

N° d'ORDRE :28/2009 – M/MT

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



MÉMOIRE

Présenté pour l'obtention du diplôme de MAGISTER

En : MATHEMATIQUES

Spécialité : Recherche Opérationnelle : Mathématique de Gestion

Par : BRAHMI Boualem

Thème

**Branch & Bound pour l'Optimisation d'une Fonction
Linéaire sur l'Ensemble de Solutions Efficaces**

Soutenu le : 29/06/2009 devant le jury composé de :

<i>M^r</i>	Mustapha	MOULAI	Professeur	U.S.T.H.B Alger	Président
<i>M^r</i>	Djamal	CHAABANE	Maître de Conférences	U.S.T.H.B Alger	Directeur de Thèse
<i>M^{me}</i>	Fatiha	MERAZKA	Maître de Conférences	U.S.T.H.B Alger	Examinatrice
<i>M^r</i>	Sadek	BOUROUBI	Maître de Conférences	U.S.T.H.B Alger	Examinateur

Table des matières

Introduction générale	6
1 Généralités	9
1.1 Rappel de quelques définitions	9
1.1.1 Ensembles et fonctions convexes	9
1.1.1.1 Ensembles convexes	9
1.1.1.2 Fonctions convexe	10
1.1.2 Minima locaux et globaux	10
1.2 Dualité	11
1.2.1 Lien primal-dual	12
1.2.2 Comparaison des solutions	12
1.3 Optimisation mathématique	12
1.3.1 Optimisation des fonctions non linéaires sous contraintes	12
1.3.1.1 Condition nécessaires d’optimalité dans le cas des contraintes quelconques	12
1.3.1.2 Optimisation des fonctions non linéaires sous contraintes non linéaires de type égalité	12
1.3.1.3 Optimisation des fonctions non linéaires sous des contraintes non linéaires, de type inégalité	14
1.3.2 Optimisation des fonctions non linéaires sous des contraintes non linéaires, de type mixte	17
1.3.2.1 Condition nécessaire d’optimalité de premier ordre	17

1.3.2.2	Condition suffisantes d'optimalité de second ordre	18
1.3.3	Optimisation des fonctions non linéaires sous des contraintes li- néaires, de type égalité	18
1.3.3.1	Condition nécessaires d'optimalité de premier ordre	18
1.4	Problématique	19
2	Optimisation multicritère	22
2.1	Concepts d'optimalité	23
2.1.1	Conditions d'existence des solutions optimales	24
2.1.2	Programmation linéaire multicritère	26
2.1.3	Méthodes de résolution	27
2.2	Détection graphique de l'efficacité	27
2.3	Caractérisation des solutions efficaces continues	30
3	L'Optimisation sur l'Ensemble des Solutions Efficaces	31
3.1	Cas continu	31
3.1.1	Caractérisation de solutions efficaces par convexe inverse	32
3.1.1.1	Représentation de convexe inverse	32
3.2	Cas discret	34
3.2.1	Formulation du problème	35
3.2.2	L'approche ε - contrainte	36
3.2.2.1	Dualité en programmation linéaire et ces applications pour MOLP	37
3.3	La dualité en programmation en nombre entier	38
3.3.1	L'approche fonction composée pour un programme multi-objectifs en nombre entier	39
4	La Résolution par l'Algorithme de Décomposition Branch and Bound	41
4.1	Terminologie et description générale	41
4.2	L'algorithme général	42
4.2.1	Illustration de la méthode sur quelque problèmes	42
4.2.2	Algorithme Branch and bound	46
4.2.3	Subdivision du simplexe	47
4.2.4	La borne inférieure	49
4.2.4.1	Estimation de la borne inférieure	50
4.2.5	Bornes supérieures	52

TABLE DES MATIÈRES

4.3	L'algorithme de Branch and Bound pour l'optimisation sur l'ensemble de solutions efficaces d'un problème linéaire bi-critère	52
4.3.1	Algorithme de Branch and bound	56
4.3.2	Exemples didactique	59
4.3.3	Convergence et finitude	64
4.4	Complexité	65
5	Expérimentations et résultats	67
5.1	Programme implémentant l'algorithme	67
5.1.1	Génération aléatoire des exemples tests	68
5.2	Résultats et interprétation	68
5.2.1	Tests sur des exemples aléatoires de petite taille	69
5.2.2	Tests sur des exemples aléatoires à matrices creuses et non creuses de taille moyenne	72
5.2.3	Tests sur des exemples aléatoires de taille moyenne	74
	Conclusion générale & perspectives	77
	Bibliographie	84
	Annexes	85
	Annexes A	86
	Annexes B	87
	Annexes C	88
	Annexes D	89

Table des figures

4.1	Séparation et évaluation pour le problème d'affectation	44
4.2	Evaluation de la solution optimale	45
4.3	A simplexe standard dans \mathcal{R}^3	49
4.4	Construction de segments dans l'iteration 1	59
4.5	Espace de décision pour l'exemple didactique 1	60
4.6	Espace des critères pour l'exemple didactique 1	61
4.7	L'iteration 1 du l'algorithme pour l'exemple 2	62
4.8	L'iteration 2 du l'algorithme pour l'exemple 2	63
4.9	L'iteration 3 du l'algorithme pour l'exemple 2	63
5.1	La courbe représentatif de (<i>Itmoy</i>) en fonction de n et $5 < m < 1$	70
5.2	La courbe représentatif de (<i>CPUmoy</i>) en fonction de n et $5 < m < 1$	71
5.3	La courbe représentatif de (<i>Itmoy</i>) en fonction de n	73
5.4	La courbe représentatif de (<i>CPUmoy</i>) en fonction de n	73
5.5	La courbe représentatif de (<i>Itmoy</i>) en fonction de m et $n = 40, 50$ et 60	74
5.6	La courbe représentatif de (<i>CPUmoy</i>) en fonction de m et $n = 40, 50$ et 60	75
5.7	La courbe représentatif de (<i>Itmoy</i>) en fonction de m et $n = 140, 160$ et 180	76
5.8	La courbe représentatif de (<i>CPUmoy</i>) en fonction de m et $n = 140, 160$ et 180	76
5.9	Etapes d'initialisation de l'algorithme 1	86
5.10	Organigramme itératif de l'algorithme 1	87
5.11	Etapes d'initialisation de l'algorithme 2	88
5.12	Organigramme itératif de l'algorithme 2	89

Liste des tableaux

4.1	Résumé de l'exécution de l'exemple deux	64
5.1	Problèmes à $2 \leq m \leq 4$ et $2 \leq n \leq 10$	70
5.2	Problèmes à des matrices creuses et non creuses	72
5.3	Problèmes à taille moyenne	74
5.4	Problèmes $100 < n < 200$	75

Introduction générale

Le monde est écrit en langage *mathématique* (Galilie) et nous pouvons ajouter que tous les problèmes de la vie courante, du plus simple au plus complexe se posent comme un problème d'optimisation. Ces problèmes sont incontournables. A cause, entre autres, de la concurrence qui n'a eu de cesse de se développer entre les différents acteurs du marché, les chercheurs ne purent pas se contenter bien longtemps de solutions uniquement vouées à satisfaire les contraintes. Pour ces raisons, le développement des modèles théoriques et des techniques traitant *des problèmes d'optimisation* peuvent être partagé en deux classes à savoir, la classe de problèmes convexes, et la classe de problèmes non convexes. Cette classification a poussé Rokhafollar à dire "*the great watershed in optimization isn't between linearity or non linearity but between convexity and non convexity*"[Bou07] , dont elle a connu une accélération spectaculaire, particulièrement après la deuxième guerre mondiale.

Durant cette période, les ingénieurs se sont trouvés face à des problèmes à la taille et à la complexité croissante, ce qui fut une motivation pour la recherche de méthodes de résolution fiables et systématiques. La plupart d'entre elles reposent sur un socle solide de résultats théoriques établissant les conditions pour leur convergence vers une solution optimale recherchée.

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

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

Généralement, les méthodes d'optimisation *multicritère* sont regroupées en deux catégories selon la nature du problème multi objectifs :

1. l'analyse multicritère est utilisée lorsque l'ensemble fini de solutions possibles est fourni *explicitement* ;
2. la programmation mathématique multicritère (ou optimisation multi-objectifs), s'impose lorsque l'ensemble des solutions de cardinal souvent infini est défini *implicitement* par la satisfaction de contraintes.

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

Ce présent travail est réparti comme suit :

En partant de quelques généralités, nous allons décrire un rappel de définitions et quelques notions importantes auxquelles nous ferons appel par la suite, ainsi que quelques propriétés.

Le deuxième chapitre comprend un rappel théorique sur les différentes notions de l'optimisation linéaire multicritère. Ainsi la caractérisation de l'ensemble des points efficaces d'un problème multi-objectifs linéaire, par un convexe inverse des contraintes dans l'espace des critères, pour le cas continue (MOLP) et l'introduction de la dualité et de la méthode ε -contrainte pour les programmes multi-objectives en nombre entier(MOIP), ont été présentés.

Dans le troisième chapitre nous introduisons la résolution du problème d'optimisation d'une fonction sur l'ensemble des solutions efficaces dans l'espace des critères par une méthode de décomposition Branch and Bound, ainsi que sa convergence.

Le quatrième chapitre a pour objectif, la mise en oeuvre de la méthode de décomposition Branch and Bound pour la résolution d'un problème d'optimisation d'une fonction linéaire sur l'ensemble des solutions efficaces d'un problème linéaire bi-critères et com-

prend les différents résultats obtenus et leurs interprétations. Nous terminerons par une conclusion générale et quelques perspectives.

1.1 Rappel de quelques définitions

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

1.1.1 Ensembles et fonctions convexes

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

1.1.1.1 Ensembles convexes

Définition 1 (*Ensemble convexe*) Soit l'ensemble $\mathcal{C} \subseteq \mathbf{R}^n$. \mathcal{C} est convexe si

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

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

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

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

1.1.1.2 Fonctions convexe

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

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

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

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

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

Propriété 1 Si C_1, \dots, C_r sont des ensembles convexes, alors l'intersection $\bigcap_{i=1}^r C_i$ est convexe.

1.1.2 Minima locaux et globaux

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

$$\min_{x \in X} f(x). \tag{1.1}$$

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

Définition 7 (minimum local). Intuitivement, un vecteur $x^* \in X$ est un minimum local de f sur X s'il a un coût plus faible que celui de ses voisins. Formellement, x^* est un minimum local de f sur X si

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

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

Le minimum local est strict si

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

Définition 8 (minimum global). Un vecteur $x^* \in X$ est un minimum global de f sur X s'il a un coût plus faible que celui de tous les autres vecteur dans X . Formellement, x^* est un minimum local de f sur X si

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

Le minimum global est stricte si

$$f(x^*) < f(x) \quad \forall x \in X.$$

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

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

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

1.2 Dualité

Définition 9 Soit un problème de programmation linéaire (appelé primal) noté P_r sous sa forme suivante :

$$\text{Max} \quad z = c^T x$$

$$Ax \leq b \quad \text{avec } c(1, n); A(m, n); x(n, 1); b(m, 1) \text{ alors le problème dual est :}$$

$$x \geq 0$$

$$\text{Min} \quad \omega = b^T y$$

$$yA \geq c^T \quad \text{avec } y(m, 1).$$

$$y \geq 0$$

1.2.1 Lien primal-dual

1.2.2 Comparaison des solutions

On prend P_r en forme canonique et D son dual et on considère x et y deux solutions quelconques respectivement du primal et du dual. Comme $x \geq 0$ et $y \geq 0$ on peut multiplier les contraintes du primal par y et du dual par x pour obtenir : $c^T x \leq y^T Ax \leq yb$, Celui est valable pour toutes solutions x du primal, y du dual. Donc l'optimum du primal est forcément inférieur ou égal à celui du dual.

Théorème 2 [SY06] *si x^* et y^* sont des solutions optimales alors l'optimum est le même.*

1.3 Optimisation mathématique

Cette partie a été consacré à la présentation des notions générales de la théorie de l'optimisation mathématiques sous contraintes, particulièrement les contraintes linéaires, dont la référence est inspiré du livre[SY06]

1.3.1 Optimisation des fonctions non linéaires sous contraintes

1.3.1.1 Condition nécessaires d'optimalité dans le cas des contraintes quelconques

considérons le problème (1.1)

Théorème 3 *soit $f : \mathbf{R}^n \rightarrow \mathbf{R}$ une fonction non linéaire de classe C^1 . Si $x^* \in X$ est un point minimum local (ou global) du problème (1.1), alors pour toute direction admissible $d \in \mathbf{R}^n$ en x^* , on a :*

$$[\nabla f(x^*)]^T d \geq 0.$$

où $\nabla f(x^*)$ désigne le gradient de f au point x^*

1.3.1.2 Optimisation des fonctions non linéaires sous contraintes non linéaires de type égalité

Considérons le problème

$$\begin{cases} \min f(x) \\ g(x) = 0 \end{cases} \quad (1.2)$$

où $f(\cdot) : \mathbf{R}^n \rightarrow \mathbf{R}$, $g(\cdot) : \mathbf{R}^n \rightarrow \mathbf{R}^m$ et $g(x) = \begin{pmatrix} g_1(x) \\ g_2(x) \\ \vdots \\ g_m(x) \end{pmatrix}$.

La fonction de lagrange est définie comme suit :

$$L(x, \lambda) = f(x) + \lambda^T g(x) = f(x) + \sum_{i=1}^m \lambda_i g_i(x).$$

Théorème 4 *soit x^* une solution optimale du problème (1.2) et supposons que les vecteurs $\nabla g_i(x^*)$, $i = 1, \dots, m$ sont linéairement indépendants, alors il existe un vecteur multiplicateur de lagrange λ^* tel que (x^*, λ^*) vérifie la condition de stationnarité de la fonction de lagrange c'est à dire*

$$\frac{\partial L(x^*, \lambda^*)}{\partial x} = 0,$$

$$\frac{\partial L(x^*, \lambda^*)}{\partial \lambda} = 0.$$

Définition 10 *Un point x^* est appelé point stationnaire du problème (1.2), s'il existe un vecteur $\lambda^* \in \mathcal{R}^m$ tel que la paire (x^*, λ^*) soit un point stationnaire de la fonction de Lagrange associée, c'est-à-dire*

$$(L1) \begin{cases} \frac{\partial L(x^*, \lambda^*)}{\partial x} = 0, \\ \frac{\partial L(x^*, \lambda^*)}{\partial \lambda} = 0. \end{cases} \iff \nabla L(x^*, \lambda^*) = 0$$

La recherche des points stationnaires du problème (1.2) revient à résoudre le système (L1).

Remarque Un point minimum de la fonction de lagrange n'est pas nécessairement un point minimum du problème(1.2).

Condition nécessaire d'optimalité de second ordre

Considérons le problème (1.2), où on suppose que les fonctions f et g sont de classe C^2 .

Théorème 5 *Soit x^* un point minimum local du problème (1.2). Supposons que les vecteurs $\nabla g_i(x^*)$, $i = 1, \dots, m$ sont linéairement indépendants et soit λ^* le vecteur multiplicateur de lagrange correspondant. Alors*

$$y^T \frac{\partial^2 L(x^*, \lambda^*)}{\partial x^2} y \geq 0,$$

pour tout y appartenant à l'hyperplan $H = \{y \in \mathbf{R}^n : y^T \nabla g_i(x^*), i = 1, \dots, m\}$.

Conditions suffisantes d'optimalité

Considérons le problème (1.2), où l'on suppose que les fonctions f et g sont de classe C^2 .

Théorème 6 *Pour qu'un point stationnaire x^* du problème (1.2) soit un point minimum local du problème (1.2), il suffit que le point x^* et le vecteur multiplicateur de Lagrange correspondant vérifient la relation*

$$y^T \frac{\partial^2 L(x^*, \lambda^*)}{\partial x^2} y > 0$$

pour tout $y \neq 0$ appartenant à l'hyperplan $H = \{y \in \mathbf{R}^n : y^T \frac{\partial g_i(x^*)}{\partial x} = 0, i = 1, 2, \dots, m\}$.

1.3.1.3 Optimisation des fonctions non linéaires sous des contraintes non linéaires, de type inégalité

Considérons le problème

$$\begin{cases} \min f(x) \\ g(x) \leq 0, \\ x \in D \end{cases} \quad (1.3)$$

où $D \subset \mathbf{R}^n$ est un ouvert non vide de \mathbf{R}^n , $f(\cdot) : \mathbf{R}^n \rightarrow \mathbf{R}$ et $g(\cdot) : \mathbf{R}^n \rightarrow \mathbf{R}^m$ et

$$g(x) = \begin{pmatrix} g_1(x) \\ \cdot \\ \cdot \\ \cdot \\ g_m(x) \end{pmatrix}.$$

Condition nécessaire d'optimalité de premier ordre

Définition 11 – *Tout point $x \in D$ vérifiant $g_i(x) \leq 0 \quad \forall i = 1, 2, \dots, m$ est appelé solution réalisable du problème (1.3).*

- La contrainte $g_i(x) \leq 0$ est dite active au point \tilde{x} si $g_i(\tilde{x}) = 0$. Elle est dite passive si $g_i(\tilde{x}) < 0$.
- On note $I_a(\tilde{x}) = \{i \in I = \{1, 2, \dots, m\} / g_i(\tilde{x}) = 0\}$: l'ensemble des contraintes actives au point \tilde{x} .

Théorème 7 supposons que

- 1 f est une fonction différentiable au point $x^* \in \text{int}(D)$,
- 2 les fonctions $g_i(\cdot), i \in I \setminus I_a(x^*)$, sont continues au point x^* ,
- 3 les $g_i(\cdot), i \in I_a(x^*)$, sont différentiable au point x^* ,
- 4 x^* est une solution optimale locale (ou globale) du problème (1.3).

Alors, il n'existe pas de vecteur $d \in \mathbf{R}^n$ tel que

$$d^T \nabla f(x^*) < 0$$

et

$$d^T \nabla g_i(x^*) < 0, \quad i = 1, \dots, m, \quad i \in I_a(x^*). \quad (1.4)$$

Définition 12 Un vecteur $d \in \mathbf{R}^n$ est une direction admissible au point x^* par rapport à la contrainte $g_i(x) \leq 0$, si

$$\begin{cases} d^T \frac{\partial g_i(x^*)}{\partial x} < 0 & \text{si } g_i(x^*) = 0; \\ d \text{ est quelconque} & \text{si } g_i(x^*) \neq 0 \end{cases}$$

Définition 13 Un vecteur $d \in \mathbf{R}^n$ est appelé direction admissible au point x^* par rapport aux contraintes $g_i(x) \leq 0, \forall i = 1, 2, \dots, m$, s'il est une direction admissible par rapport à chacune des contraintes $g_i(x) \leq 0, i = 1, 2, \dots, m$, au point x^* .

Pour trouver les directions admissibles du problème (1.3) au point $x^* \in D$, il suffit de résoudre le système linéaire (1.4).

Théorème 8 Soit x^* un point minimum local du problème (1.3) tel que les vecteurs

$$\frac{\partial g_i(x^*)}{\partial x}, \quad i \in I_a(x^*)$$

soient linéairement indépendants. Il existe alors un unique vecteur multiplicateur de Lagrange λ^* vérifiant les relations suivantes :

- a. $\lambda^* \geq 0$,
- b. $\frac{\partial L(x^*, \lambda^*)}{\partial x} = 0$,
- c. $(\lambda^*)^T g(x^*) = 0$ ($\lambda_i^* g_i(x^*) = 0, \forall i = 1, 2, \dots, m$.)

Condition nécessaire d'optimalité de second ordre

Définition 14 On dit que $x^* \in X = \{x \in D / g(x) \leq 0\}$ est ordinaire, si les vecteurs

$$\nabla g_i(x^*), \quad i \in I_a(x^*)$$

sont linéairement indépendants.

Définition 15 Une contrainte $g_i(x^*) \leq 0$, active en la solution ordinaire x^* , est dite forte (resp. faible), si $\lambda_i^* > 0$ (resp. $\lambda_i^* = 0$).

On note

$I_a^+(x^*) = \{i \in I, g_i(x^*) = 0, \lambda_i^* > 0\}$ l'ensemble des indices des contraintes fortes au point x^* , et

$I_a^0(x^*) = \{i \in I, g_i(x^*) = 0, \lambda_i^* = 0\}$ l'ensemble des contraintes faibles au point x^* .

Supposons que les fonction f et g sont de classe C^2 .

Théorème 9 Soit x^* une solution optimale ordinaire du problème (1.3) et $\lambda^* \geq 0$ le vecteur multiplicateur de lagrange correspondant. Alors,

$$y^T \frac{\partial^2 L(x^*, \lambda^*)}{\partial x^2} y \geq 0$$

$$\forall y \in H = \left\{ y \in \mathcal{R}^n : \begin{cases} y^T \frac{\partial g_i(x^*)}{\partial x} = 0 & \forall i \in I_a^+(x^*); \\ y^T \frac{\partial g_i(x^*)}{\partial x} \leq 0 & \forall i \in I_a^0(x^*) \end{cases} \right\}$$

Condition suffisantes d'optimalité

Définition 16 Une solution réalisable x^* du problème (1.3) est dite pseudo stationnaire, s'il existe un m -vecteur λ^* tel que

$$\begin{aligned}\frac{\partial L(x^*, \lambda^*)}{\partial x} &= 0, \\ \lambda^* &\geq 0, \\ \lambda_i^* g_i(x^*) &= 0, \quad \forall i = 1, \dots, m.\end{aligned}$$

Théorème 10 *Pour qu'une solution réalisable, pseudo-stationnaire, x^* soit localement optimale pour le problème (1.3), il suffit que*

$$y^T \frac{\partial g_i(x^*)}{\partial x} y > 0,$$

$$\forall y \in H = \left\{ y \in \mathcal{R}^n : \begin{cases} y^T \frac{\partial g_i(x^*)}{\partial x} = 0 & \forall i \in I_a^+(x^*); \\ y^T \frac{\partial g_i(x^*)}{\partial x} \leq 0 & \forall i \in I_a^0(x^*) \end{cases} \right\}, \|y\| \neq 0$$

1.3.2 Optimisation des fonctions non linéaires sous des contraintes non linéaires, de type mixte

Considérons le problème

$$\begin{cases} \min & f(x) \\ g(x) & \leq 0, \\ h(x) & = 0 \end{cases} \quad (1.5)$$

où $f(\cdot) : \mathbf{R}^n \rightarrow \mathbf{R}$, $g(\cdot) : \mathbf{R}^n \rightarrow \mathbf{R}^m$ et $h(\cdot) : \mathbf{R}^n \rightarrow \mathbf{R}^k$.

