

# Table des matières

Résumé	8
Introduction générale	9
<b>I Optimisation mono-objectif</b>	<b>13</b>
<b>1 L'optimisation semi-définie positive (SDP)</b>	<b>14</b>
1.1 Préliminaires matriciels . . . . .	14
1.1.1 Notations . . . . .	14
1.1.2 Matrices semi-définies positives . . . . .	15
1.2 Optimisation SDP . . . . .	16
1.3 Définitions primale et duale du problème . . . . .	16
1.3.1 Le problème primal . . . . .	16
1.3.2 Le problème dual . . . . .	17
1.4 Réalisabilités primale et duale . . . . .	19
1.5 Exemples de modélisation SDP . . . . .	20
1.6 Programmation linéaire . . . . .	21
<b>2 Minimisation des polynômes homogènes de degré 2</b>	<b>23</b>
2.1 Introduction . . . . .	23
2.2 Notions de base sur les polynômes . . . . .	23
2.2.1 Notations . . . . .	23
2.2.2 Propriétés . . . . .	24
2.2.3 Degré d'un polynôme . . . . .	25
2.2.4 Polynômes remarquables . . . . .	26
2.3 Minimisation des polynômes homogènes de degré 2 . . . . .	27
2.4 Minimisation d'un polynôme homogène d'ordre 2 sur un simplexe standard . . . . .	27

<b>3</b>	<b>La méthode adaptée pour la résolution d'un programme quadratique convexe</b>	<b>29</b>
3.1	Introduction . . . . .	29
3.2	Programmation quadratique convexe . . . . .	29
3.2.1	Les formes quadratiques . . . . .	29
3.2.2	Notion de convexité . . . . .	31
3.2.3	Problème quadratique convexe (PQC) . . . . .	32
3.3	Position du problème et définitions . . . . .	32
3.4	Formule d'accroissement de la fonction objectif . . . . .	33
3.5	Critère d'optimalité . . . . .	34
3.6	Critère de suboptimalité . . . . .	35
3.7	Méthode de résolution . . . . .	35
3.7.1	Construction de la direction d'amélioration adaptée $l$ . . . . .	36
3.7.2	Calcul du pas maximal $\theta^0$ . . . . .	37
3.7.3	Estimation de suboptimalité . . . . .	38
3.7.4	Changement de support . . . . .	38
3.8	Algorithme de résolution . . . . .	39
3.9	Exemple numérique . . . . .	40
<b>4</b>	<b>La méthode adaptée pour la résolution d'un programme linéaire</b>	<b>48</b>
4.1	Introduction . . . . .	48
4.2	Position du problème et définitions . . . . .	48
4.2.1	Position du problème . . . . .	48
4.2.2	Définitions . . . . .	49
4.3	Critère d'optimalité . . . . .	50
4.4	Critère de suboptimalité . . . . .	52
4.5	Méthode de résolution . . . . .	53
4.5.1	Construction d'une nouvelle solution réalisable . . . . .	53
4.5.2	Calcul de $\beta(\bar{x}, J_B)$ . . . . .	54
4.5.3	Changement de support . . . . .	55
4.6	Algorithme de la méthode . . . . .	56
4.7	Étude d'un exemple numérique . . . . .	58
4.8	Expériences numériques . . . . .	61
4.9	Conclusions . . . . .	67
<b>II</b>	<b>Optimisation multiobjectif</b>	<b>68</b>
<b>5</b>	<b>Préliminaires et considérations théoriques</b>	<b>69</b>
5.1	Introduction . . . . .	69
5.2	Concepts et terminologie . . . . .	69

5.2.1	Les alternatives et les critères . . . . .	69
5.2.2	Paradigme des problèmes de décision multiobjectif . . . . .	70
5.2.3	Les différentes problématiques multiobjectif . . . . .	71
5.3	Les structures de préférences . . . . .	71
5.3.1	Les relations binaires et les ordres . . . . .	71
5.3.2	Les structures de préférences . . . . .	72
5.4	L'espace de décision et l'espace objectif . . . . .	73
5.5	L'optimisation multiobjectif . . . . .	74
5.6	Les concepts d'optimalité . . . . .	76
5.6.1	Optimalité de Pareto . . . . .	76
5.6.2	Optimalité au sens de Slater . . . . .	77
5.7	Définitions . . . . .	78
5.7.1	La surface du compromis . . . . .	78
5.7.2	Représentation de la surface du compromis sur Matlab pour un problème à deux fonctions objectif . . . . .	79
5.7.3	Fonction utilité . . . . .	80
5.7.4	Point idéal . . . . .	81
5.7.5	Table des gains . . . . .	81
5.7.6	Point nadir . . . . .	82
<b>6</b>	<b>Quelques méthodes d'optimisation multiobjectif</b>	<b>84</b>
6.1	La méthode de pondération . . . . .	84
6.2	La méthode du compromis . . . . .	86
6.3	Les méthodes hybrides . . . . .	87
6.4	La méthode dite du "but à atteindre" . . . . .	87
6.5	La méthode dite du "but programmé" . . . . .	88
6.6	La méthode du Simplexe Multiobjectif . . . . .	89
6.6.1	Trouver une solution de base réalisable : . . . . .	92
6.6.2	Trouver un point extrême efficace : . . . . .	92
6.6.3	Générer l'ensemble des points extrêmes efficaces : . . . . .	92
6.6.4	Exemple numérique . . . . .	96
<b>7</b>	<b>La méthode adaptée du support pour la résolution d'un pro- gramme linéaire multiobjectif</b>	<b>100</b>
7.1	Introduction . . . . .	100
7.2	Position du problème et définitions générales . . . . .	100
7.3	Définition des solutions efficaces et leurs propriétés . . . . .	102
7.4	Définition des solutions $\epsilon$ -efficaces et leurs propriétés . . . . .	103
7.5	Procédure de recherche d'un point efficace initial . . . . .	104
7.5.1	Trouver un point efficace initial en utilisant la méthode d'Iser- mann . . . . .	104

7.5.2	Trouver un point efficace initial en utilisant la procédure de Benson-Radjef-Bibi . . . . .	106
7.6	Recherche des solutions efficaces . . . . .	107
7.6.1	Test d'efficacité d'une variable non basique . . . . .	107
7.6.2	Construction de la nouvelle solution efficace . . . . .	109
7.7	L'algorithme de la méthode . . . . .	112
7.8	Exemple numérique . . . . .	113
7.9	Exemple sur la modélisation mathématique de la chaîne logistique .	119
7.9.1	Présentation de la laiterie . . . . .	119
7.9.2	Caractéristiques des produits de la laiterie . . . . .	119
7.9.3	Résolution du modèle à l'aide de la méthode adaptée du support . . . . .	122
7.10	Conclusion . . . . .	123
<b>Conclusion générale</b>		<b>125</b>
<b>Bibliographie</b>		<b>125</b>
<b>Annexe</b>		<b>133</b>
A	Introduction . . . . .	134
B	Utilisation du langage du logiciel Matlab . . . . .	134
C	Implémentation de la méthode adaptée multiobjectif sous Matlab .	134
C.1	Génération des programmes aléatoires . . . . .	134
C.2	Méthode de Benson . . . . .	135
C.3	Choix du critère $k_0$ . . . . .	136
C.4	Génération d'une solution $\epsilon$ -efficace . . . . .	136
C.5	Test d'efficacité de la solution générée . . . . .	137

# Résumé

Ce travail est consacré à la minimisation des polynômes et ce en utilisant la méthode adaptée du support. La particularité de cette méthode est qu'elle utilise un critère de suboptimalité qui nous permet d'arrêter l'algorithme avec une précision souhaitée pour trouver des solutions dites  $\epsilon$ -optimales. Cette méthode est efficace, rapide, simple et permet une réduction de temps dans l'ensemble du processus d'optimisation. En effet, une implémentation sous la version Matlab 8.1.0.604 a été développée et les résultats obtenus sont concluants. De ce fait, nous avons aussi pu concevoir une nouvelle méthode pour résoudre un programme linéaire multiobjectif dont les variables de décision sont bornées. Notre méthode nous permet de trouver les points efficaces et  $\epsilon$ -efficaces, et ce en exploitant le critère de suboptimalité de la méthode adaptée en mono-objectif. L'algorithme de cette méthode a été programmé et ses étapes sont présentées dans l'annexe, constituant ainsi une bibliothèque utilisable pour résoudre d'autres programmes.

**Mots clés :** Minimisation, polynôme homogène, variables bornées, optimisation multiobjectif, efficacité,  $\epsilon$ -efficacité.

# Introduction générale

Le problème de la minimisation des polynômes réels fait l'objet de nombreuses techniques de résolutions : utilisation de bases de Gröbner, calcul de valeurs propres, de discriminants ou de résultants  $\dots$  Ces techniques se révèlent difficiles à mettre en oeuvre en pratique. Ceci est dû à la complexité des algorithmes calculant le minimum d'un polynôme réel. Ces dernières années se sont développées de nouvelles méthodes, dites de relaxation, calculant un minorant du minimum recherché. Parmi elles, les techniques d'optimisation semi-définie positive (SDP) qui tiennent une bonne place, comme l'ont mis en avant P. Parrilo et B. Sturmfels dans leur article [65].

L'optimisation semi-définie positive [86] s'est beaucoup développée à partir des années 1990, du fait de plusieurs découvertes. D'une part, beaucoup de problèmes pratiques ont pu être définis au moyen de ce formalisme (en recherche opérationnelle) ou ont trouvé une formalisation SDP approchée, mais précise (en optimisation combinatoire, en optimisation algébrique). Par ailleurs ces problèmes peuvent être résolus efficacement par divers algorithmes : points intérieurs (algorithmes polynomiaux), lagrangien augmenté, méthode des faisceaux (algorithme d'optimisation non différentiable), etc. L'approche semi-définie, qui conduit à des relaxations convexes mais non-linéaires, a permis d'obtenir de remarquables résultats théoriques en approximation et devient à présent utilisable en pratique (tout comme l'optimisation linéaire qui en est un cas particulier).

La SDP a tout d'abord été rendue populaire grâce aux résultats spectaculaires qu'elle a permis d'obtenir en approximation polynomiale. C'est un cas particulier de la programmation convexe, et elle peut être vue comme une généralisation de la programmation linéaire. Elle a néanmoins été exclue dans un premier temps du champ pratique car peu de logiciels permettaient de résoudre efficacement les SDP.

La thèse a pour ambition d'évaluer ce qu'une méthode alternative comme la méthode adaptée du support [40] pourrait apporter en optimisation des polynômes en général ou en optimisation semi-définie positive.

Cette méthode est itérative ; son principe est le suivant : à partir d'une solution réalisable initiale et d'un support initial, chaque itération consiste à trouver

une direction et un pas le long de cette direction pour améliorer la valeur de la fonction objective sans quitter l'ensemble réalisable déterminé par les contraintes du problème. La solution réalisable initiale et le support initial pourraient être calculés indépendamment. En plus de cela, le point de départ initial n'est pas obligé d'être un point extrême tel que dans la méthode du simplexe. Gabasov et al. [40] utilise une approche appelée la méthode adaptée où il utilise une autre métrique dite adaptée. Il considère tous les indices non optimaux pour lesquels il construit une direction d'amélioration de la fonction d'objectif et le pas maximal le long de cette direction.

Cette méthode est considérée comme une méthode intermédiaire entre les méthodes du point intérieur et la méthode du simplexe. En effet, en utilisant la méthode adaptée, le point réalisable initial peut être un point extrême, un point intérieur ou n'importe quel point sur le bord.

La méthode adaptée [40] est habituellement utilisée pour résoudre des programmes linéaires ou des programmes quadratiques convexes avec des variables bornées. En outre, la méthode intègre un critère de suboptimalité qui permet d'arrêter l'algorithme avec une précision souhaitée pour trouver des solutions  $\epsilon$ -optimale. Ce critère pourrait être utile dans des applications pratiques.

En fait, nous allons voir dans le deuxième chapitre de cette thèse que minimiser un polynôme homogène revient à la minimisation d'une fonctionnelle quadratique convexe. Ainsi, il est possible d'utiliser la méthode adaptée du support [11, 38, 41] pour la minimisation des polynômes sur un ensemble de contraintes bornées. Les détails de cette méthode sont présentés dans le chapitre 3.

Nous verrons au chapitre 1 que la programmation semi-définie positive est une généralisation de la programmation linéaire, et il existe des liens étroits entre elles. On a alors consacré le chapitre 4 pour présenter la méthode adaptée que nous avons développée pour résoudre un programme linéaire standard [23]. Cela n'est pas restrictif sachant qu'en pratique, la forme standard apparaît naturellement et très souvent.

Toutefois, la principale difficulté que l'on rencontre en optimisation mono-objectif vient du fait que modéliser un problème sous la forme d'une équation unique peut être une tâche difficile. Avoir comme but de se ramener à une seule fonction objectif peut aussi biaiser la modélisation. L'optimisation multiobjectif autorise ces degrés de liberté qui manquaient en optimisation mono-objectif. Cette souplesse n'est pas sans conséquences sur la démarche à suivre pour chercher un optimum à notre problème enfin modélisé. La recherche ne nous donnera plus une solution unique mais une multitude de solutions.

La première notion d'optimalité multicritère a été introduite par Edgeworth en

1881, elle a été utilisée de manière plus formelle par l'économiste italien V. Pareto (1848-1923). Cette notion est appelée solution efficace, optimale selon Pareto ou encore solution non-dominée, ces solutions sont appelées solutions de Pareto et l'ensemble de solutions que l'on obtient à la fin de la recherche est la surface de compromis. Par la suite, Kuhn et Tucker (1951) ont donné des résultats théoriques concernant les problèmes d'optimisation multiobjectif.

Comme nous allons le voir tout au long de notre travail, l'optimisation multiobjectif n'est pas une tâche facile, il faut choisir une méthode d'optimisation pour résoudre ce type de problème. Cette dernière tâche étant délicate en optimisation mono-objectif, on déduit qu'elle est d'une difficulté plus grande encore dans l'optimisation multiobjectif. En effet, les problèmes d'optimisation multiobjectif sont très variés et correspondent à des situations de décision très différentes. Contrairement aux problèmes d'optimisation mono-objectif où l'ordre usuel sur  $\mathbb{R}$  est total, pour un problème multiobjectif, l'ordre naturellement introduit sur l'espace des critères n'est que partiel, ce qui traduit l'impossibilité de comparer les solutions entre elles. L'ensemble des points de l'espace de recherche tels qu'il n'existe aucun point qui est strictement meilleur que tous les autres simultanément sur tous les critères est appelé front de Pareto du problème. Il s'agit de l'ensemble des meilleurs compromis réalisables entre les objectifs, et le but de l'optimisation est d'identifier cet ensemble de compromis optimal entre les critères.

Dans cette thèse, nous nous sommes justement intéressés à ce type de problèmes multiobjectif. Nous avons alors développé une nouvelle méthode [22], pour résoudre les problèmes linéaires multiobjectif, en exploitant le principe de la méthode adaptée.

Nous avons alors divisé notre thèse en deux grandes parties :

La première partie est consacrée à l'optimisation mono-objectif, nous utilisons l'optimisation semi-définie pour construire des solutions exactes pour des problèmes d'optimisation linéaires et quadratiques. Cette partie comporte quatre chapitres :

Dans le premier chapitre, nous allons donner un résumé sur les éléments de l'optimisation semi-définie [57].

Dans le deuxième chapitre, nous allons aborder le problème de minimisation d'un polynôme homogène de degré 2 et dans le troisième chapitre, nous allons le résoudre en utilisant la méthode adaptée du support pour la résolution d'un problème quadratique convexe à variables bornées.

Enfin, dans le quatrième chapitre, notre travail vise à proposer une approche basée sur la méthode adaptée du support pour résoudre des programmes linéaires mono-objectif avec des variables non négatives.

La deuxième partie est consacrée à la résolution des problèmes multicritères. Cette partie comporte trois chapitres :



Dans le cinquième chapitre, nous allons donner un rappel des notions fondamentales de la théorie de la décision multicritère et une présentation sommaire des modèles utilisés.

Dans le sixième chapitre, nous allons exposer quelques méthodes de résolution d'un programme multiobjectif.

Dans le septième et dernier chapitre, nous proposons d'étudier un programme linéaire multicritère à variables bornées, et ce en utilisant la méthode adaptée du support [22].

Nous avons clôturé notre thèse par une conclusion générale et quelques perspectives de recherche.

**Première partie**

**Optimisation mono-objectif**

# Chapitre 1

## L'optimisation semi-définie positive (SDP)

### 1.1 Préliminaires matriciels

Dans cette section, nous rappelons quelques propriétés des matrices symétriques (en particulier semi-définies positives), et effectuons quelques rappels de calcul matriciel. Ceci nous sera utile dans l'étude de la programmation semi-définie positive.

#### 1.1.1 Notations

On note :

$$\mathcal{S}^n(\mathbb{R}) = \{M \in \mathbb{R}^{n \times n} : M^T = M\},$$

l'espace vectoriel des matrices réelles d'ordre  $n$  symétriques.

On munit  $\mathcal{S}^n(\mathbb{R})$  d'une structure euclidienne en y définissant le produit scalaire standard

$$\langle \cdot, \cdot \rangle : (M, N) \in \mathcal{S}^n(\mathbb{R}) \times \mathcal{S}^n(\mathbb{R}) \mapsto \langle M, N \rangle = \text{tr}(MN) = \sum_{i=1}^n \sum_{j=1}^n M_{ij}N_{ij},$$

où  $\text{tr}(MN)$  désigne la trace d'une matrice carrée du produit matriciel  $MN$ .

Toute forme quadratique  $x^T Ax$  peut donc s'écrire  $\langle A, xx^T \rangle$ . Une matrice  $A$  de  $\mathcal{S}^n(\mathbb{R})$  est positive (resp. définie positive) si et seulement si pour tout  $z$  non nul dans  $\mathbb{R}^n$ , on a  $z^T Az \geq 0$  (resp.  $z^T Az > 0$ ).

On note :

$$\mathcal{S}_+^n(\mathbb{R}) = \{M \in \mathcal{S}^n(\mathbb{R}) : M \succcurlyeq 0\},$$

la partie de  $\mathcal{S}^n(\mathbb{R})$  formée des matrices semi-définies positives (c'est ce que signifie la notation  $M \succcurlyeq 0$ ) et on note :

$$\mathcal{S}_{++}^n(\mathbb{R}) = \{M \in \mathcal{S}^n(\mathbb{R}) : M \succ 0\},$$

la partie de  $\mathcal{S}^n(\mathbb{R})$  formée des matrices définies positives (c'est ce que signifie la notation  $M \succ 0$ ).

### 1.1.2 Matrices semi-définies positives

**Définition 1.1.1.** Soit  $A \in \mathcal{S}_n(\mathbb{R})$ .

- $A$  est dite semi-définie positive, et l'on notera  $A \in \mathcal{S}_n^+(\mathbb{R})$  ou  $A \succeq 0$  si

$$\forall x \in \mathbb{R}^n, x^T A x \geq 0.$$

- $A$  est dite définie positive, et l'on notera  $A \in \mathcal{S}_n^{++}(\mathbb{R})$  ou  $A \succ 0$  si

$$\forall x \in \mathbb{R}^n \setminus \{0\}, x^T A x > 0.$$

**Proposition 1.1.1.** Soit  $B \in M_n(\mathbb{R})$  inversible. Alors

$$A \in \mathcal{S}_n^+(\mathbb{R}) \Leftrightarrow B^T A B \in \mathcal{S}_n^+(\mathbb{R})$$

et

$$A \in \mathcal{S}_n^{++}(\mathbb{R}) \Leftrightarrow B^T A B \in \mathcal{S}_n^{++}(\mathbb{R}).$$

**Définition 1.1.2.** Un ensemble  $C \in \mathbb{R}^m$  est un cône si

$$x, y \in C \Rightarrow \forall \lambda \geq 0, \lambda(x + y) \in C.$$

Un cône est dit pointé si  $C \cap (-C) = \{0\}$ .

**Proposition 1.1.2.**  $\mathcal{S}_n^+(\mathbb{R})$  est un cône fermé dans  $\mathcal{S}_n(\mathbb{R})$ .

*Remarque 1.1.1.*  $\mathcal{S}_n^{++}(\mathbb{R})$  n'est pas un cône fermé car  $\{0\} \notin \mathcal{S}_n^{++}(\mathbb{R})$ .

*Remarque 1.1.2.* La structure de cône de  $\mathcal{S}_n^+(\mathbb{R})$  induit un ordre partiel sur  $\mathcal{S}_n(\mathbb{R})$ , dit ordre de Löwner :

$$A \succeq B \Leftrightarrow A - B \in \mathcal{S}_n^+(\mathbb{R}).$$

On notera de la même manière :

$$A \succ B \Leftrightarrow A - B \in \mathcal{S}_n^{++}(\mathbb{R}).$$

**Proposition 1.1.3.** Soient  $A, B \in \mathcal{S}_n^{++}(\mathbb{R})$ . Alors :

1.  $A + B \succ B$ .
2.  $Tr(AB) \leq Tr(A)Tr(B)$ .
3. Les valeurs propres de  $AB$  sont positives.

## 1.2 Optimisation SDP

Un problème d'optimisation SDP ou semi-définie positive (on recycle ainsi le sigle SDP anglais qui abrège semidefinite programming) est un problème d'optimisation convexe, qui étend celui de l'optimisation linéaire. Le problème a en effet une structure semblable, mais avec une inconnue qui est une matrice symétrique (au lieu d'être un vecteur de nombres réels) que l'on impose d'être semi-définie positive (au lieu de demander que les nombres réels soient positifs). Comme en optimisation linéaire, le critère à minimiser est linéaire et l'inconnue doit également satisfaire une contrainte affine.

Dans ce chapitre, nous donnons succinctement quelques éléments de la programmation semi-définie. L'accent est mis essentiellement sur les problèmes de programmation semi-définie sous forme standard. Cela n'est pas restrictif sachant qu'en pratique, la forme standard apparaît naturellement et très souvent, et surtout nous allons expliciter la relation avec l'optimisation linéaire.

## 1.3 Définitions primale et duale du problème

### 1.3.1 Le problème primal

Un problème d'optimisation SDP s'écrit de la manière suivante

$$P_{(SDP)} \quad \begin{cases} \inf_{X \in \mathcal{S}^n(\mathbb{R})} \langle C, X \rangle, \\ A(X) = b, \\ X \succeq 0, \end{cases}$$

où  $C \in \mathcal{S}^n(\mathbb{R})$ ,  $A : \mathcal{S}^n(\mathbb{R}) \rightarrow \mathbb{R}^m$  est une application linéaire et  $b \in \mathbb{R}^m$ . Il s'agit donc de minimiser un critère linéaire sur l'intersection du cône des matrices semi-définies positives et d'un sous-espace affine. C'est ce qu'on appelle **la formulation primale** d'un problème SDP.

Le critère et la contrainte d'égalité sont linéaires (ou affines), mais la contrainte d'appartenance au cône  $\mathcal{S}_+^n(\mathbb{R})$  est non linéaire, éventuellement non différentiable. Ainsi, le critère du problème étant linéaire et son ensemble admissible étant convexe, le problème SDP est en général considéré comme un problème d'optimisation convexe.

Si les problèmes d'optimisation SDP ont une structure assez semblable à celle de l'optimisation linéaire, qui les rend très vite familiers, les résultats que l'on connaît en optimisation linéaire ne s'étendent pas à l'optimisation SDP, on ne peut guère en être étonné, vu la généralité du formalisme. Il faudra y prendre garde.

On note  $\text{val}(P_{SDP})$  et  $\text{sol}(P_{SDP})$  la valeur optimale du problème ( $P_{SDP}$ ) et son ensemble de solution. On peut représenter l'application linéaire  $A$  au moyen de  $m$  matrices  $A_i \in \mathcal{S}^n(\mathbb{R})$ .

**Proposition 1.3.1.** *Un programme semi-défini est un programme convexe.*

**Exemple 1.3.1.** Considérons le problème suivant :

$$\begin{cases} \min x_{11} + 4x_{12} + 6x_{13} + 9x_{22} + 7x_{33}, \\ x_{11} + 2x_{13} + 3x_{22} + 14x_{23} + 5x_{33} = 11, \\ 4x_{12} + 16x_{13} + 6x_{22} + 4x_{33} = 19, \\ X \succeq 0. \end{cases} \quad (1.1)$$

qui est un programme semi-défini avec

$$b = \begin{pmatrix} 11 \\ 19 \end{pmatrix}, C = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 9 & 0 \\ 3 & 0 & 7 \end{pmatrix}, A_1 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 3 & 7 \\ 1 & 7 & 5 \end{pmatrix},$$

$$A_2 = \begin{pmatrix} 0 & 2 & 8 \\ 2 & 6 & 0 \\ 8 & 0 & 4 \end{pmatrix}, X = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix}.$$

### 1.3.2 Le problème dual

On se donne un produit scalaire sur  $\mathbb{R}^m$ , également noté  $\langle \cdot, \cdot \rangle$ , et l'on introduit l'opérateur  $A^* : \mathbb{R}^m \rightarrow \mathcal{S}^n(\mathbb{R})$  adjoint de  $A$ , qui est défini par

$$\forall X \in \mathcal{S}^n(\mathbb{R}), \forall y \in \mathbb{R}^m : \langle A(X), y \rangle = \langle X, A^*(y) \rangle.$$

La méthode la plus simple pour obtenir un dual de ( $P_{SDP}$ ) est d'utiliser la dualisation lagrangienne de sa contrainte d'égalité. On utilise donc le Lagrangien  $\ell : \mathcal{S}_+^n(\mathbb{R}) \times \mathbb{R}^m \rightarrow \mathbb{R}$  défini par

$$\ell(X, y) = \langle C, X \rangle - \langle y, A(X) - b \rangle$$

et on écrit ( $P_{SDP}$ ) comme un inf-sup :

$$\inf_{X \in \mathcal{S}_+^n(\mathbb{R})} \sup_{y \in \mathbb{R}^m} \ell(X, y).$$

Le dual s'obtient alors en inversant l'infimum et le supremum

$$\sup_{y \in \mathbb{R}^m} \inf_{X \in \mathcal{S}_+^n(\mathbb{R})} \ell(X, y).$$

Après calculs, le problème dual de  $(P_{SDP})$  obtenu par ce procédé consiste donc à trouver  $(y, S) \in \mathbb{R}^m \times \mathcal{S}^n(\mathbb{R})$  solution de

$$(D_{SDP}) \quad \begin{cases} \sup_{(y,S) \in \mathbb{R}^m \times \mathcal{S}^n(\mathbb{R})} \langle b, y \rangle, \\ A^*(y) + S = C, \\ S \succcurlyeq 0. \end{cases}$$

On note  $\text{val}(D_{SDP})$  et  $\text{sol}(D_{SDP})$  la valeur optimale du problème  $(D_{SDP})$  et son ensemble de solution.

L'écart entre les valeurs optimales primale et duale est ce que l'on appelle le "saut de dualité" :

$$\text{Le saut de dualité} = \text{val}(P_{SDP}) - \text{val}(D_{SDP}).$$

On dit qu'il n'y a pas de saut de dualité si le saut de dualité est nul.

Si l'on utilise le produit scalaire euclidien dans  $\mathbb{R}^m$  et si l'on prend la représentation ci-dessus de  $A$ , son adjoint  $A^*$  s'écrit

$$A^*(y) = \sum_{i=1}^m A_i y_i.$$

Dans ce cas, le problème dual peut se voir comme celui cherchant à maximiser une forme linéaire en  $y$  sur  $\mathcal{S}^n(\mathbb{R})$ , tout en imposant qu'une combinaison affine de matrices (avec coefficients  $y_i$ ) soit semi-définie positive :

$$C - \sum_{i=1}^m A_i y_i \succcurlyeq 0.$$

Cette dernière relation est ce que l'on appelle une inégalité matricielle linéaire (ou affine), notée IML en abrégé.

Le théorème suivant donne quelques conséquences simples de la dualisation lagrangienne de  $(P_{SDP})$ . Le point 1 est connu sous le nom de **dualité faible**. L'écart  $\text{val}(P_{SDP}) - \text{val}(D_{SDP})$  entre valeurs optimales primale et duale est appelé **le saut de dualité**. Le point 2 montre que  $\langle X, S \rangle$  est l'écart entre valeurs primale et duale pour un triplet admissible  $(X, y, S) \in \mathcal{A}$ . Le point 3 donne une condition suffisante d'optimalité élémentaire, mais bien utile.

**Théorème 1.3.1** (Conséquences de la dualisation lagrangienne).

1.  $\text{val}(D_{SDP}) \leq \text{val}(P_{SDP})$ ,
2.  $(X, y, S) \in \mathcal{A} \Rightarrow \langle C, X \rangle - \langle b, y \rangle = \langle X, S \rangle \geq 0$ ,
3.  $(X, y, S) \in \mathcal{A}$  et  $\langle X, S \rangle = 0 \Rightarrow X \in \text{sol}(P_{SDP})$  et  $(y, S) \in \text{sol}(D_{SDP})$ . La réciproque du point 3 est fautive en général, mais on verra qu'elle a lieu si  $\mathcal{A}^s \neq \emptyset$ . Lorsqu'elle a lieu,  $(X, y)$  est un point-selle du lagrangien  $\ell$  défini ci-dessus sur  $\mathcal{S}_+^n(\mathbb{R}) \times \mathbb{R}^m$ .

**Exemple 1.3.2.** Dans l'exemple 1.3.1, le problème dual s'écrit sous la forme suivante :

$$\begin{cases} \max 11y_1 + 19y_2, \\ y_1 \begin{pmatrix} 1 & 0 & 1 \\ 0 & 3 & 7 \\ 1 & 7 & 5 \end{pmatrix} + y_2 \begin{pmatrix} 0 & 2 & 8 \\ 2 & 6 & 0 \\ 8 & 0 & 4 \end{pmatrix} + Z = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 9 & 0 \\ 3 & 0 & 7 \end{pmatrix}, \\ Z \succeq 0. \end{cases} \quad (1.2)$$

On peut réécrire ce problème sous la forme :

$$\begin{cases} \max 11y_1 + 19y_2, \\ \begin{pmatrix} 1 - y_1 & 2 - 2y_2 & 3 - y_1 - 8y_2, \\ 2 - 2y_2 & 9 - 3y_1 - 6y_2 & -7y_1, \\ 3 - y_1 - 8y_2 & -7y_1 & 7 - 5y_1 - 4y_2, \end{pmatrix} \succeq 0. \end{cases} \quad (1.3)$$

## 1.4 Réalisabilités primale et duale

Nous nous intéressons ici à la question de la réalisabilité des problèmes primal et dual. Quand peut-on trouver une matrice  $X \succcurlyeq 0$  telle que  $A(X) = b$ ? Quand peut-on trouver un vecteur  $y \in \mathbb{R}^m$  tel que  $A^*(y) \preccurlyeq C$ ?

On note

$$\begin{aligned} \mathcal{A}_P &= \{X \in \mathcal{S}^n(\mathbb{R}) : A(X) = b, X \succcurlyeq 0\}, \\ \mathcal{A}_P^s &= \{X \in \mathcal{S}^n(\mathbb{R}) : A(X) = b, X \succ 0\}, \end{aligned}$$

les ensembles des matrices admissibles et strictement admissibles de  $(P_{SDP})$ .

On note

$$\begin{aligned} \mathcal{A}_D &= \{(y, S) \in \mathbb{R}^m \times \mathcal{S}^n(\mathbb{R}) : A^*(y) + S = C, S \succcurlyeq 0\}, \\ \mathcal{A}_D^s &= \{(y, S) \in \mathbb{R}^m \times \mathcal{S}^n(\mathbb{R}) : A^*(y) + S = C, S \succ 0\}, \end{aligned}$$

les ensembles des couples admissibles et strictement admissibles de  $(D_{SDP})$ . On note enfin

$$\mathcal{A} = \mathcal{A}_P \times \mathcal{A}_D \quad \text{et} \quad \mathcal{A}^s = \mathcal{A}_P^s \times \mathcal{A}_D^s.$$

Le résultat suivant résume cette discussion.

### Théorème 1.4.1.

- Les contraintes primales sont quasi-réalisables si, et seulement si,  $\langle b, y \rangle \geq 0$  pour tout  $y \in \mathbb{R}^m$  tel que  $A^*(y) \succcurlyeq 0$ .
- Les contraintes duales sont quasi-réalisables si, et seulement si,  $\langle C, X \rangle \geq 0$  pour tout  $X \in \mathcal{S}_+^n(\mathbb{R})$  tel que  $A(X) = 0$ .



La quasi-réalisabilité des contraintes primales et duales n'est pas une propriété très forte (elle n'assure même pas leur réalisabilité). Numériquement, il est certainement préférable d'avoir une réalisabilité plus robuste, insensible à de petites perturbations du second membre. On définit donc les concepts suivants. On dit que les contraintes primales sont fortement réalisables si

$$b \in \text{int } A(\mathcal{S}_+^n(\mathbb{R})),$$

où l'opérateur «int» désigne la prise de l'intérieur. De même, on dit que les contraintes duales sont fortement réalisables si

$$C \in \text{int } (A^*(\mathbb{R}^m) + \mathcal{S}_+^n(\mathbb{R})).$$

**Théorème 1.4.2** (Théorème réalisabilités primale et duale fortes).

- Les contraintes primales sont fortement réalisables si, et seulement si,  $A$  est surjective et  $\mathcal{A}_P^s \neq \emptyset$ .
- Les contraintes duales sont fortement réalisables si, et seulement si,  $\mathcal{A}_D^s \neq \emptyset$ .

La théorème suivant donne des conditions suffisantes pour que l'ensemble des solutions primales (resp. duales) soit non vide et borné.

**Théorème 1.4.3** (Existence de solutions).

Si  $\mathcal{A}_P \times \mathcal{A}_D^s \neq \emptyset$ , alors  $\text{sol}(P_{SDP})$  est non vide et borné et il n'y a pas de saut de dualité.

Si  $\mathcal{A}_P^s \times \mathcal{A}_D \neq \emptyset$  et si  $A$  est surjective, alors  $\text{sol}(D_{SDP})$  est non vide et borné et il n'y a pas de saut de dualité.

Si  $\mathcal{A}^s \neq \emptyset$  et si  $A$  est surjective, alors  $\text{sol}(P_{SDP})$  et  $\text{sol}(D_{SDP})$  sont non vides et bornés et il n'y a pas de saut de dualité.

## 1.5 Exemples de modélisation SDP

Beaucoup de problèmes "convexes" ont une formulation SDP. L'intérêt d'exhiber une telle formulation est de montrer que l'on peut les résoudre par des algorithmes polynomiaux, souvent efficaces. Le choix des exemples qui suivent est motivé par le but de montrer leur diversité.

- Optimisation linéaire.
- Optimisation quadratique convexe.
- Minimisation de la valeur propre maximale.
- Minimisation de la norme matricielle euclidienne.
- Positivité de polynômes.
- Optimisation robuste.
- Relaxation SDP de problèmes non convexes.

## 1.6 Programmation linéaire

En pratique, personne ne veut transformer un programme linéaire en un programme (SDP) pour la résolution numérique, mais du point de vue théorique, un programme linéaire peut être considéré comme un cas particulier de problème (SDP). En effet, un programme linéaire (PL) est un problème sous la forme :

$$\begin{cases} \min z(x) = c^T x, \\ Ax = b, \\ x \geq 0. \end{cases} \quad (1.4)$$

où  $x$  et  $c \in \mathbb{R}^n$ ,  $b \in \mathbb{R}^m$ ,  $A \in \mathbb{R}^{m \times n}$ .

Pour transformer le programme linéaire en un programme (SDP), on définit :

$$X := \text{Diag}(x), \quad A_0 := \text{Diag}(c) \text{ et } A_i := \text{Diag}(A_i, :), \forall i \in \{1, \dots, m\},$$

où  $A_i$  désigne la  $i_{\text{ème}}$  ligne de la matrice  $A$ . Nous réécrivons alors la fonction objectif linéaire sous la forme :

$$c^T x = \langle C, X \rangle,$$

et pour les contraintes, on a :

$$Ax = b \Leftrightarrow \langle A_i, X \rangle = b_i, i = 1, \dots, m,$$

ainsi pour les contraintes  $x_i \geq 0$ , on a

$$x_i \geq 0, \forall i \in \{1, \dots, m\} \Leftrightarrow X \in S_+^n \Leftrightarrow X \succeq 0.$$

Ainsi, le programme linéaire (PL) de la variable  $x$  peut être réécrit sous la forme standard d'un programme (SDP) de la variable matricielle  $X$  comme suit :

$$\begin{cases} \inf_{X \in S^n} \langle C, X \rangle, \\ A(X) = b, \\ X \succeq 0, \end{cases}$$

Notons tout d'abord qu'en optimisation linéaire, c'est-à-dire pour les problèmes de type :

$$(PL) \begin{cases} \min c^T x, \\ Ax = b, \\ x \geq 0. \end{cases} \quad \text{et} \quad (PD) \begin{cases} \max by, \\ A^T y + z = c, \\ z \geq 0. \end{cases}$$

**Lemme 1.6.1** (Lemme de Farkas). *Soit  $A$  une matrice de réels d'ordre  $m \times n$  et  $b$  un vecteur-colonne d'ordre  $m$ , alors un et un seul des systèmes linéaires suivants a une solution :*

- le système  $Ax = b$  pour  $x$  un vecteur d'ordre  $n$  vérifiant  $x \geq 0$ ,
- le système  $A^T y \geq 0$  pour  $y$  un vecteur d'ordre  $m$  vérifiant  $b^T y < 0$ .

Les résultats d'existence de solution et de dualité forte assurent que les propriétés suivantes sont équivalentes :

- Les problèmes primal (PL) et dual (DL) sont réalisables (domaines de contraintes non vides),
- (PL) est réalisable et borné (i.e., la valeur optimale est finie),
- (DL) est réalisable et borné,
- (PL) admet une solution optimale (primale),
- (DL) admet une solution optimale (duale).

De plus, dans chacun de ces cas, il n'y a pas de saut de dualité. L'absence de ce dernier et l'existence de solutions optimales peuvent être obtenus grâce à l'existence de points strictement réalisables pour le dual et le primal.

Dans le chapitre 4, nous allons proposer une approche basée sur la méthode adaptée du support pour résoudre ce type de problème, cette approche intègre un critère de suboptimalité qui permet d'arrêter l'algorithme avec une précision souhaitée pour trouver des solutions  $\epsilon$ -optimales.

# Chapitre 2

## Minimisation des polynômes homogènes de degré 2

### 2.1 Introduction

Dans ce chapitre, nous allons aborder le problème de minimisation d'un polynôme homogène de degré  $n = 2$ , et le résoudre en utilisant la méthode adaptée du support pour la résolution d'un problème quadratique convexe à variables bornées.

### 2.2 Notions de base sur les polynômes

Cette section présente les définitions et les propriétés algébriques des polynômes sur un anneau quelconque.

#### 2.2.1 Notations

On se donne  $n$  lettres  $X_1, \dots, X_n$  et un anneau  $A$  unitaire et commutatif, dans ce contexte les lettres  $X_1, \dots, X_n$  sont appelées indéterminées ou variables.

**Définition 2.2.1** (Polynôme). Un polynôme  $P$  en  $X_1, \dots, X_n$  à coefficients dans  $A$  est une combinaison linéaire finie de monômes :

$$\sum_{i_1, \dots, i_n \geq 0} a_{i_1, \dots, i_n} X_1^{i_1} \cdots X_n^{i_n},$$

où  $a_{i_1, \dots, i_n} \in A$  sont presque tous nuls, chaque produit  $a_{i_1, \dots, i_n} X_1^{i_1} \cdots X_n^{i_n}$  est appelé un terme du polynôme. L'ensemble de ces polynômes est noté  $A[X_1, \dots, X_n]$ .

**Exemple 2.2.1.** Pour  $A = \mathbb{Z}$ , l'expression  $P = 1 + X_1^2 + 2X_1^2 - 3X_1X_2 - 5X_3$  est un polynôme en  $X_1, X_2$  et  $X_3$ , qui comporte cinq termes non nuls.

**Cas particulier :** Un polynôme de la forme  $a_{i_1, \dots, i_n} X_1^{i_1} \cdots X_n^{i_n}$ , qui comporte au plus un coefficient non nul, est appelé un monôme.

On a une injection naturelle

$$a \longmapsto aX_1^0 \cdots X_n^0,$$

de  $A$  dans  $A[X_1, \dots, X_n]$ , ce monôme est noté  $a$ , et appelé une constante. En particulier les polynômes 0 et 1 sont des constantes.

Pour un polynôme

$$P = \sum_{i_1, \dots, i_n \geq 0} a_{i_1, \dots, i_n} X_1^{i_1} \cdots X_n^{i_n}.$$

**Définition 2.2.2.** On appelle support de  $P$  et on note  $\text{Supp}(P)$  le sous-ensemble

$$\text{Supp}(P) = \{(i_1, \dots, i_n) : a_{i_1, \dots, i_n} \neq 0\} \subset \mathbb{N}^n,$$

Par définition, il s'agit d'un ensemble fini.

Pour l'exemple 2.2.1, on a

$$\text{Supp}(P) = \{(0, 0, 0), (1, 0, 0), (2, 0, 0), (1, 1, 0), (0, 0, 1)\}.$$

### 2.2.2 Propriétés

On a clairement

- $P + Q = Q + P$ , (commutativité).
- $P + (Q + R) = (P + Q) + R$ , (associativité).
- $P + 0 = 0 + P$ , (existence d'un élément neutre).

ce qui montre que  $(A[X_1, \dots, X_n], +, 0)$  est un groupe abélien.

La multiplication interne (le produit de Cauchy, encore parfois noté  $P \times Q$ ) possède les propriétés élémentaires suivantes :

- $PQ = QP$ , (commutativité).
- $P(QR) = (PQ)R$ , (associativité).
- $P(Q + R) = PQ + PR$ , (distributivité).
- $P.1 = 1.P = P$ , (existence d'un élément neutre).

Ceci montre que  $(A[X_1, \dots, X_n], +, \times, 0, 1)$  est un anneau commutatif unitaire. L'injection canonique  $A \longrightarrow A[X_1, \dots, X_n]$  est un morphisme d'anneaux.

Considérons enfin la multiplication externe  $(a, P) \longrightarrow a \times P$ . Elle possède les propriétés suivantes

- $0.P = 0$ .
- $1.P = P$ .
- $a.(P + Q) = a.P + a.Q$ .

- $(a + b).P = a.P + b.P.$
- $(ab).P = a.(b.P).$

Ceci montre que  $(A[X_1, \dots, X_n], +, \cdot, 0, 1)$  possède une structure de module sur l'anneau  $A$  (si  $A = K$  est un corps, il s'agit tout simplement d'un  $K$ -espace vectoriel). L'injection canonique  $A \rightarrow A[X_1, \dots, X_n]$  est clairement un morphisme de  $A$ -modules.

Muni des trois opérations précédentes,  $A[X_1, \dots, X_n]$  est une  $A$ -algèbre, et le polynôme constant  $P = 1$  est l'élément unité de cette algèbre. L'injection canonique  $A \rightarrow A[X_1, \dots, X_n]$  est aussi un morphisme de  $A$ -algèbres.

