

RÉPUBLIQUE ALGERIENNE DÉMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université des Sciences et de la Technologie Houari Boumediène

Faculté de Mathématiques



THÈSE

Présentée pour l'obtention du **grade de DOCTEUR EN SCIENCES**

**En : MATHÉMATIQUES**

**Spécialité : Recherche opérationnelle**

**Par : AMROUCHE SALIMA**

**Sujet**

Optimisation stochastique multi-objectifs

Soutenue publiquement, le 04 / 06 / 2018, devant le jury composé de :

Mr.ABBAS Moncef,	Professeur à l'U.S.T.H.B	Président
Mr.MOULAÏ Mustapha,	Professeur à l'U.S.T.H.B	Directeur de thèse
Mr.AIDENE Mohammed,	Professeur à l'U.M.M, Tizi Ouzou	Examineur
Mr.CHAABANE Djamal,	Professeur à l'U.S.T.H.B	Examineur
Mr.OUKACHA Brahim,	Professeur à l'U.M.M, Tizi Ouzou	Examineur
Mr.RADJEF Mohamed Said,	Professeur à l'U.A.M, Béjaia	Examineur
Mr. MERAKEB Abdelkader,	Maître de Conférence/A à l'U.M.M, Tizi Ouzou	Invité

# Remerciements

J'adresse mes premiers remerciements très particulièrement à Mustpha MOULAÏ, professeur à la Faculté de Mathématiques à l'USTHB et le directeur de ma thèse, qui m'a guidé pendant plusieurs années. Je le remercie pour tous ses conseils, ses remarques judicieuses qui ont enrichi mon travail et surtout d'avoir cru en moi et de m'avoir fait confiance ainsi que son soutien moral. Sans lui ce travail n'aurait pas pu voir le jour.

Je remercie Monsieur Moncef ABBAS , professeur à la Faculté de Mathématiques à l'USTHB, de m'avoir fait l'honneur d'examiner ma thèse et de présider le jury.

Je tiens aussi à remercier Monsieur Mohamed AIDENE, professeur à l'université Mouloud Mammeri de Tizi Ouzou (UMMTO), qui m'a fait l'honneur d'examiner ma thèse.

Je remercie également Monsieur Djemal CHAABANE, professeur à la Faculté de Mathématiques à l'USTHB, pour l'intérêt qu'il a bien voulu porter à ce travail et pour m'avoir fait l'honneur d'accepter la charge de l'examineur.

Ma gratitude va ensuite à Monsieur Brahim OUKACHA , professeur à l'université Mouloud Mammeri de Tizi Ouzou (UMMTO), qui m'a fait l'honneur d'examiner ma thèse.

Mes remerciements vont également à Monsieur Mohammed Said RADJEF, professeur à l'université Abderrahmane Mira de Bejaia, pour m'avoir fait l'honneur d'accepter la charge d'examineur.

Je remercie également Monsieur Abdelkader MERAKEB, maître de conférences à l'université Mouloud Mammeri de Tizi Ouzou (UMMTO), pour m'avoir fait l'honneur d'accepter de juger mon travail en tant qu'invité.

Que toutes celles et tous ceux qui m'ont aidé puissent trouver ici ma sincère gratitude

---

et ma plus profonde reconnaissance.

Je tiens également à remercier très chaleureusement les différentes ami(e)s et collègues du L'USTHB qui ont partagé avec moi ces différentes années.

Un immense MERCI, tout à fait particulier, à ma mère, mon père, Naima, Mostafa, Malika et ses enfants, je suis consciente que cette thèse ne serait pas ce qu'elle est sans ses soutiens.

# Dédicaces

*A mes parents,*

*A mes soeurs Malika et Naïma,*

*A mon frère Mostafa,*

*A tous mes amis.*

# Table des matières

Liste des figures	vii
<b>1 Notations et abréviations</b>	<b>viii</b>
Notations et abréviations	ix
<b>Introduction générale</b>	<b>1</b>
<b>2 Optimisation linéaire</b>	<b>4</b>
2.1 Introduction . . . . .	4
2.2 Propriétés des ensembles dans $\mathbb{R}^n$ . . . . .	5
2.2.1 Ensembles convexes, polyèdres et polytopes . . . . .	5
2.2.2 Faces, Points extrêmes . . . . .	6
2.3 Formulation d'un problème de programmation linéaire . . . . .	6
2.3.1 Forme canonique du problème PL . . . . .	7
2.3.2 Forme standard du problème PL . . . . .	7
2.4 Propriétés fondamentales de la programmation linéaire . . . . .	8
2.5 Caractérisation des solutions du problème PL . . . . .	8
2.5.1 L'algorithme simplexe . . . . .	11
2.6 Dualité . . . . .	12
2.6.1 Propriétés de la dualité . . . . .	12
2.6.2 L'algorithme dual du simplexe . . . . .	13
2.7 La programmation linéaire en nombres entiers . . . . .	14

---

2.7.1	La méthode de coupure de Gomory . . . . .	15
2.7.2	Méthode par séparation et évaluation B&B . . . . .	16
2.7.3	Les méthodes de coupes . . . . .	19
2.8	Conclusion . . . . .	20
<b>3</b>	<b>Optimisation linéaire stochastique</b>	<b>21</b>
3.1	Introduction . . . . .	21
3.2	Notions fondamentales . . . . .	22
3.2.1	Variables aléatoires, scénarios et espaces de probabilité . . . . .	22
3.3	La programmation linéaire stochastique . . . . .	24
3.3.1	Décisions et étapes . . . . .	25
3.4	Types de modèles d'optimisation stochastique . . . . .	26
3.5	Modèles avec des contraintes probabilistes . . . . .	27
3.5.1	Contraintes probabilistes indépendantes . . . . .	28
3.5.2	Contraintes probabilistes jointes . . . . .	29
3.5.3	Espérance conditionnelle . . . . .	30
3.5.4	Incertitude seulement dans la fonction objectif . . . . .	30
3.6	Modèle de recours . . . . .	32
3.6.1	Différents type de recours . . . . .	34
3.7	Problème stochastique à deux étapes avec recours fixe . . . . .	35
3.7.1	Méthode "L-Shaped" . . . . .	36
3.8	La programmation Linéaire stochastique en nombres entiers . . . . .	45
3.8.1	Méthode "Integer L-shaped" . . . . .	46
3.9	Conclusion . . . . .	47
<b>4</b>	<b>Optimisation linéaire multi-objectifs</b>	<b>48</b>
4.1	Introduction . . . . .	48
4.2	Optimisation mathématique multi-objectifs . . . . .	49
4.2.1	Points particuliers . . . . .	51
4.2.2	Solutions efficaces et points non dominés . . . . .	51

4.3	La programmation linéaires multi-objectifs en variables entières . . . . .	54
4.3.1	La méthode de pondération des fonctions objectifs . . . . .	55
4.3.2	Solutions efficaces supportées et non supportées . . . . .	55
4.3.3	Méthodes graphiques pour la résolution du problème MOILP . . . . .	57
4.3.4	Méthode de Abbas et Moulai [1] . . . . .	59
4.3.5	Méthode de Chergui et Moulai [36] . . . . .	62
4.4	Conclusion . . . . .	64
<b>5</b>	<b>Optimisation linéaire stochastique multi-objectifs</b>	<b>65</b>
5.1	Introduction . . . . .	65
5.2	Programmation linéaire stochastique multi-objectifs . . . . .	66
5.2.1	Concepts de solutions efficaces . . . . .	66
5.3	Programmation linéaire stochastique multi-objectifs en nombres entiers . . . . .	68
5.3.1	Le problème déterministe équivalent de SMOILP . . . . .	69
5.3.2	Faisabilité . . . . .	70
5.3.3	Efficacité (Optimalité) . . . . .	71
5.4	Conclusion . . . . .	73
<b>6</b>	<b>Une méthode de résolution du problème SMOILP</b>	<b>74</b>
6.1	Introduction . . . . .	74
6.2	Optimisation linéaire stochastique multi-objectifs en nombres entiers . . . . .	75
6.2.1	Une méthodologie pour résoudre SMOILP . . . . .	75
6.2.2	Organigramme de la méthode SMOILP . . . . .	80
6.3	Exemple illustratif . . . . .	83
6.4	Expérimentation et résultats . . . . .	107
	<b>Conclusion générale</b>	<b>109</b>

# Liste des Figures

2.1	Arborescence correspondant à B&B . . . . .	18
3.1	La structure des données par bloc . . . . .	38
3.2	Flux d'information transférées entre le problème principal et les sous-problèmes. . . . .	43
4.1	L'espace des décisions à l'espace des objectifs . . . . .	50
4.2	Illustration des définitions . . . . .	53
4.3	Représentation des solutions dans l'espace de décisions et des objectifs . . .	56
4.4	Solutions efficaces . . . . .	58
6.1	Organigramme SMOILP . . . . .	81





# Chapitre 1

## Notations et abréviations

$\mathbb{N}$	Ensemble des entiers naturels
$\mathbb{R}$	Ensemble des nombres réels
$\mathbb{Z}$	Ensemble des entiers relatifs
$\emptyset$	Ensemble vide
$\Omega$	Ensemble de tous les événements aléatoires
$ \Omega $	Le cardinal de l'ensemble $\Omega$ (nombre de ses éléments)
$\omega$	Un élément (événement) de l'ensemble $\Omega$
$\xi$	Vecteur aléatoire, $\xi = (\xi^1, \dots, \xi^R)$ , représenté par $\Xi$
$\mathbb{P}(\xi^r)$	Probabilité de scénario $\xi^r$
$(\Omega, \Xi, \mathbb{P})$	Espace de probabilité
$\mathbb{E}_\xi$	Opérateur d'espérance
$\sigma$	Opérateur d'écart-type
$\mathbb{V}$	Opérateur de la variance, $\mathbb{V} = \sigma^2$
$\ x\ $	si $x \in \mathbb{R}^n$ , $\ x\  = \sum_{i=1}^{i=n} x_i$
$A$	Matrice déterministe ; $A \in \mathbb{R}^{m \times n}$
$b$	Vecteur déterministe ; $b \in \mathbb{R}^m$
$c$	Vecteur ligne déterministe ; $c \in \mathbb{R}^n$
$c(\xi)$	Vecteur ligne aléatoire dépendant de la variable $\xi$
$[\alpha_l]$	Le plus grand entier inférieur ou égal à $\alpha_l$
$z$	Les variables $z$ représentent les décisions de la deuxième étape prises une fois que les réalisations de $\xi$ seront observées
$r$	Indice des scénarios, $r \in \{1, \dots, R\}$ <sup>ix</sup>
$R$	Le nombre des scénarios

$\mathbb{P}^r$	La probabilité d'occurrence du scénario $r$ , $\mathbb{P}^r = \mathbb{P}(\xi^r)$ .
$W(\xi)$	Matrice de recours aléatoire dépendant de la variable $\xi$ ; $W \in \mathbb{R}^{m_0 \times n_0}$
$T(\xi)$	Matrice de technologie aléatoire dépendant de la variable $\xi$ ; $T \in \mathbb{R}^{m_0 \times n}$
$q(\xi)$	Vecteur de recours aléatoire dépendant de la variable $\xi$ ; $q \in \mathbb{R}^{n_0 \times 1}$
$(q(\xi))^t$	Vecteur transposé de recours aléatoire $q^t \in \mathbb{R}^{1 \times n_0}$
$Q(x, \xi)$	Fonction de recours qui représente le coût minimum pour corriger la décision de la première étape et satisfaire les contraintes
$Q(x)$	Espérance de la fonction de recours
$\theta$	Limite inférieure sur $Q(x)$ dans la méthode <i>L-shaped</i>
$S$	Ensemble des solutions réalisables de $PL$ ; $S = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$
$\mathbf{S}$	Ensemble des solutions entières réalisables de $PL$ avec $\mathbf{S} = S \cap \mathbb{Z}^n$ .
$B_i$ ou $J$	Ensemble des indices des variables de base
$N_i$ ou $\bar{J}$	Ensemble des indices des variables hors base
$H_i$	Ensemble des directions possibles d'amélioration des critères
s.c	Sous les contraintes
$PL$	Programme linéaire
LP	La programmation linéaire ( <i>Linear Programming</i> )
ILP	La programmation linéaire en nombres entiers ( <i>Integer Linear Programming</i> )
SLP	La programmation stochastique linéaire ( <i>Stochastic Linear Programming</i> )
SIP	La programmation stochastique en nombres entiers ( <i>Stochastic Integer Programming</i> )
SMOILP	La programmation linéaire stochastique multi-objectifs en nombres entiers ( <i>Stochastic Multi-Objectifs Integer Linear Programming</i> )
B&B	Méthode par séparation et évaluation ( <i>Branch and Bound</i> )
L-Shaped	Méthode en forme L
Integer L-Shaped	Méthode en forme L en nombres entiers

# Introduction générale

L'importance et la richesse de l'optimisation stochastique et l'optimisation multi-objectifs ont contribué fortement au règne particulier de ces deux domaines dans la recherche opérationnelle.

Les programmes stochastiques multi-objectifs sont des programmes mathématiques où quelques données incertaines sont présentes à la fois dans les objectifs à optimiser et/ou dans les contraintes du problème. L'incertitude est habituellement caractérisée par une distribution de probabilité sur les paramètres. Bien que l'incertitude soit rigoureusement définie, dans la pratique elle peut s'étendre en détail en quelques scénarios (résultats possibles des données) aux distributions communes spécifiques et précises de probabilité.

Le début de la programmation stochastique et en particulier la programmation stochastique linéaire (SLP) a vu le jour dans les années cinquante et le début des années soixante du dernier millénaire.

En 1955, Dantzig [43], fut le premier qui a formulé le problème général de la programmation linéaire avec des données incertaines. En 1961, il a discuté avec Madansky [44] à un stade précoce la possibilité de résoudre des programmes linéaires stochastiques à deux étapes.

Parmi les approches de solutions considérées comme outils de base pour la résolution du SLP, nous citons entre autres la méthode du simplexe de Dantzig [45], la décomposition Dantzig-Wolf [45], la décomposition dual de J.F. Benders [24], la méthode des coupes introduite par J.E. Kelley [63] et la méthode faisable direction proposée et discutée en détail par G. Zoutendijk [110]. Ces méthodes et leurs implémentations n'ont cessé d'être révisé et développé jusqu'à nos jours.

L'observation selon laquelle les problèmes du monde réel doivent être résolus de manière optimale selon des critères conflictuels interdisant une solution idéale (optimale pour chaque critère) a conduit au développement de l'optimisation multicritère.

De ses premières racines, qui ont été posées par Pareto à la fin du XIX<sup>ème</sup> siècle, la discipline a prospéré et grandi, en particulier au cours des trois dernières décennies. Aujourd'hui, de nombreux systèmes d'aide à la décision intègrent des méthodes pour faire face à des objectifs contradictoires. Le fondement de tels systèmes est une théorie mathématique de l'optimisation sous plusieurs objectifs.

L'introduction de variables discrètes dans les problèmes multi-objectifs rend ces problèmes beaucoup plus difficile à aborder, même s'ils sont linéaires. L'ensemble admissible n'est plus convexe et les difficultés sont importantes.

L'optimisation stochastique et l'optimisation multi-objectifs ont connu un développement rapide, impressionnant et extrêmement fructueux au cours des dernières décennies. Étonnamment, ces deux domaines ont évolué de façon assez surprenante, mais ils ont évolué en bonne partie séparément l'un de l'autre, et même aujourd'hui, leur intersection n'est abordée que par une fraction relativement faible des publications. Les problèmes stochastiques multi-objectifs (SMOP) sont apparus dans la littérature avec les travaux de Goicoechea et al. (1976) [54], Stancu-Minasian (1984) [91], Teghem et al. (1986) [95], Slowinski et Teghem (1988) [89] et ceux d'Urli et Nadeau (1990) [102]. Ces méthodes ont été successivement appliquées dans les problèmes réels.

La présence des variables entières dans les SMOP provoque de nouvelles difficultés pour trouver les solutions efficaces et développer des méthodes interactives. Bon nombre des difficultés inhérentes aux problèmes de programmation linéaire stochastique multi-objectifs en nombres entiers (SMOILP) sont déjà visibles.

En 1990, Teghem [99] fut le premier à proposer une méthode interactive: STRANGE-MOMIX pour les problèmes SMOILP. En 2006, Abbas et Belhacene [2], proposent un algorithme qui combine la technique "cutting plane" développée par Abbas et Moulaï [1] et la méthode de décomposition *L-shaped* développée par Van Slyke et Wets [106] et qui est connue également sous le nom de la décomposition de Benders [24].

Soulignons que Fouad Ben Abdelaziz a une grande contribution dans le domaine de la programmation multi-objectifs citons [17], [18], [19], [20], [21], [22], [23].

Notre travail de recherche s'inscrit dans le cadre d'un problème SMOILP, nous avons proposé un algorithme très intéressant [8] combinant la technique de coupe d'efficacité développée par Abbas et Moulaï [1] et le processus de séparation de la méthode *branch & bound* [85] et la méthode *L-shaped*.

Cette thèse comporte cinq chapitres. Le premier chapitre, a pour objectif de présenter les concepts de base de la programmation linéaire, notamment les méthodes de résolution classiques (méthodes du simplexe, dual du simplexe et "*Branch & Bound*") [39], [45], [85],[100].

Le deuxième chapitre est entièrement consacré à regrouper les généralités constituant une introduction à la programmation stochastique [60], [59], [78], [91], [88]. Nous nous intéressons particulièrement aux fondements théoriques de la programmation stochastique ainsi que leurs méthodes de transformation à des problèmes déterministes équivalentes. Nous passons en revue quelques méthodes de résolution telles que "*L-shaped*" et "*Integer L-shaped*". [32], [59], [61].

Le troisième chapitre traite le cadre général de la programmation multi-objectifs. Nous présentons des notions essentielles de l'optimisation multi-objectifs (la dominance, l'efficacité) et nous présentons quelques méthodes classiques d'optimisation multi-objectifs discrète existantes dans la littérature [76], [33], [37].

Le quatrième chapitre se focalise sur le passage d'un problème stochastique multi-objectifs SMOILP de l'état stochastique à l'état déterministe MOILP [24], [17], [20].

Le cinquième et dernier chapitre est consacré a notre nouvelle méthode exacte pour la détermination de l'ensemble des solutions efficaces avec les valeurs de recours correspondantes pour les problèmes SMOILP [8], [9], [10]. Nous donnons également un exemple numérique relatif .

Nous terminons notre thèse par une conclusion qui synthétise nos résultats trouvés et qui jette la lumière sur nos futures perspectives.

# Chapitre 2

## Optimisation linéaire

### 2.1 Introduction

L'optimisation linéaire consiste à trouver parmi un ensemble de solutions respectant des contraintes une solution qui optimise une fonction objectif. Par optimiser, on entend trouver la plus petite valeur (problème de minimisation) ou la plus grande valeur (problème de maximisation). Dans ce chapitre, nous donnons de brefs rappels sur les principaux résultats de la programmation linéaire (LP : *Linear Programming*) et de la programmation linéaire en variables entières (ILP : *Integer Linear Programming*), [42], [43], [83], [85], [39], [75], [100].

Nous commençons par énoncer quelques propriétés sur les points et les ensembles dans  $\mathbb{R}^n$ . Nous présentons une formulation du problème ILP, et décrivons les méthodes qui sont les plus souvent utilisées pour les résoudre. La principale caractéristique d'un programme linéaire réside dans sa facilité de résolution grâce au célèbre algorithme du Simplexe. Les problèmes ILP sont difficiles à résoudre, du fait notamment que le domaine admissible n'est plus convexe mais discret. Le développement du problème ILP a connu trois étapes successives, à savoir : les méthodes de coupure furent les premières à être proposées, comme conséquence directe des travaux de R.E.Gomory en 1958 [52]. Cette première approche fut ensuite remplacée au début des années 60 [68] par celle des procédures par séparation et évaluation (B&B : *Branch and Bound*) et des méthodes de coupes.

L'approche des méthodes de coupure connut vingt ans plus tard une seconde force dans le cadre de la théorie polyédrale avec la notion d'inégalités valides qui peut aussi être intégrée dans une approche *Branch and Cut*.

## 2.2 Propriétés des ensembles dans $\mathbb{R}^n$

Les notations et les terminologies utilisées dans ces sections sont tirées de [100], [85].

### 2.2.1 Ensembles convexes, polyèdres et polytopes

**Définition 2.1 (Ensemble convexe)** *Un sous-ensemble  $\mathfrak{C}$  de  $\mathbb{R}^n$  est dit convexe si*

$$\forall x, x' \in \mathfrak{C}, \forall \lambda \in [0, 1], (\lambda x + (1 - \lambda)x') \in \mathfrak{C}.$$

D'un point de vue géométrique, un convexe est donc un ensemble qui, lorsqu'il contient deux points, contient nécessairement le segment les reliant. L'ensemble vide est par convention convexe.

**Définition 2.2 (Cône convexe)** *Un sous-ensemble  $\hat{C}$  de  $\mathbb{R}^n$  est un cône convexe si*

$$\forall x^1, x^2 \in \hat{C}, \forall \lambda_1, \lambda_2 \in \mathbb{R}^+, \lambda_1 x^1 + \lambda_2 x^2 \in \hat{C}.$$

**Définition 2.3 (Polyèdre)** *Un polyèdre dans  $\mathbb{R}^n$  est un ensemble de la forme*

$$P = \{x \in \mathbb{R}^n : Ax \leq b\},$$

où  $A \in \mathbb{R}^{m \times n}$  et  $b \in \mathbb{R}^m$  sont fixés.

*Autrement dit, c'est une intersection finie de demi-espaces fermés de  $\mathbb{R}^n$ . En fait, tout polyèdre dans  $\mathbb{R}^n$  est convexe.*

**Corollaire 2.1** *Un polyèdre de  $\mathbb{R}^n$  possède un nombre fini de faces. En particulier, il contient un nombre fini de points extrémaux, appelés aussi sommets.*

**Définition 2.4 (Polytope)** *Un polyèdre convexe borné est un polytope convexe.*



## 2.2.2 Faces, Points extrêmes

**Définition 2.5 (Face)** Soit  $\mathfrak{C}$  un sous-ensemble convexe fermé de  $\mathbb{R}^n$  et  $F \subseteq \mathfrak{C}$  convexe fermé lui aussi. On dit que  $F$  est une face de  $\mathfrak{C}$  si quels que soient  $x, x' \in \mathfrak{C}$  et  $\lambda \in [0, 1]$ , si  $\lambda x + (1 - \lambda)x' \in F$  alors nécessairement  $x \in F$  et  $x' \in F$ .

**Définition 2.6 (Point extrême)** On appelle  $x^0 \in \mathfrak{C}$  un point extrême (ou extrémal) si et seulement si quels que soient  $x, x' \in \mathfrak{C}$  et  $\lambda \in [0, 1]$ , si  $\lambda x + (1 - \lambda)x' = x^0$  alors nécessairement  $x = x' = x^0$ . Autrement dit, un point extrême d'un convexe  $\mathfrak{C}$  de  $\mathbb{R}^n$  est une face réduite à un point.

## 2.3 Formulation d'un problème de programmation linéaire

La programmation linéaire est un cas particulier de la programmation mathématique pour lequel la fonction objectif (fonction de coût) et les contraintes du problème sont linéaires. En écriture matricielle, tout programme linéaire  $PL$  peut être transformé sous la forme standard suivante :

$$(PL) \left\{ \begin{array}{l} \min Z(x) = cx \\ \text{s.c} \\ Ax \left\{ \begin{array}{l} = \\ \leq \\ \geq \end{array} \right\} b \\ x \geq 0 \end{array} \right. \quad (2.1)$$

où  $c = (c_1, \dots, c_n)$  est un vecteur ligne de  $n$  colonnes, les variables de décision est un vecteurs colonnes  $x = (x_1, \dots, x_n)^t$  à  $n$  lignes ,  $A = (a_{ij})_{1 \leq i \leq m, 1 \leq j \leq n}$  est une matrice à  $m$  lignes et  $n$  colonnes et  $b = (b_1, \dots, b_m)$  est un vecteur colonne de  $m$  lignes. L'inégalité vectorielle  $x \geq 0$  signifie que chaque composante du vecteur  $x$  est positive ou nulle.

L'ensemble des contraintes  $S = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$  représente l'ensemble des solutions admissibles de PL (2.1), où  $S \subseteq \mathbb{R}^n$  est un polyèdre non vide et borné.

**Remarque 2.1** *Il est toujours possible de ramener :*

(i) *l'optimisation à une minimisation (maximiser la fonction  $Z$  est équivalent à minimiser la fonction  $-Z$ ),*

(ii) *toutes les variables à être non négatives. Une variable  $x_j$  de signe quelconque peut être décomposée en deux variables non-négatives  $x_j = x_j^+ - x_j^-$ , avec  $x_j^+ = \max(0, x_j)$  et  $x_j^- = \max(0, -x_j)$ .*

### 2.3.1 Forme canonique du problème PL

Pour obtenir cette forme équivalente, toutes les contraintes d'égalités sont mises sous forme d'inégalité suivante :

$$A_i x = b_i \Leftrightarrow \begin{cases} A_i x \geq b_i \\ -A_i x \geq -b_i \end{cases}$$

avec  $A_i$  : la  $i^{\text{ème}}$  ligne de la matrice  $A$  et  $b_i$  la  $i^{\text{ème}}$  coordonnée de vecteur  $b$ .

En notation matricielle, la forme canonique est donc :

$$\left( \begin{array}{l} \min Z(x) = cx \\ \text{s.c} \\ Ax \geq b \\ x \geq 0 \end{array} \right) \quad \left( \text{ou encore} \quad \left( \begin{array}{l} \max Z(x) = cx \\ \text{s.c} \\ Ax \leq b \\ x \geq 0 \end{array} \right) \right)$$

### 2.3.2 Forme standard du problème PL

Pour obtenir cette forme équivalente, toutes les contraintes d'inégalité sont mises sous forme d'égalité par l'introduction de variables d'écart non négatives :

$$A_i x \geq b_i \Leftrightarrow A_i x - x'_i = b_i \text{ et } x'_i \geq 0$$

$$A_i x \leq b_i \Leftrightarrow A_i x + x'_i = b_i \text{ et } x'_i \geq 0$$

En notation matricielle, la forme standard est donc :

$$(LP) \begin{cases} \min Z(x) = cx \\ \text{s.c} \\ Ax = b \\ x \geq 0 \end{cases} \quad (PL)$$

## 2.4 Propriétés fondamentales de la programmation linéaire

**Définition 2.7 (Solution admissible)** Une solution admissible (réalisable) du problème (PL) est une solution qui vérifie toutes ses contraintes.

L'ensemble des solutions admissibles noté  $S$ ,  $S = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$  est soit :

- (i) un polytope,
- (ii) un polyèdre convexe, non vide mais non borné,
- (iii) un ensemble vide.

**Définition 2.8 (Solution optimale)** Une solution optimale du problème (PL) est une solution admissible qui minimise la fonction  $Z(x)$ .

$$x^* \text{ solution optimale} \iff x^* \in S \text{ et } cx^* \leq cx, \forall x \in S.$$

## 2.5 Caractérisation des solutions du problème PL

**Définition 2.9 (Base)** Soit  $J$  un sous-ensemble d'indices colonnes  $\{1, \dots, m\}$ , avec  $|J| = m$ . Si la sous-matrice  $A^J$  déduite de  $A$ , en considérant les colonnes d'indices de  $J$ , est inversible, alors  $J$  est une base.

Nous notons par  $J$  l'ensemble des indices des variables de base,  $\bar{J}$  l'ensemble des indices des variables hors base. Sans perte de généralité, on peut supposer que les colonnes de  $A$  ont été ordonnées de manière à pouvoir écrire  $A$  sous la forme  $A = (A^J, A^{\bar{J}})$ , avec  $A^J$  la  $m \times m$ -matrice de base inversible et  $A^{\bar{J}}$  la  $m \times (n - m)$ -matrice contenant les

colonnes de  $A$  différents de  $A^J$ . Le vecteur  $x$  peut être partitionné de façon analogue en posant  $x = (x_J, x_{\bar{J}})^t$ . Les variables  $x_J$  sont appelés variables de base et les variables  $x_{\bar{J}}$  les variables hors base. Finalement le vecteur  $c$  peut lui aussi être partitionné de la même manière en  $c = (c^J, c^{\bar{J}})$ , le programme linéaire (PL) s'écrit :

$$\begin{cases} \min Z(x) = c^J x_J + c^{\bar{J}} x_{\bar{J}} \\ \text{s.c} \\ A^J x_J + A^{\bar{J}} x_{\bar{J}} = b \\ x_J \geq 0 \\ x_{\bar{J}} \geq 0 \end{cases} \quad (2.2)$$

**Définition 2.10 (Solution de base)** *On appelle solution de base du système  $Ax = b$  associée au choix de base  $J$  la solution  $x^*$  définie par:*

$$\begin{aligned} x_{\bar{J}}^* &= (0, \dots, 0)^t. \\ x_J^* &= [A^J]^{-1} b, \end{aligned}$$

*On dit que la solution optimale  $x^*$  est dégénérée si une ou plusieurs composantes de  $x_{\bar{J}}^*$  sont nulles.*

**Définition 2.11 (Solution de base réalisable)** *On dit que la solution de base  $x^*$  du système  $Ax = b$  associée au choix de la base  $J$  est une solution de base réalisable (admissible), si de plus toutes les composantes de  $x^*$  sont positives. Dans ce cas, on dit aussi que la base  $J$  est une base réalisable.*

**Définition 2.12 (Solution de base optimale)** *Soit  $J$  une base réalisable et  $x^*$  la solution de base associée à  $J$ . Si le vecteur  $c^*$  (où :  $(c^{\bar{J}})^* = c^{\bar{J}} - c^J [A^J]^{-1} A^{\bar{J}}$  et  $(c^J)^* = (0, \dots, 0)$ ) n'a que des composantes négatives ou nulles, alors  $x^*$  est une solution optimale du problème PL*

*Si la solution optimale  $x^*$  est dégénérée, alors cette solution optimale n'est pas unique. Les autres solutions des base optimales sont appelées solutions alternatives à  $x^*$ .*

**Définition 2.13** Un programme linéaire (2.2) est dit écrit sous forme canonique par rapport à une base  $J$ , si les deux conditions suivantes sont vérifiées :

- (i)  $A^J$  est, à une permutation près des colonnes, la matrice unité ;
- (ii)  $c^J = 0$ .