1.3.2.1 Condition nécessaire d'optimalité de premier ordre

Théorème 11 *Supposons que $f, g, h \in C^1$ et x^* une solution réalisable, optimale locale du problème (1.5) telle que les vecteurs*

$$\nabla g_i(x^*), i \in I_a(x^*),$$

$$\nabla g_j(x^*), j = 1, \dots, k.$$

soient linéairement indépendants. Alors, il existe des vecteurs multiplicateurs de Lagrange $\lambda^ \in \mathbf{R}^m, \mu \in \mathbf{R}^k$ tels que*

$$\left\{ \begin{array}{l} \nabla f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla g_i(x^*) + \sum_{j=1}^k \mu_j^* \nabla h_j(x^*) = 0 \\ \lambda_i^* g_i(x^*) = 0, \forall i = 1, \dots, m, \\ h(x^*) = 0, \\ \lambda^* \geq 0. \end{array} \right. \quad (1.6)$$

1.3.2.2 Condition suffisantes d'optimalité de second ordre

Théorème 12 *Supposons que $f, g, h \in C^2$ et pour la solution réalisable x^* du problème(1.5), il existe $\lambda^* \in \mathbf{R}^k$ tels que les relations (1.6) soient vérifiées. Si, de plus, pour tout $y \in \mathbf{R}^n (\|y\| \neq 0)$ tel que*

$$\begin{aligned} y^T \frac{\partial g_i(x^*)}{\partial x} &= 0, \quad \forall i \in I_a^+(x^*), \\ y^T \frac{\partial g_i(x^*)}{\partial x} &\leq 0, \quad \forall i \in I_a^0(x^*), \\ y^T \frac{\partial h_j(x^*)}{\partial x} &= 0, \quad \forall j = 1, \dots, k, \end{aligned}$$

on a

$$y^T \frac{\partial L(x^*, \lambda^*, \mu^*)}{\partial x^2} y > 0;$$

alors, x^* est un point minimum local du problème (1.5).

1.3.3 Optimisation des fonctions non linéaires sous des contraintes linéaires, de type égalité

1.3.3.1 Condition nécessaires d'optimalité de premier ordre

Considérons le problème

$$\begin{aligned} \min \quad & f(x) \\ \text{Ax} - b &= 0 \end{aligned} \quad (1.7)$$

où A est une $m \times n$ -matrice, $b \in \mathbf{R}^m$. Posons

$$A = (a_1 \ a_2 \ \dots \ a_n) = \begin{pmatrix} A_1^T \\ A_2^T \\ \cdot \\ \cdot \\ A_m^T \end{pmatrix}$$

avec a_i : vecteur colonne et A_i^T : vecteur ligne.

$$\text{Posons } g(x) = \begin{pmatrix} g_1(x) \\ g_2(x) \\ \cdot \\ \cdot \\ g_m(x) \end{pmatrix}, g_i(x) = A_i^T x - b_i \quad \forall i = 1, \dots, m.$$

Alors l'ensemble des solutions réalisables de (1.7) s'écrit :

$$X = \{x \in \mathbf{R}^n : g(x) = 0\} = \{x \in \mathbf{R}^n : g_i(x) = A_i^T x - b_i = 0, \forall i = 1, \dots, m\}.$$

Pour que $X \neq \emptyset$ et X ne soit pas constitué d'un point isolé, on pose $\text{Rang}(A) = m < n$

Proposition 1 Soit $x \in X$. Un vecteur $d \in \mathbf{R}^n$ est une direction admissible en x , si et seulement si $Ad = 0$.

De plus, pour un vecteur $d \in \mathbf{R}^n$ vérifiant $Ad = 0$, on aura $x(\alpha) = x + \alpha d \in X, \forall \alpha \in \mathbf{R}$.

On définit la fonction de Lagrange $L(x, \lambda)$ pour le problème (1.7) par

$$L(x, \lambda) = f(x) + \lambda^T g(x) + \sum_{i=1}^m \lambda_i g_i(x),$$

$$\text{où } x \in \mathbf{R}^n \text{ et } \lambda = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \cdot \\ \cdot \\ \lambda_m \end{pmatrix} \in \mathbf{R}^m.$$

1.4 Problématique

Soit un problème linéaire multi-objectifs suivant

$$\max_{x \in X} c_i^T x \quad i = 1, \dots, p. \tag{1.8}$$

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

$$X = \{x \in \mathbf{R}^n : Ax \leq b, x \geq 0\}. \tag{1.9}$$

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

On note par C la $n \times p$ matrice, et X l'ensemble réalisable de problème (1.8) c'est-à-dire

$$\min\{Cx : x \in E_X\}. \quad (1.10)$$

Depuis les années 1970, des auteurs se sont intéressés à l'optimisation d'une fonction sur l'ensemble des solutions efficaces d'un problème de programmation linéaire multi-objectifs de type (1.8)(MOLP). L'importance et la motivation de ce problème (1.10), ont été discutées en détail dans la littérature (par exemple Ecker et Kouada [EK78, EK75]). En particulier, Bensen [Ben79b, Ben82b, Ben78, Ben98, HB02], prouve que dans quelques problèmes de modélisation impliquant des objectifs multiples, les modèles d'optimisation sur l'ensemble des solutions efficaces(E_X) sont plus réalistes et appropriées que les programmes linéaires multi-objectifs habituels (MOLP).

En outre, la résolution de ces problèmes d'optimisation, avec l'ensemble efficace défini *implicitement*, évite les difficultés informatiques *d'énumérer* tous les points extrêmes efficaces.

La difficulté du problème est principalement due à la *non convexité* de l'ensemble des solutions efficaces. Les algorithmes existants pour résoudre ce problème peuvent être classés en plusieurs groupes selon leur principe de fonctionnement ; citons

- Les algorithmes de recherche de sommets adjacents [ES94].
- Les algorithmes de recherche de sommets non adjacents [NTBK05, Ben78].
- Les algorithmes basés sur la méthode de séparation et évaluation [FM00, MD05],
- Les algorithmes basés sur la méthode de relaxation Lagrangienne.
- Les algorithmes basés sur la méthode duale et de bisection.

Le problème a été étudié la première fois en 1972 par Philip [Phi72], qui décrit un algorithme basé sur le déplacement à travers les sommets efficaces adjacents dans le cas où le critère principal est une fonction linéaire ; et un bon nombre de chercheurs ont suivi cette voie. Plus tard, Isermann et Steur ont décrit une procédure semblable pour la résolution de (1.10) où la fonction objectif principale est l'un des objectifs $c^i x$ dans le problème (MOLP). Ces méthodes emploient un hyperplan de coupe qui a une arête efficace adjacente rapportant une augmentation de la fonction objectif principale.

Des résultats concernant la caractérisation des solutions efficaces ont été développés, ce qui a motivé la recherche en optimisant un critère linéaire sur l'ensemble de solutions

efficaces sans les énumérées toutes.

Dans notre travail, nous nous intéressons au problème de type (1.10) mais la fonction à minimiser n'est pas forcément linéaire, mais est en général une fonction réelle et sous la supposition (1.11); où la méthode de décomposition Branch and Bound à été utilisée, en présentant l'algorithme proposé par Thoai et la description de ces étapes.

$$X \neq E_X \neq \emptyset, \quad (1.11)$$

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

$$\min\{f(x) : x \in E_X\}, \quad (1.12)$$

avec f une fonction réelle.

Notons que l'ensemble des *solutions efficaces* E_X est en général non convexe. Le problème (1.12) rentre alors dans le cadre de l'optimisation global (ou la programmation non convexe).

Quand tout minimum local du problème (1.12) est aussi un minimum global (par exemple, dans la programmation linéaire ou convexe), les méthodes de l'optimisation locales peuvent trouver l'optimum facilement. Cependant, ce n'est pas le cas dans beaucoup de problèmes. Le champ de l'optimisation globale est consacré aux problèmes de type (1.12) dont lequel peut avoir plusieurs optima locaux à part l'optimum global. Les algorithmes construits pour résoudre ces problèmes peuvent être classés dans les chemins différents. Une classification habituelle distingue entre déterministe et méthodes stochastiques. Les méthodes stochastiques (par exemple, la recherche tabou, algorithme génétique ou recuit simulé) appliquent quelque randomfactors peuvent être piégé dans les optima locaux. Cependant, ils ne trouvent pas l'optimum global nécessairement. Dans les méthodes du déterministe (tel l'optimisation Lipschitzian ou Branche and bound) aucuns aléatoires facteurs ne sont inclus. Quelques-uns convergent à l'optimum global sous quelques conditions, mais quand l'algorithme est arrêté après un nombre fini des itérations que l'exactitude de la solution ne peut pas être sue avec la précision. Nous nous concentrerons sur une méthode de la dernière classe, dans particulier de Branch and bound méthodes qui est utilisée parmi les méthodes complètes et rigoureuses pour les problèmes du non convexe et non linéaire le plus communément.

Optimisation multicritère

La plupart des problèmes d'optimisation réels requièrent l'optimisation de plusieurs critères (ou objectifs) souvent contradictoires devront être optimisés (maximisés ou minimisés) simultanément. Alors que, pour les problèmes n'incluant qu'un seul objectif l'optimum cherché est clairement défini, celui d'un problème multicritère est mal défini dans la mesure, où il n'existe aucune décision qui maximise (minimise) toutes les fonctions objectif simultanément vu l'aspect conflictuel de ces dernières. Il existe alors différents concepts d'optimalité, selon lesquels est examiné une partie de ces décisions vérifiant certaines propriétés. Les décisions optimales retenues sont généralement un assortiment de décisions, qui se distinguent par différents compromis réalisés entre les objectifs.

Problème multicritère :

De manière générale, un problème d'optimisation multicritère s'écrit sous la forme

$$\langle X, f(X) \rangle, X \subset \mathbf{R}^n \text{ et } f(x) \subset \mathbf{R}^p \quad (2.1)$$

X est dit ensemble des décisions (actions, scénarios)

$$\begin{aligned} f : X \subset \mathbf{R}^n &\rightarrow \mathbf{R}^p \\ x \in X &\mapsto f(x) \end{aligned}$$

où $f(x) = \begin{pmatrix} f_1(x) \\ \cdot \\ \cdot \\ \cdot \\ f_p \end{pmatrix}$
 avec $f_i : X \in \mathbf{R}^n \rightarrow \mathbf{R} \quad i = 1, \dots, p$ sont les fonctions objectifs (fonctions coût, critères).

2.1 Concepts d'optimalité

En l'absence d'une décision optimale dans le problème multicritère (2.1) qui maximise toutes les fonctions objectifs, il est nécessaire pour identifier les meilleurs compromis de définir une relation d'ordre entre ces éléments. Dans le cas des problèmes d'optimisation multicritère, ces relations d'ordre sont appelées relations de dominance, elles constituent un concept crucial en optimisation multicritère appelé optimalité. Il existe cependant plusieurs concepts d'optimalité qui ont été présentés dans la littérature : l'optimalité au sens de Pareto, la cône-optimalité et l'optimalité au sens de Slater [Ste86], que nous allons définir, ainsi que l'optimalité au sens de Geoffrion [Ste86]. Pour définir clairement et formellement ces notions d'optimalité, les relations usuelles $=, <, \leq$ sont étendues aux cas des vecteurs.

Notations

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

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

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

Optimalité de Pareto

Une décision de Pareto est optimale, dans le sens où il n'existe aucune autre solution dans l'ensemble des décisions, capable d'améliorer la valeur d'un critère sans en détériorer au moins la valeur d'un autre. De manière formelle, nous avons la définition suivante :

Définition 17 Une décision $x^* \in X$ est dite maximale de Pareto (décision efficace) du

problème (2.1) s'il n'existe pas une autre décision x qui vérifie la relation : $f(x) \geq f(x^*)$

Optimalité de Slater

Plusieurs articles concernant l'optimisation multicritère portent sur l'optimalité de Pareto. Dans certains cas, un concept légèrement faible est utilisé. Il est appelé optimalité de Slater. Une décision est optimale de Slater dans le sens où il n'existe aucune autre décision dans l'ensemble des décisions, capable d'améliorer tous les critères simultanément.

Définition 18 Une décision $x^* \in X$ est dite *décision maximale de Slater (faiblement efficace)* dans le problème (2.1), s'il n'existe pas une autre décision $x \in X$ telle que $f(x) > f(x^*)$

Optimalité de Geoffrion

Un autre concept légèrement faible et renforcé par l'élimination des différences infinies entre les objectifs est appelé optimalité de Geoffrion. Il est défini comme suit :

Définition 19 Une décision $x^* \in X$ est dite *décision maximale de Geoffrion (proprement efficace)* pour le problème (2.1), si :

- x^* est une décision maximale de Pareto.
- $\exists M > 0$ tel que $\forall x \in X, \forall i \in \{1, \dots, p\}, f_i(x) > f_i(x^*)$ et pour un certain $j \in \{1, \dots, N\}, j \neq i$ tel que $f_j(x) < f_j(x^*)$, on a :

$$\frac{f_i(x) - f_i(x^*)}{f_j(x^*) - f_j(x)} < M.$$

Si on note X^P, X^S et X^G les ensembles de décisions maximales respectivement de Pareto, de Slater et de Geoffrion, les inclusions suivantes sont alors vérifiées,

$$X^G \subseteq X^P \subseteq X^S.$$

2.1.1 Conditions d'existence des solutions optimales

Conditions d'existence des décisions maximales de Pareto On définit la fonction

$$\begin{aligned} \phi : f(X) &\rightarrow R \\ f(x) &\mapsto \phi(f(x)); \end{aligned} \tag{2.2}$$

où $f(X) = \{f(x) = (f_1(x), f_2(x), \dots, f_p(x)) \in \mathbf{R}^p, x \in X\}$.

Définition 20 La fonction ϕ est strictement croissante (respectivement croissante) par rapport à la relation de préférence \geq sur l'ensemble $f(X)$, si $\forall x^1, x^2 \in X$ $f(x^1) \geq f(x^2) \Rightarrow \phi(f(x^1)) > \phi(f(x^2))$ (respectivement $\phi(f(x^1)) \geq \phi(f(x^2))$).

Définition 21 La fonction ϕ est strictement croissante (respectivement croissante) par rapport à la relation de préférence $>$ sur l'ensemble $f(X)$, si $\forall x^1, x^2 \in X f(x^1) > f(x^2) \Rightarrow \phi(f(x^1)) > \phi(f(x^2))$ (respectivement $\phi(f(x^1)) \geq \phi(f(x^2))$).

Théorème 13 [PH86]. Soit ϕ une fonction croissante par rapport à la relation de préférence \geq sur $f(X)$. Alors, pour qu'une décision $\bar{x} \in X$ soit maximale de Pareto dans le problème (2.1), il est suffisant que l'une des conditions suivantes soit vérifiée :

1. La fonction ϕ est strictement croissante par rapport à la relation de préférence \geq sur $f(X)$ et $A\bar{x} \in X$ maximise la fonction ϕ , c-à-d :

$$\phi(f_1(\bar{x}), f_2(\bar{x}), \dots, f_N(\bar{x})) = \max_{x \in X} \phi(f_1(x), f_2(x), \dots, f_p(x)), \quad (2.3)$$

2. $f(\bar{x}) \in X$ est un point maximum unique de la fonction ϕ sur $f(X)$.

En définissant la fonction

$$\phi(f_1(x), f_2(x), \dots, f_p(x)) = \left[\sum_{i=1}^p \alpha_i [f_i(x)]^k \right]^{\frac{1}{k}} ;$$

On a la conséquence suivante :

Conséquence 1 Supposons que les conditions suivantes sont vérifiées :

$$f_i(x) \geq 0 \quad \forall i = 1, \dots, p \quad \forall x \in X, \quad (2.4)$$

$$\alpha_i > 0, \forall i = 1, \dots, p \quad \text{et} \quad k > 0. \quad (2.5)$$

Alors, une décision $\bar{x} \in X$ vérifiant la relation

$$\left[\sum_{i=1}^p \alpha_i [f_i(\bar{x})]^k \right]^{\frac{1}{k}} = \max_{x \in X} \left[\sum_{i=1}^p \alpha_i [f_i(x)]^k \right]^{\frac{1}{k}} \quad (2.6)$$

est maximal de Pareto dans le problème (2.1).

Remarque : très souvent, dans les problèmes multicritère (2.1), on utilise la condition suffisante (2.6) avec $k = 1$, c-à-d :

$$\sum_{i=1}^p \alpha_i f_i(\bar{x}) = \max_{x \in X} \sum_{i=1}^p \alpha_i f_i(x) \quad (2.7)$$

Dans ce cas, la condition (2.4) n'est pas exigée.

Conditions d'existence des décisions maximales de Slater

Théorème 14 [YZ75]. Si la fonction ϕ définie dans (2.2) est strictement croissante par rapport à la relation de préférence $>$ sur $f(X)$, alors toute décision $\bar{x} \in X$ vérifiant la relation (2.3) est une décision maximale de Slater dans le problème multicritère (2.1).

Conditions d'existence des décisions maximales de Geoffrion Notons par $\langle \cdot, \cdot \rangle$ le produit cartésien.

Théorème 15 [YZ75]. Une décision $\bar{x} \in X$ est une décision maximale de Geoffrion dans le problème (2.1), si et seulement, si il existe un composé fini de vecteurs $\{\alpha_1, \alpha_2, \dots, \alpha_p\} \subseteq \Delta_N, p \leq N$, tel que pour tout $x \in X$ il existe un $j \in \{1, 2, \dots, p\}$ tel que

$$\langle \alpha^j, f(\bar{x}) \rangle \geq \langle \alpha^j, f(x) \rangle,$$

Où Δ_N est $(N - 1)$ -simplexe de \mathcal{R}^N .

On déduit du théorème 15, pour $p = 1$ la conséquence suivante :

Conséquence 2 Si $\alpha_i > 0, \forall i = 1, \dots, p$, alors toute décision $\bar{x} \in X$ vérifiant

$$\sum_{i=1}^p \alpha_i f_i(\bar{x}) \geq \sum_{i=1}^p \alpha_i f_i(x)$$

est maximale de Geoffrion dans le problème multicritère (2.1)

2.1.2 Programmation linéaire multicritère

Définition 22 Les programmes linéaires multicritère appartiennent à une classe de problèmes d'optimisation où les critères sont des fonctions linéaires, à optimiser sur un polyèdre convexe et fermé.

Définition 23 Soit X un polyèdre et $x \in X$

- x est un point extrême s'il n'existe pas un $y, z \in X \setminus \{x\}$ tel que $x = (y + z)/2$.
- x est un point extrême de la région faisable si, et seulement si, x est une solution faisable de base.

Position du problème

Considérons le problème linéaire multicritère suivant :

$$PLM : \{(C_1^T x, C_2^T x, \dots, C_p^T x) \rightarrow \max \quad Ax = b, \quad x \geq 0 \quad (2.8)$$

Où, $C_i \in \mathcal{R}^n, i = 1, \dots, p, x \in \mathcal{R}^n, A$ est une $m \times n$ -matrice, $b \in \mathcal{R}^m$.

Notons X_r l'ensemble des solutions réalisables du problème multicritère (2.8)

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

et notons

$$\dot{\Delta}_p = \left\{ \lambda = (\lambda_1, \lambda_2, \dots, \lambda_p) \in \mathbf{R}^p, \lambda_i \geq 0, \forall i = \overline{1, p}, \sum_{i=1}^p \lambda_i = 1 \right\}$$

et C est la $q \times n$ -matrice constituée des vecteurs lignes $C_i^T, i = 1, \dots, p$.

2.1.3 Méthodes de résolution

Il existe de nombreuses méthodes pour la résolution d'un problème linéaire multicritère, dont certaines utilisent l'algorithme du simplexe modifié pour prendre en compte le fait que le problème à étudier est à objectifs multiples. La méthode du simplexe multicritère notamment la méthode de Zeleny [YZ75], peut être utilisée pour générer une représentation exacte de l'ensemble des solutions efficaces. Ceci est réalisé en se déplaçant d'un point extrême efficace à un autre point extrême efficace adjacent, jusqu'à trouver tous les points extrêmes efficaces, puis générer le polytope joignant les solutions extrêmes efficaces. Cependant, il existe plusieurs tests qui permettent d'affirmer ou non l'efficacité d'un point extrême [PH86]. Dans tous les cas, les méthodes de résolution d'un programme linéaire multicritère comprennent quatre phases :

- Trouver une solution réalisable de base,
- Trouver une solution réalisable extrême efficace,
- Générer l'ensemble des solutions réalisables extrêmes efficaces,
- Générer l'ensemble des solutions réalisables efficaces.

Théorème 16 [Ste86] *Une solution réalisable $x^* \in X_r$ est une solution maximale de Pareto du problème linéaire multicritère (2.8), si et seulement si il existe un $\lambda \in \dot{\Delta}_p$ tel que x^* est une solution optimale de $P(\lambda)$, où $P(\lambda)$ est défini par :*

$$P(\lambda) \begin{cases} \max_{x \in X} C(\lambda)^T x & = & \sum_{i=1}^p \lambda_i (C_i)^T x \\ Ax & = & b \\ x & \geq & 0 \end{cases} \quad (2.9)$$

2.2 Détection graphique de l'efficacité

En considérant que l'idée de dominance se réfère aux vecteurs de l'espace critère, l'idée de l'efficacité se réfère aux points de l'espace de décision. Un point \bar{x} est efficace si son vecteur critère n'est pas dominé par un vecteur critère d'un autre point de S (l'ensemble de solutions réalisables). C'est-à-dire, d'un point efficace, il n'est pas possible de se déplacer

d'une façon réalisable afin d'augmenter un des objectifs sans diminuer au moins un des autres objectifs. Dans cet ordre d'idée, quelques concepts sont utilisés pour la détection graphique de l'efficacité. Pour tester l'efficacité d'un point donné \bar{x} , le concept d'ensemble dominant est introduit par le biais de quelques définitions.

Définition 24 (Cône.) Soit $V \in \mathbf{R}^n, V \neq \emptyset$. V est un cône si et seulement si $\alpha v \in V$ pour tout scalaire $\alpha > 0$ et tout $v \in V$. Le vecteur $\mathbf{0}$, vecteur de \mathbf{R}^n , est contenu dans chaque cône.

Un cône convexe est un cône qui est ensemble également convexe.

Définition 25 (Cône polaire.) Soit $V \in \mathbf{R}^n$ un cône. Le cône polaire non négatif de V (noté V^\geq) est le cône convexe $V^\geq = \{y \in \mathbf{R}^n | y'v \geq 0, \forall v \in V\}$.

Ce qui signifie que tous les vecteurs de V^\geq font un angle inférieur ou égale 90° avec chaque vecteur de V . Nous définissons les générateurs de cône par

Définition 26 (Vecteurs générateurs d'un cône). Soit un ensemble de vecteurs $\{v^1, \dots, v^p\}$ de \mathbf{R}^n et l'ensemble V tel que : $V = \{v \in \mathbf{R}^n | v = \sum_{i=1}^p \alpha_i v^i = 1, \alpha_i \geq 0 \forall i = 1 \dots p\}$. V est l'ensemble de toutes les combinaisons linéaires à coefficients non négatifs des $\alpha_i, i = 1, \dots, p$ et le cône convexe engendré par l'ensemble $\{v^1, \dots, v^p\}$. La dimension d'un cône $V \in \mathbf{R}^n$ est donnée par le nombre de vecteurs linéairement indépendants dans V .