### 2.2.3 Degré d'un polynôme

Si  $P$  est un polynôme non nul en  $X_1^{i_1} \cdots X_n^{i_n}$  (c'est-à-dire, un polynôme dont au moins un des coefficients est non nul),

$$\sum_{i_1, \dots, i_n \geq 0} a_{i_1, \dots, i_n} X_1^{i_1} \cdots X_n^{i_n},$$

on définit le degré total de  $P$  par la formule

$$\deg(P) = \max\{i_1 + \dots + i_n; a_i \neq 0\},$$

et le degré partiel de  $P$  par rapport à  $X_j$  est défini par

$$\deg_{X_j}(P) = \max\{i_j : a_{i_1 \dots i_n} \neq 0\},$$

pour simplifier, on le note aussi parfois  $\deg_j(P)$ .

Une convention utile est de poser

$$\deg(0) = \deg_{X_i}(0) = -\infty,$$

avec la règle

$$\max\{n, -\infty\} = n.$$

Si  $P$  et  $Q$  sont deux polynômes non nuls, les propriétés suivantes ont lieu

- Si  $\deg(P) \neq \deg(Q)$ , alors

$$P + Q \neq 0 \text{ et } \deg(P + Q) = \max\{\deg(P), \deg(Q)\}.$$

- Si  $\deg(P) = \deg(Q)$  et  $P + Q \neq 0$ , alors

$$\deg(P + Q) \leq \max\{\deg(P), \deg(Q)\}.$$

- Si  $PQ \neq 0$ , alors

$$\deg(P + Q) \leq \deg(P) + \deg(Q).$$

Avec la convention précédente les deux relations ci-dessus ont aussi lieu pour  $P + Q = 0$  ou, respectivement, pour  $PQ = 0$ .

**Exemple 2.2.2.** Si  $P = 2 + X_1^2 + X_1X_2 + X_1X_3^2 + X_1X_2^2X_3$ , on a  $\deg(P) = 4$  du fait que  $P$  contient le terme  $X_1X_2^2X_3$ , dont le degré est maximal. On a  $\deg_3(P) = 2$  car  $P$  contient le terme  $X_1X_3^2$ , de degré maximal par rapport à la troisième variable  $X_3$ .

Un argument facile conduit à la relation

$$P + Q \neq 0 \Rightarrow \deg(P + Q) = \max\{\deg(P), \deg(Q)\}.$$

## 2.2.4 Polynômes remarquables

### 1) Polynômes homogènes

**Définition 2.2.3.** Un polynôme est dit homogène si tous les monômes qui apparaissent avec un coefficient non nul ont le même degré total.

**Exemple 2.2.3.** Le polynôme  $x^5 + 2x^3y^2 + 9xy^4 - z^3$  est homogène dans  $A[x, y]$  de degré 5 car la somme des exposants est 5 pour chacun des monômes.

### Propriétés

- Si un polynôme  $P$  de  $A[X_1, \dots, X_n]$  est homogène de degré  $d$ , alors la fonction polynomiale associée est une fonction homogène de degré  $d$ , c'est-à-dire que pour tous  $\lambda, x_1, \dots, x_n \in A$  on a :

$$P(\lambda X_1, \dots, \lambda X_n) = \lambda^d P(X_1, \dots, X_n).$$

- Les polynômes homogènes d'un degré donné forment un espace vectoriel. Les polynômes homogènes de degré 1 sont les formes linéaires :

$$\sum_{i=1}^n a_i X_i = a_1 X_1 + a_2 X_2 + \dots + a_n X_n.$$

- Les polynômes homogènes de degré 2 sont les formes quadratiques :

$$\sum_{1 \leq i \leq j \leq n} a_{ij} X_i X_j = a_{11} X_1^2 + a_{12} X_1 X_2 + \dots + a_{nn} X_n^2.$$

### 2) Polynômes symétriques

**Définition 2.2.4.** Un polynôme  $P$  de  $A[X_1, \dots, X_n]$  en  $n$  variables est symétrique si pour toute permutation  $\rho$  on a

$$P(X_1, X_2, \dots, X_n) = P(X_{\rho(1)}, X_{\rho(2)}, \dots, X_{\rho(n)}).$$

**Exemple 2.2.4.** Les polynômes  $X_1 + X_2$  et  $X_1^2 + 4X_1X_2 + X_2^2$  sont des polynômes symétriques en deux variables, et  $X_1^2X_2 + X_1^2X_3 + X_1X_2^2 + X_1X_3^2 + X_2^2X_3 + X_2X_3^2$  est un polynôme symétrique en trois variables.

## 2.3 Minimisation des polynômes homogènes de degré 2

Comme nous avons pu le voir dans la section 2.2, les polynômes homogènes de degré 2 sont les formes quadratiques.

Les formes quadratiques d'une, deux et trois variables sont données respectivement par les formules suivantes :

$$Q(x) = ax^2,$$

$$Q(x, y) = ax^2 + bxy + cy^2,$$

$$Q(x, y, z) = ax^2 + by^2 + cz^2 + dxy + exz + fyz.$$

Le chapitre 3 est justement consacré à l'étude de ces formes quadratiques et à la présentation de la méthode adaptée dans le cas de la résolution d'un problème de minimisation d'une fonctionnelle quadratique  $Q$  avec un système de contraintes bornées.

Le problème que nous allons résoudre se présente sous la forme suivante :

$$\begin{cases} \min Q(x) = \frac{1}{2}x^T D x, \\ Ax = b, \\ d_1 \leq x \leq d_2, \end{cases} \quad (2.1)$$

où la matrice  $D$  associée à la forme quadratique est supposée symétrique et semi-définie positive,  $x$ ,  $d_1$  et  $d_2$  sont des  $n$ -vecteurs,  $b$  est un  $m$ -vecteur,  $A$  est une matrice d'ordre  $m \times n$ , avec  $\text{rang}(A) = m < n$ .

## 2.4 Minimisation d'un polynôme homogène d'ordre 2 sur un simplexe standard

Cette section aborde le problème de la minimisation d'un polynôme homogène de degré 2 sur un simplexe standard. Il s'agit d'un problème classique qui a des applications dans de nombreux domaines universitaires tels que la dynamique des populations, la cinétique chimique et la théorie des graphes. Il a fait l'objet des recherches menées à l'aide de la technique d'élévation du degré et la méthode de subdivision [54]. Mais contrairement à ces deux méthodes, la méthode que nous allons présenter au chapitre 3 a la particularité de résoudre exactement ce problème et de manipuler les contraintes du problème comme elles se présentent sans aucune transformation.



## 2.4 Minimisation d'un polynôme homogène d'ordre 2 sur un simplexe standard 28

**Définition 2.4.1.** On notera  $\Delta_n$  (ou simplement  $\Delta$  quand la dimension est claire) le simplexe standard de  $\mathbb{R}^n$ , défini par :

$$\Delta_n := \{(x_1, \dots, x_n) \in \mathbb{R}^n \mid \forall i, x_i \geq 0 \text{ et } \sum_{i=1}^n x_i = 1\}.$$

Le problème de minimisation d'un polynôme homogène sur un simplexe standard de  $\mathbb{R}^n$  se présente sous la forme suivante :

$$\begin{cases} \min Q(x) = \frac{1}{2}x^T D x, \\ x \geq 0, \\ \sum_{i=1}^n x_i = 1. \end{cases} \quad (2.2)$$

Par identification, résoudre le problème (2.2) revient à résoudre le programme suivant :

$$\begin{cases} \min Q(x) = \frac{1}{2}x^T D x, \\ Ax = b, \\ x \geq 0, \end{cases} \quad (2.3)$$

avec  $D$  une matrice symétrique et semi-définie positive d'ordre  $n \times n$ ,  $x$  un vecteur colonne d'ordre  $n$ ,  $A$  un vecteur ligne d'ordre  $n$  dont toutes les composantes sont égales à 1 et  $b = 1$ .

Le problème (2.3) est un problème de minimisation d'une fonctionnelle quadratique  $Q$ , avec un système de contraintes linéaire, on peut le résoudre directement par la méthode adaptée du support pour la résolution d'un programme quadratique convexe qui sera présentée au chapitre 3.

# Chapitre 3

## La méthode adaptée pour la résolution d'un programme quadratique convexe

### 3.1 Introduction

En optimisation mathématique, un problème d'optimisation quadratique est un problème d'optimisation dans lequel on minimise (ou maximise) une fonction quadratique sur un polyèdre convexe. Les contraintes peuvent donc être décrites par des fonctions linéaires. L'optimisation quadratique est la discipline qui étudie ces problèmes.

### 3.2 Programmation quadratique convexe

Dans cette section, nous rappelons quelques définitions et propriétés des fonctions quadratiques et des matrices semi-définies positives.

#### 3.2.1 Les formes quadratiques

**Définition 3.2.1** (Les Formes quadratiques). Une fonction  $Q : \mathbb{R}^n \rightarrow \mathbb{R}$  est dite forme quadratique de  $n$  variables  $x_1, x_2, \dots, x_n$  si elle s'écrit sous la forme suivante :

$$\sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j = x^T A x,$$

où  $x^T = (x_1, x_2, \dots, x_n)$  est un  $n$ -vecteur-ligne et  $A = (a_{ij}, 1 \leq i, j \leq n)$  une matrice carrée d'ordre  $n$ . Pour  $i \neq j$ , le coefficient du terme  $x_i x_j$  s'écrit  $a_{ij} + a_{ji}$ . En

vertu de cela, la matrice  $A$  peut être supposée symétrique. En effet, en définissant de nouveaux coefficients  $d_{ij}, 1 \leq i, j \leq n$ , on obtient une nouvelle matrice  $D$  symétrique telle que :

$$D = (d_{ij}, 1 \leq i, j \leq n), \quad \text{avec} \quad d_{ij} = d_{ji} = \frac{a_{ij} + a_{ji}}{2}.$$

Il est clair qu'après une redéfinition des coefficients, la valeur de la forme quadratique  $Q(x)$  reste inchangée pour tout  $x \in \mathbb{R}^n$  :

$$Q(x) = x^T Ax = x^T Dx.$$

**Définition 3.2.2** (Gradient d'une forme quadratique). Soit  $Q : \mathbb{R}^n \rightarrow \mathbb{R}$  une fonction réelle continue et différentiable. Son gradient au point  $x$  est défini par :

$$\nabla Q(x) = \begin{pmatrix} \frac{\partial Q(x)}{\partial x_1} \\ x_1 \\ \frac{\partial Q(x)}{\partial x_2} \\ x_2 \\ \vdots \\ \frac{\partial Q(x)}{\partial x_n} \\ x_n \end{pmatrix}.$$

Si  $Q$  est une forme quadratique et  $D$  sa matrice symétrique associée, c'est à dire :  $Q(x) = x^T Dx$ , alors le gradient de  $Q$  est :

$$\nabla Q(x) = 2Dx.$$

**Définition 3.2.3** (Hessienne d'une forme quadratique). Soit  $Q : \mathbb{R}^n \rightarrow \mathbb{R}$  une fonction réelle de classe  $\mathcal{C}^2$ , la matrice Hessienne de la fonction  $Q$  est définie par :

$$\nabla^2 Q(x) = \begin{pmatrix} \frac{\partial^2 Q(x)}{\partial x_1^2} & \frac{\partial^2 Q(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 Q(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 Q(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 Q(x)}{\partial x_2^2} & \cdots & \frac{\partial^2 Q(x)}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 Q(x)}{\partial x_n \partial x_1} & \frac{\partial^2 Q(x)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 Q(x)}{\partial x_n^2} \end{pmatrix}$$

Si  $Q$  est une forme quadratique et  $D$  sa matrice symétrique associée, c'est à dire :  $Q(x) = x^T Dx$ , alors la Hessienne de  $Q$  est :

$$\nabla^2 Q = 2D.$$

**Définition 3.2.4** (Matrice semi-définie positive). Soit  $D$  une matrice carrée symétrique (ie,  $D$  matrice  $n \times n$  et  $D = D^T$ ), alors :

- $D$  est dite définie positive, si  $x^T Dx > 0, \forall x \in \mathbb{R}^n, x \neq 0$ ,
- $D$  est dite semi-définie positive, si  $x^T Dx \geq 0, \forall x \in \mathbb{R}^n$ ,
- $D$  est dite définie négative, si  $x^T Dx < 0, \forall x \in \mathbb{R}^n, x \neq 0$ ,
- $D$  est dite semi-définie négative, si  $x^T Dx \leq 0, \forall x \in \mathbb{R}^n$ .

### 3.2.2 Notion de convexité

La convexité joue un rôle majeur dans la théorie de l'optimisation. Elle est un outil indispensable et efficace pour la recherche des conditions à la fois nécessaires et suffisantes d'optimalité.

**Définition 3.2.5** (Ensemble convexe). Un ensemble  $\mathcal{C}$  de  $\mathbb{R}^n$  est dit convexe, si

$$\forall x_1, x_2 \in \mathcal{C}, (1 - \lambda)x_1 + \lambda x_2 \in \mathcal{C}, \forall \lambda \in [0, 1].$$

Cette définition signifie qu'un ensemble  $\mathcal{C}$  est convexe si le segment joignant deux de ses points quelconques est contenue dans l'ensemble  $\mathcal{C}$ .

**Définition 3.2.6** (Fonction convexe). Une fonction réelle  $Q$  définie sur un convexe  $\mathcal{C} \subset \mathbb{R}^n$ , est dite convexe, si pour tous les points  $x_1, x_2 \in \mathcal{C}$ , et pour tout nombre réel positif ou nul, tel que  $0 \leq \lambda \leq 1$ , l'inégalité suivante est vérifiée :

$$Q(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda Q(x_1) + (1 - \lambda)Q(x_2).$$

**Définition 3.2.7** (Fonction strictement convexe). Une fonction réelle  $Q$  définie sur un convexe  $\mathcal{C} \subset \mathbb{R}^n$ , est dite strictement convexe, si pour tous les points  $x_1, x_2 \in \mathcal{C}$ , avec  $x_1 \neq x_2$  et pour tout nombre réel positif ou nul, tel que  $0 < \lambda < 1$ , l'inégalité stricte suivante est vérifiée :

$$Q(\lambda x_1 + (1 - \lambda)x_2) < \lambda Q(x_1) + (1 - \lambda)Q(x_2).$$

**Théorème 3.2.1.** Une fonction quadratique  $Q(x) = x^T D x$  est convexe si et seulement si sa matrice associée  $D$  est semi-définie positive.

**Définition 3.2.8** (Fonction concave).  $Q$  est dite concave si  $-Q$  est convexe.

*Remarque 3.2.1.* L'hypothèse de convexité apporte simplicité et efficacité à la théorie de l'optimisation. En particulier les conditions nécessaires d'optimalité deviennent également suffisantes, et tout le résultat acquiert un caractère global.

**Définition 3.2.9.** On dit qu'un problème de programmation mathématique est convexe (resp. strictement convexe), s'il consiste à minimiser une fonction convexe (resp. strictement convexe) sur un domaine convexe. l'étude des problèmes convexes et des algorithmes de résolution correspondants est l'objet de la programmation convexe. L'hypothèse de convexité est cruciale en optimisation. Notons que :

- Les problèmes convexes sont en général synonymes de minimisation.
- Les problèmes convexes sont les bons problèmes de la théorie de l'optimisation, des problèmes pour lesquels il existe des algorithmes de résolution efficaces.

- l'hypothèse de convexité ne garantit cependant ni l'existence ni l'unicité d'une éventuelle solution.

**Proposition 3.2.1.** *Soit  $Q$  une fonction convexe définie sur un convexe  $C \in \mathbb{R}^n$ , alors l'ensemble des points où  $Q$  atteint son minimum est convexe.*

**Proposition 3.2.2.** *Tout minimum local est un minimum global.*

**Proposition 3.2.3.** *Si la fonction  $Q$  est strictement convexe, alors son minimum global lorsqu'il existe est atteint en un seul point  $x^0$ .*

### 3.2.3 Problème quadratique convexe (PQC)

Il s'agit d'une classe de problèmes d'optimisation, où la fonction objectif est quadratique, s'écrivant sous la forme :

$$Q(x) = \frac{1}{2}x^T D x + c^T x,$$

qu'on minimise sur un polyèdre convexe fermé. Ce genre de problème est convexe dès lors que la matrice  $D$  est semi-définie positive.

*Remarque 3.2.2.* On remarquera qu'un problème linéaire est un problème quadratique dégénéré ( $D = 0$ ), et c'est toujours un problème convexe.

## 3.3 Position du problème et définitions

Dans cette section, on s'intéresse au problème de minimisation d'un polynôme homogène de degré 2, c'est à dire au problème de minimisation d'une fonctionnelle quadratique  $Q$ , avec un système de contraintes bornées, et ce en utilisant la méthode adaptée du support.

Le problème que nous allons résoudre se présente sous la forme suivante :

$$\begin{cases} \min Q(x) = \frac{1}{2}x^T D x, \\ Ax = b, \\ d_1 \leq x \leq d_2. \end{cases} \quad (3.1)$$

Comme on peut le voir ci-dessus, la fonction objective  $Q(x)$  est un polynôme homogène de degré 2.

Soient les ensembles d'indices suivants :

$$I = \{1, 2, \dots, m\}, \quad J = \{1, 2, \dots, n\}.$$

On pose  $J = J_B \cup J_N$  avec  $J_B \cap J_N = \emptyset$  et  $|J_B| = m$ .

On peut alors fractionner les vecteurs et les matrices de la manière suivante :

$$x = x(J) = (x_j, j \in J),$$

$$x = \begin{pmatrix} x_B \\ \text{---} \\ x_N \end{pmatrix}, \quad x_B = x(J_B) = (x_j, j \in J_B), \quad x_N = x(J_N) = (x_j, j \in J_N),$$

$$A = (a_j, j \in J) = (A_B | A_N), \quad A_B = A(I, J_B), \quad A_N = A(I, J_N).$$

**Définition 3.3.1.** • Un vecteur  $x$  vérifiant les contraintes du problème (3.1) est appelé *plan* ou *solution réalisable* du problème (3.1).

- Un plan  $x^0$  est dit *optimal* si

$$Q(x^0) = (x^0)^T D x^0 = \min(x^T D x),$$

où  $x$  est pris parmi tous les plans du problème (3.1).

- Un plan  $x^\epsilon$  est appelé  $\epsilon$ -*optimal* ou *suboptimal* si

$$Q(x^\epsilon) - Q(x^0) \leq \epsilon,$$

où  $\epsilon$  est un nombre positif ou nul, donné à l'avance et  $x^0$  est une solution optimale du problème (3.1).

- L'ensemble  $J_B \setminus |J_B| = m$ , est appelé *support des contraintes*.
- Le couple  $\{x, J_B\}$ , formé d'une solution réalisable  $x$  et d'un support  $J_B$ , est appelé *plan de support*.
- Le plan de support est dit *non dégénéré* si

$$d_{1j} < x_j < d_{2j}, \quad \text{pour tout } j \in J_B.$$

## 3.4 Formule d'accroissement de la fonction objectif

Soit  $\{x, J_B\}$  un plan de support du problème (3.1) et considérons un autre plan quelconque  $\bar{x} = x + \Delta x$ .

L'accroissement de la fonction objectif s'écrit alors :

$$Q(\bar{x}) - Q(x) = g^T(x) \Delta x + \frac{1}{2} (\Delta x)^T D (\Delta x),$$

où  $g(x) = Dx$  est le gradient de la fonction objectif  $Q$  au point  $x$ , avec  $g(x) = g(J) = (g_j, j \in J)$ . Il peut être fractionné de la manière suivante :

$$g = \begin{pmatrix} g_B \\ \text{---} \\ g_N \end{pmatrix}, \quad g_B = g(J_B) = (g_j, j \in J_B), \quad g_N = g(J_N) = (g_j, j \in J_N).$$

Par ailleurs, on a

$$\begin{cases} Ax = b, \\ A\bar{x} = b. \end{cases} \Leftrightarrow A\Delta x = 0.$$

En posant

$$\Delta x = \begin{pmatrix} \Delta x_B \\ \Delta x_N \end{pmatrix}, \quad \Delta x_B = \Delta x(J_B), \quad \Delta x_N = \Delta x(J_N),$$

l'égalité  $A\Delta x = 0$  peut s'écrire  $A_B\Delta x_B + A_N\Delta x_N = 0$ ,  
d'où

$$\Delta x_B = -A_B^{-1}A_N\Delta x_N. \quad (3.2)$$

La formule de l'accroissement de la fonction objectif devient alors :

$$\begin{aligned} Q(\bar{x}) - Q(x) &= \frac{1}{2} \begin{pmatrix} \Delta x_B \\ \Delta x_N \end{pmatrix}^T D \begin{pmatrix} \Delta x_B \\ \Delta x_N \end{pmatrix} + g^T(x)\Delta x \\ &= g^T(x)\Delta x + \frac{1}{2} \begin{pmatrix} -A_B^{-1}A_N\Delta x_N \\ \Delta x_N \end{pmatrix}^T D \begin{pmatrix} -A_B^{-1}A_N\Delta x_N \\ \Delta x_N \end{pmatrix} \\ &= g^T(x)\Delta x + \frac{1}{2}(\Delta x_N)^T \begin{pmatrix} -A_B^{-1}A_N \\ I_N \end{pmatrix}^T D \begin{pmatrix} -A_B^{-1}A_N \\ I_N \end{pmatrix} (\Delta x_N), \end{aligned}$$

où  $I_N = I(J_N, J_N)$  est une matrice identité d'ordre  $(n - m)$ .

Posons

$$Z = \begin{pmatrix} -A_B^{-1}A_N \\ I_N \end{pmatrix} \quad \text{et} \quad M = Z^T D Z.$$

Nous définissons le vecteur des potentiels  $u$  ainsi que le vecteur des estimations  $E$  comme suit :

$$\begin{aligned} u^T &= g_B^T A_B^{-1}, \\ E^T &= g^T - u^T A = (E_B^T, E_N^T), \end{aligned}$$

où

$$E_B^T = 0, \quad E_N^T = g_N^T - u^T A_N.$$

Finalement, l'accroissement de la fonction objectif aura la forme suivante :

$$Q(\bar{x}) - Q(x) = E_N^T \Delta x_N + \frac{1}{2} (\Delta x_N)^T M (\Delta x_N). \quad (3.3)$$

### 3.5 Critère d'optimalité

**Théorème 3.5.1.** *Soit  $\{x, J_B\}$  un plan de support des contraintes du problème (3.1). Alors les relations suivantes :*

$$\begin{cases} E_j \geq 0, & \text{si } x_j = d_{1j}, & j \in J_N, \\ E_j \leq 0, & \text{si } x_j = d_{2j}, & j \in J_N, \\ E_j = 0, & \text{si } d_{1j} < x_j < d_{2j}, & j \in J_N, \end{cases} \quad (3.4)$$

sont suffisantes pour l'optimalité du plan  $x$ .

Ces mêmes relations sont aussi nécessaires, si le plan de support des contraintes est non-dégénéré.

*Démonstration.* Voir dans [56] □

### 3.6 Critère de suboptimalité

Le nombre  $\beta(x, J_B)$  dit estimation de suboptimalité est une valeur qui estime l'écart entre la valeur optimale  $Q(x^0)$  et une autre valeur  $Q(x)$  d'un plan de support des contraintes quelconque  $\{x, J_B\}$ .

**Théorème 3.6.1.** *Soit  $\{x, J_B\}$  un plan de support des contraintes du problème (3.1) et  $\epsilon$  un nombre réel positif ou nul quelconque. Si le nombre*

$$\beta(x, J_B) = \sum_{j \in J_N, E_j > 0} E_j(x_j - d_{1j}) + \sum_{j \in J_N, E_j < 0} E_j(x_j - d_{2j}) \leq \epsilon,$$

alors  $\{x, J_B\}$  est  $\epsilon$ -optimal.

*Démonstration.* Voir dans [56] □

### 3.7 Méthode de résolution

Avant de présenter la méthode de résolution, donnons quelques définitions essentielles.

**Définition 3.7.1.** • On appelle support de la fonction objectif du problème (3.1), l'ensemble des indices  $J_S \subset J_N$ , tel que

$$\det M(J_S, J_S) \neq 0.$$

On posera  $J_{NN} = J_N \setminus J_S$ .

- L'ensemble des indices  $J_P = \{J_B, J_S\}$  est appelé support du problème (3.1).
- On appelle plan de support du problème (3.1), la paire  $\{x, J_P\}$  formée du plan  $x$  et du support  $J_P$ , il est dit accordé si  $E(J_S) = 0$ .

Étant donné un nombre réel positif ou nul quelconque  $\epsilon$  et un plan de support initial accordé  $\{x, J_P\}$ , le but de l'algorithme est alors de construire un plan  $\epsilon$ -optimal  $x^\epsilon$  ou carrément un plan optimal  $x^0$ .

L'itération de l'algorithme consiste à faire le passage de  $\{x, J_P\}$  vers  $\{\bar{x}, \bar{J}_P\}$  tel que  $Q(\bar{x}) \leq Q(x)$ . La première étape consiste à changer le plan  $x$  en un nouveau plan  $\bar{x}$  de la manière suivante :  $\bar{x} = x + \theta^0 l$ , où  $l$  est un  $n$ -vecteur, appelé direction d'amélioration adaptée et  $\theta^0$  est le pas le long de cette direction. La seconde étape sera celle du changement du support  $J_P$  en  $\bar{J}_P$ .



### 3.7.1 Construction de la direction d'amélioration adaptée $l$

Considérons la métrique suivante pour les composantes non basiques de la direction admissible  $l$  :

$$d_{1j} - x_j \leq l_j \leq d_{2j} - x_j, j \in J_{NN} = J_N \setminus J_S. \quad (3.5)$$

Cette métrique dépend du plan courant  $x$ , et de ce fait, elle est dite adaptée. Afin de calculer les composantes de la direction admissible d'amélioration  $l$ , considérons l'accroissement :

$$\Delta Q = Q(x + l) - Q(x) = E_N^T l_N + \frac{1}{2} l_N^T M l.$$

En tenant compte de la métrique (3.5), la partie linéaire de  $\Delta Q$  atteint son minimum pour les valeurs des composantes de  $l(J_{NN})$  suivantes :

$$l_j = \begin{cases} d_{1j} - x_j, & \text{si } E_j > 0, \\ d_{2j} - x_j, & \text{si } E_j < 0, \\ 0, & \text{si } E_j = 0, \end{cases} j \in J_{NN}. \quad (3.6)$$

Quant aux composantes  $l_S$ , on les calculera de manière à assurer

$$\bar{E}_j = 0, \quad j \in J_S.$$

Nous avons :

$$E_N^T = g_N^T - g_B^T A_B^{-1} A_N = (g_B^T, g_N^T) \begin{pmatrix} -A_B^{-1} A_N \\ I_N \end{pmatrix} = g^T(x) Z.$$

Comme  $l = Z l_N$ , on aura donc

$$\begin{aligned} \bar{E}_N^T &= g^T(z + \theta^0 l) Z = g^T(z) Z + \theta^0 l^T D Z \\ &= g^T(z) Z + \theta^0 l_N^T Z^T D Z \\ &= E_N^T + \theta^0 l_N^T M. \end{aligned}$$

On pose  $\delta_N = M l_N$ .

Finalement, on a

$$\bar{E}_N = E_N + \theta^0 \delta_N.$$

Puisque  $E(J_S) = 0$ , alors l'équation  $\bar{E}(J_S) = 0$  est équivalente à

$$M(J_S, J_S) l(J_S) + M(J_S, J_{NN}) l(J_{NN}) = 0.$$

D'où

$$l(J_S) = -M_S^{-1}M(J_S, J_{NN})l(J_{NN}). \quad (3.7)$$

Ensuite, on calcule  $l_B$  de manière à avoir  $Al = 0$  :

$$l_B = -A_B^{-1}(A_S l_S + A_{NN} l_{NN}). \quad (3.8)$$

### 3.7.2 Calcul du pas maximal $\theta^0$

On construit le nouveau plan  $\bar{x}$  sous la forme :

$$\bar{x} = x + \theta^0 l,$$

où  $l$  est la direction d'amélioration définie par les relations (3.6)-(3.8) et le nombre  $\theta^0$  est le pas le long de cette direction tel que :

$$\theta^0 = \min\{1, \theta_{j_0}, \theta_S, \theta_F\}.$$

Les nombres  $\theta_{j_0}$  et  $\theta_S$  se calculent de façon à ce que les relations suivantes soient vérifiées :

$$\begin{aligned} d_{1j} - x_j &\leq \theta^0 l_j \leq d_{2j} - x_j, j \in J_{NN}, \\ d_{1j} - x_j &\leq \theta^0 l_j \leq d_{2j} - x_j, j \in J_S, \\ d_{1j} - x_j &\leq \theta^0 l_j \leq d_{2j} - x_j, j \in J_B. \end{aligned}$$

Donc

$$\theta_{j_0} = \min(\theta_j, j \in J_B), \quad \theta_S = \min(\theta_j, j \in J_S),$$

avec

$$\theta_j = \begin{cases} \frac{d_{2j} - x_j}{l_j}, & \text{si } l_j > 0, \\ \frac{d_{1j} - x_j}{l_j}, & \text{si } l_j < 0, \\ \infty, & \text{si } l_j = 0. \end{cases} \quad (3.9)$$

Le nombre  $\theta^0 = 1$  représente le pas correspondant aux indices de  $J_{NN}$ .

Quant à  $\theta_F$ , il se calcule de façon que le passage de  $x$  à  $\bar{x}$  puisse assurer une relaxation maximale de la fonction objectif, tout en gardant le même signe pour  $E_j$  et  $\bar{E}_j$ , où

$$\begin{aligned} E_N^T(x) &= g^T(x)Z, \quad \bar{E}_N(\bar{x}) = \bar{E}_N(x + \theta^0 l), \\ \bar{E}_N(\bar{x}) &= E_N(x) + \theta^0 M l_N = E_N(x) + \theta^0 \delta_N. \end{aligned}$$

On posera donc  $\theta_F = \sigma_{j_*} = \min \sigma_j$ ,  $j \in J_{NN}$ , avec

$$\sigma_j = \begin{cases} -\frac{E_j}{\delta_j}, & \text{si } E_j \delta_j < 0, \\ 0, & \text{si } E_j = 0 \text{ et } \delta_j < 0, \\ \infty, & \text{sinon .} \end{cases}$$

$$\delta_j = M(j, J_N)l_N.$$

Le nouveau plan est  $\bar{x} = x + \theta^0 l$ .

### 3.7.3 Estimation de suboptimalité

Calculons la nouvelle estimation de suboptimalité  $\beta(\bar{x}, J_B)$  en fonction de  $\beta(x, J_B)$  :

$$\beta(\bar{x}, J_B) = \sum_{\substack{j \in J_N, \\ \bar{E}_j > 0}} \bar{E}_j(\bar{x}_j - d_{1j}) + \sum_{\substack{j \in J_N, \\ \bar{E}_j < 0}} \bar{E}_j(\bar{x}_j - d_{2j}).$$

On a alors

$$\beta(\bar{x}, J_B) = \beta(x, J_B) - \theta^0 \beta(x, J_B) - \theta^0 \delta_N^T l_N + (\theta^0)^2 \delta_N^T l_N.$$

Finalement, on aura la formule suivante :

$$\beta(\bar{x}, J_B) = (1 - \theta^0) \beta(x, J_B) - \theta^0 (1 - \theta^0) l^T D l.$$

### 3.7.4 Changement de support

Le changement de support s'effectue lorsque  $\theta^0 < 1$  et  $\beta(x, J_B) > \epsilon$ . Dans ce qui suit, on énumère les différents changements de support suivant la valeur que prend  $\theta^0$ .

- Pour  $\theta^0 = 1$ , le vecteur  $x^0 = x + \theta^0 l$  est une solution optimale du problème (3.1).
- Si  $\theta^0 = \theta_{j_0} < 1$ , alors contrairement à la méthode du simplexe, le choix de l'indice  $j_1$  n'est pas unique, ce qui fait la particularité de cette méthode. Lorsque ce cas se réalise pour un indice  $j_0 \in J_B$ , on a alors

$$l_{j_0} = - \sum_{j \in J_N} e_{j_0}^T A_B^{-1} a_j l_j = - \sum_{j \in J_N} x_{j_0 j} l_j \neq 0,$$

où  $e$  est un vecteur unitaire de dimension  $m$ . Il existe alors  $j_1 \in J_N$  tel que  $x_{j_0 j_1} \neq 0$ . Cette dernière condition nous assure par conséquent que  $\bar{J}_B =$

$(J_B \setminus j_0) \cup j_1$  est bel et bien un support.

Si on peut avoir  $x_{j_0 j_1} \neq 0$ , avec  $j_1 \in J_S$ , on posera donc

$$\bar{J}_B = (J_B \setminus j_0) \cup j_1, \quad \bar{J}_S = (J_S \setminus j_1).$$

Sinon, on choisira un indice  $j_1 \in J_{NN}$  tel que  $l_{j_1} \neq 0$  et on posera

$$\bar{J}_B = (J_B \setminus j_0) \cup j_1, \quad \bar{J}_S = J_S.$$

- Si  $\theta^0 = \theta_{j_s}$  : alors  $\bar{J}_B = J_B$ ,  $\bar{J}_S = J_S \setminus j_s$ ,
- Si  $\theta^0 = \theta_F$  : alors  $\bar{J}_B = J_B$ ,  $\bar{J}_S = J_S \cup j_*$ , tel que  $j_*$  est l'indice correspondant à  $\theta_F$ .

On recommencera alors une nouvelle itération avec le nouveau plan de support accordé  $\{\bar{x}, \bar{J}_B\}$ ,  $\bar{J}_P = \{\bar{J}_B, \bar{J}_S\}$  si  $\beta(\bar{x}, \bar{J}_B) > \epsilon$ .

### 3.8 Algorithme de résolution

Soit un nombre réel positif ou nul quelconque  $\epsilon$  et un plan de support initial  $\{x, J_P\}$ , où  $J_P = \{J_B, J_S\}$ .

Pour faciliter les calculs, on commence par  $J_S = \emptyset$  et ( $S = 0$ ).

Les étapes de l'algorithme sont les suivantes :

#### Début

1. Calculer le vecteur des estimations :  $E^T = (E_B^T, E_N^T)$ ,
  - Calculer  $\beta(x, J_B)$ .
    - Si  $\beta(x) = 0$ , alors le processus de résolution s'arrête avec  $\{x, J_B\}$  plan de support optimal.
    - Si  $\beta(x) \leq \epsilon$ , alors le processus de résolution s'arrête avec  $\{x, J_B\}$  un plan de support  $\epsilon$ -optimal.
    - Sinon, aller en 2.
2. Changement du plan  $x$  en  $\bar{x}$  tel que  $\bar{x} = x + \theta^0 l$ ,
  - Calculer la direction d'amélioration  $l$ .
  - Calculer le pas  $\theta^0$ .
  - Calculer  $\bar{x} = x + \theta^0 l$ .
  - Calculer  $\beta(\bar{x}) = (1 - \theta^0)(\beta(x) - \theta^0 l^T D l)$ .
    - Si  $\beta(\bar{x}) = 0$ , alors le processus de résolution s'arrête avec  $\{\bar{x}, J_B\}$  plan de support optimal.
    - Si  $\beta(\bar{x}) \leq \epsilon$ , alors le processus de résolution s'arrête avec  $\{\bar{x}, J_B\}$  un plan de support  $\epsilon$ -optimal.
    - Sinon, aller en 3.
3. Changement de support  $J_P$  en  $\bar{J}_P$  :

- Si  $\theta^0 = 1$  alors  $\bar{x} = x + \theta^0 l$ ;  $\beta(\bar{x}) = 0$ ;  $\bar{x}$  est optimal; aller a fin.
- Si  $\theta^0 = \theta_{j_0}$ , alors
  - S'il existe  $j_1 \in J_S$  tel que  $x_{j_0 j_1} \neq 0$ , alors on pose :  
 $\bar{J}_B = (J_B \setminus j_0) \cup j_1$ ,  $\bar{J}_S = J_S \setminus j_1$ ,  $\bar{J}_P = \{\bar{J}_B, \bar{J}_S\}$ .
  - Sinon on choisit  $j_1 \in J_{NN}$  tel que  $l_{j_1} \neq 0$ , alors on pose :  
 $\bar{J}_B = (J_B \setminus j_0) \cup j_1$ ,  $\bar{J}_S = J_S$ ,  $\bar{J}_P = \{\bar{J}_B, \bar{J}_S\}$ .
- Si  $\theta^0 = \theta_{j_s}$ , alors

$$\begin{aligned}\bar{J}_B &= J_B, \\ \bar{J}_S &= J_S \setminus j_s, \\ \bar{J}_P &= \{\bar{J}_B, \bar{J}_S\}.\end{aligned}$$

- Si  $\theta^0 = \theta_F$ , alors

$$\begin{aligned}\bar{J}_B &= J_B, \\ \bar{J}_S &= J_S \cup j_*, \\ \bar{J}_P &= \{\bar{J}_B, \bar{J}_S\}.\end{aligned}$$

( $S = 2$ ) , aller en 1 avec le plan de support  $\{\bar{x}, \bar{J}_P\}$ .

Fin.

### 3.9 Exemple numérique

Soit le problème de programmation quadratique convexe à variables bornées suivant :

$$\begin{cases} \min Q(x) = x_1^2 + x_1 x_2 + 6x_2^2 + x_3^2 + x_3 x_4 + 6x_4^2, \\ x_1 + 2x_2 + x_3 = 4, \\ 2x_1 + x_2 + x_4 = 5, \\ -2 \leq x_1 \leq 2, \\ -4 \leq x_2 \leq 4, \\ -6 \leq x_3 \leq 6, \\ -8 \leq x_4 \leq 8. \end{cases} \quad (3.10)$$

Par identification, on a :

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}, D = \begin{pmatrix} 2 & 1 & 0 & 0 \\ 1 & 12 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 1 & 12 \end{pmatrix}, A = \begin{pmatrix} 1 & 2 & 1 & 0 \\ 2 & 1 & 0 & 1 \end{pmatrix}, b = \begin{pmatrix} 4 \\ 5 \end{pmatrix},$$

$$d_1 = \begin{pmatrix} -2 \\ -4 \\ -6 \\ -8 \end{pmatrix}, d_2 = \begin{pmatrix} 2 \\ 4 \\ 6 \\ 8 \end{pmatrix}.$$

On a  $x^0 = (0, 0, 4, 5)$  est un plan initial réalisable pour le programme (3.10), avec

$Q(x^0) = 186$ .

Posons  $J_B = \{3, 4\}$ ,  $J_N = \{1, 2\}$ ,  $J_S = \emptyset$  et  $J_P = \{J_B, J_S\}$ .

**1<sup>ère</sup> itération :**

Soit  $\{x^0, J_P\}$  un plan de support réalisable et  $\epsilon = 0.01$ .

- Déterminons la matrice  $M$ , on a :

$$A_B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, A_N = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}, Z = \begin{pmatrix} -A_B^{-1}A_N \\ I_N \end{pmatrix} = \begin{pmatrix} -1 & -2 \\ -2 & -1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix},$$

d'où

$$M = Z^T D Z = \begin{pmatrix} -1 & -2 & 1 & 0 \\ -2 & -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 0 & 0 \\ 1 & 12 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 1 & 12 \end{pmatrix} \begin{pmatrix} -1 & -2 \\ -2 & -1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 55 & 45 \\ 32 & 19 \end{pmatrix}.$$

- Calculons le vecteur gradient  $g(x^0)$  :

$$g(x^0) = Dx = \begin{pmatrix} 0 \\ 0 \\ 13 \\ 64 \end{pmatrix}, \text{ avec } g_B = \begin{pmatrix} 13 \\ 64 \end{pmatrix} \text{ et } g_N = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

- Calculons le vecteur des estimations  $E$ , on a :

$$E_N = g_N - (g_B^T A_B^{-1} A_N)^T = \begin{pmatrix} -141 \\ -90 \end{pmatrix},$$

jusqu'à la fin de la résolution, on a toujours  $E_B = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ .

- Calculons l'estimation de suboptimalité  $\beta(x^0, J_B)$  :

$$\begin{aligned} \beta(x^0, J_B) &= \sum_{\substack{j \in J_N \\ E_j > 0}} E_j(x_j - d_{1j}) + \sum_{\substack{j \in J_N \\ E_j < 0}} E_j(x_j - d_{2j}) \\ &= E_1(x_1 - d_{21}) + E_2(x_2 - d_{22}) \\ &= -141(0 - 2) - 90(0 - 4) \\ &= 642 > \epsilon. \end{aligned}$$

**Changement de plan**

- Calculons la direction d'amélioration  $l$ , on a :

$$l_{NN} = \begin{pmatrix} l_1 \\ l_2 \end{pmatrix} = \begin{pmatrix} d_{21} - x_1 \\ d_{22} - x_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \end{pmatrix},$$

$$l(J_S) = 0, \quad (\text{car } J_S = \emptyset),$$

$$l_B = -A_B^{-1} A_{NN} l_{NN} = \begin{pmatrix} -10 \\ -8 \end{pmatrix}.$$

Ainsi, on a :

$$l = \begin{pmatrix} 2 \\ 4 \\ -10 \\ -8 \end{pmatrix}.$$

- Calculons le pas  $\theta^0$  le long de cette direction :

$$\theta^0 = \min\{1, \theta_{j_0}, \theta_S, \theta_F\},$$

$$\theta_{j_0} = \min\{\theta_j, j \in J_B\} = 1.625,$$

$$\theta_F = \sigma_{j_*} = 0.49 \quad \text{avec } j_* = 1,$$

$$\text{Donc } \theta^0 = 0.49 \quad \text{avec } j_0 = 4.$$

- On a alors le nouveau plan :

$$x^{(1)} = x^0 + \theta^0 l = \begin{pmatrix} 0.97 & 1.94 & -0.86 & 1.11 \end{pmatrix},$$

$$\text{avec } Q(x^{(1)}) = 32, 71.$$

- Calculons le nouveau vecteur gradient pour le nouveau plan de support  $\{x^{(1)}, J_P\}$  :

$$g(x^{(1)}) = Dx^{(1)} = \begin{pmatrix} 3.89 \\ 24.31 \\ -0.61 \\ 12.46 \end{pmatrix}, \quad \text{avec } g_B = \begin{pmatrix} -0.61 \\ 12.46 \end{pmatrix} \quad \text{et } g_N = \begin{pmatrix} 3.89 \\ 24.31 \end{pmatrix}.$$

$$\text{On a : } E_N = g_N - (g_B^T A_B^{-1} A_N)^T = \begin{pmatrix} -20.42 \\ 13.08 \end{pmatrix}.$$

- Calculons l'estimation de suboptimalité  $\beta(\bar{x}) = \beta(x^{(1)}, J_B)$  :

$$\beta(x^{(1)}, J_B) = 98, 71 > \epsilon.$$

- Effectuons le changement de support :

On a :

$$j_0 = 4 \quad \text{et } j_1 = 1 \in J_{NN} \quad \text{car } l_1 = 2 \neq 0,$$

d'où

$$J_B^{(1)} = (J_B \setminus j_0) \cup j_1 = \{1, 3\}, \quad J_S^{(1)} = J_S = \emptyset.$$

- Calculons le nouveau vecteur gradient pour le plan de support  $\{x^{(1)}, J_P^{(1)}\}$  :

$$g(x^{(1)}) = Dx^{(1)} = \begin{pmatrix} 3.89 \\ 24.31 \\ -0.61 \\ 12.46 \end{pmatrix},$$

avec

$$g_B = \begin{pmatrix} 3.89 \\ -0.61 \end{pmatrix}, \quad g_N = \begin{pmatrix} 24.31 \\ 12.46 \end{pmatrix}, \quad A_B = \begin{pmatrix} 1 & 1 \\ 2 & 0 \end{pmatrix}, \quad A_B^{-1} = \begin{pmatrix} 0 & 0.5 \\ 1 & -0.5 \end{pmatrix},$$

$$A_N = \begin{pmatrix} 2 & 0 \\ 1 & 1 \end{pmatrix},$$

donc

$$E_N = g_N - g_B^T A_B^{-1} A_N^T = \begin{pmatrix} \frac{6753}{290} \\ \frac{290}{2961} \\ \frac{290}{290} \end{pmatrix}.$$

On recommence alors une nouvelle itération avec le plan de support  $\{x^{(1)}, J_P^{(1)}\}$ .