**Définition 2.14** Le  $m$ -vecteur ligne  $\pi = c^J(A^J)^{-1}$  est appelé " vecteur multiplicateur relatif à la base  $J$  ".

**Définition 2.15** Le  $n$ -vecteur ligne  $c = (0; c^{\bar{J}} - \pi A^{\bar{J}})$ , noté parfois  $c(J)$ , est appelé " vecteur coût relatif à la base  $J$  " (appelé aussi " coût réduit ").

Le pivotage est une opération appliquée à la matrice augmentée  $M = \begin{pmatrix} A & b \\ c & -\pi b \end{pmatrix}$ .

Aussi, on définit la matrice de pivotage  $D(r, s)$  comme suit :

**Définition 2.16** Étant donné une  $(m+1)(n+1)$ -matrice  $M$ , où  $M = \begin{pmatrix} A & b \\ c & -\pi b \end{pmatrix}$ ;  $r$  indice ligne ( $1 \leq r \leq m$ ) et  $s$  indice colonne ( $1 \leq s \leq n$ ), tels que :  $A_r^s \neq 0$ . La matrice  $D(r, s)$  est telle que :

$$D(r, s) = (e_1, \dots, e_{r-1}, D^r, e_{r+1}, \dots, e_m, e_{m+1})$$

où

$e_i$  est le  $i^{\text{ème}}$  vecteur de la base canonique

$$D^r = \left( -\frac{A_1^s}{A_r^s}, -\frac{A_2^s}{A_r^s}, \dots, -\frac{A_{r-1}^s}{A_r^s}, \frac{1}{A_r^s}, -\frac{A_{r+1}^s}{A_r^s}, \dots, -\frac{A_m^s}{A_r^s}, -\frac{c^s}{A_r^s} \right)^t$$

$A_i^j$  est la composante de la  $i^{\text{ème}}$  ligne et la  $j^{\text{ème}}$  colonne de la matrice  $A$ .

$D(r, s)$  est appelée " matrice de pivotage sur le **pivot**  $A_r^s$  de la  $m \times n$ -matrice  $A$  ".

**Définition 2.17** On appelle "col", une application définie comme suit :

$$\begin{aligned} \text{col} : \{1, \dots, m\} &\longrightarrow J \\ i &\mapsto \text{col}(i) \text{ telle que : } A^{\text{col}(i)} = e_i \left( \begin{array}{l} \text{où } e_i \text{ est le } i^{\text{ème}} \text{ vecteur} \\ \text{de la base canonique} \end{array} \right) \end{aligned}$$

### 2.5.1 L'algorithme simplexe

L'algorithme de programmation linéaire le plus célèbre et le plus ancien consiste en la méthode du simplexe de [42].

Considérons le *PL* (2.2) écrit sous forme canonique par rapport à une base réalisable  $J$ . C'est-à-dire qu'on a :  $b \geq 0$ .

La méthode du simplexe consiste en la répétition du processus suivant jusqu'à ce qu'on obtienne soit une base optimale soit un ensemble de solutions pour lesquelles  $Z$  n'est pas borné soit un ensemble de solutions vide.

**Algorithme 2.1** (*Simplexe*)

**Étape 1.** Choisir  $s$  tel que  $c^s > 0$ , si un tel n'existe pas, la base  $J$  est optimale. Sinon aller à l'Étape 2.

**Étape 2.** Examiner  $A^s$ , si  $A^s \leq 0$  pas de solution optimale ( $Z$  n'est pas borné). Terminer. Sinon, aller à l'Étape 3.

**Étape 3.**  $\mathbb{I} = \{i : A_i^s > 0\} \neq \emptyset$ . Choisir  $i_0 \in \mathbb{I}$ , tel que :  $\frac{b_{i_0}}{A_{i_0}^s} = \min_{i \in \mathbb{I}} \left\{ \frac{b_i}{A_i^s} \right\}$ . Aller à l'Étape 4.

**Étape 4.** Effectuer le pivotage sur la matrice des coefficients du programme linéaire de manière à écrire celui-ci sous forme canonique par rapport à la nouvelle base  $J := (J \cup \{s\}) \setminus \{\text{col}(i_0)\}^{(1)}$ .

Poser  $\text{col}(i_0) = s$  où  $\text{col}(i_0)$  est l'indice de la variable de base associée à la ligne  $i_0$ .

L'élément  $A_{i_0}^s$  est appelé pivot (où  $A_{i_0}^s$  la composante de l'intersection de la colonne  $s$  et la ligne  $i_0$  de la matrice  $A$ ).

<sup>(1)</sup>On fait entrer l'indice  $s$  à l'ensemble des indices  $J$  et en fait sortir l'indice  $\text{col}(i_0)$  de  $J$ .

## 2.6 Dualité

La dualité associe à tout problème linéaire un autre problème linéaire qui est appelé problème dual du problème initial tel que les deux programmes duaux ne constituent pas deux problèmes différents mais deux aspects différents d'un même problème.

### Définition 2.18

$$\text{Etant donné un PL: } (\mathcal{P}) \begin{cases} \max Z = cx \\ \text{s.c} \\ Ax \leq b \\ x \geq 0 \end{cases}, \text{ son PL dual de est le PL: } (D) \begin{cases} \min W = yb \\ \text{s.c} \\ yA \geq c \\ y \geq 0 \end{cases}$$

- Le PL original ( $\mathcal{P}$ ) est alors souvent appelé le PL primal.
- Les programmes ( $\mathcal{P}$ ) et ( $D$ ) sont duaux l'un de l'autre et le problème dual du dual est le problème primal initial.

### 2.6.1 Propriétés de la dualité

Considérons un problème linéaire de maximisation ( $\mathcal{P}$ ) et son dual ( $D$ ).

**Théorème 2.1 (Dualité faible)** *Pour tout couple de solutions  $\bar{x}, \bar{y}$  respectivement réalisables de ( $\mathcal{P}$ ) et ( $D$ ) on a:  $c\bar{x} \leq \bar{y}b$ .*

**Théorème 2.2** *Soit  $x^*$  et  $y^*$  un couple de solutions réalisables pour ( $\mathcal{P}$ ) et ( $D$ ) respectivement tel que  $cx^* = y^*b$ , alors  $x^*$  et  $y^*$  sont respectivement des solutions optimales pour ( $\mathcal{P}$ ) et ( $D$ ).*

**Théorème 2.3 (Théorème de dualité de Gale, Kuhn et Tucker)** *Le problème primal ( $\mathcal{P}$ ) possède une solution optimale  $x^*$  si et seulement si le problème dual ( $D$ ) possède une solution optimale  $y^*$ . Dans ce cas, on a nécessairement  $Z(x^*) = w(y^*)$ .*

**Théorème 2.4 (Dualité forte)** *Si le problème primal (dual) possède une solution optimale finie, alors il en est de même pour le problème dual (primal) et de plus  $Z = w$ .*

**Théorème 2.5 (Théorème des écarts complémentaires)** Soient  $\bar{x}$  et  $\bar{y}$  des solutions réalisables respectivement du (P) et (D). Une condition nécessaire et suffisante pour que  $\bar{x}$  et  $\bar{y}$  soient solutions optimales est qu'elles vérifient les relations suivantes :

$$(c - \bar{y}^t A)\bar{x} = 0 \text{ et } \bar{y}^t (A\bar{x} - b) = 0$$

**Définition 2.19 (Solution duale associée à une base)** Soit  $J$  une base du problème (P). Les coûts réduits des variables de base étant nuls, on a  $c^J - c^J [A^J]^{-1} A^J = 0$ , qui donne  $y_J A^J = c^J$ . Ainsi,  $y_J = c^J [A^J]^{-1}$  et donc  $y_J$  est une solution réalisable de base du problème (D). Si de plus  $c_{\bar{J}} \geq 0$  alors, la base  $J$  est une base optimale si elle est duale réalisable.

## 2.6.2 L'algorithme dual du simplexe

La notion de dualité exposée dans le paragraphe précédent permet de définir d'autres algorithmes de résolution des programmes linéaires qui peuvent être considérés comme des variantes de la méthode du simplexe.

Considérons le PL (2.2) écrit sous forme canonique par rapport à une base  $J$ .

L'algorithme dual du simplexe considère une situation dans laquelle  $c^{\bar{J}} \leq 0$  avec une solution de base non réalisable  $b_i < 0$ . L'algorithme consiste alors à choisir un pivot s'il existe pour rendre  $b \geq 0$ , tout en maintenant les  $c \leq 0$ .

tel que (2.2) soit dual réalisable.

**Algorithme 2.2 (Dual du simplexe)**

**Étape 1.** Choisir  $i_0$  tel que  $b_{i_0} < 0$ . Si un tel  $i_0$  n'existe pas, alors la base  $J$  est optimale. Terminer.

Sinon aller à l'Étape 2.

**Étape 2.** Examiner  $A_{i_0}$ . Si  $A_{i_0} \geq 0$  pas de solution réalisable. Terminer.

Sinon, aller à l'Étape 3.

**Étape 3.**  $\mathbb{S} = \{s : A_{i_0}^s < 0\} \neq \emptyset$ . Choisir  $s_0 \in \mathbb{S}$  tel que:  $\frac{c^{s_0}}{A_{i_0}^{s_0}} = \min_{s \in \mathbb{S}} \left\{ \frac{c^s}{A_{i_0}^s} \right\}$ , aller à l'Étape 4.



**Étape 4.** Effectuer le pivotage sur la matrice des coefficients du programme linéaire de manière à écrire celui-ci sous forme canonique par rapport à la nouvelle base  $J := (J \setminus \{\text{col}(i_0)\}) \cup \{s_0\}$ . Aller à l'Étape 1.

## 2.7 La programmation linéaire en nombres entiers

Dans le développement de la programmation linéaire en variables entières (ILP : *Integer Linear Programming*), il existe trois étapes successives : les méthodes de coupure furent les premières à être proposées, en conséquence essentiellement aux travaux de R.E.Gomory en 1958 [52] ; cette première approche fut ensuite remplacée au début des années 60 [68] par celle des procédures par séparation et évaluation (B&B : *Branch and Bound*) qui fut ensuite largement développées pour entamer la voie de l'optimisation combinatoire. L'approche des méthodes de coupure connue vingt ans plus tard une seconde force dans le cadre de la théorie polyédrale, avec la notion d'inégalités valides qui peut aussi être nécessairement intégrée dans une approche *Branch and Cut*.

Considérons un programme linéaire (PL) pour lequel toutes les variables doivent être entières, le problème résultant, noté (ILP : *integer linear programming*) est le problème général de la programmation linéaire à variables entières :

$$(ILP) \begin{cases} \min Z(x) \\ \text{s.c} \\ x \in \mathbf{S} = S \cap \mathbb{Z}^n \end{cases} \quad (2.3)$$

où  $\mathbf{S} = S \cap \mathbb{Z}^n$  et  $S = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$

À cause de ce changement de la nature des variables, l'ensemble admissible ( $S \cap \mathbb{Z}^n$ ) n'est pas convexe et les programmes linéaires en variables entières nécessitent donc d'autres méthodes de résolution que celles des programmes linéaires à variables continues.

**Remarque 2.2** *Aucune distinction n'est faite entre les variables entières et binaires, puisque les variables entières peuvent être transformées en variables binaires.*

**Définition 2.20 (Inégalité valide de (ILP))** Une inégalité  $A_i x \leq \alpha$  ou  $A_i x \geq \alpha$ , est dite **inégalité valide** pour  $\mathbf{S}$  si elle est vérifiée par toute solution réalisable de (2.3) (c'est-à-dire tout point de  $\mathbf{S}$ ).

Une **coupe** est une inéquation valide qui n'est pas satisfaite pour tout point de  $S$ .

**Définition 2.21** Si  $\alpha$  est un scalaire quelconque on désigne par  $\lfloor \alpha \rfloor$  le plus grand entier inférieur ou égal à  $\alpha$ .

### 2.7.1 La méthode de coupure de Gomory

L'algorithme utilisant les coupes de Gomory procédé comme suit :

La première étape de l'algorithme consiste à résoudre la relaxation linéaire (PL) bien évidemment, si (2.3) n'a pas de solution admissible, il en est de même pour (2.3) et si la solution optimale  $\bar{x}$  de (PL) est entière, le problème est résolu. Supposons donc que ce ne soit pas le cas et que  $\bar{x}$  est finie.

choisir une variable  $x_j$  telle que la valeur  $\bar{x}_j$  est fractionnaire et considérer la ligne correspondante du tableau du simplexe, par exemple la ligne  $i$  :

$$x_j + \sum_{k \in N} a_{ik} x_k = \bar{x}_j$$

où  $N$  est l'ensemble des indices des variables hors-base.

La contrainte

$$\sum_{k \in N} f(\hat{a}_{ik}) x_k \geq f(\bar{x}_j) \tag{2.4}$$

est alors déduite de l'expression précédente, où  $f(r) = r - \lfloor r \rfloor$  désigne la partie fractionnaire du nombre réel  $r$ .

La contrainte (2.4) est **la coupure de Gomory**, peut être rajouter au tableau courant du simplexe.

Lorsque le problème est réoptimisé en variables continues et qu'il existe encore des variables non entières, une nouvelle coupure de Gomory est ajoutée. Autrement dit, après un nombre fini d'application de l'algorithme dual, soit

- on obtient une solution entière optimale
- on constate que le problème est impossible, c'est-à-dire qu'une condition nécessaire pour que les variables soient entières ne peut être vérifiée.

Dans la pratique, les coupes fractionnaires de Gomory convergent très lentement.

Dans cette thèse, nous proposons des procédures basées sur le B&B afin de résoudre, le plus efficacement possible, des problèmes difficiles. Les méthodes du B&B ont été spécialement élaborées pour des problèmes en variables discrètes [100], [85].

### 2.7.2 Méthode par séparation et évaluation B&B

Nous nous intéressons aux méthodes par séparation et évaluation pour ILP, qui sont basées sur l'idée de l'énumération des solutions réalisables. La séparation consiste en la division de l'ensemble des solutions réalisables en plusieurs sous-ensembles et l'évaluation repose sur la résolution des sous-problèmes engendrés par la séparation et à la suppression des sous-ensembles qui ne contiennent pas la solution optimale.

On considère le programme linéaire en nombres entiers :

$$(ILP^{(0)}) \begin{cases} \min Z(x) \\ \text{s.c} \\ x \in \mathbf{S} = S \cap \mathbb{Z}^n \end{cases} \quad (2.5)$$

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

Ce problème est associé au nœud initial 0 de l'arborescence. Au nœud  $i$  de l'arborescence sera associé un problème  $ILP^{(i)}$  en variables entières dont l'ensemble des solutions sera :  $\mathbf{S}^i = S^i \cap \mathbb{Z}^n$ ; On associe son problème relaxé<sup>(2)</sup>  $LP^{(i)}$  dont l'ensemble des solutions sera :  $S^i$

---

<sup>(2)</sup>Problème obtenu en ignorant les contraintes d'intégralité.

**Définition 2.22 (Nœud sondé)** *Un nœud  $l$  de l'arbre est sondé si  $ILP^{(l)}$  n'est pas réalisable, i.e  $\mathbf{S}^l = \emptyset$*

**Définition 2.23 (Nœud actif)** *Les nœuds créés mais non sondés, donc ceux qui restent à examiner, sont appelés les nœuds actifs.*

Une méthode de B&B sera essentiellement constituée de trois éléments principaux :

### (1) Procédure de séparation

On construit une arborescence d'évaluations de problèmes, où la racine correspond à l'évaluation du problème initial.

Soit un nœud  $i$  non sondé. Il existe donc au moins une variable non entière dans la solution optimale  $(x^i)^*$  du problème relaxé  $LP^{(i)}$ ; on choisit le premier  $x_l = \alpha_l \notin \mathbb{Z}$ .

Le nœud  $i$  est séparé en deux sous-nœuds, en imposant respectivement la contrainte supplémentaire :

$$(i) \quad x_l \leq \lfloor \alpha_l \rfloor$$

$$(ii) \quad x_l \geq \lfloor \alpha_l \rfloor + 1$$

La solution  $x^{i*}$  ne vérifie aucune de ces deux contraintes. Ces contraintes (i) et (ii) définissent respectivement deux zones disjointes de  $S^i$  contenant toutes les solutions de  $\mathbf{S}^i$ . Il est aisé (grâce à l'algorithme dual) de réoptimiser les problèmes relaxés correspondant à ces deux sous-nœuds.

$$S^{i+1} = S^i \cap \{x : x_l \leq \lfloor \alpha_l \rfloor\} \text{ et } S^{i+2} = S^i \cap \{x : x_l \geq \lfloor \alpha_l \rfloor + 1\}$$

Il sera généralement conseillé de choisir une subdivision qui correspond à une partition de  $S$ , c'est à dire qui vérifie également :

$$S^i \cap S^j = \emptyset, \quad \forall (i, j), \quad i \neq j.$$

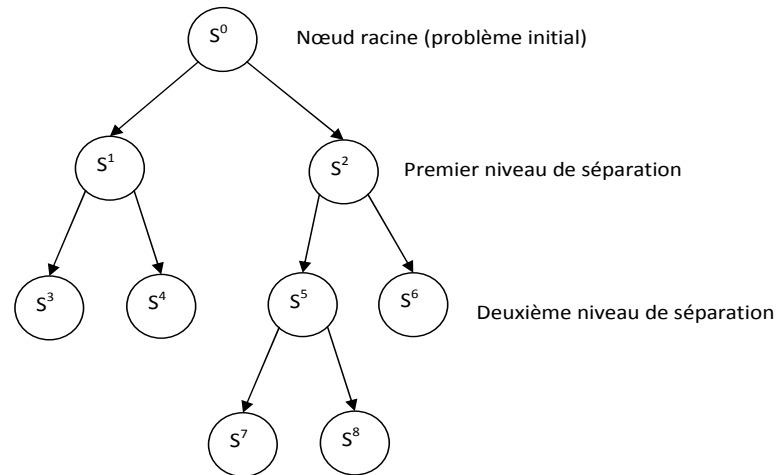


Figure 2.1 : Arborescence correspondant à B&amp;B

Ces subdivisions successives sont représentées à l'aide de l'arborescence ci-dessus.

- Le niveau 0 (appelé racine de l'arborescence) comporte un seul nœud qui correspond au sommet  $S = S^0$ .
- Le niveau 1 comporte deux nœuds qui correspondent aux sommets  $S^1$  et  $S^2$ .
- Le niveau 3 comporte quatre nœuds correspondent aux sommets  $S^3, S^4, S^5$  et  $S^6$

## (2) Procédure d'évaluation

L'évaluation permet de calculer une borne inférieure (dans le cas de la minimisation) sur la valeur de la solution optimale des sous-problèmes issus de la séparation et d'éliminer les nœuds qui ne peuvent contenir la solution optimale du problème original. La seconde partie de l'évaluation, dite d'élagage, consiste en la suppression des nœuds qui ne contiennent pas la solution optimale et à arrêter la séparation des nœuds dont la solution optimale est connue. En effet, lorsque la borne inférieure associée à un nœud est supérieure à la valeur d'une solution réalisable connue, le sous-ensemble correspondant ne peut pas contenir la solution optimale, un tel nœud est sondé. Par ailleurs, un nœud dont la solution optimale est connue ne nécessite aucun développement supplémentaire, et peut donc être stérilisé, dans le sens où seule sa solution optimale est retenue.

### (3) Stratégie de parcours

La stratégie de parcours est la règle suivant laquelle est choisi le sommet devant être séparé parmi tous les sommets pendants (les sommets non encore séparés) de l'arborescence. Parmi les stratégies de parcours les plus connues, on peut citer :

- (i) La largeur d'abord : Cette stratégie favorise les sous-problèmes les plus anciennement créés, c'est à dire les sommets les plus proches de la racine. Il est à noter que cette stratégie est peu utilisée car elle présente une efficacité plus faible que les deux autres stratégies présentées.
- (ii) La profondeur d'abord : L'exploration privilégie le sous problème le plus récemment créé, c'est-à-dire au sommet le plus éloigné de la racine. Cette stratégie permet d'avoir rapidement la solution optimale avec un peu de place mémoire.
- (iii) Le meilleur d'abord : Cette stratégie favorise l'exploration des sous-problèmes possédant la plus grande borne supérieure. Elle dirige la recherche là où la probabilité de trouver une meilleure solution est la plus grande. Elle permet aussi d'éviter l'exploration de tous les sous-problèmes qui possèdent une évaluation inférieure à la valeur optimale.

### 2.7.3 Les méthodes de coupes

Les méthodes de coupes ont pour cible d'obtenir l'enveloppe convexe des solutions entières réalisables, c'est-à-dire le plus petit polyèdre contenant toutes les solutions entières réalisables du ILP. Si l'on arrive à désigner cette enveloppe convexe, la résolution de la relaxation du ILP réduit à cet ensemble donne une solution optimale entière. La difficulté de ces méthodes demeure dans la génération de coupes efficaces. Les méthodes de coupes utilisées affichent des performances simples. Par contre, elles se révèlent efficaces lorsqu'elles sont associées à des méthodes de recherche arborescente telles que le *branch and cut* [85] (méthode combinant l'algorithme du B&B et la méthode des coupes polyédrales).

## 2.8 Conclusion

Dans ce chapitre, nous nous sommes intéressés aux problèmes de programmation linéaire et la programmation linéaire en nombres entiers lorsque toute donnée du problème est connue avec certitude. Nous avons rappelé en premier lieu des éléments essentiels de la théorie de la programmation linéaire et la programmation linéaire en nombres entiers notamment les méthodes de résolution classiques (méthodes du simplexe, dual du simplexe et B&B). Lorsque des paramètres incertains apparaissent dans un problème d'optimisation, on ne se retrouve plus dans le domaine déterministe, on passe dans un monde stochastique qui fait l'objet du chapitre 2.

# Chapitre 3

## Optimisation linéaire stochastique

### 3.1 Introduction

La notion d'incertitude dans la programmation mathématique est apparue pour la première fois dans les années 50 avec les travaux de Bellman et Dantzig [43], Cooper et Charnes [34], [35], et elle a rencontré depuis un développement rapide. Dans ce chapitre on parlera d'incertitude dans le sens d'incertitude aléatoire, où une partie de l'information est nécessaire pour la compréhension complète d'un phénomène est inconnue. L'incertitude dans les problèmes d'optimisation touche notamment les coûts de production, les prix des marchés, les pénalités en cas des violations des contrats, aussi bien que la demande des clients, les délais de livraison, les temps de traitement, la disponibilité des machines et d'autres coefficients technologiques. Il se peut que l'on connaisse partiellement certains aspects du phénomène à travers soit des scénarios, soit un historique, soit une loi de probabilité, soit des moments de la variable aléatoire (par exemple espérance mathématique, variance). Des scénarios peuvent être créés à partir des historiques, (comme par exemple les ventes des dernières années, les températures de la dernière décennie, la mortalité d'une certaine population), ou à partir de l'opinion d'experts qui peuvent prévoir le comportement du phénomène incertain. Si la loi de probabilité de la variable aléatoire est connue de façon analytique, on peut soit créer des méthodes analytiques (la plupart du temps ceci est un travail non-trivial) qui tiennent compte de cette loi de probabilité, soit générer



un ensemble de scénarios suivant cette loi.

Le lecteur pourra trouver dans les travaux suivants [104], [28], [32] et [84] un état de l'art de la programmation stochastique, d'autre part les références [107] et [69] se focalisent essentiellement sur les méthodes de résolution de programmation stochastique en nombres entiers. La littérature autour de la programmation stochastique devient de plus en plus riche, les ouvrages les plus importants dans le domaine étant ceux de [78], [27], [87] et [60] alors que [107], [73] et [69] se focalisent sur les méthodes de résolution de programmation stochastique en nombres entiers.

Dans ce chapitre, nous rappellerons les principaux résultats de la programmation linéaire stochastique en variables continues (SLP : *Stochastic Linear Programming*), nous parlerons de manière générale de toutes les définitions et tous les résultats que nous allons utiliser dans notre travail. Nous aborderons uniquement les résultats nécessaires pour l'étude et la résolution du problème de la programmation linéaire stochastique multi-objectifs en nombres entiers (SMOILP : *Stochastic MultiObjective Integer Linear Programming*) nécessaires pour la compréhension des prochains chapitres. Pour plus de détail le lecteur pourra consulter les ouvrages spécialisés [60], [28] et [64] qui ont été utilisés pour la rédaction de ce chapitre.

## 3.2 Notions fondamentales

### 3.2.1 Variables aléatoires, scénarios et espaces de probabilité

Plusieurs paramètres d'un problème peuvent être considérés comme incertains et sont donc représentés comme des variables aléatoires. Les coûts de production et de distribution dépendent généralement des coûts du carburant, qui sont aléatoires. Les demandes futures dépendent des conditions incertaines du marché. Les rendements des cultures dépendent de conditions météorologiques incertaines.

L'incertitude est représentée en termes d'expériences aléatoires avec des résultats indiqués par  $\omega$  et l'ensemble de tous les résultats est représenté par  $\Omega$ . Les résultats peuvent être combinés en sous-ensembles d'événements. Nous notons par  $\Xi$  un ensemble

d'événements aléatoires.

Enfin, à chaque événement  $A \in \Xi$ , est associé une valeur  $\mathbb{P}(A)$ , appelée probabilité, telle que :

$$(i) \quad 0 \leq \mathbb{P}(A) \leq 1;$$

$$(ii) \quad \mathbb{P}(\emptyset) = 0, \mathbb{P}(\Omega) = 1;$$

$$(iii) \quad \mathbb{P}(A_1 \cup A_2) = \mathbb{P}(A_1) + \mathbb{P}(A_2) \text{ si } A_1 \cap A_2 = \emptyset.$$

Le triplet  $(\Omega, \Xi, \mathbb{P})$  s'appelle un espace de probabilité qui doit satisfaire un nombre de conditions (pour plus de détails le lecteur est invité à consulter [38]). Il est possible de définir plusieurs variables aléatoires associées à un espace de probabilité, c'est-à-dire toutes les variables qui sont influencées par les événements aléatoires de  $A$ .

La description des variables aléatoires utilisée dans la programmation stochastique est étroitement liée à l'ensemble  $\Omega$ . Dans certains cas, les éléments  $\omega \in \Omega$  sont utilisés pour décrire quelques états de *scénarios*. Tous les éléments aléatoires dépendent d'un ensemble fini de plusieurs scénarios. Une telle situation se produit fréquemment dans les modèles stratégiques où la connaissance des résultats possibles à l'avenir est obtenue par le jugement d'experts et où seuls quelques scénarios sont examinés en détail. Dans de nombreuses situations, il est extrêmement difficile et inutile de construire  $\Omega$  et  $\Xi$  ; la connaissance des variables aléatoires est suffisante. Pour une variable aléatoire particulière, on définit sa distribution cumulative  $F_\xi(x) = \mathbb{P}(\xi \leq x)$ , ou plus précisément  $F_\xi(x) = \mathbb{P}(\{\omega : \xi(\omega) \leq x\})$ . Une variable aléatoire discrète prend un nombre fini ou dénombrable de valeurs différentes. Elle est mieux décrite par sa distribution de probabilités, qui est la liste des valeurs possibles,  $\xi^r, r \in \{1, \dots, R\}$ , avec les probabilités associées,

$$\mathbb{P}^r = \mathbb{P}(\xi^r) = \mathbb{P}(\xi = \xi^r) > 0, r = 1, \dots, R \text{ (avec } \mathbb{P}^r > 0 \text{ et } \sum_{r=1}^{r=R} \mathbb{P}(\xi^r) = 1).$$

### 3.3 La programmation linéaire stochastique

Les programmes stochastiques sont des programmes mathématiques où les valeurs de certains coefficients ne sont pas connues avec certitude. Dans la plupart des cas, ces coefficients sont des variables aléatoires caractérisant des lois de probabilité bien précise. Comme les coefficients peuvent prendre plusieurs valeurs, nous disons que chacune des réalisations possibles est un scénario. Il peut arriver qu'il existe un nombre très important ou même infini de scénarios. Or, la capacité technologique de résoudre un problème de programmation linéaire stochastique avec un nombre trop important de scénarios étant encore limitée, il devient naturel de ne retenir qu'un nombre limité de scénarios parmi tous les événements possibles.