Définition 27 (Cône semi-polaire positif) Soit $V \subset \mathbf{R}^n$ un cône convexe généré par $\{V^1, \dots, V^p\}$. Alors le cône semi-polaire positif (noté $V^>$) est le cône convexe $V^> = \{y \in \mathbf{R}^n | y'v^i \geq 0, \text{ pour tout } i \text{ et } y'v^i > 0, \text{ pour au moins un } i\} \cup \{0_{\mathbf{R}^n}\}$

Définition 28 (Ensemble de dominance). Soit $\bar{x} \in S$ et $C^>$ le cône semi-polaire positif généré par le gradient de p fonctions objectif ie

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

l'ensemble de dominance de \bar{x} est donné par

$$D_{\bar{x}} = \{\bar{x}\} \oplus C^>$$

c-à-d que

$$D_{\bar{x}} = \{x \in \mathbf{R}^n | x = \bar{x} + y, y \in C^>\}$$

Remarque : L'opération \oplus est définie par $A, B \in \mathbf{R}^n$

$$A \oplus B = \{\alpha \in \mathbf{R}^n | \alpha = a + b, a \in A, b \in B\}$$

Théorème 17 Soit $D_{\bar{x}}$ l'ensemble de dominance de $\bar{x} \in S$. Alors, \bar{x} est efficace si et seulement si $D_{\bar{x}} \cap S = \{\bar{x}\}$

Définition 29 (Point idéal) Le point idéal \bar{Z} est le point de \mathbf{R}^p de coordonnées

$$\bar{Z}_k = \max_{x \in S} (Z_k(x)) \quad k = 1, \dots, p.$$

Définition 30 (matrice des gains). Soit \hat{x}^j une solution optimale du critère Z_j une matrice $(p \times p)$ formée des éléments de $Z_{kj} = Z_k(\hat{x}^j)$ est dite matrice des gains

$$\begin{pmatrix} \bar{Z}_{11} & \cdot & \cdot & \cdot & Z_{1j} & \cdot & \cdot & \cdot & Z_{1p} \\ \cdot & & & & & & & & \cdot \\ \cdot & & & & & & & & \cdot \\ \cdot & & & & & & & & \cdot \\ Z_{k1} & \cdot & \cdot & \cdot & \bar{Z}_{kj} & \cdot & \cdot & \cdot & Z_{kp} \\ \cdot & & & & & & & & \cdot \\ \cdot & & & & & & & & \cdot \\ \cdot & & & & & & & & \cdot \\ Z_{p1} & \cdot & \cdot & \cdot & Z_{pj} & \cdot & \cdot & \cdot & \bar{Z}_p \end{pmatrix}$$

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

Définition 31 (Point anti-idéal) Le point anti-idéal \underline{Z} est le point de \mathbf{R}^p , de coordonnées

$$\underline{Z}_k = \min_{x \in S} (Z_k(x)) \quad k = 1, \dots, p.$$

Définition 32 (Point nadir). Le point nadir η de \mathbf{R}^p de coordonnées

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

où Z_{kj} est un élément de la matrice des gains, est dit point nadir.

Définition 33 (Face). Le sous-ensemble non vide du polyèdre S est une face s'il existe un polyèdre L tel que $F = L \cap S$ et S est entièrement situé dans un demi-espace fermé délimité par L , mais non inclus à L

2.3 Caractérisation des solutions efficaces continues

Il peut arriver dans la programmation mathématique multicritère que l'ensemble des solutions efficaces, qui est une partie de l'ensemble des solutions admissibles, soit très vaste et parfois même infini comme dans le cas continu. Dans une telle situation, il est souvent impossible d'énumérer toutes les solutions efficaces et même lorsque c'est possible, il est nécessaire d'aider le décideur à faire son choix parmi les solutions efficaces. La sélection d'une solution efficace spécifique comme candidat de meilleur compromis nécessite donc une certaine connaissance de la structure de préférence du décideur. Les méthodes pour aider un décideur à faire son choix parmi les solutions efficaces peuvent être classées par catégorie selon la façon dont le décideur articule ou incorpore ses préférences. Chaque modèle de préférence est généralement explicité en fixant un certain nombre de paramètres. Dans le chapitre qui suit, nous décrivons les caractérisations de l'ensemble de solutions efficaces, en utilisant la dualité dans l'espace des critères pour un problème MOLP¹ ou bien pour le cas discret MOILP².

¹Programme linéaire multi-objectifs.

²Programmes linéaire multi-objectifs en nombre entier.

L'Optimisation sur l'Ensemble des Solutions Efficaces

3.1 Cas continu

Dans ce chapitre, nous reformulons un problème équivalent au problème (1.12) du premier chapitre page 21 comme un programme inversé convexe dans \mathbf{R}^{n+p} , dont le reverse convexe des contraintes est défini par une fonction concave dans l'espace des critères \mathbf{R}^p . L'idée principale est basée sur les résultats donnés dans la référence [RHZ07], ainsi présentés comme suit :

Théorème 18 ([RHZ07]) *Soit X un sous ensemble convexe non vide de \mathbf{R}^n . Si les fonctions f_i sont strictement convexes pour tout $i = \{1, 2, \dots, p\}$ alors les assertions suivantes sont équivalentes :*

- i-** \bar{x} est un point efficace de problème MOLP.
- ii-** Il existe $\lambda \in \mathbf{R}^p$ tel que \bar{x} soit une solution optimale du problème (P_λ) suivant :

$$\min \left[f_\lambda(x) = \sum_{i=1}^p \lambda_i f_i(x) : x \in X \right]$$

où $\lambda_i \geq 0$, $i = \overline{1, p}$ et $\sum_{i=1}^p \lambda_i = 1$.

3.1.1 Caractérisation de solutions efficaces par convexe inverse

Dans cette partie, on introduit les principaux résultats qui correspondent à la caractérisation des solutions optimales d'un programme linéaire avec l'addition de problème de convexe inversé des contraintes.

3.1.1.1 Représentation de convexe inverse

Problème convexe inverse (RCP) Soient f, g deux fonctions convexes de \mathbf{R}^n dans \mathbf{R} alors le problème (RCP) s'écrit sous cette forme :

$$\begin{aligned} \min \quad & f(x) \\ \text{s.c} \quad & g(x) \geq 0 \\ & x \in X \subset \mathbf{R}^n \end{aligned}$$

La contrainte $g(x) \geq 0$ appelée contrainte de convexe inverse.

Proposition 2 [RHZ07] Sous la supposition (1.11) page 21 l'ensemble polyèdre $\Omega \in \mathbf{R}^{n+m}$ défini par

$$\omega = \{(\alpha, \beta) \mid A^T \alpha - C^T \beta \geq C^T \lambda, \alpha \geq 0, \beta \geq 0\} \quad (3.1)$$

est non vide qui a un sommet pour chaque vecteur positif $\lambda \in \mathbf{R}^p$, A est la matrice $m \times n$ des contraintes et $C(n \times 1)$ vecteur critère.

Preuve :

Sous la supposition (1.11), soit y une solution efficace de problème (1.8), alors de la définition d'une solution efficace et du théorème 18 chaque vecteur positif λ , le point y est une solution optimale du problème

$$\max \{ \lambda^T Cx : x \in X, Cx \geq Cy \}. \quad (3.2)$$

Ce dernier est équivalent au problème suivant :

$$\max \{ \lambda^T Z : Ax \leq b, -Z \leq -cy, -Cx + Z = 0, x \geq 0 \}. \quad (3.3)$$

Le dual de ce problème (dans $\alpha \in \mathbf{R}^m, \beta \in \mathbf{R}^p$ et $\delta \in \mathbf{R}^p$) est comme suit :

$$\min \{ b^T \alpha - (Cy)^T \beta : A^T \alpha - c^T \delta \geq 0, -\beta + \delta = \lambda, \alpha \geq 0, \beta \geq 0 \}. \quad (3.4)$$

Et qui est équivalent au problème suivant

$$\min \{ b^T \alpha - (Cy)^T \beta : A^T \alpha - c^T \beta \geq c^T \lambda, \alpha \geq 0, \beta \geq 0 \}. \quad (3.5)$$

Du théorème de la dualité forte en programmation linéaire, alors les problèmes (3.4) et (3.5) ont au moins une solution optimale donc l'ensemble Ω est non vide. On se basons sur la proposition 2, $\forall \lambda > 0, \forall z \in \mathbf{R}^p$, on construit le programme linéaire suivant :

$$\bar{g}(z) = \min\{b^T \alpha - z^T \beta : A^T \alpha - C^T \beta \leq c^T \lambda, 0 \leq \alpha \leq \bar{\alpha}, 0 \leq \beta \leq \bar{\beta}\} \quad (3.6)$$

Avec $\bar{\alpha} \in \mathbf{R}^m$ et $\bar{\beta} \in \mathbf{R}^p$

soit la fonction g défini comme suit

$$\begin{aligned} g & : \mathbf{R}^p \rightarrow \mathbf{R} \\ z & \mapsto g(z) = \bar{g}(z) - \lambda^T z \end{aligned} \quad (3.7)$$

Proposition 3 Soit $V(\Omega)$ l'ensemble des sommets du polyèdre Ω défini en (3.1).

Supposons que les vecteurs $\bar{\alpha}$ et $\bar{\beta}$ sont choisis d'une manière à satisfaire les conditions :

$$V(\Omega) \subset \{(\alpha, \beta) : 0 \leq \alpha \leq \bar{\alpha}, 0 \leq \beta \leq \bar{\beta}\} \quad (3.8)$$

Alors la fonction g définie en (3.7), a les propriétés suivantes :

- (a) g est une fonction concave dans \mathbf{R}^p .
- (b) $g(z)$ est définie pour chaque $z \in \mathbf{R}^p$.
- (c) $g(z) \geq 0, \forall z \in \{z \in \mathbf{R}^p : z \leq cx, x \in X\}$.
- (d) y est une solution efficace du problème (1.8) ssi $y \in X$ et $g(cy) = 0$

Preuve :

- (a) : $\bar{g}(z)$ est le minimum de $-\beta^T z + b^T \alpha$, par conséquent elle est concave, donc $g(z)$ est concave.
- (b) : de la proposition 2 et la propriété 3.8, il suit que pour chaque $z \in \mathbf{R}^p, \bar{g}(z)$ défini par (3.6) est fini donc g est fini.
- (c) : (c)-1 : D'une part si $z = Cx, x \in X$ et du problème (3.6) $\bar{g}(z)$ est égal à la valeur optimal de problème (3.5), où bien au problème équivalent (3.4), et sachant que le dual de (3.4) et le problème (3.3), ce qui implique que $g(z) = \bar{g}(z) - \lambda^T z \geq 0$.
- (c)-2 : $z \leq Cx, x \in X$, alors il existe $\hat{z} = C\hat{x}, \hat{x} \in X$ tel que $\hat{z} \geq z$. donc de (3.6) et (3.7) et $\alpha \geq 0, \beta \geq 0$ il suit que $g(z) \geq g(\hat{z}) \geq 0$
- (d) : il est évident que y est une solution efficace du problème (1.8) ssi (y, z) avec $z = Cy$ est la solution optimal du problème (3.3) ie $\bar{g}(z) = \lambda^T z$ où $g(z) = \bar{g}(z) - \lambda^T z = 0$.

On utilisons la fonction g définie en (3.7) et ses propriétés montrées dans la proposition (3) nous construisons à partir problème (1.12) le problème suivant :

$$\min\{f(x) : x \in X, \quad Cx - z = 0, \quad g(z) \leq 0\}. \quad (3.9)$$

L'équivalence de problème (3.7) et (3.9) sont affirmés comme suit :

Proposition 4 *Le point x^* est une solution optimale du problème (1.12) ssi le point $(x^*, z^*) \in \mathbf{R}^{n+p}$ avec $z^* = cx^*$ soit une solution optimal du problème (3.9).*

3.2 Cas discret

Les problèmes multi objectifs en nombre entier joue un rôle central dans le champ d'optimisation multicritère. En plus avec leur pertinence pour les applications pratiques, ils sont théoriquement intéressant dû à leur rapport intrinsèque à la théorie classique de la programmation linéaire mono objectif en nombre entier. La scalarisation des fonction objectif ne peuvent pas être utilisés pour produire toutes les solutions non dominées (supportées), là où existe des solutions non dominées (non supportées), qui sont dominées par la combinaisons convexe fractionnaire d'autres solutions; souvent les solutions non supportées occupent une part très importante dans le processus de décision donc elles ne peuvent être ignorées.

Les dernières décennies, plusieurs méthodes ont été développées pour résoudre des problèmes MOILP, dans le sens de trouver l'ensemble des solutions efficaces entières, ou au moins quelque partie considérable (Teghem et Kunsch). Les techniques utilisées sont d'une nature diverse, y compris programmation dynamique (Villareal et Karwan, 1981), branche and bound (Marcotte et Soland, 1986)[FS06],[MD05], méthodes séquentielles (Klein et Hannan, 1982,; Sylva et Crema, 2004,; Sylva et Crema, 2007). Particulièrement dans le contexte de ce mémoire la dualité en programmation multicritère ainsi la méthode ε -contrainte introduite dans Haimès et al (1971) sont utilisées[KKZ04]. Cependant, et en dépit de tels efforts, une énumération explicite de l'ensemble effectif d'un MOILP n'est pas encore recommandée même pour les raisons pratiques dans les problèmes de taille moyenne. Comme connu, excepté les fardeaux computationnels lourds impliqués dans de tels algorithmes, la dimension habituellement considérable de l'ensemble effectif a tendance à saturer le faiseur de la décision, jusqu'au point où la sélection de sa solution préférée devient presque une mission impossible. Pour éviter ces situations, une approche sensible consiste à optimiser une fonction linéaire sur l'ensemble effectif comme une façon de mesurer l'importance, ou discrimine parmi, beaucoup des solutions effectives différentes

(Benson, 1984). L'optimisation sur l'ensemble effectif est une approche intéressante dans l'optimisation multicritère qui a été étudiée entièrement dans le cas continu par beaucoup d'auteurs (voir par exemple, Benson,[Ben82b],[Ben78],[Ben98],[HB02]; Yamamoto, 2002), suivre le travail novateur de Philip en 1972[Phi72], bien que le cas discret ait par aucuns moyens vus un développement semblable. En fait, la seule méthode proposée pour l'optimisation sur l'ensemble effectif d'un MOILP est donnée dans Abbas et Chaabane [eDC06] (2006), où les différents types de coupes sont imposés dans un tel chemin que l'amélioration de la valeur objective à chaque itération est garantie. Cependant, l'expérience computationnelle concernant cet algorithme n'est pas rapportée, donc c'est difficile d'évaluer l'applicabilité pratique de telle approche[Jor08].

La théorie de la dualité a une longue tradition et une grande variété d'application (Wolsey 1981), l'introduction de cette dernière et la méthode ε -contrainte nous permettent la caractérisation de l'ensemble des solutions efficaces des programmes linéaire multi-objectifs en nombre entier (MOILP).

3.2.1 Formulation du problème

On considère le programme multicritère(MOP) suivant

$$\begin{aligned}
 & \max \quad \{Z_1 = f_1(x)\} \\
 & \quad \cdot \\
 & \quad \cdot \\
 & \quad \cdot \\
 & \max \quad \{Z_k = f_k(x)\} \\
 & \text{s.t} \quad x \in S
 \end{aligned} \tag{3.10}$$

avec $S \in \mathcal{R}^n$ l'ensemble réalisable, et $f_i(x)$, $i = 1, \dots, k$ des fonctions reals.

Le problème (3.10) est dit un problème multi objectifs linéaire si $f_i(x) = C^i x$, $\forall i = 1, \dots, k$ et

$$S = \{x \in \mathcal{R}^n : Ax \leq b, x \geq 0\} \tag{3.11}$$

et si on suppose que tout les composantes de la matrice $C := (C^i)_{i=1}^k$, la matrice A et le vecteur b sont des entiers, alors le *MOLP* peut s'écrire sous la forme.

$$\begin{aligned}
 & \max \quad Cx \\
 & \text{st} \quad Ax \leq b \\
 & \quad \quad x \geq 0.
 \end{aligned} \tag{3.12}$$

Le problème (3.10) est dit un programme linéaire multi objectifs en nombres entiers si $f_i(x) = C^i x \quad \forall i = 1, \dots, k$ et

$$S = \{x \in \mathcal{R}^n : Ax \leq b, x \geq 0 \text{ et entier}\} \quad (3.13)$$

en d'autres termes

$$\begin{aligned} \max \quad & Cx \\ \text{st} \quad & Ax \leq b \\ & x \geq 0. \text{ et entier} \end{aligned} \quad (3.14)$$

On note l'ensemble des critères réalisable Z , l'ensemble des vecteurs critères non dominés N et l'ensemble de tous les point efficaces E de(3.10) comme suit :

$$Z = \{Z \in \mathcal{R}^k : Z = f(x), x \in S\} = f(S)$$

$$N = \{z \in Z : \text{il n'existe pas } \tilde{z} \in Z \text{ tel que } \tilde{z} \geq z\}$$

$$E = \{x \in S : f(x) \in N\}$$

Et on suppose que l'ensemble Z est borné, \mathbf{R}_{\leq}^k est fermé c-a-d l'ensemble $Z + \mathbf{R}_{\leq}^k$ est fermé et l'ensembles N, E sont non vides.

3.2.2 L'approche ε - contrainte

Soit $i \in \{1, \dots, K\}$ et $e_j \in \mathbf{R}, j \in J_i = \{1, \dots, k\}/\{i\}$

Alors le programme ε - contrainte du i^{me} objectif est formulé comme suit :

$$\begin{aligned} \max \quad & f_i(x) \\ \text{st} \quad & f_j(x) \geq e_j \quad \forall j \in J_i \\ & x \in S. \end{aligned} \quad (3.15)$$

Et on suppose que la borne supérieur de $e_j, j \in J_i$ est choisie d'une manière que (3.15) est faisable et Z l'ensemble des solutions réalisable dans l'espaces des critères est supposé borné ce qui implique l'existence de la solution optimale.

Théorème 19 [TZ04] *Chaque solution optimale de (3.15) est une solution faiblement efficace de (3.10) et l'ensemble de toutes les solutions optimales de (3.15) contient au moins une solution efficace pour (3.10).*

Théorème 20 [TZ04] *Si x^* est une solution efficace de (3.10), alors il existe un indice $i \in \{1, \dots, k\}$ et une borne supérieur $e_j \in \mathbf{R}, j \in J_i$ tel que x^* est solution optimale de (3.15)*

Observons que on peut généraliser le théorème 20 pour MOP comme suit

Corollaire 1 [TZ04] *S'il existe un index i et une borne supérieur $e_j \in \mathbf{R}, j \in J_i$ sachant que x^* est une solution optimale unique de (3.15), alors x^* est efficace pour (3.10).*

3.2.2.1 Dualité en programmation linéaire et ces applications pour MOLP

On considère le programme linéaire ε -contrainte

$$\begin{aligned} \max \quad & C^i x \\ \text{st} \quad & C^j x \geq e_j, \forall j \in J_i \\ & Ax \leq b, \quad x \geq 0 \end{aligned} \quad (3.16)$$

On introduit les variables dual $u_j, j \in J_i$ et $y \in \mathbf{R}^m$ pour formuler le dual de problème (3.16) comme :

$$\begin{aligned} \min \quad & -\sum_{j \in J_i} e_j u_j + b^T y \\ \text{st} \quad & -\sum C_l^j u_j + A^T y \geq c^i \quad \forall l = 1 \dots n \\ & u_j \geq 0, \quad \forall j \in J_i, \quad y \geq 0 \end{aligned} \quad (3.17)$$

On utilise seulement les variables duaux optimaux $u_j^*, j \in J_i$ de (3.17), qui correspondent pour programme ε -contrainte de (3.16), tel qu'on peut définir le vecteur $\hat{u} \in \mathcal{R}^k$ comme

$$\hat{u}_j := \begin{cases} \hat{u}_j^* & , j \in J_i \\ 1 & j = i \end{cases}$$

Après la normalisation, on obtient $\bar{\lambda} := \frac{\hat{u}}{\|\hat{u}\|}$ avec $\hat{\lambda} \in \hat{\Lambda}$. Une solution optimale du problème (3.16) et aussi optimale pour (3.14), ce dernier est donné dans le cas de MOLP par

$$\begin{aligned} \max \quad & \bar{\lambda}^T C x \\ \text{st} \quad & Ax \leq b \\ & x \geq 0 \end{aligned} \quad (3.18)$$

Ce qui est équivalent à

$$\begin{aligned} \max \quad & \hat{u}^T C x \\ \text{st} \quad & Ax \leq b \\ & x \geq 0 \end{aligned} \quad (3.19)$$

Théorème 21 [TZ04] Soit $x^* \in S$ une solution optimale de problème(3.16). Alors de x^* , on peut construire un vecteur poids unitaire $\bar{\lambda} \in \bar{\Lambda}$ sachant que x^* soit aussi optimale pour(3.18)

et invrersement comme le confirme le théorème suivant

Théorème 22 [TZ04] Soit $x^0 \in S$ est une solution de (3.18) pour un $\bar{\lambda} \in \bar{\Lambda}$. Alors il existe un index $i \in \{1, \dots, k\}$ sachant que x^* est aussi optimal pour (3.16), avec la borne inférieure e_j est définie comme $e_j = c^j x^*$, $j \in J_i$.

Nous utiliserons plus tard que la maximisation de la fonction $\bar{\lambda}^T Cx$ dans (3.19) est équivalente à la maximisation de la fonction de l'utilité linéaire (théorèmes 21 et 22)

$$g(x) = c^i x + \sum_{j \in J_i} u^* c^j x - b^T y^* = c^i x - F^*(-C_{j_i}, b) \quad (3.20)$$

où $F^* : \mathbf{R}^{k-1+m} \rightarrow \mathbf{R}$ est la fonction linéaire, C_j la sous matrice induite de C en supprimant la i^{me} ligne et $b^T y$ est la constante qui n'a aucun impact sur la maximisation de g .

3.3 La dualité en programmation en nombre entier

Soit \mathcal{F} l'ensemble de toutes les fonctions croissante $F : \mathbf{R}^m \rightarrow \mathbf{R}$, c-à-d

$$\mathcal{F} = \{(F^* : \mathbf{R}^m \rightarrow \mathbf{R}) : F(a) \leq F(b) \forall a, b \in \mathbf{R}^m \quad a \leq b\}$$

et considérons le problème mono objectif en nombre entier

$$\begin{aligned} \max \quad & C^T x \\ \text{s.t} \quad & Ax \leq b \\ & x \geq 0 \text{ et entier} \end{aligned} \quad (3.21)$$

avec $C \in \mathbf{Z}^n$, alors son dual peut s'écrire comme suit :

$$\begin{aligned} \min \quad & F(b) \\ \text{s.t} \quad & F(Ax) \geq C^T x \\ & \forall x \geq 0 \text{ ,et entier} \\ & F \in \mathcal{F} \end{aligned} \quad (3.22)$$

Théorème 23 (Dualité faible)[TZ04] $C^T x \leq F(b)$ pour toute solutions réalisable x du primal (3.21) et toute fonction réalisable F du dual (3.22)

Théorème 24 (dualité forte)[TZ04] Si l'un des problème (3.21) ou (3.22) admet une solution optimal, alors il existe une solution optimale x^* de (3.21) et une fonction optimal dual F^* de (3.22), telle que $C^T x^* = F^*(b)$. De plus, si (3.21) est irréalizable, alors (3.22) est aussi irréalizable ou non borné et si (3.22) est irréalizable, alors (3.21) est irréalizable ou non borné.

