réalisable. Le concept de points efficaces joue un rôle important dans l'analyse et la résolution de ces problèmes.

Dans certaines situations pratiques, l'énumération de tout l'ensemble des solutions efficaces d'un problème multi-objectif n'est pas toujours recommandée car il peut s'avérer que cet ensemble efficient soit très grand et il devient impossible pour le décideur de choisir le meilleur compromis en termes de ses préférences.

L'optimisation d'un critère, qui exprime les préférences du décideur, sur l'ensemble des solutions efficaces constitue, dès lors, un sujet de recherche essentiel dans ce domaine. Ce problème est en principe très difficile à résoudre, ceci est dû principalement à la non-convexité de son ensemble réalisable et le problème appartient alors à la classe des problèmes d'optimisation globale (programmation non convexe).

Dans ce mémoire, nous nous intéresserons au problème de l'optimisation d'une fonction linéaire sur l'ensemble des solutions efficaces d'un problème multi-objectif linéaire discret. Ce problème se pose chaque fois qu'une fonction linéaire est disponible qui sert de critère pour mesurer l'importance de ou pour discriminer parmi les alternatives efficaces

disponibles en optimisant cette fonction linéaire sur l'ensemble efficace.

Ce type de problème a été étudié la première fois en 1972 par Philip [34] et depuis plusieurs chercheurs, citons en particulier : Benson [4][5][6][7][8] ,Isermann [27] , Yamamoto [41] , Ecker et Song [20] , Sayin [37] , motivés par de nombreuses applications [4][23] se sont intéressés à l'optimisation d'une fonction sur l'ensemble efficient d'un problème multi-objectif linéaire.

En 1992 Nguyen [33] propose pour la première fois une méthode de résolution des problème d'optimisation d'un critère linéaire sur l'ensemble efficace discret, cette méthode consiste à calculer une borne supérieure de la valeur optimale de la fonction objectif. En 2006, (Abbas et al.)[3] proposent un algorithme évitant l'énumération explicite de tous les points efficaces dans l'espace des variables de décision, où différents types de coupes sont imposées de telle manière que l'amélioration de la valeur optimale de la fonction objectif soit garantie à chaque itération. En 2008, Jorge [28] développe un algorithme basé sur l'analyse d'un ordre approprié de problèmes linéaires en nombres entiers pour éliminer successivement les solutions moins bonnes sur le critère principal. Très récemment , En 2010, (Chaabane et al.) [9]proposèrent une méthode de résolution dans l'espace des critères dans laquelle la valeur de la fonction objectif principal est améliorée en optimisant une somme pondérée des critères à chaque itération.

3.2 Méthodes de résolution d'un problème d'optimisation multi-objectif sur l'ensemble efficace dans le cas discret

La programmation linéaire entière multi-objectif (MOILP) est une classe importante de problèmes survenant dans les multicritères prise de décision lorsque certaines ou toutes les variables du modèle représentent des décisions discrètes. Au cours des dernières décennies, plusieurs méthodes ont été développées résoudre les problèmes (MOILP), dans le sens de trouver l'ensemble efficace, ou du moins une partie importante de celui-ci. L'optimisation dite sur l'ensemble efficace est une approche intéressante dans l'optimisation multicritères qui a été soigneusement étudié dans le cas continu par de nombreux auteurs, bien que le cas discret na pas reçu autant d'attention.

3.2.1 Définitions et notations

Le problème d'optimisation d'une fonction linéaire sur l'ensemble efficient d'un problème multi-objectif linéaire peut se formuler comme suit :

$$(P_E) \begin{cases} \text{maximiser} & \phi(x) = dx \\ \text{s.c} & \\ & x \in \text{Eff.} \end{cases}$$

Où d est un vecteur de R^n et Eff représente l'ensemble des solutions efficace du problème multi-objectif linéaire discèt suivant :

$$(MOILP) \begin{cases} \text{maximiser} & Z_i(x) = c^i x \quad i = \overline{1, p} \\ \text{s.c} & \\ & x \in D. \end{cases}$$

Où

- $D = S \cap Z^n$.
- $S = \{x \in R^n / Ax \leq bx \geq 0\}$
- $A \in R^{m \times n}, b \in R^m$
- $c^i \in R^n, \forall i = \overline{1, p}$.

D étant supposé non vide et S un polyèdre convexe borné.

Considérons le problème relaxé (en relachant la contrainte d'efficacité) suivant :

$$(P_R) \begin{cases} \text{maximiser} & \phi(x) = dx \\ \text{s.c} & \\ & x \in D. \end{cases}$$

Une caractérisation des solutions efficaces du problème multi-objectif ($MOILP$) est donnée par le théorème suivant qui servira de test d'efficacité :

Théorème 3.2.1 [19]

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

$$(P(x^*)) \begin{cases} \text{optimiser} & \theta = \sum_{i=1}^{i=p} \psi_i \\ \text{s.c} & \\ & Cx - I\psi = Cx^*; \\ & x \in D, \psi \in R^+, i = \overline{1, p}. \end{cases}$$

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

Dans le cas où la valeur de $\theta \neq 0$, alors toute solution optimale $\hat{x}$ de $(P(x^*))$ est efficace pour (P_E) et son vecteur critère domine Cx^* .

3.2.2 Méthode de Jorge [28]

L'algorithme que nous proposons ici s'avère fournir une solution optimale du problème (P_E) sans avoir à calculer l'ensemble de toutes les solutions efficaces du problème multi-objectif *MOILP*.

3.2.3 Description de la méthode

La procédure commence à résoudre le problème relaxé (P_R) . Évidemment, seulement dans un nombre réduit de cas spéciaux la solution optimale de (P_R) fournit une solution optimale de (P_E) . Donc, si ce n'était pas le cas, une nouvelle solution efficace qui domine la précédente est alors obtenue. Ensuite, dans chaque itération, le critère principal est optimisé sur le domaine restreint par des contraintes en nombres entiers qui sont incluses progressivement pour éliminer les solutions dominées par la solution efficace courante, afin de fournir une solution non dominée par les solutions détectées antérieurement jusqu'à ce qu'une solution optimale soit finalement trouvée.

3.2.4 Formulation de l'algorithme

Algorithme 3 : Algorithme de Jorge

Etape (0) :(Initialisation)

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

si (P_R) est irréalisable **alors**

(P_E) est aussi irréalisable **stop**.

sinon

soit x^l une solution optimal de (P_R) .

Etape (1) :

si x^l est efficace **alors**

$x_{opt} = x^l$ et une solution optimale de (P_E) et $\phi_{opt} = dx^l$ **stop**.

sinon

poser $\phi_{\sup} = dx^l$ et aller à **Etape (2)**

Etape (2) :

Trouver $\tilde{x}^l \in E(P)$ dont le vecteur critère domine Cx^l et soit $\tilde{x}^l$ une solution optimal du problème (T_l) suivant :

$$\max\{dx/Cx = C\tilde{x}^l, x \in D\}$$

si $d\tilde{x}^l > \phi_{\inf}$ **alors**

poser $\phi_{\inf} = d\tilde{x}^l$ et $x_{opt} = \tilde{x}^l$.

si $\phi_{\inf} = \phi_{\sup}$ **alors**

x_{opt} est une solution optimal de (P_E) **stop**.

Cette étape permet de déterminer toutes les solutions qui ont le même vecteur critère que $\tilde{x}^l$ en résolvant (T_l) .

Etape (3) :

Résoudre le problème (P_l) suivant :

$$\max\{dx/x \in D - \cup_{s=1}^l D_s\}$$

où $D_s = \{x \in \mathbb{Z}^n / C\tilde{x}^s \geq Cx\}$.

si (P_l) est irréalisable **alors**

x_{opt} est une solution optimal de (P_E) **stop**.

sinon

soit x^{l+1} une solution optimal de (P_l)

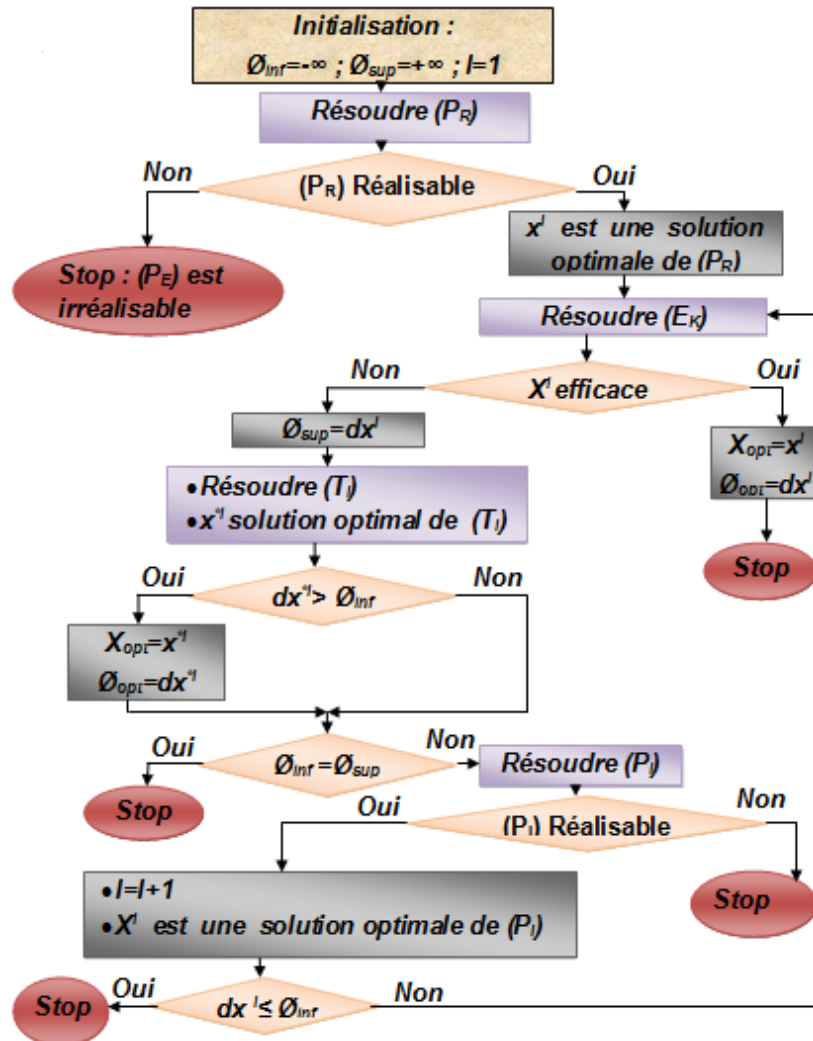
si $dx^{l+1} \leq \phi_{\inf}$ **alors**

x_{opt} est une solution optimal de (P_E) **stop**.

sinon

poser $l = l + 1$ et aller à l'étape (1)

Organigramme de la méthode



3.2.5 Exemple

Considérons le problème suivant [28] :

$$(P_E) \left\{ \begin{array}{ll} \text{maximiser} & \phi(x) = -x_1 - 2x_2 \\ \text{s.c} & \\ & -2x_1 + x_2 \leq 0 \quad ; \\ & x_1 \leq 3 \quad ; \\ & x_2 \leq 2 \quad ; \\ & x \in Eff \quad . \end{array} \right.$$

avec :

$$(MOILP) \left\{ \begin{array}{ll} \max Z_1 = x_1 - 2x_2 \\ \max Z_2 = -x_1 + 4x_2 \\ \text{s.c} & \\ & -2x_1 + x_2 \leq 0 \quad ; \\ & x_1 \leq 3 \quad ; \\ & x_2 \leq 2 \quad ; \\ & x \in Z_+ \quad . \end{array} \right.$$