Un programme linéaire stochastique est un programme linéaire en présence de coefficients aléatoires définis sur un espace de probabilité  $(\Omega, \Xi, \mathbb{P})$  de distribution connue.

Considérons le programme linéaire stochastique suivant :

$$\left\{ \begin{array}{l} \text{“min” } Z = c(\xi) x \\ \text{s.c} \\ T(\xi) x = h(\xi) \\ Ax = b \\ x \geq 0 \end{array} \right. \quad (3.1)$$

où  $x$  et  $b$  sont des vecteurs réels de dimension  $(n \times 1)$  et  $(m \times 1)$ .  $A$  est une matrice réelle de dimension  $(m \times n)$ ,  $Z$  représente la fonction à optimiser.

$\xi = (\xi^1, \dots, \xi^R)$  est le vecteur composé des variables aléatoires reflétant l'incertitude sur les événements. Les événements possibles  $\xi^r$  s'appellent souvent des *scénarios*, dont  $\mathbb{P}^r = \mathbb{P}(\xi^r)$  est la probabilité d'occurrence.

$c(\xi)$ ,  $h(\xi)$  et  $T(\xi)$  sont respectivement appelés *vecteur coût*, *vecteur second membre* et *matrice de technologie* dépendant de la variable  $\xi$  et de dimensions appropriées ( $c : 1 \times n$ ,  $h : m_0 \times 1$ ,  $T : m_0 \times n$ ).

L'ensemble des contraintes  $S = \{x : Ax = b, x \in \mathbb{R}_+^n\}$  est un polyèdre convexe.

### 3.3.1 Décisions et étapes

Dans les problèmes d'optimisation stochastique, nous nous intéressons aux décisions que l'on prend à présent qui doivent à définir (dans un sens prédéterminé) et doivent permettre de faire face aux circonstances incertaines qui se produiront après (au futur). Ce type de problème s'appelle "*here and now problems*".

Il y a un autre type de problèmes. Si on est capable de connaître la réalisation de la variable aléatoire de futur, alors on est également capable de trouver la solution optimale du présent qui correspond à ce problème. Ces problèmes s'appellent "*wait and see problems*".

Les programmes linéaires stochastiques sont des programmes linéaires dans lesquels certaines données problématiques peuvent être considérées comme incertaines. Les programmes de recours sont ceux dans lesquels certaines décisions de recours peuvent être prises après la révélation de l'incertitude. Pour être plus précis, l'incertitude des données signifie que certaines des données problématiques peuvent être représentées sous forme de variables aléatoires. On suppose qu'une description probabiliste précise des variables aléatoires est disponible, sous forme de distribution de probabilité. Comme d'habitude, les valeurs particulières que prendront les différentes variables aléatoires sont seulement connues après l'expérience aléatoire, car le vecteur  $\xi$  n'est connu qu'après l'expérience.

L'ensemble des décisions est ensuite divisé en deux étapes:

- (i) Un certain nombre de décisions doit être pris avant l'expérience. Toutes ces décisions sont appelées décisions de première étape et la période où ces décisions sont prises est appelée la *première étape*.
- (ii) Plusieurs décisions peuvent être prises après l'expérience. Elles sont appelées décisions de deuxième étape. La période correspondante est appelée la *seconde étape*.

Les décisions de première étape sont représentées par le vecteur  $x$ , tandis que les décisions de deuxième étape sont représentées par le vecteur  $z$  ou  $z(\xi)$  ou même  $z(\xi, x)$  si l'on veut souligner que les décisions de seconde étape diffèrent en fonction du résultat de l'expérience

aléatoire et de la décision de première étape. La séquence d'événements et de décisions est donc résumée comme suit:

$$x \rightarrow \xi \rightarrow z(\xi, x).$$

### 3.4 Types de modèles d'optimisation stochastique

Les approches qui dominent sur la modélisation et la résolution des problèmes de la programmation stochastique sont les suivantes :

- (i) Modèles avec des contraintes probabilistes.
- (ii) Modèles de recours.
- (iii) Modèles robustes.

Dans les sections qui suivent, on présentera brièvement les différentes approches employées dans les modèles avec des contraintes probabilistes. On passera ensuite à la grande famille des modèles de recours : le programme linéaire à deux étapes avec recours (*2-stage linear program with recourse*), qui sera utile pour la résolution de notre problème. La même logique peut être étendue à des problèmes multi-étapes quand le décideur est en face d'une décision de première étape, essayant d'anticiper l'impact de l'aspect aléatoire sur plusieurs périodes du futur. Le problème peut s'étendre en plusieurs étapes suivant l'horizon d'étude.

Dans le contexte multi-étapes, les décisions du problème sont fractionnées en plusieurs groupes, chacun étant déterminé à une étape donnée. Elles sont employées lorsque l'information sur les coefficients incertains peut être révélée sur plusieurs étapes. Citons, à titre d'exemple, des problèmes de gestion d'une chaîne de production, des problèmes de flots dans un réseau, ou encore des problèmes de gestion de stock où dans chacun de ces problèmes les quantités produites, achetées, acheminées ou stockées doivent être calculées sur plusieurs périodes d'un horizon temporel de manière dynamique.

### 3.5 Modèles avec des contraintes probabilistes

La programmation sous contraintes probabilistes due à Charnes et Cooper [35] a pour but de transformer les contraintes stochastiques en des contraintes déterministes équivalentes en considérant la probabilité de leurs réalisations, simultanément ou séparément, au moins égale à un seuil ou à des seuils choisi(s) par le décideur.

Etant donné que la méthode de la programmation sous contraintes probabilistes [35] focalise sur les contraintes, nous considérons alors un programme linéaire stochastique dont l'objectif est déterministe comme suit :

$$\left\{ \begin{array}{l} \min cx \\ \text{s.c} \\ Ax = b \\ Tx \geq \xi \\ x \geq 0 \end{array} \right. \quad (3.2)$$

où :  $c$ ,  $A$  et  $b$  sont les matrices déterministes de dimensions  $(1 \times n)$ ,  $(m \times n)$  et  $(m \times 1)$  respectivement.  $\xi = (\xi^1, \dots, \xi^{m_0})$  est un vecteur aléatoire de distribution connue sur l'espace de probabilité  $(\Omega, \Xi, \mathbb{P})$ .  $T$  est une matrice aléatoire de dimension  $(m_0 \times n)$  définie sur l'espace de probabilité  $(\Omega, \Xi, \mathbb{P})$ .

La contrainte  $Tx \geq \xi$  pourrait s'exprimer comme

$$\max \mathbb{P}(Tx \geq \xi) \quad (3.3)$$

ou encore :

$$\mathbb{P}(Tx \geq \xi) \geq \rho \quad (3.4)$$

$\rho$  étant une probabilité réglée par l'utilisateur qui reflète par exemple la fiabilité d'un système ou une tolérance acceptable. Ces contraintes sont dites jointes ou simultanées.

L'interprétation des contraintes (3.4) reflète l'exigence que la contrainte  $Tx \geq \xi$  soit satisfaite dans  $(\rho \times 100)\%$  des cas, et pas nécessairement toujours. Surtout pour

un problème où la non-satisfaction d'une contrainte est tolérée dans  $(\rho \times 100)\%$  des cas, une exigence de  $(\rho \times 100)\%$  aurait été trop contraignante. Ceci est souvent le cas dans les problèmes de dimensionnement et de fiabilité où l'exigence par rapport aux probabilités de défaillance se modélise avec les contraintes probabilistes.

Un problème plus facile à traiter serait de considérer l'incertitude sur chaque contrainte séparément et ainsi obtenir le système :

$$\mathbb{P}(T_i x \geq \xi^i) \geq \rho^i, \quad i = 1, \dots, m_0 \quad (3.5)$$

où  $T_i$  est la ligne  $i$  de la matrice  $T$  et  $\rho^i$  la probabilité que cette contrainte soit satisfaite. Ces contraintes s'appellent également "contraintes point à point".

Le choix entre les modélisation (3.4) et (3.5) dépendra du fait que les contraintes sont indépendantes les unes des autres ou non. Ceci est imposé par le problème lui même.

Par ailleurs, si les composantes du vecteur aléatoire sont indépendantes, on pourrait déduire :

$$\mathbb{P}\{Tx \geq \xi\} \geq \rho \Leftrightarrow \begin{cases} \mathbb{P}(T_1 x \geq \xi^1) \geq \rho \\ \mathbb{P}(T_2 x \geq \xi^2) \geq \rho \\ \vdots \\ \mathbb{P}(T_{m_0} x \geq \xi^{m_0}) \geq \rho \end{cases} \quad (3.6)$$

ce qui diffère de (3.5).

Il y a essentiellement deux versions différentes de la programmation sous contraintes probabilistes : (i) Contraintes probabilistes indépendantes et (ii) Contraintes probabilistes jointes.

### 3.5.1 Contraintes probabilistes indépendantes

Remplacer chaque contrainte par la probabilité de sa réalisation au moins égale à un seuil choisi par le décideur (les contraintes probabilistes sont indépendantes entre elles).

On note ici que le terme indépendantes porte sur les contraintes probabilistes et non pas nécessairement sur la variable aléatoire.

$$\left\{ \begin{array}{l} \min cx \\ \text{s.c} \\ \mathbb{P}(\xi : T_i x \geq h(\xi^i)) \geq \rho^i, \quad \forall i = 1, \dots, m_0 \\ Ax = b \\ x \geq 0 \end{array} \right. \quad (3.7)$$

où  $h(\xi^i)$  suit la même distribution que la variable aléatoire  $\xi^i$ , et  $T_i$  est la ligne  $i$  de la matrice déterministe  $T$ ;

$\rho^i$  est le seuil de la  $i^{\text{ème}}$  contrainte.

### 3.5.2 Contraintes probabilistes jointes

Remplacer l'ensemble des contraintes par la probabilité (jointe) de leurs réalisations simultanées au moins égale à un seuil convenablement choisi par le décideur.

Le modèle de Katoka [62] au cas où la matrice technologique  $T$  comprend plusieurs lignes qui contiennent des paramètres incertains, ainsi que le second membre est la suivante :

$$\left\{ \begin{array}{l} \min cx \\ \text{s.c} \\ \mathbb{P} \{ \xi : T(\xi) x \leq 0 \} \geq \rho \\ Ax = b \\ x \geq 0 \end{array} \right. \quad (3.8)$$

où  $\rho$  est le seuil de toutes les contraintes simultanément.

La convexité des contraintes jointes n'est pas triviale (exemple en [60]), mais elle peut être assurée dans certains cas [78] où la distribution jointe des éléments de la matrice  $T$  est normale.

La première version (Contraintes probabilistes indépendantes) est plus avantageuse que la deuxième (Contraintes probabilistes jointes) car le décideur peut choisir pour



chaque contrainte  $T_i x \geq h(\xi^i)$  suivant les données du problème, le seuil  $\rho^i$  tel que  $\mathbb{P}(\xi : T_i x \geq h(\xi^i)) \geq \rho^i$ .

### 3.5.3 Espérance conditionnelle

Une autre approche de modélisation serait de mesurer la moyenne des violations d'une contrainte et d'exiger qu'elle ne dépasse pas un seuil  $d$ . On reprend le problème (3.2) et on étudie la contrainte portant sur la variable aléatoire  $\xi$ . L'espérance mathématique  $\mathbb{E}\{\xi^r - T^r x\}$  exprime la moyenne de la violation de la contrainte correspondante. On pourrait également s'intéresser exclusivement à la violation moyenne aux cas où on a violé la contrainte, c'est-à-dire :

$$\mathbb{E}\{\xi^r - T^r x : \xi^r - T^r x > 0\} < d_r, \quad r = 1, \dots, R \quad (3.9)$$

Dans (3.9) on demande que la moyenne des violations de la contrainte  $r$  (cas où la contrainte a été violée) ne dépasse pas la valeur  $d_r$ . On se réfère à [78] pour le passage de la forme stochastique à la forme déterministe.

### 3.5.4 Incertitude seulement dans la fonction objectif

Dans la cas où l'aléa n'intervient que dans la fonction objectif, ce qui est mis en question n'est pas la réalisabilité du programme, puisque le domaine des solutions réalisables n'est pas affecté par les éléments aléatoires. Ce que l'on considère est la qualité de la solution obtenue et d'une certaine manière sa robustesse, à savoir les variations en termes de coûts pour différentes réalisations de la variable aléatoire.

On considère la version générique suivante :

$$\left\{ \begin{array}{l} \min h(x, \xi) \\ \text{s.c} \\ Ax = b \\ x \geq 0 \end{array} \right. \quad (3.10)$$

où  $h(x, \xi)$  est une fonction dépendant de la variable aléatoire  $\xi$  et de la variable de décision  $x$ .

Il existe différentes manières de traiter ce type de problèmes :

**(i) La classe E-modèles :**

Dans le cas où l'incertitude n'intervient que par la fonction objectif  $h(x, \xi)$ , la minimisation de l'espérance (3.11) est la méthode la plus simple. Or, ceci peut donner des décisions dont le coût réel ne soit pas représenté par le coût souhaité du modèle.

$$\left\{ \begin{array}{l} \min \mathbb{E}[h(x, \xi)] \\ \text{s.c} \\ Ax = b \\ x \geq 0 \end{array} \right. \quad (3.11)$$

**(ii) La classe V-modèles :**

La fonction objectif est une caractéristique de dispersion comme dans le modèle du type Markowitz. Les modèles du type Markowitz [74] remédient à ce problème de la grande variation des coûts de solutions : ici on minimise d'une part l'espérance des coûts en surveillant la diminution de l'écart type soit aussi minimisé. Une règle simple d'un modèle du type Markowitz est la suivante ( $\mu$  représente la moyenne et  $\sigma$  l'écart type) :

(1) Si  $\exists x : \mu(x) = \mu(x^*)$  alors il faut que  $\sigma(x) > \sigma(x^*)$  ou

(2) Si  $\exists x : \sigma(x) = \sigma(x^*)$  alors il faut que  $\mu(x) > \mu(x^*)$

Un modèle du type Markowitz pourrait être le suivant :

$$\left\{ \begin{array}{l} \min \alpha \mathbb{E}[h(x, \xi)] + \beta \sqrt{\mathbb{V}[h(x, \xi)]} \\ \text{s.c} \\ Ax = b \\ x \geq 0 \end{array} \right. \quad (3.12)$$

Pour une étude de ce type de modèles on se réfère aux travaux récents de Ruszczyński et Shapiro [81].

où  $\alpha, \beta$  des constants réels donnés.

**(iii) La classe P-modèles :**

Transformer la fonction objectif en la contrainte probabiliste suivante :

$$\left\{ \begin{array}{l} \min d \\ \text{s.c} \\ \mathbb{P} \{h(x, \xi) x \leq d\} \geq \rho \\ Ax = b \\ x \geq 0 \end{array} \right. \quad (3.13)$$

où  $\rho$  une probabilité donnée à l'avance, le plus souvent très grande.

Le décideur détermine les paramètres reflétant son attitude vis à vis du risque tel que la fonction objectif est modélisée de telle manière que le résultat correspond plus aux attentes du décideur [105].

## 3.6 Modèle de recours

Les modèles de recours en deux étapes appartiennent au cadre principale de la programmation stochastique [27], [78], [87]. Dans la programmation stochastiques en deux étapes, les variables de décision sont divisées en deux groupes, les variables de décision de la première étape sont des variables déterminées avant la réalisation d'événements aléatoires. Quand la situation devient claire (à la seconde étape), un autre groupe de variables apparaît, ces variables sont appelées variables de recours qui sont déterminées après avoir connu les valeurs réalisées d'événements aléatoires. Il s'agit alors de prendre une décision à la première étape (avant que ne soient connus les paramètres incertains du problème) pour que la valeur moyenne du coût global (coût dû aux deux prises de décision) soit optimisée.

Considérons un programme linéaire de recours à deux étapes dont les variables sont séparées en deux groupes distincts  $x$  et  $z$ . Les variables du premier groupe  $x$  sont décidées en présence d'incertitude, et celles du second groupe  $z$  le sont, une fois l'incertitude levée (il s'agit des variables de recours). Le programme linéaire de recours à deux étapes s'écrit:

$$\left\{ \begin{array}{l} \min_x \mathbb{E}_\xi [c(\xi)x] + \mathbb{E}_\xi [Q(x, \xi)] \\ \text{s.c} \\ Ax = b, \\ x \geq 0 \\ \text{avec} \\ \left\{ \begin{array}{l} Q(x, \xi) = \min_z q^t(\xi)z(\xi) \\ \text{s.c} \\ W(\xi)z = h(\xi) - T(\xi)x \\ z \geq 0 \end{array} \right. \end{array} \right. \quad (3.14)$$

où  $\xi = (\xi^1, \dots, \xi^R)$  est le vecteur composé des variables aléatoires reflétant l'incertitude sur les évènements pouvant se produire entre la première et la seconde étape. Les évènements possibles  $\xi^r$  s'appellent souvent des scénarios, dont  $\mathbb{P}^r = \mathbb{P}(\xi^r)$  est la probabilité d'occurrence.

$x$  et  $b$  sont des vecteurs de dimension  $(n \times 1)$  et  $(m \times 1)$ .  $A$  est une matrice de dimension  $(m \times n)$ ,

$\mathbb{E}_\xi$  l'espérance mathématique par rapport à la variable aléatoire  $\xi$ , alors que  $c(\xi)$ ,  $T(\xi)$ ,  $W(\xi)$  et  $q(\xi)$  sont respectivement appelées *vecteur coût*, *matrice de technologie*, *matrice de recours* et *vecteur coût* dépendant de la variable  $\xi$  et de dimensions appropriées ( $c : 1 \times n$ ,  $T : m_0 \times n$ ,  $W : m_0 \times n_0$ ,  $q : n_0 \times 1$ ).

Le coût de la décision de recours pour  $x$  fixé et une réalisation donnée  $\xi^r$  :

$$Q(x, \xi^r) = \min_z \{(q(\xi^r))^t z : W(\xi^r)z = h(\xi^r) - T(\xi^r)x, z \geq 0\} \quad (3.15)$$

Dans l'écriture du problème (3.14), on remarque que les variables de recours  $z$  dépendent non seulement des variables de décision  $x$  mais aussi de la variable  $\xi$ . La fonction de recours (3.15) étant définie pour  $x$  et  $\xi$  fixés par la valeur optimale du problème d'optimisation

auxiliaire de la seconde étape :

- (i) Le problème de la seconde étape trouve la meilleure décision du recours pour chaque donnée aléatoire possible ( $x$  la décision de la première étape).
- (ii) La solution de la première étape devrait être choisie telle que pour chaque réalisation possible des données aléatoires, une action faisable du recours peut être choisie pour corriger n'importe quelle anomalie.
- (iii) Nous définissons les deux ensembles suivants :

Ensemble de faisabilité de première étape :  $K_1 = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$ , qui représente les valeurs de  $x$  qui sont réalisables dans le problème de la première étape ( peut être connu explicitement ).

Ensemble de faisabilité de seconde étape :  $K_2 = \{x \in \mathbb{R}^n : Q(x, \xi) < \infty \text{ avec probabilité } 1\}$ , qui représente les valeurs de  $x$  qui guident à la faisabilité dans le problème de la seconde étape (difficile à exprimer analytiquement).

### 3.6.1 Différents type de recours

Dans un problème d'optimisation stochastique à deux étapes avec recours, une première décision doit être prise avant la réalisation des paramètres incertains. Lorsqu'une réalisation de ce vecteur aléatoire est connue, une décision de recours doit être mise en place. Les deux décisions sont soumises à des contraintes et un coût est associé à chacune de ces décisions.

On peut distinguer différents cas de recours en fonction de  $q(\xi)$ ,  $z(\xi)$ , et  $W(\xi)$ . Les cas qui sont cités dans la littérature sont le recours fixe, le recours complet, le recours simple et le recours relativement complet.

**Définition 3.1 (Recours fixe)** *Le recours est considéré comme étant fixe ou déterministe si les valeurs  $q$  et  $W$  d'un programme avec recours ne dépendent pas de leurs valeurs sont donc à priori connues  $W(\xi) = W$  (C'est-à-dire  $W$  n'est pas stochastique).*

**Définition 3.2 (Recours complet)** *Le recours fixe est complet si  $\forall x \in \mathbb{R}^n$  il existe  $y \in Y$  tel que  $Q(x, \xi) < +\infty, \forall \xi \in \Xi$ .*

$\{Wy : y \geq 0\} = \mathbb{R}^{m_0} \implies K_2 = \mathbb{R}^n$  Le problème de seconde étape est réalisable pour toutes les décisions de première étape.

Si pour tout  $x \geq 0$  et pour tout  $\xi \in \Xi$ . Le problème de seconde étape est réalisable, alors le recours est complet.

**Définition 3.3 (Recours relativement complet)** Le recours fixe est relativement complet si  $\forall x \in X \subseteq \mathbb{R}^n$  il existe  $y \in Y$  tels que  $Q(x, \xi) < +\infty, \forall \xi \in \Xi$ .

Si pour tout  $x \geq 0$  tel que  $Ax = b$ , et pour tout  $\xi \in \Xi$ , le problème de seconde étape est réalisable, alors le recours est relativement complet.

**Remarque 3.1** Le recours complet implique le recours relativement complet.

**Définition 3.4 (Recours simple)** Le recours fixe est simple si sa forme canonique est

$$\text{la suivante : } \begin{pmatrix} q \\ W \end{pmatrix} = \begin{pmatrix} q^+ & q^- \\ I & -I \end{pmatrix}$$

où  $I$  est la matrice identité de dimension  $n_0 \times n_0$  et  $q^+, q^-$  sont les vecteurs de pénalités de dimension  $1 \times n_0$ .

**Remarque 3.2** Le recours simple implique le recours complet.

## 3.7 Problème stochastique à deux étapes avec recours fixe

Le programme linéaire stochastique à deux étapes avec recours fixe (dû à Dantzig [43] et Beal [16]) cherche à réduire au minimum le coût de la décision de première étape plus le coût prévu du recours de la décision de seconde étape.

Soit la matrice de recours  $W$  ne dépendant pas de  $\xi$ , les pénalités sont indépendantes de la réalisation des incertitudes. On a alors un *recours fixe*. Ceci implique que le problème donné par (3.14) revient au problème de la programmation linéaire stochastique à deux étapes avec recours fixe qui est donné par la forme déterministe suivante :

$$\left\{ \begin{array}{l} \min_x \mathbb{E}_\xi [c(\xi) x] + \mathbb{E}_\xi [Q(x, \xi)] \\ \text{s.c} \\ Ax = b \\ x \geq 0 \end{array} \right. \quad (3.16)$$

où  $Q(x, \xi^r)$  est la valeur optimale du problème de seconde étape :

$$Q(x, \xi^r) = \min_z \{(q(\xi^r))^t z : Wz = h(\xi^r) - T(\xi^r)x, z \geq 0\}$$

et

$$Q(x) = \mathbb{E}_\xi [Q(x, \xi^r)] = \mathbb{E}_\xi \left[ \min_z (q(\xi))^t z \right]$$

où  $Q(x)$  est l'espérance de  $Q(x, \xi^r)$  sur l'espace de scénarios qui s'appelle également *fonction de recours*.  $Q(x)$  peut être vu comme la pénalisation du terme de violation de la contrainte d'égalité (correction par recours).

La fonction objectif de (3.16) est donné par l'espérance mathématique de l'objectif de la première étape  $c(\xi) x$  plus l'espérance mathématique de l'objectif du seconde étape  $(q(\xi))^t z$ .

### 3.7.1 Méthode "L-Shaped"

En 1962 [24], Benders a proposé une méthode de décomposition consistant à classer les variables d'un programme linéaire en deux groupes. Son objectif a été de résoudre des problèmes de programmation mixte. En 1966 [105], Wets a appliqué la décomposition de Benders sur les problèmes de recours en deux étapes. En outre, Van Slyke et Wets (1969) [103] ont développé la première méthode de SLP sous le nom "L-Shaped" destinée à résoudre les problèmes représentés en tant que modèles de recours en deux étapes.

L'idée de la méthode "L-Shaped" consiste donc à diviser les variables en deux groupes. Les variables de la première étape (un problème principal dont la variable de décision  $x$ ) sont les variables dont la valeur doit être fixée avant la réalisation des événements et ne peut en aucun cas être modifiée suivant la réalisation de la variable aléatoire. Par ailleurs,

les variables de la seconde étape (les variables de recours) sont reposées à des variables aléatoires et des variables de la première étape. Le but de cette règle est que l'on tient compte de toute information déjà disponible afin de prendre nos décisions mais en même temps, il est irréaliste de tenir compte des événements inconnus qui n'existent (au moment de la décision) que dans un espace probabiliste.

On engendre deux types de problèmes complémentaires, le problème principale et les sous-problèmes. Les sous-problèmes (un pour chaque scénario) reçoivent les valeurs des variables de la première étape et calculent les variables de recours pour chaque scénario ainsi que le coût associé. Le problème principale contenir les variables de la première étape et la fonction de recours dont la forme exacte n'est pas connue. Si une telle fonction était donnée de façon explicite, alors on connaîtrait automatiquement l'impact de chaque solution de la première étape au second. Le but est de construire cette fonction de recours à partir des solutions proposées successivement par le problème principal et en ajoutant des coupes générées de la résolution des sous-problèmes. Les coupes se construisent en utilisant les solutions des sous-problèmes.

Sous l'hypothèse de distribution discrète, nous obtenons à nouveau un LP, avec la structure de blocs particulière :

$$\left\{ \begin{array}{l} \min_x \mathbb{E}_\xi [c(\xi)x] + Q(x) \\ \text{s.c} \\ Ax = b, \\ T(\xi^r)x + Wz = h(\xi^r), \quad r = 1, \dots, R \\ x \geq 0, \quad z \geq 0. \end{array} \right. \quad (3.17)$$

La construction du problème équivalent déterministe exige une structure de blocs particulière que l'on met en évidence en réécrivant les contraintes sous sa forme matricielle suivante:

de telle sorte que, sous l'hypothèse de distribution discrète, nous obtenons à nouveau un LP, avec la structure de données spéciale indiquée sur la figure 3.1 .



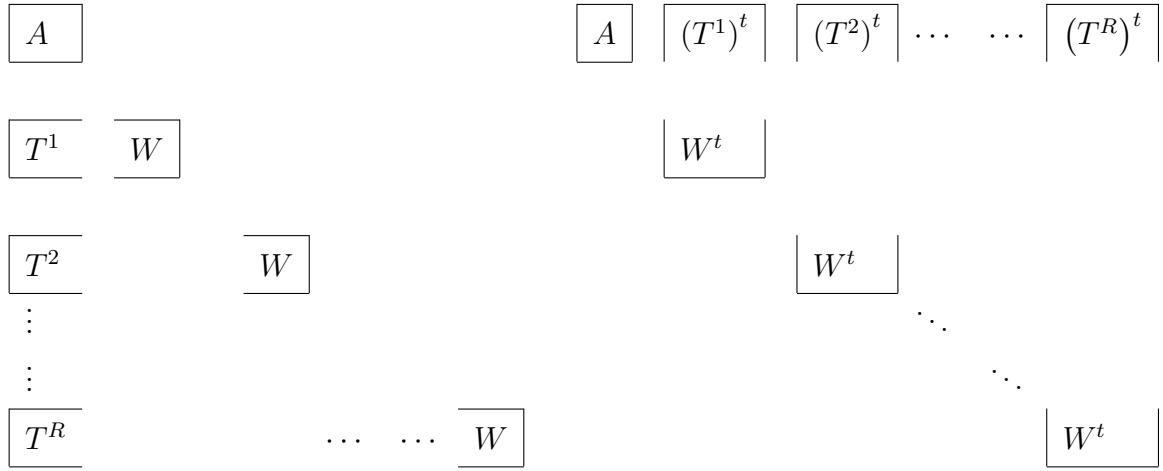


Figure 3.1 : La structure des données par bloc

La forme matricielle des contraintes (a) est une structure par bloc qui a donné le nom à la méthode "L-shaped". En prenant le dual de la forme étendue, on obtient une structure dual angulaire par bloc (b). Cette méthode a été étendue dans la programmation stochastique pour répondre aux questions de faisabilité et elle est connue sous le nom de méthode en L-shaped de Van Slyke et Wets (1969) [103].