### 2<sup>ème</sup> itération

Soit le plan de support réalisable  $\{x^{(1)}, J_P^{(1)}\}$ .

- Calculons l'estimation de suboptimalité  $\beta(x^{(1)}, J_B^{(1)})$ , on a :

$$\beta(x^{(1)}, J_B^{(1)}) = 231,45 > \epsilon.$$

- Calculons la matrice  $M$  :

$$M = Z^T D Z = \begin{pmatrix} 31 & -7 \\ -7 & 15 \end{pmatrix}.$$

### Changement de plan :

- Calculons la direction d'amélioration  $l$  :

on a :

$$l_{NN} = \begin{pmatrix} l_2 \\ l_4 \end{pmatrix} = \begin{pmatrix} -5.94 \\ -9.11 \end{pmatrix},$$

$$l_S = 0 \quad \text{car} \quad J_S^{(1)} = \emptyset,$$

$$l_B = -A_B^{-1} A_{NN} l_{NN} = \begin{pmatrix} 7.53 \\ 4.36 \end{pmatrix}.$$

donc :  $l = \begin{pmatrix} 7.53 \\ -5.94 \\ 4.36 \\ -9.11 \end{pmatrix}.$

- Calculons le pas  $\theta^0$  le long de cette direction :

$$\theta^0 = \min\{1, \theta_{j_0}, \theta_S, \theta_F\},$$

$$\theta_{j_0} = \min\{\theta_j, j \in J_B\} = 0.14 \quad \text{avec} \quad j_0 = 2,$$

$$\theta_F = \sigma_{j_*} = 0.11 \quad \text{avec} \quad j_* = 4,$$

$$\text{donc} \quad \theta^0 = 0.11.$$



- Le nouveau plan est :

$$x^{(2)} = x^{(1)} + \theta^0 l = \begin{pmatrix} 1,78 & 1,31 & -0,4 & 0,14 \end{pmatrix},$$

avec  $Q(x^{(2)}) = 15,81$ .

Calculons à présent le vecteur gradient associé au plan de support  $\{x^{(2)}, J_P^{(1)}\}$  :

$$g(x^{(2)}) = Dx^{(2)} = \begin{pmatrix} 4.87 \\ 17.5 \\ -0.66 \\ 1.28 \end{pmatrix},$$

$$g_B = \begin{pmatrix} 17.5 \\ -0.66 \end{pmatrix} \quad \text{et} \quad g_N = \begin{pmatrix} 4.87 \\ 1.28 \end{pmatrix},$$

$$\text{donc } E_N = g_N - (g_B^T A_B^{-1} A_N)^T = \begin{pmatrix} -32.11 \\ -17.54 \end{pmatrix}.$$

- On a :  $\beta(x^{(2)}, J_B^{(1)}) = 224.24 > \epsilon$ .

**changement de Support :**

$\theta_{j_0} = \min\{\theta_j, j \in J_B\} = 0.16$  avec  $j_0 = 1$ ,

$j_1 = 2 \in J_{NN}$  avec  $l_2^{(1)} \neq 0$ ,

$J_B^{(2)} = (J_B^{(1)} \setminus j_0) \cup j_1 = \{2, 3\}$  et  $J_S^{(2)} = J_S = \emptyset$ .

- Calculons le nouveau vecteur gradient pour le plan de support  $\{x^{(2)}, J_P^{(2)}\}$  :

$$g(x^{(2)}) = Dx^{(2)} = \begin{pmatrix} 4.87 \\ 17.5 \\ -0.66 \\ 1.28 \end{pmatrix},$$

avec

$$g_B = \begin{pmatrix} 17.5 \\ -0.66 \end{pmatrix}, \quad g_N = \begin{pmatrix} 4.87 \\ 1.28 \end{pmatrix}, \quad A_B = \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}, \quad A_B^{-1} = \begin{pmatrix} 0 & 1 \\ 1 & -2 \end{pmatrix}$$

$$\text{et } A_N = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix},$$

$$\text{donc } E_N = g_N - (g_B^T A_B^{-1} A_N)^T = \begin{pmatrix} -32.11 \\ -17.54 \end{pmatrix}.$$

### 3<sup>ème</sup> itération

Soit le plan de support réalisable  $\{x^{(2)}, J_P^{(2)}\}$ .

- Calculons l'estimation de suboptimalité  $\beta(x^{(2)}, J_B^{(2)})$ , on a :

$$\beta(x^{(2)}, J_B^{(2)}) = 144,92 > \epsilon.$$

- Déterminons la matrice  $M$ , on a :

$$M = Z^T D Z = \begin{pmatrix} 106 & 70 \\ 70 & 58 \end{pmatrix}.$$

### Changement de plan :

- Calculons la direction d'amélioration  $l$  :

on a :

$$l_{NN} = \begin{pmatrix} l_1 \\ l_4 \end{pmatrix} = \begin{pmatrix} 0.22 \\ 7.86 \end{pmatrix},$$

$$l(J_S) = 0,$$

$$l_B = -A_B^{-1} A_{NN} l_{NN} = \begin{pmatrix} -8.3 \\ 16.38 \end{pmatrix},$$

$$\text{d'où } l = \begin{pmatrix} 0.22 \\ -8.3 \\ 16.38 \\ 7.86 \end{pmatrix}.$$

- Calculons le pas  $\theta^0$  le long de cette direction :

$$\theta^0 = \min\{1, \theta_{j_1}, \theta_S, \theta_F\},$$

$$\theta_{j_1} = \min\{\theta_j, j \in J_B\} = 0.39 \text{ avec } j_0 = 3,$$

$$\theta_F = \sigma_{j_*} = 0.03 \text{ avec } j_* = 4,$$

$$\text{donc } \theta^0 = 0.03.$$

- Le nouveau plan est :

$$x^{(3)} = x^{(2)} + \theta^0 l = \begin{pmatrix} 1.78 & 1.06 & 0.09 & 0.38 \end{pmatrix},$$

avec  $Q(x^{(3)}) = 12.73$ .

Calculons à présent le vecteur gradient associé au plan de support  $\{x^{(3)}, J_P^{(2)}\}$  :

$$g(x^{(3)}) = D x^{(3)} = \begin{pmatrix} 4.62 \\ 14.5 \\ 0.56 \\ 4.65 \end{pmatrix},$$

$$\text{avec } g_B = \begin{pmatrix} 14.5 \\ 0.56 \end{pmatrix} \text{ et } g_N = \begin{pmatrix} 4.62 \\ 4.65 \end{pmatrix},$$

$$\text{donc } E_N = g_N - (g_B^T A_B^{-1} A_N)^T = \begin{pmatrix} -22.7 \\ -8.73 \end{pmatrix}$$

- On a :  $\beta(x^{(3)}, J_B^{(2)}) = 71.52 > \epsilon$ .

**Changement de support :**

$$\theta_{j_0} = \min\{\theta_j, j \in J_B\} = 0.39 \text{ avec } j_0 = 3,$$

$$j_1 = 1 \in J_{NN} \text{ avec } l_2^{(2)} \neq 0,$$

$$J_B^{(3)} = (J_B^{(2)} \setminus j_0) \cup j_1 = \{1, 2\} \text{ et } J_S^{(2)} = J_S = \emptyset.$$

- Calculons le nouveau vecteur gradient pour le plan de support  $\{x^{(3)}, J_P^{(3)}\}$  :

$$g(x^{(3)}) = Dx^{(3)} = \begin{pmatrix} 4.62 \\ 14.5 \\ 0.56 \\ 4.65 \end{pmatrix},$$

avec

$$g_B = \begin{pmatrix} 4.62 \\ 14.5 \end{pmatrix}, \quad g_N = \begin{pmatrix} 0.56 \\ 4.65 \end{pmatrix}, \quad A_B = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}, \quad A_B^{-1} = \begin{pmatrix} -0.33 & 0.66 \\ 0.66 & -0.33 \end{pmatrix}$$

$$\text{et } A_N = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

$$\text{donc } E_N = g_N - (g_B^T A_B^{-1} A_N)^T = \begin{pmatrix} -7.56 \\ 6.4 \end{pmatrix}.$$

**4<sup>ème</sup> itération**

Soit le plan de support réalisable  $\{x^{(3)}, J_P^{(3)}\}$ .

- Calculons l'estimation de suboptimalité  $\beta(x^{(3)}, J_B^{(3)})$ , on a :

$$\beta(x^{(3)}, J_B^{(3)}) = 98,31 > \epsilon.$$

- Déterminons la matrice  $M$ , on a :

$$M = Z^T D Z = \begin{pmatrix} 106 & 70 \\ 70 & 58 \end{pmatrix}.$$

**Changement de plan :**

- Calculons la direction d'amélioration  $l$  :

on a :

$$l_{NN} = \begin{pmatrix} l_3^{(3)} \\ l_4^{(3)} \end{pmatrix} = \begin{pmatrix} 5.91 \\ -8.38 \end{pmatrix},$$

$$l(J_S) = 0,$$

$$l_B = -A_B^{-1} A_{NN} l_{NN} = \begin{pmatrix} 7.55 \\ -6.73 \end{pmatrix},$$

$$\text{d'où } l = \begin{pmatrix} 7.55 \\ -6.73 \\ 5.91 \\ -8.38 \end{pmatrix}.$$

- Calculons le pas  $\theta^0$  le long de cette direction :

$$\theta^0 = \min\{1, \theta_{j_1}, \theta_S, \theta_F\},$$

$$\theta_{j_0} = \min\{\theta_j, j \in J_B\} = 0.03 \text{ avec } j_0 = 1,$$

$$\theta_F = \sigma_{j_*} = 0.08 \text{ avec } j_* = 4 ,$$

$$\text{donc } \theta^0 = 0.03.$$

- Le nouveau plan est :

$$x^{(4)} = x^{(3)} + \theta^0 l = \begin{pmatrix} 2 & 0.85 & 0.27 & 0.13 \end{pmatrix},$$

$$\text{avec } Q(x^{(4)}) = 10.37.$$

Au bout de la 5<sup>ème</sup> itération, on trouve :

$$x^{(5)} = x^{(4)} + \theta^0 l = (2, 0.61, 0.78, 0.39) \text{ tel que } \beta(x^{(5)}, J_B^{(4)}) = 0,$$

de cette dernière égalité, on peut déduire que le plan de support  $\{x^{(5)}, J_B^{(4)}\}$  est optimal pour notre programme (3.10) , avec  $Q(x^{(5)}) = 9.27$ .

# Chapitre 4

## La méthode adaptée pour la résolution d'un programme linéaire

### 4.1 Introduction

Ce chapitre est consacré à présenter la méthode adaptée pour résoudre des programmes linéaires mono-objectif avec des variables non négatives. La méthode adaptée [40] est habituellement utilisée pour résoudre des programmes linéaires avec des variables bornées, l'avantage de cette méthode réside dans sa rapidité. En outre, la méthode intègre un critère de suboptimalité qui permet d'arrêter l'algorithme avec une précision souhaitée pour trouver des solutions  $\epsilon$ -optimales. Ce critère pourrait être utile dans des applications pratiques. Ce travail est soumis pour publication [23], il est motivé par sa nécessité dans un autre travail de programmation multiobjectif, dont les détails seront présentés dans le chapitre 7.

### 4.2 Position du problème et définitions

#### 4.2.1 Position du problème

Considérons le problème linéaire suivant :

$$\begin{cases} \max z(x) = c^T x, \\ Ax = b, \\ x \geq 0. \end{cases} \quad (4.1)$$

On définit les indices des contraintes et des variables de décision par :  $I =$

$\{1, 2, \dots, m\}$  et  $J = \{1, 2, \dots, n\}$ , tel que :

$c = (c_j, j \in J)$  et  $x = (x_j, j \in J)$  sont des vecteurs d'ordre  $n$ ,  $b = (b_i, i \in I)$  est un vecteur d'ordre  $m$ ,  $A = A(I, J)$  est une matrice d'ordre  $m \times n$ , tel que  $\text{rang} A = m < n$ .

Pour résoudre le problème (4.1) considérons le programme suivant :

$$\begin{cases} \max z(x) = c^T x, \\ Ax = b, \\ 0 \leq x \leq M_{\mathbb{R}^n}, \end{cases} \quad (4.2)$$

avec  $M_{\mathbb{R}^n}$  un vecteur dont toutes les composantes sont égales à  $M$ ;  $M$  étant un nombre réel positif aussi grand que l'on veut.

Soit  $J_B$  un sous-ensemble de  $J$  tel que  $J = J_B \cup J_N$  avec  $J_B \cap J_N = \phi$  et  $|J_B| = |I| = m$ .

En vertu de la partition de  $J$ , on peut fractionner les vecteurs  $c$  et  $x$ , ainsi que la matrice  $A$  de la manière suivante :

$$x = \begin{pmatrix} x_B \\ -- \\ x_N \end{pmatrix}, \quad x_B = x(J_B) = (x_j, j \in J_B), \quad x_N = x(J_N) = (x_j, j \in J_N),$$

$$c = \begin{pmatrix} c_B \\ -- \\ c_N \end{pmatrix}, \quad c_B = c(J_B), \quad c_N = c(J_N),$$

$$A = A(I, J) = (A_B | A_N), \quad A_B = A(I, J_B), \quad A_N = A(I, J_N).$$

### 4.2.2 Définitions

- Un vecteur  $x$  vérifiant les contraintes du problème (4.2) est appelé *plan* ou *solution réalisable* du problème. Ainsi l'ensemble des solutions réalisables est défini par :

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

- Un plan  $x^0$  est dit *optimal* dans le problème (4.2), si

$$z(x^0) = c^T x^0 = \max c^T x.$$

- D'autre part, un plan  $x^\epsilon$  est dit  $\epsilon$ -*optimal* ou *suboptimal* ( $\epsilon \in \mathbb{R}, \epsilon \geq 0$ ) dans le problème (4.2), si

$$z(x^0) - z(x^\epsilon) = c^T x^0 - c^T x^\epsilon \leq \epsilon,$$

où  $x^0$  est une solution *optimale* du problème (4.2).

- L'ensemble  $J_B \subset J$ ,  $|J_B| = m$ , est dit *support* si

$$\det A_B = \det A(I, J_B) \neq 0.$$

- Le couple  $\{x, J_B\}$  formé du plan  $x$  et du support  $J_B$  est appelé *plan de support*.
- Le plan de support est dit *non dégénéré* si  $0 < x_j < M$ ,  $\forall j \in J_B$ .
- On définit le vecteur des potentiels  $u$  par

$$u^T = c_B^T A_B^{-1},$$

ainsi que le vecteur des estimations  $E$  par

$$E^T = (E_B^T, E_N^T) = u^T A - c^T,$$

avec  $E_B = E(J_B) = 0$  et  
 $E_N^T = E^T(J_N) = c_B^T A_B^{-1} A_N - c_N^T.$

*Remarque 4.2.1.* Une solution réalisable de support est différente d'une solution réalisable de base. En effet, les variables non basiques d'une solution réalisable de base sont toutes nulles alors que celles d'une solution réalisable de support sont quelconques, c'est-à-dire positives ou nulles.

Ce chapitre est structuré comme suit : d'abord, nous allons définir le critère d'optimalité d'une solution réalisable. Ensuite, dans la section 4.4, nous allons donner la condition de suboptimalité d'une solution réalisable. Nous développons la méthode de résolution dans la section 4.5. L'algorithme détaillé est donné dans la section 4.6 et un exemple numérique pour démontrer l'applicabilité de la méthode suggérée est présenté dans la section 4.7, la section 4.8 est consacrée à présenter les expériences numériques. Enfin, nous clôturons notre chapitre par une conclusion dans la section 4.8.

### 4.3 Critère d'optimalité

Soit  $\{x, J_B\}$  un plan de support du problème (4.2) et considérons un autre plan quelconque  $\bar{x} = x + \Delta x$ . L'accroissement de la fonction objectif s'écrit alors :

$$\Delta z = c^T \bar{x} - c^T x = c^T \Delta x = -(c_B^T A_B^{-1} A_N - c_N^T) \Delta x_N = -E_N^T \Delta x_N = - \sum_{j \in J_N} E_j \Delta x_j.$$

Le critère d'optimalité pour le problème (4.2) est donné par le théorème suivant :

**Théorème 4.3.1.** Soit  $\{x, J_B\}$  un plan de support du problème (4.2).

Les relations suivantes :

$$\begin{cases} E_j \geq 0, & \text{si } x_j = 0, \quad j \in J_N, \\ E_j \leq 0, & \text{si } x_j = M, \quad j \in J_N, \\ E_j = 0, & \text{si } 0 < x_j < M, \quad j \in J_N, \end{cases} \quad (4.3)$$

sont suffisantes pour que le point  $x$  soit optimal.

Si le plan de support est non-dégénéré, alors ces mêmes relations sont aussi nécessaires pour que  $x$  soit optimal c'est-à-dire on aura l'équivalence.

*Démonstration. Suffisance :*

Soit  $\{x, J_B\}$  un plan de support réalisable du problème (4.2), vérifiant les relations (4.3). Pour tout plan  $\bar{x}$  du problème (4.2), la formule d'accroissement nous permet d'écrire :

$$\begin{aligned} z(\bar{x}) - z(x) &= - \sum_{j \in J_N, E_j > 0} E_j(\bar{x}_j - x_j) - \sum_{j \in J_N, E_j < 0} E_j(\bar{x}_j - x_j) \\ &= \sum_{j \in J_N, E_j > 0} E_j(-\bar{x}_j) + \sum_{j \in J_N, E_j < 0} E_j(M - \bar{x}_j). \end{aligned}$$

Comme  $0 \leq \bar{x} \leq M$ , on a  $\bar{x} \geq 0$  et  $\bar{x} - M \leq 0$ ; d'où  $z(\bar{x}) \leq z(x)$  pour tout plan  $\bar{x}$ . Le vecteur  $x$  est, par conséquent, une solution optimale du problème (4.2).

*Nécessité :*

Soit  $\{x, J_B\}$  un plan de support optimal non-dégénéré du problème (4.2) et supposons que les relations (4.3) ne sont pas vérifiées, c-à-d, qu'il existe au moins un indice  $j_0 \in J_N$ , tel que

$$\begin{aligned} E_{j_0} &> 0, \text{ pour } x_{j_0} > 0, \quad j_0 \in J_N, \\ &\text{ou} \\ E_{j_0} &< 0, \text{ pour } x_{j_0} < M, \quad j_0 \in J_N. \end{aligned}$$

On construit alors un autre plan  $\bar{x} = x + \theta^0 l$ , où  $\theta^0$  est un nombre réel positif, et  $l$  est un vecteur de direction que l'on construit comme suit :

$$\begin{cases} l_{j_0} = -\text{sign} E_{j_0}, \\ l_j = 0, \quad j \neq j_0, \quad j \in J_N. \end{cases}$$

D'autre part, on doit avoir

$$A\bar{x} = A(x + \theta^0 l) = b$$

$\Updownarrow$



$$Al = A_B l_B + A_N l_N = 0.$$

Donc

$$A_B l_B + A_N l_N = 0.$$

D'où

$$l_B = -A_B^{-1} A_N l_N = A_B^{-1} a_{j_0} \text{sign} E_{j_0}.$$

On a donc

$$\begin{cases} l_{j_0} = -\text{sign} E_{j_0}, \\ l_j = 0, j \neq j_0, j \in J_N, \\ l_B = A_B^{-1} a_{j_0} \text{sign} E_{j_0}. \end{cases}$$

Le vecteur  $\bar{x}$  vérifie la contrainte principale  $A\bar{x} = b$ . Pour que  $\bar{x}$  soit un plan du problème (4.2), il doit en plus vérifier l'inégalité  $0 \leq \bar{x} \leq M \Leftrightarrow 0 \leq x + \theta^0 l \leq M$ . Soit, en écrivant composante par composante

$$\begin{cases} -x_j \leq \theta^0 l_j \leq M - x_j, & j \in J_B, \\ -x_{j_0} \leq \theta^0 l_{j_0} \leq M - x_{j_0}. \end{cases} \quad (4.4)$$

Comme le plan de support est non-dégénéré, on trouve toujours un nombre  $\theta^0$  positif assez petit tel que le vecteur  $\bar{x}$  sera un plan pour le problème (4.2). Avec ce nouveau plan  $\bar{x}$ , on calcule :

$$\begin{aligned} z(\bar{x}) - z(x) &= \sum_{j \in J_N} E_j (\bar{x}_j - x_j) \\ &= -\theta^0 \sum_{j \in J_N} E_j l_j \\ &= -\theta^0 E_{j_0} l_{j_0} \\ &= \theta^0 E_{j_0} \text{sign} E_{j_0} \\ &= \theta^0 |E_{j_0}|. \end{aligned}$$

On a ainsi trouvé un plan  $x \neq \bar{x}$  tel que  $z(\bar{x}) > z(x)$ ; ce qui contredit le fait que  $x$  soit un plan optimal, donc les relations (4.3) sont forcément vérifiées si le plan de support est non-dégénéré.  $\square$

## 4.4 Critère de suboptimalité

Afin d'évaluer la différence entre la valeur optimale  $z(x^0)$  et une valeur  $z(x)$  pour toute solution de support réalisable  $\{x, J_B\}$ , on utilise la formule suivante :

$$\begin{aligned} \beta(x, J_B) &= \sum_{j \in J_N, E_j > 0} E_j (x_j) + \sum_{j \in J_N, E_j < 0} E_j (x_j - M) \\ &= \sum_{j \in J_N} E_j x_j - M \sum_{j \in J_N, E_j < 0} E_j, \end{aligned}$$

appelée *estimation de suboptimalité*.

De là, il s'ensuit que

$$\Delta z(x) = z(\bar{x}) - z(x) \leq \beta(x, J_B),$$

et pour  $\bar{x} = x^0$ , on a

$$z(x^0) - z(x) \leq \beta(x, J_B).$$

De cette dernière inégalité, on déduit le théorème suivant :

**Théorème 4.4.1.** (*Condition de Suboptimalité*).

Soit  $\{x, J_B\}$  un plan de support du problème (4.2) et  $\epsilon$  un nombre réel positif ou nul.

Si  $\beta(x, J_B) \leq \epsilon$ , alors  $x$  est  $\epsilon$ -optimal pour le problème (4.2).

Dans le cas particulier où  $\epsilon = 0$ ,  $x$  est optimal.

*Démonstration.*  $z(x^0) - z(x) \leq \beta(x, J_B) \leq \epsilon \Rightarrow z(x^0) - z(x) \leq \epsilon$ . D'où  $x$  est  $\epsilon$ -optimal.

□

## 4.5 Méthode de résolution

### 4.5.1 Construction d'une nouvelle solution réalisable

La méthode de résolution consiste à faire un changement de plan pour augmenter  $z(x)$  et un changement de support pour diminuer  $\Theta(\lambda)$ ;  $\Theta(\lambda)$  étant la fonction duale de  $z(x)$ .

La construction du nouveau plan  $\bar{x} = x + \theta^0 l$  consiste à choisir  $l \in \mathbb{R}^n$ , appelée direction d'amélioration, et un nombre non négatif  $\theta^0$  qui est le pas maximal le long de cette direction tel que  $z(\bar{x}) \geq z(x)$ .

$$l_j = \begin{cases} -x_j, & \text{si } E_j > 0, \\ M - x_j, & \text{si } E_j < 0, \\ 0, & \text{si } E_j = 0, \quad j \in J_N, \end{cases} \quad (4.5)$$

et  $l(J_B) = -A_B^{-1} A_N l(J_N)$ , pour avoir  $A\bar{x} = b$  et  $\bar{x}$  vérifie  $0 \leq \bar{x} \leq M$ .

D'autre part, le pas maximal  $\theta^0$  le long de la direction  $l$  doit vérifier la relation suivante :

$$-x_j \leq \theta^0 l_j \leq M - x_j, \quad j \in J_B.$$

Par conséquent, on a

$$\theta_j = \begin{cases} \frac{M-x_j}{l_j}, & \text{si } l_j > 0, \\ \frac{-x_j}{l_j}, & \text{si } l_j < 0, \\ \infty, & \text{si } l_j = 0, j \in J_B, \end{cases}$$

avec

$$\theta_{j_0} = \min_{j \in J_B} \theta_j,$$

le pas maximal sera

$$\theta^0 = \min(1, \theta_{j_0}).$$

Le nouveau plan  $\bar{x}$  s'écrit donc  $\bar{x} = x + \theta^0 l$ .

### 4.5.2 Calcul de $\beta(\bar{x}, J_B)$

La valeur de l'estimation de suboptimalité pour le nouveau plan  $\bar{x}$  est calculée comme suit :

$$\begin{aligned} \beta(\bar{x}, J_B) &= \sum_{E_j > 0, j \in J_N} E_j \bar{x}_j + \sum_{E_j < 0, j \in J_N} E_j (\bar{x}_j - M) \\ &= \sum_{E_j > 0, j \in J_N} E_j (x_j + \theta^0 l_j) + \sum_{E_j < 0, j \in J_N} E_j (x_j + \theta^0 l_j - M) \\ &= \sum_{E_j > 0, j \in J_N} E_j x_j + \sum_{E_j > 0, j \in J_N} E_j \theta^0 l_j + \sum_{E_j < 0, j \in J_N} E_j (x_j - M) + \sum_{E_j < 0, j \in J_N} E_j \theta^0 l_j \\ &= \beta(x, J_B) + \sum_{E_j > 0, j \in J_N} E_j \theta^0 l_j + \sum_{E_j < 0, j \in J_N} E_j \theta^0 l_j \\ &= \beta(x, J_B) + \theta^0 \left( \sum_{E_j > 0, j \in J_N} E_j l_j + \sum_{E_j < 0, j \in J_N} E_j l_j \right). \end{aligned}$$

Ainsi, on obtient :

$$\beta(\bar{x}, J_B) = \beta(x, J_B) + \theta^0 \left( \sum_{E_j > 0, j \in J_N} E_j l_j + \sum_{E_j < 0, j \in J_N} E_j l_j \right). \quad (4.6)$$

En remplaçant les  $l_j$  donnés par les relations (4.5), on a

$$\beta(\bar{x}, J_B) = (1 - \theta^0) \beta(x, J_B).$$

De cette dernière expression, on déduit que :

- Si  $\theta^0 = 1$  alors  $\bar{x}$  est optimal pour le problème (4.2).
- Si  $\beta(\bar{x}, \bar{J}_B) < \epsilon$ , on peut arrêter l'algorithme avec  $\bar{x}$  est  $\epsilon$ -optimal pour le problème (4.2).
- Si  $\beta(\bar{x}, \bar{J}_B) > \epsilon$ , on recommencera alors une nouvelle itération avec le nouveau plan de support  $\{\bar{x}, \bar{J}_B\}$ , on changera  $J_B$  de la manière suivante :

### 4.5.3 Changement de support

Le changement de support consiste à faire un changement de coplan  $E$  vers  $\bar{E}$ , et du vecteur des potentiels  $u$  vers  $\bar{u}$  de telle sorte que :

$$\beta(\bar{x}, J_B) \leq \beta(\bar{x}, \bar{J}_B).$$

Pour cela, on pose :

$$\bar{E}_j = E_j + \sigma_0 t_j,$$

$$\bar{u}_j = u_j + \sigma_0 t_j,$$

où  $t$  est la direction de diminution de la fonction duale,  $\sigma_0$  est le pas maximal le long de cette direction.

Après calcul du plan  $\bar{x} = x + \theta^0 l$ , le pas  $\theta^0$  est donné par :

$$\theta^0 = \min(1, \theta_{j_0}) = \theta_{j_0}, \quad j_0 \in J_B.$$

On cherchera un indice  $j_1 \in J_N$  qui va entrer dans la base à la place de  $j_0$ .

Pour cela, posons :

$$t_j = \begin{cases} -\text{sign}(l_{j_0}), & \text{si } j = j_0, \\ 0, & \text{si } j \in J_B \setminus j_0, \end{cases}$$

$$t^T(J_N) = t^T(J_B)A_B^{-1}A_N,$$

et calculons

$$\sigma_0 = \sigma_{j_1} = \min_{j \in J_N}(\sigma_j)$$

avec

$$\sigma_j = \begin{cases} \frac{-E_j}{t_j} & , \quad \text{si } E_j t_j < 0, \\ 0 & , \quad \text{si } (E_j = 0 \text{ et } x_j \neq 0 \text{ pour } t_j > 0) \text{ ou} \\ & (E_j = 0 \text{ et } x_j \neq M \text{ pour } t_j < 0), \quad j \in J_N, \\ \infty & , \quad \text{sinon .} \end{cases}$$

On a  $\bar{E}_{j_1} = 0$ .

Le nouveau support est :  $\bar{J}_B = (J_B \setminus j_0) \cup j_1$ .

On peut remarquer que :

$$\begin{aligned}
\beta(\bar{x}, \bar{J}_B) &= \sum_{\bar{E}_j > 0, j \in J_N} \bar{E}_j \bar{x}_j + \sum_{\bar{E}_j < 0, j \in J_N} \bar{E}_j (\bar{x}_j - M) \\
&= \sum_{E_j > 0, j \in J_N} (E_j + \sigma_0 t_j) \bar{x}_j + \sum_{E_j < 0, j \in J_N} (E_j + \sigma_0 t_j) (\bar{x}_j - M) \\
&= \sum_{E_j > 0, j \in J_N} E_j \bar{x}_j + \sum_{E_j < 0, j \in J_N} E_j (\bar{x}_j - M) \\
&\quad + \sigma_0 \sum_{E_j > 0, j \in J_N} t_j \bar{x}_j + \sigma_0 \sum_{E_j < 0, j \in J_N} t_j (\bar{x}_j - M) \\
&= \beta(\bar{x}, J_B) + \sigma_0 \left( \sum_{E_j > 0, j \in J_N} t_j \bar{x}_j + \sum_{E_j < 0, j \in J_N} t_j (\bar{x}_j - M) \right) \\
&= (1 - \theta^0) \beta(\bar{x}, J_B) + \sigma_0 \left( \sum_{E_j > 0, j \in J_N} t_j \bar{x}_j + \sum_{E_j < 0, j \in J_N} t_j (\bar{x}_j - M) \right).
\end{aligned}$$

Posons

$$\begin{aligned}
\alpha_0 &= \sum_{E_j > 0, j \in J_N} t_j \bar{x}_j + \sum_{E_j < 0, j \in J_N} t_j (\bar{x}_j - M). \\
&= (1 - \theta^0) \sum_{j \in J_N} t_j l_j \\
&= (1 - \theta^0) t_{j_0} l_{j_0}.
\end{aligned}$$

On a

$$\alpha_0 = (1 - \theta^0) t_{j_0} l_{j_0} = \begin{cases} (1 - \theta^0) l_{j_0} & \text{si } t_{j_0} = 1, \\ (\theta^0 - 1) l_{j_0} & \text{si } t_{j_0} = -1. \end{cases}$$

Donc

$$\beta(\bar{x}, \bar{J}_B) = (1 - \theta^0) \beta(\bar{x}, J_B) + \sigma_0 \alpha_0.$$

De cette dernière expression, on déduit que :

- Si  $\beta(\bar{x}, \bar{J}_B) > \epsilon$ , on recommencera alors une nouvelle itération avec le nouveau plan de support  $\{\bar{x}, \bar{J}_B\}$ .
- Si  $\beta(\bar{x}, \bar{J}_B) < \epsilon$ , on peut arrêter l'algorithme avec  $\bar{x}$  est  $\epsilon$ -optimal.
- Si  $\beta(\bar{x}, \bar{J}_B) = 0$ , alors on arrête la procédure en ayant trouvé notre solution optimale.

## 4.6 Algorithme de la méthode

Soit un plan de support initial  $\{x, J_B\}$  et  $\epsilon \geq 0$ .

1. Calculer le vecteur des estimations :

$$E^T = (E_B^T, E_N^T) = c_B^T A_B^{-1} A - c^T.$$

2. Tester l'optimalité de la solution du support réalisable  $\{x, J_B\}$ .
  - Calculer la valeur de suboptimalité  $\beta(x, J_B)$ ,
  - Si  $\beta(x, J_B) = 0$ , alors on peut arrêter l'algorithme,  $\{x, J_B\}$  est optimal ;
  - Si  $\beta(x, J_B) \leq \epsilon$ , on peut arrêter l'algorithme avec  $\{x, J_B\}$  est  $\epsilon$ -optimal ;
  - Si  $\beta(x, J_B) > \epsilon$ , aller à (3).
3. Changer la solution réalisable  $x$  par  $\bar{x} : \bar{x} = x + \theta^0 l$ .
  - Calculer le vecteur  $l$ .
  - Calculer le pas maximal  $\theta^0$ .
  - Choisir l'indice  $j_0$ .
    - (i) Si  $\theta^0 = \infty$  alors la fonction objective n'est pas bornée, on arrête l'algorithme.
    - (ii) Sinon, calculer le nouveau point  $\bar{x} = x + \theta^0 l$ .
4. Tester l'optimalité de la nouvelle solution réalisable  $\bar{x}$ .
  - Calculer  $\beta(\bar{x}, J_B) = (1 - \theta^0)\beta(x, J_B)$  ;
  - Si  $\theta^0 = 1$ , alors on peut arrêter l'algorithme avec  $\{\bar{x}, J_B\}$  est optimal,
  - Si  $\beta(\bar{x}, J_B) \leq \epsilon$ , alors on peut arrêter l'algorithme avec  $\{\bar{x}, J_B\}$  est  $\epsilon$ -optimal,
  - Si  $\beta(\bar{x}, J_B) > \epsilon$ , aller à (5).
5. Changement de support  $J_B$  par  $\bar{J}_B$ 
  - Calculer le vecteur  $t$  ;
  - Calculer  $\sigma_0$ , et déterminer l'indice  $j_1$  ;
  - Déterminer le nouveau support  $\bar{J}_B = (J_B \setminus j_0) \cup j_1$  ;
  - Calculer  $\beta(\bar{x}, \bar{J}_B)$  :
    - Si  $\beta(\bar{x}, \bar{J}_B) = 0$ , alors on peut arrêter l'algorithme avec  $\{\bar{x}, \bar{J}_B\}$  est optimal ;
    - Si  $\beta(\bar{x}, \bar{J}_B) \leq \epsilon$ , alors on peut arrêter l'algorithme avec  $\{\bar{x}, \bar{J}_B\}$  est  $\epsilon$ -optimal ;
    - Si  $\beta(\bar{x}, \bar{J}_B) > \epsilon$ , aller à (1), on passe à une nouvelle itération avec le nouveau plan de support  $\{\bar{x}, \bar{J}_B\}$ .

## 4.7 Étude d'un exemple numérique

Nous allons illustrer le développement théorique de la méthode en considérant le programme linéaire suivant :

$$\begin{cases} \max z(x) = -x_1 + 3x_2, \\ 3x_1 - 2x_2 \leq 7, \\ -x_1 + 4x_2 \leq 9, \\ -2x_1 + 3x_2 \leq 6, \\ x_1, x_2 \geq 0. \end{cases} \quad (4.7)$$

On introduit les variables d'écart, ce qui conduit au problème suivant :

$$\begin{cases} \max z(x) = -x_1 + 3x_2, \\ 3x_1 - 2x_2 + x_3 = 7, \\ -x_1 + 4x_2 + x_4 = 9, \\ -2x_1 + 3x_2 + x_5 = 6, \\ x_1, x_2, x_3, x_4, x_5 \geq 0. \end{cases} \quad (4.8)$$

En s'inspirant de notre méthode, résoudre le P.L (4.8) revient à résoudre le P.L suivant :

$$\begin{cases} \max z(x) = -x_1 + 3x_2, \\ 3x_1 - 2x_2 + x_3 = 7, \\ -x_1 + 4x_2 + x_4 = 9, \\ -2x_1 + 3x_2 + x_5 = 6, \\ 0 \leq x_i \leq M, \text{ pour } i = \overline{1,5}. \end{cases} \quad (4.9)$$

On a

$$A = \begin{pmatrix} 3 & -2 & 1 & 0 & 0 \\ -1 & 4 & 0 & 1 & 0 \\ -2 & 3 & 0 & 0 & 1 \end{pmatrix}, \quad c^T = \begin{pmatrix} -1 & 3 & 0 & 0 & 0 \end{pmatrix}, \quad b = \begin{pmatrix} 7 \\ 9 \\ 6 \end{pmatrix}.$$

Posons  $J_N = \{1, 2\}$  et  $J_B = \{3, 4, 5\}$ .

Le point  $x^0 = (0 \ 0 \ 7 \ 9 \ 6)^T$  est une solution réalisable pour notre P.L (4.9).

**1<sup>ère</sup> iteration**

- Calculons  $E^T = (1 \ -3 \ 0 \ 0 \ 0)$ .
- Calculons la valeur de suboptimalité :

$$\beta(x^0, J_B) = 3M - 3.$$

- Calculons le vecteur  $l$  :

$$l = \begin{pmatrix} 0 \\ M-1 \\ 2M-1 \\ -4M+4 \\ -3M+4 \end{pmatrix}.$$

- Calculons  $\theta$  :

$$\theta = \begin{pmatrix} 1 \\ 1 \\ \frac{M-9}{2M-1} \\ \frac{-5}{-4M+4} \\ \frac{-3}{-3M+4} \end{pmatrix}.$$

- Choix de  $j_0$  :

$$\theta_{j_0} = \frac{-3}{-3M+4}, \text{ avec } j_0 = 5.$$

- Calculons  $\theta^0$  :

$$\theta^0 = \min\left(1, \frac{-3}{-3M+4}\right) = \frac{-3}{-3M+4}.$$

- Calculons  $x^{(1)}$  :

$$x^{(1)} = x^0 + \theta^0 l = \begin{pmatrix} 0 & 2 & 11 & 1 & 0 \end{pmatrix}^T.$$

- Calculons la nouvelle valeur de suboptimalité  $\beta(x^{(1)}, J_B)$  :

$$\beta(x^{(1)}, J_B) = \frac{(-3M+7)(3M-3)}{(-3M+4)} > 0.$$

Ainsi  $\{x^{(1)}, J_B\}$  n'est pas optimal.

#### Changement de support

- Calculons le vecteur  $t$  :

$$t = \begin{pmatrix} -2 \\ 3 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

- Calculons  $\sigma$  :

$$\sigma = \begin{pmatrix} 1/2 \\ 1 \end{pmatrix}.$$

- Déterminons l'indice  $j_1$  :

$$\sigma_0 = \sigma_{j_1} = \frac{1}{2},$$

d'où  $j_1 = 1$ .



- Notre nouveau support est :

$$J_B^{(1)} = (J_B \setminus j_0) \cup j_1 = \{3, 4, 1\}.$$

- Calculons la nouvelle valeur de suboptimalité  $\beta(x^{(1)}, J_B^{(1)})$  :

$$\beta(x^{(1)}, J_B^{(1)}) = \frac{3}{2}(M - 2) > 0.$$

Ainsi  $\{x^{(1)}, J_B^{(1)}\}$  n'est pas optimal.

### 2<sup>ème</sup> itération

- Calculons

$$E^T = \left( 0 \quad \frac{-3}{2} \quad 0 \quad 0 \quad \frac{1}{2} \right).$$

- Calculons  $l$  :

$$l = \begin{pmatrix} \frac{3M-6}{2} \\ M-2 \\ \frac{10-5M}{2} \\ \frac{10-5M}{2} \\ 0 \end{pmatrix}.$$

- Calculons  $\theta^0$  :

$$\theta = \begin{pmatrix} \frac{2M}{3M-6} \\ 1 \\ \frac{-22}{10-5M} \\ \frac{-2}{10-5M} \\ 1 \end{pmatrix},$$

$$\theta_{j_0} = \frac{-2}{10-5M},$$

alors

$$j_0 = 4.$$

Nous avons :

$$\theta^0 = \min\left(1, \frac{-2}{10-5M}\right) = \frac{-2}{10-5M}.$$

- Calculons  $x^{(2)}$  :

$$x^{(2)} = x^{(1)} + \theta^0 l = \left( \frac{3}{5} \quad \frac{12}{5} \quad 10 \quad 0 \quad 0 \right)^T.$$

- Calculons  $\beta(x^{(2)}, J_B^{(1)})$  :

$$\beta(x^{(2)}, J_B^{(1)}) = \frac{3}{10}(5M - 12) > 0.$$

Ainsi  $\{x^{(2)}, J_B^{(1)}\}$  n'est pas optimal.

**Changement de support :**

- Calculons

$$t = \begin{pmatrix} 0 \\ \frac{5}{2} \\ 0 \\ 1 \\ -\frac{1}{2} \end{pmatrix}.$$

- Calculons  $\sigma$  :

$$\sigma = \begin{pmatrix} 0.6 \\ 1 \end{pmatrix}.$$

- Calculons  $\sigma_0$  :

$$\sigma_0 = \sigma_{j_1} = 0.6,$$

alors

$$j_1 = 2.$$

- Notre nouveau support est :

$$J_B^{(2)} = (J_B^{(1)} \setminus j_0) \cup j_1 = \{3, 2, 1\}.$$

- Calcul de la nouvelle valeur de suboptimalité  $\beta(x^{(2)}, J_B^{(2)})$  :

$$\beta(x^{(2)}, J_B^{(2)}) = 0.$$

Alors, on arrête l'algorithme avec  $\{x^{(2)}, J_B^{(2)}\}$  est optimal pour le problème (4.8).