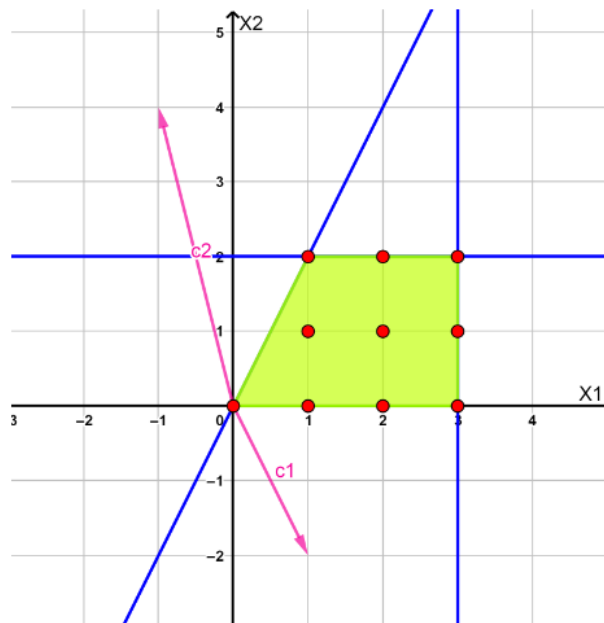


FIGURE 3.1: Présentation de l'ensemble de décision du MOILP

Les solutions réalisables du problème *MOILP* représenté dans la figure(3.1) sont :

$$D = \{(0, 0), (0, 1), (0, 2), (0, 3), (1, 1), (1, 2), (2, 1), (2, 2), (3, 1), (3, 2)\} .$$

Les solutions efficaces représentées dans la figure (3.2) et obtenues par la méthode du cône positif sont : $Eff = \{(2, 0); (3, 0); (2, 1); (3, 1); (1, 2); (2, 2); (3, 2)\}$

On effet, on remarque que les solutions $(0, 0); (1, 0); (1, 1)$ sont dominées par $(2, 1); (3, 1); (3, 2)$ respectivement, voir figure(3.2)

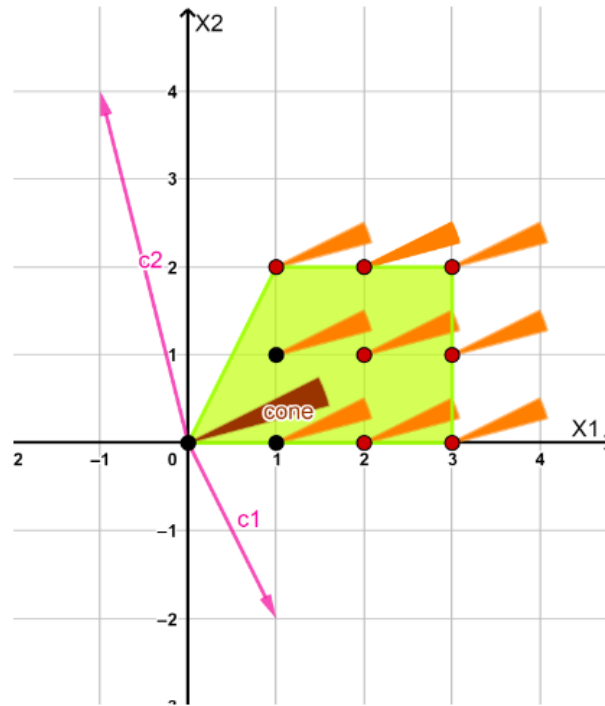


FIGURE 3.2: Les solutions efficaces du MOILP

Dans le tableau suivant on donne les valeurs des fonctions objectifs du problème $MOILP$ et celle de la fonction principale (P_E) pour les différentes solutions efficaces trouvées.

$x \in Eff$	(2,0)	(3,0)	(2,1)	(3,1)	(1,2)	(2,2)	(3,2)
$Z_1(x)$	2	3	0	1	-3	-2	-1
$Z_2(x)$	-2	-3	2	1	7	6	5
$\phi(x)$	-2	-3	-4	-5	-5	-6	-7

TABLE 3.1: Tableau de valeurs

D'après le tableau, on remarque que la solution optimale est donc $(2, 0)$ avec une valeur objectif de -2.

Déroulement de l'algorithme

1^{ère} itération :

Initialisation :

On pose : $\phi_{inf} = -\infty$; $\phi_{sup} = +\infty$; $l = 1$.

On résoud le problème relaxé (P_R) suivant :

$$(P_R) \left\{ \begin{array}{ll} \text{maximiser} & \phi(x) = -x_1 - 2x_2 \\ \text{s.c} & \\ & -2x_1 + x_2 \leq 0 \quad ; \\ & x_1 \leq 3 \quad ; \\ & x_2 \leq 2 \quad ; \\ & x \geq 0 \quad . \end{array} \right.$$

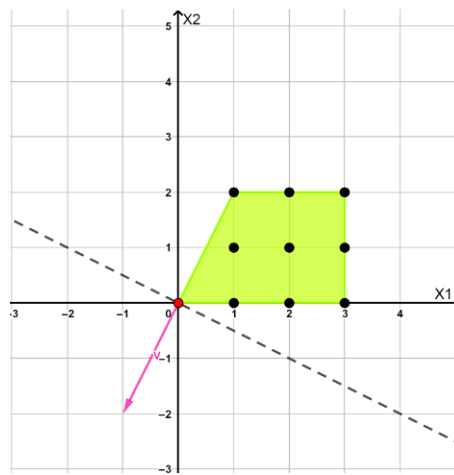


FIGURE 3.3: Résolution graphique de problème relaxé

On obtient la solution $x^1 = (0, 0)$ comme solution optimale de P_R . Avec $Z(x^1) = (0, 0)$ son image dans l'espace de résultat.

Etape (1) :

Dans cette étape on va faire un test d'efficacité d' **Ecker et kouada**[19] par la résolution de problème suivant :

$$P(x^1) \left\{ \begin{array}{l} \max \theta = \psi_1 + \psi_2 \\ s.c \\ x_1 - 2x_2 - \psi_1 = 0 \quad ; \\ -x_1 + 4x_2 - \psi_2 = 0 \quad ; \\ -2x_1 + x_2 \leq 0 \quad ; \\ x_1 \leq 3 \quad ; \\ x_2 \leq 2 \quad ; \\ x \geq 0 \quad , x \in Z \quad , \psi_1, \psi_2 \in R \end{array} \right.$$

Soit $\hat{x}^1 = (3, 1)$ la solution de ce problème , $\hat{x}^1 \neq x^1$ alors $x^1 = (0, 0)$ n'est pas efficace et $\hat{x}^1 = (3, 1)$ est une solution efficace générer par le test d'efficacité, donc on pose : $\phi_{sup} = dx^1 = 0$

Etape (2) :

$\hat{x}^1 \in Eff$, on résoudre le problème (T_1) suivant pour déterminer tous le solutions alternatives à $\hat{x}^1$

$$(T_1) \left\{ \begin{array}{l} \max \phi(x) = -x_1 - 2x_2 \\ s.c \\ x_1 - 2x_2 = 1 \quad ; \\ -x_1 + 4x_2 = 1 \quad ; \\ -2x_1 + x_2 \leq 0 \quad ; \\ x_1 \leq 3 \quad ; \\ x_2 \leq 2 \quad ; \\ x \geq 0 \quad x \in Z \end{array} \right.$$

On obtient $\tilde{x}^1 = (3, 1)$, et donc $\tilde{x}^1 = \hat{x}^1$. C'est-à-dire qu'il n'y a pas une solution alternative à $\hat{x}^1$ qui améliore la valeur de la fonction objectif.

On a $\phi(\tilde{x}^1) = -5 > \phi_{inf} = -\infty$, alors $\phi_{inf} = -5$ et $x_{opt} = \tilde{x}^1 = (3, 1)$.

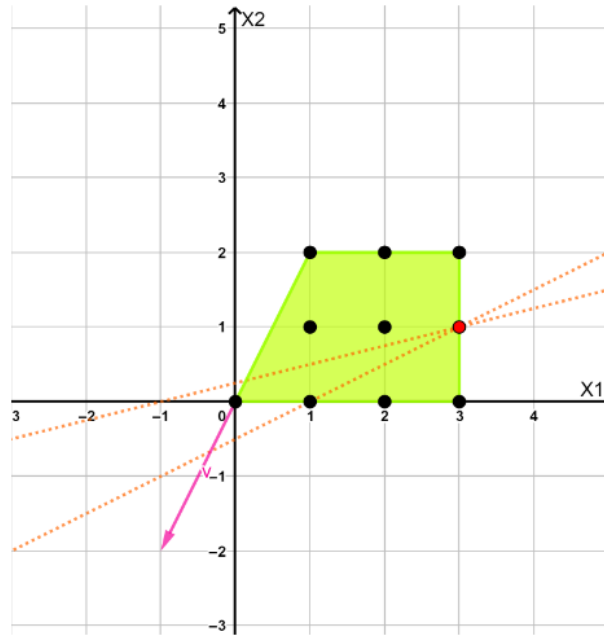


FIGURE 3.4: Résolution graphique de T_1

Etape (3) :

On résoud le problème (P_1) suivant pour déterminer l'amélioration d'aux moins un critère.

$$(P_1) \left\{ \begin{array}{ll} \text{maximiser} & \phi(x) = -x_1 - 2x_2 \\ \text{s.c} & \\ & -2x_1 + x_2 \leq 0 \quad ; \\ & x_1 \leq 3 \quad ; \\ & x_2 \leq 2 \quad ; \\ & x_1 - 2x_2 \geq 2y_1^1 - 3(1 - y_1^1) \quad ; \\ & -x_1 + 4x_2 \geq 2y_2^1 - 7(1 - y_2^1) \quad ; \\ & y_1^1 + y_2^1 = 1 \quad ; \\ & x \geq 0, \quad y_1^1 \in \{0, 1\}, \quad y_2^1 \in \{0, 1\} \quad . \end{array} \right.$$

La résolution de P_1 donne la solution $x^2 = (2, 0)^t$ qui améliore le premier critère, on la résumé dans le tbleau suivant :

x	(3,1)	(2,0)
Z_1	1	2
Z_2	1	-2

On a $\phi(x^2) = -2$, $x_{opt} = x^2$ et $l = l + 1 = 2$.

2^{ème} itération :

On teste l'efficacité de x^2 **Etape (1)** :

$$P(x^2) \left\{ \begin{array}{ll} \max & \theta = \psi_1 + \psi_2 \\ \text{s.c} & \\ & x_1 - 2x_2 - \psi_1 = 2 \quad ; \\ & -x_1 + 4x_2 - \psi_2 = -2 \quad ; \\ & -2x_1 + x_2 \leq \quad ; \\ & x_1 \leq 3 \quad ; \\ & x_2 \leq 2 \quad ; \\ & x \geq 0; x \in Z; \psi_1 \in R; \psi_2 \in R \quad . \end{array} \right.$$

Par la résolution du problème $(P(x^2))$, on trouve que $x^2 = (2, 0)^t$ est une solution efficace donc on arrête l'algorithme avec :

$x_{opt} = x^2$ et $\phi(x_{opt}) = -2$.