La structure de blocs indique que, si l'on fixe les variables  $x$  à une valeur  $\hat{x}$ , alors le problème de minimisation devient séparable par scénario. Cette observation traduit encore la propriété de *non-anticipativité*, puisque les variables  $x$  représentant la décision immédiate sont participantes pour l'ensemble des contraintes associées aux différents scénarios. Une fois  $x$  fixé à  $\hat{x}$ , il suffit donc de résoudre indépendamment les  $R$  sous problèmes pour obtenir la solution du problème principal.

**Remarque 3.3** *On s'aperçoit comment un tel programme (3.17) peut déjà avoir de grandes dimensions.*

*Le nombre de scénarios  $R$  peut être grand, puisque si on a  $V$  variables aléatoires avec  $R_v$  valeurs possibles chacune, on obtient  $R = \prod_{v=1}^V R_v$  scénarios.*

*Par exemple, pour  $V = 5$  variables aléatoires, avec  $R_v = 10$  valeurs possibles chacune, on*

obtient  $R = 10^5$  scénarios, déjà pour un problème simple. Or, si on relaxait la propriété de la non-anticipativité, la structure du programme serait beaucoup plus abordable, puisque l'on pourrait le décomposer en  $R$  sous-problèmes indépendants les uns des autres.

**Définition 3.5 (non-anticipativité)** *Le fait que l'on ne se permet pas de faire dépendre  $x$  de  $\xi$ , c'est-à-dire que pour tout  $\xi^i$  d'avoir une décision  $x^i$  différente, s'appelle la non-anticipativité et c'est la propriété qui couple les variables de la première avec celles de seconde étape. Si on ne tenait pas compte de cette propriété, alors pour toute réalisation de la variable aléatoire  $\xi^i$  on changerait à volonté les variables de la première étape, en choisissant chaque fois une valeur  $x^i$  telle qu'elle minimise la fonction du coût.*

le problème (3.17) équivalent à (3.18) :

$$\left\{ \begin{array}{l} \min_x \mathbb{E}_\xi [c(\xi)x] + \mathbb{E}_\xi \left[ \min_z (q(\xi))^t z \right] \\ \text{s.c} \\ Ax = b \\ W(\xi^r)z = h(\xi^r) - T(\xi^r)x, \quad r = 1, \dots, R \\ x \geq 0, \quad z \geq 0 \end{array} \right. \quad (3.18)$$

Les éléments de données de la première étape  $A$ ,  $b$  et  $c$  sont déterministes parce qu'ils reflètent l'état et les paramètres du système à l'instant présent. Nous supposons que le vecteur aléatoire  $\xi$  a un support fini et  $r = 1, \dots, R$ , indexent ses réalisations possibles et que  $\mathbb{P}^r$  soit leurs probabilités. Dans cette hypothèse, nous pouvons maintenant écrire le programme équivalent déterministe sous la forme extensive. Cette formulation est créée en associant un ensemble de décisions de seconde étape à chaque réalisation de  $\xi$  c'est-à-dire à chaque réalisation de  $q(\xi^r)$ ,  $h(\xi^r)$ , et  $T(\xi^r)$ .

A chaque itération on injecte la variable  $x$  à tout les sous-problème de la seconde étape. On demande de chaque sous-problème deux informations : la valeur de sa fonction objectif et le vecteur des variables duales optimales. Ces informations se regroupent en une coupe, constituant une combinaison linéaire de tous les sous-problèmes.

L'algorithme procède en ajoutant des coupes dont l'existence et la validité est assurée par la théorie de la dualité (trouver une direction d'amélioration). Au départ on résout

le problème (3.18). On récupère la solution  $(x^*, \theta)$  et on fait passer cette solution à chaque sous-problème de deuxième étape:

Primal  $Q(x^*, \xi^r)$

$$\begin{cases} \min (q(\xi^r))^t z \\ \text{s.c} \\ Wz = h(\xi^r) - T(\xi^r)x \\ z \geq 0 \end{cases}$$

Considérons à présent le problème, dual du problème primal  $Q(x^*, \xi^r)$  :

Dual

$$\begin{cases} \max \pi^t [h(\xi^r) - T(\xi^r)x^*] \\ \text{s.c} \\ \pi^t W \leq (q(\xi^r))^t \\ \pi \in \mathbb{R} \end{cases} \quad (3.19)$$

L'ensemble des contraintes  $\mathbf{P} = \{\pi : \pi^t W \leq (q(\xi^r))^t\}$  est indépendant de  $x$ .

Soit  $\{\pi^\alpha : \alpha \in I\}$  l'ensemble des points extrêmes de  $\mathbf{P}$  et  $\{\sigma^\delta : \delta \in \Delta\}$  l'ensemble des arêtes extrêmes [77].

(i) Si  $\mathbf{P} = \emptyset$  alors  $Q(x^*, \xi^r)$  est illimité ( $Q(x^*, \xi^r) = -\infty$ ) ou non réalisable ( $Q(x^*, \xi^r) = +\infty$ ).

(ii) Si  $\mathbf{P} \neq \emptyset$  alors  $Q(x^*, \xi^r)$  est non réalisable ou admet une solution optimale.

Le lemme de *farkas* est une conséquence immédiate du théorème fort de dualité, ce qui apporte un état nécessaire et suffisant pour la faisabilité d'un système des contraintes linéaires et peut être énoncé comme la proposition suivante :

**Proposition 3.1** *L'ensemble  $\{z : Wz = h(\xi^r) - T(\xi^r)x^*, z \geq 0\} \neq \emptyset$*

$$\text{si et seulement si } \sigma^t W \leq 0 \implies \sigma^t [h(\xi^r) - T(\xi^r)x^*] \leq 0.$$

Nous concluons que  $Q(x^*, \xi^r)$  est non réalisable si et seulement si  $\mathbf{P}$  admet une arête extrême  $\sigma_r$  telle que :  $(\hat{\sigma}_r)^t [h(\xi^r) - T(\xi^r)x^*] > 0$  ; Autrement la valeur optimale de  $Q(x^*, \xi^r)$  est obtenu par  $(\pi_r)^t [h(\xi^r) - T(\xi^r)x^*]$ , où  $\pi$  est un point extrême de  $\mathbf{P}$ . Puis nous vérifions la faisabilité pour le problème de seconde étape, nous devons trouver une direction  $\sigma_r$  en résolvant le programme suivant :

$$\begin{cases} \max \sigma^t [h(\xi^r) - T(\xi^r)x^*] \\ \text{s.c} \\ \sigma^t W \leq 0 \\ \|\sigma\|_1 \leq 1 \end{cases} \quad (3.20)$$

La norme sur  $\sigma$  pourrait être la norme  $\|\cdot\|_1$  ou la norme  $\|\cdot\|_2$ . On choisit la norme  $\|\cdot\|_1$  pour rester dans le cas linéaire tel que la dernière contrainte est ajoutée pour limitée (bornée)  $\sigma$ .

Si pour un certain  $\xi^r, r \in \{1, \dots, R\}$ ,  $(\hat{\sigma}_r)^t [h(\xi^r) - T(\xi^r)x^*] > 0$ , où  $\hat{\sigma}_r$  est la solution optimale de (3.20), nous avons trouvé  $\xi^r$  pour lequel  $x^*$  n'est pas une solution réalisable du problème du seconde étape. Dans ce cas, nous ajoutons au programme initial (3.18) la coupe de faisabilité suivante :

$$(\hat{\sigma}_r)^t [h(\xi^r) - T(\xi^r)x] \leq 0 \quad (3.21)$$

On note bien que la coupe (3.21) est équivalent à (3.22) :

$$D_r x \geq d_r \quad (3.22)$$

où  $d_r = (\hat{\sigma}_r)^t h(\xi^r)$ ,  $D_r = (\hat{\sigma}_r)^t T(\xi^r)$  et  $\hat{\sigma}_r$  est la solution optimale de (3.20) et le nouveau programme est résolu en utilisant la méthode dual simplexe

**Remarque 3.4** *Comme il existe un nombre fini de points extrêmes et de directions extrêmes dual alors il y a seulement un nombre fini de contraintes de faisabilité et d'optimalité. Par conséquent, l'algorithme L-shaped doit converger de façon finie.*

Si tous les problèmes admettent une solution, on essayera de faire mieux en obtenant une direction d'amélioration de la fonction de recours (ceci pour chaque scénario

séparément) et tracer ainsi le plan de la fonction de recours au point courant  $x^*$ . Pour tracer cet hyperplan on a besoin :

- (i) de la valeur de la fonction objectif pour chaque sous-problème (3.19) auquel on a passé la solution  $x^*$ ;
- (ii) le vecteur dual optimal  $\hat{\sigma}_r$  du sous problème (3.19) (correspondant au scénario  $r$ ).

Si on se restreint au sous-problème (3.19), l'hyperplan d'appui au point  $x^*$  est donné par la formule suivante :

$$Q_r(x) = Q(x, \xi^r) = (\hat{\pi}_r)^t [h(\xi^r) - T(\xi^r)x]$$

La fonction de recours est la combinaison linéaire de la fonction de chaque sous problème différent. On a donc :

$$Q(x^*) = \sum_{r=1}^R \mathbb{P}^r Q(x^*, \xi^r) = \sum_{r=1}^R \mathbb{P}^r Q_r(x^*)$$

où  $\mathbb{P}^r = \mathbb{P}(\xi^r)$  est la probabilité d'occurrence du scénario  $r$ .

La fonction  $Q(x)$  étant convexe. La *coupe d'optimalité* (3.23) à rajouter dans le problème principal, qui regroupe toute information de tout sous-problème est la suivante :

$$\theta \geq \sum_{r=1}^R \mathbb{P}^r (\hat{\pi}_r)^t [h(\xi^r) - T(\xi^r)x] \quad (3.23)$$

La coupe (3.23) est équivalent à (3.24) :

$$\theta \geq e_r - E_r x \quad (3.24)$$

où  $e_i = \sum_{r=1}^R p^r (\hat{\pi}_r)^t h(\xi^r)$ ,  $E_i = \sum_{r=1}^R p^r (\hat{\pi}_r)^t T(\xi^r)$  et  $\hat{\pi}_r$  est la solution optimale de (3.19)

A chaque itération on rajoute au problème principal soit une coupe de faisabilité, soit une coupe d'optimalité. Au cours des itérations,  $\theta$  sera le maximum point d'une famille d'inégalités de la fonction de recours. Si on arrive à trouver un  $\theta^*$  dont la valeur vaudra  $Q(x^*)$ , alors on aura trouvé l'optimum, qui minimise la fonction réelle du recours  $Q(x)$ .

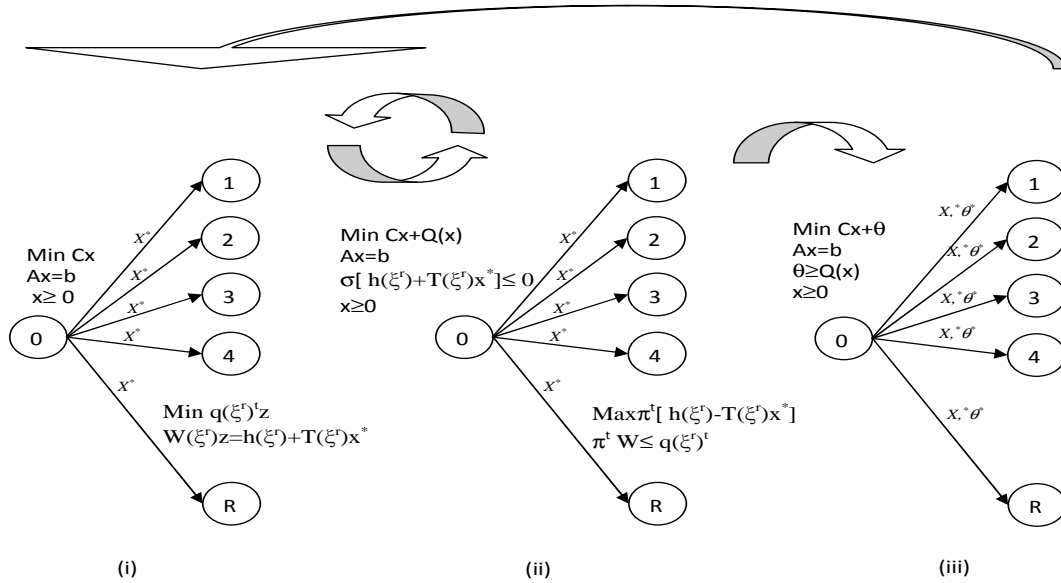


Figure 3.2 : Flux d'information transférées entre le problème principal et les sous-problèmes.

La figure (3.2) montre le flux d'information entre le problème principal et les sous problèmes (correspondant aux différents scénarios).

- (i) **Première itération** : Les solutions  $x^*$  sont transférées du problème principal à tout sous-problème (3.20)
- (ii) **Deuxième itération** : Les vecteurs optimal  $\hat{\sigma}_r$  sont envoyés de sous-problème (3.20) au problème principal ; à partir ces informations une nouvelle coupe agrégée (3.21) ou (3.22) se rajoute au problème principal.
- (iii) **Troisième itération** : Les vecteurs dual optimal  $\hat{\pi}_r$  sont envoyés de tout sous-problème (3.19) au problème principal ; à partir ces informations une nouvelle coupe agrégée (3.23) ou (3.24) se rajoute au problème principal.

**Description de l'algorithme "L-shaped"**

**Algorithme 3.1** (*L-shaped*)

**Étape 1.**  $N_l := M_l := v := 0$ ,  $\bar{z} := \infty$ ,  $\theta = -\infty$  ; résoudre le problème initial  $(P_0)$  :

$$(P_0) \begin{cases} \min cx \\ s.c \\ Ax = b \\ x \geq 0 \end{cases}$$

**Étape 2.** Faire  $l := l + 1$ ; résoudre le problème actuel :

$$(P_l) \begin{cases} \min cx + \theta \\ s.c \\ Ax = b \\ D_l x \geq d_l, \quad l = 1, \dots, N_l \\ E_l x + \theta \geq e_l, \quad l = 1, \dots, M_l \\ x \geq 0, \theta \in \mathbb{R} \end{cases}$$

où  $D_l x \geq d_l$  coupe de faisabilité et  $E_l x + \theta \geq e_l$  coupe d'optimalité.

$N_l$  est l'ensemble des indices d'itération où les contraintes de faisabilité ont été générées.

$M_l$  est l'ensemble des indices d'itération où les contraintes d'optimalité ont été générées.

Si le problème actuel n'a pas de solution réalisable, terminer. Sinon,  $(x^l, \theta^l)$  une solution optimale.

**Étape 3.** Pour  $r = 1, \dots, R$ , résoudre le programme (3.20) en utilisant la solution  $(x^l, \theta^l)$ ; trouver  $\sigma^r$  une solution de (3.20) et déterminer les valeurs du vecteur  $D_r = (\sigma^r)^t T(\xi^r)$  et du coefficient  $d_r = (\sigma^r)^t h(\xi^r)$ . Si ce programme n'est pas faisable pour un scénario  $r$  alors générer la coupe de faisabilité  $D_r x \geq d_r$  et l'ajouter au problème  $(P_l)$ . Faire  $N_l := N_l + 1$  et aller à l'Étape 2. Sinon (si faisable pour tout les scénarios) aller à l'Étape 4.

**Étape 4.** Pour  $r = 1, \dots, R$ , résoudre le programme dual (3.19) en utilisant la solution  $(x^l, \theta^l)$ ; trouver  $\pi^r$  une solution de (3.19) puis calculer  $e_l = \sum_{r=1}^R p^r (\pi_r^l)^t h(\xi^r)$  et  $E_l = \sum_{r=1}^R p^r (\pi_r^l)^t T(\xi^r)$ . Si  $\theta^l \leq e_l - E_l x^l$  ajouter au problème actuel la coupe d'optimalité

$E_l x + \theta \geq e_l$ , faire  $M_l := M_l + 1$  et aller à l'Étape 2. Sinon aller à l'Étape 5.

**Étape 5.** Si  $\theta^l > e_l - E_l x^l$  arrêter l'algorithme car  $(x^l, \theta^l)$  est la solution optimale. Calculer  $Q(x^l)$  et  $\bar{z} = cx^l + Q(x^l)$  et mettre à jour  $\bar{z}$ .

## 3.8 La programmation Linéaire stochastique en nombres entiers

Afin de caractériser les différentes situations possibles de SILP (SILP : *Stochastic Integer Linear Programming*), nous utilisons la notation  $a/b/c$  pour représenter une classe particulière de programmes en deux étapes [69]. La première lettre caractérise les variables de la première étape : soit  $C$  si toutes les variables de première étape sont continues,  $I$  si elles sont toutes entières ( $B$  si elles sont tout binaires), ou  $M$  si certaines variables de la première étape sont continues et d'autres sont entières (problèmes mixtes). De même, la deuxième lettre caractérise les variables de seconde étape avec une représentation identique. La troisième lettre est soit  $D$  quand  $\xi$  a un support fini (distribution discrète), soit  $C$  quand  $\xi$  est une variable aléatoire continue. Ainsi,  $I/C/D$  représente un programme stochastique en nombres entiers (toutes les variables doivent être entières) à deux étapes avec une distribution discrète.

Dans cette section, nous supposons la programmation Linéaire stochastique en nombres entiers de la classe  $I/C/D$  présenté à une étape donnée qui dit le problème courant (actuel) :

$$(P_l) \left\{ \begin{array}{l} \min cx + \theta \\ \text{s.c} \\ Ax = b \\ D_l x \geq d_l, \quad l = 1, \dots, N_l \\ E_l x + \theta \geq e_l, \quad l = l, \dots, M_l \\ x \geq 0, \theta \in \mathbb{R} \\ x \text{ vecteur entier} \end{array} \right.$$



### 3.8.1 Méthode "Integer L-shaped"

La version de la méthode "L-Shaped" où les variables de la première étape sont entières (binaires) a été présentée par Laporte et Louveaux en 1993 [70]. Le choix des variables de la première étape se fait par branch & bound alors que les coupes au second niveau (d'optimalité et de faisabilité) se calculent comme avant.

**Algorithme 3.2** (*IntegerL-shaped*)

**Étape 1.**  $N_l := M_l := l := 0, \bar{z} := \infty, \theta = -\infty$  ; le seul nœud non sondé correspond au problème initial

**Étape 2.** Sélectionner un nœud non sondé dans la liste et aller à l'Étape 3 ; s'il n'existe pas, terminer.

**Étape 3.** Faire  $l := l + 1$  ; résoudre le problème relaxé actuel :

$$\left\{ \begin{array}{l} \min cx + \theta \\ s.c \\ Ax = b \\ D_l x \geq d_l, \quad l = 1, \dots, N_l \\ E_l x + \theta \geq e_l, \quad l = l, \dots, M_l \\ x \geq 0, \theta \in \mathbb{R} \end{array} \right.$$

où  $D_l x \geq d_l$  coupe de faisabilité et  $E_l x + \theta \geq e_l$  coupe d'optimalité..

$N_l$  est l'ensemble des indices d'itération où les contraintes de faisabilité ont été générées.

$M_l$  est l'ensemble des indices d'itération où les contraintes d'optimalité ont été générées.

Si le problème actuel n'a pas de solution réalisable, alors le nœud  $l$  est sondé, aller à l'Étape 2. Sinon,  $(x^l, \theta^l)$  une solution optimale. Aller à l'Étape 4.

**Étape 4.** Pour  $r = 1, \dots, R$ , résoudre le programme (3.20) en utilisant la solution  $(x^l, \theta^l)$  ; trouver  $\sigma^r$  une solution de (3.20) et déterminer les valeurs du vecteur  $D_r = (\sigma^l)^t T(\xi^r)$  et du coefficient  $d_r = (\sigma^l)^t h(\xi^r)$ . Si ce programme n'est pas faisable pour un scénario  $r$  alors générer la coupe de faisabilité  $D_r x \geq d_r$  et l'ajouter au problème  $(P_l)$ . Faire  $N_l := N_l + 1$ , aller à l'Étape 3. Sinon (si faisable pour tout les scénarios) aller à l'Étape 5.

**Étape 5.** Vérifier les restrictions d'intégralité. En cas de violation, créer deux nouvelles

branches en suivant la procédure de séparation et évaluation ; ajouter les nouveaux nœuds à la liste des nœuds suspendus; retourner à l'Étape 2. Sinon, aller à l'Étape 6.

**Étape 6.** Calculer  $Q(x^l)$  et  $\bar{z} = cx^l + Q(x^l)$ . Si  $z^l < \bar{z}$ , mettre à jour  $\bar{z}$ . Aller à l'Étape 7.

**Étape 7.** Si  $\theta^l \geq e_l - E_l x^l$ , alors sonder le nœud  $l$  et aller à l'Étape 2. Sinon ajouter au problème actuel la coupe d'optimalité  $E_l x + \theta \geq e_l$ , faire  $M_l := M_l + 1$  et aller à l'Étape 3.

## 3.9 Conclusion

Nous avons présenté les principaux résultats bibliographiques relatifs à la prise en compte d'incertitude affectant les coefficients de programmes linéaires. Ainsi, ne se retrouvant pas dans le domaine déterministe, nous sommes passés au monde stochastique. La façon dont les données sont disponibles détermine la représentation mathématique du problème. Nous avons introduit dans le présent chapitre différentes manières de modéliser des problèmes stochastiques en observant le contexte dans lequel chaque modélisation peut être justifiée. Parmi les modèles on retrouve les modèles avec des contraintes probabilistes et aussi les modèles en deux étapes avec recours. Nous avons passé en revue deux méthodes de résolution : "L-shaped" pour les problèmes continus et "Integer L-shaped" pour les problèmes discrets.

# Chapitre 4

## Optimisation linéaire multi-objectifs

### 4.1 Introduction

Un programme linéaire multi-objectifs (MOLP : *MultiObjective Linear Programming*) est constitué d'un système de contraintes linéaires définissant un domaine de solutions réalisables et d'un ensemble de fonctions linéaires à optimiser (maximiser ou minimiser) définissant des objectifs conflictuels.

Contrairement à l'optimisation mono objectif, la solution d'un problème multi-objectifs n'est pas une solution unique, mais un ensemble de solutions, connu comme l'ensemble des solutions efficaces, non dominantes ou pareto optimales<sup>(1)</sup>. Toute solution de ce type est optimale dans le sens qu'aucune amélioration ne puisse être faite sur un objectif sans dégradation d'au moins un autre objectif ([92], [41], [46], [66]).

Si les variables sont astreintes à ne prendre que des valeurs entières, on parle de programme linéaire multi-objectifs en nombres entiers (MOILP : *MultiObjective Integer Linear Programming*).

---

<sup>(1)</sup>Dans les problèmes multi-objectifs, on ne parle plus de solutions optimales, mais plutôt de solutions efficaces réalisant un bon compromis.

Plusieurs travaux concernant les méthodes de résolution de MOILP ont été publiées au cours de ces dernières trente années. Teghem et Kunsch ([96], [97], 1986) ont présenté une étude sur les méthodes interactives pour la programmation linéaire multi-objectifs (entier et mixte) publiée jusqu'à la fin de 1985 (la première méthode date de 1980). Durant la même période, une autre approche est due à Rasmussen ([79], 1986), en mettant l'accent sur la programmation multi-objectifs en 0-1, par considération des méthodes interactives et non-interactives. Clímaco et al. ([40], 1997) ont présenté un aperçu sur des méthodes interactives pour un cas particulier de MOILP (mixtes), où environ vingt méthodes interactives sont caractérisées et résumées.

## 4.2 Optimisation mathématique multi-objectifs

Nous définissons un problème multi-objectifs comme un problème de décision qui consiste à optimiser (maximiser ou minimiser) simultanément  $k$  fonctions objectifs notées  $f_i$ ,  $i = 1, \dots, k$  où  $k \geq 2$ , sur un ensemble de solutions  $S$ .

Ce problème peut être formulé mathématiquement comme suit :

$$\left\{ \begin{array}{l} \text{“opt”} \\ x \in S \end{array} [f_1(x), \dots, f_k(x)] \right.$$

Le symbole “ ” signifie qu'il n'est généralement pas possible de trouver dans  $S$  une action qui optimise simultanément les  $k$  objectifs, puisqu'on rencontre souvent des objectifs contradictoires<sup>(2)</sup>.

- Les  $n$  variables de décision  $x_i$  ( $i = 1, \dots, n$ ) constituent une solution représentée par le vecteur de décision  $x = (x_1, x_2, \dots, x_n)^t$
- $f(x) = (f_1(x), f_2(x), \dots, f_k(x))$  est le vecteur de  $k$  critères  $f_i, i = 1, \dots, k$ .
- L'ensemble d'actions  $S$  est un sous ensemble de  $\mathbb{R}^n$  décrit implicitement par des inéquations et/ou des équations, appelées *contraintes*.

---

<sup>(2)</sup>Deux objectifs sont contradictoires lorsque la diminution d'un objectif entraîne une augmentation de l'autre objectif.

- L'ensemble  $Z_S = f(S)$  qui est la projection de l'espace  $S$  sur l'espace des critères.
- L'ensemble  $\mathbb{R}^n$  qui contient  $S$  est dit *espace de décision*,
- L'ensemble  $\mathbb{R}^k$  qui contient  $Z_S$  est dit *espace des critères*,
- Si les  $k$  fonctions objectifs sont linéaires en  $x$ , i.e  $f_i(x) = c^i x, \forall i = 1, \dots, k$ , nous obtenons un problème de MOLP

$$\left\{ \begin{array}{l} \min Z^1(x) = c^1 x \\ \min Z^2(x) = c^2 x \\ \vdots \\ \min Z^k(x) = c^k x \\ \text{s.c} \\ x \in S \end{array} \right. \quad (4.1)$$

avec  $S = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$ ;  $A, c^i$  (pour tout  $i = \{1, \dots, k\}$ ),  $x$  et  $b$  sont des matrices déterministes de dimensions respectives  $(m \times n)$ ,  $(1 \times n)$ ,  $(n \times 1)$  et  $(m \times 1)$ , où  $n$  est le nombre de variables,  $m$  est le nombre de contraintes du système, et  $k$  est le nombre de fonctions objectifs.

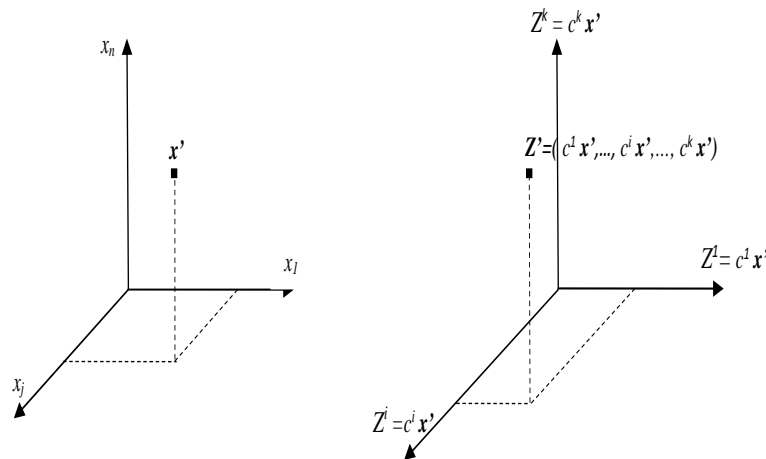


Figure 4.1 : L'espace des décisions à l'espace des objectifs

### 4.2.1 Points particuliers

#### Point “idéal”

Les coordonnées de point idéal  $Z^{(I)}$  du problème (4.1) sont obtenues en optimisant chaque fonction objectif séparément.

$$Z^{(I)} = (\hat{Z}^1, \dots, \hat{Z}^k) \in \mathbb{R}^k \text{ tel que : } \forall i = 1, \dots, k, \hat{Z}^i = \min_{x \in S} \{Z^i(x) = c^i x\}.$$

#### Points “nadir”

Les coordonnées de point nadir  $Z^{(N)} = (\overset{\circ}{Z}^1, \dots, \overset{\circ}{Z}^k)$  du problème (4.1) correspondent aux pires valeurs obtenues par chaque fonction objectif, lorsque l'on restreint l'espace des solutions à la surface de compromis.

Soit  $x_i^*$  une solution optimale obtenue en optimisant le critère  $Z^i$  sur  $S$ .

$$\hat{Z}^i = \min_{x \in S} \{Z^i(x) = c^i x\} = c^i x_i^*, \forall i = 1, \dots, k$$

et

$$\overset{\circ}{Z}^i = \max_{x \in S/Z^j = \hat{Z}^j} \{c^i x\}, \forall i = 1, \dots, k, \text{ avec } i \neq j.$$

- En particulier, le point Nadir  $Z^{(N)}$  est défini en bi-objectif par :

$$Z^{(N)} = (\overset{\circ}{Z}^1, \overset{\circ}{Z}^2) \text{ tel que } \forall i = \{1, 2\}, \overset{\circ}{Z}^i = \max_{x \in S/Z^j = \hat{Z}^j} \{c^i x\} \text{ avec } i \neq j$$

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

### 4.2.2 Solutions efficaces et points non dominés

En raison de la nature conflictuelle des fonctions objectifs, il n'existe pas de solution admissible optimisant tous les objectifs simultanément. Par conséquent, la notion mono-objectif de solution optimale ne s'applique donc plus. L'optimalité dans un contexte multi-objectifs est basée sur la notion de dominance et d'efficacité au sens de Pareto dont nous rappelons le formalisme dans les définitions suivantes restreintes au cas de minimisation vu dans (4.1).