Le programme dual en nombre entier et en particulier, les fonctions optimaux dual F^* sont généralement utilisées pour l'analyse de sensibilité

Corollaire 2 [TZ04] Soit x^*, F^* les solutions optimales respectivement pour le primal (3.21) et le dual (3.22). Si le second membre b de (3.21) est modifier et devient $b' \neq b$, la fonction F^* reste aussi une fonction réalisable duale pour le problème modifier, satisfait $Z' \leq F^*(b')$ avec Z' : la valeur optimal de l'objective de problème modifier.

3.3.1 L'approche fonction composée pour un programme multi-objectifs en nombre entier

considérons le problème (3.14) (MOIP)

$$\begin{aligned} \max \quad & Cx \\ \text{st} \quad & Ax \leq b \\ & x \geq 0 \text{ et entier.} \end{aligned} \tag{3.23}$$

En appliquant l'approche ε -contrainte pour le problème (3.14), en choisissant une fonction objectif $f(x) = c^i x, i \in \{1, \dots, k\}$ et un vecteur $e = (e_j)_{j \in J_i} \in \mathbf{R}^{k-1}$ de bornes inférieures des objectives restants. Le problème (3.15) est donc donné par

$$\begin{aligned} \max \quad & C^i x \\ \text{s.t} \quad & C_{j_i} x \geq e \\ & Ax \leq b \\ & x \geq 0 \text{ et entier.} \end{aligned} \tag{3.24}$$

Et son dual est comme suit :

$$\begin{aligned} \min \quad & F(-e, b) \\ \text{s.t} \quad & F(-C_{j_i} x, Ax) \geq c^i x \quad \forall x \geq 0 \text{ et entier} \\ & F \in \mathcal{F} \end{aligned} \tag{3.25}$$

En utilisant la fonction optimale dual F^* de (3.25) pour définir une fonction d'utilité pour (MOIP) comme une fonction composée donnée par :

$$\mathcal{G}(x) := c^i x - F^*(-C_{j_i} x, b). \quad (3.26)$$

En se basant sur les théorèmes 25, 26 le problème (3.14) se transforme comme un programme composé en nombre entier suivant

$$\begin{aligned} \max \quad & c^i x - F^*(-C_{j_i} x, b) \\ \text{s.t} \quad & Ax \leq b \\ & x \leq 0, \text{ et entier.} \end{aligned} \quad (3.27)$$

Théorème 25 [TZ04] Soit x^* une solution optimale pour (3.24) et F^* une fonction optimale pour (3.25), alors x^* est aussi optimale pour (3.27).

Théorème 26 [TZ04] x^* est une solution efficace de (3.14) ssi il existe un indice $i \in \{1, \dots, k\}$ et une fonction croissante $F^* \in F$ sachant que x^* est optimale pour (3.27).

La Résolution par l'Algorithme de Décomposition Branch and Bound

4.1 Terminologie et description générale

Les problèmes de l'optimisation discrets NP-difficile sont souvent des travaux immense qui exigent des mêmes algorithmes de l'escient, et le paradigme, B& B est l'un des outils principaux dans construction de ceux-ci. Un algorithme B&B cherche dans l'espace complet de solutions d'un problème donné pour la meilleure solution. Cependant, l'énumération explicite est normalement impossible dû au nombre exponentiel croissant de solutions potentielles. Rappelons tout d'abord le principe général de séparation et évaluation. Quand on doit minimiser une fonction f sur un domaine K , on cherche à éviter l'exploration exhaustive de K . Pour ce faire, on se donne une décomposition de K en sous-ensembles (si possible disjoints) qui correspond au terme séparation. On suppose être capable d'évaluer efficacement un minorant des valeurs de f sur chaque sous-ensemble (ce qui correspond au terme évaluation). Au lieu de faire une recherche exhaustive dans K , on commence par une région donnée, dans laquelle la recherche est conduite de façon exhaustive, ce qui conduit à un majorant pour le minimum. On compare alors ce majorant au minorant d'une région non explorée jusqu'à présent. Si le minorant est plus grand que le majorant actuel, on peut éliminer la région entière sans avoir besoin de l'explorer. Dans le cas contraire, on met à jour le majorant grâce à l'exploration exhaustive de la région. Les performances d'un algorithme exploitant la séparation et l'évaluation dépendent de

plusieurs facteurs : la vitesse de calcul des minorants dans une région, la qualité de ces minorants, l'ordre de parcours des régions, etc.

4.2 L'algorithme général

Par convenance, on représente l'exécution de la méthode de branch-and-bound à travers une arborescence. La racine de cette arborescence représente l'ensemble de toutes les solutions du problème considéré. Dans ce qui suit, nous résumons la méthode de branch-and-bound sur des problèmes de minimisation. Pour appliquer la méthode de branch-and-bound, nous devons être en possession :

1. d'un moyen de calcul d'une borne inférieure correspondant à une solution partielle
2. d'une stratégie de subdivision de l'espace de recherche pour créer des espace de recherche de plus en plus petits.
3. d'un moyen de calcul d'une borne supérieure à trouvés au moins pour une solution.

La méthode commence par considérer le problème de départ avec son ensemble de solutions, appelé la racine. Des procédures de bornes inférieures et supérieures sont appliquées à la racine. Si ces deux bornes sont égales, alors une solution optimale est trouvée, et on s'arrête là. Sinon, l'ensemble des solutions est divisée en deux ou plusieurs sous-problèmes, devenant ainsi des enfants de la racine. La méthode est ensuite appliquée récursivement à ces sous-problèmes, engendrant ainsi une arborescence. Si une solution optimale est trouvée pour un sous-problème, elle est réalisable, mais pas nécessairement optimale, pour le problème de départ. Comme elle est réalisable, elle peut être utilisée pour éliminer toute sa descendance : si la borne inférieure d'un noeud dépasse la valeur d'une solution déjà connue, alors on peut affirmer que la solution optimale globale ne peut être contenue dans le sous-ensemble de solution représenté par ce noeud. La recherche continue jusqu'à ce que tous les noeuds soient explorés ou stérilisés.

4.2.1 Illustration de la méthode sur quelque problèmes

Soient n personnes à affecter à n tâches. Le coût de l'affectation de i^{ime} personne à la tâche j est noté par C_{ij} . Le problème consiste à affecter chaque tâche à une seule personne de telle manière à minimiser le coût total . Bien entendu, une tâche ne peut être exécutée que par une seule personne.

Illustration. Soit la matrice suivante des coûts

$$C = \begin{bmatrix} \text{t\^ache1} & \text{t\^ache2} & \text{t\^ache3} & \text{t\^ache4} & \\ 9 & 2 & 7 & 8 & \text{personne a} \\ 6 & 4 & 3 & 7 & \text{personne b} \\ 5 & 8 & 1 & 8 & \text{personne c} \\ 7 & 6 & 9 & 4 & \text{personne d} \end{bmatrix}$$

Si par exemple, on décide de faire l'affectation suivante $(1, a); (2, c); (3, b)$ et $(4, d)$; on aura le coût total suivant : $9 + 8 + 3 + 4 = 24$.

Par contre, si on faisait l'affectation suivante : $(1, b); (2, c); (3, d)$ et $(4, a)$; on aura le coût total suivant $6 + 8 + 9 + 8 = 31$.

L'application de la méthode de branch-and-bound nécessite une fonction de borne inférieure.

Une des ces fonctions de borne inférieure pourrait être celle-ci : Il est clair que, quelque soit la solution, son coût ne peut pas être plus petit que la somme des plus petits éléments de chacune des lignes de la matrice des coûts. Dans notre cas, $2 + 3 + 1 + 4 = 10$.

On commence par la racine qui correspond à l'état de " pas d'élément encore choisi de la matrice des coûts". La valeur de la borne inférieure est alors égale à 10. Les noeuds au premier niveau de l'arbre correspondent au 4 jobs de la première ligne de la matrice à partir du moment que chacun d'eux est un choix potentiel de la première composante de la solution. Le plus prometteur parmi eux est le noeud 2 car ayant la plus petite borne inférieure, comme illustré par la Figure 4.1

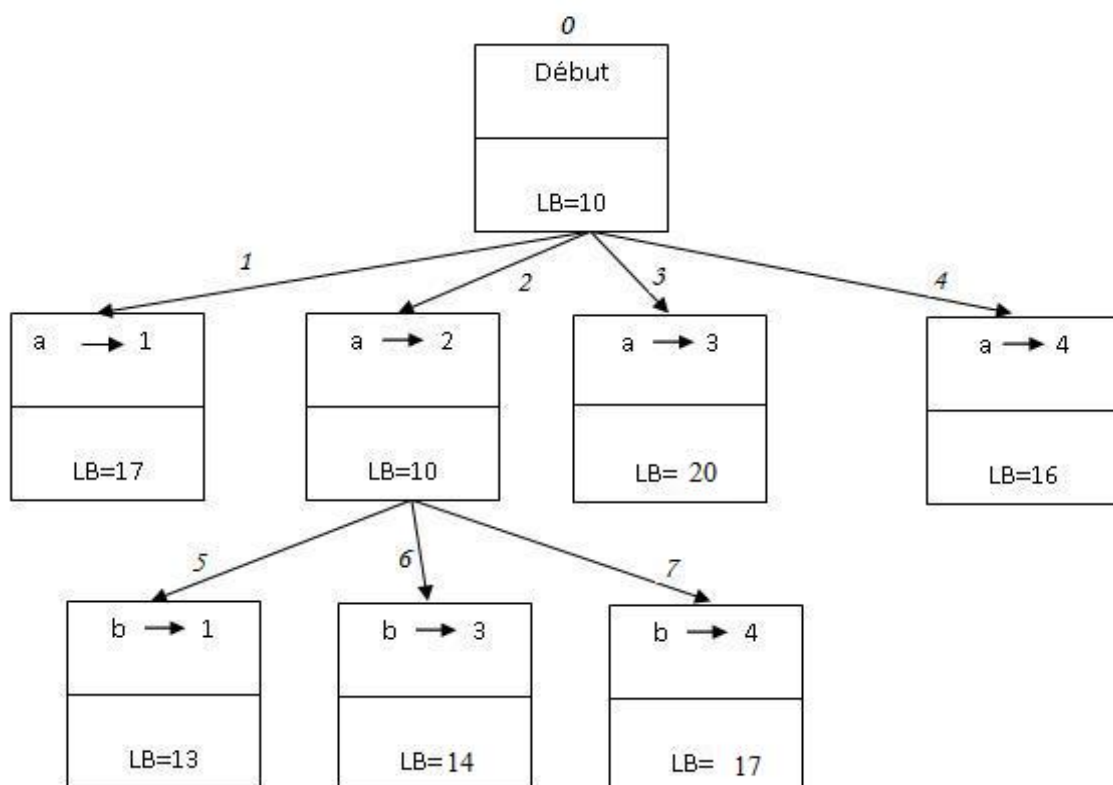


FIG. 4.1 – Séparation et évaluation pour le problème d'affectation

En continuant, on génère les noeuds comme illustrés par la Figure 4.2. jusqu'à arriver à la solution $(2, a); (1, b); (3, c) \text{ et } (4, d)$ engendrant un coût de valeur 13. Remarquons que, quand on arrive à noeud 8, on choisit le couple $(c, 3)$. Cela nous laisse obligatoirement avec le couple $(4, d)$. C'est la première solution complète qu'on génère à ce stade de calculs. À partir de là, soit on génère une solution dont le coût est supérieur à la solution courante (comme c'est le cas pour le noeud 9) soit on ignore les noeuds car ne pouvant pas contenir une solution meilleure que celle dont on dispose (comme c'est le cas des noeuds 1, 3, 4, 6 et 7). Notons que c'est par pur hasard que nous avons généré du premier coup la solution optimale.

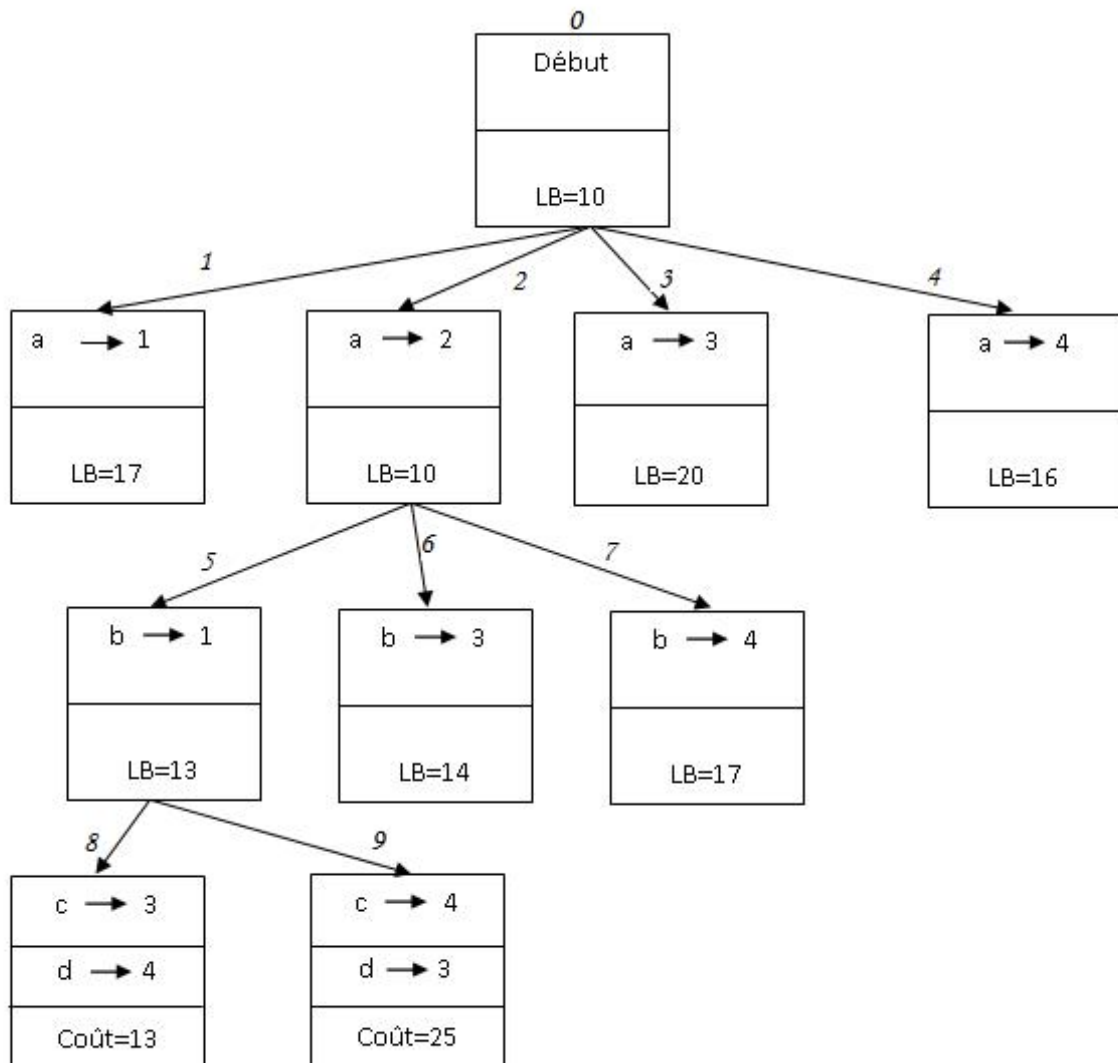


FIG. 4.2 – Evaluation de la solution optimale

4.2.2 Algorithme Branch and bound

Cet algorithme est désigné pour la résolution des programmes d'optimisation global de type

$$\min\{f_0(x) : x \in X, g_i(x) \leq 0, i = 1, \dots, m\}, \quad (4.1)$$

où g_i est une fonction concave $i = 1, \dots, m$ X est un Polytope (polyèdre borné) défini par

$$\mathcal{X} = \{x \in \mathcal{R}^n : Ax - b \leq 0\} \quad (4.2)$$

où A est une matrice $m \times n$ et b est m -vecteur.

On note par F l'ensemble des solutions réalisables de (4.1). l'algorithme de type $B\&B$ s'énonce comme suit :

Algorithme 1.

◇ **Initialisation**

1. Construire un simplexe $S^1 \supseteq \mathcal{X}$;
2. Calculer une borne inférieure $\mu(S^1)$ pour f_0 sur $S^1 \cap F$;
3. Déterminer l'ensemble $Q(S^1) \subset S^1 \cap F$ (noter que $Q(S^1)$ peut être vide) ;
4. Pour $Q^1 = Q(S^1)$, $\gamma_1 = \min\{f_0(x) : x \in Q^1\}$, où γ_1 est une borne supérieure de la valeur optimale du problème considéré ; $\gamma_1 = +\infty$ si $Q^1 = \emptyset$;
5. Choisir un point $\xi^1 \in Q^1$ telle que $f_0(\xi^1) = \gamma_1$
6. Poser $\mu^1 = \mu(S^1)$, $\mathcal{R}^1 = \{S^1\}$, $k = 1$;

◇ **Itération k**

1. – Si $\gamma_k = \mu_k$, alors arrêter, ξ^k est la solution optimale recherchée et γ^k est l'objectif optimal.
- Sinon
déviser S^k en r simplexes S_1^k, \dots, S_r^k satisfaisant :

$$\bigcup_{i=1}^r S_i^k = S^k, \text{ int}S_i^k \cap \text{int}S_j^k = \emptyset, \forall i \neq j, i, j = 1, \dots, r$$

2. Pour chaque $i = 1, \dots, r$, calculer une borne inférieure $\mu(S_i^k)$ pour f_0 sur $S_i^k \cap F$ satisfaisant $\mu(S_i^k) \geq \mu_k$, et déterminer un ensemble $Q(S_i^k) \subset S_i^k \cap F$
3. Poser $Q^{k+1} = Q^k \cup \{Q(S_i^k) : i = 1, \dots, r\}$, $\gamma_{k+1} = \min\{f_0(x) : x \in Q^{k+1}\}$.
4. Choisir $\xi^{k+1} \in Q^{k+1}$ telle que $f_0(\xi^{k+1}) = \gamma_{k+1}$.
5. Poser $\mathcal{R}^{k+1} = \mathcal{R}^k \setminus \{S^k\} \cup \{S_i^k : i = 1, \dots, r\}$.

6. Supprimer tous les $S \in \mathcal{R}^{k+1}$ telle que $\mu(S) \geq \gamma_{k+1}$ et réinitialiser l'ensemble \mathcal{R}^{k+1} , c'est à dire, $\mathcal{R}^{k+1} = \{S \in \mathcal{R}^{k+1} : \mu(S) \leq \gamma_{k+1}\}$.
7. – Si $\mathcal{R}^{k+1} = \emptyset$, poser $\mu_{k+1} = \gamma_{k+1}$.
– Sinon poser $\mu_{k+1} = \min\{\mu(S) : S \in \mathcal{R}^{k+1}\}$.
8. Choisir $S^{k+1} \in \mathcal{R}^{k+1}$ telle que $\mu(S^{k+1}) = \mu_{k+1}$.
9. Poser $k = k + 1$ et aller à (1).

Pour la compréhension et l'implémentation de cet algorithme, on doit spécifier trois opérations basiques :

- (A) Subdivisions des simplexes,
- (B) Estimation des bornes inférieures
- (C) Calcul des bornes supérieures.

4.2.3 Subdivision du simplexe

Définition 34 ***r-simplexe*** : On appelle *r-simplexe* l'enveloppe convexe de $r + 1$ points affinement indépendants dans \mathbf{R}^n . Les *r-simplexes* sont donc exactement les polytopes de dimension r avec $r + 1$ sommets.

Il s'agit de subdiviser l'espace en plusieurs sous espaces et souvent on utilise les subdivisions, en triangle ou bien rectangle utilisé par exemple dans [HT97], dans notre cas on a utilisé la subdivision du simplexe.

Dans ce qui suit, on suppose que X est un ensemble borné qu'on peut diviser en un n -simplexe dans l'espace des critères $S \subset \mathcal{R}^p$. La subdivision suivante est souvent utilisée en optimisation globale.

À chaque itération de l'algorithme $(p - 1)$ -simplexe S se divise en r $(p - 1)$ -simplexe S_1, \dots, S_r telle que :

$$\bigcup_{i=1}^r S_i = S, \text{ int}S_i \cap \text{int}S_j = \emptyset, \forall i \neq j, i, j = 1, \dots, r.$$

, De plus, pour la systématique performance de subdivision dans l'algorithme de B&B, il est exigé que le premier simplexe $S^1 \supset X$, et les simplices générés ; l'algorithme à la même structure. En utilisant la supposition suivante :

$$\exists v^0 \in \mathcal{R}^p \leq \min\{c_i^t x : x \in X\} (i = 1, \dots, p) \quad (4.3)$$

une condition suffisante pour l'existence de borne inférieure v_i pour chaque critère i est la bornétude de X .

on note par F l'ensemble faisable de problème 3.9 c-à-d

$$F = \{(x, z) \in \mathcal{R}^{n+p} : x \in X, Cx - z = 0, g(z) \leq 0\} \quad (4.4)$$

due souvent à la non convexité du polyèdre F , on construit le polyèdre convexe F^0 contenant F , en tenant compte de la proposition suivante

Proposition 5 *Les insertions suivantes sont équivalente*

- (a) $g(v^0) > 0$;
- (b) *l'ensemble faisable F est incluse dans l'ensemble F^0 tel que*

$$F^0 = \{(x, z) \in \mathcal{R}^{n+p} : x \in X, z = v^0 + \sum t_i e^i, t_i \geq 0 (i = 1, \dots, p)\}$$

sachant que $e^i (i = 1, \dots, p)$ est le vecteur unitaire de \mathcal{R}^{n+p} .

Preuve :

Un cône $\mathcal{C} \in \mathcal{R}^n$ est un ensemble fermé sous l'opération de prendre des rayons à travers ses points, c-à-d $\lambda x \in \mathcal{C}$ avec $\lambda \geq 0$ et $x \in \mathcal{C}$. Nous voyons que l'ensemble \mathcal{C} est un cône convexe ssi $\lambda_1 x_1 + \lambda_2 x_2 \in \mathcal{C}$ avec $\lambda_1; \lambda_2 \geq 0$ et $x_1; x_2 \in \mathcal{C}$. Notons que chaque sous espace linéaire est un cône convexe. Donc nous pouvons utiliser le terme cône comme un terme court pour cône convexe.

(a) Du fait que $v^0 \leq Cx, \forall x \in X$, cela suit que v^0 n'est pas un point efficace et cela implique par la proposition 3(c)-(d) du chapitre 3 que $g(v^0) > 0$.