3.2.6 Méthode de Chaabane et Pirlot [9]

Chaabane et Pirlot [9] ont proposé un algorithme exacte pour maximiser une fonction linéaire notée ϕ sur l'ensemble des solutions efficaces d'un problème *MOILP* sans avoir à énumérer explicitement tous les éléments de cet ensemble. Deux techniques sont utilisées : la première consiste à réduire progressivement le domaine admissible en ajoutant plus de contraintes en éliminant tous les points dominés par une solution courrante ; la deuxième, lorsque la nouvelle solution obtenue en maximisant la fonction ϕ dans le domaine réduit n'est pas efficace, une procédure d'exploration est appliqué sur les arêtes incident à cette solution afin de trouver une nouvelle alternative des solutions efficaces si elles existent. L'algorithme produit non seulement une valeur optimale de la fonction linéaire mais aussi un sous-ensemble de solutions non dominées dans la direction de ϕ et ça peut être utile dans la pratique.

3.2.7 Description de la méthode

La procedure commence par une solution initiale efficace x^0 obtenue par la résolution du problème $(P(x^*))$, avec x^* est la solution optimale de problème relaxé (P_R) . x^0 va être utilisé pour la résolution de problème (T_l) suivant :

$$(T_l) : \max\{dx/cx = cx^0, x \in D\}$$

Où C est le vecteur des critères. $\tilde{x}^l$ est la solution optimale de (T_l) , $\hat{x}^l$ est considéré comme la première solution efficace.

On initialise $x_{opt} = \hat{x}^l$ et $\phi_{opt} = d\hat{x}^l$.

À chaque itération l , on résoud les problèmes (P_l) définie par :

$$(P_l) : \max\{\phi(x) = dx/x \in D \setminus \bigcup_{s=1}^l D_s\}$$

Où $D_s = \{x \in Z^n / cx \leq c\tilde{x}^s\}$, avec $s \in \{1, \dots, l\}$.

et ceci pour éliminer toutes les solutions alternatives à $\hat{x}^l$.

Définition 3.2.1 Soit x^* une solution optimale de (P_R) , solution réalisable $\tilde{x} \in D$ est dit être une solution alternative à x^* si $\phi(x) = \phi(\tilde{x})$ et $x^* \neq \tilde{x}$.

On définit le problème P_λ^k suivant :

$$\begin{cases} \text{maximiser} & Z_\lambda(x) = \sum_{i=1}^{i=p} \lambda_i Z_i \\ \text{s.c} & \\ \lambda \in [0, 1]; & \sum_{i=1}^p \lambda_i = 1; \\ x \in D^k. & \end{cases}$$

tel que : $D^k = H^k \cap \{x \in D / dx \geq dx_{opt} + 1\}$ et

$$H^k = H^{k-1} \cap \left\{ \begin{array}{l} x \in D / Z_i(x) \geq (Z_i(x_{opt}) + 1)y_i^k - M_i(1 - y_i^k), \quad i = \overline{1, p} \\ \sum_{i=1}^{i=p} y_i^k \geq 1, y_i^k \in \{0, 1\}, i = \overline{1, p}. \end{array} \right\}$$

où

- M_i est une borne inférieure de la i^{me} fonction objectif dans S.
- y_i^k est une variable binaire associée à chaque critère et définie par :

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

- La contrainte $\sum_{i=1}^{i=p} y_i^k \geq 1$ signifie qu'au moins un des critères est amélioré.
- La contrainte supplémentaire $dx \geq dx_{opt} + 1$ permet de se déplacer vers une autre solution meilleure sur ϕ si elle existe.

Formulation de l'algorithme

Algorithme 4 : Algorithme de Chaabane et Pirlot

Initialisation :

$\phi_{opt} \leftarrow -\infty ; l \leftarrow 0 .$

Etape 1 :

Résoudre (P_R) .

si $S = \emptyset$ **alors**

 | P_E n'admet pas de solution

sinon

 | Soit x^l une solution optimale de (P_R) .

si x^l efficace test **alors**

$x_{opt} \leftarrow x^l ; \phi_{opt} \leftarrow \phi(x^l).$

sinon

$\tilde{x}^l$ solution efficace (test d'efficacité).

Etape 2 : Résoudre le problème (T_l) et soit $\tilde{x}^l$ sa solution optimale.

si $\phi(\tilde{x}^l) > \phi_{opt}$ **alors**

 | $x_{opt} \leftarrow \tilde{x}^l ; \phi_{opt} \leftarrow \phi(\tilde{x}^l) ; l \leftarrow l + 1$

 Résoudre le problème (P_l) et soit x^l sa solution optimale.

si $\phi(\tilde{x}^l) < \phi_{opt}$ **alors**

 | **stop.**

sinon si x^l efficace **alors**

$x_{opt} \leftarrow x^l ; \phi_{opt} \leftarrow \phi(x^l),$ **stop.**

sinon

 | $J_l = \{j \in N_l / \phi_j - c_j = 0\}$

 | N_l ensemble d'indices des variable de base de x^l .

si $J_l = \emptyset$ **alors**

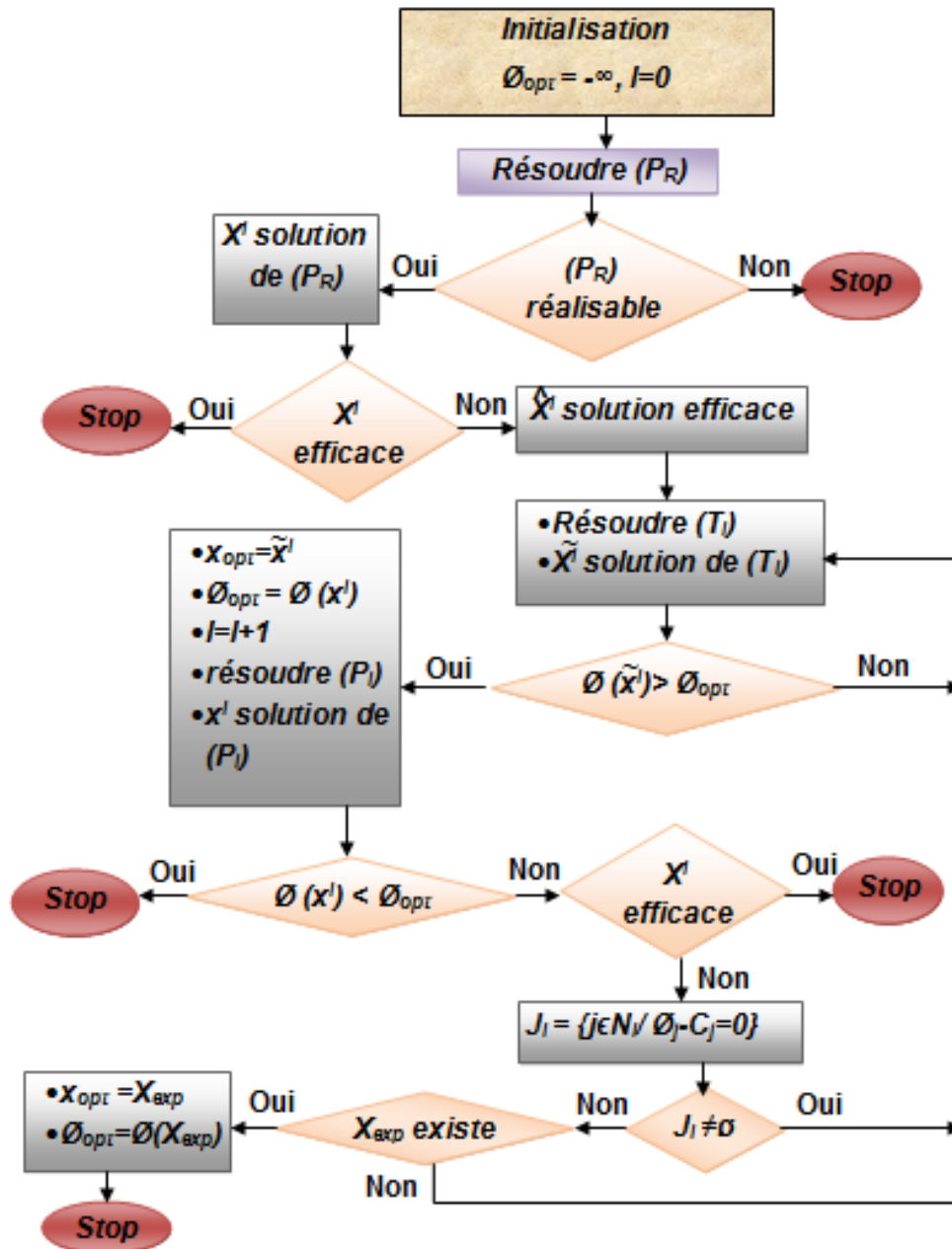
 | Aller à **Etape 2 :**

sinon si x_{exp} existe **alors**

 | $x_{opt} \leftarrow x_{exp} ; \phi_{opt} \leftarrow \phi(x_{exp}),$ **stop.**

 | x_{exp} une solution explorer de x^l .

Diagramme de la méthode de Chaabane et Pirlot



3.2.8 Exemple

On considère le problème suivant :

$$\begin{cases} \max \phi(x) = -x_1 - 3x_2 \\ s.c \\ x \in Eff \end{cases}$$

où :

$$(MOILP) \begin{cases} \max Z_1 = 2x_1 - x_2 \\ \max Z_2 = -x_1 + 2x_2 \\ s.c \\ x_1 + x_2 \leq 10 \quad ; \\ x_1 \leq 5 \quad ; \\ x_2 \leq 7 \quad ; \\ x \in N \quad . \end{cases}$$

Les solutions réalisables du problème *MOILP* sont représentées dans la figure (3.5).