## 4.8 Expériences numériques

Afin d'effectuer une comparaison numérique entre la méthode adaptée et les méthodes de Linprog Solver incorporées dans Matlab Toolbox, une implémentation sous la version Matlab 8.1.0.604 a été développée. Nous avons résolu les 620 problèmes test listés dans le Tableau 4.1, le Tableau 4.2 et le Tableau 4.3 sur un ordinateur personnel avec Intel Celeron N2830 Cadencé avec CPU 2.17 GHz et 2 Go de RAM, fonctionnant sous Windows 7. La comparaison est basée sur des problèmes de programmation linéaire générés aléatoirement. La caractéristique de ces problèmes est que la solution optimale  $x$  et le vecteur  $c$  sont connus auparavant. Dans notre implémentation, les paramètres d'entrée de la procédure qui génèrent des problèmes linéaires aléatoires sont  $n$  et  $m$ , respectivement, ils représentent le nombre de variables et le nombre de contraintes. Le solveur de programmes linéaires de Matlab est Linprog, il comporte trois algorithmes : interior-point-legacy,

interior-point et dual-simplex. Les résultats numériques sont rapportés dans les tableaux 4.1, 4.2 et 4.3 où les  $T(s)$  et les «Itérs» représentent respectivement la moyenne du temps CPU en secondes et la moyenne du nombre d'itérations des dix problèmes générés pour chaque taille de problème.

- ✓ La Table 4.1 rapporte les résultats numériques pour des problèmes linéaires, où chaque taille de problème vérifie  $n - m \leq 5$ .  $T(s)$  Ratio et  $Itérs$  Ratio représentent, respectivement, les rapports de temps de CPU (temps de CPU de Linprog solver) / (temps de CPU de la méthode Adaptée) et les rapports d'itérations (nombre d'itérations de Linprog solver / nombre d'itérations de la méthode adaptée) pour la taille correspondante.

Dimension du problème $n \times m$	La méthode adaptée		Linprog Solver		Linprog/Adaptée	
	<i>Itérs</i>	$T(s)$	<i>Itérs</i>	$T(s)$	<i>Itérs</i> Ratio	$T(s)$ Ratio
$10 \times 10$	1	$5.2552e^{-4}$	4	0.1087	4	200
$10 \times 8$	2	0.0483	7	0.0716	3.5	9.4203
$10 \times 6$	6	0.0366	8	0.0433	1.3333	1.1830
$10 \times 2$	8	0.0675	8	0.04821	1	0.7142
$10 \times 1$	8	0.0776	8	0.04993	1	0.6434
$100 \times 100$	1	0.0192	4	0.1945	4	10.1302
$100 \times 98$	2	0.0769	6	0.1082	3	1.4070
$100 \times 96$	7	0.1009	9	0.1937	1.2857	1.9197
$100 \times 95$	7	0.1091	7	0.1419	1	1.3006
$200 \times 200$	1	0.1105	4	0.3410	4	3.0859
$200 \times 198$	2	0.1040	6	0.3582	3	3.4442
$200 \times 196$	6	0.1975	7	0.3937	1.1666	1.9934
$200 \times 195$	7	0.2509	7	0.4352	1	1.7345
$300 \times 300$	1	0.1322	4	0.9757	4	7.3804
$300 \times 298$	2	0.1560	6	1.1976	3	7.6769
$300 \times 296$	6	0.5309	7	1.3690	1.1666	2.5786
$300 \times 295$	8	0.5494	8	1.5316	1	2.7877
$400 \times 400$	1	0.1920	4	2.1882	4	11.3968
$400 \times 398$	2	0.3152	6	2.9825	3	9.4622
$400 \times 396$	6	1.4099	7	3.3398	1.1666	2.3688
$400 \times 395$	7	1.6920	7	3.3879	1	2.0023
$500 \times 500$	1	0.2131	4	4.013	4	18.8315
$500 \times 498$	2	0.4118	6	5.7130	3	13.8732
$500 \times 496$	5	1.3280	6	5.6122	1.2000	4.2260
$500 \times 495$	7	2.1543	7	5.7565	1	2.6720
Mean	4.3846	0.3953	6.2692	1.5617	2.2128	12.4244

TABLE 4.1 – Résultats numériques pour les problèmes linéaires test

Nous traçons le temps CPU pour les deux méthodes. Le graphe est affiché dans la Figure 4.1.

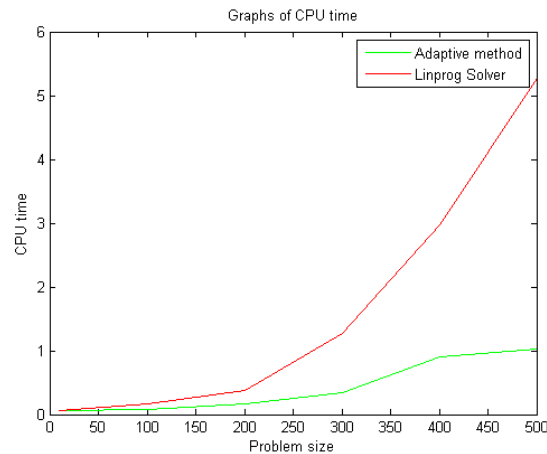


FIGURE 4.1 – Graphe du temps CPU

Lorsque la différence entre le nombre de contraintes et le nombre de variables de décision de tous les problèmes test qui ont été résolus est faible, nous notons que Linprog Solver nécessite plus d'itérations par rapport à la méthode adaptée. Donc, notre méthode adaptée dépasse Linprog Solver pour résoudre presque tous les problèmes test générés de manière aléatoire. En effet, la méthode adaptée est en moyenne 2.2128 fois plus rapide que Linprog Solver en termes de temps CPU et résout les problèmes d'essai 12.4244 fois plus rapidement que Linprog Solver en termes de nombre d'itérations. En résumé, nous constatons que lorsque  $n - m \leq 5$ , notre méthode est très efficace par rapport au Linprog Solver.

- ✓ La Table 4.2 rapporte les résultats numériques pour des problèmes linéaires de grande dimension.

Dimension du problème $n \times m$	La méthode adaptée		Linprog Solver	
	<i>Itérs</i>	<i>T(s)</i>	<i>Itérs</i>	<i>T(s)</i>
600 × 600	1	0.2728	-	-
700 × 700	1	0.2820	-	-
800 × 800	1	0.4337	-	-
900 × 900	1	0.5538	-	-
1000 × 1000	1	0.6011	-	-
1000 × 998	2	2.4118	-	-
1000 × 996	5	6.5611	-	-
1000 × 994	10	19.1011	-	-
1000 × 992	21	38.3422	-	-
1000 × 990	28	59.0012	-	-
2000 × 2000	1	4.6192	-	-
2000 × 1998	2	20.7356	-	-
2000 × 1996	4	48.1953	-	-
2000 × 1995	6	84.7458	-	-
3000 × 3000	1	26.9410	-	-

TABLE 4.2 – Résultats numériques pour les problèmes test avec de grande dimension

Nous constatons ici que l'efficacité de notre approche est clairement démontrée. Donc, quand nous avons  $n > 500$  Linprog Solver nous affiche "Maximum variable size allowed by the program is exceeded" mais notre méthode nous retourne la solution optimale et la valeur optimale de la fonction objective.

- ✓ La Table 4.3 rapporte des résultats numériques pour des problèmes linéaires lorsque nous spécifions dans les options de Linprog Solver l'utilisation de la méthode du simplexe.

Dimension du problème $n \times m$	La méthode adaptée		La méthode du simplexe		Adaptée/ Simplexe	Simplex/ Adaptive
	<i>Itérs</i>	<i>T(s)</i>	<i>Itérs</i>	<i>T(s)</i>	<i>Itérs</i> Ratio	<i>T(s)</i> Ratio
100 × 100	1	0.0192	1	0.4903	1	25.5364
100 × 90	23	0.2459	18	0.4089	1.2777	1.6628
100 × 85	24	0.3592	19	0.3561	1.2631	0.9913
100 × 82	40	0.3688	25	0.3824	1.6000	1.0368
200 × 200	1	0.1105	1	2.9224	1	26.4470
200 × 190	23	0.7941	16	2.8482	1.4375	3.5867
200 × 180	60	2.3421	25	3.0760	2.4000	1.3133
200 × 172	76	2.4519	44	2.5114	1.7272	1.0242
300 × 300	1	0.1322	1	10.4257	1	78.863
300 × 290	19	1.7033	11	11.7611	1.7272	6.9048
300 × 280	58	4.9132	39	10.8693	1.4871	2.2122
300 × 270	111	15.5494	48	10.3319	2.3125	0.6644
400 × 400	1	0.1920	1	35.8304	1	186.6166
400 × 390	19	3.9109	19	31.2115	1	7.9806
400 × 380	76	10.4119	44	39.3398	1.7272	3.7783
400 × 370	98	22.9960	66	30.7865	1.4848	1.3387
400 × 360	136	23.4680	87	31.9438	1.5632	1.3611
500 × 500	1	0.2131	1	87.2340	1	409.3571
500 × 490	30	22.2344	-	83.2719	-	3.7451
500 × 480	70	22.6654	-	69.7956	-	3.0793
500 × 470	85	34.5860	-	77.1454	-	2.2305
500 × 460	154	41.8694	-	58.4778	-	1.3966
Mean	50.3181	9.6153	25.8888	27.3373	1.4448	35.0512

TABLE 4.3 – Résultats numériques pour les problèmes aléatoires

Nous traçons le graphe du temps CPU pour les deux méthodes. Le graphe est illustré dans la Figure 4.2.

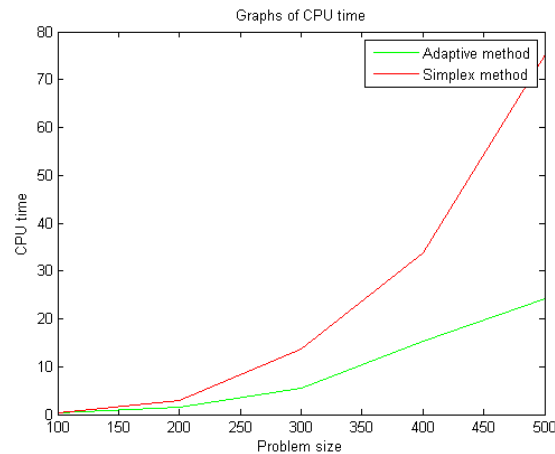


FIGURE 4.2 – Graphe du temps CPU

Notons que le nombre d'itérations est plus grand en utilisant notre méthode, mais le temps d'exécution est inférieur dans la majorité des problèmes générés.

En effet, la méthode adaptée est en moyenne 35.0512 fois plus rapide que la méthode du simplexe en termes de temps CPU. Mais la méthode du simplexe résout les problèmes test 1.4448 fois plus vite que la méthode adaptée en termes d'itérations. Ainsi, notre méthode reste meilleure que la méthode du simplexe étant donnée que le temps d'exécution est beaucoup mieux avec notre méthode.

Dans nos futurs travaux, nous effectuerons les expériences numériques. Nous allons inclure une phase de recherche d'une solution réalisable initiale. Nous comparerons notre algorithme sur des problèmes linéaires réels "benchmarks" afin d'obtenir des résultats plus raffinés.

## 4.9 Conclusions

Dans ce chapitre, nous avons proposé une méthode pour résoudre un programme linéaire mono-objectif avec des variables non négatives.

La particularité de notre méthode est qu'elle utilise un critère de suboptimalité qui nous permet d'arrêter l'algorithme avec une précision souhaitée. Notre méthode est efficace, rapide, simple et permet une réduction de temps dans l'ensemble du processus d'optimisation. En effet, l'étude de la section 8 indique que lorsque la différence entre le nombre de variables de décision et le nombre de contraintes est assez faible, la méthode adaptée est jusqu'à 12 fois plus rapide que Linprog solver en termes de nombre d'itérations et jusqu'à 35 fois plus rapide que la méthode du simplexe en termes de temps CPU.



Deuxième partie

**Optimisation multiobjectif**

# Chapitre 5

## Préliminaires et considérations théoriques

### 5.1 Introduction

Le but de ce chapitre est de présenter les différents concepts utilisés en programmation multiobjectif en précisant les types de décision qu'il est possible de prendre.

### 5.2 Concepts et terminologie

#### 5.2.1 Les alternatives et les critères

##### Les alternatives

L'ensemble des alternatives est l'ensemble des solutions réalisables. Cet ensemble peut être défini de deux manières :

- explicitement, par un ensemble fini d'alternatives de cardinal relativement faible,
- implicitement, par un ensemble de propriétés ou de conditions que les alternatives doivent vérifier (c'est le cas d'un ensemble d'alternatives spécifié par ses variables de décision vérifiant un système de contraintes explicites).

##### Les objectifs (Les critères)

Un objectif indique le sens de l'amélioration qu'un décideur souhaite apporter à un système lors d'un changement d'état. Il reflète l'aspiration du décideur. Les trois manières de poursuivre un objectif sont de le maximiser, de le minimiser ou de le maintenir dans un certain état. Des exemples industriels classiques de ces situations

sont : maximiser le profit, minimiser le coût ou maintenir un équilibre économique. Des auteurs ajoutent à ces situations d'autres types d'objectifs comme : près d'une cible (but), plus grand ou plus petit qu'un certain seuil, dans un intervalle, etc...

### 5.2.2 Paradigme des problèmes de décision multiobjectif

On peut répartir les méthodes d'optimisation multiobjectif en trois grandes familles. Ces familles sont définies en fonction de l'instant où l'on prend la décision d'effectuer ou non un compromis entre les fonctions objectif. Ces trois familles sont :

#### 1) Les méthodes d'optimisation a priori :

Dans ces méthodes, le compromis que l'on désire effectuer entre les différentes fonctions objectif a été déterminé avant l'exécution de la méthode d'optimisation. Pour obtenir la solution de notre problème, il suffira de faire une et une seule recherche et, en fin d'optimisation, la solution obtenue reflétera le compromis que l'on désirait effectuer entre les fonctions objectif avant de lancer la recherche. Ces méthodes sont intéressantes dans le sens où il suffit d'une seule recherche pour trouver la solution. Cependant, il ne faut pas oublier le travail de modélisation du compromis qui a été effectué avant d'obtenir ce résultat. Il ne faut pas, non plus, oublier que le décideur est "humain" et qu'il sera susceptible de constater que la solution obtenue en fin d'optimisation ne le satisfait finalement pas et qu'il souhaite maintenant, après avoir examiné cette solution, une solution qui opère un autre compromis entre les fonctions objectif.

#### 2) Les méthodes d'optimisation progressive :

Dans ces méthodes, on cherchera à questionner le décideur au cours de l'optimisation afin que celui-ci puisse réorienter la recherche vers des zones susceptibles de contenir des solutions qui satisfassent les compromis qu'il souhaite opérer entre les fonctions objectif. Ces méthodes, bien qu'appliquant une technique originale pour modéliser les préférences du décideur, présentent l'inconvénient de monopoliser l'attention du décideur tout au long de l'optimisation. Cet inconvénient n'est pas majeur lorsque l'on a affaire à des problèmes d'optimisation pour lesquels la durée d'une évaluation de la fonction objectif n'est pas importante. Pour les autres problèmes, l'utilisation d'une telle méthode peut être délicate. Il est en effet difficile de monopoliser l'attention du décideur pendant une période qui peut s'étendre sur plusieurs heures en lui posant, de plus, des questions toutes les dix minutes.

### 3) Les méthodes a posteriori :

Dans ces méthodes, on va chercher un ensemble de solutions bien réparties dans l'espace de solutions. Le but sera ensuite de proposer ces solutions au décideur pour qu'il puisse sélectionner la solution qui le satisfait le plus en jugeant les différentes solutions proposées. Ici, il n'est plus nécessaire de modéliser les préférences du décideur. On se contente de produire un ensemble de solutions que l'on transmettra au décideur. Il y a donc un gain de temps non négligeable vis-à-vis de la phase de modélisation des préférences de la famille des méthodes a priori. L'inconvénient qu'il nous faut souligner est que, maintenant, il faut générer un ensemble de solutions bien réparties. Dans notre travail, nous nous sommes intéressés à ce type de méthodes.

## 5.2.3 Les différentes problématiques multiobjectif

### 1) Problématique de choix multiobjectif

Cette problématique vise à identifier un ensemble aussi réduit que possible d'alternatives (éventuellement une alternative) meilleures que les autres. Cet ensemble est constitué d'alternatives entre lesquelles il est difficile de choisir. L'optimisation est un exemple de cette problématique.

### 2) Problématique de rangement multiobjectif

Cette problématique vise à ranger les alternatives suivant un ordre de préférence. Elle établit un ordre ou préordre complet ou partiel sur l'ensemble des alternatives.

## 5.3 Les structures de préférences

La plupart des modèles de décision multiobjectif utilisent la notion fondamentale de structure de préférence. Pour l'introduire, nous rappelons les définitions suivantes :

### 5.3.1 Les relations binaires et les ordres

Soit  $\mathcal{S}$  un ensemble. Une relation binaire sur  $\mathcal{S}$  est un sous-ensemble  $\mathcal{R}$  de  $\mathcal{S} \times \mathcal{S}$ . Nous introduisons quelques propriétés de relations binaires.

**Définition 5.3.1.** Une relation binaire  $\mathcal{R}$  sur  $\mathcal{S}$  est appelée

- réflexive si  $(s, s) \in \mathcal{R}$  pour tout  $s \in \mathcal{S}$ ,
- irreflexive si  $(s, s) \notin \mathcal{R}$  pour tout  $s \in \mathcal{S}$ ,

- symétrique si  $(s^1, s^2) \in \mathcal{R} \Rightarrow (s^2, s^1) \in \mathcal{R}$  pour tous  $s^1, s^2 \in \mathcal{S}$ ,
- asymétrique si  $(s^1, s^2) \in \mathcal{R} \Rightarrow (s^2, s^1) \notin \mathcal{R}$  pour tous  $s^1, s^2 \in \mathcal{S}$ ,
- antisymétrique si  $(s^1, s^2) \in \mathcal{R}$  et  $(s^2, s^1) \in \mathcal{R} \Rightarrow s^2 = s^1$  pour tous  $s^1, s^2 \in \mathcal{S}$ ,
- transitive si  $(s^1, s^2) \in \mathcal{R}$  et  $(s^2, s^3) \in \mathcal{R} \Rightarrow (s^1, s^3) \in \mathcal{R}$  pour tous  $s^1, s^2, s^3 \in \mathcal{S}$ ,
- négativement transitive si  $(s^1, s^2) \notin \mathcal{R}$  et  $(s^2, s^3) \notin \mathcal{R} \Rightarrow (s^1, s^3) \notin \mathcal{R}$  pour tous  $s^1, s^2, s^3 \in \mathcal{S}$ ,
- connexe si  $(s^1, s^2) \in \mathcal{R}$  ou  $(s^2, s^1) \in \mathcal{R}$  pour tous  $s^1, s^2 \in \mathcal{S}$  avec  $s^1 \neq s^2$ ,
- fortement connexe (ou total) si  $(s^1, s^2) \in \mathcal{R}$  ou  $(s^2, s^1) \in \mathcal{R}$  pour tous  $s^1, s^2 \in \mathcal{S}$ ,

**Définition 5.3.2.** Une relation binaire  $\mathcal{R}$  sur un ensemble  $\mathcal{S}$  est

- une relation d'équivalence si elle est réflexive, symétrique et transitive,
- un préordre (quasi-ordre) si elle est réflexive et transitive.

Au lieu de  $(s^1, s^2) \in \mathcal{R}$ , nous écrirons aussi  $s^1 \mathcal{R} s^2$ .

**Définition 5.3.3.** La relation binaire  $\mathcal{R}$  sur  $\mathcal{S}$  est

- un préordre total si elle est réflexive, transitive et connectée,
- un ordre total s'il s'agit d'un préordre total antisymétrique,
- un ordre faible strict s'il est asymétrique et négativement transitif.

A partir des préordres totaux, des ordres faibles stricts peuvent être obtenus et vice versa, comme le montre la proposition suivante :

**Définition 5.3.4.** La relation binaire  $\mathcal{R}$  est

- un ordre partiel si elle est réflexive, transitive et antisymétrique,
- un ordre partiel strict si elle est asymétrique et transitive (ou, de manière équivalente, s'il est irreflexive et transitive).

### 5.3.2 Les structures de préférences

Nous adoptons la définition suivante pour une structure de préférence :

**Définition 5.3.5.** Une structure de préférence  $\mathcal{R}$  est définie sur  $\mathcal{S}$  par une relation binaire  $\mathcal{P}$  asymétrique et une relation binaire  $\mathcal{I}$  réflexive et symétrique.  $\mathcal{P}$  et  $\mathcal{I}$  sont deux sous-ensembles disjoints de  $\mathcal{S} \times \mathcal{S}$ .

$$s^1 \mathcal{R} s^2 \Leftrightarrow s^1 \mathcal{P} s^2 \text{ ou } s^1 \mathcal{I} s^2 \quad (\mathcal{R} = \mathcal{P} \cup \mathcal{I}).$$

Si une paire  $(s^1, s^2)$  appartient à la relation  $\mathcal{P}$ , alors " $s^1$  est préférable à  $s^2$ ". Nous utilisons aussi la notation  $s^1 \prec s^2$ , ou  $s^1 \succ s^2$ .

Si une paire  $(s^1, s^2)$  appartient à la relation  $\mathcal{I}$ , alors " $s^1$  est indifférent à  $s^2$ ". Nous

utilisons aussi la notation  $s^1 \sim s^2$ .

$\mathcal{R}$  est interprété comme une préférence ou indifférence. La notation  $(s^1, s^2) \in \mathcal{R}$  se lit " $s^1$  est préféré ou indifférent à  $s^2$ ". Généralement,  $s^1 \mathcal{R} s^2$  est noté par  $s^1 \preceq s^2$  ou  $s^1 \succeq s^2$ .

Si une paire  $(s^1, s^2) \notin \mathcal{P}$  et  $(s^1, s^2) \notin \mathcal{I}$  alors  $s^1$  et  $s^2$  sont incomparables. La relation d'incomparabilité  $\mathcal{R}$  est défini par

$$\mathcal{R} = \{(s^1, s^2) / (s^1, s^2) \in \mathcal{S} \times \mathcal{S}, (s^1, s^2) \notin \mathcal{P} \text{ et } (s^1, s^2) \notin \mathcal{I}\}.$$

- **La structure de préordre total :** Cette structure est utilisée dans la plupart des modèles classiques d'économie. Dans ce cas, la relation de préférence  $\mathcal{P}$  est un ordre faible, et la relation d'indifférence  $\mathcal{I}$  est la non appartenance à la relation  $\mathcal{P}$ , c'est-à-dire

$$\mathcal{I} = \{(s^1, s^2) / ((s^1, s^2) \in \mathcal{S} \times \mathcal{S}, (s^1, s^2) \notin \mathcal{P} \text{ et } (s^1, s^2) \notin \mathcal{P})\}.$$

S'il n'y pas d'ex aequo ( $\mathcal{I} = \emptyset$ ),  $\mathcal{R}$  est un ordre strict.

- **La structure d'ordre partiel :** Cette structure est utilisée lorsqu'on range du meilleur au moins bon, sans ex aequo, les éléments de certains sous-ensembles d'alternatives. Cette structure autorise les situations d'incomparabilité entre deux alternatives.
- **La structure de préordre partiel :** Cette structure est utilisée lorsqu'on range du meilleur au moins bon, avec d'éventuels ex aequo, les éléments de certains sous ensembles d'alternatives. Cette structure autorise les situations d'incomparabilité et d'indifférence entre deux alternatives.

## 5.4 L'espace de décision et l'espace objectif

Dans cette section, nous introduisons informellement les notions fondamentales d'espace de décision (ou de variable) et de critère (ou d'objectif) dans lesquelles sont contenues les alternatives et leurs images sous les cartographies de fonctions objectives.

Considérons l'exemple suivant :

**Exemple 5.4.1.** Nous voulons acheter une nouvelle voiture et avons identifié quatre modèles que nous aimons : une VW Golf, une Opel Astra, une Ford Focus et une Toyota Corolla. La décision sera prise en fonction du prix, de la consommation d'essence et de la puissance. Nous préférons une voiture économique et puissante avec une faible consommation d'essence. Dans ce cas, nous sommes confrontés à un problème de décision avec quatre alternatives et trois critères. Les caractéristiques des quatre voitures sont montrées dans la table (5.1) (les données sont inventées).

Comment décide-t-on, laquelle des quatre voitures est la «meilleure» alternative, quand la voiture la plus puissante est aussi celle qui consomme le plus d'essence, de sorte que nous ne pouvons pas acheter une voiture à la fois économique et puissante. Cependant, nous observons qu'avec l'un des trois critères seuls, le choix est facile.

		Alternatives			
		VW	Opel	Ford	Toyota
Critères	Prix(1,000 Euros)	16.2	14.9	14.0	15.2
	Consommation ( $\frac{1}{100km}$ )	7.2	7.0	7.5	8.2
	Puissance (kW)	66.0	62.0	55.0	71.0

TABLE 5.1 – Critères et alternatives

Si nous ne prenons en considération que le prix et la consommation d'essence, il est facile de voir que Opel et Ford sont les choix efficaces. Pour les deux, il n'y a pas d'alternative qui est à la fois moins cher et consomme moins d'essence. De plus, Toyota et VW sont plus chers et consomment plus d'essence qu'Opel.

On note  $S = \{VW, Opel, Ford, Toyota\}$  l'ensemble réalisable, ou l'ensemble des alternatives du problème de décision. L'espace, dont l'ensemble réalisable  $S$  est un sous-ensemble s'appelé l'espace de décision.

Si on note le prix par  $f_1$  et la consommation d'essence par  $f_2$ , puis les tracés  $f_i : S \rightarrow \mathbb{R}$  sont des critères ou des fonctions objectives. Le problème d'optimisation peut être énoncé mathématiquement comme :

$$\min_{x \in S} (f_1(x), f_2(x))$$

L'image de  $S$  sous  $f = (f_1, f_2)$  est notée par :

$$Z := f(S) := \{z \in \mathbb{R}^2 : z = f(x), x \in S\},$$

est appelée l'image de l'ensemble réalisable, ou l'ensemble réalisable dans l'espace critère. L'espace à partir duquel les valeurs de critère sont prises est appelé l'espace critère.

## 5.5 L'optimisation multiobjectif

[35] La forme générale d'un problème d'optimisation multiobjectif est :

$$(P) : \begin{cases} (f_1(x), f_2(x), \dots, f_p(x)) \rightarrow \text{"maximiser"} \\ x \in S. \end{cases}$$

où on a  $p$  fonctions objectif  $f_k : \mathbb{R}^n \rightarrow \mathbb{R}$  avec  $k \in \{1, \dots, p\}$  et  $p \geq 2$ .

On note :

- Le vecteur des fonctions objectifs par  $f(x) = (f_1(x), f_2(x), \dots, f_p(x))^T \in \mathbb{R}^p$ .
- Le vecteur des variables de décision par

$$x = (x_1, x_2, \dots, x_n)^T,$$

appartenant à une région admissible non vide :

$$S = \{x \in \mathbb{R}^n / g(x) \leq 0_{\mathbb{R}^m}, h(x) = 0_{\mathbb{R}^r}\},$$

avec  $g(x) = (g_1(x), g_2(x), \dots, g_m(x))^T$  et  $h(x) = (h_1(x), h_2(x), \dots, h_r(x))^T$ .

Comme on peut le voir ici, on n'a plus un seul objectif à atteindre, mais  $p$  fonctions objectif. Le but que l'on se fixe dans la résolution d'un problème d'optimisation multiobjectif est de maximiser au mieux les différents objectifs sachant que dans un problème d'optimisation multicritère, on rencontre souvent des objectifs contradictoires (deux objectifs sont contradictoires lorsque la diminution d'un objectif entraîne une augmentation de l'autre objectif).

*Remarque 5.5.1* (La multiplicité des solutions). Lorsque l'on cherche à obtenir une solution optimale à un problème d'optimisation multiobjectif donné, on s'attend souvent à trouver une solution et une seule. En fait, on rencontre rarement ce cas de figure. La plupart du temps, on trouve une multitude de solutions, du fait que certains des objectifs sont contradictoires. Donc, quand on résoudra un problème d'optimisation multiobjectif, on obtiendra une grande quantité de solutions. Ces solutions ne seront pas optimales, au sens où elles ne maximiseront pas tous les objectifs du problème. Un concept intéressant, qui nous permettra de définir les solutions obtenues, est le compromis.

En effet, les solutions que l'on obtient lorsque l'on a résolu le problème sont des solutions de compromis. Elles maximisent un certain nombre d'objectifs tout en dégradant les performances sur d'autres objectifs.

**Définition 5.5.1.** • On note l'image de la région admissible par :

$$Z = f(S) = \{z \in \mathbb{R}^p / z = f(x), x \in S\} \subseteq \mathbb{R}^p.$$

- Les éléments de  $Z$  sont dits vecteurs critères et ils sont notés par  $z = (z_1, z_2, \dots, z_p)$ , où  $z_k = f_k(x)$ ,  $k \in \{1, \dots, p\}$ .

*Remarque 5.5.2.* On suppose que toutes les fonctions objectifs sont à maximiser. Si une fonction objectif  $f_k$  est à minimiser, il est équivalent à maximiser la fonction  $-f_k$ .

**Définition 5.5.2.** Lorsque toutes les fonctions objectifs et les contraintes qui définissent  $S$  sont linéaires, le problème d'optimisation multiobjectif est dit linéaire. On le note P.L.M.O (problème de programmation linéaire multi-objectifs).



**Définition 5.5.3.** Le vecteur  $z \in \mathbb{R}^p$  est dit résultat si et seulement si  $z = f(x), x \in S$ . Notons l'ensemble des résultats par :

$$Z = \{z \in \mathbb{R}^p, \exists x \in S, z = f(x)\}.$$

**Définition 5.5.4.** Étant donnés deux résultats  $z^1$  et  $z^2$ , on dit que  $z^1$  domine  $z^2$  si et seulement si :

$$z_k^1 \geq z_k^2, \forall k \in \{1, 2, \dots, p\} \text{ et } z_{k_0}^1 > z_{k_0}^2 \text{ pour un certain indice } k_0.$$

## 5.6 Les concepts d'optimalité

### 5.6.1 Optimalité de Pareto

**Définition 5.6.1.** Un vecteur de décision  $x^* \in S$  est une solution optimale de Pareto (efficace ou non dominée), s'il n'existe pas un autre vecteur  $x \in S$  tel que  $f_k(x) \geq f_k(x^*)$  pour tout  $k = 1, \dots, p$  et  $f_{k_0}(x) > f_{k_0}(x^*)$  pour au moins une fonction objectif  $f_{k_0}$ .

Un vecteur critère  $z^* \in Z$  est optimal de Pareto, s'il n'existe pas un autre vecteur critère  $z \in Z$  tel que  $z_k \geq z_k^*$  pour tout  $k \in \{1, \dots, p\}$  et  $z_{k_0} > z_{k_0}^*$  pour au moins une composante  $z_{k_0}$ . Ce qui revient à dire que  $z^*$  est optimal de Pareto si son vecteur de décision est optimal de Pareto.

**Exemple 5.6.1.** Soit le programme linéaire multiobjectif suivant :

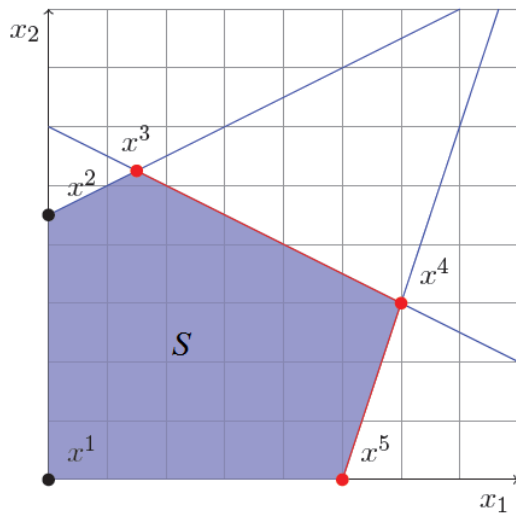
$$(MOLP1) : \begin{cases} \min f_1(x) = -x_1 - x_2, \\ \min f_2(x) = -2x_1 + 2x_2, \\ -x_1 + 2x_2 \leq 9, \\ 6x_1 - 2x_2 \leq 30, \\ x_1 + 2x_2 \leq 12, \\ x_i \geq 0, i = 1, 2. \end{cases}$$

L'ensemble réalisable de (MOLP1) et son image sont représentés sur les figures (a) et (b) :

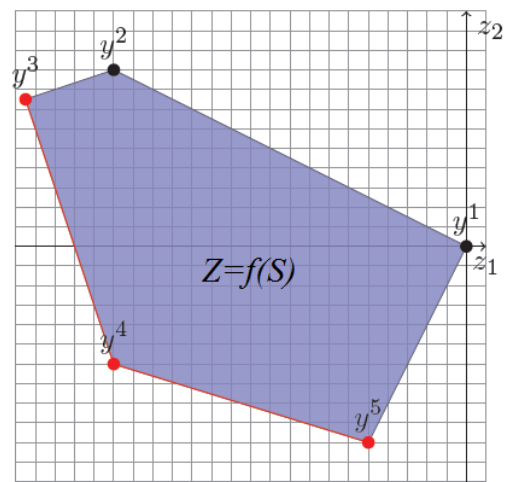
En particulier, le polytope  $S$  est délimité par cinq solutions extrêmes :

- $x^1 = (0, 0)$  dont l'image est  $y^1 = (0, 0)$ ,
- $x^2 = (0, 4.5)$  dont l'image est  $y^2 = (-18, 9)$ ,
- $x^3 = (1.5, 5.25)$  dont l'image est  $y^3 = (-22.5, 7.5)$ ,
- $x^4 = (6, 3)$  dont l'image est  $y^4 = (-18, -6)$ ,
- $x^5 = (5, 0)$  dont l'image est  $y^5 = (-5, -10)$ ,

Ici, une infinité de solutions sont efficaces : les segments  $[x^3x^4]$  et  $[x^4x^5]$ . On dit que ces solutions sont optimales de Pareto (efficaces ou non dominées).



(a) Ensemble réalisable de (MOPL1) et solutions efficaces.



(b) Image de l'ensemble réalisable de (MOPL1) et points non dominés.

FIGURE 5.1 – Ensemble réalisable du problème MOPL1 dans l'espace des décisions et l'espace des objectifs.

### 5.6.2 Optimalité au sens de Slater

**Définition 5.6.2.** Un vecteur de décision  $x^* \in S$  est une solution optimale au sens de Slater (ou optimale faible de Pareto), s'il n'existe pas un autre vecteur  $x \in S$  tel que  $f_k(x) > f_k(x^*)$  pour tout  $k \in \{1, \dots, p\}$ .

Un vecteur critère  $z^* \in Z$  est optimal de Slater (ou optimal faible de Pareto), s'il n'existe pas un autre vecteur critère  $z \in Z$  tel que  $z_k > z_k^*$  pour tout  $k \in \{1, \dots, p\}$ . Ce qui revient à dire que  $z^*$  est optimal de Slater si son vecteur de décision est optimal de Slater.

*Remarque 5.6.1.* Comme on peut le remarquer :

$$Z_{Pareto} \subseteq Z_{Slater}$$

*Remarque 5.6.2.* Bien que la programmation mono-objectif est étudiée dans l'espace de décision, la programmation multiobjectif est étudiée dans l'ensemble des critères. Une des raisons est que l'espace des critères est souvent de dimension plus petite que celle de l'espace de décision. Une autre raison est que les décideurs sont souvent intéressés par les valeurs des critères.

## 5.7 Définitions

### 5.7.1 La surface du compromis

Le petit nombre de solutions que l'on a sélectionné en utilisant la règle de classement basée sur la définition de la dominance forme ce que l'on appelle la surface de compromis (ou front de Pareto).

**Exemple 5.7.1.** [18] Dans le cas d'un problème à deux objectifs :

$$\begin{cases} \min f_1(x), \\ \min f_2(x), \\ x \in S. \end{cases}$$

On représente la surface du compromis noté **P** sur la figure suivante :

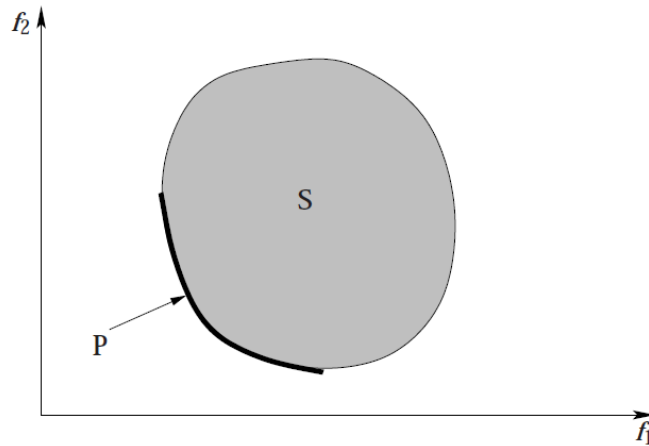


FIGURE 5.2 – Représentation de la surface de compromis.

### 5.7.2 Représentation de la surface du compromis sur Matlab pour un problème à deux fonctions objectif

Nous allons représenter la surface du compromis pour le programme linéaire bi-objectif suivant :

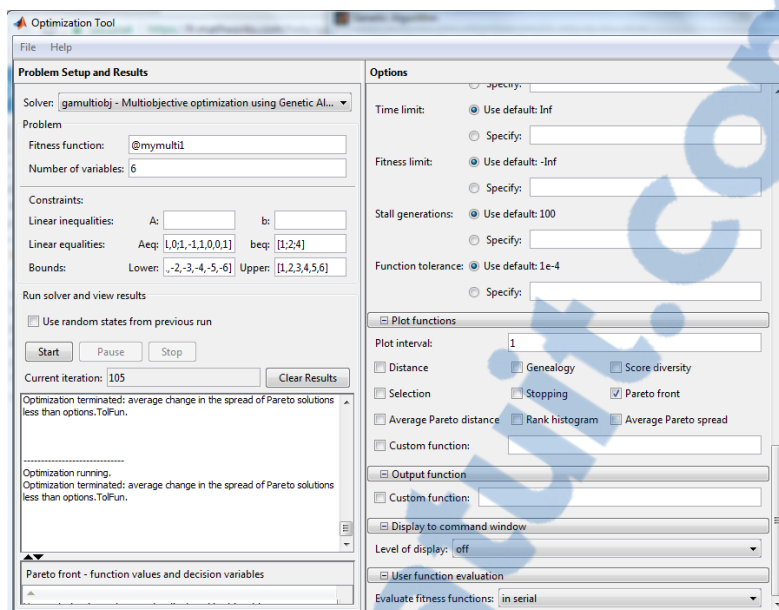
$$\left\{ \begin{array}{l} \min z_1(x) = x_1 + 2x_2, \\ \min z_2(x) = x_1 - 2x_3, \\ x_1 + x_2 + x_4 = 1, \\ x_2 + x_5 = 2, \\ x_1 - x_2 + x_3 + x_6 = 4, \\ -1 \leq x_1 \leq 1, \\ -2 \leq x_2 \leq 2, \\ -3 \leq x_3 \leq 3, \\ -4 \leq x_4 \leq 4, \\ -5 \leq x_5 \leq 5, \\ -6 \leq x_6 \leq 6. \end{array} \right. \quad (5.1)$$

Effectuons l'optimisation avec optimisation App en suivant les étapes suivante :

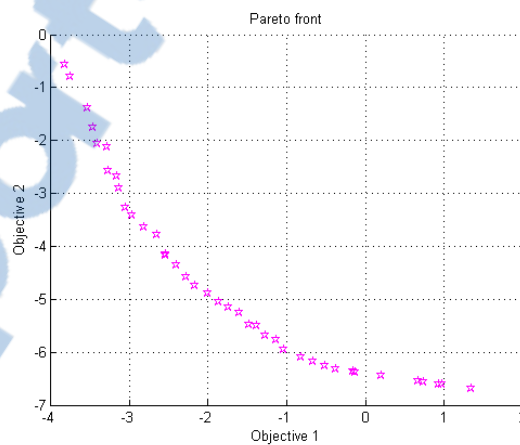
1. Il faut créer le fichier de fonction présenter ci-dessous avant de procéder et stocker en tant que mymultil.m :

```
function f = mymultil(x)
    f(1) = x(1)+2*x(2);
    f(2) = x(1)-2*x(3);
end
```

2. Pour définir le problème d'optimisation, il faut démarrez l'application de l'optimisation et le définir comme sur la figure ci-dessous :



3. Définir les options pour le problème.
4. Exécutez l'optimisation en cliquant sur Démarrer sous exécuter, afficher les résultats.
5. Un complot s'affiche dans une fenêtre de la figure.



### 5.7.3 Fonction utilité

Le décideur prend ses décisions en se basant sur une fonction. Cette fonction est appelée la fonction utilité.

**Définition 5.7.1.** Une fonction  $U : \mathbb{R}^n \rightarrow \mathbb{R}$  représentant les préférences du décideur parmi les vecteurs critères est appelée fonction utilité.

*Remarque 5.7.1.* Si on a l'expression mathématique de la fonction utilité, le problème d'optimisation multiobjectifs devient facile à résoudre en utilisant une méthode d'optimisation mono-objectif.

Cependant, cette méthode est inutilisable en pratique. La raison est qu'il est extrêmement difficile, ou impossible, pour le décideur de représenter mathématiquement par une fonction ses préférences. Et même si cette fonction est connue, elle peut être difficile à optimiser à cause de sa nature complexe.

**Théorème 5.7.1.** Soit la fonction utilité  $U : \mathbb{R}^n \rightarrow \mathbb{R}$  croissante monotone. Supposons que  $U$  atteint son maximum en  $z^0 \in Z$ . Alors,  $z^0$  est optimal de Pareto.

*Démonstration.* Soit  $z^0 \in Z$  une solution maximale d'une fonction utilité croissante monotone  $U$ . Supposons que  $z^0$  n'est pas optimal de Pareto. Alors, il existe un vecteur critère  $z \in Z$  tel que  $z_k \geq z_k^0$  pour tout  $k \in \{1, \dots, p\}$  et  $z_{k_0} > z_{k_0}^0$  pour au moins un indice  $k_0$ . Comme  $U$  est croissante monotone, on a  $U(z) > U(z^0)$ . Donc  $U$  n'atteint pas son maximum en  $z^0$ . Cette contradiction implique que  $z^0$  est optimal de Pareto.  $\square$

*Remarque 5.7.2.* Le théorème 5.7.1 donne une relation entre les solutions optimales de Pareto et les fonctions utilités.

#### 5.7.4 Point idéal

**Définition 5.7.2.** Les composantes  $z_k^0$  du vecteur critère idéal  $z^0 \in \mathbb{R}^p$  sont obtenues en maximisant chacune des fonctions objectifs individuellement, soumis aux contraintes, c'est-à-dire, en résolvant le programme linéaire mono-objectif suivant :

$$\begin{cases} \max f_k(x) \\ x \in S. \end{cases}$$

pour  $k \in \{1, \dots, p\}$ .

*Remarque 5.7.3.* Si le vecteur critère idéal est admissible c'est-à-dire  $z^0 \in Z$  alors il serait une solution du problème multiobjectif ( $P$ ). Cependant, en général, il n'en est pas ainsi à cause du caractère conflictuel existant entre les objectifs.

#### 5.7.5 Table des gains

Une table des gains est formée en utilisant les vecteurs de décision obtenus lors du calcul du vecteur critère idéal.