(b) soit $(x, z) \in F$. Alors par la définition $x \in X$ et $z = cx$, de la supposition 4.3, il suit directement que $z \geq v^0$, par conséquent, z peut s'écrire comme suit

$$z = v^0 + \sum t_i e^i, t_i \geq 0, i = 1, \dots, p$$

c-à-d $(x, z) \in F^0$

l'ensemble F^0 est généré par $(p - 1)$ -simplexe standard, défini comme suit

$$S^0 = [e^1, \dots, e^p] = \{z \in \mathcal{R}^p : z \geq 0, \sum_{i=0}^p z_i = 1\} \quad (4.5)$$

Par exemple pour le cas $p = 3$ voir la figure 4.3 e_i est le vecteur unité de $\mathcal{R}^p, i = 1, \dots, p$, donc on utilise la notation $F^0 = F(S^0)$, en générale, pour chaque $(p - 1)$ -simplexe $S = [s^1, \dots, s^p] \subset S^0$, on définit l'ensemble $F(S) \subset \mathcal{R}^{n+p}$ pour

$$F(S) = \{(x, z) \in \mathcal{R}^{n+p} : x \in X, z = v^0 + \sum_{i=1}^p t_i s^i, t_i \geq 0 (i = 1, \dots, p)\} \quad (4.6)$$

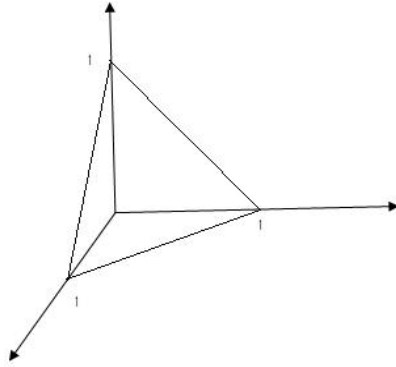


FIG. 4.3 – A simplexe standard dans \mathcal{R}^3

chaque simplexe S est divisé en plusieurs $(p - 1)$ -simplexe s_1, \dots, s_r sachant que

$$\cup_{j=1}^r s_j = S \text{ et } \text{ints}_j \cap \text{ints}_i = \emptyset \text{ pour } i \neq j$$

Alors, en conséquence, l'ensemble $F(S)$ est divisé en sous ensembles $F(s_1), \dots, F(s_r)$ tel que :

$$\cup_{j=1}^r F(s_j) = F(S) \text{ et } \text{int}F(s_j) \cap \text{int}F(s_i) = \emptyset \text{ pour } i \neq j$$

4.2.4 La borne inférieure

Pour chaque simplexe S généré, on calcul la borne inférieure U de la fonction f sur l'ensemble $F \cap F(S)$, telle que

$$U(S) \leq \min\{f(x) : (x, z) \in F \cap F(S)\}. \quad (4.7)$$

De la proposition 5(a) et comme g est concave, alors il existe au moins un point $v^i \in K_i$ tel que

$$K_i = \{z \in \mathcal{R}^p : z = \omega + t_i s^i, t_i \geq 0\}. \quad (4.8)$$

sachant que $g(v^i) = 0$, soit

$$I(s) = \{i; \exists v^i = v^0 + \omega_i s^i \text{ avec } g(v^0) = 0\} \quad (4.9)$$

Notons que pour chaque $i \in I(s)$ le nombre ω est calculé par la formule suivante :

$$\omega_i = \max\{\omega : g(v^0 + \omega s^i) \geq 0, \omega > 0\} \quad (4.10)$$

La méthode pour calculer la borne inférieure est basée sur les résultats suivantes :

Proposition 6 *La borne inférieure $U(S)$ est obtenue comme suit*

- (a) si $I(S) = \emptyset$, alors $U(S) = +\infty$;
 (b) Si $I(S) \neq \emptyset$, alors $U(S)$ est calculée par la résolution de programme suivant (avec $x \in \mathcal{R}^n$) et $t \in \mathcal{R}^p$

$$U(S) = \min f(x) \text{ s.t. } \begin{cases} Cx & - \sum_{i \in I(S)} \omega_i s^i t_i - \sum_{i \notin I(S)} s^i t_i = v^0 \\ x & \in X \\ \sum_{i \in I(S)} t_i \geq 1, & t \geq 0 \end{cases} \quad (4.11)$$

pour chaque $i \in I(S)$, ω_i est déterminé par 4.10

Preuve

Puisque la fonction g est concave, alors l'ensemble

$$G_+ = \{z \in \mathcal{R}^p : g(z) > 0\}$$

est convexe.

- (a) Soit $I(S) = \emptyset$. De 4.6, chaque $z \in F(S)$ est représenté uniquement par $z = v^0 + \sum_{i=1}^p s^i t_i, t_i \geq 0$ ($i = 1, \dots, p$). Comme les rayons $K_i, (i = 1, \dots, p)$ définis par 4.8, appartiennent à G_+ , ce qui suit que l'ensemble $F(S)$, est aussi convexe.
 (b) Supposons que $I(S) \neq \emptyset$. Alors chaque point $z \in F(S)$ est représenté par

$$z = v^0 + \sum_{i \in I(S)} \omega_i s^i t_i + \sum_{i \notin I(S)} s^i t_i, \quad t_i \geq 0 \quad (i = 1, \dots, p).$$

Pour la démonstration des algorithmes de type $B\&B$, la caractérisation la plus importante dans le processus de partition est le concept d'exhaustivité. Une suite d'ensemble des partition $\{S^k\}, S^k \supset S^{k+1}, \forall k$, est dite exhaustive si S^k tend vers un point unique quand $k \rightarrow +\infty$, c'est à dire : $\bigcap_{k=1}^{\infty} S^k = \{x\}$

Ainsi, dans l'algorithme $B\&B$, un processus de partitionnement est dit exhaustif si toute suite de partition générée à travers l'algorithme est exhaustive.

4.2.4.1 Estimation de la borne inférieure

Soit S un n -simplexe généré à travers l'algorithme ayant $n + 1$ sommets v^1, \dots, v^{n+1} . Supposons que S est généré à partir du simplexe S' telle décrit ci-dessus, et supposons

que la borne inférieure $\mu(S')$ de f_0 sur l'ensemble $S' \cap F$ est connue ; rappelons que F est l'ensemble des solutions réalisables du problème (4.1). On calculera une borne inférieure $\mu(S)$ de f_0 sur l'ensemble $S \cap F$.

Soit v^0 un point de S , par exemple,

$$v^0 = \frac{1}{n+1}(v^1 + \dots + v^{n+1});$$

pour chaque $i = 0, \dots, m$ soit s^i le sous gradient de g_i au point v^0 . De plus, soit U la matrice $n \times (n+1)$ avec les colonnes v^1, \dots, v^{n+1} et \tilde{U} la matrice $(n+1) \times (n+1)$ avec les colonnes $(v^1, 1), \dots, (v^{n+1}, 1)$, c'est à dire,

$$U = (v^1, \dots, v^{n+1}), \quad \tilde{U} = \begin{pmatrix} v^1 & \dots & v^{n+1} \\ 1 & \dots & 1 \end{pmatrix}$$

Ainsi, pour chaque $i = 0, \dots, m$, la fonction affine

$$l_i(x) = (x - v^0)s^i + g_i(v^0) = s^i x + (g_i(v^0) - s^i v^0)$$

est un sous estimateur de la fonction convexe g_i , et la fonction affine

$$\varphi_i(x) = -\sum_{j=1}^{n+1} h_i(v^j) \tilde{U}_j^{-1}(x, 1)$$

où \tilde{U}_j^{-1} est la $j^{\text{ième}}$ ligne de la matrice \tilde{U}^{-1} , est un sous estimateur de la fonction $-h_i$. Généralement, φ_i est dite l'enveloppe convexe de $-h_i$ sur S . Ainsi, une borne inférieure de f_0 sur l'ensemble $S \cap F$ peut être calculée en résolvant le problème de programmation linéaire suivant :

$$\min\{l_0(x) + \varphi_0(x) : x \in (\mathcal{X} \cap S), l_i(x) + \varphi_i(x) \leq 0, i = 1, \dots, m\}. \quad (4.12)$$

En posant

$$x = \sum_{j=1}^{n+1} \lambda_j v^j = U\lambda, \quad (4.13)$$

le programme linéaire (4.12) peut être reformulé d'une manière équivalente comme le programme linéaire suivant avec les nouvelles variables $\lambda = (\lambda_1, \dots, \lambda_{n+1})$

$$\begin{aligned} & \min \sum_{j=1}^{n+1} (s^0 v^j - h_0(v^j)) \lambda_j \\ \text{S} \cdot \text{à} : & (AU)\lambda + b \leq 0 \\ & \sum_{j=1}^{n+1} (s^i v^j - h_i(v^j)) \lambda_j + (g_i(v^0) - s^i v^0) \leq 0, \quad i = 1, \dots, m \\ & \sum_{j=1}^{n+1} \lambda_j = 1 \\ & \lambda_j \geq 0, \quad j = 1, \dots, n+1 \end{aligned} \quad (4.14)$$

Finallement, soit $\tilde{\mu}(S)$ la valeur optimale du problème (4.14), si elle existe, on peut calculer une borne inférieure par

$$\mu(S) = \begin{cases} \max\{\mu(S'); (\tilde{\mu}(S) + (g_0(v^0) - s^0 v^0))\}, & \text{Si } \tilde{\mu}(S) < +\infty, \\ +\infty & \text{ailleurs.} \end{cases} \quad (4.15)$$

Notons qu'on a $\tilde{\mu}(S) < +\infty$ quand l'ensemble des solutions réalisables du problème (4.14) est non vide. Si le problème (4.14) n'a pas de solution réalisable, il s'ensuit directement que $S \cap F = \emptyset$.

Pour le cas $\tilde{\mu}(S) < +\infty$, soit $\lambda(S)$ une solution optimale de (4.14). Le point

$$x = U\lambda(S) \quad (4.16)$$

est utilisé pour l'établissement de la convergence de l'algorithme.

4.2.5 Bornes supérieures

Pour chaque simplexe généré à travers l'algorithme satisfaisant $\mu(S) < +\infty$, on essaiera de trouver les points réalisable du problème (4.1) contenu dans S . Ceci sera possible en choisissant un sous ensemble de S , par exemple $\{v^0, v^1, \dots, v^{n+1}\}$ et $x(S)$ une collection de points réalisables trouvés dans S notée par $Q(S)$. On pose $Q(S) = \emptyset$ quand aucun point réalisable n'est trouvé durant la recherche dans S . À travers l'algorithme, plusieurs points réalisable peuvent être trouvés; de cette façon, la borne supérieure de a valeur optimale du problème (4.1) peut être perfectionnée itérativement.

Remarque : Pour la résolution du problème d'optimisation d'une fonction linéaire sur l'ensemble des solution efficaces, on c'est limité pour le cas continu(MOLP) seulement, pour le cas discret(MOILP) voir [Jor08, eDC06].

4.3 L'algorithme de Branch and Bound pour l'optimisation sur l'ensemble de solutions efficaces d'un problème linéaire bi-critère

Dans cette partie on considère que le problème (1.8) est un programme linéaire bi-critère($p = 2$), on le note par P , telle que C est une matrice $2 * n$ avec les lignes C_1^T et C_2^T . L'ensemble polyhedral compact et non vide de X est supposé sous cette forme

$$X = \{x \in \mathbf{R}^n | Ax = a, \quad x \geq 0\}.$$

Avec A est une matrice $n * m$ et a est un vecteur colonne de dimension m . On note E_X l'ensemble de solutions efficaces dans l'espace de décision et par E_Y l'ensemble de solutions efficaces dans l'espace des critères, supposés aussi non vide et compact, sachant que (1.12) possède au moins une solution optimale globale.

Considérons le problème

$$(PR) \quad \min f(x) \quad s.t \quad x \in X$$

le problème relaxé de (P) . Soit x^R la solution optimale de (PR) . Si $x^R \in E_X$, alors x^R est la solution optimale de (P) , dont il y a plusieurs chemins pour résoudre un programme linéaire approprié, vérifiant que si ou pas un $x \in X$ est une solution efficace.

Le cas favorable c'est quand (1.8) est complètement efficace c-à-d $E_X = X$ d'où $E_Y = Y$ alors x^R est sûrement efficace donc c'est une solution optimale de (P) .

Même si nous n'avons aucun des cas favorable précités, le problème bi-critère possède quelques propriétés spéciales qui peuvent exploitées pour résoudre (P) ; pour chaque $i = 1, 2$ soit

$$M_i = \max \langle C_i, x \rangle \tag{4.17}$$

$$s.t \quad x \in X \tag{4.18}$$

Et soit

$$m_i = \max \langle C_i, x \rangle \tag{4.19}$$

$$s.t \langle C_{3-i}, x \rangle = M_{3-i} \tag{4.20}$$

$$x \in X \tag{4.21}$$

De plus soit

$$I = \{b \in R | m_2 \leq b \leq M_2\}$$

pour chaque $b \in I$, soit

$$(P_b) \quad w(b) = \max \langle c_1, x \rangle$$

$$s.t \quad \langle c_2, x \rangle \geq b$$

$$x \in X$$

Le point $x^\circ \in \mathbf{R}^n$ satisfait $x^\circ \in X_E$ si et ssi x° est la solution optimale du problème (P_b) pour un $b \in I$, la fonction $w(b)$ est continue, concave, linéaire et décroissante sur I . Supposons que (1.8) n'est pas complètement efficace. Alors l'ensemble Y_E dans l'espace des critères consiste à l'un des sommets de Y , arrêtes de Y , ou bien c'est l'union d'un nombre fini des arrêtes connectées de Y [HB02].

Pour un $y = (y_1, y_2)^T \in Y$, $y \in E_Y$ ssi $y_2 \in I$ et $y_1 = w(y_2)$.

Si $m_2 = M_2$ alors $Y_E = \{(M_1, M_2)^T\}$, $X_E = \{x \in X | \langle c_i, x \rangle = M_i, i = 1, 2\}$

Et le problème (P) peut se réduire à un problème d'optimisation de contraintes linéaire suivant :

$$(P_I) \quad \min \quad f(x) \\ \text{s.t.} \quad \langle c_i, x \rangle = M_i \quad i = 1, 2 \\ x \in X$$

d'où la solution optimale de (P_I) est aussi optimale pour (P) , autrement il existe une relation d'équivalence entre le bord de E_Y et la fonction linéaire w sur I .

Soit

$$m_2 = b_1 < b_2 < \dots < b_K = M_2,$$

alors $y^k = (w(b_k), b_k)^T$, $k = 1, \dots, K - 1$ sont les point extrêmes efficaces de E_Y et les segments

$$F_y^k = \{y \in Y | y = \lambda y^k + (1 - \lambda)y^{k-1}, 0 \leq \lambda \leq 1\}, \quad k = 1, \dots, K,$$

sont les arrêtes efficaces de E_Y , de plus on peut remarquer que l'ensemble

$$F_x^k = \{x \in X | cx \in F_y^k\} \quad k = 1, \dots, K - 1,$$

est la face maximale efficace de E_X (benson). Étant donné un point efficace extrême (α_k, b_k) de E_Y , $1 \leq k \leq K - 1$, l'arrête F_y^k peut s'écrire comme suit

$$F_y^k = \{y \in Y | y_1 + \mu_k y_2 = \langle a, q^k \rangle\}. \quad (4.22)$$

où $\langle a, q^k \rangle$ est la solution optimale de

$$(P_{b,\alpha}) \quad \max \quad \mu \\ \text{s.t.} \quad -b\mu \langle a, q \rangle = \alpha \\ -c_2\mu + A^T q \geq c_1 \\ \mu \geq 0$$

Avec $b = b_k$ et $\alpha = \alpha_k = w(b_k)$

Après la résolution de $(P_{b,\alpha})$, le point extrême efficace $(\alpha_{k+1}, b_{k+1})^T$ de E_Y , peut être

obtenu par $b_{k+1} = \langle c_2, \bar{x}^{k+1} \rangle$ et $\alpha_{k+1} = \langle c_1, \bar{x}^{k+1} \rangle$

où \bar{x}^{k+1} est la solution optimale du programme linéaire

$$(V_k) \quad \begin{array}{ll} \max & \langle c_2, x \rangle \\ \text{s.t} & \langle c_1, x \rangle + \mu_k \langle c_2, x \rangle = \langle a, q^k \rangle \\ & x \in X \end{array}$$

$$E_X = \bigcup_{k=1}^{K-1} F_x^k$$

Le problème (P) peut se réduire à un nombre fini de problèmes, où f est à minimiser sur F_x^k , qui peut s'écrire comme suit :

$$(F_k) \quad \begin{array}{ll} \min & f(x) \\ \text{s.t} & \langle c_1, x \rangle + \mu_k \langle c_2, x \rangle = \langle a, q^k \rangle \\ & x \in X \end{array}$$

A condition que l'ensemble des solutions réalisable n'est pas complètement efficace, dont on peut construire le bord efficace de E_Y et l'arrêtes efficaces E_X par une approche énumérative proposé par Benson[Ben79b]. En commençant avec $k = 1$ au point extrême $y = (\alpha_1, b_1)^T = (M_1, m_2)^T$ de Y_E et on résout les sous problèmes (P_{b_k}, α_k) et (V_k) jusqu'à un point extrême $(\alpha_k, b_k)^T$ de Y_k avec $b_k = M_2$ est atteint. Pendant ce temps étape par étape le bord efficace F_y^k défini par (4.22) et l'arrêtes efficaces définies comme l'ensemble faisable de (V_k) et (F_k) sont aussi obtenus.

Une extension de la précédente énumérative méthode à été proposé par Benson pour résoudre (P). Là donc à l'iteration k , de plus de problème (P_{b_k}, α_k) et V_k , le problème (F_k) aussi sera résolu. Naturellement l'algorithme peut terminer avant la construction de tous les faces efficace de X_E par l'utilisation des bornes supérieurs et inférieurs appropriés. A un point donner de l'algorithme, soit x^c la solution retenue c-à-d la meilleur solution trouvée et soit $UB = f(x^c)$ la borne supérieur de la valeur optimale de (P). Pour $i = 1, \dots, K - 1$, soit x^j une solution optimale de (F_j) . Alors à l'iteration k , x^c peut on la d'écrire comme l'un de $\{x^1, \dots, x^k\}$, qui correspond à la pus petite valeur de f . Après avoir déterminé le point efficace extrême (α_{k+1}, b_{k+1}) de Y_E par la résolution de (V_k) cherchons la solution optimale x^R pour le problème relaxé

$$(PR_b) \quad \begin{array}{ll} \min & f(x) \\ \text{s.t} & \langle c_2, x \rangle \geq b \\ & x \in X \end{array}$$

avec $b = b_{k+1}$. Soit $LB = f(x)$, si $LB \geq UB$, alors arrêter, x^c est la solution optimale de (P). Dans le cas $x^R \in X_E$, on peut arrêter le processus si $f(x^R) < f(x^c)$. Alors x^R est la solution optimale de (P).

4.3.1 Algorithme de Branch and bound

Proposition 7 [FM00] Soit $y^1(y_1^1, y_2^1)$ et $y^2(y_1^2, y_2^2)$ deux points efficaces de Y telles que $y_2^1 < y_2^2$. Alors pour chaque $y(y_1, y_2) \in E_Y$

$$y_2^1 \leq y_2 \leq y_2^2 \quad (4.23)$$

ssi

$$(y_2^2 - y_2^1)(y_1 - y_1^1) + (y_1^1 - y_1^2)(y_2 - y_2^1) \geq 0 \quad (4.24)$$

Preuve

Soit $y = (y_1, y_2)^T \in Y$, on a $y \in E_Y$ si et ssi $y_2 \in I$ et $y_1 = w(y_2)$. Pour $y \in E_Y$ soit $\lambda = (y_2 - y_2^1)/(y_2^2 - y_2^1)$ c'est claire que $\lambda \in [0, 1]$ si et ssi y vérifie les inégalité (4.23). Alors pour chaque $\lambda \in [0, 1]$, $y_2 = \lambda y_2^2 + (1 - \lambda)y_2^1 = w(y_2)$, $y_1^i = w(y_2^i)$ pour $i = 1, 2$ et w concave sur I , on a

$$\begin{aligned} y_1 &= w(\lambda y_2^2 + (1 - \lambda)y_2^1) \\ &\geq \lambda w(y_2^2) + (1 - \lambda)w(y_2^1) \\ &= \lambda y_1^2 + (1 - \lambda)y_1^1 \\ y_1 - y_1^1 &\geq [(y_2 - y_2^1)/(y_2^2 - y_2^1)](y_1^2 - y_1^1) \\ (y_1 - y_1^1)(y_2^2 - y_2^1) &\geq (y_2 - y_2^1)(y_1^2 - y_1^1) \\ (y_1 - y_1^1)(y_2^2 - y_2^1)(y_2 - y_2^1)(y_1^2 - y_1^1) &\geq 0 \end{aligned}$$

Géométriquement la proposition 7 localise tous les points efficaces qui accomplit (4.23) et peuvent être localisés dans le demi espace fermé reliant les deux points y_1 et y_2 , c'est aussi facile de voir que tout $y \in Y$ (4.24) implique (4.23). Par conséquent pour déterminer la borne inférieur du problème

$$\min f(x) \quad (4.25)$$

$$s.t \ y_2^1 \leq \langle c_2, x \rangle \leq y_2^2 \quad (4.26)$$

$$x \in X_E \quad (4.27)$$

Il suffit de résoudre le problème relaxé

$$\min f(x) \quad (4.28)$$

$$s.t \ \langle (y_2^2 - y_2^1)c_1 + (y_1^1 - y_1^2)c_2, x \rangle \geq y_1^1 y_2^2 - y_2^1 y_1^2 \quad (4.29)$$

$$x \in X \quad (4.30)$$

au lieu de

$$\min f(x) \quad (4.31)$$

$$s.t \ y_2^1 \leq \langle c_2, x \rangle \leq y_2^2 \quad (4.32)$$

$$x \in X \quad (4.33)$$

Chaque point réalisable pour (4.29) et (4.30) est réalisable aussi pour (4.31) et (4.33).

Proposition 8 [FM00] *Soit y^1 et y^2 deux points efficaces extrême de Y sachant que $y_2^1 < y_2^2$ et soit \bar{x} une solution optimale du problème linéaire lexicographique*

$$\text{lexmax}\{\langle (y_2^2 - y_2^1)c_1 + (y_1^1 - y_1^2)c_2, x \rangle, \langle c_1, x \rangle\} \quad (4.34)$$

$$s.t \ x \in X \quad (4.35)$$

*et soit $\bar{y} = c\bar{x}$ si $\bar{y} = y^1$, alors y^1 et y^2 sont les points limite du bord efficace de Y . **Si non** \bar{y} est le point extrême efficace de Y sachant que $y_2^1 < \bar{y}_2 < y_2^2$*

Preuve

Le problème (4.34) et (4.35) peut on le transformer sous la forme suivante

$$\text{lexmax}\{\langle (y_2^2 - y_2^1, y_1^1 - y_1^2)^T, y \rangle, y_1\} \quad (4.36)$$

$$s.t \ y \in Y \quad (4.37)$$

On note par Y_1 l'ensemble des solutions optimales du (4.36) sur Y . Puisque $y_2^2 - y_2^1 > 0$ et ω est strictement décroissante sur I et on a aussi $y_1^1 - y_1^2 > 0$. Par conséquent, Y_1 est un ensemble des point extrêmes efficaces ou une arête efficace de Y .