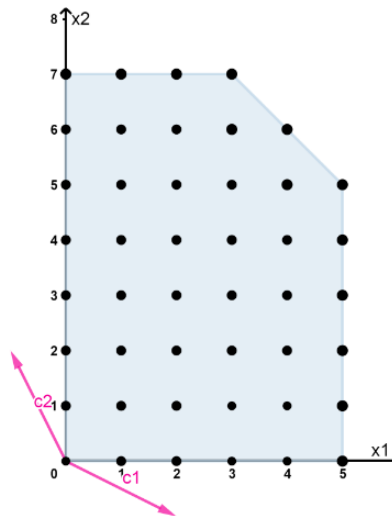


FIGURE 3.5: Ensemble de décision du MOILP

Les solutions efficaces représentées dans la figure (3.6) sont obtenues par la méthode du cône positif et sont :

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

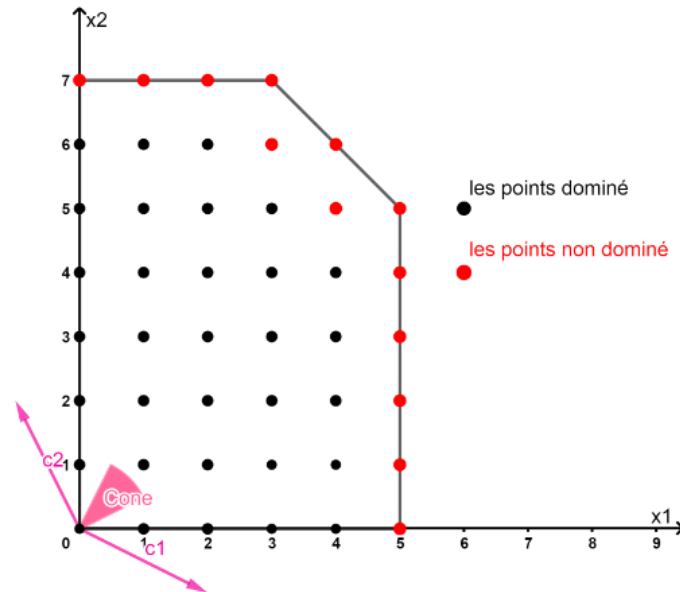


FIGURE 3.6: Résolution graphique par la méthode des cônes (Chaabane)

Dans ce tableau on a présenté les valeurs des fonctions objectifs du problème ($MOILP$) et du problème principale (P_E).

Eff	(0,7)	(1,7)	(2,7)	(3,7)	(3,6)	(4,6)	(4,5)
Z_1	-7	-5	-3	-1	0	2	3
Z_2	14	13	12	11	9	8	6
ϕ	-21	-22	-23	-24	-21	-22	-19
Eff	(5,5)	(5,4)	(5,3)	(5,2)	(5,1)	(5,0)	-
Z_1	5	6	7	8	9	10	-
Z_2	5	3	1	-1	-3	-5	-
ϕ	-20	-17	-14	-11	-8	-5	-

Déroulement de l'algorithme

Initialisation :

$\phi_{opt} = -\infty$; $l=0$; $H^0 = D$.

Résoudre le problème relaxé suivant :

$$(P_R) \left\{ \begin{array}{ll} \max \phi(x) = -x_1 - 3x_2 & \\ s.c & \\ x_1 + x_2 \leq 10 & ; \\ x_1 \leq 5 & ; \\ x_2 \leq 7 & ; \\ x \in N & . \end{array} \right.$$

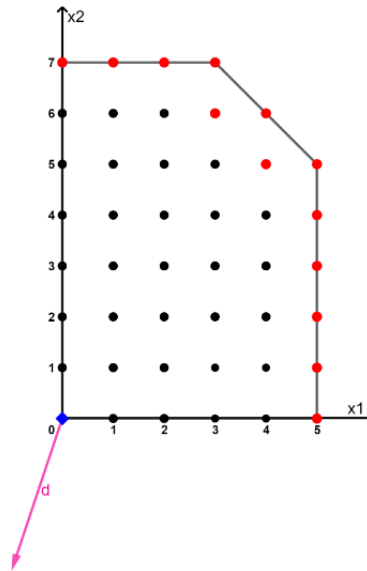
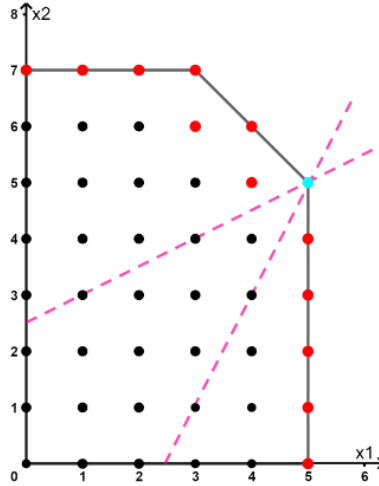


FIGURE 3.7: Résolution graphique du problème relaxé (Chaabane)

On obtient $x^0 = (0, 0)$ comme solution optimale de (P_R) . $-M_1 = -7$, $-M_2 = -5$.
En appliquant le test d'efficacité de **Ecker et kouada** [19] par la résolution de problème suivant :

$$(P(x^0)) \left\{ \begin{array}{ll} \max \theta = \psi_1 + \psi_2 & \\ 2x_1 - x_2 - \psi_1 = 0 & ; \\ -x_1 + 2x_2 - \psi_2 = 0 & ; \\ x_1 + x_2 \leq 10 & ; \\ x_1 \leq 5 & ; \\ x_2 \leq 7 & ; \\ x_1, x_2 \in D; \psi_1, \psi_2 \in \mathbb{R} & . \end{array} \right.$$



On trouve que x^0 n'est pas une solution efficace. $\hat{x}^0 = (5, 5)$ est la solution efficace obtenu par le test d'efficacité [19]

1^{er}Itération :

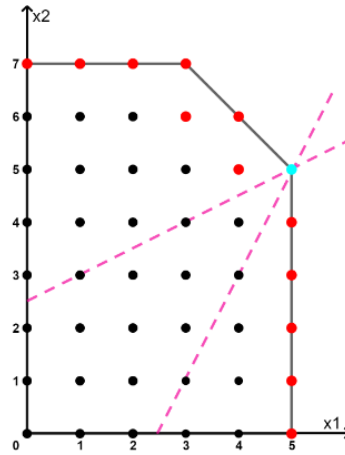
On cherche les solutions alternatives de $\hat{x}^0$ par la résolution du problème T_0 suivant :

$$(T_0) \left\{ \begin{array}{ll} \max \phi(x) = -x_1 - 3x_2 & \\ s.c & \\ 2x_1 - x_2 = 5 & ; \\ -x_1 + 2x_2 = 5 & ; \\ x_1 + x_2 \leq 10 & ; \\ x_1 \leq 5 & ; \\ x_2 \leq 7 & ; \\ x \in N & . \end{array} \right.$$

La solution $\hat{x}^0 = (5, 5)$ est sa solution optimale avec $\phi(\hat{x}^0) = -20$. C'est-à-dire qu'il n'y a pas de solutions alternatives pour $\hat{x}^0$ qui améliore la fonction objectif.

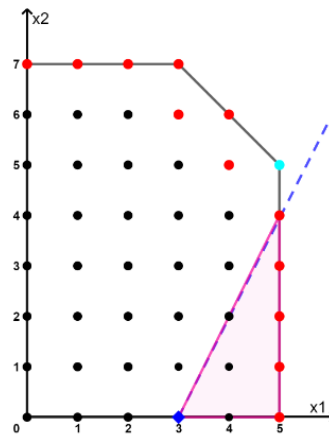
$\phi(\hat{x}^0) > \phi_{opt}$, donc $x_{opt} = (5, 5)$ et $\phi_{opt} = -20$.

$l = 1$



Pour chercher une amélioration d'un critère on résoud le problème (P_1) suivant :

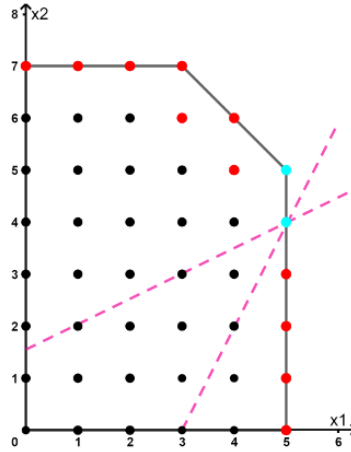
$$(P_1) \left\{ \begin{array}{ll} \max \phi(x) = -x_1 - 3x_2 & \\ s.c & \\ x_1 + x_2 \leq 10 & ; \\ x_1 \leq 5 & ; \\ x_2 \leq 7 & ; \\ 2x_1 - x_2 \geq 6y_1^1 - 7(1 - y_1^1) & ; \\ -x_1 + 2x_2 \geq 6y_2^1 - 5(1 - y_2^1) & ; \\ y_1^1 + y_2^1 \geq 1 & ; \\ x \in \mathbb{N}, (y_1^1, y_2^1) \in \{0, 1\}^2 & . \end{array} \right.$$



$x^1 = (3, 0)$ sa solution optimale avec $Z(x^1) = (6, -3)$ et $y = (0, 1)$ (une amélioration par rapport au premier critère .

Comme $\phi(x^1) = -3 > \phi_{opt}$ donc on teste l'efficacité de x^l par la résolution du problème suivant :

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



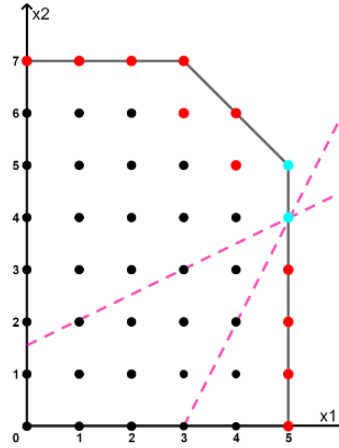
On obtient $\theta^* \neq 0$ et $\hat{x}^1 = (5, 4)$ une solution optimale de $(P(x^1))$. Alors $x^1 = (3, 0)$ n'est pas efficace .

À partir de tableau optimale de x^1 nous avons $Z(\hat{x}^1) = (6, 3)$ et puis $J_1 = \{j \in N_1 \mid c_j - d_j = 0\} = \emptyset$.

2^{ème} Itération :

Résoudre le problème T_1 suivant :

$$(T_1) \left\{ \begin{array}{ll} \max \phi(x) = -x_1 - 3x_2 & \\ s.c & \\ 2x_1 - x_2 = 6 & ; \\ -x_1 + 2x_2 = 3 & ; \\ x_1 + x_2 \leq 10 & ; \\ x_1 \leq 5 & ; \\ x_2 \leq 7 & ; \\ x \in N & . \end{array} \right.$$

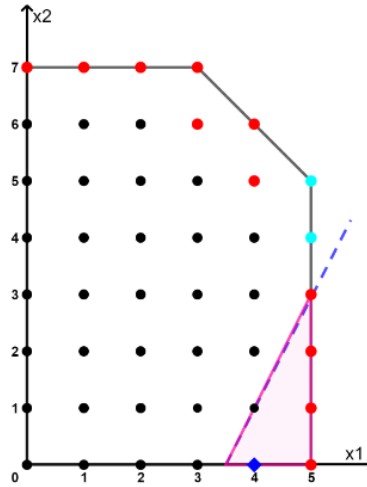


$\tilde{x}^1 = (5, 4)$ est sa solution optimale avec $\phi(\tilde{x}^1) = -17 > \phi_{opt}$. Alors $x_{opt} = (5, 4)$ et $\phi_{opt} = -17$.