**Définition 4.1 (Dominance)** Soient deux vecteurs critères  $Z, Z' \in \mathbb{R}^k$ . On dit que  $Z$  domine  $Z'$  si et seulement si  $Z^i \leq (Z')^i$  pour tout  $i \in \{1, \dots, k\}$  et  $Z^i < (Z')^i$  pour au moins un indice  $i \in \{1, \dots, k\}$ .

Autrement dit,  $Z$  est au moins aussi bon que  $Z'$  sur tous les critères, et meilleur que lui sur au moins un objectif.

**Définition 4.2 (Dominance forte)** Soient deux vecteurs critères  $Z, Z' \in \mathbb{R}^k$ . On dit que  $Z$  domine fortement  $Z'$  si et seulement si  $Z^i \leq (Z')^i$  pour tout  $i \in \{1, \dots, k\}$ . Si  $Z$  domine fortement  $Z'$ , alors  $Z$  est meilleur que  $Z'$  sur tous les objectifs.

**Définition 4.3 (Solution efficace)** Une solution  $\hat{x} \in S$  est une solution efficace pour le problème (4.1) s'il n'existe pas de solution  $x \in S$ , telle que  $cx \leq c\hat{x}$  et  $cx \neq c\hat{x}$ . Une solution  $\hat{x} \in S$  est dite efficace si son vecteur objectif n'est pas dominé par le vecteur objectif d'une autre solution de  $S$ . On dit aussi que  $\hat{x}$  est : admissible, optimum de Pareto ou non inférieure.

**Définition 4.4 (Solution faiblement efficace)** Une solution  $\hat{x} \in S$  est dite faiblement efficace pour le problème (4.1) s'il n'existe aucune autre solution admissible  $x \in S$  telle que  $cx < c\hat{x}$ . On dit alors que son vecteur objectif  $c\hat{x}$  n'est pas fortement dominé.

**Définition 4.5 (Solution fortement efficace)** Une solution  $\hat{x} \in S$  est une solution fortement efficace pour le problème (4.1) s'il n'existe pas de solution  $x \in S$  telle que  $x \neq \hat{x}$  et  $cx \leq c\hat{x}$ .

Une solution  $\hat{x}$  est fortement efficace s'il n'existe pas de solution telle que le vecteur objectif, qui lui est associé, soit aussi bon que celui de  $x$ .

Remarquons que l'efficacité forte implique l'efficacité, qui implique à son tour l'efficacité faible.

**Théorème 4.1 (Caractérisation d'une solution efficace [25])** Soit  $x^*$  une solution réalisable pour le problème (4.1).  $x^*$  est une solution efficace pour le problème (4.1) si et seulement si la valeur optimale de la fonction objectif  $\varphi$  est nulle dans le programme linéaire suivant :

$$(PE) \begin{cases} \max \varphi = \sum_{i=1}^{i=k} v_i \\ c^i x + v_i = c^i x^*, i = 1, \dots, k \\ x \in S \\ v_i \geq 0, i = 1, \dots, k \end{cases}$$

Dans nos travaux, nous cherchons à calculer un ensemble non dominé complet, ce qui s'apparente au contexte a posteriori.

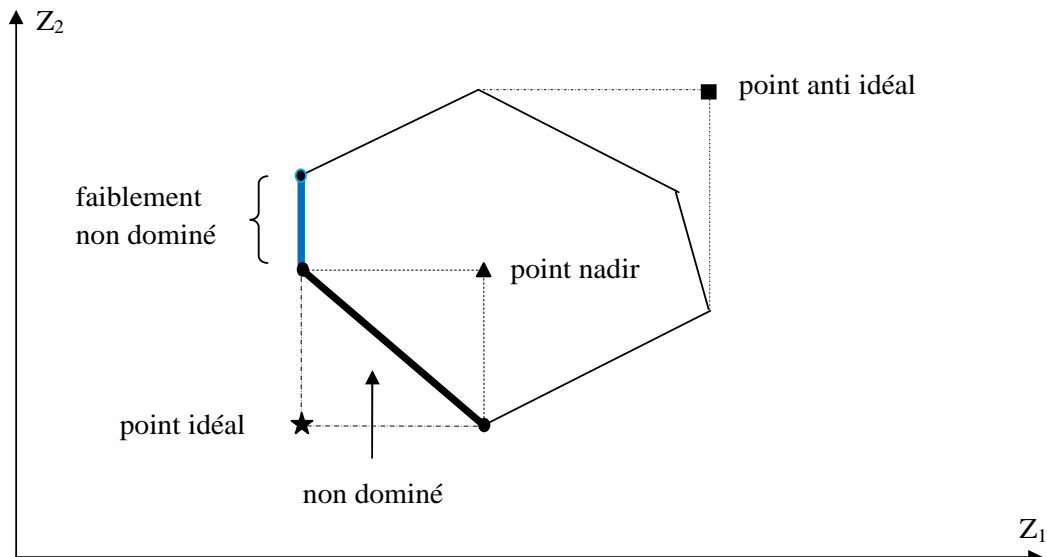


Figure 4.2 : Illustration des définitions

Dans la figure 4.2 , il n'y a pas un seul point non dominé mais plutôt un ensemble de points non dominés.

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



### 4.3 La programmation linéaires multi-objectifs en variables entières

Un programme linéaire multi-objectifs en nombres entiers MOILP peut être formulé comme suit :

$$\left\{ \begin{array}{l} \min Z^1(x) = c^1x, \\ \min Z^2(x) = c^2x, \\ \vdots \\ \min Z^k(x) = c^kx, \\ Ax = b, \\ x \geq 0, \\ x \text{ vecteur entier.} \end{array} \right. \quad (4.2)$$

avec  $A$  une  $m \times n$ -matrice,  $b$  un  $m$ -vecteur à coefficients entiers et  $c^i$  un vecteur ligne de  $\mathbb{R}^n$ , pour tout  $i = 1, \dots, k$ .

Désignons par :

$\mathbf{S}$  : L'ensemble des solutions réalisables entières de (4.2).

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

À chaque solution réalisable  $x$  dans  $\mathbf{S}$ , on associe son image  $Z(x) = \begin{pmatrix} Z^1(x) \\ Z^2(x) \\ \vdots \\ Z^k(x) \end{pmatrix}$  dans  $\mathbb{R}^k$  et

on construit donc, l'ensemble :  $Z_S = \{Z \in \mathbb{R}^k : Z = cx, x \in S\}$ , où  $c$  est la  $k \times n$ -matrice composée des vecteurs lignes  $c^i$  de  $\mathbb{R}^n$ ,  $i = 1, \dots, k$ .

Supposons que l'ensemble  $S$  est un polyèdre convexe de  $\mathbb{R}^n$  et le caractère linéaire des critères permet d'établir que  $Z_S$  est également un polyèdre de  $\mathbb{R}^k$ , dont les points extrêmes correspondent à ceux de  $S$ .

### 4.3.1 La méthode de pondération des fonctions objectifs

Cette méthode est une des premières méthodes élaborées pour la génération de solutions efficaces. Son principe de base consiste à transformer le problème multi-objectifs en un problème mono objectif, en additionnant toutes les fonctions objectifs après avoir affecté un coefficient de poids à chacune d'elles. Ces coefficients représentent l'importance relative que le décideur attribue à chaque objectif, donnant lieu à une seule fonction objectif définie par la somme pondérée des  $k$  objectifs initiaux, telle que le problème (4.2) se transforme de la manière suivante :

$$(P_\lambda) \begin{cases} \min \sum_{i=1}^{i=k} \lambda_i [c^i x] \\ x \in S. \end{cases} \quad (4.3)$$

où les poids vérifient :  $\lambda_i \geq 0, \lambda_i \in [0, 1]$  et  $\sum_{i=1}^{i=k} \lambda_i = 1$ .

**Théorème 4.2 (Geoffrion [50])**  $x^*$  est une solution efficace pour (4.1) si et seulement s'il existe  $\lambda \in \Lambda = \left\{ \lambda \in \mathbb{R}^k : \lambda_i \geq 0, i = 1, \dots, k, \sum_{i=1}^{i=k} \lambda_i = 1 \right\}$ , tel que  $x^*$  est une solution optimale du problème paramétrique  $(P_\lambda)$  (le problème (4.3)).

### 4.3.2 Solutions efficaces supportées et non supportées

D'après le théorème 4.2, les solutions efficaces du problème (4.1) obtenues en résolvant le problème paramétrique  $(P_\lambda)$ ,  $\lambda \in \Lambda$  sont appelées **solutions efficaces supportées**. Différents poids fournissent différentes solutions supportées. Les solutions efficaces du problème (4.1) qui ne sont pas solutions du problème  $(P_\lambda)$ ,  $\lambda \in \Lambda$  sont appelées **solutions efficaces non supportées**.

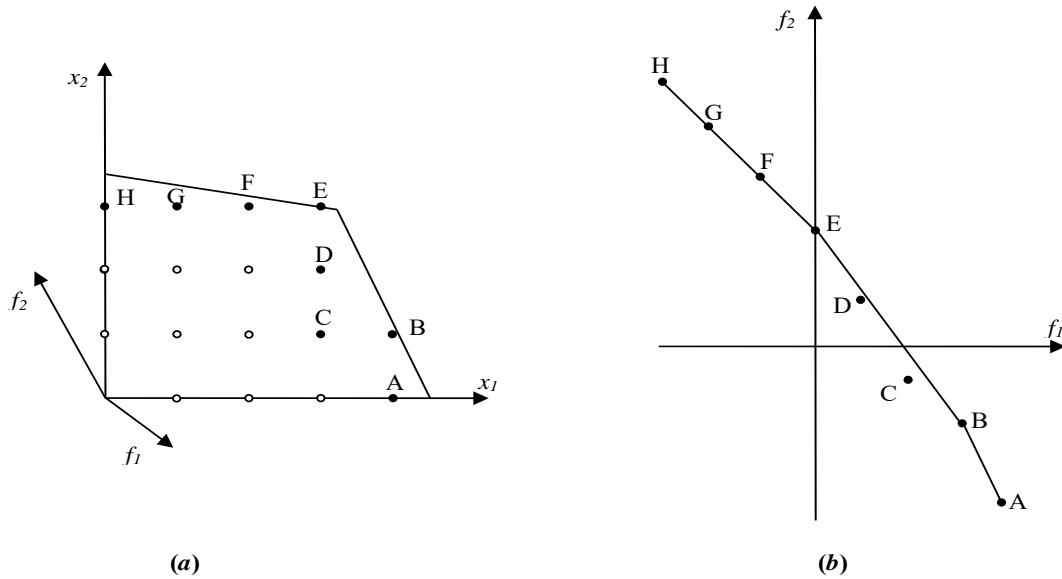


Figure 4.3 : Représentation des solutions dans l'espace de décisions et des objectifs

La figure 4.3 illustre la différence entre ces types de solutions dans (4.2) :

(a) montre le problème dans l'espace de décision, alors que (b) indique les solutions non dominées correspondantes dans l'espace des objectifs. Les solutions A, B, E, F, G et H sont des solutions non dominées supportées (situées sur la frontière de l'enveloppe convexe de  $Z_S$ ) et parmi celles-ci, A, B, E et H sont les solutions extrêmes de l'enveloppe convexe  $Z_S$  de la frontière efficace, tandis que F et G sont les solutions non extrêmes. De plus, C et D sont des solutions non dominées et non supportées.

L'optimisation d'une simple somme pondérée de fonctions objectifs (4.3) à l'aide de la méthode classique B&B, pour la programmation linéaire en nombres entiers (4.2), donne des solutions extrêmes non dominées supportées (A, B, E et H). Une exploration plus approfondie de l'arborescence de B&B permet de calculer des solutions non dominées supportées mais non extrêmes (F et G).

### 4.3.3 Méthodes graphiques pour la résolution du problème MOILP

#### Méthode basée sur le concept d'ensembles dominants

Il s'agit dans ce paragraphe de reprendre dans le cas de variables discrètes un concept introduit par R.E. Steuer [92] utilisé pour détecter graphiquement l'ensemble des solutions efficaces d'un problème MOILP dans  $\mathbb{R}^2$  et  $\mathbb{R}^3$ .

Pour tester l'efficacité en un point  $x \in S$ , R.E. Steuer a introduit le concept d'ensembles dominants au sens de la définition 4.1.

**Définition 4.6** Soit  $\hat{x} \in S$ . Nous définissons comme étant  $C^{\geq}$  la région semi-positive du cône généré par les gradients des  $k$  fonctions objectifs où  $C^{\geq} = \{y \in \mathbb{Z}^n : Cy \geq 0, Cy \neq 0\} \cup \{0 \text{ de } \mathbb{Z}^n\}$ .

**Définition 4.7** L'ensemble dominant  $D_{\hat{x}} = \{\hat{x}\} \oplus C^{\geq}$  est donné par la somme des ensembles  $\{\hat{x}\}$  et  $C^{\geq}$ .

Une autre façon d'écrire l'ensemble dominant est  $D_{\hat{x}} = \{x \in \mathbb{Z}^n : x = \hat{x} + y, Cy \geq 0, Cy \neq 0\}$ . L'ensemble dominant contient tout les points dont les vecteurs critères dominent le vecteur critère de  $\hat{x} \in S$ .

Le théorème 4.3 montre l'importance de l'ensemble dominant  $D_{\hat{x}}$  dans la détection des points efficaces.

**Théorème 4.3** [92] Soit  $D_{\hat{x}}$  l'ensemble dominant en  $\hat{x} \in S$ . Alors,  $\hat{x}$  est efficace si et seulement si  $D_{\hat{x}} \cap S = \{\hat{x}\}$ .

Le théorème 4.3 fournit un test permettant de détecter les points efficaces et pouvant être géométriquement visualisé : Si l'intersection de l'ensemble dominant avec la région réalisable contient seulement  $\hat{x}$ , alors  $\hat{x}$  est efficace. S'il existe d'autres points appartenant à l'intersection de ces deux ensembles, alors  $\hat{x}$  est inefficace.

**Exemple 4.1** Dans la figure 4.4,  $x^1$  n'est pas une solution efficace parce que  $D_{x^1} \cap S \neq \{x^1\}$ ,  $x^2$  est une solution efficace parce que  $D_{x^2} \cap S = \{x^2\}$ .

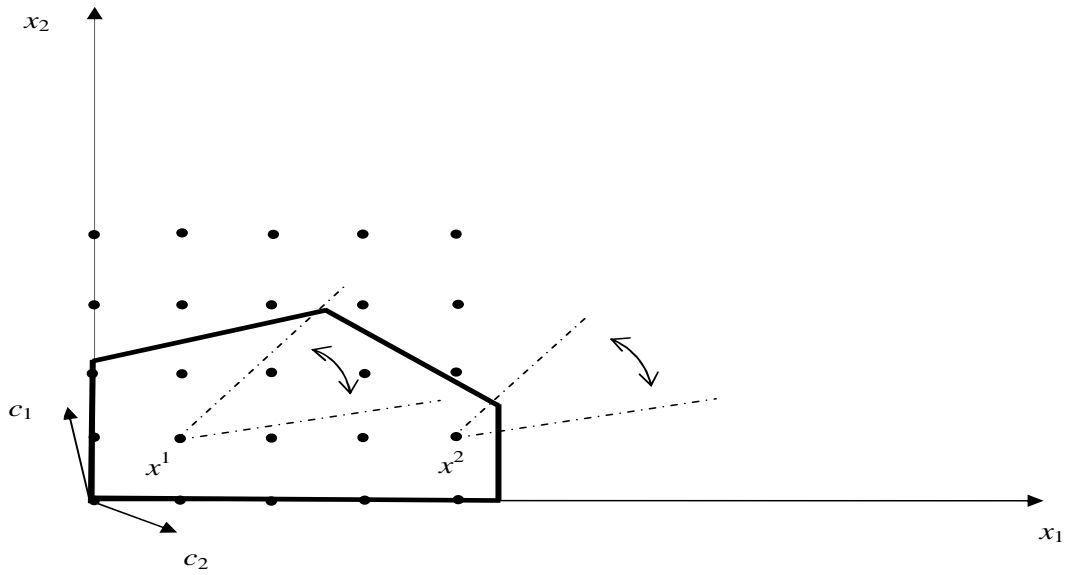


Figure 4.4 : Solutions efficaces

### Méthode basée sur le concept d'ensembles dominés

Pour tester l'efficacité en un point  $\hat{x} \in S$ , on introduit le concept d'ensembles dominés au sens des définitions 1.3.9.

**Définition 4.8** Soit  $\hat{x} \in S$  et soit  $C^{\leq}$  la région semi-négative du cône généré par les gradients des  $k$  fonctions objectifs où :

$$C^{\leq} = \{y \in \mathbb{R}^n : Cy \leq 0, Cy \neq 0\}$$

**Définition 4.9** L'ensemble dominé  $\hat{D}_{\hat{x}} = \{\hat{x}\} \oplus C^{\leq}$  est donné par la somme des ensembles  $\{\hat{x}\}$  et  $C^{\leq}$ .

Nous définissons aussi les ensembles :  $U = \{x \in \mathbb{Z}^n : Cx = 0\}$  et  $U_{\hat{x}} = \{\hat{x}\} \oplus U$ .

#### 4.3.4 Méthode de Abbas et Moulai [1]

Utilisant les coupes fractionnaires de Gomory, l'algorithme développé par les auteurs range les solutions réalisable du problème (4.2) dans l'ordre décroissant des valeurs de l'un des objectifs (ranking), sans omettre les solutions entières alternatives.

On considère le problème :

$$(P_1) : \min \{ Z^1 = C^1 x : x \in \mathbf{S} \}$$

avec  $\mathbf{S} = \{x \in \mathbb{Z}^n : Ax = b, x \geq 0\}$ , dont le problème relaxé est

$$(P_1R) : \min \{ Z^1 = C^1 x : x \in S \}$$

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

Pour des besoins relatifs à la description de la méthode, on définit les paramètres suivants pour  $l \geq 1$  :

$$S_l = \{x \in \mathbb{R}^{n_l} : A_l x = b_l; A_l \in \mathbb{R}^{n_l \times m_l}; b_l \in \mathbb{R}^{m_l}; x \geq 0\}$$

comme étant la région courante tronquée de  $S$  obtenue par application de la coupe :

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

et éventuellement des coupes successives de Gomory, où  $j_{l-1}$  un indice hors base quelconque.

- $x^l$  : la  $l^{\text{ème}}$  solution optimale entière du problème  $(P_1)$  obtenue sur  $S_l$ .
- $B_l$  : une base de  $S_l$ .
- $N_l$  : l'ensemble des indices des variables hors base de  $x^l$ .
- $\hat{A}_l = (A_{B_l})^{-1} A_l = (\hat{a}_{f(i)j}), i = 1, \dots, m_l; j = 1, \dots, n_l$  où  $f(i) \in \{1, \dots, n_l\}$  est l'indice de la variable de base associée à la ligne  $i$  ;  $i \in \{1, \dots, m_l\}$  à l'étape  $l$ .
- $\hat{C}_l = C_l - \pi_l A_l$ , avec  $\pi_l = (C_{B_l})(A_{B_l})^{-1}$ .

- $\Gamma_l = \{j \in N_l : (\hat{c}_l^1)_j < 0 \text{ et } (\hat{c}_l^i)_j > 0 \text{ pour au moins un critère } i ; i \in \{2, \dots, k\}\}.$
- $\Omega_l = \{j \in N_l : (\hat{c}_l^1)_j = 0\}.$
- $\Psi_l = \{j \in N_l : (\hat{c}_l^1)_j < 0 \text{ et } (\hat{c}_l^i)_j < 0 \text{ pour au moins un critère } i ; i \in \{2, \dots, k\}\}.$

**Définition 4.10** Une arête  $E^{j_l}$ ,  $j_l \in N_l$ , incidente à  $x^l$  est définie comme étant l'ensemble suivant :

$$E^{j_l} = \left\{ x \in \mathbb{R}^{n_l} : \begin{cases} x_i = x_i^l - \theta_{j_l} \hat{a}_{f(i)j_l}, i \in B_l \\ x_{j_l} = \theta_{j_l} \\ x_l = 0, \forall l \in N_l \setminus \{j_l\} \end{cases} \right\}$$

où :  $0 \leq \theta_{j_l} \leq \theta = \min_{i \in B_l} \left\{ \frac{x_i^l}{\hat{a}_{f(i)j_l}} : \hat{a}_{f(i)j_l} > 0 \right\}.$

Les points entiers se trouvant sur l'arête  $E^{j_l}$  sont identifiés de telle sorte que  $\theta_{j_l}$  soit entier et  $\theta_{j_l} \times \hat{a}_{f(i)j_l}$  entier,  $\forall i \in B_l$ .

On note par  $nb_{j_l}$  le nombre de solutions entières se trouvant sur l'arête  $E^{j_l}$  y compris  $x^l$ .

Notons par  $SND$  l'ensemble des solutions potentiellement non dominées de (4.2) générées jusqu'à l'étape  $l$  ;  $l \geq 1$ .

**Algorithme 4.1 (AM)**

**Étape 1.** Résoudre le problème  $(P_1)$  et trouver la solution optimale entière  $x^1$  sur  $S_1$ .

Construire l'ensemble  $\Omega_1$ .

**Étape 2.** Si  $\Omega_1 = \emptyset$ ,  $x^1$  est l'unique solution optimale sur  $S_1$ .

Soit  $(z_1^1; z_1^2, \dots, z_1^k)$  le vecteur critère correspondant, alors  $SND := \{(z_1^1; z_1^2, \dots, z_1^k)\}.$

Tronquer le point  $x^1$  par la coupe de Dantzig :

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

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

Si  $\Omega_1 \neq \emptyset$ , choisir un indice quelconque  $j_1 \in \Omega_1$  et calculer le nombre  $\theta$  de l'opération pivot.

(a) Si  $\theta \geq 1$ , déterminer toutes les solutions entières alternatives à  $x^1, y_1^q; q = 2, \dots, nb_{j_1}$ , le long de l'arête  $E^{j_1}$  et mettre à jour SND.

Comme les solutions alternatives ont la même valeur de  $z^1$  que celle de  $x^1$ , le premier point potentiellement non dominé est choisit comme le  $k$ -uplet ayant la plus grande valeur de  $z^2$ , sinon choisir celui qui a la plus grande valeur de  $z^3$  et ainsi de suite jusqu'à l'obtention du premier  $k$ -uplet potentiellement non dominé. Tronquer l'arête  $E^{j_1}$  par la coupe :

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

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

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

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

**Étape l.** ( $l \geq 3$ )

Choisir un indice  $j_{l-1} \in \Omega_{l-1}$  et explorer l'arête correspondante à la recherche de solutions entières  $y_{l-1}^q; q = 2, \dots, nb_{j_{l-1}}$  alternatives à  $x^{l-1}$ . Mettre à jour l'ensemble SND.

L'arête  $E^{j_{l-1}}$  est tronquée par la coupe :  $\sum_{j \in N_{l-1} \setminus \{j_{l-1}\}} x_j \geq 1$

Après application de la méthode duale du simplexe et éventuellement, des coupes successives de Gomory, la solution optimale entière obtenue sur la région  $S_l$  est  $x^l$ .

**Test d'arrêt.** Le processus se termine quand l'opération pivot de la méthode duale du simplexe devient impossible, indiquant que la région courante ne contient aucun point entier.



### 4.3.5 Méthode de Chergui et Moulai [36]

La méthode de coupes multiples pour l'élimination d'un ensemble dominé a été proposée par Chergui et Moulai en 2008 , ils ont élaborés un algorithme d'énumération de toutes les solutions entières non dominées. C'est une méthode B&B qui fait appel aux coupes efficaces pour passer d'un vecteur entier à un autre, elle fait appel à la méthode du simplexe et dual du simplexe pour résoudre le  $PL$  suivant :

$$(P_0) \left\{ \begin{array}{l} \text{"min"} Z^1(x) = c^1 x \\ Ax = b \\ x \geq 0 \\ x \text{ vecteur entier} \end{array} \right. \quad (4.4)$$

avec  $A \in \mathbb{Z}^{m+n}$  ;  $b \in \mathbb{Z}^m$

On suppose que le domaine de réalisabilité  $S$  est non vide et compact.

Soit  $x_l^*$  une solution entière de (4.2), on associe à  $x_l^*$  les paramètres suivants :

$B_l$  : ensemble des indices des variables de base,

$N_l$  : ensemble des indices des variables hors base,

$H_l$  : ensemble des directions possibles d'amélioration des critères,

$$H_l = \{j \in N_l : \exists i \in \{1, \dots, k\} ; \hat{c}_j^i < 0\} \cup \quad (4.5)$$

$$\{j \in N_l : \hat{c}_j^i = 0, \forall i \in \{1, \dots, k\}\} \quad (4.6)$$

avec  $\hat{c}_j^i$  la  $j^{\text{ème}}$  coordonnée du coût réduit du  $i^{\text{ème}}$  critère.

La relation suivante définit une **coupe efficace**,

$$\sum_{j \in H_l} x_j \geq 1 \quad (4.7)$$

du fait que, pour le problème (4.2), son adjonction à  $S$  supprime au moins une solution réalisable continue de  $S$ , sans supprimer des solutions réalisables entières efficaces de  $S$ .

Les ensembles suivant :

$$S_0 = S \text{ et } S_{l+1} = \left\{ x \in S_l : \sum_{j \in H_l} x_j \geq 1 \right\}$$

avec  $\sum_{j \in H_l} x_j \geq 1$  une coupe efficace pour le problème (4.2).

On définit le programme paramétrique :

$$(P_l) \begin{cases} \text{“min” } Z^1(x) = c^1 x \\ x \in S_l \end{cases} \quad (4.8)$$

L’algorithme de génération de toutes les solutions entières efficaces de (4.2) est présenté dans les étapes suivantes :

**Algorithme 4.2** (*Solutions efficaces pour MOILP*)

**Étape 1.** *Initialisation*

$S_0 := S$  ;  $l := 0$  ;  $Eff := \emptyset$  (*Eff ensemble des solutions entières efficaces de (4.2)*)

Résoudre le programme linéaire  $(P_0)$  au nœud 0 : Soit  $\bar{x}_0$  la solution optimale obtenue.

Si  $\bar{x}_0$  n’est pas entière, aller à l’Étape 3. Sinon, aller à l’Étape 4.

**Étape 2.** *Étape générale*

Tant qu’il existe un nœud non encore sondé dans l’arborescence, choisir le nœud de plus grand indice  $l$  non encore sondé et résoudre le programme linéaire correspondant  $(P_l)$ .

Si  $(P_l)$  est non réalisable, alors le nœud  $l$  est sondé.

Sinon, soit  $x_l$  une solution optimale de  $(P_l)$ . Si  $x_l$  n’est pas entière, aller à l’Étape 3.

Sinon, poser  $x_l^*$  la solution entière trouvée et aller à l’Étape 4.

**Étape 3.** Soit  $x_{lj}$  une coordonnée fractionnaire de  $x_l$ .

Séparer le nœud  $l$  en deux nouveaux nœuds :

Ajouter la contrainte  $x_j \leq \lfloor x_{lj} \rfloor$  au premier nœud, la contrainte  $x_j \geq \lfloor x_{lj} \rfloor + 1$  au second nœud et aller à l’Étape 2.

**Étape 4.** Si  $cx_l^*$  n’est pas dominé par  $cx$  pour toute solution  $x \in Eff$ , alors

$$Eff := Eff \cup \{x_l^*\}.$$

Si’il existe  $x \in Eff$  telle que  $cx$  est dominé par  $cx_l^*$ , alors

$$Eff := [Eff \setminus \{x\}] \cup \{x_l^*\}.$$

Déterminer les ensembles  $B_l$ ,  $N_l$  et  $H_l$ . Si  $H_l = \emptyset$  ; alors le nœud correspondant est sondé, aller à l’Étape 2.

**Étape 4.** Rajouter la contrainte  $\sum_{j \in H_l} x_j \geq 1$  pour obtenir l’ensemble  $S_{l+1}$ .

*Résoudre le programme linéaire obtenu ( $P_{l+1}$ ) par la méthode dual simplexe et poser  $x_{l+1}^*$  la solution optimale trouvée. Si  $x_{l+1}^*$  entière, aller à l'Étape 4. Sinon, aller à l'Étape 3.*

## 4.4 Conclusion

Dans ce chapitre, nous avons passé en revue les méthodes d'optimisation multi-objectifs en nombres entiers en accordant une importance particulière aux méthodes utilisées dans nos travaux de recherche ; notamment la méthode de Chergui et Moulai [36] et la méthode de Abbas et Moulai [1].

# Chapitre 5

## Optimisation linéaire stochastique multi-objectifs

### 5.1 Introduction

La programmation linéaire stochastique a été étudiée par quelques auteurs pour le problème mono-objectif [35], [62], [59], [69], [60], [78], [27], [32], mais il existe moins d'articles traitant le cas multi-objectifs SMOLP [90], [94], [20], [31]. Ces problèmes sont souvent plus difficiles que leurs versions déterministe continue, sans doute en raison de l'ajout de l'incertitude à la difficulté des problèmes multicritères.