	$f_1(x)$	$f_2(x)$	$\dots$	$f_p(x)$
$x^{(1)}$	$f_1(x^{(1)})$	$f_2(x^{(1)})$	$\dots$	$f_p(x^{(1)})$
$x^{(2)}$	$f_1(x^{(2)})$	$f_2(x^{(2)})$	$\dots$	$f_p(x^{(2)})$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$x^{(p)}$	$f_1(x^{(p)})$	$f_2(x^{(p)})$	$\dots$	$f_p(x^{(p)})$

TABLE 5.2 – Table des gains

$$\left\{ \begin{array}{l} x^{(1)} \text{ correspond au } \left\{ \begin{array}{l} \max f_1(x) \\ x \in S. \end{array} \right. \\ x^{(2)} \text{ correspond au } \left\{ \begin{array}{l} \max f_2(x) \\ x \in S. \end{array} \right. \\ \vdots \\ x^{(k)} \text{ correspond au } \left\{ \begin{array}{l} \max f_p(x) \\ x \in S. \end{array} \right. \end{array} \right.$$

*Remarque 5.7.4.* On peut voir que les composantes du point idéal sont la diagonale de la table 5.2 c'est-à-dire :

$$z^0 = \begin{pmatrix} f_1(x^{(1)}) \\ f_2(x^{(2)}) \\ \vdots \\ f_p(x^{(p)}) \end{pmatrix}$$

### 5.7.6 Point nadir

Les bornes inférieures de l'ensemble optimal de Pareto correspondent aux composantes du point appelé nadir, et noté  $z^{nad}$ . On peut les estimer à partir de la table des gains.

**Définition 5.7.3.** Le point critère nadir est le vecteur dont chaque composante  $z_k^{nad}$  correspond à la valeur minimale de la  $k^{\text{ème}}$  colonne de la table des gains. En d'autres termes :

$$\left\{ \begin{array}{l} \min(f_1(x^{(1)}), f_1(x^{(2)}), \dots, f_1(x^{(p)})) = z_1^{nad}, \\ \min(f_2(x^{(1)}), f_2(x^{(2)}), \dots, f_2(x^{(p)})) = z_2^{nad}, \\ \vdots \\ \min(f_p(x^{(1)}), f_p(x^{(2)}), \dots, f_p(x^{(p)})) = z_p^{nad}. \end{array} \right.$$

**Exemple 5.7.2** (Représentation du Point idéal et du point nadir). [18] Soit le problème multiobjectif suivant :

$$\begin{cases} \min f_1(x), \\ \min f_2(x), \\ x \in S. \end{cases}$$

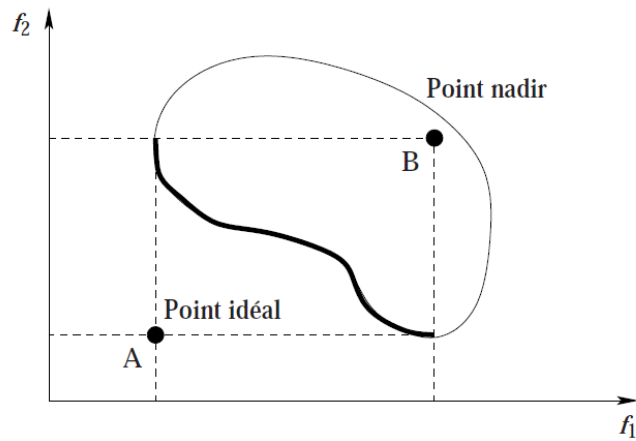


FIGURE 5.3 – Représentation du point idéal et du point “nadir”.



# Chapitre 6

## Quelques méthodes d'optimisation multiobjectif

Dans ce chapitre, nous allons exposer quelques méthodes utilisées dans la résolution d'un programme multiobjectif. Nous insisterons principalement sur la méthode du simplexe multiobjectif en déterminant ses assises théoriques.

### 6.1 La méthode de pondération

Elle consiste à transformer un problème multiobjectif en un problème à un objectif en agrégeant les différents critères sous la forme d'une somme pondérée :

$$(P_\omega) : \begin{cases} f_{eq}(x) = \sum_{k=1}^p \omega_k f_k(x) \rightarrow \text{maximiser} \\ x \in S. \end{cases}$$

Les  $\omega_k$  appelés poids, vérifient :

$$\begin{cases} \sum_{k=1}^p \omega_k = 1, \\ \omega_k \geq 0. \end{cases}$$

Il est clair que la résolution d'un problème pour un vecteur poids  $\omega$  fixé ne permet de calculer que quelques solutions Pareto optimales. Pour obtenir un ensemble contenant un grand nombre de solutions Pareto optimales, il faut résoudre plusieurs fois le problème en changeant à chaque fois la valeur de  $\omega$ .

*Remarque 6.1.1.*

- Cette approche a l'avantage de pouvoir réutiliser tous les algorithmes classiques dédiés aux problèmes d'optimisation mono-objectif.
- Cependant cette approche a aussi deux inconvénients importants. Le premier est du au fait que pour avoir un ensemble de points bien répartis sur le front Pareto,

les différents vecteurs  $\omega$  doivent être choisis judicieusement. Il est donc nécessaire d'avoir une bonne connaissance du problème. Le deuxième inconvénient provient du fait que cette méthode ne permet pas, de calculer intégralement la surface de compromis lorsque celle-ci n'est pas convexe.

**Théorème 6.1.1.** *Supposons que  $x^*$  soit une solution optimale pour le problème pondéré  $(P_\omega)$ . On a alors :*

1. *Si  $\omega_k \geq 0, k = 1, \dots, p$  alors  $x^*$  est une solution faiblement efficace pour  $(P)$  ;*
2. *Si  $\omega_k > 0, k = 1, \dots, p$  alors  $x^*$  est une solution efficace pour  $(P)$  ;*
3. *Si  $\omega_k \geq 0, k = 1, \dots, p$  et  $x^*$  est l'unique solution optimale du problème pondéré  $(P_\omega)$  alors  $x^*$  est une solution efficace pour  $(P)$  ;*

*Démonstration.* 1. Soit  $x^*$  une solution optimale pour le problème pondéré  $(P_\omega)$ . Supposons qu'elle n'est pas faiblement efficace. Dans ce cas, il existe une solution  $x \in S$  telle que  $f_k(x) < f_k(x^*), \forall k \in \{1, 2, \dots, p\}$ . Ayant  $\omega_{k_0} > 0$  pour au moins un  $k$ . D'où :

$$\sum_{k=1}^p \omega_k f_k(x) < \sum_{k=1}^p \omega_k f_k(x^*)$$

Ceci contredit le fait que  $x^*$  soit solution du problème  $P_\omega$ . Donc  $x^*$  est une solution faiblement efficace pour  $(P)$ .

2. Soit  $x^*$  une solution optimale pour le problème pondéré  $(P_\omega)$ . Supposons qu'elle n'est pas efficace. Dans ce cas, il existe une solution  $x \in S$  telle que  $f_k(x) \leq f_k(x^*), \forall k \in \{1, 2, \dots, p\}$  et  $f_{k_0}(x) < f_{k_0}(x^*)$  pour au moins une composante  $f_{k_0}$ . Comme  $\omega_k > 0, \forall k \in \{1, 2, \dots, p\}$  nous avons :

$$\sum_{k=1}^p \omega_k f_k(x) < \sum_{k=1}^p \omega_k f_k(x^*)$$

Ceci contredit le fait que  $x^*$  soit solution du problème  $(P_\omega)$ . Donc  $x^*$  est une solution faiblement efficace pour  $(P)$ .

3. Soit  $x^*$  une solution optimale pour le problème pondéré  $(P_\omega)$ . Supposons qu'elle n'est pas efficace. Dans ce cas, il existe une solution  $x \in S$  telle que  $f_k(x) \leq f_k(x^*), \forall k \in \{1, 2, \dots, p\}$  et  $f_k(x) < f_k(x^*)$  pour au moins une composante  $f_k$ . Puisque les  $\omega_k \geq 0, \forall k \in \{1, 2, \dots, p\}$ , on a :

$$\begin{aligned} \sum_{k=1}^p \omega_k f_k(x) &\leq \sum_{k=1}^p \omega_k f_k(x^*) \\ \Rightarrow f_{eq}(x) &\leq f_{eq}(x^*) \end{aligned} \tag{6.1}$$

D'autre part, l'unicité de  $x^*$  implique que :

$$f_{eq}(x^*) < f_{eq}(x^\circ), \forall x^\circ \in S \quad (6.2)$$

Ces deux dernières inégalités 6.1 et 6.2 sont contradictoires. Ainsi  $x^*$  est efficace pour le problème  $(P)$ . □

## 6.2 La méthode du compromis

Après avoir étudié les méthodes qui permettent de fusionner les fonctions objectif en une seule (on parle aussi d'“agrégation” des fonctions objectif), nous allons présenter des méthodes qui permettent de transformer un problème d'optimisation multiobjectif en un problème d'optimisation monobjectif comportant quelques contraintes supplémentaires.

La démarche est la suivante :

- on choisit un objectif à optimiser prioritairement ;
- on choisit un vecteur de contraintes initial ;
- on transforme le problème en conservant l'objectif prioritaire et en transformant les autres objectifs en contraintes d'inégalité.

On appelle aussi cette méthode la méthode de la contrainte  $\epsilon$ .

Dans le problème  $(P)$ , on suppose que la fonction objectif prioritaire est la fonction de rang 1. Puis on choisit un vecteur de contraintes  $\epsilon_k, k \in \{2, \dots, p\}, \epsilon_k \geq 0$ .

On transforme alors le problème  $(P)$  de la manière suivante :

$$\left\{ \begin{array}{l} \max f_1(x), \\ f_2(x) \leq \epsilon_2, \\ \vdots \\ f_p(x) \leq \epsilon_p, \\ x \in S. \end{array} \right.$$

Les principaux inconvénients de cette méthode sont les suivants : elle est gourmande en temps de calcul et la programmation de l'algorithme peut être extrêmement difficile s'il y a trop de fonctions contraintes.

Cependant, la relative simplicité de l'énoncé de la méthode l'a rendue populaire.

**Exemple 6.2.1.** Soit le problème multiobjectif suivant :

$$\left\{ \begin{array}{l} \min f_1(x) = x^2, \\ \min f_2(x) = (x - 2)^2, \\ x \in [0, 2]. \end{array} \right.$$

En appliquant la méthode du compromis, on obtient le problème suivant :

$$\begin{cases} \min f_1(x) = x^2, \\ f_2(x) = (x - 2)^2 \leq \epsilon, \\ x \in [0, 2]. \end{cases}$$

Pour résoudre ce problème, on donne différentes valeurs à la variable  $\epsilon$  pour avoir la surface de compromis.

### 6.3 Les méthodes hybrides

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

La méthode hybride la plus connue est la méthode de Corley. Cette méthode utilise la méthode de pondération des fonctions objectif et la méthode du compromis.

On transforme le problème ( $P$ ) de la manière suivante :

$$\begin{cases} \max \sum_{k=1}^p w_k f_k(x), \\ f_k(x) \leq \epsilon_k, k = \overline{1, p}, \\ x \in S. \end{cases}$$

Cette méthode permet de combiner les avantages des deux méthodes citées précédemment (la méthode de pondération des fonctions objectif et la méthode du compromis). Elle est ainsi efficace sur différents types de problèmes, qu'ils soient convexes ou non convexes. Cependant, une difficulté survient : le nombre de paramètres à déterminer a été multiplié par deux. Il sera beaucoup plus difficile à l'utilisateur d'exprimer sa préférence en jouant sur l'ensemble des paramètres.

### 6.4 La méthode dite du "but à atteindre"

Contrairement à la méthode de pondération des fonctions objectif, cette méthode n'impose pas de travailler sur un domaine réalisable convexe. La démarche suivie est la suivante :

- on choisit un vecteur de fonctions objectif initial  $F$  ;
- on choisit une direction de recherche (on fournit, en quelque sorte, des coefficients de pondération, comme pour la méthode de pondération des fonctions objectif)  $w$  ;
- on cherche ensuite à minimiser un coefficient scalaire  $\lambda$  qui représente l'écart par rapport à l'objectif initial  $f$  que l'on s'était fixé.

On part du problème ( $P$ ), on choisit un vecteur de fonctions objectif initial  $F \in \mathbb{R}^p$ , un ensemble de coefficients  $w_k, k \in \{1, \dots, p\}$ , et l'on désigne par  $\lambda$  un nombre

scalaire que l'on cherchera à minimiser.

Le problème ( $P$ ) devient alors :

$$\begin{cases} \max \lambda, \\ f_1(x) - w_1\lambda \leq F_1, \\ \vdots \\ f_p(x) - w_p\lambda \leq F_p, \\ x \in S. \end{cases}$$

Le principal avantage de cette méthode est qu'elle permet de travailler avec un ensemble de solutions pas nécessairement convexe. Cependant, il peut exister des formes de domaines pour lesquelles la solution dépend du choix du but initial.

## 6.5 La méthode dite du “but programmé”

Cette méthode (se rapproche beaucoup de la méthode du but à atteindre. La principale différence est qu'après avoir transformé la forme du problème d'optimisation, on se retrouve avec des contraintes d'égalité et non plus des contraintes d'inégalité.

La démarche est la suivante :

- on choisit un vecteur de fonctions objectif initial  $F$  ;
- à chaque objectif, on associe deux nouvelles variables que l'on appelle les “déviations” par rapport au vecteur de fonctions objectif initial fixé ;
- on minimise ensuite une des deux variables. Le choix de la variable se fait en fonction du type de dépassement que l'on veut (au-dessus ou au-dessous de l'objectif que l'on s'est fixé).

On part du problème ( $P$ ), on choisit un vecteur de fonctions objectif initial  $F \in \mathbb{R}^p$ . On associe aussi un ensemble de variables  $d_k^+$  et  $d_k^-$  à chaque fonction objectif  $f_k(x)$ ,  $k \in \{1, \dots, p\}$  (les “déviations”).

On obtient alors le problème suivant :

$$\begin{cases} \max \lambda, \\ f_1(x) = F_1 + d_1^+ - d_1^-, \\ \vdots \\ f_p(x) = F_p + d_p^+ - d_p^-, \\ x \in S. \end{cases}$$

*Remarque 6.5.1.* La méthode du but programmé semblable à celle du but à atteindre, soulève les mêmes critiques. Dans certains cas, on peut “rater” des solutions cachées dans des concavités.

## 6.6 La méthode du Simplexe Multiobjectif

Dans cette partie, nous allons exposer la méthode du simplexe multiobjectif qui est basée sur la méthode du simplexe pour résoudre un programme linéaire multiobjectif (PLMO), cette dernière a la particularité de pouvoir manipuler les objectifs comme ils se présentent sans aucune scalarisation<sup>1</sup> pour trouver les points extrêmes optimaux de Pareto.

Considérons le problème de programmation linéaire dont la forme standard est la suivante :

$$\begin{cases} \max Cx, \\ x \in S. \end{cases} \quad (6.3)$$

où  $S$  est l'ensemble des solutions réalisables défini comme suit :

$$S = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}. \quad (6.4)$$

On définit les indices des contraintes et des variables de décision par :  $I = \{1, 2, \dots, m\}$ ,  $J = \{1, 2, \dots, n\}$ , tel que :

$J = J_B \cup J_N$  avec  $J_B \cap J_N = \emptyset$ ,  $|J_B| = m$ .

Donc,  $x = x(J) = (x_j, j \in J)$  est un vecteur d'ordre  $n$ ,  $b = (b_i, i \in I)$  est un vecteur d'ordre  $m$ ,  $A = A(I, J)$  est une matrice d'ordre  $m \times n$ , tel que  $\text{rang} A = m < n$ .

$J_B$  étant l'ensemble de base (ou ensemble basique) défini par  $J_B = \{j_1, \dots, j_m\}$  où  $j$  dénote l'indice des variables, tel que les colonnes  $(A_j)_{j \in J_B}$  forment une base, les variables  $(x_i)_{i \in B}$  sont les variables de base.

De même  $J_N$  étant l'ensemble des indices hors base défini par  $J_N = J \setminus J_B$  et les variables  $(x_i)_{i \in N}$  sont les variables hors base (ou non basique).

On définit la fonction multicritère comme suit :

$$Z(x) = \begin{pmatrix} z_1(x) \\ z_2(x) \\ \vdots \\ z_p(x) \end{pmatrix} = \begin{pmatrix} c_1^T x \\ c_2^T x \\ \vdots \\ c_p^T x \end{pmatrix} = Cx, \quad (6.5)$$

où  $C = C(K, J)$  avec  $K = \{1, 2, \dots, p\}$  est une matrice d'ordre  $p \times n$ , dont les lignes sont les  $n$ -vecteurs  $c_k^T, k \in \{1, 2, \dots, p\}$ .

On peut alors fractionner les vecteurs et les matrices de la manière suivante :

$$x = \begin{pmatrix} x_B \\ \text{---} \\ x_N \end{pmatrix}, \quad x_B = x(J_B) = (x_j, j \in J_B), \quad x_N = x(J_N) = (x_j, j \in J_N),$$

---

1. La scalarisation est le fait de ramener un problème multiobjectif à un problème mono-objectif.

$$C = \begin{pmatrix} C_B \\ \text{---} \\ C_N \end{pmatrix}, \quad C_B = C(K, J_B), \quad C_N = C(K, J_N),$$

$$A = A(I, J) = (a_{ij}, \quad 1 \leq i \leq m, \quad 1 \leq j \leq n),$$

$$A = (A_B | A_N), \quad A_B = A(I, J_B), \quad A_N = A(I, J_N).$$

Le problème consiste à trouver l'ensemble de toutes les solutions efficaces c'est-à-dire les solutions optimales de Pareto.

Pour  $\lambda \in \mathbb{R}_+^p$ , on note  $PL(\lambda)$  le programme linéaire somme pondérée suivant :

$$\max\{\lambda^T Cx : Ax = b, x \geq 0\}.$$

- La solution  $x$  vérifiant :  $x_N = 0$  et  $x_B = A_B^{-1}b$ , est dite solution basique associée à la base  $B$ .  
Si  $x$  est une solution de base satisfaisant toutes les contraintes du problème alors  $x$  est une solution de base réalisable.
- On définit la matrice des estimations par  $W = C_B A_B^{-1} A - C$  d'ordre  $p \times n$  avec  $W = (W_B, W_N)$  correspond donc au fractionnement du plan basique.

*Remarque 6.6.1.* Géométriquement, une solution de base correspond à un sommet du polyèdre défini par  $Ax = b$ .

**Théorème 6.6.1.** *La solution réalisable  $x^0 \in S$  est dite efficace dans le problème (6.3), s'il n'existe aucune autre solution réalisable  $\bar{x} \in S$  telle que*

$$C\bar{x} \geq Cx^0 \text{ et } C\bar{x} \neq Cx^0.$$

On note l'ensemble des solutions efficaces par  $S^E$ .

**Théorème 6.6.2.** *La solution réalisable  $x^0 \in S$  est dite faiblement efficace dans le problème (6.3), s'il n'existe aucune autre solution réalisable  $\bar{x} \in S$  telle que*

$$C\bar{x} > Cx^0.$$

**Théorème 6.6.3.** *La solution réalisable  $x^0 \in S$  est efficace pour le problème (6.3) si et seulement si  $\exists \lambda \in \mathbb{R}^p, \lambda > 0$  tel que*

$$\begin{cases} \lambda^T Cx^0 = \max \lambda^T Cx \\ x \in S. \end{cases} \quad (6.6)$$

**Définition 6.6.1.** Une solution réalisable basique  $x^0$  est dite non dégénérée si :

$$x_j \geq 0, \forall j \in J_B.$$

**Propriété 6.6.1.** Soit  $\lambda \in \mathbb{R}^p$  et  $\lambda > 0$ . Pour que  $\lambda \in \Lambda(x^0)$ , il est nécessaire et suffisant que

$$\begin{cases} \lambda^T Cx^0 = \max \lambda^T Cx, \\ x \in S. \end{cases} \quad (6.7)$$

La démonstration de cette propriété provient directement de la définition du cône  $\Lambda(x^0)$  et du théorème 6.6.3.

*Remarque 6.6.2.* Le cône  $\Lambda(x^0)$  est appelé cône des poids optimaux du plan basique efficace  $x^0$ .

**Propriété 6.6.2.** Dans le cas où  $\text{rang } C = n$  le plan basique  $x^0$  est une solution efficace unique dans  $S$ , si et seulement si  $\Lambda(x^0) = \mathbb{R}_+^p$ , avec  $\mathbb{R}_+^p$  le quadrant positif de  $\mathbb{R}^p$ .

**Propriété 6.6.3.** Soit  $S^{EB} = \{x^1, x^2, \dots, x^r\}$  l'ensemble des solutions réalisables basiques efficaces du problème (6.3). Alors

$$\bigcup_{k=1}^r \Lambda(x^k) = \mathbb{R}_+^p,$$

où  $\Lambda(x^k)$  est le cône des poids optimaux du plan basique efficace  $x^k$ .

*Démonstration.*

L'inclusion  $\bigcup_{k=1}^r \Lambda(x^k) \subset \mathbb{R}_+^p$  est évidente.

Démontrons l'inclusion inverse c'est à dire que  $\mathbb{R}_+^p \subset \bigcup_{k=1}^r \Lambda(x^k)$ . Soit  $\lambda \in \mathbb{R}_+^p$ . Considérons le problème suivant :

$$\widehat{\lambda}^T Cx \rightarrow \max, x \in S. \quad (6.8)$$

Avec  $S$  un ensemble borné et fermé, alors la solution de (6.8) existe. Le problème (6.8) possède alors au moins une solution réalisable basique optimale  $\hat{x}$ .

Du théorème 6.6.3, on conclut que  $\hat{x} \in S^{EB}$ . En vertu de la propriété 6.6.1, on a  $\widehat{\lambda} \in \Lambda(\hat{x})$ . Donc  $\mathbb{R}_+^p \subset \bigcup_{k=1}^r \Lambda(x^k)$ .  $\square$

Dans le cas de la programmation linéaire multiobjectif, il existe de nombreuses méthodes de résolution qui utilisent toutes l'algorithme du simplexe, modifié pour la circonstance.

Les points extrêmes de l'ensemble des solutions réalisables de notre problème correspondent aux solutions de base réalisables si celles-ci ne sont pas dégénérées. Cela permet l'utilisation du tableau du simplexe pour déterminer les points extrêmes correspondant aux solutions non-dominées. On va donc énumérer les solutions de base grâce au simplexe modifié, tester si ces solutions correspondent à des solutions efficaces (non-dominées), puis générer le polytope engendré par les solutions efficaces.

Ainsi, la méthode de résolution comporte trois phases :



### 6.6.1 Trouver une solution de base réalisable :

Cette phase est la phase 1 de la programmation linéaire mono-objectif. Elle consiste à rechercher une solution de base réalisable initiale  $x^0$  en résolvant le système des contraintes du programme.

### 6.6.2 Trouver un point extrême efficace :

En s'inspirant de la méthode de H. Isermann, la procédure proposée par H.P. Benson [7] pour la recherche d'un point initial extrême efficace consiste à résoudre deux problèmes de la programmation linéaire. Notons que si  $S = \emptyset$ , alors le problème (6.3) ne possède aucune solution efficace, donc on suppose que  $S \neq \emptyset$ . Soit le programme linéaire suivant :

$$\begin{cases} \min -z^T C x^0 + u^T b, \\ w^T + z^T C - u^T A = -e^T C, \\ w, z \geq 0, \end{cases} \quad (6.9)$$

où  $x^0 \in S$ ,  $w \in \mathbb{R}^n$  et  $z, u \in \mathbb{R}^p$

La méthode de H.P. Benson est alors donnée par les étapes suivantes :

Étape (1) : Trouver un point réalisable  $x^0 \in S$ .

Étape (2) : S'il en existe, trouver une solution optimale  $(w^0, z^0, u^0)$  de (6.9), et aller à l'étape (3), sinon arrêter.

Étape (3) : Obtenir un point extrême en résolvant le programme suivant :

$$\begin{cases} \max (z^0 + e)^T C x, \\ x \in S. \end{cases}$$

Pour voir la preuve de la validité de ces deux procédures, veuillez consulter [69].

### 6.6.3 Générer l'ensemble des points extrêmes efficaces :

Dans cette phase, on localise tous les points extrêmes efficaces en effectuant un pivotage efficace à partir de chaque variable non basique efficace en calculant à partir du premier point extrême efficace un point voisin et on teste s'il est efficace en utilisant le test d'efficacité de Yu et Zeleny [105, 106] défini par le théorème suivant :

**Théorème 6.6.4.** *Soit le programme linéaire :*

$$\begin{cases} \max g = e^T v, \\ Cx - v = Cx^*, \\ x \in S, \\ v \geq 0. \end{cases} \quad (6.10)$$

Si  $\max g = 0$ , alors  $x^*$  est efficace. Sinon,  $x^*$  est non efficace.

*Démonstration.* Par définition,  $x^*$  est efficace si et seulement s'il n'existe pas un point  $x \in S$  tel que  $Cx \geq Cx^*$  avec au moins une inégalité stricte. Donc, si  $\max g > 0$ , alors une des composantes de  $v$  n'est pas nulle et  $x^*$  est non efficace; au contraire, si  $\max g = 0$ , alors toutes les composantes de  $v$  sont nulles et  $x^*$  est efficace.  $\square$

### Méthode de Yu et Zeleny [106]

Yu et Zeleny calculent à partir d'un point efficace un point extrême, puis ils testent si celui-ci est efficace. S'il ne l'est pas, ils reviennent au point efficace, puis le procédé est itéré. Cependant, Yu et Zeleny ne testent pas toutes les bases car certaines sont efficaces ou dominées de manière évidente.

Soient  $J_B$  l'ensemble des indices des variables de base et  $J_N$  l'ensemble des indices des variables hors base.

Après réarrangement des colonnes de la matrice  $A$ , le système  $Ax = b$  s'écrit

$$(A_B, A_N) \begin{pmatrix} x_B \\ x_N \end{pmatrix} = b,$$

où  $A_B = A(I, J_B)$ ,  $A_N = A(I, J_N)$ ,  $x_B = x(J_B)$  et  $x_N = x(J_N)$ .

D'après les formules habituelles de résolution d'un programme linéaire, on a

$$x_B = A_B^{-1}b - A_B^{-1}A_Nx_N = A_B^{-1}b,$$

car  $x_N = 0$ .

On pose

$$Y = (y_{ij}, i \in J_B, j \in J_N) = A_B^{-1}A_N.$$

La valeur de la fonction objectif pour la solution de base réalisable  $x = (b, 0)^T$ ,  $b \in \mathbb{R}^m$ ,  $0 \in \mathbb{R}^{n-m}$ , est

$$z_k = \sum_{r \in J_B} C_{kr}b_r, \quad k \in \{1, \dots, p\}.$$

On a

$$W_{kj} = C_{kj} - \sum_{r \in J_B} C_{kr}y_{rj}, \quad j \in J_N, k \in \{1, \dots, p\}$$

et on pose

$$\theta_j = \min \left\{ \frac{b_r}{y_{rj}} : y_{rj} > 0; r \in J_B \right\}, \quad \forall j \in J_N.$$

Si  $W_{kj} \leq 0$ ,  $\forall j \in J_N$ , alors la solution courante ( $x = (b, 0)^T$ ) est maximale pour le critère  $k$ .

Si on introduit la  $j^{\text{ème}}$  colonne dans la base, on va obtenir une nouvelle solution  $\hat{x}$  et un nouveau vecteur  $\hat{Z}$ , à l'aide de la relation suivante :

$$\hat{Z} = Z + \theta_j W_j.$$

Le test d'efficacité présenté ci-haut n'est pas toujours nécessaire, car certaines solutions sont efficaces ou non efficaces d'une manière évidente ; et ce, en se basant sur les remarques suivantes :

**Remarque 1 :** Soit  $x$  une solution de base réalisable.

- S'il existe un  $j \in J_N$  tel que  $W_j \geq 0$  avec au moins une inégalité stricte, alors la solution  $x$  est dominée par la solution  $\hat{x}$ . En effet, si  $\theta_j W_j > 0$  alors,  $\hat{Z} = Z + \theta_j W_j \geq Z$ , avec au moins une des inégalités qui est stricte.
- S'il existe un  $j \in J_N$  tel que  $W_j \leq 0$  avec au moins une inégalité stricte, alors l'introduction de la  $j^{\text{ème}}$  colonne en base mène à une solution  $\hat{x}$  dominée par la solution  $x$ . En effet, comme  $\theta_j W_j < 0$  alors,  $\hat{Z} = Z + \theta_j W_j \leq Z$ , avec au moins une des inégalités qui est stricte.

**Remarque 2 :** Soit  $x$  une solution de base réalisable, et  $Z$  la valeur qui lui correspond. S'il existe  $j, l \in J_N$ , tel que  $\theta_j W_j \leq \theta_l W_l$  avec au moins une des inégalités qui est stricte, alors l'introduction de la  $j^{\text{ème}}$  colonne en base mène à une solution dominée par celle qui résulte de l'introduction de la  $l^{\text{ème}}$  colonne, donc il est inutile d'effectuer le changement de base introduisant la colonne  $j$ . Cette propriété découle de la relation suivante :

$$\theta_j W_j \leq \theta_l W_l \Rightarrow Z + \theta_j W_j \leq Z + \theta_l W_l.$$

**Remarque 3 :** S'il existe un indice  $k \in \{1, \dots, p\}$  tel que  $W_{kj} \leq 0, \forall j \in J_N$ , alors le  $i^{\text{ème}}$  critère est à son maximum. Si, de plus, il n'existe pas de  $j \in J_N$  tel que  $W_j = 0$ , alors la solution basique correspondante est non dominée.

En définitif, seules les colonnes non comparables à zéro, et les colonnes, telles que  $\theta_{j_1} W_{j_1}$  et  $\theta_{j_2} W_{j_2}$  ne sont pas comparables,  $\forall j_1, j_2 \in J_N$ , sont candidates pour une introduction en base. Dans ce cas, on ne peut rien dire sur l'efficacité de la solution  $\bar{x}$  correspondante. Pour cela, on applique le test d'efficacité énoncé ci-haut par ces auteurs.

En tenant compte de ces remarques, Yu et Zeleny ont construit une procédure, qui permet d'obtenir tous les sommets efficaces sans en oublier un.

Dans la description suivante de l'algorithme du Simplex multiobjectif, qui trouve toutes les bases efficaces et toutes les solutions réalisables basiques efficaces, nous devons stocker une liste  $L1$  de bases efficaces à traiter et une liste de bases efficaces  $L2$  pour la sortie, ainsi qu'une liste  $EN$  de variables non-basiques efficaces.

**Algorithme 6.6.1.** Entrée : données  $A, b, C$  d'un P.L.M.O.

1. Trouver une solution réalisable.
  - Si elle existe, alors aller en 2.
  - Sinon Arrêter, le problème est irréalisable.
2. Poser  $s = 1$ .
3. Calculer  $x^s$ , le  $s^{\text{ème}}$  point extrême efficace.
4. Vérifier si  $W_i^s \geq 0$  pour au moins un  $i$ .
  - Si oui, aller en 3.
  - Sinon, aller en 6.
5. Peut-on améliorer un objectif ?
  - Si oui, aller en 6.
  - Sinon, aller en 11.
6. Y a-t-il un  $j \in J_N$  tel que  $W_j^s \leq 0$  ?
  - Si oui, aller en 7.
  - Sinon, aller en 9.
7. Considérer tous les  $j \in J_N$ .
8. Vérifier si l'introduction de la  $j^{\text{ème}}$  colonne correspondante conduit à une base inexplorée ?
  - Si oui, poser  $s = s + 1$  et aller en 3.
  - Sinon, aller en 9.
9. Y a-t-il une base inexplorée déjà stockée ?
  - Si oui, aller en 3.
  - Sinon, arrêter, tous les sommets ont été calculés.
10. Considérer le programme (6.6.4).
  - Si  $\max g = 0$ , aller en 11.
  - Sinon, aller en 12.
11. La solution  $x^s$  est efficace, aller en 12.
12. Calculer  $\theta_j = \min (x_i^s / a_{ij}, a_{ij} > 0), \forall j \in J_N^s$ .
13. Y a-t-il  $j \in J_N^s$  tel que  $\theta_j W_j^s \geq \theta_k W_k^s, \forall k \in J_N^s$  ?
  - Si oui, aller en 8.
  - Sinon, aller en 14.
14. Y a-t-il  $j \in J_B^s$  tel que  $W_j^s \geq 0$  ?
  - Si oui, aller en 15.
  - Sinon, aller en 9.
15. Vérifier si  $s \leq n - m$ .
  - Si oui, aller en 16.
  - Sinon, aller en 9.
16. Stocker les indices de base correspondants qui conduisent à une base inexplorée, et aller en 9.

### 6.6.4 Exemple numérique

[34] Soit le programme linéaire multiobjectif suivant :

$$\left\{ \begin{array}{l} \max f_1(x) = x_1 + 2x_2, \\ \max f_2(x) = x_1 - 2x_3, \\ \max f_3(x) = -x_1 + x_3, \\ x_1 + x_2 \leq 1, \\ x_2 \leq 2, \\ x_1 - x_2 + x_3 \leq 4, \\ x_1 \geq 0, \\ x_2 \geq 0, \\ x_3 \geq 0. \end{array} \right. \quad (6.11)$$

On introduit les variables d'écart  $x_4$ ,  $x_5$  et  $x_6$  pour avoir la forme standard d'un P.L.M.O, on obtient :

$$\left\{ \begin{array}{l} \max f_1(x) = x_1 + 2x_2, \\ \max f_2(x) = x_1 - 2x_3, \\ \max f_3(x) = -x_1 + x_3, \\ x_1 + x_2 + x_4 = 1, \\ x_2 + x_5 = 2, \\ x_1 - x_2 + x_3 + x_6 = 4, \\ x_i \geq 0, i = \overline{1, 6}. \end{array} \right.$$

Par identification, on a :

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix}, A = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & -1 & 1 & 0 & 0 & 1 \end{pmatrix}, b = \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix}, C = \begin{pmatrix} 1 & 2 & 0 & 0 & 0 & 0 \\ 1 & 0 & -2 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

**La phase 1 :**

$J_B = \{4, 5, 6\}$  est un ensemble de base, la solution de base correspondante est  $x^0 = (0, 0, 0, 1, 2, 4)$ . Et on a :

$$W = C_B A_B^{-1} A - C = \begin{pmatrix} 1 & 0 & 0 & 2 & 0 & 0 \\ 1 & 0 & 2 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \end{pmatrix}$$

**La phase 2 :**

On va appliquer la méthode de H.P.Benson pour trouver notre premier point extrême efficace.

En utilisant le programme (6.9) et par identification on obtient le programme suivant :

$$\left\{ \begin{array}{l} \min u_1 + 2u_2 + 4u_3 \\ w_1 + z_1 + z_2 - z_3 - u_1 - u_3 = -1, \\ w_2 + u_3 + 2z_1 - u_1 - u_2 = -2, \\ w_3 + z_3 - 2z_2 = 1, \\ w_4 - u_1 = 0, \\ -u_2 + w_5 = 0, \\ w_6 - u_3 = 0, \\ z_i \geq 0, i = \overline{1, 3}, \\ w_j \geq 1, j = \overline{1, 6}. \end{array} \right.$$

Pour résoudre ce programme, on applique la méthode du simplexe mono-objectif. A la fin, on trouvera le point suivant :

$$u^0 = (2, 0, 0), z^0 = (0, 0, 1), w^0 = (2, 0, 0, 2, 0, 0)$$

Pour obtenir un point extrême efficace pour le problème (6.11), on résout le programme suivant :

$$\left\{ \begin{array}{l} (z^0 + e)^T Cx \longrightarrow \max, \\ x \in S. \end{array} \right.$$

Par identification on obtient :

$$\left\{ \begin{array}{l} \max 2x_2 \\ x_1 + x_2 + x_4 = 1, \\ x_2 + x_5 = 2, \\ x_1 - x_2 + x_3 + x_6 = 4, \\ x_i \geq 0, i = \overline{1, 6}. \end{array} \right.$$

Ainsi notre point extrême efficace est  $x^{(1)} = (0, 1, 0, 0, 1, 5)$ , l'ensemble de base correspondant à ce point est  $J_{B^1} = \{2, 5, 6\}$ , et l'ensemble hors base est  $J_N^{(1)} = \{1, 3, 4\}$  avec :

$$W^{(1)} = \begin{pmatrix} 1 & 0 & 0 & 2 & 0 & 0 \\ -1 & 0 & 2 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \end{pmatrix}, A^{(1)} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & -1 & 1 & 0 \\ 2 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}, b^{(1)} = \begin{pmatrix} 1 \\ 1 \\ 5 \end{pmatrix}.$$

**La phase 3 :**

On fait entrer la variable  $x_1$  en base et on fait sortir la variable  $x_2$ . Notre ensemble de base devient  $J_B^{(2)} = \{1, 5, 6\}$ , donc l'ensemble hors base est  $J_N^{(2)} = \{2, 3, 4\}$  avec :

$$W^{(2)} = \begin{pmatrix} 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 & 0 \\ 0 & -1 & -1 & -1 & 0 & 0 \end{pmatrix}, \quad A^{(2)} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & -2 & 1 & -1 & 0 & 1 \end{pmatrix}, \quad b^{(2)} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}.$$

On pourrait résumer ces résultats dans le tableau suivant :

$w_1$	0	-1	0	1	0	0	1
$w_2$	0	1	2	1	0	0	1
$w_3$	0	-1	-1	-1	0	0	-1
$x_2$	1	1	0	1	0	0	1
$x_5$	0	1	0	0	1	0	2
$x_6$	0	-2	1	-1	0	1	3

Le point extrême correspondant à cette base est :  $x^{(2)} = (1, 0, 0, 0, 2, 3)$ .

**Test de Yu et Zeleny**

Testons si ce point est efficace pour le problème (6.11) en utilisant le test de Yu et Zeleny :

$$\begin{cases} \max g = e^T v, \\ Cx - I_k v = Cx^2, \\ x \in S, \\ v \in \mathbb{R}^3, v_i \geq 0, i = \overline{1, 3}. \end{cases}$$

Par identification on a :

$$\begin{cases} \max g = v_1 + v_2 + v_3 \\ x_1 + 2x_2 - v_1 = 1, \\ x_1 - 2x_3 - v_2 = 1, \\ -x_1 + x_3 - v_3 = -1, \\ x_1 + x_2 + x_4 = 1, \\ x_2 + x_5 = 2, \\ x_1 - x_2 + x_3 + x_6 = 4, \\ x_i \geq 0, i = \overline{1, 6}, \\ v_i \geq 0, i = \overline{1, 3}. \end{cases} \quad (6.12)$$

En résolvant le problème (6.12) on trouve que :  $\max g = 1$  d'où  $x^{(2)}$  est dominée.

*Remarque 6.6.3.* Puisque  $x^{(2)}$  n'est pas efficace, on revient à notre premier point extrême efficace c'est à dire  $x^{(1)} = (0, 1, 0, 0, 1, 5)$  avec  $J_B^{(1)} = \{2, 5, 6\}$  et  $J_N^{(1)} = \{1, 3, 4\}$ . On fait entrer la variable  $x_3$  en base et on fait sortir la variable  $x_6$ . Notre ensemble de base devient  $J_B^{(3)} = \{2, 3, 6\}$ , et on suit les mêmes étapes que pour  $x^{(2)}$ .

Pour  $J_B^{(3)} = \{1, 5, 6\}$ , le tableau de la base est le suivant :

$w_1$	0	-1	0	1	0	0	1
$w_2$	0	1	2	1	0	0	1
$w_3$	0	-1	-1	-1	0	0	-1
$x_2$	1	1	0	1	0	0	1
$x_5$	0	1	0	0	1	0	2
$x_6$	0	-2	1	-1	0	1	3

Le point extrême correspondant à cette base est :  $x^{(3)} = (1, 0, 0, 0, 2, 3)$ .

Si on trouve que  $x^{(3)}$  est efficace, on continue la procédure à partir de ce dernier sinon on revient à notre premier point extrême  $x^{(1)}$  et on fait entrer une nouvelle variable hors base.

On continue la procédure et à la fin de l'algorithme, nous avons identifié trois bases efficaces  $J_B^{(1)} = \{2, 5, 6\}$ ,  $J_B^{(2)} = \{1, 5, 6\}$  et  $J_B^{(3)} = \{2, 3, 5\}$ . Les solutions efficaces réalisables de base correspondantes sont respectivement  $x^{(1)} = (0, 1, 0, 0, 1, 5)$ ,  $x^{(2)} = (1, 0, 0, 0, 1, 0)$  et  $x^{(3)} = (0, 1, 5, 0, 1, 0)$ . Notez que les bases  $\{1, 5, 6\}$  et  $\{2, 3, 5\}$  ne sont pas adjacentes, car au moins deux pivots sont nécessaires pour obtenir l'un de l'autre. Ils sont cependant connectés via  $\{2, 5, 6\}$ ,



# Chapitre 7

## La méthode adaptée du support pour la résolution d'un programme linéaire multiobjectif

### 7.1 Introduction

Dans ce chapitre, nous proposons d'étudier un programme linéaire multicritère à variables bornées, et ce en utilisant la méthode adaptée du support [40]. Cette méthode est connue en optimisation mono-objectif pour son efficacité, et le grand gain en temps et en espace mémoire qu'elle présente.

Cette méthode a fait l'objet d'une publication dans le journal "Turkish Journal of Mathematics"[22]. La contribution de notre travail se situe beaucoup plus au niveau algorithmique qu'au niveau théorique. En effet, nous nous sommes attachés à concevoir une méthode algorithmique permettant de traiter efficacement des problèmes tout en respectant un cadre théorique solide justifiant rigoureusement notre méthode.

### 7.2 Position du problème et définitions générales

La forme standard d'un problème linéaire multicritère à variables bornées est définie par :

$$\begin{cases} \max Z(x) = Cx, \\ x \in S, \end{cases} \quad (7.1)$$

où  $S$  est l'ensemble des solutions réalisables défini comme suit :

$$S = \{x \in \mathbb{R}^n : Ax = b, d_1 \leq x \leq d_2\}. \quad (7.2)$$

On définit les indices des contraintes et des variables de décision par :  $I = \{1, 2, \dots, m\}$ ,

$J = \{1, 2, \dots, n\}$ , tel que :

$J = J_B \cup J_N$  avec  $J_B \cap J_N = \emptyset$ ,  $|J_B| = m$ .

Donc,  $x = x(J) = (x_j, j \in J)$ ,  $d_1 = d_1(J) = (d_{1j}, j \in J)$  et  $d_2 = d_2(J) = (d_{2j}, j \in J)$  sont des vecteurs d'ordre  $n$ ,  $b = (b_i, i \in I)$  est un vecteur d'ordre  $m$ ,  $A = A(I, J)$  est une matrice d'ordre  $m \times n$ , tel que  $\text{rang} A = m < n$ .