$l = 2$

Résoudre le problème (P_2) suivant :

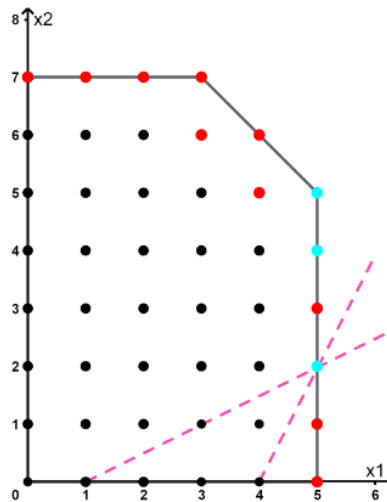
$$(P_2) \left\{ \begin{array}{ll} \max \phi(x) = -x_1 - 3x_2 & \\ s.c & \\ x_1 + x_2 \leq 10 & ; \\ x_1 \leq 5 & ; \\ x_2 \leq 7 & ; \\ 2x_1 - x_2 \geq 7y_1^2 - 7(1 - y_1^2) & ; \\ -x_1 + 2x_2 \geq 4y_2^2 - 5(1 - y_2^2) & ; \\ y_1^2 + y_2^2 \geq 1 & \\ x \in N, (y_1^2, y_2^2) \in \{0, 1\}^2 & . \end{array} \right.$$



$x^2 = (4, 0)$ une solution optimale pour P_2 avec $Z(x^2) = (8, -4)$ et $Y = \{1, 0, 1, 0\}$ (une autre amélioration par rapport au premier critère).

$\phi(x^2) = -4 > \phi_{opt}$, on test l'efficacité[19] de x^2 par la résolution du problème suivant :

$$(P(x^2)) \left\{ \begin{array}{l} \max \theta = \psi_1 + \psi_2 \\ s.c \\ 2x_1 - x_2 - \psi_1 = 8 \\ -x_1 + 2x_2 - \psi_2 = -4 \\ x_1 + x_2 \leq 10 \quad ; \\ x_1 \leq 5 \quad ; \\ x_2 \leq 7 \quad ; \\ x_1, x_2 \in D; \psi_1, \psi_2 \in \mathbb{R} \quad . \end{array} \right.$$



Soit $\hat{x}^2 = (5, 2)$ une solution efficace avec $Z(\hat{x}^1) = (8, -1)$ obtenu de test d'efficacité de x^2 . Donc x^2 n'est pas efficace.

À partir de tableau optimale de x^2 $J_2 = \{j \in N_2 \mid c_j - Z_j = 0\} = \emptyset$.

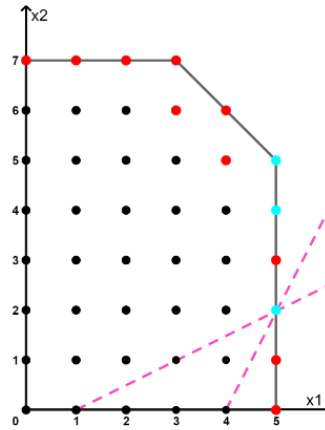
3^{ème} Itération :

Résolution de problème (T_2)

$$(T_2) \left\{ \begin{array}{ll} \max \phi(x) = -x_1 - 3x_2 & \\ \text{s.c} & \\ x_1 + x_2 \leq 10 & ; \\ x_1 \leq 5 & ; \\ x_2 \leq 7 & ; \\ 2x_1 - x_2 = 8 & ; \\ -x_1 + 2x_2 = -1 & ; \\ x \in N & . \end{array} \right.$$

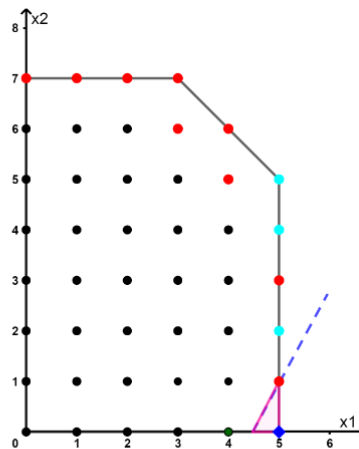
La solution $\tilde{x}^2 = (5, 2)$ est optimale de (T_2) avec $\phi(\tilde{x}^2) = -11 > \phi_{opt}$. Donc $x_{opt} = (5, 2)$ et $\phi_{opt} = -11$.

$l = 3$



Résoudre le problème (P_3) suivant :

$$(P_3) \left\{ \begin{array}{l} \max \phi(x) = -x_1 - 3x_2 \\ s.c \\ x_1 + x_2 \leq 10 \quad ; \\ x_1 \leq 5 \quad ; \\ x_2 \leq 7 \quad ; \\ 2x_1 - x_2 \geq 9y_1^3 - 7(1 - y_1^2) \quad ; \\ -x_1 + 2x_2 \geq -5(1 - y_2^2) \quad ; \\ y_1^3 + y_2^3 \geq 1 \\ x \in N, (y_1^3, y_2^3) \in \{0, 1\}^2 \quad . \end{array} \right.$$



La solution optimale de ce problème est $x^3 = (5, 0)$ avec $Z(x^3) = (10, -5)$ et

$Y = \{1, 0, 1, 0, 1, 0\}$ (une autre amélioration par rapport au premier critère)
 $\phi(x^3) = -5 > \phi_{opt}$.On test l'efficacité de x^3 par la résolution du problème suivant :

$$(P(x^3)) \left\{ \begin{array}{l} \max \theta = \psi_1 + \psi_2 \\ s.c \\ 2x_1 - x_2 - \psi_1 = 10 \quad ; \\ -x_1 + 2x_2 - \psi_2 = -5 \quad ; \\ x_1 + x_2 \leq 10 \quad ; \\ x_1 \leq 5 \quad ; \\ x_2 \leq 7 \quad ; \\ x_1, x_2 \in D; \psi_1, \psi_2 \in \mathbb{R} \quad . \end{array} \right.$$

$\hat{x}^3 = (5, 0) = x^3$ est une solution efficace. Donc l'algorithme est terminé avec : $x_{opt} = (5, 2)$ et $\phi_{opt} = -5$.

3.2.9 La Méthode Chergui et Ouail [12]

Chergui et Ouail ont proposé un algorithme exact pour résoudre le problème (P_E) , l'algorithme fournit une solution optimale de (P_E) sans avoir à énumérer l'ensemble de toutes les solutions efficaces du problème multi-objectif $(MOILP)$. La méthode est basée sur le principe de séparation et évaluation "Branch & Bound" et l'utilisation d'une coupe efficace à chaque itération de l'algorithme en mesure de supprimer des solutions entières qui ne sont pas efficaces pour le problème multi-objectif $(MOILP)$.

L'approche adoptée pour ce travail pour générer la solution optimale du problème central (P) est basée sur la résolution du problème unicritère linéaire (P_l) à chaque étape l ($l \geq 0$) de l'algorithme. Ce problème (P_l) est formulé par :

$$(P_l) \left\{ \begin{array}{l} \max \phi(x) = d^l x \\ s.c \\ x \in S_l \end{array} \right.$$

$S_0 = S$ et sans contrainte d'intégrité des variables.

La direction de croissance de chaque critère Z_i , $i = \overline{1, p}$ du problème $(MOILP)$ est déterminée à l'aide de leur gradient $\bar{c}_i$ et celle du critère principal ϕ du problème (P_E) est déterminée par $\bar{d}$. La méthode utilise ces informations pour construire une coupe en mesure de supprimer des solutions entières qui ne sont pas efficaces pour le problème multi-objectif $(MOILP)$.

Définition 3.2.2 Coupe efficace

Une coupe est dite efficace pour le problème $(MOILP)$ si son adjonction au domaine S n'élimine pas les solutions réalisables entières et efficaces de S .

Avant de déterminer l'expression mathématique de cette coupe efficace, il faut d'abord définir les ensembles suivants en x_l^* :

$B_l, (N_l)$ est défini comme l'ensemble des indices des variables de base (hors base) respectivement de x_l^*

$$H = H_l^1 \cup H_l^2 \quad (3.1)$$

$$H_l^1 = \{j \in J_l / \exists i \in \{1, \dots, p\}; \bar{c}_j^i > 0 \text{ et } \bar{c}_j \geq 0\} \quad (3.2)$$

$$H_l^2 = \{j \in J_l / \bar{c}_j^i = 0, \forall i = \overline{1, p} \text{ et } \bar{c}_j \geq 0\} \quad (3.3)$$

Où $\bar{\gamma}_j^i$ est le gradient de vecteur d.

$$S_{l+1} = \left\{ x \in S_l / \sum_{j \in H_l} x_j \geq 0 \right\} \quad (3.4)$$

La coupe efficace $\sum_{j \in H_l} x_j \geq 1$ élimine les solutions entières non efficaces.

Description de la méthode

On pose $\phi_{opt} = -\infty$. La méthode consiste à résoudre le programme (P_l) [3.2.9] à chaque étape de l'algorithme. Chaque programme (P_l) [3.2.9] correspond au nœud l dans une arborescence structurée.

Un nœud l de l'arborescence est sondé si le programme correspondant (P_l) [3.2.9] n'est pas réalisable ou que l'ensemble H_l est vide (c'est à dire qu'aucun critère ne peut être amélioré et donc il n'aura plus de solutions efficaces).

Si la solution optimale x_l^* du programme (P_l) n'est pas entière, soit x_j une composante de x_l^* telle que $x_j = \alpha_j$ où α_j est un nombre fractionnaire. Le nœud l de l'arborescence est alors séparé en deux nœuds qui lui sont imposées par les contraintes additionnelles $x_j \leq \lfloor \alpha_j \rfloor$ et $x_j \geq \lfloor \alpha_j \rfloor + 1$ où $\lfloor \alpha_j \rfloor$ indique la partie entière du nombre réel α_j . Dans chaque nœud, le programme linéaire obtenu doit être résolu jusqu'à obtention d'une solution entière si elle existe.

En présence d'une solution entière, la solution est testée pour son efficacité; si elle est efficace, ϕ_{opt} est mis à jour et le nœud correspondant est sondé, sinon la coupe $\sum_{j \in H_l} x_j \geq 1$ est rajoutée au programme et le nouveau programme est résolu.

La méthode se termine quand tous les nœuds créés sont sondés. La solution entière optimale du problème (P) est alors x_{opt} qui correspond à ϕ_{opt} .

Formulation de l'algorithme

Algorithme 5 : Algorithme de Chergui et Ouail

Etape (1) : (Initialisation)

$l = 1$, $\phi_{opt} = -\infty$

Etape (2) :

Tant qu'il existe un nœud non encore sondé, choisir le nœud non sondé le plus récemment créé et résoudre le programme linéaire correspondant (Pl) en utilisant la méthode dual du simplexe.

(On utilise la méthode du simplexe pour résoudre le problème initial (Pl), pour $l = 0$)

si (Pl) *n'est pas réalisable* **alors**

le nœud correspondant est sondé ;

sinon

soit x_l^* sa solution optimale obtenue ;

si $\phi_{opt} \geq \phi(x_l^*)$ **alors**

le nœud est sondé. Aller à l'étape (2);

sinon si x_l^* *est entière* **alors**

aller à l'étape (2a) ;

sinon

aller à l'étape (2b).

Etape (2a)

Tester l'efficacité de x_l^* .

si x_l^* *est efficace* **alors**

mettre à jour $\phi_{opt} = \phi(x_l^*)$ et $x_{opt} = x_l^*$; le nœud l est sondé.

Aller à l'étape (2);

sinon

Déterminer les ensembles N_l , H_l^1 , H_l^2 et H_l , construire la coupe efficace ;

si $H_l = \emptyset$ **alors**

le nœud l est sondé puisque il n'y aura plus d'autre solutions efficaces.