Certaines célèbres approches traitant ce problème le transforment d'abord en un programme déterministe multi-objectifs, qui est par la suite résolu par des méthodes interactives (comme celles présentées dans **PROTRADE** de Goicoechea et al.[54], **STRANGE** de Teghem et al.[99] et **PROMISE** d'Urli et Nadeau [102]). Ces méthodes ont été testées avec succès dans des contextes réels.

## 5.2 Programmation linéaire stochastique multi-objectifs

Considérons le programme linéaire stochastique multi-objectifs SMOLP donné sous la forme mathématique suivante :

$$\left\{ \begin{array}{l} \text{“min”}_x Z^i = c^i(\xi) x, \quad i = 1, \dots, k \\ Ax = b \\ T(\xi) x = h(\xi) \\ x \geq 0 \end{array} \right. \quad (5.1)$$

où :

$k$  est nombre de fonctions objectifs,  $k \geq 2$ .

$c^i(\xi)$ ,  $T(\xi)$  et  $h(\xi)$  sont des matrices aléatoires de dimensions respectives  $(1 \times n)$ ,  $(m_0 \times n)$  et  $(m_0 \times 1)$  définies sur un certain espace de probabilité  $(\Omega, \Xi, \mathbb{P})$ .

Les variables aléatoires dans  $c^i$ ,  $T$  et  $h$  ont une distribution discrète conjointe, définie par les réalisations  $(c^i(\xi^r), T(\xi^r), h(\xi^r))$ , avec la probabilités  $\mathbb{P}^r = \mathbb{P}(\xi^r)$ , où  $r = 1, \dots, R$ ,  $(\mathbb{P}^r > 0 \text{ et } \sum_{r=1}^{r=R} \mathbb{P}^r = 1)$ .

$x = (x_1, x_2, \dots, x_n)^t$  est le vecteur représentant les  $n$  variables de décision.

$A$  et  $b$  sont les matrices déterministes parce qu'ils reflètent l'état et les paramètres du système à l'instant présent, de diemnsions respectives  $m \times n$  et  $m \times 1$ .

Le fait que  $x$  ne dépend pas de  $\xi$  traduit la propriété de non-anticipativité de la prise de décision, c'est-à-dire que les décisions devant s'appliquer de suite ne peuvent pas varier en fonction d'informations non encore disponibles.

### 5.2.1 Concepts de solutions efficaces

Les concepts de solutions efficaces de SMOLP ont fait l'objet des ouvrages spécialisés suivants : [54] [91], [89], [17], [18], [31]. Considérons le problème (5.1)

$$\left\{ \begin{array}{l} \text{“min”}_x Z^i = c^i(\xi) x, \quad i = 1, \dots, k \\ x \in \mathbb{D} \end{array} \right. \quad (5.2)$$

avec  $\mathbb{D} = \{x : x \in \mathbb{R}^n, Ax = b, T(\xi) x = h(\xi), x \geq 0\}$  est non-vide, compact et convexe.

Dans ce cas, le concept vient de l'obtention de l'espérance de chaque objectif stochastique et décrit le problème multi-objectifs relatif à la minimisation de ces espérances.

**Définition 5.1 (Solution efficace en espérance)** Une solution  $\hat{x} \in \mathbb{D}$  est une solution efficace pour le problème (5.1), si elle est pareto efficace pour le problème :

$$\begin{cases} \text{“min” } \mathbb{E}[c^i(\xi)x], & i = 1, \dots, k \\ x \in \mathbb{D} \end{cases}$$

où  $\mathbb{E}[c^i(\xi)x]$  est l'espérance de la  $i^{\text{ème}}$  fonction objectif.

**Définition 5.2 (Solution efficace en variance minimum)** Une solution  $\hat{x} \in \mathbb{D}$  est une solution efficace avec une variance minimum pour le problème (5.1), si elle est efficace pour le problème :

$$\begin{cases} \text{“min” } \sigma^2[c^i(\xi)x], & i = 1, \dots, k \\ x \in \mathbb{D} \end{cases}$$

où  $\sigma^2[c^i(\xi)x] = \mathbb{V}[c^i(\xi)x]$  est la variance de la  $i^{\text{ème}}$  fonction objectif.

**Définition 5.3 (Solution  $\mathbb{E}\sigma$ -efficace)** Une solution  $\hat{x} \in \mathbb{D}$  est une solution d'espérance et d'écart-type efficace pour le problème (5.1) si elle est efficace pour le problème :

$$\begin{cases} \text{“min” } \mathbb{E}[c^i(\xi)x], & i = 1, \dots, k \\ \text{“min” } \sigma[c^i(\xi)x], & i = 1, \dots, k \\ x \in \mathbb{D} \end{cases}$$

où  $\sigma[c^i(\xi)x]$  est l'écart-type de la  $i^{\text{ème}}$  fonction objectif.

**Définition 5.4 (Solution efficace avec un risque minimum de niveaux  $\mu_1, \dots, \mu_k$ )**

Une solution  $\hat{x} \in \mathbb{D}$  est une solution efficace avec un risque minimum de niveaux  $\mu_1, \dots, \mu_k$  pour le problème (5.1) si elle est efficace pour le problème :

$$\begin{cases} \text{“max” } \mathbb{P}[c^i(\xi)x \leq \mu_i], & i = 1, \dots, k \\ x \in \mathbb{D} \end{cases}$$

**Définition 5.5 (Solution  $\beta$ -efficace)** Une solution  $\hat{x} \in \mathbb{D}$  est une solution efficace avec probabilités  $\beta_1, \dots, \beta_k$  (où une solution  $\beta$ -efficace) pour le problème (5.1), s'il existe  $\hat{\mu} = (\hat{\mu}_1, \dots, \hat{\mu}_k)$  tel que :  $\begin{pmatrix} \hat{x} \\ \hat{\mu} \end{pmatrix}$  est une solution efficace pour le problème :

$$\begin{cases} \text{“min”}_{x, \mu} \mu_i, & i = 1, \dots, k \\ s.c \\ \mathbb{P}(c^i(\xi) x \leq \mu_i) \geq \beta_i, & i = 1, \dots, k \\ x \in \mathbb{D} \end{cases}$$

où  $\beta_1, \dots, \beta_k$  sont les seuils de probabilité qui sont fixés à priori par le décideur.

### 5.3 Programmation linéaire stochastique multi-objectifs en nombres entiers

Un problème de la programmation linéaire stochastique multi-objectifs en nombres entiers SMOILP peut être formulé comme suit :

$$(P) \begin{cases} \text{“min”}_x Z^i = c^i(\xi) x, & i = 1, \dots, k \\ Ax = b \\ T(\xi) x = h(\xi) \\ x \in \mathbb{N}^n \end{cases} \quad (5.3)$$

où  $c^i, T$  et  $h$  sont des matrices aléatoires définies sur un espace de probabilité  $(\Omega, \Xi, \mathbb{P})$ ;  $c^i(\xi) \in \mathbb{R}^{1 \times n}$ ,  $T(\xi) \in \mathbb{R}^{m_0 \times n}$  et  $h(\xi) \in \mathbb{R}^{m_0 \times 1}$ .

$A$  et  $b$  sont les matrices déterministes, avec :  $A \in \mathbb{Z}^{m \times n}$  et  $b \in \mathbb{Z}^m$ .

### 5.3.1 Le problème déterministe équivalent de SMOILP

Considérons une distribution discrète et finie  $\{(\xi^r, \mathbb{P}^r), r = 1, \dots, R\}$  des données aléatoires où  $R$  est le nombre de réalisations. Progressivement, à chaque réalisation  $\xi^r$  de  $\xi$  nous associons  $k$  fonctions objectifs  $(Z^i)^r = c^i(\xi^r) x, i = 1, \dots, k$ , ensuite une matrice  $T(\xi^r)$  et enfin un vecteur  $h(\xi^r)$ ; en prenant en considération les scénarios différents qui affectent les  $k$  objectifs et les contraintes stochastiques.

Le même principe du recours utilisé dans la programmation stochastique mono-objectif avec la matrice de recours déterministe  $W$  est utilisée dans le présent document ([60], [27], [29]). Évidemment, nous supposons que le décideur peut indiquer d'une manière satisfaisante les pénalités  $q^r = q(\xi^r)$  des variables violées  $z^r, r = 1, \dots, R$  des contraintes, tout en préservant la dimension raisonnable du problème déterministe.

Contrairement à la méthode STRANGE, où une fonction objectif supplémentaire est créée pour pénaliser les contraintes violées, la fonction du recours  $Q(x, \xi^r)$  est ajoutée à chaque fonction objectif. Cette pénalité est donnée par :

$$Q(x, \xi^r) = \min_z \{(q^r)^t z : W(\xi^r) z = h(\xi^r) - T(\xi^r) x, z \geq 0\}.$$

Alors le décideur doit réduire au minimum la valeur d'espérance de tous les coûts :

$$\tilde{Z}^i = \mathbb{E}[Z^i + Q(x, \xi)], i = 1, \dots, k.$$

Nous obtenons le problème déterministe suivant :

$$\left\{ \begin{array}{l} \text{“min”}_x \tilde{Z}^i = \acute{Z}^i + Q(x), \quad i = 1, \dots, k \\ Ax = b \\ x \geq 0, x \text{ vecteur entier} \end{array} \right. \quad (5.4)$$

où

$$\acute{Z}^i = \mathbb{E}[Z^i] = \sum_{r=1}^R \mathbb{P}(\xi^r) Z_r^i = \sum_{r=1}^R \mathbb{P}^r c^i(\xi^r) x = \mathbb{E}[c^i(\xi) x]$$

et

$$Q(x) = \mathbb{E}[Q(x, \xi)] = \sum_{r=1}^R \mathbb{P}(\xi^r) Q(x, \xi^r) = \sum_{r=1}^R \mathbb{P}^r (q^r)^t z^r$$

sont respectivement les valeurs d'espérance de  $Z^i$  et de la fonction du recours  $Q(x, \xi)$ .



Nous nous attendons que lors de la deuxième étape, le programme  $Q(x, \xi^r)$  d'être réalisable pour toute réalisation de  $\xi^r$ ,  $r = 1, \dots, R$  de  $\xi$  selon  $(m_0 \times n_0)$ -matrice de recours  $W(\xi^r)$ . Ceci ne doit pas être vrai pour toutes les décisions  $x \in S_0 = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$  de la première étape.

### 5.3.2 Faisabilité

Supposons que la matrice du recours  $W$  est fixée. Nous clarifions la question “ *comment décider si une donnée  $x = x^0$  est réalisable* ”, pour la seconde étape du problème, pour toutes les réalisations possibles de  $\xi$ . Alors, il est plus simple de travailler avec le dual du programme de la deuxième étape :

$$\max_{\pi} \{ \pi^t [h(\xi^r) - T(\xi^r) x^0] : \pi^t W \leq (q^r)^t \} \quad (5.5)$$

L'ensemble des contraintes  $\mathbf{P} = \{ \pi : \pi^t W \leq (q^r)^t \}$  est indépendant de  $x$ .

Soit  $\{ \pi^t : t \in I \}$  l'ensemble des points extrêmes de  $\mathbf{P}$  et  $\{ \sigma^\delta : \delta \in \Delta \}$  l'ensemble des arêtes<sup>(1)</sup> extrêmes [77].

- (i) Si  $\mathbf{P} = \emptyset$  alors  $Q(x^0, \xi^r)$  est illimité ( $Q(x^0, \xi^r) = -\infty$ ) ou non réalisable.
- (ii) Si  $\mathbf{P} \neq \emptyset$  alors  $Q(x^0, \xi^r)$  est non réalisable ou admet une solution optimale.

Le lemme de *FARKAS* est une conséquence immédiate du théorème fort de dualité, ce qui apporte un état nécessaire et suffisant pour la faisabilité d'un système des contraintes linéaires et peut être énoncé comme la proposition suivante :

**Proposition 5.1** *L'ensemble  $\{z : Wz = h(\xi^r) - T(\xi^r) x^0, z \geq 0\} \neq \emptyset$  si et seulement si*

$$\sigma^t W \leq 0 \implies \sigma^t [h(\xi^r) - T(\xi^r) x^0] \leq 0.$$

---

<sup>(1)</sup>Une arête associée à deux point  $x$  et  $x'$  est l'ensemble des combinaisons convexes des points  $x$  et  $x'$ .

Nous concluons que  $Q(x^0, \xi^r)$  est non réalisable si et seulement si  $\mathbf{P}$  admet une arête extrême  $\sigma$  telle que :  $\sigma^t [h(\xi^r) - T(\xi^r)x^0] > 0$  ; autrement la valeur optimale de  $Q(x^0, \xi^r)$  est obtenu par  $\pi^t [h(\xi^r) - T(\xi^r)x^0]$ , où  $\pi$  est un point extrême de  $\mathbf{P}$ . Puis nous vérifions la faisabilité pour le problème de la deuxième étape, nous devons trouver une direction  $\sigma$  en résolvant le programme suivant :

$$\max_{\sigma} \{ \sigma^t [h(\xi^r) - T(\xi^r)x^0] : \sigma^t W \leq 0, \|\sigma\|_1 \leq 1 \} \quad (5.6)$$

On choisit la norme 1 pour rester dans le cas linéaire, tel que la dernière contrainte est ajoutée pour limiter (borner)  $\sigma$ ; sinon la valeur maximale sera  $+\infty$  et ceci est de notre intérêt.

Si pour un certain  $\xi^r$ ,  $r \in \{1, \dots, R\}$ ,  $\hat{\sigma}_r^t [h(\xi^r) - T(\xi^r)x^0] > 0$ , où  $\hat{\sigma}_r$  est la solution optimale de (5.6), nous avons trouvé  $\xi^r$  pour lequel  $x^0$  n'est pas une solution réalisable du problème de la deuxième étape. Dans ce cas-ci, nous ajoutons au (5.4) **la coupe de faisabilité**

$$\hat{\sigma}_r^t [h(\xi^r) - T(\xi^r)x] \leq 0 \quad (5.7)$$

au programme (5.4) et le nouveau programme est résolu en utilisant la méthode dual simplexe.

### 5.3.3 Efficacité (Optimalité)

Supposons que toutes les coupes de faisabilité déterminées sont rajoutées au problème (5.4), nous pouvons le reformulé en présentant une nouvelle variable  $\theta$  tel que nous résultons le problème suivant :

$$\left\{ \begin{array}{l} \text{“min” } \tilde{Z}^i = \hat{Z}^i + \theta, \quad i = 1, \dots, k \\ x \in \mathbb{S}, \\ \theta \geq Q(x), \\ x \text{ vecteur entier.} \end{array} \right. \quad (5.8)$$

avec  $\mathbb{S} = \{x \in \mathbb{R}^n : Ax = b, \sigma_r^t T(\xi^r)x \geq \sigma_r^t h(\xi^r), r \in \{1, \dots, R\}, x \geq 0\}$

$\mathbb{S}$  est un polyèdre non vide et compact dans  $\mathbb{R}^n$ .

Nous traitons un problème de recherche de l'ensemble  $Eff$  de toutes les solutions entières de (5.8) qui sont efficaces dans le sens de la définition suivante :

**Définition 5.6** Une solution  $x^0 \in \mathbb{S}$  est une solution efficace pour (5.8) s'il n'existe pas de solution  $x^1 \in \mathbb{S}$  tel que  $\tilde{Z}^i(x^1) \leq \tilde{Z}^i(x^0)$ ,  $i \in \{1, \dots, k\}$  et  $\tilde{Z}^i(x^1) < \tilde{Z}^i(x^0)$  pour au moins une valeur de  $i \in \{1, \dots, k\}$  et pour toute réalisation  $\xi^r$ ,  $r = 1, \dots, R$ .

En mettant en considération le programme mono-objectif (5.9), la première étape est de déterminer une solution réalisable entière de ce problème.

$$\left\{ \begin{array}{l} \text{“min” } \tilde{Z}^1 = \acute{Z}^1 + \theta \\ x \in \mathbb{S} \\ \theta \geq Q(x) \\ x \text{ entier.} \end{array} \right. \quad (5.9)$$

Algorithmiquement, nous ne pouvons pas utiliser  $\theta \geq Q(x)$  comme contrainte puisque  $Q(x)$  définit implicitement par un grand nombre de problème d'optimisation.

Nous résolvons le problème (5.9) sans  $\theta \geq Q(x)$  et obtenons une solution faisable  $(x^0, \theta^0)$ , (initialiser :  $\theta = -\infty$ ).

Les solutions optimales  $(\hat{\pi}_r, r = 1, \dots, R)$  de dual (5.5) sont utilisées pour calculer l'espérance de la fonction du recours  $Q(x^0)$  donnée par:

$$Q(x^0) = \sum_{r=1}^R \mathbb{P}^r Q(x^0, \xi^r) = \sum_{r=1}^R \mathbb{P}^r (\hat{\pi}_r)^t [h(\xi^r) - T(\xi^r)x^0].$$

Si  $\theta^0 \geq Q(x^0)$ ,  $x^0$  est optimale pour (5.9), sinon **la coupe d'optimalité**

$$\theta \geq \sum_{r=1}^R \mathbb{P}^r (\hat{\pi}_r)^t [h(\xi^r) - T(\xi^r)x] \quad (5.10)$$

est ajoutée au programme (5.9) et le nouveau programme est résolu en utilisant la méthode dual simplexe.

## 5.4 Conclusion

Dans la suite de ce travail, nous avons élaboré une nouvelle méthode de résolution du problème stochastique multi-objectifs en nombres entiers de la classe  $I/C/D$ . Nous avons proposé un algorithme efficace [8] combinant la technique de coupe d'efficacité développée par Abbas et Moulai [1] et le processus de séparation de la méthode branch & bound et la méthode L-shaped.

# Chapitre 6

## Une méthode de résolution du problème SMOILP

### 6.1 Introduction

Dans ce chapitre, nous décrivons une nouvelle méthode pour la résolution du problème de SMOILP (on considère travailler sur un espace d'incertitude décrit par un ensemble fini de scénarios) combinant la technique de coupe d'efficacité développée par Abbas et Moulaï dans [1] (son but initial était de résoudre les problèmes MOILP) et le processus de séparation de la méthode B&B [85] et la méthode *L-shaped* développée par Van Slyke et Wets dans [103]. La méthode L-Shaped est destinée à résoudre les problèmes représentés en tant que modèles de recours en deux étapes (on suppose que l'on dispose d'un ensemble fini de réalisations que l'on appellera également des *scénarios*).

Les seules contributions notables dans notre domaine d'intérêt sont, à notre connaissance, la méthode interactive: **STRANGE-MOMIX** pour le problème du SMOILP qui est développé par Teghem, 1990 [99], Abbas et Bellahcene [2] ont proposé la méthode *Cutting plane* pour SMOILP en 2006, il s'agit de l'adaptation au contexte mixte de la méthode *the cutting plane technique* pour MOILP développée par Abbas et Moulai dans [1] et la méthode *L-shaped* détaillée dans [60].

Le but de ce chapitre est de développer une nouvelle procédure pour la résolution

du problème de la programmation linéaire stochastique entière à objectifs multiples basée sur la méthode de deux étapes avec recours fixe et la méthode d'évaluation et séparation détaillés dans les chapitres précédents.

Nous terminons par une illustration de l'algorithme par un exemple numérique.

## 6.2 Optimisation linéaire stochastique multi-objectifs en nombres entiers

Nous proposons à présent une résolution exacte [6] du problème SMOILP (5.3), vu plus haut dans le chapitre 4, sous sa forme équivalente suivante :

$$(P) \begin{cases} \text{“min” } \tilde{Z}^i = \hat{Z}^i + \theta, & i = 1, \dots, k \\ x \in \mathbb{S}, \\ \theta \geq Q(x), \\ x \text{ vecteur entier.} \end{cases} \quad (6.1)$$

### 6.2.1 Une méthodologie pour résoudre SMOILP

Dans cette section, nous présentons une méthode exacte [8], basée sur le processus de branchement et utilisant une coupe efficace pour générer toutes les solutions efficaces entières pour le problème (P). Tout d'abord, la méthode proposée est présentée en détails, l'algorithme pour résoudre les problèmes de programmation linéaire multi-objectifs stochastiques en nombres entiers est ensuite détaillé.

Tout au long de ce qui suit, nous notons par :

$Eff$  : ensemble de solutions entières efficaces de (P),

$l$ ,  $N_l$  et  $M_l$  : entiers positifs où  $N_l$  et  $M_l$  indiquent le nombre de coupes de faisabilité et coupes d'optimalité, respectivement, ajoutées jusqu'à l'étape  $l$ .

### Description de la méthode

Commençons par  $\theta = -\infty$  sans coupe de faisabilité et d'optimalité [60].

La fonction objectif  $\mathbb{E}[c^1(\xi)x]$  (ou toute autre fonction objectif  $\mathbb{E}[c^i(\xi)x]$  ;  $i \in \{1, \dots, k\}$  à la place de  $\mathbb{E}[c^1(\xi)x]$ ) est minimisée sous les contraintes déterministes, pour l'obtention d'une solution optimale du problème principal relaxé, puis une solution entière est trouvée par l'application du processus de séparation retrouvé dans la méthode B&B.

Dès qu'une solution entière  $x$  est trouvée, nous testons si pour certaines réalisations  $\xi^r$  ( $r \in \{1, \dots, R\}$ ) les problèmes de la deuxième étape, générés par cette solution entière (5.6), ne sont pas réalisables. Ensuite, une coupe de faisabilité (5.7) est introduite et le problème principal est à réoptimiser pour obtenir une autre solution entière réalisable  $x'$ .

En utilisant  $x'$ , pour toutes les réalisations  $\xi^r$  ( $r \in \{1, \dots, R\}$ ), nous résolvons les programmes (5.6). De plus, nous calculons le recours  $Q(x')$ , afin de tester si  $\theta < Q(x')$ . Dans le cas affirmatif, une coupe d'optimalité (5.10) est introduite au problème courant, qui est encore à réoptimiser et le processus est à itérer. Sinon, la solution entière  $x'$  est comparée aux solutions déjà trouvées et donc l'ensemble de toutes les solutions potentiellement efficaces de  $Eff$  est mis à jour.

En présence d'une solution entière efficace, la coupe efficace (4.7) est construite en utilisant les directions de croissance de tous les critères de SMOILP, puis elle est ajoutée au problème principal pour éliminer non seulement des solutions non entières du domaine, mais aussi des solutions entières qui ne sont pas efficaces (4.7) assure une amélioration pour le critère  $i$  ce qui permet de pouvoir repartir à la recherche d'autres solutions non dominées). La recherche des solutions efficaces se fait dans chaque sous-domaine créé. Le processus s'arrête lorsque tous les domaines créés ont été explorés. Un domaine sera considéré comme un domaine exploré, s'il ne contient pas de solutions efficaces lorsqu'aucun critère ne puisse améliorer, auquel cas  $H_l$  est vide, ou bien le problème courant n'a pas de solution réalisable. La recherche de solutions efficaces ne s'arrête que si tous les domaines créés ont été explorés.

Pour ce faire, nous définissons un programme ( $P_l$ ) obtenu au nœud  $l$  correspondant à

la recherche d'une solution entière nouvelle, comme suit :

$$(P_l) \begin{cases} \min \mathbb{E} [c^1(\xi) x] + \theta \\ \text{s.c} \\ x \in S_l \\ \sigma_r^t [h(\xi^r) - T(\xi^r) x] \leq 0, \quad r = 1, \dots, N_l \\ \theta^l \geq Q(x), \quad l = 1, \dots, M_l \end{cases} \quad (6.2)$$

où  $N_l$  et  $M_l$  indiquent le nombre de coupes de faisabilité et coupes d'optimalité, respectivement, ajoutées jusqu'à l'étape  $l$ .

Un noeud  $l$  de l'arbre est sondé si  $(P_l)$  n'est pas réalisable ou bien  $H_l = \emptyset$ .

Si la solution optimale  $\tilde{x}^l$  du programme  $(P_l)$  n'est pas entière, dont la valeur  $x_j$  soit un composant de  $\tilde{x}^l$  de telle sorte que  $x_j = \alpha_j$ ; où  $\alpha_j$  est un nombre fractionnaire. Le noeud  $l$  de l'arborescence est ensuite séparé en deux noeuds qui sont imposés par les contraintes supplémentaires  $x_j \leq \lfloor \alpha_j \rfloor$  et  $x_j \geq \lfloor \alpha_j \rfloor + 1$ ; où  $\lfloor \alpha_j \rfloor$  est le plus grand entier inférieur à  $\alpha_j$ : Dans chaque noeud, le programme linéaire obtenu doit être résolu, jusqu'à ce qu'une solution entière soit trouvée. En présence d'une solution optimale entière, une coupe efficace  $\sum_{j \in H_l} x_j \geq 1$  est ensuite ajoutée pour supprimer les solutions en nombres entiers qui ne sont pas efficaces et le nouveau programme est résolu en utilisant la méthode du dual simplexe. La méthode se termine lorsque tous les noeuds créés sont sondés.

En conclusion, nous avons construit la solution  $(x, \theta)$  qui est entière réalisable et optimale pour (5.3) et vérifie le définition 5.6.

**Algorithme 6.1** (*Solutions efficaces pour SMOILP*)

**Étape 1. (Initialisation)**

On initialise le compteur d'itérations  $l = 0, \mathbb{F} = \{P_0\}, N_0 = M_0 = 0, \theta = -\infty$ .

Créer le premier noeud avec le programme linéaire déterministe  $(P_0)$  suivant :

$$(P_0) \begin{cases} \min \mathbb{E} [c^1(\xi) x] \\ \text{s.c} \\ x \in S_0 \end{cases}$$



Soit  $Eff = \emptyset$  l'ensemble de solutions entières efficaces, vide au départ.

Résoudre le programme linéaire  $(P_0)$  au nœud 0 de l'arborescence.

**Étape 2. (Étape générale)**

Tant que l'arborescence contient des nœuds non sondé, faire :

- Choisir le nœud  $l$  le plus récemment créé qui n'est pas encore sondé et résoudre le programme linéaire correspondant  $(P_l)$
- Si le programme  $(P_l)$  n'admet pas de solutions réalisables, alors le nœud  $l$  est sondé,
- Sinon, soit  $\tilde{x}_l$  la solution optimale trouvée,
- Si  $\tilde{x}_l$  n'est pas entière, aller à l'Étape 3. Sinon, aller à l'Étape 4.

**Étape 3. (Procédure de branchement)**

Choisir une coordonnée  $x_j$  de  $\tilde{x}_l$  dont la valeur  $x_j = \alpha_j$  où  $\alpha_j$  est un nombre fractionnaire et séparez le nœud actif  $l$  de l'arbre en deux nœuds  $h$  et  $h'$ ,  $h \geq l+1$ ;  $h' \geq l+1$  et  $h \neq h'$ .

Dans le tableau du simplexe courant, la contrainte supplémentaire  $x_j \leq \lfloor \alpha_j \rfloor$  est ajoutée et un nouveau domaine est considéré dans le nœud  $h$  et de même, la contrainte  $x_j \geq \lfloor \alpha_j \rfloor + 1$  est ajoutée pour obtenir un autre domaine considéré dans le nœud  $h'$ , et obtenir les sous problèmes  $(P_h)$ ,  $(P_{h'})$ . Faire  $\mathbb{F} = \mathbb{F} \cup \{P_h, P_{h'}\} \setminus \{P_l\}$ . Aller à l'Étape 2

**Étape 4. (Procédure de faisabilité de la seconde étape)**

Résoudre les sous-problèmes (5.6).

Si  $\sigma_r^t [h(\xi^r) - T(\xi^r) \tilde{x}^l] > 0$ , rajouter la coupe de faisabilité (5.7) au problème  $(P_l)$ , aller à l'Étape 2

Sinon, aller à l'Étape 5.

**Étape 5. (Procédure d'optimalité de la seconde étape)**

Résoudre les problèmes (5.5)

Si  $Q(x^l) > \theta$ ; Rajouter la coupe d'optimalité (5.10) au problème  $(P_l)$  aller à l'Étape 2

Sinon, aller à l'Étape 6.

**Étape 6. (Coupe d'efficacité)**

Si  $\tilde{Z}(\tilde{x}_l)$  n'est pas dominé par  $\tilde{Z}(x)$  pour tous  $x \in Eff$ , alors  $Eff := Eff \cup \{\tilde{x}_l\}$ .

S'il existe  $x \in Eff$  tel que  $\tilde{Z}(\tilde{x}_l)$  domine  $\tilde{Z}(x)$ , alors  $Eff := Eff \cup \{\tilde{x}_l\} \setminus \{x\}$ .

Déterminer les ensembles  $N_l$  et  $H_l$  :

Si  $H_l = \emptyset$ , alors le noeud  $l$  est sondé, aller à l'Étape 2.

Sinon, rajouter la coupe (4.7) :  $\sum_{j \in H_l} x_j \geq 1$  pour obtenir un nouveau programme linéaire  $(P_{l+1})$ , aller à l'Étape 2.