On définit la fonction multicritère comme suit :

$$Z(x) = \begin{pmatrix} z_1(x) \\ z_2(x) \\ \vdots \\ z_p(x) \end{pmatrix} = \begin{pmatrix} c_1^T x \\ c_2^T x \\ \vdots \\ c_p^T x \end{pmatrix} = Cx, \quad (7.3)$$

où  $C = C(K, J)$  avec  $K = \{1, 2, \dots, p\}$  est une matrice d'ordre  $p \times n$ , dont les lignes sont les  $n$ -vecteurs  $c_k^T, k \in \{1, 2, \dots, p\}$ .

On peut alors fractionner les vecteurs et les matrices de la manière suivante :

$$x = \begin{pmatrix} x_B \\ -- \\ x_N \end{pmatrix}, \quad x_B = x(J_B) = (x_j, j \in J_B), \quad x_N = x(J_N) = (x_j, j \in J_N),$$

$$C = \begin{pmatrix} C_B \\ -- \\ C_N \end{pmatrix}, \quad C_B = C(K, J_B), \quad C_N = C(K, J_N),$$

$$A = A(I, J) = (a_{ij}, 1 \leq i \leq m, 1 \leq j \leq n),$$

$$A = (A_B | A_N), \quad A_B = A(I, J_B), \quad A_N = A(I, J_N).$$

Nous donnons les définitions suivantes :

- Un vecteur  $x$  vérifiant les contraintes du problème (7.1) est appelé *plan* ou *solution réalisable* du problème.
- L'ensemble  $J_B \subset J$ ,  $|J_B| = m$ , est dit *support* si

$$\det A_B = \det A(I, J_B) \neq 0.$$

- Le couple  $\{x, J_B\}$  formé du plan  $x$  et du support  $J_B$  est appelé *plan de support*.
- Le plan de support est un *point extrême* si  $x_j = d_{1j}$  ou  $x_j = d_{2j}, \forall j \in J_N$ .
- Le plan de support est dit *non dégénéré* si

$$d^{1j} < x_j < d^{2j}, \quad j \in J_B.$$

- La matrice  $E^T = C_B A_B^{-1} A - C$  d'ordre  $p \times n$  est dite *la matrice des estimations*, avec  $E = (E_B^T, E_N^T)$ , tels que  $E_B = E(K, J_B)$  et  $E_N = E(K, J_N)$ .

– On définit la matrice des potentiels comme suit :  $U = C_B A_B^{-1}$ .

*Remarque 7.2.1.* Dans notre travail, on suppose que  $S$  est un ensemble borné et que le problème est non-dégénéré, c'est-à-dire  $d_1 < x < d_2$ , donc toutes les solutions réalisables ont au moins  $m$  composantes non critiques, avec  $m = \text{rang}(A)$ .

*Remarque 7.2.2.* Une solution réalisable de support est différente d'une solution réalisable de base. En effet, les variables non basiques d'une solution réalisable de base sont toutes nulles alors que celles d'une solution réalisable de support sont quelconques, c'est-à-dire positives ou nulles.

## 7.3 Définition des solutions efficaces et leurs propriétés

D'abord, nous rappelons les théorèmes suivants :

**Théorème 7.3.1.** *La solution réalisable  $x^0 \in S$  est dite efficace dans le problème (7.1), s'il n'existe aucune autre solution réalisable  $\bar{x} \in S$  telle que*

$$C\bar{x} \geq Cx^0 \text{ et } C\bar{x} \neq Cx^0.$$

*On note l'ensemble des solutions efficaces par  $S^E$ .*

**Théorème 7.3.2.** *La solution réalisable  $x^0 \in S$  est dite faiblement efficace dans le problème (7.1), s'il n'existe aucune autre solution réalisable  $\bar{x} \in S$  telle que*

$$C\bar{x} > Cx^0.$$

**Théorème 7.3.3.** *(Critère d'efficacité) Soient  $\{x, J_B\}$  un plan de support du problème (7.1) et  $k \in K$ .*

*Si*

$$\begin{cases} E_{kj} \geq 0, & \text{si } x_j = d_{1j}, \quad j \in J_N, \\ E_{kj} \leq 0, & \text{si } x_j = d_{2j}, \quad j \in J_N, \\ E_{kj} = 0, & \text{si } d_{1j} < x_j < d_{2j}, \quad j \in J_N, \end{cases} \quad (7.4)$$

*alors le plan de support  $\{x, J_B\}$  est optimal pour l'objectif  $k$ , le plan  $x$  est alors dit faiblement efficace pour le problème (7.1).*

*Si le plan de support est non-dégénéré, alors ces mêmes relations sont aussi nécessaires pour que  $x$  soit faiblement efficace c'est à dire on aura l'équivalence.*

Pour discuter sur certaines des propriétés théoriques de base et d'autres approches du problème, voir, par exemple, [82, 37, 36].

Le problème de la programmation linéaire multicritère à variables bornées peut être considéré comme le problème de recherche de toutes les solutions qui sont efficaces ou faiblement efficaces du problème (7.1) pour  $C, A, b, d_1$  et  $d_2$  donnés. L'algorithme que nous allons développer permet de déterminer aussi des solutions  $\epsilon$ -efficaces.

## 7.4 Définition des solutions $\epsilon$ -efficaces et leurs propriétés

**Définition 7.4.1.** Soit  $\epsilon \in \mathbb{R}^p, \epsilon \geq 0$ . Une solution réalisable  $x^\epsilon \in S$  est dite  $\epsilon$ -efficace pour le problème (7.1) s'il n'existe aucune autre solution réalisable  $\bar{x} \in S$  telle que  $c_k^T \bar{x} - c_k^T x^\epsilon \leq \epsilon_k, \forall k \in K$ .

On note l'ensemble des solutions  $\epsilon$ -efficaces du problème (7.1) par  $S_\epsilon^E$ .

Les propriétés suivantes sont les résultats directs de la définition des solutions  $\epsilon$ -efficaces :

- Propriété 7.4.1.**
1.  $S^E \subset S_\epsilon^E, \forall \epsilon > 0$  et  $S^E = S_\epsilon^E$ , pour  $\epsilon = 0$ .
  2. Si  $\epsilon_1 > \epsilon_2 > 0$ , alors  $S_{\epsilon_2}^E \subset S_{\epsilon_1}^E$ .

**Lemme 7.4.1.** La solution réalisable  $x^\epsilon \in S$  est dite  $\epsilon$ -efficace pour le problème (7.1) si, et seulement si il existe  $x^0 \in S$  de telle sorte que tous les vecteurs  $\lambda \in \mathbb{R}_+^p, \sum_{k=1}^p \lambda_k = 1$ , satisfont la condition  $\lambda^T C x^0 = \max_{x \in S} \lambda^T C x$ , Nous avons l'inégalité suivante :

$$\lambda^T (C x^0 - C x^\epsilon) \leq \epsilon.$$

**Définition 7.4.2.** La valeur

$$\beta_k(x, J_B) = \sum_{j \in J_N, E_{kj} > 0} E_{kj}(x_j - d_{1j}) + \sum_{j \in J_N, E_{kj} < 0} E_{kj}(x_j - d_{2j}),$$

est appelée *estimation de suboptimalité* de l'objectif  $k, k \in \{1, \dots, p\}$ .

**Théorème 7.4.1.** (Caractérisation des solutions  $\epsilon$ -efficace)

Soit  $\{x, J_B\}$  un plan de support du problème (7.1) et  $\epsilon$  un vecteur réel positif ou nul arbitraire de dimension  $p$ .

S'il existe  $k \in \{1, \dots, p\}$  tel que  $\beta_k(x, J_B) \leq \epsilon_k$ , alors  $x$  est  $\epsilon_k$ -faiblement efficace du problème (7.1).

Si  $\beta(x, J_B) = (\beta_k(x, J_B), k \in \{1, \dots, p\}) \leq \epsilon$ , alors  $x$  est  $\epsilon$ -efficace.

## 7.5 Procédure de recherche d'un point efficace initial

### 7.5.1 Trouver un point efficace initial en utilisant la méthode d'Isermann

En s'inspirant de la procédure proposée par H.Isermann pour trouver un point efficace initial, nous proposons une procédure qui va tenir compte des contraintes du programme (7.1). La résolution des programmes linéaires introduits dans la procédure va se faire en utilisant la méthode adaptée du support pour la résolution d'un programme linéaire [70, 40].

La procédure est donnée dans les étapes suivantes :

**Étape (1) :** Trouver une solution optimale  $(u^0, v^0, \gamma^0, \alpha^0, \beta^0)$  pour le programme suivant en utilisant la méthode directe de support pour la résolution d'un programme linéaire à variables non nulle :

$$\begin{cases} \min(b - Ad_1)^T u + (d_2 - d_1)^T \gamma, \\ u^T A - v^T C + \gamma^T - \alpha^T = 0, \\ v - \beta = e, \\ \alpha, \gamma, \beta \geq 0. \end{cases} \quad (7.5)$$

où  $e \in \mathbb{R}^k$  est un vecteur dont tous les éléments sont égaux à un. Ensuite aller à l'étape (2).

Sinon arrêter la procédure.

**Étape (2) :** Obtenir un point efficace pour le problème (7.1) qui est la solution optimale du programme :

$$\begin{cases} \max(v^0)^T Cx, \\ Ax = b, \\ d_1 \leq x \leq d_2, \end{cases} \quad (7.6)$$

en utilisant la méthode adaptée du support pour la résolution d'un programme linéaire à variables bornées [40].

Pour établir la validité de cette procédure, nous prouvons les résultats suivants :

**Théorème 7.5.1.** *Si le programme (7.5) admet une solution optimale  $(u^0, v^0, \gamma^0, \alpha^0, \beta^0)$ , alors le programme (7.15) admet une solution optimale qui est un point efficace pour le problème (7.1).*

*Démonstration.* Soit  $\lambda \in \mathbb{R}^p, \lambda > 0$ , et considérez le programme linéaire suivant :

$$\begin{cases} \max \lambda^T Cx, \\ Ax = b, \\ d_1 \leq x \leq d_2. \end{cases} \quad (7.7)$$

On pose  $y = x - d_1$  dans le programme linéaire (7.7). On a :

$$\begin{cases} \max \lambda^T C y + \lambda^T C d_1, \\ A y = b - A d_1, \\ y \leq d_2 - d_1, \\ y \geq 0. \end{cases} \quad (7.8)$$

Soit  $(u^0, v^0, \gamma^0, \alpha^0, \beta^0)$  la solution optimale de (7.5). Le programme dual de (7.6) est donné par

$$\begin{cases} \min u^T (b - A d_1) + \gamma^T (d_2 - d_1), \\ u^T A + \gamma^T \geq (v^0)^T C, \\ \gamma \geq 0. \end{cases} \quad (7.9)$$

Comme  $(u^0, v^0, \gamma^0, \alpha^0, \beta^0)$  est la solution optimale du programme (7.5), alors  $(u^0, \gamma^0)$  est une solution réalisable pour le programme (7.9). Puisque l'ensemble  $S$  est non vide, le programme (7.6) est un problème réalisable. De la théorie de la dualité, le programme (7.8), avec  $\lambda = v^0$ , admet une solution optimale.  $\square$

Le théorème suivant permet de trouver une solution efficace du problème multicritère (7.1) en résolvant un programme linéaire à variables bornées :

**Théorème 7.5.2.** [29, 25] *Le programme linéaire suivant :*

$$\begin{cases} \max e^T C x, \\ x \in S. \end{cases} \quad (7.10)$$

*admet une solution optimale si et seulement si, le programme (7.1) possède une solution efficace.*

**Théorème 7.5.3.** *Le programme (7.5) admet une solution optimale si et seulement si, le programme (7.1) possède une solution efficace.*

*Démonstration.* Le programme dual de (7.5) est donné par :

$$\begin{cases} \max e^T z, \\ A y = b - A d_1, \\ -C y + z = 0, \\ y \leq d_2 - d_1, \\ y \geq 0, \\ z \geq 0. \end{cases} \quad (7.11)$$

Si on effectue le changement de variables  $y + d_1 = x$ , on obtient le programme suivant :

$$\begin{cases} \max e^T z, \\ A x = b, \\ z = C x - C d_1, \\ d_1 \leq x \leq d_2, \\ z \geq 0. \end{cases} \quad (7.12)$$

On a  $z = Cx - Cd_1 \geq 0$ , alors le programme (7.12) est équivalent au programme suivant :

$$\begin{cases} \max e^T Cx - e^T Cd_1, \\ Ax = b, \\ d_1 \leq x \leq d_2. \end{cases} \quad (7.13)$$

Or, comme  $e^T Cd_1$  est une valeur constante, alors le programme (7.13) est équivalent au programme (7.11).

En appliquant le théorème 7.5.2 et d'après la théorie de la dualité, on établit le théorème.  $\square$

Le théorème suivant décrit une classe de problèmes linéaires multiobjectif pour lesquels la procédure d'Isermann est valide.

**Théorème 7.5.4.** *La méthode d'Isermann est valable si  $T = \{x \in S, Cx \geq 0\}$  est non vide.*

Nous proposons donc d'utiliser une nouvelle méthode pour générer un premier point extrême efficace présenté dans la section qui suit.

### 7.5.2 Trouver un point efficace initial en utilisant la procédure de Benson-Radjef-Bibi

Inspiré par le processus proposé par H.P.Benson, les auteurs de [82] ont développé une procédure pour trouver une solution efficace initiale. Mais dans leur méthode, ils utilisent la méthode directe du support pour résoudre les programmes linéaires mono-critère introduits dans la procédure. Ici, nous utilisons la même procédure, mais nous résolvons les programmes linéaires mono-critère en utilisant la méthode adaptée du support [70, 40].

La procédure utilisée est donnée dans les étapes suivantes :

**Étape (1) :** Trouver une solution réalisable  $x^0 \in S$ .

**Étape (2) :** Si  $S \neq \emptyset$ , trouver une solution optimale  $(u^0, w^0, \gamma^0, \alpha^0)$  pour le problème suivant :

$$\begin{cases} \min u^T(-Cx^0 + Cd_1) + w^T(b - Ad_1) + \gamma^T(d_2 - d_1), \\ u^T C - w^T A - \gamma^T + \alpha^T = -e^T C, \\ u, \alpha, \gamma \geq 0, \end{cases} \quad (7.14)$$

et aller à l'étape (3).

Sinon, arrête la procédure, le problème (7.1) n'est pas réalisable.

**Etape (3) :** Obtenir un point efficace du programme (7.1) en résolvant le programme suivant, et ce en utilisant la méthode adaptée du support [40] :

$$\begin{cases} \max (u^0 + e)^T Cx, \\ Ax = b, \\ d_1 \leq x \leq d_2. \end{cases} \quad (7.15)$$

## 7.6 Recherche des solutions efficaces

Dans cette phase, nous adoptons le principe de la méthode adaptée du support, dont la stratégie est la suivante :

On localise les points efficaces en faisant entrer toutes les variables non basiques correspondantes à une solution efficace en base. Ainsi à partir du premier point efficace, on calcule un point voisin et on teste s'il est efficace. S'il ne l'est pas, on revient à un point efficace, puis le procédé est itéré.

### 7.6.1 Test d'efficacité d'une variable non basique

Pour tester l'efficacité d'un point  $x^*$  dans le programme linéaire multiobjectif (7.1), nous reprenons le test développé par S.Radjeff et M.O.Bibi [82, 70].

On introduit un vecteur colonne  $v$  de dimension  $p$  et on définit le programme linéaire suivant :

$$\begin{cases} \max g = e^T v, \\ Ax = b, \\ Cx - v = Cx^*, \\ d_1 \leq x \leq d_2, \\ v \geq 0. \end{cases} \quad (7.16)$$

**Théorème 7.6.1.** *Si  $\max g = 0$ , alors  $x^*$  est efficace. Sinon,  $x^*$  est non efficace.*

Le programme (7.16) est un programme linéaire hybride; on le résout par la méthode directe de support pour la résolution d'un P.L à variables hybrides [73]. Pour ce, on introduit les notations suivantes pour le transformer en un P.L à variables bornées que nous proposons de résoudre par la méthode adaptée du support :

$$\bar{A} = \begin{pmatrix} A \\ C \end{pmatrix}, \bar{H} = \begin{pmatrix} 0 \\ -I \end{pmatrix}, \bar{b} = \begin{pmatrix} b \\ Cx^* \end{pmatrix},$$

où  $I$  est la matrice identité d'ordre  $k$ .



Le programme (7.16) devient alors

$$\begin{cases} h = e^T v \longrightarrow \max, \\ \bar{A}x + \bar{H}v = \bar{b}, \\ d_1 \leq x \leq d_2, \\ v \geq 0. \end{cases} \quad (7.17)$$

où  $e$  est un  $k$ -vecteur dont tous les éléments sont égaux à 1.

Cependant, deux cas particuliers peuvent se présenter :

**Cas particulier 1 :** Dans le cas où tous les éléments de la matrice  $C$  sont positifs ou nuls, on obtient le programme test :

$$\begin{cases} h = e^T v \longrightarrow \max, \\ Ax = b, \\ Cx - v = Cx^*, \\ d_1 \leq x \leq d_2, \\ Cd_1 - Cx^* \leq v \leq Cd_2 - Cx^*. \end{cases} \quad (7.18)$$

On introduit les notations suivantes :

$$y = \begin{pmatrix} x \\ v \end{pmatrix}, \bar{A} = \begin{pmatrix} A & 0 \\ C & -I \end{pmatrix}, \bar{b} = \begin{pmatrix} b \\ Cx^* \end{pmatrix}, \bar{C} = \begin{pmatrix} 0 & e \end{pmatrix},$$

$$\bar{d}_1 = \begin{pmatrix} d_1 \\ Cd_1 - Cx^* \end{pmatrix}, \bar{d}_2 = \begin{pmatrix} d_2 \\ Cd_2 - Cx^* \end{pmatrix}.$$

On obtient alors le programme test suivant :

$$\begin{cases} h = \bar{C}y \longrightarrow \max, \\ \bar{A}y = \bar{b}, \\ \bar{d}_1 \leq y \leq \bar{d}_2. \end{cases} \quad (7.19)$$

En utilisant la méthode adaptée du support pour la résolution d'un P.L. à variables bornées [13], on résout le programme linéaire (7.19).

**Cas particulier 2 :** Dans le cas où tous les éléments de la matrice  $C$  sont négatifs ou nuls, on a le programme test suivant :

$$\begin{cases} h = e^T v \longrightarrow \max, \\ Ax = b, \\ Cx - v = Cx^*, \\ d_1 \leq x \leq d_2, \\ Cd_2 - Cx^* \leq v \leq Cd_1 - Cx^*. \end{cases} \quad (7.20)$$

On introduit les notations suivantes :

$$y = \begin{pmatrix} x \\ v \end{pmatrix}, \bar{A} = \begin{pmatrix} A & 0 \\ C & -I \end{pmatrix}, \bar{b} = \begin{pmatrix} b \\ Cx^* \end{pmatrix}, \bar{C} = \begin{pmatrix} 0 & e \end{pmatrix},$$

$$\bar{d}_1 = \begin{pmatrix} d_1 \\ Cd_2 - Cx^* \end{pmatrix}, \bar{d}_2 = \begin{pmatrix} d_2 \\ Cd_1 - Cx^* \end{pmatrix}.$$

On obtient alors le programme test suivant :

$$\begin{cases} h = \bar{C}y \rightarrow \max, \\ \bar{A}y = \bar{b}, \\ \bar{d}_1 \leq y \leq \bar{d}_2. \end{cases} \quad (7.21)$$

En utilisant la méthode adaptée du support pour la résolution d'un P.L. à variables bornées [13], on résout le programme linéaire (7.21).

### 7.6.2 Construction de la nouvelle solution efficace

La méthode de recherche des points efficaces consiste à faire entrer les variables hors base correspondantes au premier point efficace trouvé dans la deuxième phase une après l'autre.

Le changement de plan consiste à augmenter  $z_{k_0}(x)$  et le Changement du support consiste à diminuer  $\Theta_{k_0}(\lambda)$ , ( $\Theta_{k_0}(\lambda)$  étant la fonction duale de  $z_{k_0}(x)$ ).

La construction du nouveau plan  $\bar{x} = x + \theta^0 l$ , consiste à choisir  $l \in \mathbb{R}^n$ , appelée direction d'amélioration, et un nombre non négatif  $\theta^0$  qui est le pas maximal le long de cette direction tel que  $z_{k_0}(\bar{x}) \geq z_{k_0}(x)$ .

Soit  $k_0$  le critère vérifiant la relation :

$$\sum_{j \in J_N} |E_{k_0 j}| = \max_{k=1,p} \left( \sum_{j \in J_N} |E_{kj}| \right).$$

Sur  $J_N$ , on pose  $\theta = 1$ , et

$$l_j = \begin{cases} d_{1j} - x_j, & \text{si } E_{k_0 j} > 0, \\ d_{2j} - x_j, & \text{si } E_{k_0 j} < 0, \\ 0 & \text{si } E_{k_0 j} = 0, \quad j \in J_N. \end{cases} \quad (7.22)$$

Sur  $J_B$ , on pose  $l(J_B) = -A_B^{-1} A_N l(J_N)$  pour avoir  $A\bar{x} = b$ , pour que  $\bar{x}$  vérifie  $d_1 \leq \bar{x} \leq d_2$ . D'autre part, le pas maximal  $\theta^0$  le long de la direction  $l$  doit vérifier :

$$d_{1j} - x_j \leq \theta^0 l_j \leq d_{2j} - x_j, \quad j \in J_B.$$

Par conséquent, on a

$$\theta_j = \begin{cases} \frac{d_{2_j} - x_j}{l_j}, & \text{si } l_j > 0, \\ \frac{d_{1_j} - x_j}{l_j}, & \text{si } l_j < 0, \\ \infty, & \text{si } l_j = 0, \quad j \in J_B. \end{cases}$$

avec

$$\theta_{j_0} = \min_{j \in J_B} \theta_j,$$

où  $j_0$  étant l'indice candidat à sortir de la base.

Le pas maximal sera

$$\theta^0 = \min(1, \theta_{j_0}).$$

Le nouveau plan  $\bar{x}$  s'écrit donc  $\bar{x} = x + \theta^0 l$ .

**Calcul de  $\beta(\bar{x}, J_B)$**

Pour  $k \in \{1, \dots, p\}$ , on a :

$$\begin{aligned} \beta_k(\bar{x}, J_B) &= \sum_{E_{kj} > 0, j \in J_N} E_{kj}(\bar{x}_j - d_{1_j}) + \sum_{E_{kj} < 0, j \in J_N} E_{kj}(\bar{x}_j - d_{2_j}) \\ &= \beta_k(x, J_B) + \theta^0 \left( \sum_{E_{kj} > 0, j \in J_N} E_{kj} l_j + \sum_{E_{kj} < 0, j \in J_N} E_{kj} l_j \right). \end{aligned}$$

Ainsi, on obtient :

$$\beta_{k_0}(\bar{x}, J_B) = \beta_{k_0}(x, J_B) + \theta^0 \left( \sum_{E_{k_0j} > 0, j \in J_N} E_{k_0j} l_j + \sum_{E_{k_0j} < 0, j \in J_N} E_{k_0j} l_j \right).$$

En remplaçant  $l_j$  donné par les relations (7.22), on a

$$\beta_{k_0}(\bar{x}, J_B) = (1 - \theta^0) \beta_{k_0}(x, J_B).$$

De cette dernière expression on déduit que :

- Si  $\theta^0 = 1$  alors  $\bar{x}$  est optimal pour l'objectif  $k_0$  ; on dit aussi que  $\bar{x}$  est un point efficace de Slater ; on appelle le programme test pour tester l'efficacité de ce point ; si ce point est efficace, on considère alors toutes les variables non basique.
- Si  $\beta_{k_0}(\bar{x}, \bar{J}_B) < \epsilon_{k_0}$ , on peut arrêter l'algorithme avec  $\bar{x}$  is  $\epsilon$ -optimal pour l'objectif  $k_0$  ; on dit aussi que  $\bar{x}$  est  $\epsilon$ -faiblement efficace.
- Si  $\beta_{k_0}(\bar{x}, \bar{J}_B) > \epsilon_{k_0}$ , on recommencera alors une nouvelle itération avec le nouveau plan de support  $\{\bar{x}, \bar{J}_B\}$ , on changera  $J_B$  de la manière suivante :

### Changement de support

Le changement de support consiste à faire un changement de coplan  $E$  vers  $\bar{E}$ , et du vecteur des potentiels  $U$  vers  $\bar{U}$  de telle sorte que :

$$\beta_{k_0}(\bar{x}, J_B) \leq \beta_{k_0}(\bar{x}, \bar{J}_B).$$

Pour cela on pose :

$$\begin{aligned}\bar{E}_{k_0j} &= E_{k_0j} + \sigma_0 t_j, \\ \bar{U}_{k_0j} &= U_{k_0j} + \sigma_0 t_j,\end{aligned}$$

où  $t$  est la direction de diminution de la fonction duale,  $\sigma_0$  est le pas maximal le long de cette direction.

#### Calcul de $t$ et $\sigma_0$

Après calcul du plan  $\bar{x} = x + \theta^0 l$ , le pas  $\theta^0$  est donné par :

$$\theta^0 = \min(1, \theta_{j_0}) = \theta_{j_0}, \quad j_0 \in J_B.$$

On cherchera un indice  $j_1 \in J_N$  qui va entrer dans la base à la place de  $j_0$ .

Pour cela posons :

$$t_j = \begin{cases} -\text{sign}(l_{j_0}), & \text{si } j = j_0, \\ 0, & \text{si } j \in J_B \setminus j_0, \end{cases}$$

$$t(J_N) = t(J_B) A_B^{-1} A_N.$$

et calculons

$$\sigma_0 = \sigma_{j_1} = \min_{j \in J_N}(\sigma_j),$$

avec

$$\sigma_j = \begin{cases} \frac{-E_{k_0j}}{t_j}, & \text{si } E_{k_0j} t_j < 0, \\ 0, & \text{si } (E_{k_0j} = 0 \text{ et } x_j \neq d_{1_j} \text{ pour } t_j > 0) \text{ ou} \\ & (E_{k_0j} = 0 \text{ et } x_j \neq d_{2_j} \text{ pour } t_j < 0), \quad j \in J_N \\ \infty, & \text{sinon.} \end{cases}$$

On a  $\bar{E}_{k_0j_1} = 0$ .

Le nouveau support est  $\bar{J}_B = (J_B \setminus j_0) \cup j_1$ .

On peut remarquer que :

$$\begin{aligned}\beta_{k_0}(\bar{x}, \bar{J}_B) &= \sum_{\bar{E}_{k_0j} > 0, j \in \bar{J}_N} \bar{E}_{k_0j}(\bar{x}_j - d_{1_j}) + \sum_{\bar{E}_{k_0j} < 0, j \in \bar{J}_N} \bar{E}_{k_0j}(\bar{x}_j - d_{2_j}) \\ &= (1 - \theta^0) \beta_{k_0}(x, J_B) + \sigma_0 \left( \sum_{E_{k_0j} > 0, j \in \bar{J}_N} t_j(\bar{x}_j - d_{1_j}) + \sum_{E_{k_0j} < 0, j \in \bar{J}_N} t_j(\bar{x}_j - d_{2_j}) \right).\end{aligned}$$

De cette dernière expression, on déduit que :

- Si  $\beta_{k_0}(\bar{x}, \bar{J}_B) > \epsilon_{k_0}$ , on recommencera alors une nouvelle itération avec le nouveau plan de support  $\{\bar{x}, \bar{J}_B\}$ .
- Si  $\beta_{k_0}(\bar{x}, \bar{J}_B) < \epsilon_{k_0}$ , on peut arrêter l'algorithme avec  $\bar{x}$  est  $\epsilon$ -optimal pour l'objectif  $k_0$ ; on dit aussi que  $\bar{x}$  est  $\epsilon$ -faiblement efficace.
- Si  $\beta_{k_0}(\bar{x}, \bar{J}_B) = 0$ , alors on arrête la procédure en ayant trouvé un point efficace. On appelle alors le programme test (donné dans la section précédente) pour vérifier l'efficacité de ce point, puis on recommence le procédé en considérant une autre variable non basique.

## 7.7 L'algorithme de la méthode

Les étapes de la méthode de recherche de points efficaces sont résumées dans l'algorithme suivant :

- I. Trouver le premier point efficace en utilisant la procédure suivante :
  - Trouver une solution optimale  $(u^0, w^0, \gamma^0, \alpha^0)$  du programme (7.14).
  - Obtenir un point efficace solution optimale du programme (7.15).  
Soit  $x^1$  le point trouvé, aller en II.
- II. Poser  $s = 1$  :
  - (1) Soit  $\{x^s, J_B\}$  une solution de support réalisable et  $\epsilon \geq 0$ .
    - Calculer  $U = C_B A_B^{-1}$ ,
    - Calculer  $E = (E_B, E_N) = U A - C$ ,
  - (2) Choisir le critère  $k_0$ ,
    - Calculer  $\beta_{k_0}(x^s, J_B)$  :
      - Si  $\beta_{k_0}(x^s, J_B) > \epsilon_{k_0}$ , aller en (3).
      - Si  $\beta_{k_0}(x^s, J_B) < \epsilon_{k_0}$ , on peut arrêter l'algorithme avec  $x^s$  est une solution  $\epsilon_{k_0}$ -faiblement efficace, aller en (5).
      - Si  $\beta_{k_0}(x^s, J_B) = 0$ , alors on arrête l'algorithme avec  $x^s$  est une solution faiblement efficace, aller en (5).
  - (3) Calculer une nouvelle solution réalisable en utilisant la procédure suivante :
    - Calculer le vecteur  $l$ ,
    - Déterminer l'indice  $j_0$  et le pas maximal  $\theta^0$ ,  
– Poser  $s = s + 1$ .
    - Calculer  $x^s = x^{s-1} + \theta^0 l$ ,
    - Calculer  $\beta_{k_0}(x^s, J_B) = (1 - \theta^0)\beta_{k_0}(x^{s-1}, J_B)$  :
      - Si  $\beta_{k_0}(x^s, J_B) > \epsilon_{k_0}$ , aller en (4).
      - Si  $\beta_{k_0}(x^s, J_B) < \epsilon_{k_0}$ , on peut arrêter l'algorithme avec  $x^s$  est  $\epsilon_{k_0}$ -faiblement efficace.

- Si  $\theta^0 = 1$ , alors on peut arrêter l'algorithme ayant trouvé une solution faiblement efficace, aller en (7).
- (4) Changer de support :
- Calculer le vecteur  $t$ ,
  - Calculer  $\sigma_0$ , et déterminer l'indice  $j_1$ ,
  - Déterminer le nouveau support  $\bar{J}_B = (J_B \setminus j_0) \cup j_1$ ,
  - Calculer  $\beta(x^s, \bar{J}_B)$ ,
    - Si  $\beta_{k_0}(x^s, \bar{J}_B) > \epsilon_{k_0}$ , aller en (5).
    - Si  $\beta_{k_0}(x^s, \bar{J}_B) < \epsilon_{k_0}$ , on peut arrêter l'algorithme avec  $x^s$  est  $\epsilon_{k_0}$ -faiblement efficace.
    - Si  $\beta_{k_0}(x^s, \bar{J}_B) = 0$ , alors on peut arrêter l'algorithme en ayant trouvé une solution faiblement efficace, aller en (7).
- (5) Vérifier si l'introduction de la  $j^{\text{ème}}$  colonne correspondante conduit à une base inexplorée ?
- Si oui, aller en (6).
  - Sinon, arrêter l'algorithme, les points sont tous trouvés.
- (6) Peut-on améliorer un autre objectif ?
- Si oui, aller en (1).
  - Sinon, arrêter l'algorithme, les points sont tous trouvés.
- (7) Considerons le programme (7.16) avec  $x^* = x^s$  :
- Si  $\max g = 0$ , la solution  $x^s$  est efficace.
  - Sinon, aller en (6).

## 7.8 Exemple numérique

Soit le problème de programmation linéaire à variables bornées suivant :

$$\left\{ \begin{array}{l} \max z_1(x) = x_1 + 2x_2, \\ \max z_2(x) = x_1 - 2x_3, \\ \max z_3(x) = -x_1 + x_3, \\ x_1 + x_2 + x_4 = 1, \\ x_2 + x_5 = 2, \\ x_1 - x_2 + x_3 + x_6 = 4, \\ -1 \leq x_1 \leq 1, \\ -2 \leq x_2 \leq 2, \\ -3 \leq x_3 \leq 3, \\ -4 \leq x_4 \leq 4, \\ -5 \leq x_5 \leq 5, \\ -6 \leq x_6 \leq 6. \end{array} \right. \quad (7.23)$$

On a  $x^0 = (0, 0, 0, 1, 2, 4)$  est une solution réalisable.

### I. Calcul du premier point efficace :

D'abord, on doit résoudre le programme (7.14) ; par identification on a :

$$\left\{ \begin{array}{l} \min(-u_1 + 5u_2 - 2u_3 + 8w_1 + 9w_2 + 12w_3 + 2\gamma_1 + 4\gamma_2 + 6\gamma_3 + 8\gamma_4 + 10\gamma_5 + 12\gamma_6), \\ u_1 + u_2 - u_3 - w_1 - w_3 - \gamma_1 + \alpha_1 = -1, \\ 2u_1 - w_1 - w_2 + w_3 - \gamma_2 + \alpha_2 = -2, \\ -2u_2 + u_3 - w_3 - \gamma_3 + \alpha_3 = 1, \\ -w_1 - \gamma_4 + \alpha_4 = 0, \\ -w_2 - \gamma_5 + \alpha_5 = 0, \\ -w_3 - \gamma_6 + \alpha_6 = 0, \\ u \geq 0, \alpha \geq 0, \gamma \geq 0. \end{array} \right.$$

Pour résoudre ce programme, on pose :

$$w = w^+ - w^- \text{ avec } w^+ = \max(0, w) \text{ et } w^- = \max(0, -w),$$

Ainsi, on obtient le programme suivant :

$$\left\{ \begin{array}{l} \min(-u_1 + 5u_2 - 2u_3 + 8w_1^+ + 9w_2^+ + 12w_3^+ - 8w_1^- - 9w_2^- - 12w_3^- \\ \quad + 2\gamma_1 + 4\gamma_2 + 6\gamma_3 + 8\gamma_4 + 10\gamma_5 + 12\gamma_6), \\ u_1 + u_2 - u_3 - w_1^+ - w_3^+ + w_1^- + w_3^- - \gamma_1 + \alpha_1 = -1, \\ 2u_1 - w_1^+ - w_2^+ + w_3^+ + w_1^- + w_2^- - w_3^- - \gamma_2 + \alpha_2 = -2, \\ -2u_2 + u_3 - w_3^+ + w_3^- - \gamma_3 + \alpha_3 = 1, \\ -w_1^+ + w_1^- - \gamma_4 + \alpha_4 = 0, \\ -w_2^+ + w_2^- - \gamma_5 + \alpha_5 = 0, \\ -w_3^+ + w_3^- - \gamma_6 + \alpha_6 = 0, \\ u \geq 0, w^+ \geq 0, w^- \geq 0, \alpha \geq 0, \gamma \geq 0. \end{array} \right.$$

La solution optimale de ce programme est :

$$(0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0).$$

Pour obtenir notre premier point efficace résolvons le programme suivant :

$$\left\{ \begin{array}{l} \max(u^0 + e)^T Cx, \\ Ax = b, \\ d_1 \leq x \leq d_2. \end{array} \right.$$

Le problème à résoudre est donc :

$$\left\{ \begin{array}{l} \max(2x_2), \\ x_1 + x_2 + x_4 = 1, \\ x_2 + x_5 = 2, \\ x_1 - x_2 + x_3 + x_6 = 4, \\ -1 \leq x_1 \leq 1, \\ -2 \leq x_2 \leq 2, \\ -3 \leq x_3 \leq 3, \\ -4 \leq x_4 \leq 4, \\ -5 \leq x_5 \leq 5, \\ -6 \leq x_6 \leq 6. \end{array} \right.$$

La solution optimale de ce programme en utilisant la méthode adaptée du support est :  $(1, 2, -1, -2, 0, 6)$ . Ainsi notre premier point efficace est :

$$x^{(1)} = (1, 2, -1, -2, 0, 6).$$

Posons  $J_B = \{3, 4, 5\}$  et  $J_N = \{1, 2, 6\}$ , et aller en II.

**II. Recherche de tous les points efficaces associés à notre premier point efficace :**

**1<sup>ère</sup> itération :** Soit  $\{x^1, J_B\}$  une solution de support réalisable et  $\epsilon = \begin{pmatrix} 0.01 \\ 0.01 \\ 0.01 \end{pmatrix}$ .

- Calculons la matrice des estimations  $E$  :

$$E = C_B A_B^{-1} A - C.$$

On a :

$$E_N = \begin{pmatrix} -1 & -2 & 0 \\ -3 & 2 & -2 \\ 2 & -1 & 1 \end{pmatrix} \text{ et } E_B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

donc

$$E = \begin{pmatrix} -1 & -2 & 0 & 0 & 0 & 0 \\ -3 & 2 & 0 & 0 & 0 & -2 \\ 2 & -1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

- Déterminer le critère  $k_0$  such as :

$$\sum_{j \in J_N} |E_{k_0 j}| = \max \left( \sum_{j \in J_N} |E_{1j}|, \sum_{j \in J_N} |E_{2j}|, \sum_{j \in J_N} |E_{3j}| \right) = \sum_{j \in J_N} |E_{2j}| = 7,$$

d'où  $k_0 = 2$ .



- On a  $\beta_2(x^{(1)}, J_B) = 8 > \epsilon_2$ , alors le point  $x^{(1)}$  n'est pas optimal pour l'objectif  $k_0 = 2$ .
- Calculer la direction d'amélioration  $l$  :

$$\begin{cases} l_N = \begin{pmatrix} l_1 \\ l_2 \\ l_6 \end{pmatrix} = \begin{pmatrix} 0 \\ -4 \\ 0 \end{pmatrix}, \\ l_B = \begin{pmatrix} l_3 \\ l_4 \\ l_5 \end{pmatrix} = \begin{pmatrix} -4 \\ -4 \\ 4 \end{pmatrix} \end{cases}$$

- Calculer le pas maximal

$$\theta_{j_0} = \min_{j \in J_B} \theta_j = \min(\theta_3, \theta_4, \theta_5) = \theta_3 = 0.5,$$

donc  $j_0 = 3$  et  $\theta^0 = \min(1, \theta_{j_0}) = 0.5$ .

- Calculer  $x^{(2)}$  :

$$x^{(2)} = x^{(1)} + \theta^0 l = \begin{pmatrix} 1 \\ 0 \\ -3 \\ 0 \\ 2 \\ 6 \end{pmatrix}$$

- Calculer  $\beta_2(x^{(2)}, J_B) = 4 > \epsilon_2$ , donc  $\{x^{(2)}, J_B\}$  n'est pas efficace pour le problème (7.23).

#### Changement de support

- $t = \begin{pmatrix} 1 \\ -1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix},$

- $\sigma = \begin{pmatrix} 3 \\ 2 \\ 0 \\ 0 \\ 0 \\ 2 \end{pmatrix},$

- $\sigma_0 = \sigma_{j_1} = \sigma_2 = 2$ , donc  $j_1 = 2$ .

Notre nouveau support est :  $J_B^{(1)} = (J_B \setminus j_0) \cup j_1 = \{2, 4, 5\}$ .

$\beta_2(x^{(2)}, J_B) = 0$ , alors  $\{x^{(2)}, J_B^{(1)}\}$  est optimal pour l'objectif  $k_0 = 2$ , donc  $x^{(2)}$  est une solution faiblement efficace pour le problème (7.23).

- Considérons le programme (7.16). Par identification, on obtient le programme suivant :

$$\left\{ \begin{array}{l} \max g = v_1 + v_2 + v_3, \\ x_1 + 2x_2 - v_1 = 1, \\ x_1 - 2x_3 - v_2 = 7, \\ -x_1 + x_3 - v_3 = -4, \\ x_1 + x_2 + x_4 = 1, \\ x_2 + x_5 = 2, \\ x_1 - x_2 + x_3 + x_6 = 4, \\ -1 \leq x_1 \leq 1, \\ -2 \leq x_2 \leq 2, \\ -3 \leq x_3 \leq 3, \\ -4 \leq x_4 \leq 4, \\ -5 \leq x_5 \leq 5, \\ -6 \leq x_6 \leq 6, \\ v \geq 0. \end{array} \right.$$

On trouve  $\max g = 0$ . Par conséquent la solution  $x^{(2)}$  est efficace pour notre problème(7.23).

**2<sup>ème</sup> itération :** Soit  $\{x^{(2)}, J_B^{(1)}\}$  une solution de support réalisable.

- Calculer la matrice des estimations  $E$  :

$$E_N = \begin{pmatrix} -3 & -2 & -2 \\ -1 & 2 & 0 \\ 1 & -1 & 0 \end{pmatrix} \text{ et } E_B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

donc

$$E = \begin{pmatrix} -3 & 0 & -2 & 0 & 0 & -2 \\ -1 & 0 & 2 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \end{pmatrix}.$$

- Déterminer le critère  $k_0$  tel que :

$$\sum_{j \in J_N} |E_{k_0 j}| = \max \left( \sum_{j \in J_N} |E_{1j}|, \sum_{j \in J_N} |E_{2j}|, \sum_{j \in J_N} |E_{3j}| \right) = \sum_{j \in J_N} |E_{1j}| = 7,$$

donc  $k_0 = 1$ .

- On a  $\beta_1(x^{(2)}, J_B^{(1)}) = 12 > \epsilon_1$ , alors le point  $x^{(2)}$  n'est pas optimal pour l'objectif  $k_0 = 1$ .

- Calculer la direction d'amélioration  $l$  :

$$\begin{cases} l_N = \begin{pmatrix} l_1 \\ l_3 \\ l_6 \end{pmatrix} = \begin{pmatrix} 0 \\ 6 \\ 0 \end{pmatrix}, \\ l_B = \begin{pmatrix} l_2 \\ l_4 \\ l_5 \end{pmatrix} = \begin{pmatrix} 6 \\ -6 \\ -6 \end{pmatrix} \end{cases}$$

- Calculer le pas maximal

$$\theta_{j_0} = \min_{j \in J_B} \theta_j = \min(\theta_2, \theta_4, \theta_5) = \theta_2 = \frac{1}{3},$$

donc  $j_0 = 2$  et  $\theta^0 = \min(1, \theta_{j_0}) = \frac{1}{3}$ .

- Calculer  $x^{(3)}$  :

$$x^{(3)} = x^{(2)} + \theta^0 l = \begin{pmatrix} 1 \\ 2 \\ -1 \\ -2 \\ 0 \\ 6 \end{pmatrix}.$$

On peut remarquer que  $x^{(3)} = x^{(1)}$ , donc on déduit que  $x^{(1)}$  est optimal pour l'objectif  $k_0 = 1$ . Ainsi  $x^{(3)}$  est efficace pour le problème (7.23).

Soit  $J_B^{(2)} = \{1, 2, 3\}$  une base inexplorée.