Aller à l'étape (2) ;

sinon

Ajouter la coupe efficace :

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

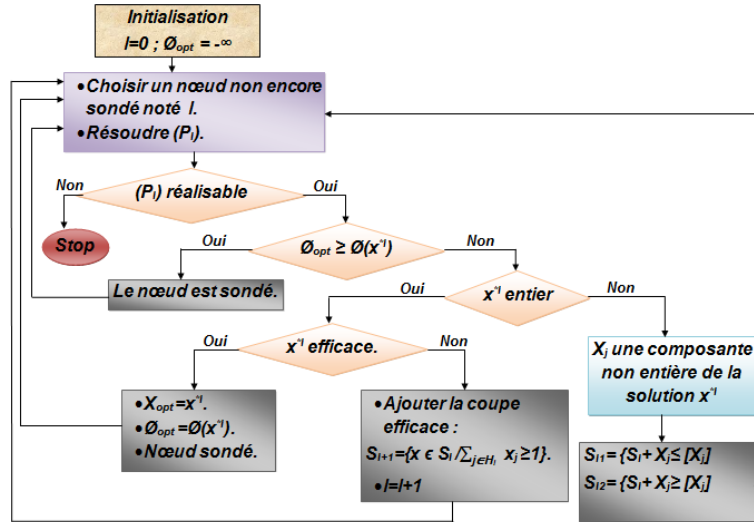
et aller à l'étape (2);

Etape (2b) :

Appliquer le processus de branchement "Branch & Bound" et séparer le nœud

actuel l en deux nouveaux nœuds $l_1, l_1 > l + 1$ et $l_2, l_2 > l + 1$. Aller à l'étape (2).

Organigramme de la méthode



3.2.10 Exemple

Pour illustrer l'utilisation de l'algorithme, on considère le problème suivant :

$$(MOILP) \left\{ \begin{array}{ll} \max Z_1 = 3x_1 - 4x_2 & \\ \max Z_2 = 2x_1 - 8x_2 & \\ s.c & \\ 6x_1 + x_2 \leq 5 & ; \\ -2x_1 + x_2 \leq 8 & ; \\ x_2 & ; \\ x \in \mathbb{Z} & . \end{array} \right.$$

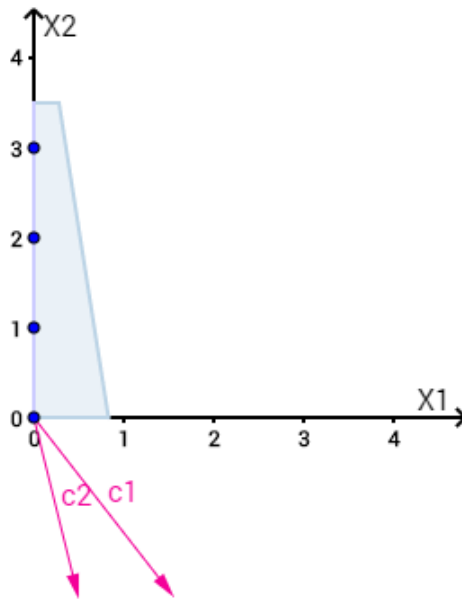


FIGURE 3.8: Résolution graphique du MOILP

Les solution réalisable du *MOILP* représenter dans la figure (3.8) sont :
 $D = \{(0, 0); (0, 1); (0, 2); (0, 3)\}$.

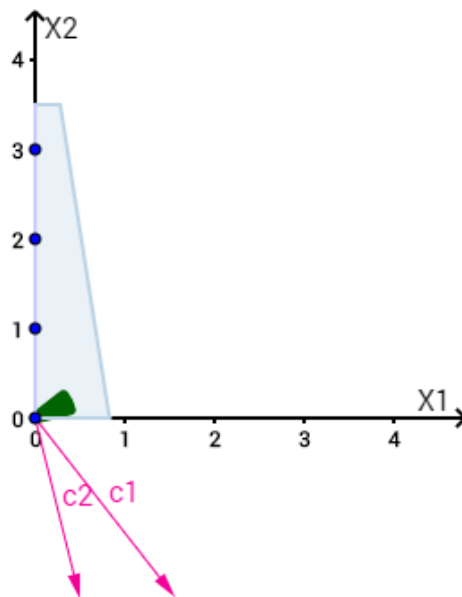


FIGURE 3.9: Solution efficace du MOILP

Les solutions efficaces du problème *MOILP* obtenues par la méthode de cône représentés dans la figure (3.9) sont :

$$Eff = \{(0, 0); (0, 1); (0, 2); (0, 3)\}$$

Soit le problème relaxé P_R suivant :

$$(P_R) \begin{cases} \max \phi = -3x_1 + 5x_2 \\ s.c \\ 6x_1 + x_2 \leq 5 & ; \\ -2x_1 + x_2 \leq 8 & ; \\ x_2 & ; \\ x \in \mathbb{D} & . \end{cases}$$

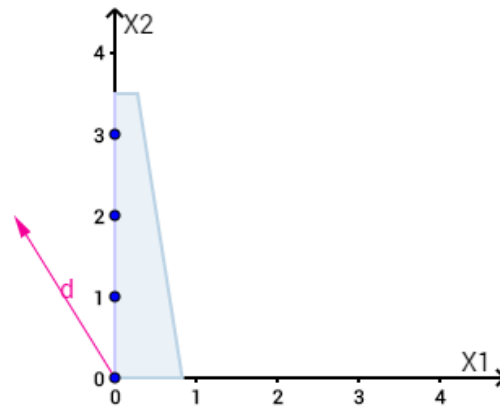


FIGURE 3.10: Résolution graphique du problème relaxé

Déroulement de l'algorithme

Initialisation :

$$\phi_{opt} = -\infty, l = 1.$$

La résolution de problème relaxé (P_R) nous donne le tableau suivant :

T_1	x_1	x_5	b
x_2	0	$\frac{1}{2}$	$\frac{7}{2}$
x_3	6	$\frac{1}{2}$	$\frac{3}{2}$
x_4	-2	$\frac{1}{2}$	$\frac{9}{2}$
$\bar{d}$	-3	$\frac{-5}{2}$	

La solution trouvée est $x_1^* = (0, \frac{7}{2})$ n'est pas entière, on applique alors le processus de Branch & Bound. Deux nœuds sont créés :

$$N_1 : x_2 \leq 3$$

$$N_2 : x_2 \geq 4$$

La contrainte $x_2 \geq 4$ est rajoutée au Tableau 1 et on obtient le Tableau 2.

T_2	x_1	x_1	b
x_2	0	1	3
x_3	6	-1	2
x_4	-2	-1	5
x_5	0	2	1
$\bar{d}$	-3	-5	
$\bar{c}_1$	2	4	
$\bar{c}_2$	3	8	

La solution trouvée $x_2^* = (0, 3)$ est entière et efficace, $\phi(x_2^*) = 15$. Donc le nœud est sondé. La contrainte $x_2 \leq 3$ est rajoutée au Tableau 1 et on obtient le Tableau 3.

T_3	x_3	x_7	b
x_2	$\frac{1}{7}$	$\frac{6}{7}$	$\frac{17}{7}$
x_6	$-\frac{1}{7}$	$-\frac{6}{7}$	$\frac{4}{7}$
x_4	$\frac{1}{7}$	$-\frac{8}{7}$	$\frac{45}{7}$
x_5	$-\frac{2}{7}$	$-\frac{12}{7}$	$\frac{15}{7}$
x_1	$\frac{1}{7}$	$-\frac{1}{7}$	$\frac{3}{7}$
$\bar{d}$	$-\frac{2}{7}$	$\frac{33}{7}$	

On obtient la solution $x_3^* = (\frac{3}{7}, \frac{17}{7})$ qui n'est pas entière. Alors en appliquant le processus de Branch & Bound, deux nœuds sont créés.

$$N_1 : x_1 \geq 1.$$

$$N_2 : x_1 \leq 0.$$

La contrainte $x_1 \geq 1$ est rajoutée au Tableau 3. Le tableau 4 est obtenu :

T_4	x_7	x_8	b
x_2	1	1	2
x_6	-1	-1	1
x_4	-1	1	6
x_5	-2	-2	3
x_1	0	1	0
x_3	-1	-7	3
$\bar{d}$	-5	-2	
$\bar{c}_1$	4	1	
$\bar{c}_2$	8	6	

La solution trouvée est entière $x_4^* = (0, 2)$ et efficace et $\phi(x_4^*) = 10$. Alors $\phi_{opt} = 10$ et $x_{opt} = (0, 2)$

La contrainte $x_1 \leq 0$ est rajoutée au Tableau 3. On obtient une solution $x_5^*(0, 3)$ entière et efficace et $\phi(x_5^*) = 15$, $\phi(x_5^*) > \phi_{opt}$. Alors $\phi_{opt} = 15$ et $x_{opt} = (0, 3)$.

Tous les nœuds sont sondé donc $x_{opt} = (0, 3)$, $\phi_{opt} = 15$.

Pour résumer cet exemple, nous présentons une arborescence qui représente les états des nœuds pendant le processus figure(3.11).

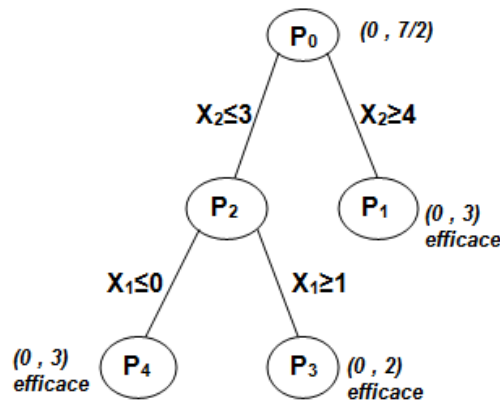


FIGURE 3.11: Arborescence des nœuds relatif à l'exemple

Exemple de Jorge

On résoud l'exemple de Jorge [28] avec notre méthode :

$$(MOILP) \left\{ \begin{array}{ll} \max Z_1 = x_1 - 2x_2 & ; \\ \max Z_2 = -x_1 + 4x_2 & ; \\ s.c & \\ -2x_1 + x_2 \leq 0 & ; \\ x_1 \leq 3 & ; \\ x_2 \leq 2 & ; \\ x \in Z_+ & . \end{array} \right.$$

$$(P_R) \left\{ \begin{array}{ll} \max \phi(x) = -x_1 - 2x_2 & ; \\ s.c & \\ -2x_1 + x_2 \leq 0 & ; \\ x_1 \leq 3 & ; \\ x_2 \leq 2 & ; \\ x \geq 0 & . \end{array} \right.$$

La résolution du problème relaxé nous obtenons le tableau suivant :

T_1	x_2	x_3	b
x_4	$\frac{1}{2}$	$\frac{1}{2}$	3
x_1	$-\frac{1}{2}$	$-\frac{1}{2}$	0
x_5	1	0	2
$\bar{d}$	$-\frac{5}{2}$	$-\frac{1}{2}$	
$\bar{c}_1$	$\frac{1}{2}$	$-\frac{3}{2}$	
$\bar{c}_2$	$-\frac{1}{2}$	$\frac{7}{2}$	

La solution optimale trouvé est $x_1^* = (0, 0)$ qu'elle est entière mais n'est pas efficace.