**Théorème 6.1** *Supposons que  $H_l \neq \emptyset$  au point entier courant  $x_l^*$ . Si  $x$  est une solution entière efficace dans le domaine  $S_l \setminus \{x_l^*\}$ , alors  $x \in S_{l+1}$ .*

**Preuve.** Soit  $x$  une solution entière se trouvant dans le domaine  $S_l \setminus \{x_l^*\}$  telle que  $x \notin S_{l+1}$ , alors  $\sum_{j \in H_l} x_j = 0$ , ce qui implique que  $x_j = 0$  pour tout  $j \in H_l$ . A partir du tableau du simplexe correspondant à la solution optimale  $x_l^*$ , les critères sont évalués comme suit:

$$Z^i(x) = \sum_{j \in N_l} \bar{c}_j^i x_j + \bar{c}^i, \forall i \in \{1, \dots, k\}, \text{ où } \bar{c}^i = Z^i(x_l^*).$$

$$\text{Ainsi, on peut écrire : } Z^i(x) = \sum_{j \in N_l \setminus H_l} \bar{c}_j^i x_j + \bar{c}^i, \forall i \in \{1, \dots, k\}$$

D'autre part  $\bar{c}_j^i \geq 0$  pour tout  $j \in N_l \setminus H_l$  et  $\bar{c}_j^i > 0$  pour au moins un critère, implique

$$\text{que } \sum_{j \in N_l \setminus H_l} \bar{c}_j^i x_j \geq 0 \Rightarrow \sum_{j \in N_l \setminus H_l} \bar{c}_j^i x_j + \bar{c}^i \geq \bar{c}^i.$$

Pour tout critère  $Z^i, i \in \{1, \dots, k\}$ , l'inégalité suivante est obtenue :

$$Z^i(x) = \sum_{j \in N_l} \bar{c}_j^i x_j + \bar{c}^i \Rightarrow Z^i(x) \geq \bar{c}^i \Rightarrow Z^i(x) \geq Z^i(x_l^*)$$

Par conséquent,  $Z^i(x) \geq Z^i(x_l^*)$  pour tout  $i \in \{1, \dots, k\}$  et  $Z^i(x) > Z^i(x_l^*)$  pour au moins un indice  $i$ . Donc  $Z(x_l^*)$  domine  $Z(x)$  et la solution  $x$  n'est pas efficace. ■

**Corollaire 6.1** *Supposons que  $H_l \neq \emptyset$  à la solution entière courante  $x_l^*$ , alors la contrainte  $\sum_{j \in H_l} x_j \geq 1$  définit une coupe efficace pour le programme  $(P_l)$ .*

**Preuve.** D'après le théorème précédent, aucune solution efficace n'est supprimée lorsque la contrainte  $\sum_{j \in H_l} x_j \geq 1$  est rajoutée. On peut alors dire que c'est une contrainte valide efficace. De plus,  $x_l^*$  ne vérifie pas cette contrainte puisque  $x_j = 0$ , pour tout  $j \in H_l$ . On conclut que la contrainte  $\sum_{j \in H_l} x_j \geq 1$  est une coupe efficace ■

**Proposition 6.1** *Si  $H_l = \emptyset$  au point entier courant  $x_l^*$ , alors  $S_l \setminus \{x_l^*\}$  est un domaine exploré.*

**Preuve.**  $H_l = \emptyset$  signifie que  $x_l^*$  est une solution entière optimale pour tous les critères ( $x_l^*$  constitue un point idéal localement dans le domaine  $S_l$ ) et donc  $S_l \setminus \{x_l^*\}$  ne contient pas de solutions efficaces. ■

**Théorème 6.2** *L'algorithme de recherche de toutes les solutions efficaces entières du problème (P) décrit ci-dessus converge en un nombre fini d'étapes.*

**Preuve.** L'ensemble des solutions possibles du problème (P), étant compact, contient un nombre fini de solutions entières. Chaque fois que l'on calcule un nombre entier  $x_l^*$  optimal, la coupe efficace est ajoutée. Ainsi, selon le théorème et corollaire ci-dessus, au moins la solution  $x_l^*$  est éliminée lorsqu'on étudie un quelconque sous-problème ( $P_k$ );  $k > l$ , mais aucune solution efficace n'est supprimée. ■

## 6.2.2 Organigramme de la méthode SMOILP

Pour résoudre un problème linéaire stochastique en nombres entiers à objectifs multiples, une nouvelle méthode basée sur trois principes :

- (i) Principe à deux étapes avec recours entier dont on ajoute à chaque itération une coupe de faisabilité ou coupe d'optimalité;
- (ii) Principe de séparation et évaluation en utilisant le concept d'ensembles dominés;
- (iii) Principe de coupe efficace pour générer toutes les solutions entières efficaces du problème (6.1).

Désignons par :

$Eff$  : ensemble des solutions entières et efficaces de (6.1) ;

$ND = \{ \tilde{Z}(x^i) / x^i \in Eff \}$  ;

$\mathbb{F}$  : une liste contenant les sous problèmes non sondés ;

$l, N_l, M_l$  : entiers positifs.

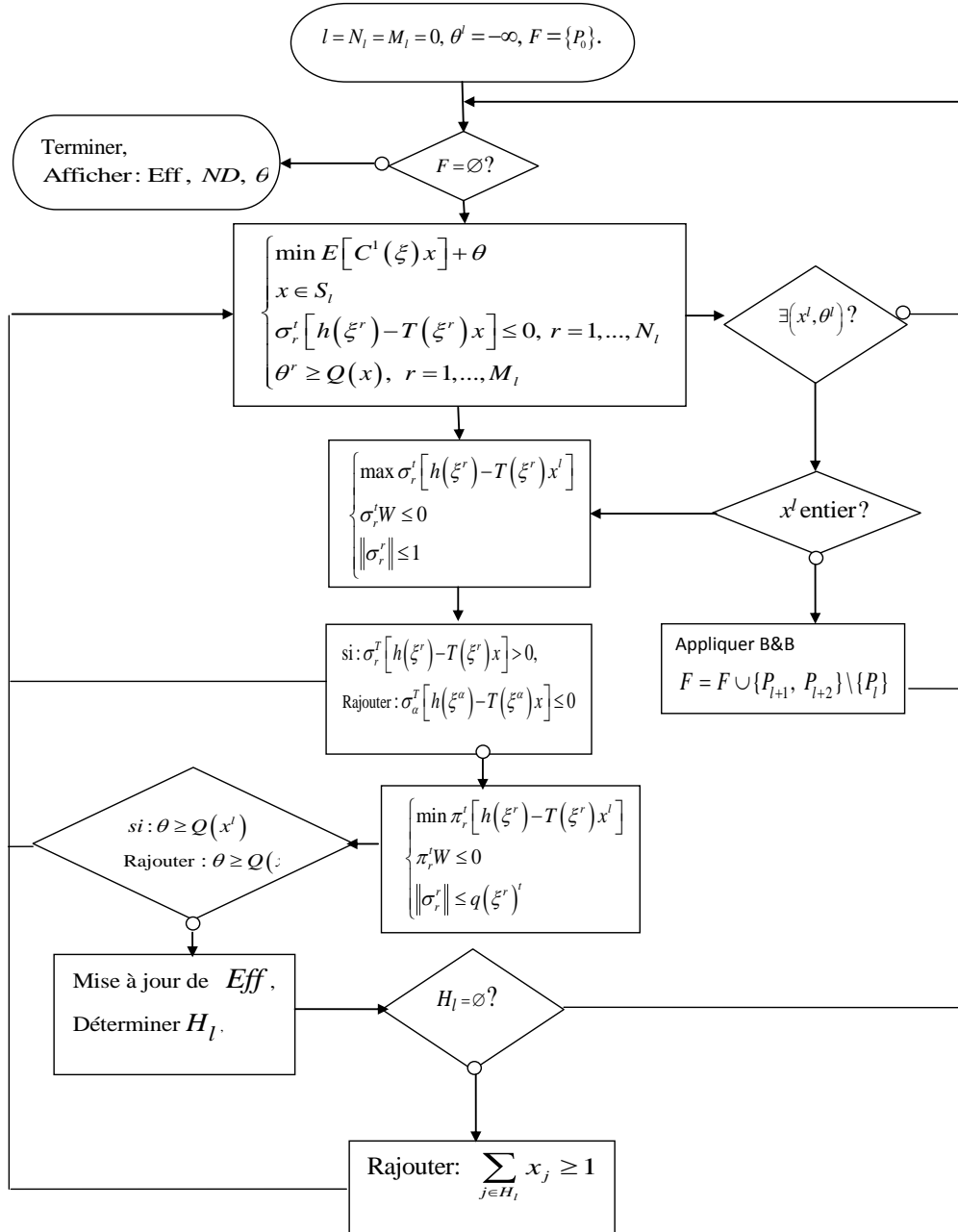


Figure 6.1 : Organigramme SMOILP

**Procédure d'initialisation** Considérer le problème ILP suivant :

$$(P_l) \begin{cases} \min \mathbb{E}[c^1(\xi)x] + Q(x) \\ x \in S_l \\ \sigma_r^t [h(\xi^r) - T(\xi^r)x] \leq 0, \quad r = 1, \dots, N_l \\ \theta^e \geq Q(x), \quad e = 1, \dots, M_l \\ x \geq 0. \end{cases}$$

Initialiser :  $\theta = -\infty$ ,  $l = N_l = M_l = 0$ ,  $\mathbb{F} = \{P_0\}$ ,  $Eff = \emptyset$  et  $ND = \emptyset$ .

Aller à la procédure de choix ;

**Procédure de choix** Si  $\mathbb{F} = \emptyset$ , afficher  $Eff$ ,  $ND$ , terminer.

Sinon, sélectionner le problème  $(P_l)$  de plus fort indice  $l$  dans la liste  $\mathbb{F}$ , aller à la procédure d'évaluation ;

**Procédure générale** Résoudre le programme linéaire  $(P_l)$  au nœud  $l$  de l'arborescence et poser  $x$  la solution optimale trouvée.

Si  $x$  n'est pas une solution entière, aller à la procédure de séparation. Sinon, aller à la procédure de faisabilité

S'il n'existe pas une telle solution  $x$ , faire  $\mathbb{F} = \mathbb{F} \setminus \{(P_l)\}$  et aller à la procédure de choix.

**Procédure de séparation** Choisir une coordonnée  $j$  de  $x$  dont la valeur  $\alpha_j$  n'est pas entière et séparer le nœud  $l$  en deux nouveaux nœuds : ajouter la contrainte  $x_j \leq \lfloor \alpha_j \rfloor$  au premier nœud, la contrainte  $x_j \geq \lfloor \alpha_j \rfloor + 1$  au second nœud et aller à la procédure de choix.

**Procédure de faisabilité** Résoudre les problèmes (5.6) :

$$\max_{\sigma} \{ \sigma^t [h(\xi^r) - T(\xi^r)x^l] : \sigma^t W \leq 0, \|\sigma\|_1 \leq 1 \}, r \in \{1, \dots, R\}$$

Si  $\sigma_r^t [h(\xi^r) - T(\xi^r)x^l] > 0$ , rajouter au problème  $(P_l)$  la coupe de faisabilité (5.7) :

$$\sigma_r^t [h(\xi^r) - T(\xi^r)x] \leq 0$$

Si  $\sigma_r^t [h(\xi^r) - T(\xi^r)x^l] \leq 0$ , aller à la procédure d'optimalité ;

**Procédure d'optimalité** Résoudre le problème (5.5) :

$$\max_{\pi} \{ \pi^t [h(\xi^r) - T(\xi^r)x^l] : \pi^t W \leq (q^r)^t \}, r \in \{1, \dots, R\}$$

Si  $Q(x^l) > \theta$  ; Rajouter au problème  $(P_l)$  la coupe d'optimalité (5.10)

$$\theta \geq \sum_{r=1}^R \mathbb{P}^r (\pi_r)^t [h(\xi^r) - T(\xi^r)x]. \text{ Aller à la procédure générale.}$$

Sinon, aller à la procédure d'efficacité;

**Procédure d'efficacité** S'il existe  $x \in Eff$  tel que  $\tilde{Z}(x^l)$  domine  $\tilde{Z}(x)$ , alors  $Eff := Eff \cup \{x\} \setminus \{x^l\}$ , aller à la procédure de choix.

Sinon ( $\tilde{Z}(x^l)$  n'est pas dominé par  $\tilde{Z}(x)$  pour tous  $x \in Eff$ ), alors  $Eff := Eff \cup \{x^l\}$ .

Déterminer les ensembles  $N_l$  et  $H_l$ :

Si  $H_l = \emptyset$ , alors le noeud  $l$  est sondé, aller à la Procédure de choix.

Sinon, rajouter la coupe (4.7)  $\sum_{j \in H_l} x_j \geq 1$  au programme  $(P_l)$ . Aller à la procédure générale.

## 6.3 Exemple illustratif

Comme illustration de la méthode proposée, nous présentons un exemple ([2] et [8]) de SMOILP avec une structure semblable à celle du problème  $(P)$ ,  $k = 3$ ,  $n_0 = 4$ ,  $m_0 = m = n = 2$ .

Deux contraintes déterministes:

$$\begin{cases} -4x_1 + 2x_2 \geq -8 \\ x_1 + x_2 \leq 5 \end{cases} \Leftrightarrow \begin{cases} 4x_1 - 2x_2 \leq 8 \\ x_1 + x_2 \leq 5 \end{cases}$$

Les contraintes déterministes sont données par :

$$A = \begin{pmatrix} 4 & -2 \\ 1 & 1 \end{pmatrix}; b = \begin{pmatrix} 8 \\ 5 \end{pmatrix}$$

La matrice de recours fixe est donnée par :

$$W(\xi) = W = \begin{pmatrix} -2 & -1 & 2 & 1 \\ 3 & 2 & -5 & -6 \end{pmatrix};$$

Deux scénarios ( $R = 2$ ) affectent les trois objectifs et les contraintes stochastiques.

Supposons que les deux réalisations de  $\xi$  sont équiprobables  $\mathbb{P}(\xi^1) = \frac{1}{2}$ ,  $\mathbb{P}(\xi^2) = \frac{1}{2}$

$$q(\xi^1) = (1, 0, 6, 2)^t, \quad q(\xi^2) = (5, 3, 2, 1)^t.$$

Les contraintes stochastiques sont données par :

$$T(\xi^1) = \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix}, \quad T(\xi^2) = \begin{pmatrix} 1 & 0 \\ 3 & 4 \end{pmatrix}, \quad h(\xi^1) = \begin{pmatrix} 3 \\ 5 \end{pmatrix}, \quad h(\xi^2) = \begin{pmatrix} 6 \\ 1 \end{pmatrix};$$

Le problème devient :

$$\left\{ \begin{array}{l} \min -9x_1 + 4x_2 \\ \min 3x_1 - 5x_2 \\ \min 8x_1 - 11x_2 \\ \text{s.c} \\ 4x_1 - 2x_2 \leq 8 \\ x_1 + x_2 \leq 5 \\ x_1 + 2x_2 = 3 \\ -2x_1 + x_2 = 5 \\ x_1, x_2 \in \mathbb{N} \end{array} \right.$$

Scénario 1

$$\left\{ \begin{array}{l} \min 3x_1 - 2x_2 \\ \min 7x_1 + x_2 \\ \min -4x_1 + 9x_2 \\ \text{s.c} \\ 4x_1 - 2x_2 \leq 8 \\ x_1 + x_2 \leq 5 \\ x_1 = 6 \\ 3x_1 + 4x_2 = 1 \\ x_1, x_2 \in \mathbb{N} \end{array} \right.$$

Nous calculons l'espérance de chaque fonction objectif :

$$(Z')^1 = \mathbb{E}[c^1(\xi)] = \mathbb{P}(\xi^1) c^1(\xi^1) + \mathbb{P}(\xi^2) c^1(\xi^2) = (-3, 1),$$

$$(Z')^2 = \mathbb{E}[c^2(\xi)] = \mathbb{P}(\xi^1) c^2(\xi^1) + \mathbb{P}(\xi^2) c^2(\xi^2) = (5, -2),$$

$$(Z')^3 = \mathbb{E}[c^3(\xi)] = \mathbb{P}(\xi^1) c^3(\xi^1) + \mathbb{P}(\xi^2) c^3(\xi^2) = (2, -1).$$

Formulons le problème déterministe associé :

$$\left\{ \begin{array}{l} \min -3x_1 + x_2 \\ \min 5x_1 - 2x_2 \\ \min 2x_1 - x_2 \\ \text{s.c} \\ 4x_1 - 2x_2 \leq 8 \\ x_1 + x_2 \leq 5 \\ x_1, x_2 \in \mathbb{N} \end{array} \right.$$

Créer le premier nœud avec le programme relaxé avec un seul critère  $(P_0)$ ,  $F = \{P_0\}$

$$(P_0) \left\{ \begin{array}{l} \min \mathbb{E}[c^1(\xi)x] \\ 4x_1 - 2x_2 + x_3 = 8 \\ x_1 + x_2 + x_4 = 5 \\ x_1, x_2, x_3, x_4 \geq 0 \end{array} \right.$$

$Eff = \emptyset$  : Ensemble des solutions entières efficaces du programme  $(P)$ .

- Première itération

La résolution du problème  $(P_0)$  donne le tableau du simplexe suivant:

Tableau 1

$B$	$Rhs$	$x_3$	$x_4$
$x_1$	3	$\frac{1}{6}$	$\frac{2}{6}$
$x_2$	2	$-\frac{1}{6}$	$\frac{4}{6}$
$Z'_1$	7	$\frac{4}{6}$	$\frac{2}{6}$
$Z'_2$	-11	$-\frac{7}{6}$	$-\frac{2}{6}$
$Z'_3$	-4	$-\frac{3}{6}$	0

La solution obtenue  $x = (3, 2)$  est une solution entière optimale de la première étape pour  $(P_0)$ .

$Rhs$  : La valeur de vecteur  $b$  après les opérations de changement de base.

- Pour tester la faisabilité du problème de deuxième étape, nous résolvons les sous problème (5.6) (correspondant aux scénarios  $\xi^1$  et  $\xi^2$ ) auquel on a passé la solution  $x = (3, 2)$ :



$$h(\xi^1) - T(\xi^1)x = \begin{pmatrix} 3 \\ 5 \end{pmatrix} - \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} -4 \\ 9 \end{pmatrix}$$

$$h(\xi^2) - T(\xi^2)x = \begin{pmatrix} 6 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 3 \\ -16 \end{pmatrix}$$

$$\left\{ \begin{array}{l} \max -4\sigma_1^1 + 9\sigma_1^2 \\ -2\sigma_1^1 + 3\sigma_1^2 \leq 0 \\ -\sigma_1^1 + 2\sigma_1^2 \leq 0 \\ 2\sigma_1^1 - 5\sigma_1^2 \leq 0 \\ \sigma_1^1 - 6\sigma_1^2 \leq 0 \\ \sigma_1^1 + \sigma_1^2 \leq 1 \end{array} \right. , \text{ le maximum est atteint en : } \sigma_1^t = (\sigma_1^1, \sigma_1^2) = \left(\frac{2}{3}, \frac{1}{3}\right)$$

$$\left\{ \begin{array}{l} \max 3\sigma_2^1 - 16\sigma_2^2 \\ -2\sigma_2^1 + 3\sigma_2^2 \leq 0 \\ -\sigma_2^1 + 2\sigma_2^2 \leq 0 \\ 2\sigma_2^1 - 5\sigma_2^2 \leq 0 \\ \sigma_2^1 - 6\sigma_2^2 \leq 0 \\ \sigma_2^1 + \sigma_2^2 \leq 1 \end{array} \right. , \text{ le maximum est atteint en : } \sigma_2^t = (\sigma_2^1, \sigma_2^2) = (0, 0)$$

$$\sigma_1^t [h(\xi^1) - T(\xi^1)x] = \left(\frac{2}{3}, \frac{1}{3}\right) \begin{pmatrix} -4 \\ 9 \end{pmatrix} = \frac{1}{3}$$

$$\sigma_2^t [h(\xi^2) - T(\xi^2)x] = (0, 0) \begin{pmatrix} 3 \\ -16 \end{pmatrix} = 0$$

$\sigma_1^t [h(\xi^1) - T(\xi^1)x] > 0$ , ceci signifie que le problème de deuxième étape n'est pas réalisable pour  $\xi^1$ , alors nous créons une coupe de faisabilité de la forme (5.7):  $\frac{5}{3}x_2 \geq \frac{11}{3}$  ou  $\frac{-5}{18}x_3 + \frac{10}{9}x_4 + x_5 = \frac{-1}{3}$ . Nous rajoutons cette coupe au problème de première étape.

La résolution du problème principal avec la nouvelle coupe rajoutée donne le tableau du simplexe suivant:

Tableau 2

$B$	$Rhs$	$x_4$	$x_5$
$x_1$	$\frac{14}{5}$	1	$\frac{1}{5}$
$x_2$	$\frac{11}{5}$	0	$-\frac{1}{5}$
$x_3$	$\frac{6}{5}$	-4	$-\frac{6}{5}$
$Z'_1$	$\frac{31}{5}$	3	$\frac{4}{5}$
$Z'_2$	$-\frac{48}{5}$	-5	$-\frac{7}{5}$
$Z'_3$	$-\frac{17}{5}$	-2	$-\frac{3}{5}$

, le minimum est atteint en  $x = \left(\frac{14}{5}, \frac{11}{5}\right)$ .

$x = \left(\frac{14}{5}, \frac{11}{5}\right)$  est optimale pour la première étape de programme, mais elle n'est pas entière. La séparation se fait donc par rapport à la variable  $x_1$  et on obtient les deux sous problèmes : Le processus de séparation commence:

**Nœud 1:**  $S'_1 = \{x \in S_0 : x_1 \geq \lfloor \frac{14}{5} \rfloor + 1\}$  et  $x_1 \geq 3 \iff x_4 + \frac{3}{5}x_5 + x_6 = -\frac{1}{5}$

Nous ajoutons cette contrainte et nous résolvons le nouveau programme ( $P_1$ ) mais il étant non réalisable, alors le nœud correspondant est sondé.

**Nœud 2:**  $S_1 = \{x \in S_0 : x_1 \leq \lfloor \frac{14}{5} \rfloor\}$  et  $x_1 \leq 2 \iff -x_4 - \frac{3}{5}x_5 + x_6 = -\frac{4}{5}$

Cette contrainte est rajoutée et la méthode dual simplexe est appliquée. La solution entière optimale du programme ( $P_2$ ) est obtenue dans le tableau suivant :

Tableau 3

$B$	$Rhs$	$x_5$	$x_6$
$x_1$	2	0	1
$x_2$	$\frac{11}{5}$	$-\frac{1}{5}$	0
$x_3$	$\frac{22}{5}$	$-\frac{2}{5}$	-4
$x_4$	$\frac{4}{5}$	$\frac{1}{5}$	-1
$Z'_1$	$\frac{19}{5}$	$\frac{1}{5}$	3
$Z'_2$	$-\frac{28}{5}$	$-\frac{2}{5}$	-5
$Z'_3$	$-\frac{9}{5}$	$-\frac{1}{5}$	-2

, le minimum est atteint en  $x = (2, \frac{11}{5})$  mais il n'est pas entier. La séparation se fait donc par rapport à la variable  $x_2$  et on obtient les deux sous problèmes:

**Nœud 3:**  $S'_2 = \{x \in S_1 : x_2 \leq \lfloor \frac{11}{5} \rfloor\}$  et  $x_2 \leq 2 \Rightarrow x_2 + x_7 = 2 \Rightarrow \frac{3}{5} x_5 + x_7 = -\frac{1}{5}$

Nous ajoutons cette contrainte et nous résolvons le nouveau programme ( $P_3$ ) mais il est impossible. Alors, ce nœud est sondé.

**Nœud 4:**  $S_2 = \{x \in S_1 : x_2 \geq \lfloor \frac{11}{5} \rfloor + 1\}$  et  $x_2 \geq 3 \Rightarrow -x_2 + x_7 = -3 \Rightarrow \frac{-3}{5} x_5 + x_7 = -\frac{4}{5}$

Cette contrainte est rajoutée et la méthode dual simplexe est appliquée. La solution entière optimale du programme ( $P_4$ ) est obtenue dans le tableau suivant :

Tableau 4

$B$	$Rhs$	$x_6$	$x_7$
$x_1$	2	1	0
$x_2$	3	0	-1
$x_3$	6	-4	-2
$x_4$	0	-1	1
$x_5$	4	0	-5
$Z'_1$	3	3	1
$Z'_2$	-4	-5	-2
$Z'_3$	-1	-2	-1

, le minimum est atteint en  $x = (2, 3)$ .

- Pour tester la faisabilité du problème de deuxième étape, nous résolvons les sous problème (5.6) (correspondant aux scénarios  $\xi^1$  et  $\xi^2$ ) auquel on a passé la solution  $x = (2, 3)$ :

$$h(\xi^1) - T(\xi^1)x = \begin{pmatrix} 3 \\ 5 \end{pmatrix} - \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \end{pmatrix} = \begin{pmatrix} -5 \\ 6 \end{pmatrix}$$

$$h(\xi^2) - T(\xi^2)x = \begin{pmatrix} 6 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 4 \\ -17 \end{pmatrix}$$

$$\left\{ \begin{array}{l} \max -5\sigma_1^1 + 6\sigma_1^2 \\ -2\sigma_1^1 + 3\sigma_1^2 \leq 0 \\ -\sigma_1^1 + 2\sigma_1^2 \leq 0 \\ 2\sigma_1^1 - 5\sigma_1^2 \leq 0 \\ \sigma_1^1 - 6\sigma_1^2 \leq 0 \\ \sigma_1^1 + \sigma_1^2 \leq 1 \end{array} \right. , \text{ le maximum est atteint en: } \sigma_1^t = (\sigma_1^1, \sigma_1^2) = (0, 0)$$

$$\left\{ \begin{array}{l} \max 4\sigma_2^1 - 17\sigma_2^2 \\ -2\sigma_2^1 + 3\sigma_2^2 \leq 0 \\ -\sigma_2^1 + 2\sigma_2^2 \leq 0 \\ 2\sigma_2^1 - 5\sigma_2^2 \leq 0 \\ \sigma_2^1 - 6\sigma_2^2 \leq 0 \\ \sigma_2^1 + \sigma_2^2 \leq 1 \end{array} \right. , \text{ le maximum est atteint en: } \sigma_2^t = (\sigma_2^1, \sigma_2^2) = (0, 0)$$

$\sigma_1 = \sigma_2 = 0$ , ceci veut dire que la nouvelle solution  $x = (2, 3)$  obtenue dans le tableau 6 est la solution réalisable de problème de deuxième étape.

- Pour tester l'optimalité de  $x = (2, 3)$  pour le problème de deuxième étape, nous résolvons les sous problème (5.5) (correspondant aux scénarios  $\xi^1$  et  $\xi^2$ ) auquel on a passé la solution  $x = (2, 3)$ :

$$\left\{ \begin{array}{l} \max -5\pi_1^1 + 6\pi_1^2 \\ -2\pi_1^1 + 3\pi_1^2 \leq 1 \\ -\pi_1^1 + 2\pi_1^2 \leq 0 \\ 2\pi_1^1 - 5\pi_1^2 \leq 6 \\ \pi_1^1 - 6\pi_1^2 \leq 2 \end{array} \right. , \text{ le maximum est atteint en: } \pi_1^t = (\pi_1^1, \pi_1^2) = \left(-1, -\frac{1}{2}\right)$$

$$\left\{ \begin{array}{l} \max 4\pi_2^1 - 17\pi_2^2 \\ -2\pi_2^1 + 3\pi_2^2 \leq 5 \\ -\pi_2^1 + 2\pi_2^2 \leq 3 \\ 2\pi_2^1 - 5\pi_2^2 \leq 2 \\ \pi_2^1 - 6\pi_2^2 \leq 1 \end{array} \right. , \text{ le maximum est atteint en: } \pi_2^t = (\pi_2^1, \pi_2^2) = (1, 0)$$

$$Q(x, \xi^1) = \pi_1^t [h(\xi^1) - T(\xi^1)x] = 2$$

$$Q(x, \xi^2) = \pi_1^t [h(\xi^2) - T(\xi^2)x] = 4$$

$$Q(x) = \frac{1}{2}Q(x, \xi^1) + \frac{1}{2}Q(x, \xi^2) = 3$$

$\theta = -\infty < Q(x)$ , nous introduisons la coupe d'optimalité de la forme (5.10):

$\theta \geq \frac{1}{4} - \frac{1}{2}x_1 + \frac{5}{4}x_2$  ou  $\frac{1}{2}x_6 + \frac{5}{4}x_7 - \theta + s_1 = -3$  et nous réoptimisons le problème précédent.