**3<sup>ème</sup> itération :** Soit  $\{x^{(1)}, J_B^{(2)}\}$  une solution de support réalisable.

Calculer la matrice des estimations  $E$  :

$$E_N = \begin{pmatrix} 1 & 1 & 0 \\ 3 & -5 & -2 \\ -2 & 3 & 1 \end{pmatrix} \text{ et } E_B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

donc

$$E = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 3 & -5 & -2 \\ 0 & 0 & 0 & -2 & 3 & 1 \end{pmatrix}.$$

- Déterminer le critère  $k_0$  tel que :

$$\sum_{j \in J_N} |E_{k_0 j}| = \max \left( \sum_{j \in J_N} |E_{1j}|, \sum_{j \in J_N} |E_{2j}|, \sum_{j \in J_N} |E_{3j}| \right) = \sum_{j \in J_N} |E_{2j}| = 10,$$

alors  $k_0 = 2$ . Ainsi, on ne peut pas améliorer un autre objectif, alors on peut arrêter l'algorithme.

- Les solutions efficaces trouvées sont :

$$x^{(1)} = \begin{pmatrix} 1 \\ 0 \\ -3 \\ 0 \\ 2 \\ 6 \end{pmatrix} \quad \text{et} \quad x^{(2)} = \begin{pmatrix} 1 \\ 2 \\ -1 \\ -2 \\ 0 \\ 6 \end{pmatrix}.$$

## 7.9 Exemple sur la modélisation mathématique de la chaîne logistique

Pour procéder à l'application de la méthode adaptée, on va reprendre l'exemple étudié dans [6] qui est le cas pratique d'une entreprise algérienne "la laiterie Rio" dont les produits (les yaourts) sont considérés comme sensibles car leur durée de vie est très courte. Cette remarque fait que les prévisions des ventes se font à très court terme, ce qui permettra à la laiterie RIO de faire face à la demande des clients en y garantissant la qualité requise.

### 7.9.1 Présentation de la laiterie

La laiterie de Rio est une société à responsabilité limitée depuis juin 2004 avec un capital estimé à 500000DA. Elle a été créée en 1999 et revêt un caractère familial. Cette entreprise est versée dans la production de yaourts et emploie 26 travailleurs (11 hommes et 15 femmes). Elle a son siège dans la ville de Tlemcen.

### 7.9.2 Caractéristiques des produits de la laiterie

L'étude des caractéristiques des trois produits fabriqués par la laiterie Rio est une opération très importante, pour la prévision des ventes ainsi que pour le processus de modélisation des chaînes logistiques. En effet, il devient difficile de déterminer la méthode adéquate de prévision si on ne connaît pas la nature du produit et la période de prévision. En plus, le processus de modélisation ne peut se réaliser sans connaître les différents objectifs à atteindre et les conditions objectives ou contraintes imposées par les déterminants des produits tels que le temps nécessaire pour l'approvisionnement, la production et la distribution ainsi que la capacité dont dispose l'entreprise et qui limite le volume de production (machines, équipements, heures de travail déterminées par la main-d'oeuvre disponible dans l'entreprise etc...).

Il convient de noter également qu'il existe des caractéristiques communes à ces trois sortes de produits telles que les étapes de la production, la plupart de leurs

## 7.9 Exemple sur la modélisation mathématique de la chaîne logistique 120

composants, la période de validité ainsi que des caractéristiques différentes telles que la qualité, le coût de revient et le profit résultant de la vente de chaque unité de ces produits.

Après une étude détaillée des coûts d'achat des matières premières, de leur stockage, des coûts de production et de distribution des produits finis, des différentes étapes de la production du yaourt ainsi que sa durée de validité, on a dressé le tableau suivant :

	Yaourt aromatisé emballé dans des pots de plastique simple	Yaourt aromatisé emballé dans des pots TONIC	Yaourt brassé aux fruits emballé dans des pots TONIC
Durée de production de 2300 pots	25h et 45m	26h et 15m	27h et 15m
Prix de vente d'un pot en (DA)	36	40	52
Coût de revient d'un pot (DA)	30.68	33.55	43.62
Profit unitaire (DA)	5.32	6.45	8.38
Qualité du produit	acceptable	bonne	très bonne
Durée de validité	30 jours	30 jours	30 jours

TABLE 7.1 – Caractéristiques des produits de la laiterie Rio

Après l'analyse des ventes hebdomadaires des trois produits pour les années 2007 et 2008 et l'obtention des modèles de prévision pour chaque produit, on a calculé les ventes prévisibles de la première semaine de janvier 2009 comme suit :

Produits	Yaourt aromatisé emballé dans des pots de plastique simple	Yaourt aromatisé emballé dans des pots TONIC	Yaourt brassé aux fruits emballé dans des pots TONIC
Ventes hebdomadaires prévisibles	17459	4164	11177

TABLE 7.2 – Les ventes prévisibles de la première semaine de l'année 2009

Le problème, que rencontre la gestion de la chaîne logistique des produits de la laiterie, est représenté par la façon dont est déterminée la quantité produite hebdomadairement et à quel moment il faut produire pour réaliser les objectifs. La planification de la production se fait par semaine car la durée de vie des produits est courte. Le stockage de la production ne doit pas dépasser la semaine pour permettre aux détaillants d'écouler leur marchandise.

## 7.9 Exemple sur la modélisation mathématique de la chaîne logistique 121

Les objectifs sont les suivants :

1. maximisation du profit total ;
2. optimisation de la qualité des ventes ;
3. minimisation du coût total de la chaîne logistique.

La qualité a été évaluée comme suit :

- très bonne : 16/20,
- bonne : 14/20,
- acceptable : 12/20.

$$\begin{cases} \min z_1(x) = 43.62x_1 + 33.55x_2 + 30.68x_3 + 8220, \\ \max z_2(x) = 8.38x_1 + 6.45x_2 + 5.32x_3 - 820, \\ \max z_3(x) = 16x_1 + 14x_2 + 20x_3, \end{cases} \quad (7.24)$$

- $x_1$  : la quantité produite de yaourt brassé aux fruits emballé dans les pots TONIC.
- $x_2$  : la quantité produite de yaourt aromatisé emballé dans les pots TONIC.
- $x_3$  : la quantité produite de yaourt aromatisé emballé dans les pots de plastique simple.

Mais il existe plusieurs contraintes qui limitent la réalisation du niveau optimal de ces objectifs et qui sont représentées par :

- le volume de production qui ne doit pas dépasser le volume des ventes prévisibles pour la première semaine du mois de Janvier 2009 ;
- la contrainte de la capacité de production et les caractéristiques des produits ;
- le volume de production du yaourt brassé aux fruits doit dépasser 5000 pots, celui du yaourt aromatisé emballé dans des pots TONIC 2000 pots tandis que le volume de production du yaourt aromatisé emballé dans des pots de plastique simple doit, lui, dépasser 9000 pots ; ceci parce que la demande hebdomadaire des trois produits, durant les dernières semaines de l'année 2008, n'est pas descendue au-dessous de ces quantités ;
- la contrainte des heures de travail hebdomadaires disponibles : le volume horaire hebdomadaire disponible de la production a été estimé à environ 2160 minutes (pour le jeudi, samedi, dimanche et lundi), le calcul du temps consacré à la production d'une unité des trois produits donnant les résultats suivants :

$x_1$  : 0.124 mn ;  $x_2$  : 0.098 mn ;  $x_3$  : 0.085 mn.

$$\begin{cases} x_1 + x_2 + x_3 \leq 27600, \\ 0.124x_1 + 0.098x_2 + 0.085x_3 \leq 2160, \\ 5000 \leq x_1 \leq 11177, \\ 2000 \leq x_2 \leq 4164, \\ 9000 \leq x_3 \leq 17459. \end{cases} \quad (7.25)$$

### 7.9.3 Résolution du modèle à l'aide de la méthode adaptée du support

Pour résoudre ce problème considérons le programme suivant :

$$\left\{ \begin{array}{l} \max Z_1(x) = -43.62x_1 - 33.55x_2 - 30.68x_3, \\ \max Z_2(x) = 8.38x_1 + 6.45x_2 + 5.32x_3, \\ \max Z_3(x) = 16x_1 + 14x_2 + 20x_3, \\ x_1 + x_2 + x_3 + x_4 = 27600, \\ 0.124x_1 + 0.098x_2 + 0.085x_3 + x_5 = 2160, \\ 5000 \leq x_1 \leq 11177, \\ 2000 \leq x_2 \leq 4164, \\ 9000 \leq x_3 \leq 17459, \\ 0 \leq x_4 \leq M, \\ 0 \leq x_5 \leq M. \end{array} \right. \quad (7.26)$$

Avec  $Z_1(x) = -(z_1(x) + 8220)$ ,  $Z_2(x) = z_2(x) + 820$  et  $M$  un nombre réel positif aussi grand que l'on veut. On a  $x^0 = (6000, 3000, 10000, 8600, 272)$  est une solution réalisable.

#### I. Calcul du premier point efficace :

D'abord, on doit résoudre le programme (7.14), la solution optimale de ce programme est :

$$(0, 2.2959, 0, 0, 0, 0, 0, 0, 1.1456, 0, 0, 0, 1.7088, 0, 0, 0).$$

Pour obtenir notre premier point efficace résolvons le programme (7.15), la solution optimale de ce programme en utilisant la méthode adaptée du support est :  $(5612, 4164, 9224, 8600, 272)$ . Ainsi notre premier point efficace est :  $x^{(1)} = (7959.1, 4164, 9000, 6476.9, 0)$  et on a :

$$\left\{ \begin{array}{l} z_1(x^{(1)}) = 771213.78, \\ z_2(x^{(1)}) = 133214.22, \\ z_3(x^{(1)}) = 293640, \\ x_1 = 7959, \\ x_2 = 4164, \\ x_3 = 9000. \end{array} \right.$$

Les résultats obtenus peuvent être interprétés comme suit : la laiterie Rio doit produire 7959 pots de yaourt brassé aux fruits, 4164 pots de yaourt aromatisé emballés dans les pots TONIC ainsi que 9000 pots de yaourt aromatisé emballés dans les pots de plastique simple. Cette production nécessite un montant de 771213,78DA, représentant les coûts de gestion de la chaîne logistique. Elle réalise un profit optimal estimé à 133214,22 DA et une meilleure qualité des ventes.

## II. Recherche de tous les points efficaces associés à notre premier point efficace :

En posant  $J_B = \{2, 4\}$  et  $J_N = \{1, 3, 5\}$ , on obtient le point :  
 $x^{(2)} = (5000, 2000, 9000, 11600, 579)$ , et on a :

$$\begin{cases} z_1(x^{(2)}) = 553100, \\ z_2(x^{(2)}) = 103500, \\ z_3(x^{(2)}) = 288820, \\ x_1 = 5000, \\ x_2 = 2000, \\ x_3 = 9000. \end{cases}$$

Les résultats obtenus peuvent être interprétés comme suit : la laiterie Rio doit produire 5000 pots de yaourt brassé aux fruits, 2000 pots de yaourt aromatisé emballés dans les pots TONIC ainsi que 9000 pots de yaourt aromatisé emballés dans les pots de plastique simple. Cette production nécessite un montant de 553100 DA, représentant les coûts de gestion de la chaîne logistique. Elle réalise un profit optimal estimé à 103500 DA et une meilleure qualité des ventes.

*Remarque 7.9.1.* En changeant de support on peut trouver un autre point efficace.

## 7.10 Conclusion

Ce chapitre est consacré à développer une nouvelle méthode pour résoudre un programme linéaire multiobjectif à variables bornées en utilisant la méthode adaptée du support, basée sur l'utilisation d'une métrique adaptée. La méthode construite génère un grand gain en temps et en espace mémoire. Ceci est soutenu par le fait que nous utilisons la méthode adaptée pour résoudre nos programmes linéaires mono-objectif, et il a été montré que cette méthode est très efficace, spécialement dans le cas de problèmes dégénérés. Il convient de noter que l'utilisation de la méthode du simplexe est inappropriée pour ce type de problèmes.

D'abord, nous avons introduit une nouvelle procédure pour trouver notre premier point efficace. Par la suite, nous avons développé une procédure détaillée pour calculer les solutions efficaces et les solutions faiblement efficaces. Nous avons également caractérisé les solutions  $\epsilon$ -efficace pour un programme linéaire multiobjectif. Ainsi, nous avons exploité le critère de suboptimalité de la méthode adaptée dans la programmation mono-objectif pour obtenir les solutions  $\epsilon$ -efficaces et les solutions  $\epsilon$ -faiblement efficace de notre problème. De plus, nous avons proposé un algorithme global pour décrire les étapes majeures dans la mise en œuvre de la méthode. Enfin, nous avons présenté deux exemples illustrant l'applicabilité de la méthode.



# Conclusion générale

Cette thèse de recherche concerne la minimisation des polynômes par une nouvelle méthode dite méthode adaptée du support.

Notre objectif a été de montrer que la méthode adaptée du support peut être utilisée efficacement dans la résolution de programmes semi-définis positifs, et en particulier pour résoudre exactement des problèmes linéaires et des problèmes quadratiques convexes. L'excellence des résultats obtenues par l'utilisation de cette approche ont pu être démontrées dans le chapitre 4. De ce fait, nous avons pu concevoir une nouvelle méthode pour résoudre un programme linéaire multiobjectif dont les variables de décision sont bornées.

Pour cela, on a commencé par rappeler les notions fondamentales de l'optimisation semi-définie positive, ensuite on a défini quelques théorèmes et propriétés concernant l'optimisation quadratique convexe pour la minimisation des polynômes homogènes de degré 2 avec des contraintes linéaires, on a aussi proposé la méthode adaptée pour la résolution des programmes linéaires mono-objectif avec des variables non négatives. Enfin, on a présenté les différents concepts utilisés en optimisation linéaire multiobjectif et nous avons étudié un programme linéaire multiobjectif à variables bornées, et ce en utilisant la méthode adaptée du support. Ainsi, nous avons proposé une nouvelle procédure de recherche d'un point efficace initial, un nouveau test d'efficacité d'une variable non basique et une nouvelle procédure de recherche de tous les points efficaces associés à notre premier point efficace, et ça en s'inspirant du travail développé par S.Radjef et M.O.Bibi [82, 70]. L'algorithme nous permet aussi de trouver les points  $\epsilon$ -efficaces, et ce en exploitant le critère de suboptimalité de la méthode adaptée en mono-objectif.

Notons que la méthode adaptée du support a des propriétés intéressantes. En effet :

- Elle est innovante dans la mesure où elle essaye de s'attaquer à des situations multiobjectif pour lesquelles les méthodes de résolutions sont absentes ou artificielles. Ces situations englobent les problèmes d'optimisation en présence de nombreux critères.
- Elle est axiomatisée. D'une part, elle est fondée sur des concepts tels que

l'efficacité et l' $\epsilon$ -efficacité qui sont motivés et justifiés par des interprétations intelligibles. D'autre part, l'analyse et les choix effectués pendant la résolution reposent sur des principes qui peuvent sembler raisonnables.

- Elle est une extension d'une ancienne approche qui a déjà montré ses preuves comme nous avons pu le voir dans le quatrième chapitre.
- Elle est évolutive puisque ces fondements peuvent être facilement modifiés pour satisfaire au mieux d'autre programme, comme nous avons pu le voir dans le cas de la minimisation d'un polynômes homogènes de degré 2, pour convenir à la description de situations plus concrètes.

Dans l'état actuel de nos travaux, la réalisation de cette approche n'est pas achevée. Néanmoins les simulations numériques que nous avons effectuées montrent l'efficacité de cette méthode.

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

- L'extension de cette méthode pour un programme linéaire multiobjectif à variables hybrides.
- L'extension de cette dernière pour la minimisation de polynôme homogène de degré  $n \geq 2$ .
- Enfin, sur le plan général, d'étudier les possibilités d'élargir les concepts proposés au cadre de l'optimisation semi-défini multiobjectif.

Pour conclure, nous mentionnons qu'actuellement notre approche n'est encore qu'à un état embryonnaire. Des améliorations tant sur le plan théorique que sur le plan pratique sont prévues pour que l'on puisse utiliser cette méthode d'une manière opérationnelle.

# Bibliographie

- [1] N. Abassi. Méthode de support pour la minimisation d'une fonctionnelle quadratique convexe. Master's thesis, Université A. Mira, Béjaia, 2004.
- [2] S. Achmanov. *Programmation linéaire*. Edition Mir, Moscou, 1984.
- [3] K. Ait-Yahia and F. Benkerrou. *Méthodes multicritères d'aide à la planification de la production au niveau de la laiterie de Djurdjura. Mémoire d'ingénieur*. Université de Béjaïa, 1998.
- [4] P. Armand and C. Malivert. Determination of the efficient set in multiobjective linear programming. *Journal of optimization theory and applications*, 70(3) :467–490, 1991.
- [5] E. M. L. Beale. On quadratic programming. *Naval Research Logistics Quarterly*, 6(3) :227–243, 1959.
- [6] Mostefa Belmokaddem and Omar Benatek. Prevision des ventes et efficacite des chaines logistiques-essai de modelisation. *les cahiers du cread*, 92 :5–24, 2010.
- [7] H.P. Benson. Finding an initial efficient extreme point for a linear multiple objective program. *Journal of Operational Research Society*, 32(6) :495–498, 1981.
- [8] M. Bentobache. Nouvelle méthode pour la résolution des problèmes de programmation linéaire sous forme canonique et à variables bornées. Master's thesis, Université de Béjaia, 2005.
- [9] A.B. Berkelaar. Complementary and interior point methodology. Master's thesis, Delft University of Technology, 1995.
- [10] A.B. Berkelaar, Benjamin Jansen, Kees Roos, and Tamas Terlaky. *An interior point approach to parametric convex quadratic programming*. Erasmus University Rotterdam, The Netherlands, 1997.
- [11] M.O. Bibi. *Methods for solving linear-quadratic problems of optimal control*. PhD thesis, University of Minsk, 1985.
- [12] M.O. Bibi. Support method for solving a linear-quadratic problem with polyhedral constraints on control. *Optimization*, 37 :139–147, 1996.

- [13] M.O. Bibi. *Cours de Post-graduation sur la méthode adaptée en programmation linéaire*. Université de Béjaïa, 1998.
- [14] Mohand Ouamer Bibi and Mohand Bentobache. A hybrid direction algorithm for solving linear programs. *International Journal of Computer Mathematics*, 92(1) :201–216, 2015.
- [15] O. Bouarab. Une contribution sur la programmation linéaire multiobjectifs floue. Master’s thesis, Université de Tizi Ouzou, 1992.
- [16] B. Brahmi and M.O Bibi. Dual support method for solving convex quadratic programs. *Optimization*, 59(6) :851–872, 2010.
- [17] Y. Collette and P. Siarry. *Optimisation multiobjectif*. Editions Eyrolles, Paris, 2002.
- [18] Yann Collette. *Contribution à l’évaluation et au perfectionnement des méthodes d’optimisation multiobjectif : application à l’optimisation des plans de rechargement de combustible nucléaire*. PhD thesis, Université Paris-Est Créteil Val de Marne (UPEC), 2002.
- [19] G.B. Dantzig. *Applications et prolongements de la programmation linéaire*. Dunod, Paris, 1966.
- [20] Zohra Sabrina Delhoum and Sonia Radjef. Solution of a linear multiobjective problem by the adaptive method. *International Conference on Analysis and Its Applications*, 12-15 July 2016.
- [21] Zohra Sabrina Delhoum and Sonia Radjef. The adaptive method to solve linear multiobjective programming problems with bounded variables. *The 11th International Conferences on Multiple Objective Programming and Goal Programming*, December 2015.
- [22] Zohra Sabrina Delhoum, Sonia Radjef, and Fatima Boudaoud. Generation of efficient and  $\epsilon$ -efficient solutions in multiple objective linear programming. *Turkish Journal of Mathematics*, 2017.
- [23] Zohra Sabrina Delhoum, Sonia Radjef, and Fatima Boudaoud. An adaptive method for standard linear programs. *soumis pour publication*, 2018.
- [24] Sonia Radjef Douar. cours recherche opérationnelle 2, master 2. *Université USTO MB*, 2014.
- [25] J.G. Ecker and N.S. Hegner. On computing an initial efficient extreme point. *Journal of Operational Research Society*, 29 :1005–1007, 1978.
- [26] J.G. Ecker and N.S. Hegner. Generating the efficient set for multiple objective linear programs. ARO Report 79-1, Transactions of the twenty-fourth conference of army mathematicians, 1979.

- [27] J.G. Ecker, N.S. Hegner, and I.A. Kouada. Generating all maximal efficient faces for multiple objective linear programs. *Journal of Optimization Theory and Applications*, 30(3) :353–381, March 1980.
- [28] J.G. Ecker and I.A. Kouada. Finding efficient points for linear multiobjective problem. *Mathematical Programming*, 8 :375–377, 1975.
- [29] J.G. Ecker and I.A. Kouada. Finding all efficient extreme points for multiple objective linear programs. *Mathematical Programming*, 14 :249–261, 1978.
- [30] J.G. Ecker and J.H. Song. Optimizing a linear function over an efficient set. *Journal of Optimization Theory and Applications*, 83(3) :541–563, 1994.
- [31] F.Y. Edgeworth. *Mathematical psychics*. University Microfilms International (Out of print), 1987 (the original edition in 1881).
- [32] R.R. Egudo and M.A. Hanson. Multiobjective duality with invexity. *Journal of mathematical analysis and applications*, 126 :469–477, 1987.
- [33] Matthias Ehrgott. *Multicriteria Optimization*. Springer, 2005.
- [34] Matthias Ehrgott. *Multicriteria optimization*. Springer Science & Business Media, 2006.
- [35] Z.S. Delhoum et S. Radjef. Thèse de master : Sur la programmation linéaire multiobjectif. 2015.
- [36] J.P. Evans and R.E. Steuer. Generating efficient extreme points in linear multiple objective programming : Two algorithms and computing experience. In *Multiple Criteria decision making*. J.L. Cochrane and M. Zeleny, University of South Carolina Press, Columbia, 349-365, 1973.
- [37] J.P. Evans and R.E. Steuer. A revised simplex method for linear multiple objective programs. *Mathematical Programming*, 5(1) :54–72, 1973.
- [38] A. Faradji. Algorithmes de minimisation d’une fonctionnelle quadratique. Master’s thesis, Université de Tizi Ouzou, 1998.
- [39] J. Fliege and A. Heseleer. Constructing approximations to the efficient set of convex quadratic multiobjective problems. *University Dortmund, Germany*, 2002.
- [40] R. Gabasov. *Adaptive method of linear programming*. Preprints of the university of Karlsruhe, Germany, 1993.
- [41] R. Gabasov et al. *Constructive methods of optimization*. P.I.-University Press, Minsk, 1984.
- [42] R. Gabasov et al. Solution of linear quadratic extremal problems. *Soviet Math. Dokl.*, 31 :99–103, 1985.
- [43] R. Gabasov, F.M. Kirillova, and V.M. Rakestsky. *Constructive methods of optimization. P.IV : Convex problems*. University Press, Minsk, 1987.

- [44] T. Gal. A general method for determining the set of all efficient solutions to a linear vector-maximum problem. *European Journal of Operational Research*, 1(5) :306–322, 1977.
- [45] A.G. Hadigheh, O. Ramanko, and T. Terlaky. Sensitivity analysis in convex quadratic optimization. *Advanced optimization laboratory, Hamilton, Canada*, 2003.
- [46] Christoph Heermann, Bernd Weyers, and Jörg Fliege. A new adaptive algorithm for convex quadratic multicriteria optimization. *University Dortmund, Germany*, 2004.
- [47] C. Hillermeier. *Nonlinear Multiobjective optimization*. Birkhäuser Verlag, 2001.
- [48] H. Isermann. The enumeration of the set of all efficient solutions for a linear multiple objective program. *Journal of Operational Research*, 28(3) :711–725, 1977.
- [49] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4 :374–395, 1984.
- [50] R. L. Keeney and H. Raiffa. *Decisions with multiple objectives : preferences and value tradeoff*. Cambridge University Press, Cambridge.
- [51] E.A. Kostina and O.I. Kostyuhova. An algorithm for solving quadratic programming problems with linear equality and inequality constraints. *Computational mathematics and mathematical physics*, 41(7) :960–973, 2001.
- [52] E.A. Kostina and O.I. Kostyuhova. *A primal-dual active set method for convex quadratic programming problems*. Preprints of the university of Karlsruhe, Germany, 2003.
- [53] Z. Laouabdia-Sellami. Quelques aspects de la programmation de but linéaire et ses applications. Master’s thesis, Université de Annaba, 1991.
- [54] Richard Leroy. *Certificats de positivité et minimisation polynomiale dans la base de Bernstein multivariée*. PhD thesis, Université Rennes 1, 2008.
- [55] D.T. Luc. Connectedness of the efficient point sets in quasiconcave vector maximization. *Journal of mathematical analysis and applications*, 122 :346–354, 1987.
- [56] Bezoui Madani. Méthode adaptée de programmation quadratique convexe : Théorie et applications, 2011.
- [57] Agnès Maher. *Programmation semi-définie positive. Méthodes et algorithmes pour le management d’énergie*. PhD thesis, Paris 11, 2013.
- [58] R. Malki and B. Nezreg. *Implémentation d’un logiciel pour la résolution d’un problème de programmation convexe par la méthode adaptée, Mémoire d’ingéniorat*. Institut d’informatique, Université de Tizi Ouzou, 1999.

- [59] F. Margot. *Composition de polytopes combinatoires : une approche par projection*. Presses polytechniques et universitaires romandes, 1995.
- [60] K. Miettinen. *On the methodology of multiobjective optimization with applications*. PhD thesis, University of Jyväskylä, Finland, 1994.
- [61] K. Miettinen. *Nonlinear multiobjective optimization*. Kluwer Academic Publishers, Boston, 1999.
- [62] M. Minoux. *Programmation mathématique : Théorie et algorithmes, Tome 1, 2*. Collection technique et scientifique des télécommunications, 1983.
- [63] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, 1999.
- [64] W. Pareto. Economie mathématique. Dans l'"Encyclopédie des sciences mathématiques", sous la direction de J. Molk, Tome. 1, Vol. 4, fascicule 4.
- [65] Pablo A Parrilo and Bernd Sturmfels. Minimizing polynomial functions. *Algorithmic and quantitative real algebraic geometry, DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 60 :83–99, 2003.
- [66] J. Philip. Algorithms for the vector maximization problem. *Mathematical Programming*, 2 :207–229, 1972.
- [67] M. Prevot and S. Holin. Programmation linéaire multiobjectifs. Technical Report 31, Collection de l'I.M.E., Dijon, France, 1986.
- [68] M.S. Radjef. *Cours de Post-graduation sur les notions d'optimalité en optimisation multicritère*. Université de Béjaïa, 1998.
- [69] S. Radjef. *Thèse de Magister : Sur la programmation linéaire multiobjectifs*. 2001.
- [70] S. Radjef. *Thèse de Doctorat : Application de la méthode adaptée aux problèmes multicritères*. 2011.
- [71] S. Radjef. The direct support method to solve a multiobjective linear program. *The 23rd European Conference on operational research, EURO 2009*, July 5-8, 2009, Bonn, Germany, Book of abstracts, Page 270.
- [72] S. Radjef and M.O Bibi. About the direct support method in linear programming. *10ème Congrès de la société française de recherche opérationnelle et d'aide à la décision, ROADEF 2009*, 2009, France, Livre des résumés, Pages 382-383.
- [73] S. Radjef and M.O Bibi. An effective generalization of the direct support method. *Mathematical Problems of Engineering*, 2011 :Article ID 374390, 18 pages, doi :10.1155/2011/374390, 2011.
- [74] S. Radjef and M.O Bibi. La méthode directe du support pour la résolution d'un programme linéaire à variables mixtes. *Actes des quatrièmes journées francophones sur la recherche opérationnelle, Francoro 4, Suisse*, August 18-21, 2004.

- [75] S. Radjef and M.O Bibi. Solution of a convex quadratic program via the direct support method. *Proceedings COSI 2007, Université USTO, Oran*, June 11-13, 2007.
- [76] S. Radjef and M.O Bibi. Solution of a convex quadratic program with mixed variables by the direct support method. *Proceedings of the international conference MOAD'2007, ISBN :978-9947-0-1958-0*, November, 2007, Béjaïa, Algérie.
- [77] S. Radjef and M.O Bibi. An effective generalization of the direct support method. *Actes du colloque international Modélisation Stochastique et Statistique, MSS'10, Université USTHB, Alger*, November 21-23, 2010.
- [78] S. Radjef and M.O Bibi. A new algorithm for linear programming problems. *Proceedings of ORP 3 Meeting, Guimaraes, Portugal*, September 12-15, 2007.
- [79] S. Radjef, M.O Bibi, and M.S Radjef. Résolution d'un programme linéaire multi-objectifs à variables bornées par la méthode directe du support. *Proceeding COSI 2004, Université de Tizi Ouzou*, August 18-21, 2004.
- [80] S. Radjef and M. Hamga. *Etude comparative entre deux méthodes de résolution d'un programme quadratique convexe, Mémoire d'ingénieur*. Université de Béjaïa, 1998.
- [81] Sonia Radjef and M.O Bibi. A new algorithm for linear multiobjective programming problems with bounded variables. *Arabian Journal of Mathematics*.
- [82] Sonia Radjef and M.O Bibi. A new algorithm for linear multiobjective programming problems with bounded variables. *Arabian Journal of Mathematics*, DOI 10.1007/s40065-013-0094-x.
- [83] Sonia Radjef and Mohand Ouamer Bibi. The direct support method to solve generalized linear multiobjective problems. *Journal of Advanced Research in Applied Mathematics*, Vol.5, 2013.
- [84] Sonia Radjef and Mohand Ouamer Bibi. The direct support method to solve generalized linear multiobjective problems. *Journal of Advanced Research in Applied Mathematics (JARAM)*, Vol. 5, Issue. 4, pp. 95-111, doi : 10.5373/jaram.1576.100812., 2013.
- [85] R.T. Rockafellar. *Convex analysis*. Princeton University Press, 1970.
- [86] Frédéric Roupin. *Algorithmes Combinatoires et Relaxations par Programmation Linéaire et Semidéfinie. Application à la Résolution de Problèmes Quadratiques et d'Optimisation dans les Graphes*. PhD thesis, Université Paris-Nord-Paris XIII, 2006.
- [87] B. Roy and D. Bouyssou. *Aide multicritère à la décision : Méthodes et cas*. Editions Economica, 1993.



- [88] Y. Sawaragi, H. Nakayama, and T. Tanino. *Theory of multiobjective optimization*. Academic Press, Inc., r. bellman edition, 1985.
- [89] L. Seiford and P.L. Yu. Potential solutions of linear systems : the multi-criteria multiple constraint levels program. *Journal of mathematical analysis and applications*, 69(2) :283–303, 1979.
- [90] M. Simmonnard. *Programmation linéaire. Techniques du calcul économique*. 2ème Edition Dunod, Paris, 1973.
- [91] R.E. Steuer. *Multiple criteria optimization : Theory, Computation, and Application*. John Wiley and Sons, New York, 1986.
- [92] R.E. Steuer. The adbase multiple objective linear programming package. Technical report, University of Georgia, 1996.
- [93] L.W.G. Strijbosch. A simplified M.O.L.P algorithm. *Computers*, 18(8) :709–716, 1991.
- [94] S.J Sugden. *A Class of Direct Search Methods for Nonlinear Integer Programming*. PhD thesis, Bond University, School of Information and Computing Sciences, 1992.
- [95] Wenyu Sun and Ya-Xiang Yuan. *Optimization Theory and methods : Non-linear programming*. Springer, 2006.
- [96] E.-G. Talbi. *A Taxonomy of Hybrid Metaheuristics*. Rapport technique TR-183, LIFL, Université de Lille 1, 1998.
- [97] A. B. Touati. Problèmes d’optimisation multicritère. Master’s thesis, Université de Annaba, 1998.
- [98] Thomas Vincent. *Caractérisation des solutions efficaces et algorithmes d’énumération exacts pour l’optimisation multiobjectif en variables mixtes binaires*. 2003.
- [99] P. Vincke. *L’aide multicritère à la décision*. Editions de l’université de Bruxelles, Ellipses, Belgique, 1989.
- [100] D. De Werra and al. *Recherche opérationnelle pour ingénieur I*. Press Polytechniques et universitaires romandes, Lausanne, 2003.
- [101] P. Wolfe. The simplex method for quadratic programming. *Econometrica*, 27 :382–398, 1959.
- [102] Y. Ye and E. Tse. An extension of karmarkar’s algorithm to convex quadratic programming. *Math. Prog.*, 44 :157–179, 1989.
- [103] E.A. Yildirim and S.J. Wright. Warm-start strategies in interior-point methods for linear programming. *Technical report 1258, School of Operations research and industrial engineering, Cornell University*, 2000.

- 
- [104] P.L. Yu. *Multiple criteria decision making : five basic concepts*, chapter X, pages 662–697. Elsevier Science, North Holland, 1989.
- [105] P.L. Yu and M. Zeleny. The set of all non-dominated solutions in linear cases and a multicriteria simplex method. *Journal of mathematical analysis and applications*, 49(2) :430–468, 1975.
- [106] M. Zeleny. *Linear multiobjective programming*. Springer-Verlag, Berlin, 1974.

# Annexe

## A Introduction

Afin d'effectuer une simulation numérique de la méthode adaptée multiobjectif, une implémentation sous la version Matlab 8.1.0.604 a été développée. La simulation est basée sur des problèmes de programmation linéaire à variables bornées générés aléatoirement. La caractéristique de ces problèmes est que la solution optimale  $x$ , et la matrice des fonctions objectif  $C$  sont connus auparavant.

Dans notre implémentation, les paramètres d'entrée de la procédure qui génèrent des problèmes linéaires aléatoire sont  $n$ ,  $m$  et  $p$ , respectivement, ils représentent le nombre de variables, le nombre de contraintes et le nombre des fonctions objectifs.

## B Utilisation du langage du logiciel Matlab

Les critères qui nous ont poussé à choisir le langage du logiciel Matlab sont les suivants :

- Matlab est un logiciel parfaitement dédié à la résolution de problèmes d'analyse numérique ou d'optimisation. En effet, il permet d'effectuer des calculs matriciels ou de visualiser les résultats sous forme graphique.
- Il possède une grande variété de méthodes scientifiques prédéfinies.
- Il ne nécessite ni dimensionnement ni déclaration de type. Contrairement aux langages de programmation classiques, les fonctions du Matlab permettent de manipuler directement et interactivement ces données matricielles.

## C Implémentation de la méthode adaptée multiobjectif sous Matlab

### C.1 Génération des programmes aléatoires

Pour générer des programmes aléatoires, nous avons utilisé la fonction "randint" prédefinie sous Matlab.

## C Implémentation de la méthode adaptée multiobjectif sous Matlab135

**Entrée :**  $n, m, p$ ;

$n$  : Le nombre des variables,

$m$  : Le nombre de contraintes,

$p$  : Le nombre de critère à optimiser,

**Sortie :**  $C, A, d_1, d_2, x^0, b$ ;

$C$  : une matrice d'ordre  $n \times p$ ,

$A$  : une matrice d'ordre  $m \times n$ ,

$b$  : un vecteur d'ordre  $m$ ,

$d_1$  : un vecteur d'ordre  $n$ ,

$d_2$  : un vecteur d'ordre  $n$ ,

$x^0$  : un vecteur d'ordre  $n$ .

### C.2 Méthode de Benson

**Entrée :**  $C, A, d_1, d_2, x^0, b$ ,

Effectuer le changement de variables suivant :

$c1 = -\text{horzcat}((-C*x-C*d1)', (b-A*d1)', -(b-A*d1)', \text{zeros}(n,1)', (d2-d1)')'$ ;

$b1 = C'*\text{ones}(k,1)$ ;

$A1 = [\text{horzcat}(C', -A', A', \text{eye}(n,n)', -\text{eye}(n,n))]$ ;

Résoudre le programme suivant :

$$\begin{cases} \max(c^1)^T x, \\ A^1 x = b^1, \\ x \geq 0. \end{cases} \quad (7.27)$$

en ayant recours au solveur linprog, prédéfini sous Matlab et cela en utilisant le code suivant :

```
options = optimoptions('linprog','Algorithm','simplex');
```

```
[x,fval,exitflag,output] = linprog(-c,A,b,Aeq,beq,lb,ub,x0,options);
```

Poser

```
x=x(1:k)';
```

```
c=((x(1:k)+ones(k,1))'*C)';
```

```
ub=d2;
```

```
lb=d1;
```

```
Aeq=A;
```

```
beq=b;
```

```
A=[];
```

```
b=[];
```

## C Implémentation de la méthode adaptée multiobjectif sous Matlab136

Résoudre le programme suivant :

$$\begin{cases} \max c^T x, \\ Ax = b, \\ d_1 \leq x \leq d_2. \end{cases} \quad (7.28)$$

en utilisant le solveur linprog pour la résolution d'un programme linéaire à variables bornée.

**Sortie :**  $x^1$ ,

$x^1$  : la solution initiale de notre programme multiobjectif,

### C.3 Choix du critère $k_0$

Choisir le critère à optimiser  $k_0$  et ce en créant une fonction matlab :

**Entrée :**  $C, A, J_B$ ,

$J_B$  un vecteur d'ordre  $m$  ;

**Sortie :**  $k_0$ .

### C.4 Génération d'une solution $\epsilon$ -efficace

Générer une solution efficace en créant un fonction matlab qu'on appellera "soleff" :

**Entrée :**  $C, A, b, J_B, \epsilon, d_1, d_2, k_0$  ;

$\epsilon$  : un vecteur d'ordre  $n$ ,

**Sortie :**  $x, Z(x), \beta(x, J_B)$ . Cette fonction intègre les sous-fonctions suivante :

- la fonction "checking-inputs" pour vérifier si les dimensions des matrices et des vecteurs introduit dans la fonction "soleff" sont correctes :

**Entrée :**  $C, A, b, x, d1, d2, JB, \epsilon$  ;

**Sortie :**  $n$ .

- la fonction "suboptimality" pour calculer la valeur de suboptimalité :

**Entrée :**  $x, E, J_N, d_1, d_2, k_0$  ;

**Sortie :**  $\beta(x, J_B)$ .

- la fonction "finding-theta0" pour trouver le pas  $\theta^0$  et l'indice  $j_0$  :

**Entrée :**  $J_B, x, d_1, d_2, L, J_N$  ;

**Sortie :**  $\theta^0, j_0$ .

- la fonction "suboptimality0" pour calculer la valeur de suboptimalité du nouveau plan  $\bar{x}$  :

**Entrée :**  $x, J_B$  ;

**Sortie :**  $\beta(\bar{x}, J_B)$ .

- la fonction "change-support" pour calculer le nouveau support  $\bar{J}_B$  :

**Entrée :**  $J_B, J_N, j_0, j_1$  ;

**Sortie :**  $J_B$ .

## C Implémentation de la méthode adaptée multiobjectif sous Matlab137

---

- la fonction "finding-j1" pour calculer  $\sigma_0$  et l'indice  $j_1$  :  
**Entrée** :  $L, E, J_N, J_B, x, A, d_1, d_2, j_0, k_0$  ;  
**Sortie** :  $j_1, t, \sigma_0$ .
- la fonction "suboptimality1" pour calculer la valeur de suboptimalité  $\beta(\bar{x}, \bar{J}_B)$  :  
**Entrée** :  $x, J_N, \theta^0, t, \sigma_0, \beta(x, J_B), E, k_0$  ;  
**Sortie** :  $\beta(\bar{x}, \bar{J}_B)$ .

### C.5 Test d'efficacité de la solution générée

Tester l'efficacité des solutions générées tout au long de la fonction "soleff", et ce en utilisant la sous-fonction "testeff" :

- **Entrée** :  $C, A, b, d_1, d_2, x$ ,  
 $x$  étant la solution générée,
- **Sortie** :  $g_{max}$ .
  - Si  $g_{max} = 0$ , la solution  $x$  est efficace,
  - sinon la solution  $x$  n'est pas efficace, aller à l'étape suivante.

## Generation of efficient and $\epsilon$ -efficient solutions in multiple objective linear programming

Zohra Sabrina DELHOUM<sup>1</sup>, Sonia RADJEF<sup>2,3,\*</sup>, Fatima BOUDAUD<sup>1</sup>

<sup>1</sup>Department of Mathematics, Faculty of Exact and Applied Sciences, Mathematics and Applications Laboratory, University of Oran 1, Ahmed Ben Bella, Oran, Algeria

<sup>2</sup>Department of Mathematics, Faculty of Mathematics and Computer Science, Oran University of Science and Technology - Mohamed Boudiaf (USTO-MB), El M'naouer, Oran, Algeria

<sup>3</sup>Lamos Research Unit, University of Bejaia, Bejaia, Algeria

Received: 04.02.2017

Accepted/Published Online: 25.09.2017

Final Version: 08.05.2018

**Abstract:** We develop an algorithm to solve a multiple objective linear programming problem with bounded variables. It is based on the scalarization theorem of optimal solutions of multiobjective linear programs and the single objective adaptive method. We suggest a process for the search for the first efficient solution without having to calculate a feasible solution, and we elaborate a method to generate efficient solutions, weakly efficient solutions, and  $\epsilon$ -efficient solutions. Supporting theoretical results are established and the method is demonstrated on a numerical example.

**Key words:** Multiobjective linear program, bounded variables,  $\epsilon$ -optimality criterion, adaptive method, efficiency, weak efficiency,  $\epsilon$ -efficiency

### 1. Introduction

Multiobjective programming has applications in many academic areas such as in the fields of engineering, economics, mining, life sciences, and finance. It has been a topic of research since the 1960s. However, while this type of optimization has great interest, the simplex algorithm has long been the most exact method used to solve multiobjective linear programming problems [7]. More recently, algorithms to solve these types of programs in objective space have been developed [2]. Since the discovery of interior point algorithms to solve linear programs in polynomial time, efforts to apply such methods to deal with multiobjective programming are evident. Another focus of interest is based on scalarization. Duality theory for multiobjective linear programming has been also studied.

On the other hand, single linear programming with bounded variables models many problems in real situations and major effort has been made in solving such problems. It is also clear that the multiple objective paradigm appears naturally in this context. The study of the combination of these two areas of optimization is therefore relevant; it is the main objective of this paper.

In parallel, in the 1980s, Loridan [12] introduced the notion of  $\epsilon$ -efficient solutions for multiobjective programs. White [16] then proposed several concepts of approximate solutions for these programs and drafted methods for their generation. In the last decades,  $\epsilon$ -efficient solutions have been examined in the literature by

\*Correspondence: soniaradjef@yahoo.fr

2000 AMS Mathematics Subject Classification: 90C05, 65K05, 90C29

many authors. This paper deals with the generation of  $\epsilon$ -efficient solutions in the case of multiobjective linear programming with bounded variables.

Following the preceding work in [15], a new method is proposed. The authors develop a method to solve a multiple objective linear programming problem with upper and lower bounded decision variables. The method is an extension of the direct support method [14, 15], known in single objective linear programming. This method is considered as an intermediate method between the interior methods and the simplex one. Indeed, using the direct support method, the initial feasible solution can be an extreme point, an interior point, or any point on the edge. This method [11] is usually used to solve linear programs with bounded variables.