La contrainte $x_2 + x_3 \geq 1$ est rajoutée au Tableau 1. Le Tableau 2 est obtenu :

T_2	x_2	x_6	b
x_4	0	$\frac{1}{2}$	$\frac{5}{2}$
x_1	0	$-\frac{1}{2}$	$\frac{1}{2}$
x_5	1	0	2
x_3	1	-1	1
$\bar{d}$	-2	$-\frac{1}{2}$	

On obtient la solution $x_2^* = (\frac{1}{2}, 0)$ qui n'est pas entière, alors appliquant le processus de Branch & Bound. Deux nœuds sont créés.

$$N_1 = x_1 \geq 0.$$

$$N_2 = x_1 \leq 0.$$

La contrainte $x_1 \geq 1$ est rajouté au Tableau 2. On obtient le tableau 3.

T_3	x_2	x_7	b
x_4	0	1	2
x_1	0	-1	1
x_5	1	0	2
x_3	1	-2	2
x_6	0	-2	1
$\bar{d}$	-2	1	
$\bar{c}_1$	1	-2	
$\bar{c}_2$	-1	4	

On obtient la solution $x_3^* = (1, 0)$ qui est entière et n'est pas efficace, alors on ajoute la contrainte $x_2 + x_7 \geq 1$. Le tableau 4 est obtenu :

T_4	x_2	x_8	b
x_4	-1	1	1
x_1	1	-1	2
x_5	1	0	2
x_3	3	-2	4
x_6	2	-2	3
x_7	1	-1	1
$\bar{d}$	-1	-1	

On obtient la solution $x_5^* = (2, 0)$ qui est entière et efficace et $\phi(x_5^*) = -2$. Alors $x_{opt} = (2, 0)$, $\phi_{opt} = -2$. Le nœud est sondé. Il reste encore un nœud non sondé, ajoutant la contrainte $x_1 \leq 0$ au Tableau 2 on obtient la solution $x_6^* = (2, 0)$ qui est entière et efficace, alors $\phi(x_6^*) = -2$, le nœud est sondé.

Tous les nœuds de l'arborescence sont sondés.

Pour résumer cet exemple, nous présentons une arborescence qui représente les états des nœuds pendant le processus figure(3.12).

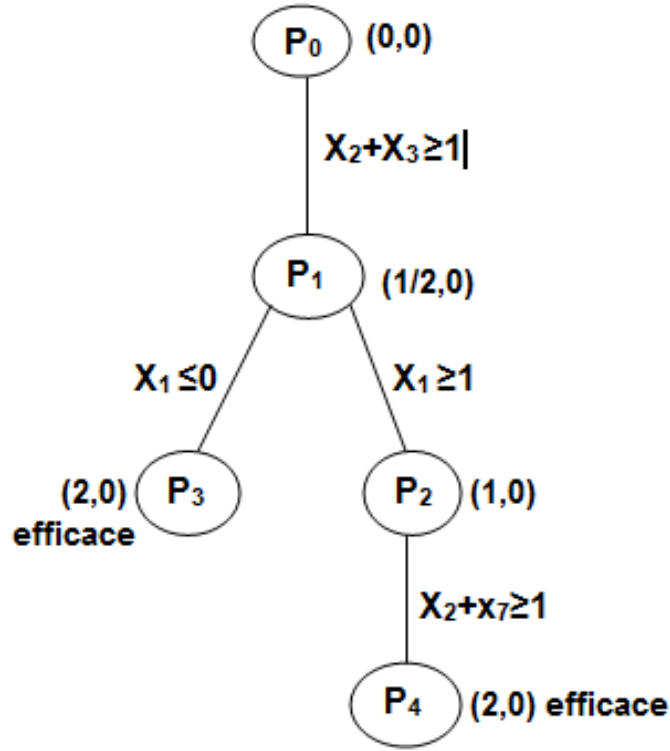


FIGURE 3.12: Arborécescence des nœuds relatif à l'exemple de Jorge

3.3 Conclusion

Ce chapitre traite un problème d'optimisation d'une fonction donné sur l'ensemble des solutions efficaces d'un problème multi-objectif, noté (P_E) . le problème (P_E) est en principe très difficile à résoudre, ceci est dû principalement à la non-convexité de son ensemble réalisable $E(P)$ et le problème appartient alors à la classe des problèmes d'optimisation globale (programmation non convexe).

Nous avons présenté trois méthodes de résolution du problème (P_E) dans le cas de variables entières, à savoir la méthode de Jorge [28], la méthode Chaabane et Pirlot[9] et la méthode de Chergui et Ouail [12].

Dans le chapitre suivant, une étude expérimentale sera faite pour les deux méthodes Jorge [28] et la méthode de Chergui et Ouail [12].

4

Implémentation et résultats

4.1 Introduction

L'évolution des outils informatiques a profondément influencé les méthodes de travail des ingénieurs et des chercheurs. Le traitement numérique des données, leur visualisation, ainsi que les techniques de modélisation et de simulation se sont notamment généralisés. Dans ce domaine, un logiciel commercial est devenu, ces dernières années, presque incontournable : il s'agit de Matlab de la société The Mathworks. Ce dernier met à la disposition de l'utilisateur un environnement performant pour mener à bien les calculs numériques.

Nous implémentons deux méthodes étudiée précédemment, la méthode de Jorge [28] et la méthode de Chergui et Ouail [12], sur le logiciel de programmation mathématique Matlab afin de pouvoir comparer l'efficacité de ces deux méthodes. Pour ce, une génération aléatoire des deux méthodes sur différentes instances a été réalisée.

4.2 Le logiciel Matlab

MATLAB (matrix laboratory) est un langage de programmation et un environnement de développement ; il est utilisé à des fins de calcul numérique. Développé par la société The MathWorks, MATLAB permet de manipuler des matrices, d'afficher des courbes et des données, de mettre en œuvre des algorithmes, de créer des interfaces utilisateurs, et peut s'interfacer avec d'autres langages comme le C, C++ et Java.

Les utilisateurs, de MATLAB sont de milieux très différents comme l'ingénierie, les sciences, et l'économie dans un contexte aussi bien industriel que pour la recherche.

4.2.1 Utilité du Matlab

- ✓ Matlab est disponible sur tous les systèmes d'exploitation standards (Windows, Unix/Linux, MacOSX...).
- ✓ Matlab a été originellement créé pour l'algèbre linéaire (matlab signifie " matrix laboratory "). Ainsi, l'objet de base est la matrice, et de nombreuses opérations matricielles sont déjà implémentées et optimisées.
- ✓ Un programme Matlab ne diffère pas fondamentalement d'un programme dans un langage classique. La seule différence importante tient dans le fait que la majorité des opérations par défaut dans Matlab étant vectorisées, il est généralement préférable d'exploiter cette caractéristique, par exemple en évitant des boucles dans les programmes.
- ✓ Matlab offre de nombreux moyens de communication entre un programme et l'utilisateur, par exemple : On peut afficher un message, une valeur à l'écran avec l'instruction `disp` : `disp('Ceci est un test')`.
- ✓ les instructions `tic` et `toc` permettent d'avoir une estimation du temps de calcul écoulé entre le `tic` et le `toc`).
- ✓ Matlab contient une option "**help**" qui aide l'utilisateur à obtenir l'aide sur Matlab par la syntaxe "help suivi par le nom de la fonction que vous recherchez."

4.3 Choix du langage

Les deux méthodes ont été développées en utilisant le logiciel MATLAB 2015, qui est un environnement puissant, complet et facile à utiliser, il est destinée au calcul scientifique. Il apporte aux ingénieurs, chercheurs et à tout, scientifique un système interactif intégrant le calcul numérique.

Le choix s'est porté sur l'emploi de ce langage, car il répond aux critères suivants :

- ✖ La maniabilité du langage : constitué d'un ensemble de possibilités faisant en sorte que le programmeur travaille avec aisance.
- ✖ Le bagage du langage : il contient une interface graphique et des fonctions prédéfinies telles que **intlinprog** qui permet de résoudre un problème de programmation linéaire en nombres entiers.

4.3.1 Programation avec Matlab

Pour d'écrire les fonctions sur Matlab, il existe deux méthodes :

- ✓ Soit directement dans la fenêtre de commande.
- ✓ Soit en utilisant l'éditeur de développement de Matlab, en sauvegardant les programmes dans des fichiers texte avec l'extension ".m".

Les programmes sauvegardés dans les fichiers Matlab (*.m) sont alors directement utilisables comme des fonctions Matlab à partir de la fenêtre de commande. En fait, l'appel à ces fonctions dans plusieurs d'autres fonctions différentes. Pour cela, le fichier doit se trouver dans le même répertoire Matlab, qui est en pratique le dossier Work.

Création d'une fonction Matlab

La création d'une fonction dans Matlab se fait par la syntaxe suivante :

$$function[s_1, s_2, \dots] = nonfonction(e_1, e_2, \dots)$$

Les variables $s_1 ; s_2 ; \dots$, sont les arguments de sortie (S) de la fonction et les variables $e_1 ; e_2 ; \dots$, sont les arguments d'entrée (E).

4.4 Résultats et discussion

Nous avons mis en œuvre l'algorithme proposé par Chergui et Ouail [12] ainsi que celui proposé par Jorge [28] afin de pouvoir faire une comparaison entre ces deux méthodes. Les deux algorithmes ont été mis en œuvre sous le langage de programmation Matlab R2015b, qui est un langage de programmation interactive, simple, efficace et optimisé pour le traitement des matrices. L'application a été déployée sur une machine équipée d'un processeur Intel(R)Core(TM) i3,3217U CPU,1.80GHZ et 4Go de mémoire.

Nous avons testé les algorithmes sur des instances générées aléatoirement suivant une loi uniforme à l'aide de la fonction prédéfinie en Matlab `Randi([ $v_{min}, v_{max}$ ],  $n, m$ )` qui renvoie une $n \times m$ matrice à coefficient entiers indépendamment uniformément distribués dans l'intervalle $[v_{min}, v_{max}]$.

$[v_{min}, v_{max}]$ est fixé à $[-20, 20]$ pour la matrice A et pour les coefficients des critères du problème (MOILP).

Pour chaque contrainte j , la valeur du second membre b_j est fixée à la partie entière de la somme des coefficients de cette contrainte divisé par 3.

Pour chaque instance (n, m, p) une séquence de 10 problèmes a été résolue par les deux algorithmes proposés, les résultats de l'expérience sont présentés dans le tableau suivant où :

- n : représente le nombre de variables du problème initiale.
- m : représente le nombre de contraintes.
- p : représente le nombre de critères du problème multi-objectif *MOILP*.
- 'Moyenne' : représente le temps moyen de calcul en secondes pour une instance donnée.
- 'Maximum' : rerésente le temps de calcul maximal en secondes pour une instance donnée.