Pour chaque étape de la méthode de B&B pour résoudre le problème (P) dans le cas bi-critère, un ensemble fini de segments noté \mathcal{N} dans l'espace des critères Y est retenus. Le segment S est déterminé par les points limites $y^1(S)$ et $y^2(S)$, qui sont des points efficaces extrêmes tel que $y_2^1(S) < y_2^2(S)$. Et en enregistrant aussi une solution optimale $x^R(S)$, pour chaque segment active, du problème relaxé (4.28)-(4.30) avec $y^i = y^i(S)$ $i = 1, 2$ et la valeur $B(S) = f(x^R)$.

L'algorithme de B&B pour résoudre (P) dans le cas bi-critère est présenté comme suit

Algorithme 2.

▷ **Initialisation**

Etape 1. Calculer M_1 et m_2 , on résolvant (4.17)-(4.18) pour $i = 1$ et (4.19)-(4.21) pour $i = 2$. Soit x^c la solution optimale de (4.19) et (4.21) et soit $UB \leftarrow f(x^c)$

Etape 2. Calculer M_2 par la résolution de (4.17)-(4.18) pour $i = 2$. **Si** $m_2 = M_2$ aller à l'étape 5 **sinon** calculer m_1 par la résolution de (4.19)-(4.21). pour $i = 1$. Soit \bar{x} une solution optimale de (4.19)-(4.21). **Si** $f(\bar{x}) < UB$ alors $UB \leftarrow f(\bar{x})$ et $x^c \leftarrow \bar{x}$. **Sinon** aller à l'étape suivante.

Etape 3. Soit le segment S_0 , défini par les deux points $y^1(S_0) = (M_1, m_2)^T$ et $y^2(S_0) = (m_1, M_2)^T$. Trouver la solution optimale \bar{x} pour le problème (4.28)-(4.30), avec $y^i = y^i(S_0)$, $i = 1, 2$.

Etape 4. **Si** $f(\bar{x}) = UB$ arrêter x^c est la solution optimale de problème (P) , **sinon** $B(S_0) \leftarrow f(\bar{x})$, $x^R(S_0) \leftarrow \bar{x}$, $k \leftarrow 1$, $N_1 \leftarrow \{S_0\}$ et aller à l'iteration k .

Etape 5. Trouver la solution, optimale x^I du problème (P_I) et arrêter le processus de résolution, x^I est la solution optimale de (P) .

▷ **Iteration** $k \geq 1$

Etape k 1. Sélectionner $S_k \in \operatorname{argmin}\{B(S) | S \in N_k\}$. **Si** $x^R(S_k) \in E_X$ alors arrêter le processus de résolution, avec x^R solution optimale du problème (P) ; **sinon** aller à l'étape suivante.

Etape k 2 Trouver la solution optimale \bar{x}^k du problème (4.34)-(4.35), avec $y^i = y^i(S_k)$, $i = 1, 2$ et soit $\bar{y}^k = C\bar{x}^k$. **Si** $f(\bar{x}^k) < UB$, alors $UB \leftarrow f(\bar{x}^k)$ et $x^c \leftarrow \bar{x}^k$.

Etape k 3. Soient les deux segments S_{k1}, S_{k2} définis par $y^1(S_{k1}) = y^1(S_k)$, $y^2(S_{k1}) = y^1(S_{k2}) = \bar{y}^k$ et $y^2(S_{k2}) = y^2(S_k)$. $j \leftarrow 1$ et $N_{k+1} \leftarrow N_{k+1} \cup \{S_{k1}\} \cup \{S_{k2}\} \setminus \{S_k\}$.

Etape k 4. Considérons le problème (4.28)-(4.30) défini par $y^i = y^i(S_{kj})$, $i = 1, 2$. **Si** $x^R(S_k)$ est réalisable pour (4.28)-(4.30), alors $x^R(S_{kj}) \leftarrow x^R(S_k)$, $B(S_{kj}) \leftarrow B(S_k)$ et aller à l'étape **k 5**. **Sinon** trouver la solution optimale $x^R(S_{kj})$ pour (4.28)-(4.30), $B(S_{kj}) \leftarrow f(x^R(S_{kj}))$.

Etape k 5. **Si** $j = 1$ alors $j \leftarrow 2$, aller l'étape **k 4**. **Sinon** aller à l'étape suivante.

Etape k 6. Supprimer tout segment $S \in N_{k+1}$ tel que $B(S) \geq UB$. Notons par N_{k+1} l'ensemble restant. **Si** $N_{k+1} = \emptyset$ alors arrêter le processus de résolution, x^c est la solution optimale du problème (P) , **sinon** $k \leftarrow k + 1$ et aller à l'étape **k 1**.

La construction de segments dans l'iteration 1 est représenté dans la figure 4.4 suivante :

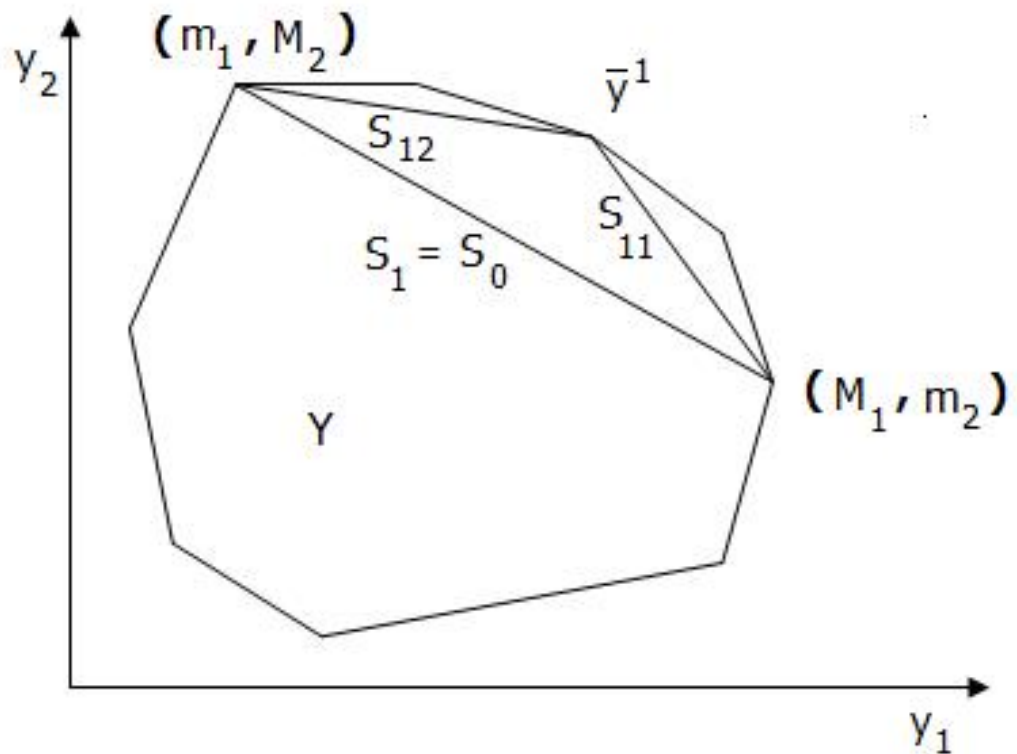


FIG. 4.4 – Construction de segments dans l'iteration 1

4.3.2 Exemples didactique

Nous illustrons le fonctionnement de l'algorithme sur deux exemples numériques dont le premier présenté dans [ES94, Cha06].

Exemple 1 : Considérons le problème multi-objectifs suivant :

$$(MOLP) \left\{ \begin{array}{l} \max z_1 = x_1 - 3x_2 \\ \max z_2 = x_1 + 3x_2 \\ s.c \quad x_1 + 2x_2 \leq 8 \\ \quad \quad 2x_1 + x_2 \leq 7 \\ \quad \quad x_1 - 2x_2 \leq 1 \\ \quad \quad x_1 \geq 0 \\ \quad \quad x_2 \geq 0 \end{array} \right.$$

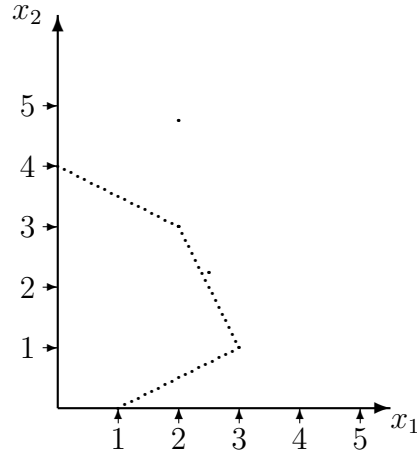


FIG. 4.5 – Espace de décision pour l'exemple didactique 1

le problème principal (P_1) est le suivant :

$$(P_1) \begin{cases} \min f(x) & = -3x_1 - 2x_2 \\ \text{s.c} & (x_1, x_2)^T \in E(MOLP) \end{cases}$$

Initialisation Etape 1 $M_1 = 1$; $m_2 = 1$; $x^c = (1, 0)^T$; $UB := -3$;

Etape 2 $M_2 = 12$; puisque $M_2 \neq m_2$ alors $m_1 = -12$; $\bar{x} = (2, 3)^T$ $f(\bar{x}) = -12$; $UB := -12$ et $x^c := \bar{x} = (2, 3)^T$

Etape 3 $S_0 := \{y^1(s_0) = (M_1, m_2)^T; y^2(s_0) = (m_1, M_2)^T\}$

Etape 4 $f(\bar{x}) \neq UB$ alors $B(s_0) := f(\bar{x})$; $x^R(s_0) := \bar{x}$; $k := 1$; $N_1 := \{s_0\}$

Iteration 1 Etape 1.1 x^R est une solution efficace (procédure de teste d'efficacité) de (MOLP), alors $x^c = (2, 3)^T$ est la solution optimale pour le problème (P_1)

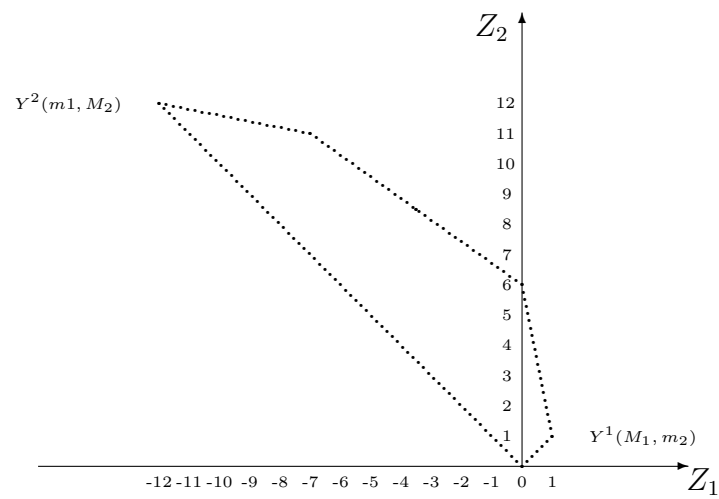


FIG. 4.6 – Espace des critères pour l'exemple didactique 1

Exemple 2 : Considérons le problème multi-objectifs suivant :

$$(MOLP_1) \left\{ \begin{array}{l} \max \quad z_1 = 5x_1 + 12x_2 + 17x_3 + 10x_4 - 13x_5 - 4x_6 + 17x_7 + 17x_8 - 4x_9 + 16x_{10} \\ \max \quad z_2 = -18x_1 - 6x_2 + 13x_3 - 20x_4 - 14x_5 - 12x_6 - 12x_7 + 4x_8 - 9x_9 - 12x_{10} \\ s.c \quad \begin{cases} 3x_1 + 169x_2 + 136x_3 + 61x_4 + 30x_5 + 99x_6 + 68x_7 + 168x_8 + 139x_9 + 35x_{10} \leq 2728 \\ 149x_1 + 105x_2 + 76x_3 + 38x_4 + 140x_5 + 180x_6 + 58x_7 + 114x_8 + 124x_9 + 196x_{10} \leq 3540 \\ 89x_1 + 41x_2 + 166x_3 + 39x_4 + 76x_5 + 164x_6 + 68x_7 + 74x_8 + 159x_9 + 54x_{10} \leq 2817 \\ 186x_1 + 134x_2 + 101x_3 + 136x_4 + 172x_5 + 129x_6 + 107x_7 + 141x_8 + 191x_9 + 50x_{10} \leq 4047 \\ 93x_1 + 168x_2 + 142x_3 + 61x_4 + 171x_5 + 164x_6 + 145x_7 + 109x_8 + 105x_9 + 175x_{10} \leq 4008 \\ 84x_1 + 4x_2 + 86x_3 + 108x_4 + 119x_5 + 132x_6 + 62x_7 + 89x_8 + 176x_9 + 147x_{10} \leq 3041 \\ x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10} \geq 0 \end{cases} \end{array} \right.$$

le problème principal (P_2) est le suivant :

$$(P_2) \left\{ \begin{array}{l} \min f(x) = dx = 18x_1 - 11x_2 + 4x_3 - x_4 + 16x_5 + 10x_6 - 2x_7 - 19x_8 + 13x_9 - 2x_{10} \\ s.c \quad (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10})' \in E(MOLP_1) \end{array} \right.$$

La première iteration de l'algorithme appliqué pour l'exemple deux génère trois points efficace extrêmes(A,B,C) dans l'espace des critères, comme le montre la figure 4.7. Dont les deux point A et B sont les points extrêmes efficaces limite d'une suite de toutes les arrêtes efficaces adjacentes et le point C n'est pas optimal pour P_2 .

En deuxième iteration le point extrême D est génère, ainsi les deux segments S_{21} et S_{22} ,

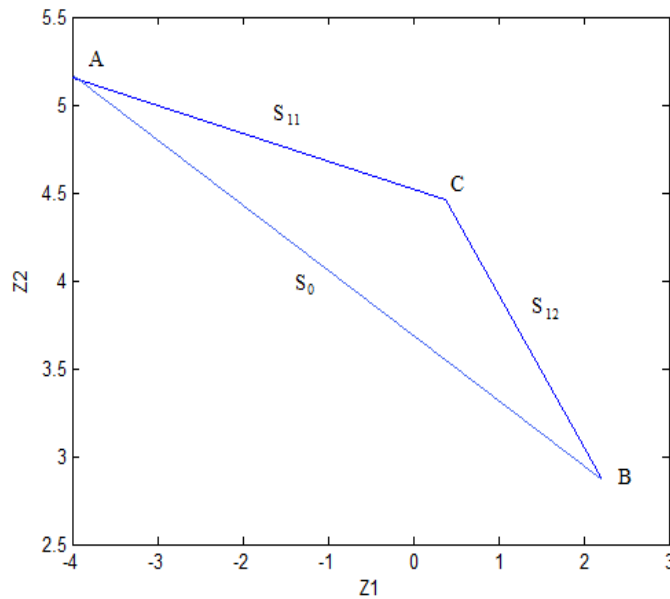


FIG. 4.7 – L'itération 1 du l'algorithme pour l'exemple 2

comme le montre la figure 4.8. Le point E est généré en troisième itération avec les deux segments S_{31} et S_{32} , comme le montre le figure 4.9. Le déroulement de l'algorithme, séparation(génération de segments) et évaluation(la borne inférieure ainsi la borne supérieure), est donné dans le tableau 4.1.

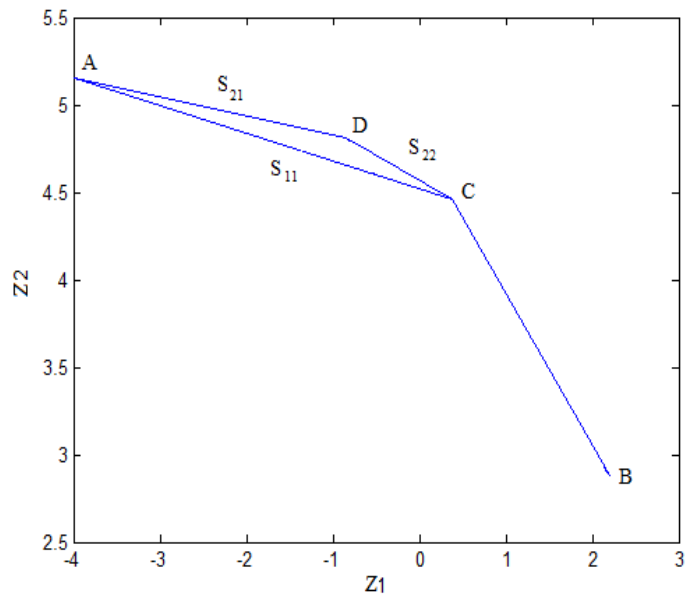


FIG. 4.8 – L'itération 2 de l'algorithme pour l'exemple 2

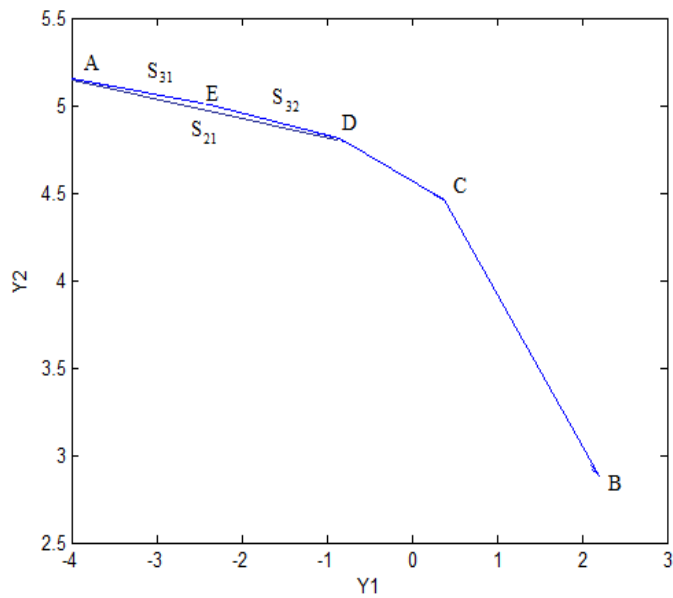


FIG. 4.9 – L'itération 3 de l'algorithme pour l'exemple 2

		Iteration 1			Iteration 2		Iteration 3	
y^1	z_1	515.2377	515.2377	445.9828	515.2377	481.1747	515.2377	500.5391
	z_2	-399.7151	-399.7151	38.3858	-399.7151	-84.2216	-399.7151	-237.1867
Points		A ↑	A ↑	C ↑	A ↑	D ↑	A ↑	E ↑
Segments		S_0 ↓	S_{11} ↓	S_{12} ↓	S_{21} ↓	S_{22} ↓	S_{31} ↓	S_{32} ↓
Points		B ↓	C ↓	B ↓	D ↓	C ↓	E ↓	D ↓
y^2	z_1	288.4880	445.9828	288.4880	481.1747	445.9828	500.5391	481.1747
	z_2	220.6084	38.3858	220.6084	-84.2216	38.3858	-237.1867	-84.2216
x_1		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
x_2		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
x_3		0.0000	0.0000	11.3368	0.0000	11.3368	0.0000	0.0000
x_4		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
x_5		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
x_6		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
x_7		0.0000	11.7875	0.0000	20.0165	0.0000	22.1850	22.1850
x_8		14.3783	10.0321	4.8328	7.8369	4.8328	7.2584	7.2584
x_9		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
x_{10}		8.9269	6.8875	10.6937	1.4365	10.6937	0.0000	0.0000
Efficace		Non	Non	Oui	Non	Non	Oui	Oui
borne inf		-291.0419	-227.9590	-67.8636	-191.8073	-67.8636	-182.2804	-182.2804
borne Sup		-141.2075			-141.2075		-182.2804	
Solutions		Rejetées					Optimale	
f^*		//					-182.2804	

TAB. 4.1 – Résumé de l'exécution de l'exemple deux

4.3.3 Convergence et finitude

Proposition 9 [FM00] *L'algorithme 2 est fini et termine toujours avec une solution optimale au problème (P)*

Preuve :

Les deux points limites de S_0 déterminés à l'étape d'initialisation sont clairement des points extrêmes efficaces de Y . Si $m_2 = M_2$, alors E_Y contient un seul point, l'ensemble faisable de (P_I) est un ensemble efficace dans l'espace de décision X . Donc on résolvant (P_I) , on obtient la solution optimale de (P).

Dans le cas $m_2 < M_2$, les deux limites de S_0 sont aussi les points limites de la série de

consecutive d'arrêtes efficaces adjacentes de Y et $I = [m_2, M_2]$. Comme chaque point de E_X est réalisable pour (4.29)-(4.30), obtenus à partir des points limites de S_0 , dans le cas $f(\bar{x}) = UB$. En effet la solution optimale de (P) est obtenue.

Supposons maintenant que l'algorithme ne termine pas à l'étape d'initialisation. On montre que les deux point limites de chaque segment de N_k sont des point extrêmes efficaces de Y et ça est vrai pour S_0 . Si le segment S_k sélectionné dans l'étape k_1 est une arrête efficace de Y , alors automatiquement $x^R(S_k) \in E_X$, donc $x^R(S_k)$ est la solution optimale de (P). Sinon par la proposition 8 \bar{y}^k déterminé à l'étape k_2 est point efficace extrême de Y et $y_2^1(S_k) < \bar{y}_2^k < y_2^2(S_k)$. En effet, les points limites des segments récemment construits sont des points extrêmes efficaces de Y .

Comme Y a un nombre fini de point extrêmes, l'algorithme 2 termine après un nombre fini d'iteration soit à l'étape k_1 avec une solution optimale ou bien à l'étape k_6 par la détection d'une solution titulaire x^c , qui est optimale pour (P).

4.4 Complexité

La théorie de la complexité se propose d'étudier de manière formelle la difficulté des problèmes en informatique. On distingue complexité spatiale (évaluation de l'espace mémoire nécessaire au fonctionnement d'un programme) et complexité temporelle (évaluation du temps de traitement). C'est à ce deuxième point que nous nous intéressons.

En théorie de la complexité on ne traite que des problèmes de décision, c'est à dire qui revient à répondre à une question par oui ou par non. On peut toutefois parler de complexité pour un problème d'optimisation, puisqu'il se ramène simplement à un problème de décision. Par exemple, dans un problème où l'on cherche à minimiser une variable entière n , il suffira de choisir un autre entier p et de traiter le problème de décision qui consiste à comparer n à p .

La méthode par séparation et évaluation est basée sur une énumération implicite et intelligente de l'ensemble des solutions réalisables. Pour cela, la séparation consiste à décomposer le problème initial en plusieurs sous-problèmes qui sont à leur tour décomposables. Ce processus peut se visualiser sous forme d'un arbre d'énumération. Pour chaque sous-problème (noeud de l'arbre), la procédure d'évaluation calcule (dans le cas d'un problème de minimisation) une borne inférieure de la solution obtenue à partir de ce sous-problème. Au préalable, une borne supérieure de la solution optimale a été calculée et est utilisée pour éviter l'exploration de noeuds dont la valeur de la borne inférieure est supérieure à la

valeur de la borne supérieure. Cette borne supérieure est réactualisée lorsqu'une solution réalisable de valeur inférieure est trouvée. Ainsi, l'exploration de certaines branches de l'arbre est coupée, ce qui permet de ne pas énumérer réellement toutes les solutions.