In [15], the authors suggested an efficiency test of a nonbasic variable and a new process to search the initial efficient extreme point. Using the direct support method principle, they prescribed an algorithm to generate all the efficient extreme points, the  $\epsilon$ -efficient extreme points, and the  $\epsilon$ -weakly efficient extreme points of the problem.

The authors, in [15], used the simplex metric when solving mono-objective programs.

In this work, we use the principle of the adaptive method [10], which is considered more general than the direct support method in single objective linear programming. Indeed, we suggest to use another metric called the adapted metric, where we consider all suboptimal indexes by which we build improvement of the objective function and the maximum step along this direction. This method avoids the preliminary transformation of the constraints and it enables us to treat the problem as it stands without making any changes. It manipulates the bounds as they are initially expressed. It is easy to use and it generates a large benefit in time and memory space. Comparisons with algorithms well known in the field of single objective programming [10, 11] have shown that this method is more efficient than the other approaches in this context.

We also exploit the suboptimality criterion of this adaptive method to find  $\epsilon$ -efficient solutions of our multiple objective problem.

Another contribution of our work is that we propose a new process to search for the first efficient solution without calculating a feasible solution.

The article is structured as follows. First, we state our purpose and some definitions. The basic concepts of efficiency, of  $\epsilon$ -efficiency, and their proprieties are established in Section 3 and Section 4. In Section 5 we give a process to find a first efficient solution. Section 6 deals with the development of a method to generate the efficient solutions. The detailed algorithm is given in Section 7 and a numerical example to demonstrate the applicability of the suggested method is given in Section 8. Finally, we give a conclusion in Section 9.

## 2. Basic terminology and problem formulation

We consider the following multiobjective linear programming problem with bounded variables:

$$\begin{cases} \max Z(x) = Cx, \\ x \in S. \end{cases} \quad (2.1)$$

The feasible set is defined as follows:

$$S = \{x \in \mathbb{R}^n : Ax = b, d_1 \leq x \leq d_2\}. \quad (2.2)$$

The indices of constraints and decision variables are respectively denoted by:

$I = \{1, 2, \dots, m\}$ ,  $J = \{1, 2, \dots, n\}$ , such that:

$J = J_B \cup J_N$  with  $J_B \cap J_N = \phi$ ,  $|J_B| = m$ .



Thus,  $x = x(J) = (x_j, j \in J)$ ,  $d_1 = d_1(J) = (d_{1j}, j \in J)$ , and  $d_2 = d_2(J) = (d_{2j}, j \in J)$  are  $n$ -vectors;  $b = (b_i, i \in I)$  is an  $m$ -vector; and  $A = A(I, J)$  is an  $m \times n$ -matrix, such that  $\text{rang} A = m < n$ .

We define the criterion function as follows:

$$Z(x) = \begin{pmatrix} z_1(x) \\ z_2(x) \\ \vdots \\ z_p(x) \end{pmatrix} = \begin{pmatrix} c_1^T x \\ c_2^T x \\ \vdots \\ c_p^T x \end{pmatrix} = Cx, \tag{2.3}$$

where  $C = C(K, J)$  with  $K = \{1, 2, \dots, p\}$  is a  $p \times n$ -matrix, whose rows are the  $n$ -vectors  $c_k^T, k \in K$ .

Then we can split the vectors and matrices as follows:

$$x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}, \quad x_B = x(J_B) = (x_j, j \in J_B), \quad x_N = x(J_N) = (x_j, j \in J_N),$$

$$C = \begin{pmatrix} C_B \\ - \\ C_N \end{pmatrix}, \quad C_B = C(K, J_B), \quad C_N = C(K, J_N),$$

$$A = A(I, J) = (a_{ij}, 1 \leq i \leq m, 1 \leq j \leq n), \quad A = (A_B | A_N), \quad A_B = A(I, J_B), \quad A_N = A(I, J_N).$$

We give the following definitions:

- The set  $J_B \subset J, |J_B| = m$ , is said to be *support* if  $\det A_B = \det A(I, J_B) \neq 0$ .
- The couple  $\{x, J_B\}$  formed by the feasible solution  $x$  and the support  $J_B$  is called *the support feasible solution*. It is said to be *nondegenerate* if

$$d_{1j} < x_j < d_{2j}, \quad \forall j \in J_B.$$

- The  $p \times n$ -matrix  $E^T = C_B A_B^{-1} A - C$  is called *the reduced cost matrix*, where  $E = (E_B^T, E_N^T)$ , such that  $E_B = E(K, J_B), E_N = E(K, J_N)$ .
- The potential matrix is defined by:  $U = C_B A_B^{-1}$ .

**Remark 2.1** *In this work, we suppose that  $S$  is a bounded set and the problem is nondegenerate; therefore, all the feasible solutions have at least  $m$  noncritical components, with  $m = \text{rang}(A)$ .*

**Remark 2.2** *The support feasible solution is a more general concept than one of the basic feasible solutions. A support feasible solution can be an interior point, a boundary point, or an extreme point of  $S$ , while a basic feasible solution is always an extreme point.*

### 3. Efficient solutions and their properties

**Definition 3.1** A feasible decision  $x^0 \in \mathbb{R}^n$  is said to be an efficient solution (or Pareto optimal solution) for the problem (2.1) if there is no other feasible solution  $x \in S$  such that  $Cx \geq Cx^0$  and  $Cx \neq Cx^0$ .

Let  $S^E$  be the efficient solutions set of the problem (2.1).

**Definition 3.2** A feasible decision  $x^0 \in \mathbb{R}^n$  is said to be a weakly efficient solution (or Slater optimal solution) for the problem (2.1) if there is no other feasible solution  $x \in S$  such that  $Cx > Cx^0$ .

Then we recall the following classical theorems:

**Theorem 3.3** A feasible decision  $x^0$  is efficient if and only if:

$$\exists \lambda \in \mathbb{R}^p, \lambda > 0 : \lambda^T Cx^0 = \max_{x \in S} \lambda^T Cx.$$

**Theorem 3.4** A feasible decision  $x^0$  is weakly efficient if and only if:

$$\exists \lambda \in \mathbb{R}^p, \lambda \geq 0 : \lambda^T Cx^0 = \max_{x \in S} \lambda^T Cx.$$

**Theorem 3.5** (Efficiency criterion) Let  $\{x, J_B\}$  be a support feasible solution for the problem (2.1) and  $k \in K$ .

If

$$\begin{cases} E_{kj} \geq 0, & \text{if } x_j = d_j^-, \quad j \in J_N, \\ E_{kj} \leq 0, & \text{if } x_j = d_j^+, \quad j \in J_N, \\ E_{kj} = 0, & \text{if } d_j^- < x_j < d_j^+, \quad j \in J_N, \end{cases} \quad (3.1)$$

then  $x$  is a weakly efficient solution for the problem (2.1).

If the support feasible solution is nondegenerate, then those relations are also necessary to have  $x$  weakly efficient.

For a discussion of some basic theoretical properties and other approaches to the problem see, for example, the references [1, 5, 8, 9, 15].

Multiple objective linear programming with bounded variables involves determining the whole set of the efficient and all weakly efficient solutions of the problem (2.1) for given  $C, A, b, d_1$  and  $d_2$ .

Our method also determines  $\epsilon$ -efficient solutions.

### 4. $\epsilon$ -Efficient solutions and their properties

**Definition 4.1** Let  $\epsilon \in \mathbb{R}^p, \epsilon \geq 0$ . A feasible decision  $x^\epsilon \in S$  is said to be  $\epsilon$ -efficient for the problem (2.1) if there exists an efficient feasible solution  $x \in S$  such that  $c_k^T x - c_k^T x^\epsilon \leq \epsilon_k, \forall k \in K$ .

Let  $S_\epsilon^E$  be the  $\epsilon$ -efficient solutions set of the problem (2.1).

The following proprieties are the direct results of the definition of  $\epsilon$ -efficient solutions:

**Proposition 4.2** 1.  $S^E \subset S_\epsilon^E, \forall \epsilon > 0$  and  $S^E = S_\epsilon^E, \text{ for } \epsilon = 0$ .

2. If  $\epsilon_1 > \epsilon_2 > 0$ , then  $S_{\epsilon_2}^E \subset S_{\epsilon_1}^E$ .

**Lemma 4.3** [3] A feasible solution  $x^\epsilon \in S$  is said to be  $\epsilon$ -efficient for the problem (2.1) if and only if there exists an efficient solution  $x^0 \in S$  such that for all vectors  $\lambda \in \mathbb{R}_+^p$ ,  $\sum_{k=1}^p \lambda_k = 1$ , satisfying the condition  $\lambda^T C x^0 = \max_{x \in S} \lambda^T C x$ , the following inequality holds:

$$\lambda^T (C x^0 - C x^\epsilon) \leq \epsilon.$$

**Definition 4.4** The value

$$\beta_k(x, J_B) = \sum_{j \in J_N, E_{kj} > 0} E_{kj}(x_j - d_j^-) + \sum_{j \in J_N, E_{kj} < 0} E_{kj}(x_j - d_j^+)$$

is called the  $\epsilon$ -efficiency formula of the objective  $k$ ,  $k \in K$ .

**Theorem 4.5** (Characterization of an  $\epsilon$ -efficient solution)

Let  $\{x, J_B\}$  be a support feasible solution of the problem (2.1) and  $\epsilon$  an arbitrary vector of  $\mathbb{R}_+^p$ .

If there exists  $k \in \{1, \dots, p\}$  such as  $\beta_k(x, J_B) \leq \epsilon_k$ , then  $x$  is  $\epsilon_k$ -weakly efficient for the problem (2.1).

If  $\beta(x, J_B) = (\beta_k(x, J_B), k \in \{1, \dots, p\}) \leq \epsilon$ , then  $x$  is  $\epsilon$ -efficient.

## 5. Finding an initial efficient point

### 5.1. Finding an initial efficient point using Isermann's procedure

Inspired by the Isermann's procedure to find an initial efficient solution, we give a procedure by taking into account the specificity of the constraints of the problem (2.1). The single linear programs introduced in the procedure will be resolved using the adaptive method [10, 13].

The procedure is given in the following steps:

**Step (1):** Get the optimal solution  $(u^0, v^0, \gamma^0, \alpha^0, \beta^0)$  for the following linear program:

$$\begin{cases} \min(b - A d_1)^T u + (d_2 - d_1)^T \gamma, \\ u^T A - v^T C + \gamma^T - \alpha^T = 0, \\ v - \beta = e, \\ \alpha, \gamma, \beta \geq 0, \end{cases} \tag{5.1}$$

where  $e \in \mathbb{R}^p$  is a vector whose entries are each unity. Then go to step (2).

Otherwise, stop the process.

**Step (2):** Get an efficient solution for the problem (2.1) by finding the optimal solution for the following program:

$$\begin{cases} \max(v^0)^T C x, \\ Ax = b, \\ d_1 \leq x \leq d_2, \end{cases} \tag{5.2}$$

using the adaptive method [10].

To establish the validity of this procedure, we prove the following results.

**Theorem 5.1** *If the program (5.1) admits an optimal solution  $(u^0, v^0, \gamma^0, \alpha^0, \beta^0)$ , then the program (5.2) admits an optimal solution. Besides, this solution is efficient for the program (2.1).*

**Proof** Let  $\lambda \in \mathbb{R}^p, \lambda > 0$ , and consider the following linear program:

$$\begin{cases} \max \lambda^T Cx, \\ Ax = b, \\ d_1 \leq x \leq d_2. \end{cases} \quad (5.3)$$

We put  $y = x - d_1$  in the linear program (5.3). We have:

$$\begin{cases} \max \lambda^T Cy + \lambda^T Cd_1, \\ Ay = b - Ad_1, \\ y \leq d_2 - d_1, \\ y \geq 0. \end{cases} \quad (5.4)$$

Let  $(u^0, v^0, \gamma^0, \alpha^0, \beta^0)$  be the optimal solution of (5.1). The dual of the program (5.2) is given by

$$\begin{cases} \min u^T(b - Ad_1) + \gamma^T(d_2 - d_1), \\ u^T A + \gamma^T \geq (v^0)^T C, \\ \gamma \geq 0. \end{cases} \quad (5.5)$$

As  $(u^0, v^0, \gamma^0, \alpha^0, \beta^0)$  is an optimal solution of the program (5.1), then  $(u^0, \gamma^0)$  is a feasible solution for the program (5.5). Since the set  $S$  is not empty, the program (5.2) is feasible. From the duality theory, the program (5.4), with  $\lambda = v^0$ , admits an optimal solution.  $\square$

The following theorem allows us to find an efficient solution for the multiple objective program (2.1) by solving one linear program with bounded variables.

**Theorem 5.2** [4, 6] *The following linear program:*

$$\begin{cases} \max e^T Cx, \\ x \in S, \end{cases} \quad (5.6)$$

*has an optimal solution if and only if the multiple objective program (2.1) admits an efficient solution.*

**Theorem 5.3** *The program (5.1) has an optimal solution if and only if the multiple objective program (2.1) admits an efficient solution.*

**Proof** We give the dual program of (5.1) by

$$\begin{cases} \max e^T z, \\ Ay = b - Ad_1, \\ -Cy + z = 0, \\ y \leq d_2 - d_1, \\ y \geq 0, \\ z \geq 0. \end{cases} \quad (5.7)$$

If we put  $y + d_1 = x$ , then we get the following program:

$$\begin{cases} \max e^T z, \\ Ax = b, \\ z = Cx - Cd_1, \\ d_1 \leq x \leq d_2, \\ z \geq 0. \end{cases} \quad (5.8)$$

We have  $z = Cx - Cd_1 \geq 0$ , and the program (5.8) is equivalent to the following one:

$$\begin{cases} \max e^T Cx - e^T Cd_1, \\ Ax = b, \\ d_1 \leq x \leq d_2. \end{cases} \quad (5.9)$$

However, as  $e^T Cd_1$  is a constant value, then the program (5.9) is equivalent to the program (5.6).

By applying Theorem 5.2 and according to the duality theory, the theorem is established.  $\square$

The following theorem describes a class of multiobjective linear problems for which Isermann's procedure is valid.

**Theorem 5.4** *Isermann's method is valid if  $T = \{x \in S, Cx \geq 0\}$  is nonempty.*

We propose to use the new method for generating an initial efficient point presented in the next section.

### 5.2. Finding an initial efficient point using the Benson–Radjef–Bibi procedure

The authors in [15] developed a procedure to find an initial efficient solution. However, in their method, they used the direct support method to solve the single linear programs introduced in the procedure. Here, we use the same procedure but we solve the single linear programs using the adaptive method [10, 13].

The procedure is given by the following steps:

**Step (1):** Find a feasible solution  $x^0 \in S$ .

**Step (2):** If  $S \neq \emptyset$ , find the optimal solution  $(u^0, w^0, \gamma^0, \alpha^0)$  of the following linear program:

$$\begin{cases} \min u^T (-Cx^0 + Cd_1) + w^T (b - Ad_1) + \gamma^T (d_2 - d_1), \\ u^T C - w^T A - \gamma^T + \alpha^T = -e^T C, \\ u, \alpha, \gamma \geq 0, \end{cases} \quad (5.10)$$

and go to step (3).

Otherwise, stop the process, as the problem (2.1) is infeasible.

**Step (3):** Get an efficient solution for the problem (2.1) by finding the optimal solution of the following program using the adaptive method [10]:

$$\begin{cases} \max(u^0 + e)^T Cx, \\ Ax = b, \\ d_1 \leq x \leq d_2. \end{cases} \tag{5.11}$$

### 6. Generating efficient points

In this phase, we use the adaptive method principle. Starting from the first efficient solution, we calculate a neighbor solution, and we test whether it is efficient. If it is not, we return to another efficient solution and we reiterate the process. Then a test of efficiency of a nonbasic variable is necessary.

#### 6.1. Efficiency test of a nonbasic variable

To test the efficiency of a solution  $x^*$  in the multiple objective linear program (2.1), we use the test developed by Radjef and Bibi [13, 15]. Here, the solution of the single objective programs is done using the adaptive method. We introduce the vector  $v$  of dimension  $p$  and we define the following linear program:

$$\begin{cases} \max g = e^T v, \\ Ax = b, \\ Cx - v = Cx^*, \\ d_1 \leq x \leq d_2, \\ v \geq 0. \end{cases} \tag{6.1}$$

**Theorem 6.1** *If  $\max g = 0$ , then  $x^*$  is efficient. Otherwise,  $x^*$  is not efficient.*

#### 6.2. Construction of the new efficient solution

To find the efficient solution, we introduce into the basis, one by one, the nonbasic variables of the initial efficient solution found in the first phase.

To calculate the new efficient solution  $\bar{x} = x + \theta^0 l$ , we choose a direction of improvement  $l \in \mathbb{R}^n$  and a maximum step  $\theta^0$  along this direction such as  $z_{k_0}(\bar{x}) \geq z_{k_0}(x)$ .

Let  $k_0$  be the criterion satisfying the following relation:

$$\sum_{j \in J_N} |E_{k_0 j}| = \max_{k=1,p} \left( \sum_{j \in J_N} |E_{kj}| \right).$$

In  $J_N$ , we set  $\theta = 1$ , and we put

$$l_j = \begin{cases} d_{1j} - x_j, & \text{if } E_{k_0 j} > 0, \\ d_{2j} - x_j, & \text{if } E_{k_0 j} < 0, \\ 0 & \text{if } E_{k_0 j} = 0, \quad j \in J_N. \end{cases} \tag{6.2}$$

In  $J_B$ , we put  $l(J_B) = -A_B^{-1}A_Nl(J_N)$  to get  $A\bar{x} = b$ , so that  $\bar{x}$  satisfies  $d_1 \leq \bar{x} \leq d_2$ . On the other hand, the maximum step  $\theta^0$  along the direction  $l$  must verify

$$d_{1_j} - x_j \leq \theta^0 l_j \leq d_{2_j} - x_j, j \in J_B.$$

Therefore, we have

$$\theta_j = \begin{cases} \frac{d_{2_j} - x_j}{l_j}, & \text{if } l_j > 0, \\ \frac{d_{1_j} - x_j}{l_j}, & \text{if } l_j < 0, \\ \infty, & \text{if } l_j = 0, \end{cases} j \in J_B,$$

with

$$\theta_{j_0} = \min_{j \in J_B} \theta_j,$$

where  $j_0$  is the candidate index to come out of the basis.

The maximal step is

$$\theta^0 = \min(1, \theta_{j_0}).$$

The new feasible solution is  $\bar{x} = x + \theta^0 l$ .

### 6.2.1. Calculation of $\beta(\bar{x}, J_B)$

For  $k \in \{1, \dots, p\}$ , we have:

$$\begin{aligned} \beta_k(\bar{x}, J_B) &= \sum_{E_{kj} > 0, j \in J_N} E_{kj}(\bar{x}_j - d_{1_j}) + \sum_{E_{kj} < 0, j \in J_N} E_{kj}(\bar{x}_j - d_{2_j}) \\ &= \beta_k(x, J_B) + \theta^0 \left( \sum_{E_{kj} > 0, j \in J_N} E_{kj} l_j + \sum_{E_{kj} < 0, j \in J_N} E_{kj} l_j \right). \end{aligned}$$

Then

$$\beta_{k_0}(\bar{x}, J_B) = \beta_{k_0}(x, J_B) + \theta^0 \left( \sum_{E_{k_0j} > 0, j \in J_N} E_{k_0j} l_j + \sum_{E_{k_0j} < 0, j \in J_N} E_{k_0j} l_j \right).$$

We replace  $l_j$  given by the relations (6.2). Thus, we have

$$\beta_{k_0}(\bar{x}, J_B) = (1 - \theta^0)\beta_{k_0}(x, J_B).$$

From this expression, we deduce that:

- If  $\theta^0 = 1$  then  $\bar{x}$  is optimal for the objective  $k_0$ ; we also say that  $\bar{x}$  is Slater efficient. We consider all the nonbasic variables.
- If  $\beta_{k_0}(\bar{x}, \bar{J}_B) < \epsilon_{k_0}$ , we have  $\bar{x}$  an  $\epsilon$ -optimal for the objective  $k_0$ ; we also say that  $\bar{x}$  is  $\epsilon$ -weakly efficient.
- If  $\beta_{k_0}(\bar{x}, \bar{J}_B) > \epsilon_{k_0}$ , we start a new iteration with the new support solution  $\{\bar{x}, \bar{J}_B\}$ , and we change  $J_B$  as follows:

**6.2.2. Change of support**

Changing support consists of the change of  $E$  to  $\bar{E}$ , and  $U$  to  $\bar{U}$ , so that:

$$\beta_{k_0}(\bar{x}, J_B) \leq \beta_{k_0}(\bar{x}, \bar{J}_B).$$

To this end, we put:

$$\bar{E}_{k_0j} = E_{k_0j} + \sigma_0 t_j,$$

$$\bar{U}_{k_0j} = U_{k_0j} + \sigma_0 t_j,$$

where  $t$  is the reduction direction of the dual function and  $\sigma_0$  is the maximal step along this direction.

**Calculation of  $t$  and  $\sigma_0$  :**

The step  $\theta^0$  is given by:

$$\theta^0 = \min(1, \theta_{j_0}) = \theta_{j_0}, \quad j_0 \in J_B.$$

We seek an index  $j_1 \in J_N$ , which will enter into the basis instead of  $j_0$ .

To this end, we put:

$$t_j = \begin{cases} -sign(l_{j_0}), & \text{if } j = j_0, \\ 0, & \text{if } j \in J_B \setminus j_0, \end{cases}$$

$$t(J_N) = t(J_B)A_B^{-1}A_N,$$

and we calculate

$$\sigma_0 = \sigma_{j_1} = \min_{j \in J_N}(\sigma_j),$$

with

$$\sigma_j = \begin{cases} \frac{-E_{k_0j}}{t_j}, & \text{if } E_{k_0j}t_j < 0, \\ 0, & \text{if } (E_{k_0j} = 0 \text{ and } x_j \neq d_{1_j} \text{ for } t_j > 0) \text{ or} \\ & (E_{k_0j} = 0 \text{ and } x_j \neq d_{2_j} \text{ for } t_j < 0), \quad j \in J_N, \\ \infty, & \text{else .} \end{cases}$$

We have  $\bar{E}_{k_0j_1} = 0$ .

The new support is  $\bar{J}_B = (J_B \setminus j_0) \cup j_1$ .

We can notice that:

$$\begin{aligned} \beta_{k_0}(\bar{x}, \bar{J}_B) &= \sum_{\bar{E}_{k_0j} > 0, j \in \bar{J}_N} \bar{E}_{k_0j}(\bar{x}_j - d_{1_j}) + \sum_{\bar{E}_{k_0j} < 0, j \in \bar{J}_N} \bar{E}_{k_0j}(\bar{x}_j - d_{2_j}) \\ &= (1 - \theta^0)\beta_{k_0}(x, J_B) + \sigma_0 \left( \sum_{E_{k_0j} > 0, j \in \bar{J}_N} t_j(\bar{x}_j - d_{1_j}) + \sum_{E_{k_0j} < 0, j \in \bar{J}_N} t_j(\bar{x}_j - d_{2_j}) \right). \end{aligned}$$

From this expression, we deduce that:

- If  $\beta_{k_0}(\bar{x}, \bar{J}_B) > \epsilon_{k_0}$ , we call the test program to verify the efficiency of this solution; if it is efficient, we will begin a new iteration with the new support feasible solution  $\{\bar{x}, \bar{J}_B\}$ .



- If  $\beta_{k_0}(\bar{x}, \bar{J}_B) < \epsilon_{k_0}$ , we have  $\bar{x}$  an  $\epsilon$ -optimal solution for the objective  $k_0$ ; we also say that  $\bar{x}$  is  $\epsilon$ -weakly efficient.
- If  $\beta_{k_0}(\bar{x}, \bar{J}_B) = 0$ , then we find the Slater efficient solution. Then we consider another nonbasic variable to start a new iteration.

### 7. Algorithm of the method

The steps of the method to search for the efficient solutions are given in the following algorithm:

I. Find the first efficient solution by using the following procedure:

- Find an optimal solution  $(u^0, w^0, \gamma^0, \alpha^0)$  of the program (5.10).
- Obtain an optimal solution solution of the program (5.11).

Let  $x^1$  be the obtained solution, go to II.

II. Set  $s = 1$ :

(1) Let  $\{x^s, J_B\}$  a support feasible solution and  $\epsilon \geq 0$ .

- Calculate  $U = C_B A_B^{-1}$ .
- Calculate  $E = (E_B, E_N) = UA - C$ .

(2) Choose the criterion  $k_0$ .

- Calculate  $\beta_{k_0}(x^s, J_B)$ :
  - If  $\beta_{k_0}(x^s, J_B) > \epsilon_{k_0}$ , go to (3).
  - If  $\beta_{k_0}(x^s, J_B) < \epsilon_{k_0}$ , then  $x^s$  is  $\epsilon_{k_0}$ -weakly efficient, go to (6).
  - If  $\beta_{k_0}(x^s, J_B) = 0$ , then  $x^s$  is weakly efficient, go to (5).

(3) Calculate the new feasible solution by using the following procedure:

- Calculate the vector  $l$ .
- Determine the index  $j_0$  and the maximal step  $\theta^0$ .
- Set  $s = s + 1$ .
- Calculate  $x^s = x^{s-1} + \theta^0 l$ ,  $s^{th}$  feasible solution.
- Calculate  $\beta_{k_0}(x^s, J_B) = (1 - \theta^0)\beta_{k_0}(x^{s-1}, J_B)$ :
  - If  $\beta_{k_0}(x^s, J_B) > \epsilon_{k_0}$ , go to (4).
  - If  $\beta_{k_0}(x^s, J_B) < \epsilon_{k_0}$ , then  $x^s$  is  $\epsilon_{k_0}$ -weakly efficient, go to (6).
  - If  $\theta^0 = 1$ , then  $x^s$  is weakly efficient solution, go to (5).

(4) Change the support:

- Calculate the vector  $t$ .
- Calculate  $\sigma_0$ , and determine index  $j_1$ .
- Determine the new support  $\bar{J}_B = (J_B \setminus j_0) \cup j_1$ .
- Calculate  $\beta(x^s, \bar{J}_B)$ .
  - If  $\beta_{k_0}(x^s, \bar{J}_B) > \epsilon_{k_0}$ , go to (7).

- If  $\beta_{k_0}(x^s, \bar{J}_B) < \epsilon_{k_0}$ , then  $x^s$  is  $\epsilon_{k_0}$ -weakly efficient, go to (6).
- If  $\beta_{k_0}(x^s, \bar{J}_B) = 0$ , then  $x^s$  is weakly efficient solution, go to (5).

(5) Introduction of the  $j$ th corresponding column leads to an unprocessed basis?

- If so, go to (6).
- Else, stop, all the solutions are found.

(6) Can we improve another objective?

- If so, go to (1).
- Else, stop, all the solutions are found.

(7) Consider the program (6.1) with  $x^* = x^s$ :

- If  $\max g = 0$ , the solution  $x^s$  is efficient.
- Else, go to (6).

### 8. Numerical example

Consider the following linear programming problem with bounded variables:

$$\left\{ \begin{array}{l} \max z_1(x) = x_1 + 2x_2, \\ \max z_2(x) = x_1 - 2x_3, \\ \max z_3(x) = -x_1 + x_3, \\ x_1 + x_2 + x_4 = 1, \\ x_2 + x_5 = 2, \\ x_1 - x_2 + x_3 + x_6 = 4, \\ -1 \leq x_1 \leq 1, \\ -2 \leq x_2 \leq 2, \\ -3 \leq x_3 \leq 3, \\ -4 \leq x_4 \leq 4, \\ -5 \leq x_5 \leq 5, \\ -6 \leq x_6 \leq 6. \end{array} \right. \tag{8.1}$$

We have  $x^0 = ( 0 \ 0 \ 0 \ 1 \ 2 \ 4 )$  as a feasible solution.

#### I. Calculation of the first Pareto efficient solution:

First, we solve the program (5.10); by identification we have:

$$\begin{cases} \min(-u_1 + 5u_2 - 2u_3 + 8w_1 + 9w_2 + 12w_3 + 2\gamma_1 + 4\gamma_2 + 6\gamma_3 + 8\gamma_4 + 10\gamma_5 + 12\gamma_6), \\ u_1 + u_2 - u_3 - w_1 - w_3 - \gamma_1 + \alpha_1 = -1, \\ 2u_1 - w_1 - w_2 + w_3 - \gamma_2 + \alpha_2 = -2, \\ -2u_2 + u_3 - w_3 - \gamma_3 + \alpha_3 = 1, \\ -w_1 - \gamma_4 + \alpha_4 = 0, \\ -w_2 - \gamma_5 + \alpha_5 = 0, \\ -w_3 - \gamma_6 + \alpha_6 = 0, \\ u \geq 0, \alpha \geq 0, \gamma \geq 0. \end{cases}$$

To solve this program, we put:

$$w = w^+ - w^- \text{ with } w^+ = \max(0, w) \text{ and } w^- = \max(0, -w),$$

and so we get the following program:

$$\begin{cases} \min(-u_1 + 5u_2 - 2u_3 + 8w_1^+ + 9w_2^+ + 12w_3^+ - 8w_1^- - 9w_2^- - 12w_3^- + 2\gamma_1 + 4\gamma_2 + 6\gamma_3 + 8\gamma_4 + 10\gamma_5 + 12\gamma_6), \\ u_1 + u_2 - u_3 - w_1^+ - w_3^+ + w_1^- + w_3^- - \gamma_1 + \alpha_1 = -1, \\ 2u_1 - w_1^+ - w_2^+ + w_3^+ + w_1^- + w_2^- - w_3^- - \gamma_2 + \alpha_2 = -2, \\ -2u_2 + u_3 - w_3^+ + w_3^- - \gamma_3 + \alpha_3 = 1, \\ -w_1^+ + w_1^- - \gamma_4 + \alpha_4 = 0, \\ -w_2^+ + w_2^- - \gamma_5 + \alpha_5 = 0, \\ -w_3^+ + w_3^- - \gamma_6 + \alpha_6 = 0, \\ u \geq 0, w^+ \geq 0, w^- \geq 0, \alpha \geq 0, \gamma \geq 0. \end{cases}$$

The optimal solution to this program is:

$$(0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 2 \ 0 \ 0 \ 0 \ 0).$$

To obtain the first Pareto efficient solution, we solve the following program:

$$\begin{cases} \max(u^0 + e)^T Cx, \\ Ax = b, \\ d_1 \leq x \leq d_2. \end{cases}$$

Identification yields:

$$\left\{ \begin{array}{l} \max(2x_2), \\ x_1 + x_2 + x_4 = 1, \\ x_2 + x_5 = 2, \\ x_1 - x_2 + x_3 + x_6 = 4, \\ -1 \leq x_1 \leq 1, \\ -2 \leq x_2 \leq 2, \\ -3 \leq x_3 \leq 3, \\ -4 \leq x_4 \leq 4, \\ -5 \leq x_5 \leq 5, \\ -6 \leq x_6 \leq 6. \end{array} \right.$$

The optimal solution to this program by the adaptive method is: ( 1 2 -1 -2 0 6 ). Then our first Pareto efficient solution is: ( 1 2 -1 -2 0 6 ).

We set  $J_B = \{3, 4, 5\}$  and  $J_N = \{1, 2, 6\}$ , and go to II.

**II. Search all Pareto efficient solutions associated to our first Pareto efficient solution:**

**First iteration:** Let  $\{x^1, J_B\}$  be a support feasible solution and  $\epsilon = \begin{pmatrix} 0.01 \\ 0.01 \\ 0.01 \end{pmatrix}$ .

- Calculate the reduced cost matrix  $E$ :

$$E = C_B A_B^{-1} A - C.$$

We have:

$$E_N = \begin{pmatrix} -1 & -2 & 0 \\ -3 & 2 & -2 \\ 2 & -1 & 1 \end{pmatrix} \text{ and } E_B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Thus,

$$E = \begin{pmatrix} -1 & -2 & 0 & 0 & 0 & 0 \\ -3 & 2 & 0 & 0 & 0 & -2 \\ 2 & -1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

- Determine the criterion  $k_0$  such as:

$$\sum_{j \in J_N} |E_{k_0 j}| = \max\left( \sum_{j \in J_N} |E_{1j}|, \sum_{j \in J_N} |E_{2j}|, \sum_{j \in J_N} |E_{3j}| \right) = \sum_{j \in J_N} |E_{2j}| = 7,$$

so  $k_0 = 2$ .

- We have  $\beta_2(x^1, J_B) = 8 > \epsilon_2$ , and then  $x^1$  is not optimal for the objective  $k_0 = 2$ .

- Calculate the direction of improvement  $l$ :

$$\begin{cases} l_N = ( l_1 & l_2 & l_6 ) = ( 0 & -4 & 0 ), \\ l_B = ( l_3 & l_4 & l_5 ) = ( -4 & -4 & 4 ). \end{cases}$$

- Calculate the maximum step:

$$\theta_{j_0} = \min_{j \in J_B} \theta_j = \min(\theta_3, \theta_4, \theta_5) = \theta_3 = 0.5,$$

so  $j_0 = 3$  and  $\theta^0 = \min(1, \theta_{j_0}) = 0.5$ .

- Calculate  $x^2$ :

$$x^2 = x^1 + \theta^0 l = ( 1 \quad 0 \quad -3 \quad 0 \quad 2 \quad 6 )$$

- Calculate  $\beta_2(x^2, J_B) = 4 > \epsilon_2$ , so  $\{x^2, J_B\}$  is not optimal for the objective  $k_0 = 2$ .

**Changing the support:**

- $t = ( 1 \quad -1 \quad 1 \quad 0 \quad 0 \quad 1 ),$

- $\sigma = ( 3 \quad 2 \quad 0 \quad 0 \quad 0 \quad 2 ),$

- $\sigma_0 = \sigma_{j_1} = \sigma_2 = 2$ , so  $j_1 = 2$ .

Our new support is:  $J_B = (J_B \setminus j_0) \cup j_1 = \{2, 4, 5\}$ .

$\beta_2(x^2, J_B) = 0$ , and then  $\{x^2, J_B\}$  is optimal for the objective  $k_0 = 2$ , so  $x^2$  is a Slater efficient solution for the problem (8.1).

- Consider the program (6.1). By identification, we have the following program:

$$\begin{cases} \max g = v_1 + v_2 + v_3, \\ x_1 + 2x_2 - v_1 = 1, \\ x_1 - 2x_3 - v_2 = 7, \\ -x_1 + x_3 - v_3 = -4, \\ x_1 + x_2 + x_4 = 1, \\ x_2 + x_5 = 2, \\ x_1 - x_2 + x_3 + x_6 = 4, \\ -1 \leq x_1 \leq 1, \\ -2 \leq x_2 \leq 2, \\ -3 \leq x_3 \leq 3, \\ -4 \leq x_4 \leq 4, \\ -5 \leq x_5 \leq 5, \\ -6 \leq x_6 \leq 6, \\ v \geq 0. \end{cases}$$

We found  $\max g = 0$ . Hence, the solution  $x^2$  is Pareto efficient for our program (8.1).

**Second iteration:** Let  $\{x^2, J_B\}$  be a support feasible solution. We have:

$$E_N = \begin{pmatrix} -3 & -2 & -2 \\ -1 & 2 & 0 \\ 1 & -1 & 0 \end{pmatrix} \text{ and } E_B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Thus,

$$E = \begin{pmatrix} -3 & 0 & -2 & 0 & 0 & -2 \\ -1 & 0 & 2 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \end{pmatrix}.$$

- Determine the criterion  $k_0$  such as:

$$\sum_{j \in J_N} |E_{k_0j}| = \max\left(\sum_{j \in J_N} |E_{1j}|, \sum_{j \in J_N} |E_{2j}|, \sum_{j \in J_N} |E_{3j}|\right) = \sum_{j \in J_N} |E_{1j}| = 7,$$

so  $k_0 = 1$ .

- We have  $\beta_1(x^2, J_B) = 12 > \epsilon_1$ , and then  $x^2$  is not optimal for the objective  $k_0 = 1$ .
- Calculate the direction of improvement  $l$ :

$$\begin{cases} l_N = (l_1 \ l_3 \ l_6) = (0 \ 6 \ 0), \\ l_B = (l_2 \ l_4 \ l_5) = (6 \ -6 \ -6). \end{cases}$$

- Calculate the maximum step:

$$\theta_{j_0} = \min_{j \in J_B} \theta_j = \min(\theta_2, \theta_4, \theta_5) = \theta_2 = \frac{1}{3},$$

so  $j_0 = 2$  and  $\theta^0 = \min(1, \theta_{j_0}) = \frac{1}{3}$ .

- Calculate  $x^3$ :

$$x^3 = x^2 + \theta^0 l = (1 \ 2 \ -1 \ -2 \ 0 \ 6).$$

We can notice that  $x^3 = x^1$ , so we can deduce that  $x^1$  is optimal for the objective  $k_0 = 1$ . Moreover,  $x^3$  is Pareto efficient for our program (8.1).

Let  $J_B = \{1, 2, 3\}$  be an unprocessed basis.

**Third iteration:** Let  $\{x^1, J_B\}$  be a support feasible solution.

Calculate the reduced cost matrix  $E$ :

$$E_N = \begin{pmatrix} 1 & 1 & 0 \\ 3 & -5 & -2 \\ -2 & 3 & 1 \end{pmatrix} \text{ and } E_B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Thus,

$$E = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 3 & -5 & -2 \\ 0 & 0 & 0 & -2 & 3 & 1 \end{pmatrix}.$$

- Determine the criterion  $k_0$  such as:

$$\sum_{j \in J_N} |E_{k_0 j}| = \max\left(\sum_{j \in J_N} |E_{1j}|, \sum_{j \in J_N} |E_{2j}|, \sum_{j \in J_N} |E_{3j}|\right) = \sum_{j \in J_N} |E_{2j}| = 10,$$

so  $k_0 = 2$ . Thus, we cannot improve another objective and so we stop the algorithm.

- The Pareto efficient solutions found are:

$$x^1 = (1 \ 0 \ -3 \ 0 \ 2 \ 6) \text{ and } x^2 = (1 \ 2 \ -1 \ -2 \ 0 \ 6).$$

## 9. Conclusion

This paper was devoted to developing a new method to solve a multiobjective linear program with bounded variables using the adaptive method, based on the use of an adapted metric. The constructed method generates a large benefit in time and memory space. This is supported by the fact that we use the adaptive method to solve our single linear programs, and it was showed that this method is very efficient, especially in the case of degenerate problems. It is worth noticing that the use of the simplex method is inappropriate for this kind of problem.

We first introduced a new procedure to find a first efficient solution. Subsequently, we developed a detailed procedure to calculate the efficient solutions and the weakly efficient solutions. We have also characterized the  $\epsilon$ -efficient solutions for a multicriteria linear programming problem with bounded variables. We exploited the suboptimal criterion of the adaptive method in single objective programming to get the  $\epsilon$ -efficient solutions and the  $\epsilon$ -weakly efficient solutions to the problem. Moreover, we provided a global algorithm to describe major steps in implementing the method. Finally, we presented a numerical example showing the applicability of the method.

## References

- [1] Benayoun R, de Montgolfier J, Tergeny J, Larichev O. Linear programming with multiple objective functions: step method (STEM). *Math Program* 1971; 1: 366-375.
- [2] Benson HP. An outer approximation algorithm for generating all efficient extreme points in the outcome set of a multiple objective linear programming problem. *J Glob Optim* 1998; 13: 1-24.
- [3] Dimkov MP. Research on the multicriteria linear programming problems. In: Gabasov R, Kirillova FM, editors. *Optimal Control Problems*. Moscow, USSR: Nauka i Technika, 1981, pp. 25-42.
- [4] Ecker JG, Hegner NS. On computing an initial efficient extreme point. *J Oper Res Soc* 1978; 29: 1005-1007.
- [5] Ecker JG, Kouada IA. Finding efficient points for linear multiple objective programs. *Math Program* 1975; 8: 375-377.
- [6] Ecker JG, Kouada IA. Finding all efficient extreme points for multiple objective linear programs. *Math Program* 1978; 14: 249-261.

- [7] Ehrgott M, Wiecek MM. Multiobjective programming. In: Figueira J, Greco S, Ehrgott M, editors. *Multiple Criteria Decision Analysis: State of the Art Surveys*. New York, NY, USA: Springer, 2005, pp. 708.
- [8] Evans JP, Steuer RE. A revised simplex method for linear multiple objective programs. *Math Program* 1973; 5: 54-72.
- [9] Evans JP, Steuer RE. Generating efficient extreme points in linear multiple objective programming: two algorithms and computing experience. In: Cochrane JL, Zeleny M, editors. *Multiple Criteria Decision Making*. Columbia, SC, USA: University of South Carolina Press, 1973, pp. 349-365.
- [10] Gabasov R. *Adaptive Method of Linear Programming*. Preprint. Karlsruhe, Germany: University of Karlsruhe, 1993.
- [11] Gabasov R, Kirillova FM, Tyatyushkin AI. *Constructive methods of optimization*. Minsk, Belarus: P.I.-University Press, 1984.
- [12] Loridan P.  $\epsilon$ -Solutions in vector minimization problems. *J Optimiz Theory App* 1984; 43: 265-276.
- [13] Radjef S. *Application de la méthode adaptée aux problèmes multicritère*. PhD, A Mira University, Bejaia, Algeria, 2011 (in French).
- [14] Radjef S, Bibi MO. An effective generalization of the direct support method. *Math Probl Eng* 2011; 2011: 374390.
- [15] Radjef S, Bibi MO. A new algorithm for linear multiobjective programming problems with bounded variables. *Arab J Math* 2014; 3: 79-92.
- [16] White DJ. Epsilon efficiency. *J Optimiz Theory App* 1986; 49: 319-337.



## **Résumé**

Ce travail est consacré à la minimisation des polynômes et ce en utilisant la méthode adaptée du support. La particularité de cette méthode est qu'elle utilise un critère de suboptimalité qui nous permet d'arrêter l'algorithme avec une précision souhaitée pour trouver des solutions dites  $\varepsilon$ -optimales. Cette méthode est efficace, rapide, simple et permet une réduction de temps dans l'ensemble du processus d'optimisation. En effet, une implémentation sous la version Matlab 8.1.0.604 a été développée et les résultats obtenus sont concluants. De ce fait, nous avons aussi pu concevoir une nouvelle méthode pour résoudre un programme linéaire multiobjectif dont les variables de décision sont bornées. Notre méthode nous permet de trouver les points efficaces et  $\varepsilon$ -efficaces, et ce en exploitant le critère de suboptimalité de la méthode adaptée en mono-objectif. L'algorithme de cette méthode a été programmé et ses étapes sont présentées dans l'annexe, constituant ainsi une bibliothèque utilisable pour résoudre d'autres programmes.

## **Mots clés:**

Minimisation; Linéaire; Quadratique; Polynôme homogène; Variables bornées; Optimisation multiobjectif; Méthode adaptée; Pareto; Efficacité;  $\varepsilon$ -efficacité.