			Ouail et Chergui		Jorge	
n	m	p	Moyenne	Maximum	Moyenne	Maximum
3	2	3	0.032277	0.047834	0.06956464	0.10816
3	5	4	0.03765082	0.10392	0.10350564	0.14602
5	5	4	0.05852782	0.18471	0.13693364	0.16687
5	8	4	0.0677403	0.11327	0.15857	0.22504
5	8	10	0.05186991	0.072963	0.28229273	0.30771
12	11	16	0.25163518	0.5036	1.1939	1.4842
12	17	16	0.59685	4.6842	1.68632727	2.3983
18	22	20	0.68909182	1.5023	3.96340909	5.1167
25	22	30	1.75605455	3.8274	11.0927727	14.7381
15	40	56	0.79010909	1.2805	10.5880909	17.3619

TABLE 4.1: Résultats de l'expérience numérique

D'après les résultats du tableau, on remarque que :

- La méthode Chergui et Ouail [12] est plus rapide que celle de Jorge [28].
- Le temps de calcul des deux méthodes augmente rapidement avec la taille des donnés, en particulier avec l'augmentation du nombre de variables, en revanche l'effet d'augmenter le nombre de contraintes pour un nombre fixe de variables est moins perceptible.
- Le nombre de critères ne fait pas augmenté le temps de calcul de façon significatif.
- La complexité algorithmique de la méthode dépend fortement du nombre de variables n et de contraintes m utilisées.

4.5 Conclusion

Dans ce dernier chapitre, nous avons essayer de présenter une comparaison entre deux méthodes existante dans la littérature résolvant le problème d'optimisation d'une fonction

linéaire sur l'ensemble efficient d'un problème multi-objectif linéaire discret, les méthodes prises en considération étaient celles de Jorge [28] et de Chergui et Ouail [12] présentées dans le chapitre précédent. Les résultats obtenus ont montré que la méthode de Chergui et Ouail [12] est meilleure, par rapport au temps d'exécution que celle de Jorge [28].

Conclusion Général

Les problèmes d'optimisation multi-objectif sont de plus en plus étudiés car ils ont un grand intérêt dans l'industrie, où plusieurs objectifs concurrents doivent être satisfaits. Ils sont la plupart du temps NP-difficiles même si leur version mono-objectif ne l'est pas.

Contrairement aux problèmes d'optimisation mono-objectif, il existe généralement plusieurs solutions Pareto optimales ou efficaces en optimisation multi-objectif, c'est-à-dire des solutions qu'on ne peut améliorer sur un objectif sans les détériorer sur un autre. Comme il n'existe aucune solution meilleure en tout point qu'une autre, un compromis différent selon les personnes doit être choisi.

Le choix est donc subjectif, et il est indispensable de proposer l'ensemble des choix possibles afin de ne pas exclure une possibilité. L'optimisation multi-objectif est donc avant tout un outil d'aide à la décision, et c'est une personne qui prendra la décision finale.

Dans cette étude, nous nous sommes intéressés aux problèmes de programmation multi-objectif en nombres entiers. L'intérêt de tels problèmes résulte du fait que dans de nombreuses situations réelles modélisables par la programmation mathématique, les variables de décision ne peuvent prendre que des valeurs entières.

Le décideur doit réaliser une analyse de l'ensemble des solutions efficaces du problème pour sélectionner les solutions efficaces préférées, et cela s'avère impossible lorsque cet ensemble est large. Dans le cas où les préférences du décideur sont modélisées, explicitement, par une fonction, alors le problème du choix des solutions efficient peut être énoncé sous forme d'un problème d'optimisation de cette fonction sur l'ensemble des solutions efficaces. Ce type de problèmes est très difficile à résoudre vu que son ensemble d'admissibilité n'est pas convexe.

Nous avons présenté une méthode qui permet de résoudre ce type de problème et ceci a été possible grâce au principe Branch & Bound renforcé par une coupe efficace et un test permettant ainsi de réduire l'espace de recherche.

L'étude expérimentale a montré que notre méthode donne des résultats satisfaisante. Une comparaison avec une méthode existante dans la littérature, qui est la méthode de Jorge [28] a été faite et les résultats montrent que la méthode que nous proposons est

meilleure que celle proposée par Jorge en considérant les mêmes conditions de programmation.

Bibliographie

- [1] M. Abbas and D. Chaabane, An algorithm for solving multiple objective integer linear programming problem, *RAIRO Operations Research* 36, pp. 351-364, 2002.
- [2] M. Abbas and M. Moulai, Solving multiple objective integer linear programming, *Journal of the Italian Operations Research Society (Ricerca Operativa)* 29, pp. 15- 38, 1999.
- [3] Abbas M. and Chaabane D. Optimizing a linear function over an integer efficient set, *European Journal of Operational Research* 174, n^02 ; 1140-1161, 2006.
- [4] Benson H.P. Optimization over the efficient set, *Journal of Mathematical Analysis and Applications* 98, 562-580, 1984.
- [5] Benson H.P. An All-Linear Programming Relaxation Algorithm for Optimizing over the Efficient Set, *Journal of Global Optimization*, Vol.1, 83-104, 1991.
- [6] Benson H.P. A finite, Nonadjacent Extreme point Search Algorithm over the efficient Set, *Journal of Optimization Theory and Applications*, Vol.73, 47-64, 1992.
- [7] Benson H.P. A bisection-extreme point search algorithm for optimizing over the efficient set in the linear dependence case, *Journal of Global Optimization* 3, 95-111, 1993.
- [8] Benson H.P. and Sayin S. Optimizing over the Efficient Set : Four Special Cases, *Journal of optimization Theory and Applications*, Vol.80, n^01 ; 3 i 17 :, 1994.
- [9] Chaabane D. and Pirlot M. A method for optimizing over the integer efficient set, *Journal of industrial and management optimization*, Vol.6, N_04 ; 811i823; 2010 :
- [10] D. Chaabane and F. Mebrek, Optimization of a linear function over the set of stochastic efficient solutions, Springer Verlag, Berlin Heidelberg, 2012.
- [11] Charnes A. and Cooper W.W. Management models and industrial applications of linear programming, John Wiley, New York, 1961.

- [12] M.E Chergui et F Z Ouail, "Optimisation vectirielle discrète en avenir certain", Thèse de doctorat en mathématiques, option recherche opérationnelle, USTHB, Algerie 2018.
- [13] Collette Y. et Siarry P. Optimisation multiobjectif, Editions Eyrolles, 2002.
- [14] H.W. Corley, A New Scalar Equivalence for Pareto Optimization, IEEE Transactions on Automatic Control 25, No-4, 829-830, 1980.
- [15] A. Crema and J. Sylva, A method for finding the set of nondominated vectors for multiple objective integer linear programs, European Journal of Operational Research, in press, 2003.
- [16] Dantzig G.B. Linear programming and extensions, Princeton University Press, Princeton, 1963.
- [17] Deb K. Multi-Objective Optimization using Evolutionary Algorithms, New-York : John Wiley, 2001.
- [18] Dhaenens-Flipo C. Optimisation Combinatoire Multi-Objectif : Apport des Méthodes Coopératives et Contribution à l'Extraction de Connaissances, thèse d'Habilitation à diriger des Recherches de l'U.S.T.L, Lille, 2005.
- [19] J.G. Ecker and I.A. Kouada, Finding Efficient Points for Linear Multiple Objective Programs, Mathematical Programming 8, No-8, 375-377, 1975.
- [20] Ecker J.G. and Song J.H. Optimizing a linear function over an efficient set, Journal of Optimization Theory and Applications : Vol.83, n°3; 541 i 563 ; 1994.
- [21] Ehrgott M. and Gandibleux X. A survey and annotated bibliography of multiobjective combinatorial optimization, OR Spektrum, 22 :425-460, 2000.
- [22] Ehrgott M. Multicriteria Optimization, Lecture Notes in Economics and Mathematical Systems, numéro 491, Springer Verlag, Berlin, Heidelberg, 2000.
- [23] Fülöp J. 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.
- [24] Geoffrion A.M. Proper Efficiency and the Theory of Vector Maximization, Journal of Mathematical Analysis and Applications 22, No-3, 618-630, 1968.
- [25] Gomory R.E. Outline of an Algorithm for Integer Solutions to Linear Programs, Bulletin of the AMS 64, 275-278, 1958.
- [26] R. Gupta and R. Malhotra, Multi-criteria integer linear programming problem, Cahiers de CERO 34, (1992).

- [27] Isermann H. Steuer R.E. Computational Experience Concerning Payoff Tables and Minimum Criterion Values over the Efficient Set, *European Journal of Operational Research*. Vol.33, 91-97, 1987.
- [28] Jorge J.M. An algorithm for optimizing a linear function over an integer efficient set, *European Journal of Operational Research* 195, 98-103, 2009.
- [29] D. Klein and E. Hannan, An algorithm for multiple objective integer linear programming problem, *European Journal of Operational Research* 9, pp. 378-385, 1982.
- [30] T.C. Koopmans, Analysis and Production as an Efficient Combination of Activities, *Activity Analysis of Production and Allocation*, Yale University Press, New Haven, London, 33-97, 1971 (originally published in 1951).
- [31] H.W. Kuhn and A.W. Tucker, Nonlinear Programming, *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, Berkeley, Los Angeles, 481-492, 1951.
- [32] Miettinen K. Nonlinear Multiobjective Optimization, Kluwer Academic Publishers, Boston, 1999.
- [33] Nguyen N.C. An Algorithm for Optimizing a Linear Function over the Integer Efficient Set, *Konrad-Zuse-Zentrum für Informationstechnik Berlin*, 1992.
- [34] Philip J. Algorithms for the vector maximization problem, *Mathematical programming* 2, 207-229, 1972.
- [35] Roy B. et Bouyssou D. Aide multicritère à la décision : méthodes et cas, éditions Economica, 1993.
- [36] Roy B. Paradigms and challenges, *Multiple Criteria Decision Analysis- State of the Art Surveys*, Springer, 3-24, 2005.
- [37] Sayin S. Optimizing over the efficient set using a topdown search of faces, *Operations Research* 48, 6572, 2000.
- [38] Stanisław Walukiewicz. Integer programming. Mathematics and its applications (Springer-Science+Business Media, B.V.). Varsovie, 1990.)
- [39] R.E. Steuer, Multiple Criteria Optimization : Theory, Computation and Applications, John Wiley and Sons, New-York, 1985.
- [40] Ulungu E.L. and Teghem J. Multi-objective combinatorial optimization : a survey, *Journal of Multi-Criteria Decision Analysis*, 3, 83-104, 1994.
- [41] Yamamoto Y. Optimization over the Efficient Set : Overview, *Journal of Global Optimization*. Vol.22, n°1 i 4 ; 285-317, 2002.

Abstract : Optimizing a linear function over the efficient set of a multiobjective integer linear programming (MOILP) problem is a topic of unquestionable practical as well as mathematical interest within the field of multiple criteria decision making. As known, those problems are particularly difficult to deal with due to the discrete nature of the efficient set, which is not explicitly known, nor a suitable implicit description is available. In this memory an exact algorithms are presented to optimize a linear function over the efficient set of a MOILP.

Résumé : L'optimisation d'une fonction linéaire sur l'ensemble efficace d'un problème multi-objectif linéaire sur un entier (MOILP) est un sujet d'intérêt pratique et mathématique incontestable dans le domaine de la prise de décision à critères multiples. Comme on le sait, ces problèmes sont particulièrement difficiles à traiter en raison de la nature discrète de l'ensemble efficace, qui n'est pas explicitement connue, ni d'un système implicite approprié. Dans ce mémoire nous présentons, des algorithmes exact pour optimiser une fonction linéaire par rapport à l'ensemble efficace d'un MOILP.