Dans notre travail, les noeuds représente des simplices pour le premier algorithme et le deuxième c'est des segments, tel que le nombre de noeuds générés dépend de nombres de sommets efficaces extrêmes, donc de la region efficace (les arrêtes efficaces).

En effet le pire des cas c'est d'explorer toute l'arborescence, c-à-d de générer toutes les points extrêmes efficaces.

Expérimentations et résultats

Dans ce chapitre, nous donnons quelques détails sur l'implantation de l'algorithme de l'optimisation sur l'ensemble des solutions efficaces (cas de programme linéaire à deux critères), qu'on a conçu sous l'environnement MATLAB 7.0.

5.1 Programme implémentant l'algorithme

La méthode programmée est basée essentiellement sur l'optimisation d'un critère linéaire sur l'ensemble des solutions efficaces. L'architecture de l'algorithme est donnée sous forme d'organigrammes dans l'annexes (C,D) les procédures essentielles sont les suivantes :

- Procédure d'initialisation : Elle permet de déterminer les deux points extrêmes efficaces, dont ces dernières sont des point limites d'une chaîne d'arrêtes consécutif efficaces ; d'évaluer les bornes inférieure et supérieure de la fonction f . En optimisant chaque critère individuellement sur l'ensemble de solutions réalisable, créant un segment qui relie les deux points limites efficaces.
- Procédure de séparation : il s'agit de créer des segments (sous-ensemble), donc de subdiviser l'espace réalisable contenant l'ensemble de solutions efficaces, dans l'espace des critères en plusieurs sous espaces. Dont le premier sous espace est généré dans l'étape d'initialisation et chaque sous espace est divisé en deux sous espaces, où on peut représenter cette procédure par un arbre binaire, tel que chaque noeud

représente un sous espaces et chaque arrête représente une subdivision.

- Procédure d'évaluation : permet d'évaluer les bornes inférieures et supérieures de f sur chaque sous espace.
- procédures de testes : Elle permet de tester
 1. Si la solution optimale trouvée sur l'ensemble réduit est efficace, alors on arrête le processus de résolution.
 2. Si la borne inférieure de f sur un sous espace est supérieur a la borne supérieure, alors on supprime le segment correspond a ce sous espace, ça revient à supprimer une branche de l'arbre.
 3. Si l'ensemble de segment est vide, alors on arrête le processus de résolution.

5.1.1 Génération aléatoire des exemples tests

Les exemples aléatoires testés ont été générés par les procédures suivantes :

Etape 1 Lecture de la dimension de l'exemple :

n : Nombre de variable.

m : Nombre de contraintes.

Etape 2 Génération des éléments de la matrice des contraintes a_{ij} selon la loi uniforme

$U_{[-20,20]}$.

Etape 3 Le vecteur b est calculé de façon que le polyèdre généré ne sont pas vide[Cha06]

Soit $A_i = k \sum_{j=1}^n a_{ij}$, $\forall i \in \{1, \dots, m\}$; avec $k \in \mathcal{N} \setminus \{0\}$

$b_i = A_i + \delta_i$ avec δ_i aléatoire dans $U_{[0,20]}$.

Etape 4 Génération de la matrice des coefficients coût C_{ij} aléatoire dans $U_{[-20,20]}$. $\forall i \in \{1, 2\}, j \in \{1, \dots, n\}$. De même pour le critère principal.

Le menu principal du programme est constitué de trois sous menus principaux permettant, premièrement aux utilisateurs d'introduire les données, deuxièmement d'exécuter des Benchmarks et dernièrement de générer des exemples tests.

5.2 Résultats et interprétation

L'exemple présenté ci-après à été traité dans [NTBK05], où le critère principal est la combinaison linéaire des deux critères secondaires.

Le critère principal : $f = (-2, -2, -5, 0, -2, -1, -1, -3, -2, -4)$

La matrice des critères : $C = \begin{pmatrix} 0 & 2 & 3 & 0 & 2 & 0 & 0 & 0 & 3 & 0 \\ 2 & 0 & 2 & 0 & 0 & 1 & 1 & 3 & -1 & 4 \end{pmatrix}$

La matrice des contraintes : $A = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 8 & 6 & 8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 5 & 0 & 1 & 0 & 0 & 0 & 3 & 5 \\ 0 & 0 & 5 & 7 & 0 & 1 & 3 & 0 & 0 & 0 \\ 7 & 0 & 5 & 0 & 8 & 0 & 2 & 1 & 0 & 0 \\ 0 & 7 & 0 & 4 & 3 & 0 & 0 & 0 & 6 & 0 \\ 0 & 1 & 0 & 0 & 4 & 6 & 0 & 3 & 0 & 7 \\ 0 & 0 & 8 & 0 & 7 & 0 & 0 & 0 & 0 & 6 \\ 0 & 8 & 2 & 0 & 0 & 0 & 5 & 8 & 3 & 0 \\ 0 & 3 & 8 & 0 & 0 & 0 & 6 & 1 & 0 & 0 \end{pmatrix}; b = \begin{pmatrix} 51 \\ 56 \\ 81 \\ 77 \\ 81 \\ 93 \\ 77 \\ 76 \\ 64 \\ 100 \end{pmatrix}$

L'ensemble des solutions non dominées (dans l'espace des critères) : $y^1 = (10.672, 76.280)$; $y^2 = (16.419, 73.206)$; $y^3 = (52.169, 39.741)$; $y^4 = (53.859, 37.750)$; $y^5 = (66.403, 19.410)$; $y^6 = (66.475, 19.298)$; $y^7 = (69.361, 3.714)$. On l'exécute par l'algorithme 2 : nous avons constaté que la solution optimale

$\bar{x} = (4.5268, 0.0000, 9.3333, 0.0000, 0.0000, 11.2513, 0.0000, 2.6456, 8.0562, 0.2222)^T$ et l'objectif optimal $f^* = -91.9096$ sont les même; le nombre d'iterations c'est 1; le temp d'exécution $Time = 0.2180secondes$.

Les tableaux 5.1, 5.3, 5.4 et 5.2 présentés ci-après, résument respectivement les caractéristiques des exemples générés aléatoirement, pour des tests; de petite, moyenne de taille et problèmes a matrices creuses et non creuses. Les deux premières colonnes contiennent respectivement le nombre de variables et le nombre de contraintes. Les deux autres colonnes résumes respectivement les nombres moyennes, d'iterations($Itmoy$) et du temps moyen d'exécution($CPUmoy$) de dix tests pour chaque exemple. Ces exemples ont été testés en utilisant un micro-ordinateur de type Pentium dual core 1.4GH avec une RAM de 1GB sous un système d'exploitation windows vista.

5.2.1 Tests sur des exemples aléatoires de petite taille

Les figures 5.1 et 5.2 représentent respectivement l'évaluation du nombre moyen d'iterations et temps moyen d'exécution en fonction de nombre de variables, pour les problèmes de petite taille. On remarque une légère augmentation du nombre moyen d'iterations($Itmoy$)

m	n	$Itmoy$	$CPUmoy$	m	n	$Itmoy$	$CPUmoy$	m	n	$Itmoy$	$CPUmoy$
2	2	1	0.1123	3	2	1	0.103	4	3	1.1	0.1374
2	3	1	0.0923	3	3	1	0.081	4	4	1.3	0.1169
2	4	1.3	0.1109	3	4	1.2	0.1374	4	5	1.2	0.1358
2	5	1.3	0.1481	3	5	1.1	0.106	4	6	1.7	0.201
2	6	1.3	0.1363	3	6	1.2	0.1276	4	7	1.4	0.2401
2	7	1.1	0.1421	3	7	1.5	0.238	4	8	1.8	0.2231
2	8	1.4	0.1678	3	8	1.4	0.2	4	9	1.8	0.4666
2	9	1.4	0.1717	3	9	1.4	0.2266	4	10	1.8	0.2266
2	10	1.1	0.1951	3	10	1.6	0.195				

TAB. 5.1 – Problèmes à $2 \leq m \leq 4$ et $2 \leq n \leq 10$

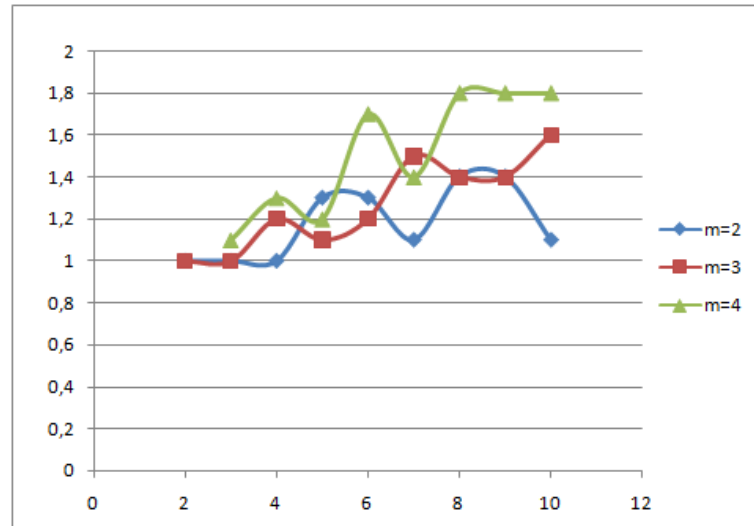


FIG. 5.1 – La courbe représentatif de ($Itmoy$) en fonction de n et $5 < m < 1$.

et le temps moyen d'exécution ($CPUmoy$) en fonction du nombre de variables, autrement dit une tendance faible. Nous avons constaté que pour les exemples tests de petites taille, le nombre moyen d'iterations ainsi que le temps moyen d'exécution sont significatives. Notons bien que la plupart des exemples la solution est obtenue dans l'étape d'initialisation.

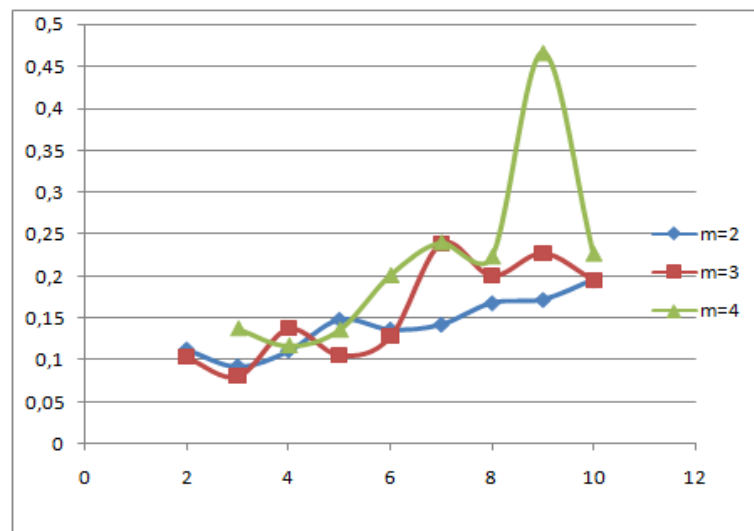


FIG. 5.2 – La courbe représentatif de (CPU_{moy}) en fonction de n et $5 < m < 1$.

5.2.2 Tests sur des exemples aléatoires à matrices creuses et non creuses de taille moyenne

		Matrices creuse		Matrice non creuse	
m	n	$Itmoy$	$CPUmoy$	$Itmoy$	$CPUmoy$
10	10	2.8	0.4525	2.5	0.48
15	15	2.7	0.5101	4.6	0.67
20	20	2.9	0.5086	4.7	0.9172
25	25	2.5	0.5788	6.9	1.2153
30	30	4.2	1.0578	5.7	1.3868
35	35	5	1.3353	12	2.9431
40	40	5.1	1.4448	7.8	2.4166
45	45	4.1	1.3668	11.1	3.41
50	50	6.1	2.067	8.3	3.345
55	55	7.9	3.5754	19.5	8.4738
60	60	10.2	3.9327	10.9	6.04
65	65	3.7	2.0111	12.2	6.16
70	70	7.2	4.03227	14.1	9.2585
75	75	5.4	3.2822	11.6	9.25
80	80	5.7	4.1962	15.3	12.644
85	85	8.1	6.3354	21.4	14.48
90	90	6.5	7.2508	16.1	15.28
95	95	6.5	6.0887	17.3	21.222
100	100	5.6	6.5925	22	21.06

TAB. 5.2 – Problèmes à des matrices creuses et non creuses

Les figures 5.3 et 5.4 représentent respectivement, l'évaluation du nombre moyen d'itérations et temps moyen d'exécution en fonction de nombre de variable (ou bien nombre de contraintes) pour les problèmes, de moyenne taille, a matrices creuses et non creuses. On remarque clairement qu' à partir de $n > 50$, les problèmes tests à matrices creuses convergent plus rapidement que ceux des problèmes tests à matrices non creuses en terme de nombre moyen d'itérations ainsi que temps moyen d'exécution et aussi on remarque que le temps $CPUmoy$ est proportionnelle a $Itmoy$. Ceci est probablement dû au gain du temps obtenu dans les premières itérations du simplexe.

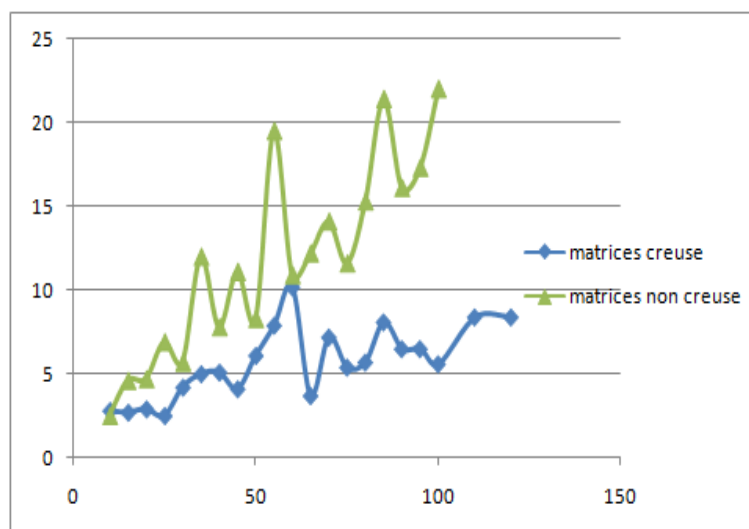


FIG. 5.3 – La courbe représentatif de ($Itmoy$) en fonction de n

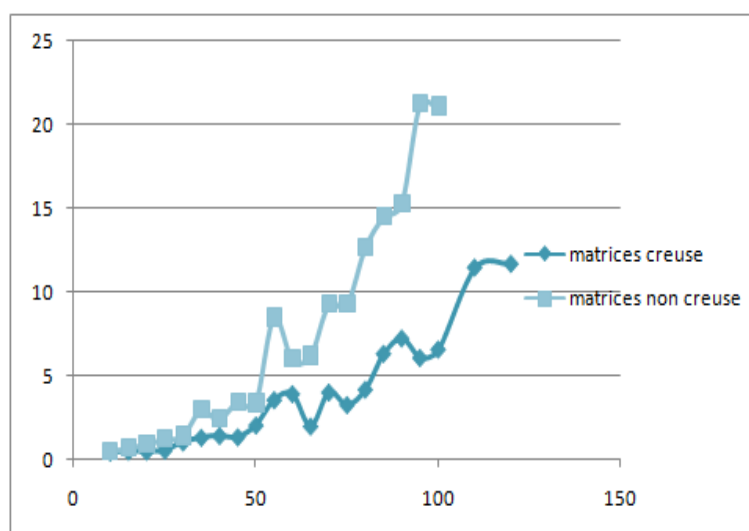


FIG. 5.4 – La courbe représentatif de ($CPU moy$) en fonction de n

5.2.3 Tests sur des exemples aléatoires de taille moyenne

n	m	$Itmoy$	CPU_{moy}	n	m	$Itmoy$	CPU_{moy}	n	m	$Itmoy$	CPU_{moy}
30	30	7.1	1.6896	50	20	7.4	1.4321	60	30	10	2.6676
35	30	9.4	2.045	50	25	7.3	1.6363	60	35	8, 5	2.4727
40	20	10.9	2.159	50	30	5	1.7816	60	40	9.8	3.2307
40	25	12.9	2.6725	50	35	9	2.35	60	45	13.4	5.3802
40	30	6.4	1.7067	50	40	15.4	4.424	60	50	9	4.0077
40	35	7.5	2.095	50	45	18.7	3.8718	60	55	12.3	6.3713
40	40	10.5	3.184	50	50	6.9	2.836	60	60	9.2	5.19

TAB. 5.3 – Problèmes à taille moyenne

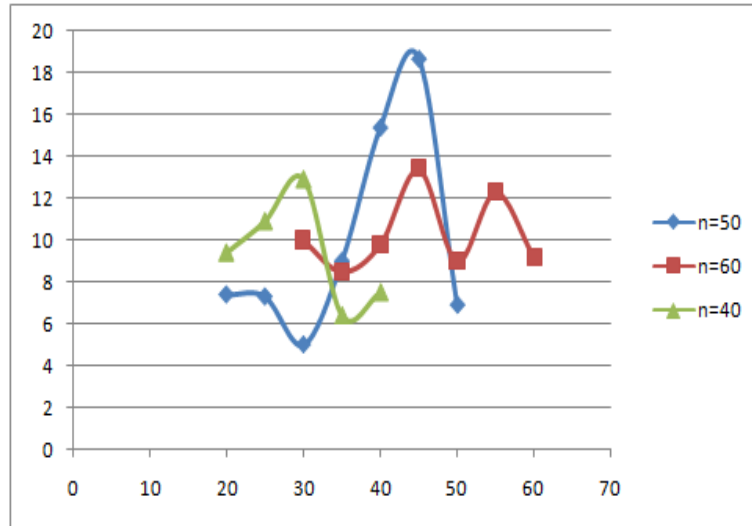


FIG. 5.5 – La courbe représentatif de ($Itmoy$) en fonction de m et $n = 40, 50$ et 60

Nous avons remarqué, que le nombre moyen d'itérations est significatif pour les exemples tests de petite et moyenne taille, ce qui confirme l'efficacité de la méthode de branch and bound pour la résolution des problèmes d'optimisation de tailles pareille, particulièrement pour les problèmes d'optimisation global. En effet le choix du critère de séparation dans Branch and Bound est essentiel pour la réduction de certain branches.

Pour les exemples tests de grande taille $n > 200, m > 100$, nous avons remarqué que le programme peut converger vers une solution, qui n'est pas forcément optimale pour le problème principal et ça est du a l'excès du nombre d'itérations durant l'execution de la procedure du simplexe (linprog), qui dépasse le nombre maximum conçu pour cette dernière.

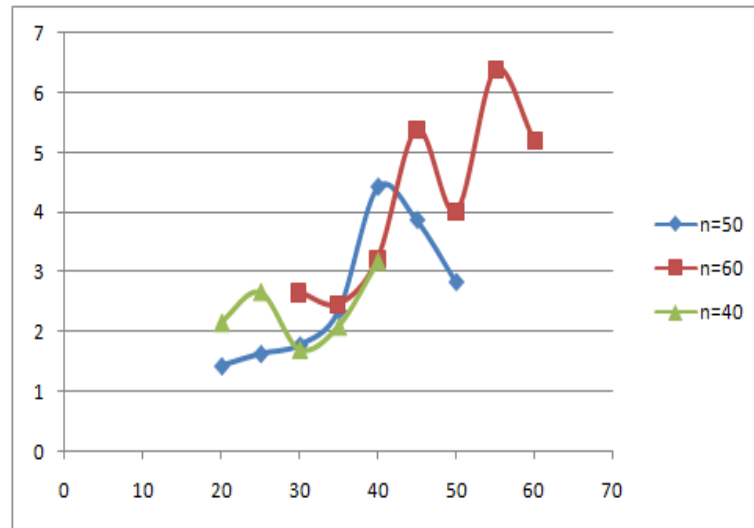


FIG. 5.6 – La courbe représentatif de (CPU_{moy}) en fonction de m et $n = 40, 50$ et 60

n	m	$Itmoy$	$CPUmoy$	n	m	$Itmoy$	$CPUmoy$
120	90	14.6	18.4126	160	130	44	120.51
120	100	37	50.6313	160	140	15	46.0716
120	110	13.4	23.7	160	150	27	166.9999
120	120	21.8	45.7189	160	160	21.7	92.3802
140	110	23.9	42.96	180	140	16.8	62.1702
140	120	27.6	59.4828	180	150	18	81.0235
140	130	15.5	42.7828	180	160	60	243.9184
140	140	42.2	127.1698	180	170	55.2	286.89
160	110	23.8	50.8467	180	180	77.5	328.85
160	120	38	90.36				

TAB. 5.4 – Problèmes $100 < n < 200$

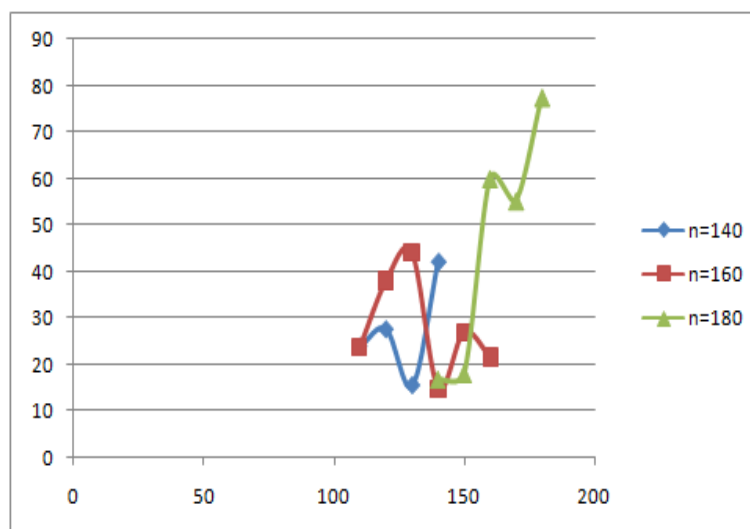


FIG. 5.7 – La courbe représentatif de (*Itmoy*) en fonction de m et $n = 140, 160$ et 180

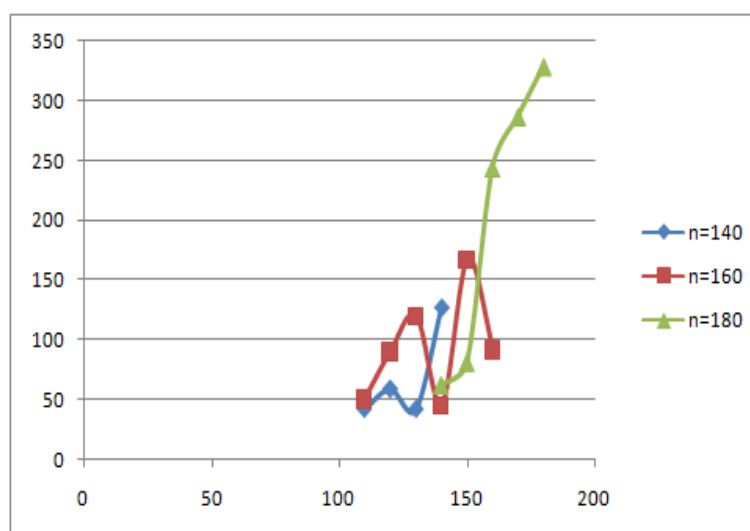


FIG. 5.8 – La courbe représentatif de (*CPUmoy*) en fonction de m et $n = 140, 160$ et 180

Conclusion générale & perspectives

Notre travail a consisté en la résolution des problèmes d'optimisation d'une fonction linéaire sur l'ensemble des solutions efficaces d'un problème linéaire multi-objectifs par la méthode de décomposition branch and bound. Pour cela, dans le premier chapitre nous avons d'abord présenté quelques généralités sur l'optimisation global (ensembles et fonctions convexe) et la dualité.

Nous avons rappelé dans le chapitre deux, certains concepts et définitions utilisés dans le cadre de résolution des problèmes multicritère. Le troisième chapitre est consacré à la caractérisation de l'ensemble des solutions efficaces, on utilisons la dualité, dans l'espace des critères pour le cas continu, ainsi en introduisant la méthode ε -contrainte pour le cas discret. Par la suite nous avons décrit deux algorithmes de type branch and bound pour la résolution du problème posé, dont le premier proposé par N. V. THOAI(2007) conçu pour le cas général(p critères) et le deuxième proposé par J. FULOP AND L. D. MUU[FM00] pour le cas d'un problème linéaire bi-critère suivi de deux exemples didactique.

En dernier lieu, nous avons implémenté l'algorithme deux sous MATLAB 7.0, suivi par des exemples traités dans la littérature(Benchmark). Sur des exemples tests générés aléatoirement, nous avons déterminé le nombre moyen, d'iterations et temps d'execution, de dix tests pour chaque exemple. Nous avons constaté que la méthode de branch and bound est très efficace pour les problèmes de petite et moyenne taille.

En effet, l'efficacité de cette technique (branch and bound) dépend essentiellement du

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

En guise de perspective de recherche, nous proposons les directions suivantes :

- Appliquer des métha-heuristiques pour la résolution de ce problème.
- Developper des méthodes de résolution sur l'ensemble solution efficaces pour le cas quadratique (voir par exemple [Bra06]).
- Appliquer des nouvelles méthodes, particulièrement pour des problèmes de grande taille par exemple les méthodes des points intérieurs ou extérieurs.

Bibliographie

- [AH99] B. Aghezzaf and M. Hachimi. Second-order optimality conditions in multiobjective optimization problems. *Journal of Optimization Theory and Applications*, 102(1) :37–50, 1999.
- [AJCGB06] R. Minguez Antonio J. Conejo, E. Castillo and R. Garcia-Bertrand. *Decomposition Techniques in Mathematical Programming*. Engineering and Science Applications, 2006.
- [BB97] H.P. Benson and G. M. Boger. Multiplicative programming problems : Analysis and efficient point search heuristic. *Journal of Optimization Theory and Application*, 94(02) :487–510, 1997.
- [Bel04] A. Metrane K. Belkeziz. Optimisation d’une fonction linéaire sur l’ensemble des solution efficace d’un problème multicritère quadratique convexe. *Annal mathématique Blaise Pascal*, 11, 2004.
- [Ben78] H.P. Benson. A finite, non adjacent extreme-points search algorithm for optimization over the efficient set. *Journal of Optimization Theory and Application*, 26, 1978.
- [Ben79a] H. P. Benson. Vector maximization with two objective functions. *Journal of Optimization Theory and Application*, 1979.
- [Ben79b] H.P. Benson. Existence of efficient solutions for vector maximization problems. *Journal of Optimization Theory and Application*, 26, 1979.
- [Ben82a] H. P. Benson. Algorithms for parametric nonconvex programming. *Journal of Optimization Theory and Application*, 38(3), 1982.

- [Ben82b] H.P. Benson. Admissible points of a convex polyhedron. *Journal of Optimization Theory and Applications*, 38, 1982.
- [Ben98] H.P. Benson. Further analysis of an outcome set-based algorithm for multiple-objective linear programming. *Journal of Optimization Theory and Application*, 97, 1998.
- [BM98] S. Bolintineanu and M. El Maghri. Second-order efficiency conditions and sensitivity of efficient points. *Journal of Optimization Theory and Applications*, 98(3) :569–592, 1998.
- [BM06] V. Vassilev B. Metev. Estimating the minimum of a function over the efficient set of a molp problem some experiments. *Institute of Information Technologies, 1113 Sofia*, 06(01), 2006.
- [Bol93] S. Bolintineanu. Minimization of a quasi-concave function over an efficient set. *Mathematical Programming*, 1993.
- [Bor99] J.M Borwein. *Convex Analysis and Nonlinear Optimisation Theory and Examples*. Gargnano, Italy, 1999.
- [Bou07] Gasmi Bouthaina. Contribution à l'étude des méthodes de résolution des méthode d'optimisation quadratiques. Master's thesis, Université Hadj lakhdar Batna(Alegerie), 2007.
- [Bra06] B. Brahmi. Méthodes primale et duale pour la résolution des problèmes de programmation quadratique convexe. Master's thesis, Département de Recherche Opérationnelle, Université A. Mira de Béjaia, 2006.
- [BS94] H. P. Benson and S. Sayin. Optimization over the efficient set : Four special cases. *Journal of Optimization Theory and Application*, 80(1), 1994.
- [Cha06] D. Chaabane. *Contribution a l'Optimisation Multicritère en Variables Discretes*. PhD thesis, Faculté Polytechnique de Mons, 2006.
- [CSZ07] G. Zhang C. Shi, J. Lu and H. Zhou. An extended branch and bound algorithm for linear bilevel programming. *European Journal of Operational Research*, 2007.
- [Dah97] Geir Dahl. *An Introduction to Convexity, Polyhedral Theory and Combinatorial Optimization*. University of Oslo Department of Informatics, 1997.
- [Dia06] T. Diallo. Etude et illustration de méthodes itératives d'optimisation non linéaire. Master's thesis, Ecole Polytechnique Fédérale de Lausanne Faculté Sciences de Base Chaire de Recherche Opérationnelle Sud Est, 2006.

- [DL08] G. Daniel and D. T. Luc. Generating the weakly efficient set of nonconvex multiobjective problems. *Journal of Global Optimization*, 41 :517–538, 2008.
- [DYCL06] C. Y. Ku D. Y. Chan and M. C. Li. A method to improve integer linear programming problem with branch-and-bound procedure. *Applied Mathematics and Computation*, 179 :484–493, 2006.
- [Eck72] J. G. Ecker. Decomposition in separable geometric programming. *Journal of Optimization Theory and Application*, 09(03), 1972.
- [eDC06] M. Abbas et D. Chaabane. Optimizing a linear function over an integer efficient set. *European Journal of Operational Research*, 174 :1140–1161, 2006.
- [Ehr05] Matthias Ehrgott. *Multicriteria Optimization*, volume 491. Lecture Notes in Economics and Mathematical Systems, 2005.
- [EK75] J.G. Ecker and I.A. Kouada. Finding efficient points for linear multiple objective programs. *Mathematical Programming*, 08 :375–377, 1975.
- [EK78] J.G. Ecker and I.A. Kouada. Finding all efficient extreme points for multiple objective linear programs. *Mathematical Programming*, 14 :249–261, 1978.
- [ER08] M. Ehrgott and S. Ruzika. Improved ε -constraint method for multiobjective programming. *Journal of Optimization Theory and Applications*, 138 :375–396, 2008.
- [ES94] J. G. Ecker and J. H. Song. Optimizing a linear function over an efficient set. *Journal of Optimization Theory and Application*, 83(03) :541–563, 1994.
- [Fli05] C. D. Flipo. Optimisation combinatoire multi-objectif :apport des méthodes coopératives et contribution à l’extraction de connaissances. Master’s thesis, Université des Sciences et Technologies de Lille U.F.R. d’I.E.E.A., 2005.
- [FM00] J. Fulop and L. D. Muu. Branch-and-bound variant of an outcome-based algorithm for optimizing over the efficient set of a bicriteria linear programming problem. *Journal of Optimization Theory and Applications*, 105 :37–54, 2000.
- [FS06] O. Spanjaard F. Sourd. Multi-objective branch-and-bound. application to the bi-objective spanning tree problem. *Laboratoire d’Informatique de Paris 6*, 2006.
- [HALT01] N. V. Thoai H. A. Le Thi, D. T. Pham. Combination between global and local methods for solving an optimization problem over the efficient set. *European Journal of Operational Research*, 2001.

- [HB02] Erjiang Sun H.P. Benson. A weight set decomposition algorithm for finding all efficient extreme points in the outcome set of a multiple objective linear program. *European Journal of Operational Research*, 139, 2002.
- [Hor88] R. Horst. Deterministic global optimization with partition sets whose feasibility is not known : Application to concave minimization, reverse convex constraints, dc-programming, and lipschitzian optimization. *Journal of Optimization Theory and Applications*, 58(1), JULY 1988.
- [HT97] R. Horst and N. V. Thoai. Utility function programs and optimization over the efficient set in multiple-objective decision making. *Journal of Optimization Theory and Application*, 92(3) :605–631, 1997.
- [JC02] A. Zilinskas J. Clausen. Subdivision, sampling, and initialization strategies for simplicial branch and bound in global optimization. *Computers and Mathematics with Applications*, 44 :943–955, 2002.
- [Jor08] J. M. Jorge. An algorithm for optimizing a linear function over an integer efficient set. *European Journal of Operational Research*, 2008.
- [KKZ04] J. Tind K. Klamroth and S. Züst. Integer programming duality in multiple objective programming. *Journal of Global Optimization*, 29 :1–18, 2004.
- [LP97] Shih-Mim Liu and G. P. Papavassilopoulos. A parallel algorithm for linear programs with an additional reverse convex constraint. *Journal of Parallel and Distributed Computing*, 45 :91–103, 1997.
- [LTHA16] L. D. Muu L. T. Hoai An, P. D. Tao. Numerical solution for optimization over the efficient set by d.c. optimization algorithms. *Operations Research Letters*, 19 :117–128, 1916.
- [MD05] G. Mavrotas and D. Diakoulaki. Multi-criteria branch and bound : A vector maximization algorithm for mixed 0-1 multiple objective linear programming. *Applied Mathematics and Computation*, 171 :53–71, 2005.
- [MSY01] I. Takahashi M. Shigeno and Y. Yamamoto. Minimum maximal flow problem an optimization over the efficient. *Scientific Research C(2)136500601 of the Ministry of Education, Culture, Sports, Science and Technology of Japan.*, 2001.
- [Nou06] K. Nouredine. Résolution numérique d’un jeu bi-matriciel multicritère. Master’s thesis, Département de Recherche Opérationnelle, Université A. Mira de Béjaia, 2006.

- [NTBK05] D. P. Vu N. T. Bach Kim, N. T. Minh Hue. The efficient outcome set of a bi-criteria linear programming and application. *AMS 1991 Mathematics Subject Classification*, 2005.
- [Pen08] J. P. Penot. Critical duality. *Journal of Global Optimization*, 40 :319–338, 2008.
- [PH86] M. Privot and S. Holin. Programmation linéaire multiobjectifs. *Institut de Mathématiques Economiques et Librairie de l'université DIJON, FRANCE*, 1986.
- [Phi72] J. Philip. Algorithme for the vector maximization problem. *Mathematical Programming*, 1972.
- [RHZ07] Y. Yamamoto R. Horst, N.V. Thoai and D. Zenke. On optimization over the efficient set in linear multicriteria programming. *Journal of Optimization Theory and Application*, 2007.
- [Ste86] R. E. Steuer. *Multiple Criteria Optimization : Theory, Computation and Application*. John Wiley, New York, 546 pp, 1986.
- [SxY06] Wenyu Sun and Ya xiang Yuan. *Optimization Theory and Methods Non-linear Programming*, volume 1. Combinatorial Optimization Ding-Zhu Du (University of Texas at Dallas), 2006.
- [SY06] W. Sun and Y. X. Yuan. *Optimization Theory and Methods Nonlinear Programming*, volume 1. Springer Science+Business Media, LLC, 233 Spring Street, New York, 2006.
- [Tho] N. V. Thoai. Global optimization over the efficient set. Department of Mathematics University of Trier 54286 Trier, Germany.
- [Tho00] N. V. Thoai. Conical algorithm in global optimization for optimizing over efficient sets. *Journal of Global Optimization*, 2000.
- [Tho02] N.V Thoai. Convergence and application of a decomposition method using duality bounds for nonconvex global optimization. *Journal of Optimization Theory and Applications*, 113(1) :165–193, 2002.
- [THP05] Hoang Tuy and N.T. Hoai-Phuong. Optimization under composite monotonic constraints and constrained optimization over the efficient set. Technical report, Institute of Mathematics, VAST, 18 Hoang Quoc Viet, 10307 Hanoi, Vietnam, 2005.

- [TZ04] Kathrin Klamroth Jorgen Tind and Sibylle Zust. Integer programming duality in multiple objective programming. *Journal of Global Optimization*, 29 :1–18, 2004.
- [VJD06] G.M. Lee V. Jeyakumar and N. Dinh. Characterizations of solution sets of convex vector minimization problems. *European Journal of Operational Research*, 174 :1380–1395, 2006.
- [XL02] Y. Xua and D. Lib. A nonlinear lagrangian dual for integer programming. *Operations Research Letters*, 30 :401–407, 2002.
- [Yam01] Y. Yamamoto. Optimization over the efficient set. *Institute of Policy and Planning Sciences, University of Tsukuba, Tsukuba, Japan*, 2001.
- [YZ75] P. L. Yu and M. Zeleny. The set of all nondominated solutions in linear cases and a multicriteria simplex method. *Journal of Mathematical Analysis and Applications*, 49 :430–468, 1975.

Annexes

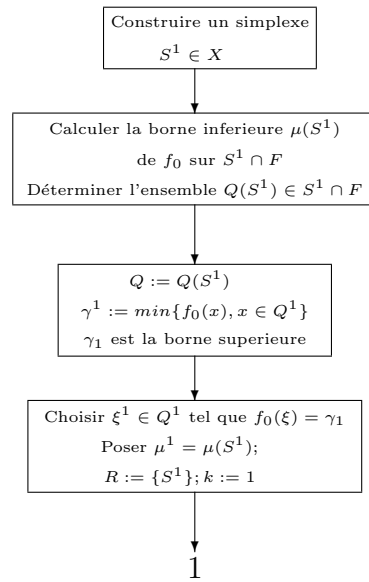


FIG. 5.9 – Etapes d’initialisation de l’algorithme 1

Annexe A

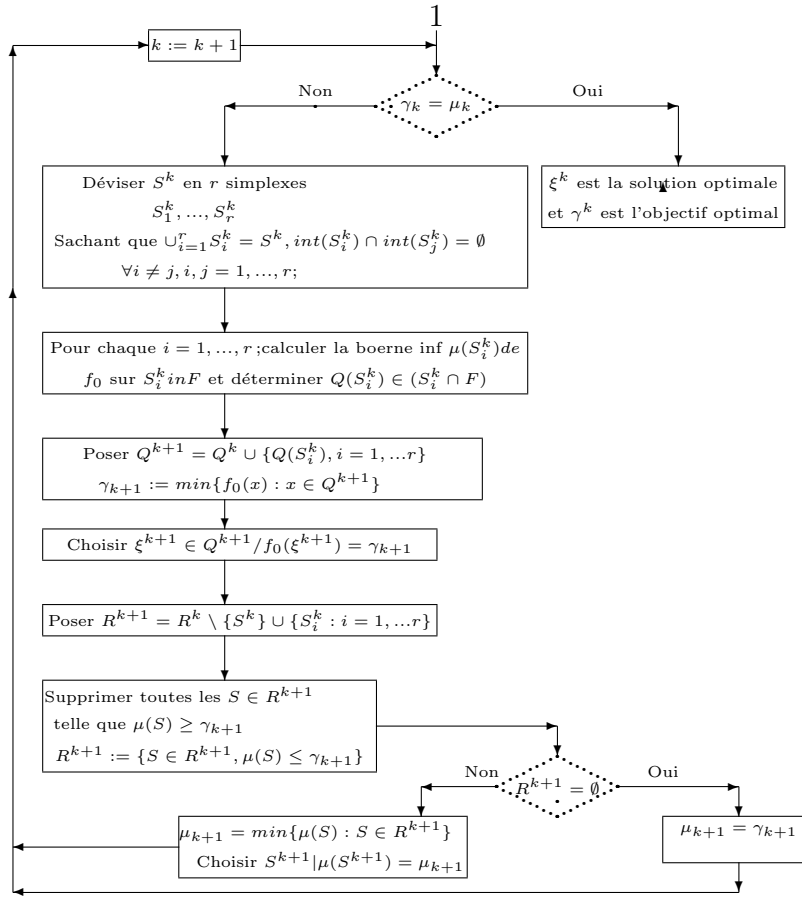


FIG. 5.10 – Organigramme itératif de l'algorithme 1

Annexe B

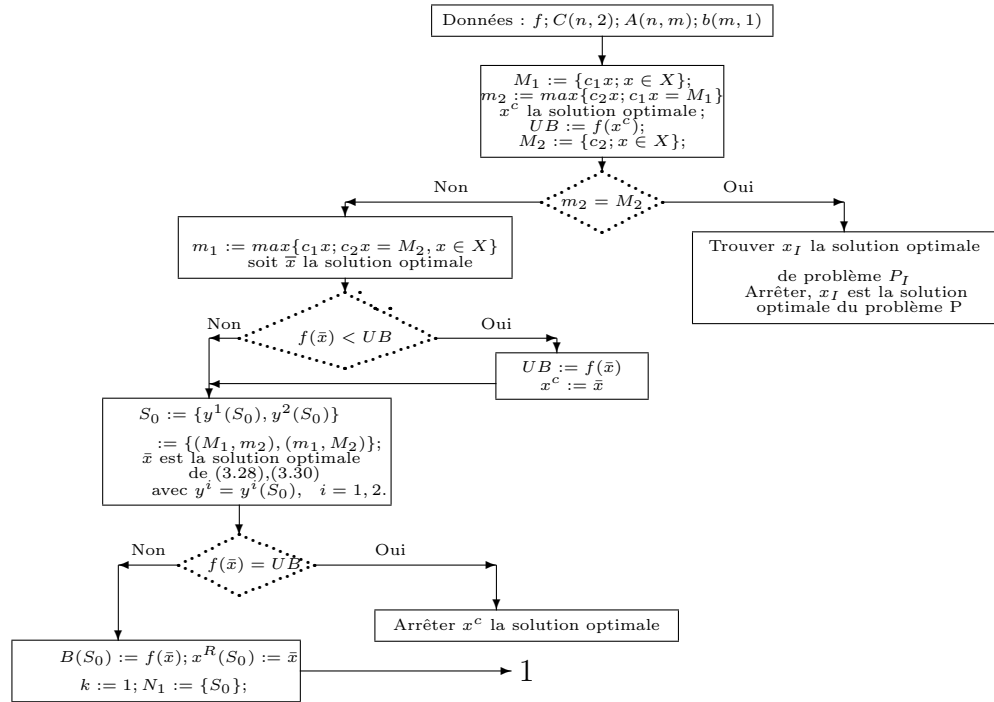


FIG. 5.11 – Etapes d’initialisation de l’algorithme 2

Annexe C

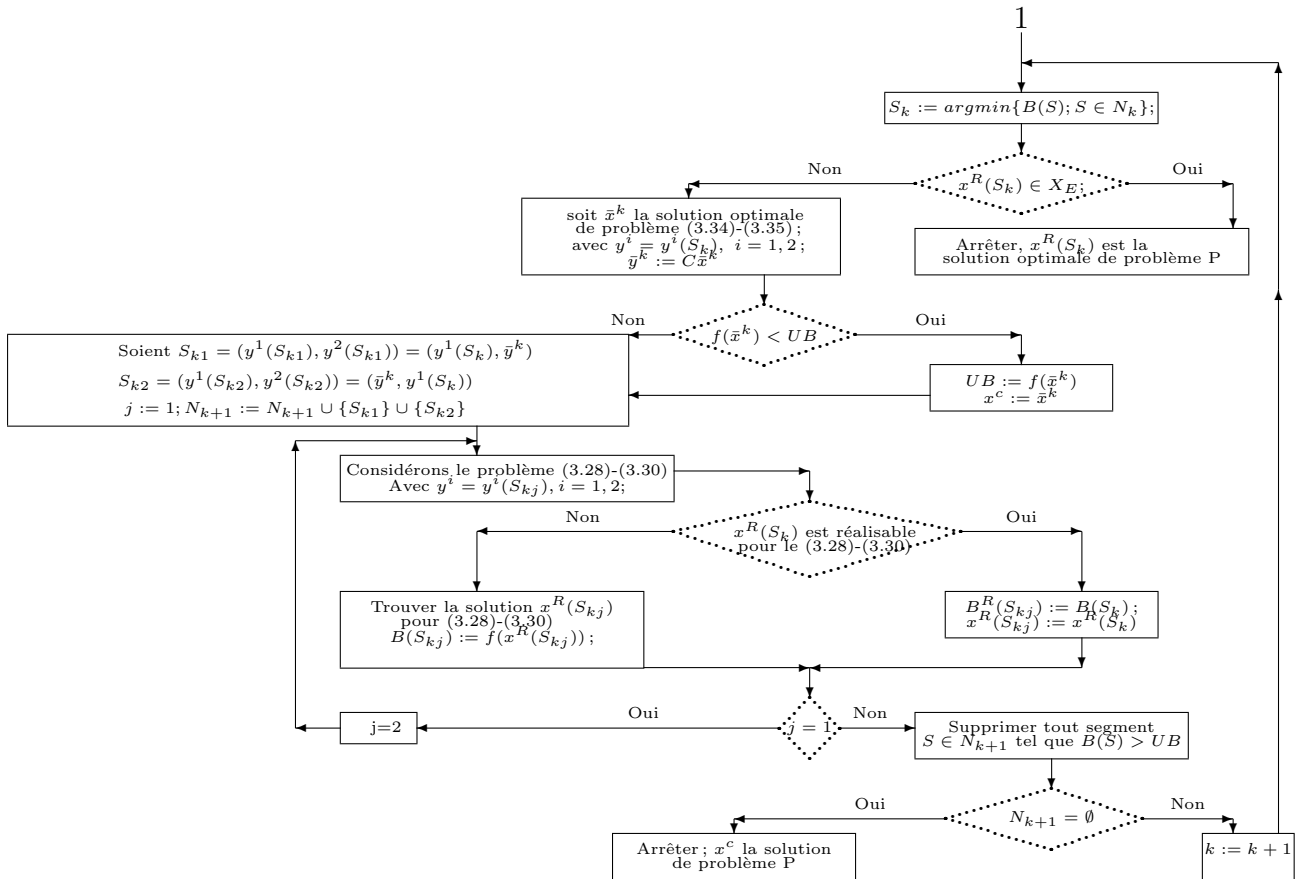


FIG. 5.12 – Organigramme itératif de l’algorithme 2

Annexe D