Tableau 5

$B$	$Rhs$	$x_6$	$x_7$	$s_1$
$x_1$	2	1	0	0
$x_2$	3	0	-1	0
$x_3$	6	-4	-2	0
$x_4$	0	-1	1	0
$x_5$	4	0	-5	0
$\theta$	3	$-\frac{1}{2}$	$-\frac{5}{4}$	-1
$Z'_1$	3	3	1	0
$Z'_2$	-4	-5	-2	0
$Z'_3$	-1	-2	-1	0

alors  $x^1 = (2, 3)$  est une solution entière efficace du programme  $(P)$  dont le vecteur critère est  $(\mathbb{E}[C^1(\xi)]x^1, \mathbb{E}[C^2(\xi)]x^1, \mathbb{E}[C^3(\xi)]x^1) = (-3, 4, 1)$  avec la pénalité  $\theta^1 = 3$ . Ainsi l'ensemble des solutions efficaces  $Eff = \{(2, 3)\}$

- $N_3 = \{6, 7\}$ ,  $H_3 = \{6, 7\} \neq \emptyset$

$$S_4 = \left\{ x \in S_3 \mid \sum_{j \in H_3} x_j \geq 1 \right\} = \{x \in S_3 \mid x_6 + x_7 \geq 1\} \text{ et } x_6 + x_7 \geq 1 \Leftrightarrow -x_6 - x_7 + x_8 = -1.$$

Nous ajoutons cette coupe efficace de la forme (4.7)  $-x_6 - x_7 + x_8 = -1$  et nous résolvons le nouveau programme de nœud correspondant, on obtient :

Tableau 6

$B$	$Rhs$	$x_4$	$x_8$	$s_1$
$x_1$	$\frac{3}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0
$x_2$	$\frac{7}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$	0
$x_3$	9	-1	-3	0
$x_6$	$\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$	0
$x_5$	$\frac{13}{2}$	$\frac{5}{2}$	$-\frac{5}{6}$	0
$\theta$	$\frac{31}{8}$	$\frac{3}{8}$	$-\frac{7}{8}$	-1
$x_7$	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$	0
$Z'_1$	1	1	2	0
$Z'_2$	$-\frac{1}{2}$	$-\frac{3}{2}$	$-\frac{7}{2}$	0
$Z'_3$	$\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{3}{2}$	0

le minimum est atteint en  $x = \left(\frac{3}{2}, \frac{7}{2}\right)$  mais ce n'est pas entier. La séparation se fait donc par rapport à la variable  $x_1$  et on obtient les deux sous problèmes :

**Nœud 5:**  $S'_4 = \{x \in S_3 : x_1 \geq \lfloor \frac{3}{2} \rfloor + 1\}$  et  $x_1 \geq 2 \Leftrightarrow \frac{1}{2}x_4 + \frac{1}{2}x_8 + x_9 = \frac{-1}{2}$

Nous ajoutons cette contrainte et nous résolvons le nouveau programme ( $P_5$ ) mais il est impossible. Alors, ce nœud est sondé.

**Nœud 6:**  $S_4 = \{x \in S_3 : x_1 \leq \lfloor \frac{3}{2} \rfloor\}$  et  $x_1 \leq 1 \Leftrightarrow -\frac{1}{2}x_4 - \frac{1}{2}x_8 + x_9 = \frac{-1}{2}$

Cette contrainte est rajoutée et la méthode dual simplexe est appliquée. La solution entière optimale du programme ( $P_6$ ) est obtenue dans le tableau suivant :

Tableau 7

$B$	$Rhs$	$x_8$	$x_9$	$s_1$
$x_1$	1	0	1	0
$x_2$	3	-1	1	0
$x_3$	10	-2	-2	0
$x_6$	1	0	-1	0
$x_5$	4	-5	5	0
$\theta$	$\frac{14}{4}$	$-\frac{5}{4}$	$\frac{3}{4}$	-1
$x_7$	0	-1	1	0
$x_4$	1	1	-2	0
$Z'_1$	0	1	2	0
$Z'_2$	1	-2	-3	0
$Z'_3$	1	-1	-1	0

le minimum est atteint en  $x = (1, 3)$

- Pour tester la faisabilité du problème de deuxième étape, nous résolvons chaque sous-problème (5.6) (correspondant aux scénarios  $\xi^1$  et  $\xi^2$ ) auquel on a passé la solution  $x = (1, 3)$  :

$$h(\xi^1) - T(\xi^1)x = \begin{pmatrix} 3 \\ 5 \end{pmatrix} - \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} -4 \\ 4 \end{pmatrix}$$

$$h(\xi^2) - T(\xi^2)x = \begin{pmatrix} 6 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 5 \\ -14 \end{pmatrix}$$

$$\left\{ \begin{array}{l} \max -4\sigma_1^1 + 4\sigma_1^2 \\ -2\sigma_1^1 + 3\sigma_1^2 \leq 0 \\ -\sigma_1^1 + 2\sigma_1^2 \leq 0 \\ 2\sigma_1^1 - 5\sigma_1^2 \leq 0 \\ \sigma_1^1 - 6\sigma_1^2 \leq 0 \\ \sigma_1^1 + \sigma_1^2 \leq 1 \end{array} \right. , \text{ le maximum est atteint en: } \sigma_1^t = (\sigma_1^1, \sigma_1^2) = (0, 0)$$



$$\left\{ \begin{array}{l} \max 5\sigma_2^1 - 14\sigma_2^2 \\ -2\sigma_2^1 + 3\sigma_2^2 \leq 0 \\ -\sigma_2^1 + 2\sigma_2^2 \leq 0 \\ 2\sigma_2^1 - 5\sigma_2^2 \leq 0 \\ \sigma_2^1 - 6\sigma_2^2 \leq 0 \\ \sigma_2^1 + \sigma_2^2 \leq 1 \end{array} \right. , \text{ le maximum est atteint en: } \sigma_2^t = (\sigma_2^1, \sigma_2^2) = (0, 0)$$

$\sigma_1 = \sigma_2 = 0$ , ceci veut dire que la nouvelle solution  $x = (1, 3)$  obtenue dans le tableau 7 est réalisable pour le problème de deuxième étape.

- Pour tester l'optimalité de  $x = (1, 3)$  pour le problème de deuxième étape, nous résolvons les sous problème (5.5) (correspondant aux scénarios  $\xi^1$  et  $\xi^2$ ) auquel on a passé la solution  $x = (1, 3)$ :

$$\left\{ \begin{array}{l} \max -5\pi_1^1 + 6\pi_1^2 \\ -2\pi_1^1 + 3\pi_1^2 \leq 1 \\ -\pi_1^1 + 2\pi_1^2 \leq 0 \\ 2\pi_1^1 - 5\pi_1^2 \leq 6 \\ \pi_1^1 - 6\pi_1^2 \leq 2 \end{array} \right. , \text{ le maximum est atteint en: } \pi_1^t = (\pi_1^1, \pi_1^2) = (-1, -\frac{1}{2})$$

$$\left\{ \begin{array}{l} \max 4\pi_2^1 - 17\pi_2^2 \\ -2\pi_2^1 + 3\pi_2^2 \leq 5 \\ -\pi_2^1 + 2\pi_2^2 \leq 3 \\ 2\pi_2^1 - 5\pi_2^2 \leq 2 \\ \pi_2^1 - 6\pi_2^2 \leq 1 \end{array} \right. , \text{ le maximum est atteint en: } \pi_2^t = (\pi_2^1, \pi_2^2) = (1, 0)$$

$$Q(x, \xi^1) = \pi_1^t [h(\xi^1) - T(\xi^1)x] = 2$$

$$Q(x, \xi^2) = \pi_2^t [h(\xi^2) - T(\xi^2)x] = 5$$

$$Q(x) = \frac{1}{2}Q(x, \xi^1) + \frac{1}{2}Q(x, \xi^2) = \frac{7}{2}$$

$\theta = Q(x)$ , alors  $x^2 = (1, 3)$  est une solution entière efficace du programme ( $P$ ) avec la pénalité  $\theta^2 = \frac{7}{2}$ .

L'ensemble des solutions efficaces  $Eff = \{(2, 3), (1, 3)\}$  puisque  $\tilde{Z}(1, 3)$  n'est pas dominé par  $\tilde{Z}(2, 3)$ .

- $N_6 = \{8, 9\}, H_6 = \{8, 9\}$

$$\bullet S_7 = \left\{ x \in S_6 : \sum_{j \in H_6} x_j \geq 1 \right\} = \{x \in S_6 : x_8 + x_9 \geq 1\} \text{ et } x_8 + x_9 \geq 1 \Leftrightarrow -x_8 - x_9 + x_{10} = -1$$

Nous ajoutons cette coupe efficace (4.7)  $-x_8 - x_9 + x_{10} = -1$  et nous résolvons le nouveau programme de nœud correspondant.

Tableau 8

$B$	$Rhs$	$x_9$	$x_{10}$	$s_1$
$x_1$	1	1	0	0
$x_2$	4	2	-1	0
$x_3$	12	0	-2	0
$x_6$	1	1	0	0
$x_5$	9	10	-5	0
$\theta$	$\frac{19}{4}$	2	$-\frac{5}{4}$	-1
$x_7$	1	2	-1	0
$x_4$	0	-3	1	0
$x_8$	1	1	-1	0
$Z'_1$	-1	1	1	0
$Z'_2$	3	-1	-2	0
$Z'_3$	2	0	-1	0

, le minimum est atteint en  $x = (1, 4)$

- Pour tester la faisabilité du problème de deuxième étape, nous résolvons chaque sous problème (5.6) (correspondant aux scénarios  $\xi^1$  et  $\xi^2$ ) auquel on a passé la solution  $x = (1, 4)$  :

$$h(\xi^1) - T(\xi^1)x = \begin{pmatrix} 3 \\ 5 \end{pmatrix} - \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 4 \end{pmatrix} = \begin{pmatrix} -6 \\ 3 \end{pmatrix}$$

$$h(\xi^2) - T(\xi^2)x = \begin{pmatrix} 6 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 1 \\ 4 \end{pmatrix} = \begin{pmatrix} 5 \\ -18 \end{pmatrix}$$

$$\left\{ \begin{array}{l} \max -6\sigma_1^1 + 3\sigma_1^2 \\ -2\sigma_1^1 + 3\sigma_1^2 \leq 0 \\ -\sigma_1^1 + 2\sigma_1^2 \leq 0 \\ 2\sigma_1^1 - 5\sigma_1^2 \leq 0 \\ \sigma_1^1 - 6\sigma_1^2 \leq 0 \\ \sigma_1^1 + \sigma_1^2 \leq 1 \end{array} \right. , \text{ le maximum est atteint en: } \sigma_1^t = (\sigma_1^1, \sigma_1^2) = (0, 0)$$

$$\left\{ \begin{array}{l} \max 5\sigma_2^1 - 18\sigma_2^2 \\ -2\sigma_2^1 + 3\sigma_2^2 \leq 0 \\ -\sigma_2^1 + 2\sigma_2^2 \leq 0 \\ 2\sigma_2^1 - 5\sigma_2^2 \leq 0 \\ \sigma_2^1 - 6\sigma_2^2 \leq 0 \\ \sigma_2^1 + \sigma_2^2 \leq 1 \end{array} \right. , \text{ le maximum est atteint en: } \sigma_2^t = (\sigma_2^1, \sigma_2^2) = (0, 0)$$

$\sigma_1 = \sigma_2 = 0$ , ceci veut dire que la nouvelle solution  $x = (1, 4)$  obtenue dans le tableau 8 est réalisable pour le problème de deuxième étape.

- Pour tester l'optimalité de  $x = (1, 4)$  pour le problème de deuxième étape, nous résolvons les sous problème (5.5) (correspondant aux scénarios  $\xi^1$  et  $\xi^2$ ) auquel on a passé la solution  $x = (1, 4)$  :

$$\left\{ \begin{array}{l} \max -6\pi_1^1 + 3\pi_1^2 \\ -2\pi_1^1 + 3\pi_1^2 \leq 1 \\ -\pi_1^1 + 2\pi_1^2 \leq 0 \\ 2\pi_1^1 - 5\pi_1^2 \leq 6 \\ \pi_1^1 - 6\pi_1^2 \leq 2 \end{array} \right. , \text{ le maximum est atteint en : } \pi_1^t = (\pi_1^1, \pi_1^2) = \left(-1, -\frac{1}{2}\right)$$

$$\left\{ \begin{array}{l} \max 5\pi_2^1 - 18\pi_2^2 \\ -2\pi_2^1 + 3\pi_2^2 \leq 5 \\ -\pi_2^1 + 2\pi_2^2 \leq 3 \\ 2\pi_2^1 - 5\pi_2^2 \leq 2 \\ \pi_2^1 - 6\pi_2^2 \leq 1 \end{array} \right. , \text{ le maximum est atteint en : } \pi_2^t = (\pi_2^1, \pi_2^2) = (1, 0)$$

$$Q(x, \xi^1) = \pi_1^t [h(\xi^1) - T(\xi^1)x] = \frac{9}{2}$$

$$Q(x, \xi^2) = \pi_1^t [h(\xi^2) - T(\xi^2)x] = 5$$

$$Q(x) = \frac{1}{2}Q(x, \xi^1) + \frac{1}{2}Q(x, \xi^2) = \frac{19}{4}$$

$\theta = Q(x)$ , alors  $x^3 = (1, 4)$  est une solution entière efficace du programme  $(P)$  avec  $\theta^3 = \frac{19}{4}$ .

L'ensemble efficace  $Eff = \{(2, 3), (1, 3), (1, 4)\}$  puisque  $\tilde{Z}(1, 4)$  n'est pas dominé par  $\{\tilde{Z}(2, 3), \tilde{Z}(1, 3)\}$ .

- $N_7 = \{9, 10\}, H_7 = \{9, 10\} \neq \emptyset$

$$S_8 = \left\{ x \in S_7 : \sum_{j \in H_7} x_j \geq 1 \right\} = \{x \in S_7 : x_9 + x_{10} \geq 1\} \text{ et } x_9 + x_{10} \geq 1 \Leftrightarrow -x_9 - x_{10} + x_{11} = -1$$

Nous ajoutons cette coupe efficace (4.7)  $-x_9 - x_{10} + x_{11} = -1$  et nous résolvons le nouveau programme de nœud correspondant.

Tableau 9

$B$	$Rhs$	$x_7$	$x_{11}$	$s_1$
$x_1$	$\frac{1}{3}$	$-\frac{1}{3}$	$\frac{1}{3}$	0
$x_2$	3	-1	0	0
$x_3$	$\frac{38}{3}$	$-\frac{2}{3}$	$-\frac{4}{3}$	0
$x_6$	$\frac{1}{3}$	$-\frac{1}{3}$	$\frac{1}{3}$	0
$x_5$	4	-5	0	0
$\theta$	$\frac{23}{6}$	$-\frac{13}{12}$	$-\frac{1}{6}$	-1
$x_{10}$	$\frac{1}{3}$	$-\frac{1}{3}$	$-\frac{2}{3}$	0
$x_4$	$\frac{5}{3}$	$\frac{4}{3}$	$-\frac{1}{3}$	0
$x_8$	$\frac{2}{3}$	$-\frac{2}{3}$	$-\frac{1}{3}$	0
$x_9$	$\frac{2}{3}$	$\frac{1}{3}$	$-\frac{1}{3}$	0
$Z'_1$	-2	0	1	0
$Z'_2$	$\frac{13}{3}$	$-\frac{1}{3}$	$-\frac{5}{3}$	0
$Z'_3$	$\frac{7}{3}$	$-\frac{1}{3}$	$-\frac{2}{3}$	0

le minimum est atteint en  $x = (\frac{1}{3}, 3)$  mais n'est pas entier. La séparation se fait donc par rapport à la variable  $x_1$  et on obtient les deux sous problèmes:

**Nœud 7:**  $S'_8 = \{x \in S_7 : x_1 \geq \lfloor \frac{1}{3} \rfloor + 1\}$  et  $x_1 \geq 1 \Leftrightarrow -\frac{1}{3}x_7 + \frac{1}{3}x_{11} + x_{12} = -\frac{2}{3}$

Nous ajoutons cette contrainte et nous résolvons le nouveau programme ( $P_7$ ) mais il est impossible. Alors, ce nœud est sondé.

**Nœud 8:**  $S_8 = \{x \in S_7 : x_1 \leq \lfloor \frac{1}{3} \rfloor\}$  et  $x_1 \leq 0 \Leftrightarrow \frac{1}{3}x_7 - \frac{1}{3}x_{11} + x_{12} = -\frac{1}{3}$

Cette contrainte est rajoutée et la méthode dual simplexe est appliquée. La solution entière optimale du programme ( $P_8$ ) est obtenue dans le tableau suivant:

Tableau 10

$B$	$Rhs$	$x_7$	$x_{12}$	$s_1$
$x_1$	0	0	1	0
$x_2$	3	-1	0	0
$x_3$	14	-2	-4	0
$x_6$	0	0	1	0
$x_5$	4	-5	0	0
$\theta$	4	$-\frac{5}{4}$	$-\frac{1}{2}$	-1
$x_{10}$	1	-1	-2	0
$x_4$	2	1	-1	0
$x_8$	1	-1	-1	0
$x_9$	1	0	-1	0
$x_{11}$	1	-1	-3	0
$Z'_1$	-3	1	3	0
$Z'_2$	6	-2	-5	0
$Z'_3$	3	-1	-2	0

, le minimum est atteint en  $x = (0, 3)$

- Pour tester la faisabilité du problème de deuxième étape, nous résolvons chaque sous

problème (5.6) (correspondant aux scénarios  $\xi^1$  et  $\xi^2$ ) auquel on a passé la solution  $x = (0, 3)$  :

$$h(\xi^1) - T(\xi^1)x = \begin{pmatrix} 3 \\ 5 \end{pmatrix} - \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 3 \end{pmatrix} = \begin{pmatrix} -3 \\ 2 \end{pmatrix}$$

$$h(\xi^2) - T(\xi^2)x = \begin{pmatrix} 6 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 0 \\ 3 \end{pmatrix} = \begin{pmatrix} 6 \\ -11 \end{pmatrix}$$

$$\left\{ \begin{array}{l} \max -3\sigma_1^1 + 2\sigma_1^2 \\ -2\sigma_1^1 + 3\sigma_1^2 \leq 0 \\ -\sigma_1^1 + 2\sigma_1^2 \leq 0 \\ 2\sigma_1^1 - 5\sigma_1^2 \leq 0 \\ \sigma_1^1 - 6\sigma_1^2 \leq 0 \\ \sigma_1^1 + \sigma_1^2 \leq 1 \end{array} \right. , \text{ le maximum est atteint en : } \sigma_1^t = (\sigma_1^1, \sigma_1^2) = (0, 0)$$

$$\left\{ \begin{array}{l} \max 6\sigma_2^1 - 11\sigma_2^2 \\ -2\sigma_2^1 + 3\sigma_2^2 \leq 0 \\ -\sigma_2^1 + 2\sigma_2^2 \leq 0 \\ 2\sigma_2^1 - 5\sigma_2^2 \leq 0 \\ \sigma_2^1 - 6\sigma_2^2 \leq 0 \\ \sigma_2^1 + \sigma_2^2 \leq 1 \end{array} \right. , \text{ le maximum est atteint en : } \sigma_2^t = (\sigma_2^1, \sigma_2^2) = \left(\frac{5}{7}, \frac{2}{7}\right)$$

$$\sigma_2^t [h(\xi^2) - T(\xi^2)x] = \frac{8}{7}$$

$\sigma_1^t [h(\xi^2) - T(\xi^2)x] > 0$ , ceci veut dire que le problème de deuxième étape (5.6) est pas réalisable pour  $\xi^2$ . Alors nous créons la coupe de faisabilité de la forme (5.7):  $\frac{11}{7} x_1 + \frac{8}{7} x_2 \geq \frac{32}{7}$  et nous réoptimisons le programme précédent. ceci implique que la solution  $x = (0, 4)$  obtenue dans le tableau 11 est réalisable pour le problème de deuxième étape.

Tableau 11

$B$	$Rhs$	$x_{12}$	$x_{13}$	$s_1$
$x_1$	0	1	0	0
$x_2$	4	$-\frac{11}{8}$	$-\frac{1}{8}$	0
$x_3$	16	$-\frac{54}{8}$	$-\frac{2}{8}$	0
$x_6$	0	1	0	0
$x_5$	9	$-\frac{55}{8}$	$-\frac{5}{8}$	0
$\theta$	$\frac{21}{4}$	$-\frac{71}{32}$	$-\frac{5}{32}$	-1
$x_{10}$	2	$-\frac{27}{8}$	$-\frac{1}{8}$	0
$x_4$	1	$\frac{3}{8}$	$\frac{1}{8}$	0
$x_8$	2	$-\frac{19}{8}$	$-\frac{1}{8}$	0
$x_9$	1	-1	0	0
$x_{11}$	2	$-\frac{35}{8}$	$-\frac{1}{8}$	0
$x_7$	1	$-\frac{11}{8}$	$-\frac{1}{8}$	0
$Z'_1$	-4	$\frac{35}{8}$	$\frac{1}{8}$	0
$Z'_2$	8	$-\frac{62}{8}$	$-\frac{2}{8}$	0
$Z'_3$	4	$-\frac{27}{8}$	$-\frac{1}{8}$	0

, le minimum est atteint en  $x = (0, 4)$

- Pour tester la faisabilité du problème de deuxième étape, nous résolvons chaque sous problème (5.6) (correspondant aux scénarios  $\xi^1$  et  $\xi^2$ ) auquel on a passé la solution  $x = (0, 4)$  :

$$h(\xi^1) - T(\xi^1)x = \begin{pmatrix} 3 \\ 5 \end{pmatrix} - \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 4 \end{pmatrix} = \begin{pmatrix} -5 \\ 1 \end{pmatrix}$$

$$h(\xi^2) - T(\xi^2)x = \begin{pmatrix} 6 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 0 \\ 4 \end{pmatrix} = \begin{pmatrix} 6 \\ -15 \end{pmatrix}$$

$$\left\{ \begin{array}{l} \max -5\sigma_1^1 + \sigma_1^2 \\ -2\sigma_1^1 + 3\sigma_1^2 \leq 0 \\ -\sigma_1^1 + 2\sigma_1^2 \leq 0 \\ 2\sigma_1^1 - 5\sigma_1^2 \leq 0 \\ \sigma_1^1 - 6\sigma_1^2 \leq 0 \\ \sigma_1^1 + \sigma_1^2 \leq 1 \end{array} \right. , \text{ le maximum est atteint en: } \sigma_1^t = (\sigma_1^1, \sigma_1^2) = (0, 0)$$

$$\left\{ \begin{array}{l} \max 6\sigma_2^1 - 15\sigma_2^2 \\ -2\sigma_2^1 + 3\sigma_2^2 \leq 0 \\ -\sigma_2^1 + 2\sigma_2^2 \leq 0 \\ 2\sigma_2^1 - 5\sigma_2^2 \leq 0 \\ \sigma_2^1 - 6\sigma_2^2 \leq 0 \\ \sigma_2^1 + \sigma_2^2 \leq 1 \end{array} \right. , \text{ le maximum est atteint en: } \sigma_2^t = (\sigma_2^1, \sigma_2^2) = (0, 0)$$

$\sigma_1 = \sigma_2 = 0$ , ceci implique que la solution  $x = (0, 4)$  obtenue dans le tableau 11 est réalisable pour le problème de deuxième étape.

- Pour tester l'optimalité de  $x = (0, 4)$  pour le problème de deuxième étape, nous résolvons les sous problème (5.5) (correspondant aux scénarios  $\xi^1$  et  $\xi^2$ ) auquel on a passé la solution  $x = (0, 4)$ :

$$\left\{ \begin{array}{l} \max -5\pi_1^1 + \pi_1^2 \\ -2\pi_1^1 + 3\pi_1^2 \leq 1 \\ -\pi_1^1 + 2\pi_1^2 \leq 0 \\ 2\pi_1^1 - 5\pi_1^2 \leq 6 \\ \pi_1^1 - 6\pi_1^2 \leq 2 \end{array} \right. , \text{ le maximum est atteint en : } \pi_1^t = (\pi_1^1, \pi_1^2) = \left(-1, -\frac{1}{2}\right)$$

$$\left\{ \begin{array}{l} \max 6\pi_2^1 - 15\pi_2^2 \\ -2\pi_2^1 + 3\pi_2^2 \leq 5 \\ -\pi_2^1 + 2\pi_2^2 \leq 3 \\ 2\pi_2^1 - 5\pi_2^2 \leq 2 \\ \pi_2^1 - 6\pi_2^2 \leq 1 \end{array} \right. , \text{ le maximum est atteint en : } \pi_2^t = (\pi_2^1, \pi_2^2) = (1, 0)$$

$$Q(x, \xi^1) = \pi_1^t [h(\xi^1) - T(\xi^1)x] = \frac{9}{2}$$

$$Q(x, \xi^2) = \pi_1^t [h(\xi^2) - T(\xi^2)x] = 6$$



$$Q(x) = \frac{1}{2}Q(x, \xi^1) + \frac{1}{2}Q(x, \xi^2) = \frac{21}{4}$$

$\theta = Q(x)$ , alors,  $x^4 = (0, 4)$  est une solution entière efficace du programme  $(P)$  avec la pénalité  $\theta^4 = \frac{21}{4}$ .

L'ensemble efficace  $Eff = \{(2, 3), (1, 3), (1, 4), (0, 4)\}$  puisque  $\tilde{Z}(0, 4)$  n'est pas dominé par  $\{\tilde{Z}(2, 3), \tilde{Z}(1, 3), \tilde{Z}(1, 4)\}$

- la coupe efficace est facilement construite et rajouter au tableau courant d'une solution entière.  $N_9 = \{12, 13\}$ ,  $H_9 = \{12, 13\}$

$$S_{10} = \left\{ x \in S_9 : \sum_{j \in H_9} x_j \geq 1 \right\} = \{x \in S_{10} : x_{12} + x_{13} \geq 1\} \text{ et } x_9 + x_{10} \geq 1 \Leftrightarrow -x_{12} - x_{13} + x_{14} = -1$$

Nous ajoutons cette coupe efficace (4.7)  $-x_{12} - x_{13} + x_{14} = -1$  et nous résolvons le nouveau programme de nœud correspondant.

Tableau 12

$B$	$Rhs$	$x_{12}$	$x_{14}$	$s_1$
$x_1$	0	1	0	0
$x_2$	$\frac{33}{8}$	$-\frac{10}{8}$	$-\frac{1}{8}$	0
$x_3$	$\frac{25}{4}$	$-\frac{52}{8}$	$-\frac{2}{8}$	0
$x_6$	0	1	0	0
$x_5$	4	$-\frac{50}{8}$	$-\frac{5}{8}$	0
$\theta$	$\frac{173}{32}$	$-\frac{66}{32}$	$-\frac{5}{32}$	-1
$x_{10}$	$\frac{17}{8}$	$-\frac{26}{8}$	$-\frac{1}{8}$	0
$x_4$	$\frac{7}{8}$	$\frac{2}{8}$	$\frac{1}{8}$	0
$x_8$	$\frac{17}{8}$	$-\frac{18}{8}$	$-\frac{1}{8}$	0
$x_9$	1	-1	0	0
$x_{11}$	$\frac{17}{8}$	$-\frac{34}{8}$	$-\frac{1}{8}$	0
$x_7$	$\frac{33}{8}$	$-\frac{10}{8}$	$-\frac{1}{8}$	0
$x_{13}$	1	1	-1	0
$Z'_1$	$-\frac{33}{8}$	$\frac{34}{8}$	$\frac{1}{8}$	0
$Z'_2$	$\frac{33}{4}$	$-\frac{60}{8}$	$-\frac{2}{8}$	0
$Z'_3$	$\frac{33}{8}$	$-\frac{26}{8}$	$-\frac{1}{8}$	0

La solution optimale  $x = (0, \frac{33}{8})$  du programme  $(P_{10})$  est obtenue dans le tableau 12, qui n'est pas entière. Le processus de séparation commence :

**Nœud 9:**  $S'_{11} = \{x \in S_{10} : x_2 \leq \lfloor \frac{39}{8} \rfloor\}$  et  $x_2 \leq 4 \Leftrightarrow \frac{1}{2}x_{12} + \frac{7}{8}x_{41} + x_{15} = \frac{-7}{8}$   
 Nous ajoutons cette contrainte et nous résolvons le nouveau programme  $(P_9)$  mais ceci impossible. Alors, ce nœud est sondé.

**Nœud 10:**  $S_{11} = \{x \in S_{10} : x_2 \geq \lfloor \frac{39}{8} \rfloor + 1\}$  et  $x_2 \geq 5 \Leftrightarrow \frac{-1}{2}x_{12} - \frac{7}{8}x_{14} + x_{15} = \frac{-1}{8}$   
 Cette contrainte est rajoutée et la méthode dual simplexe est appliquée. La solution entière optimale du programme  $(P_{11})$  est obtenue dans le tableau suivant :

Tableau 13

$B$	$Rhs$	$x_{12}$	$x_{15}$	$s_1$
$x_1$	0	1	0	0
$x_2$	5	0	-1	0
$x_3$	$\frac{39}{8}$	-4	-2	0
$x_6$	0	1	0	0
$x_5$	$\frac{67}{8}$	0	-5	0
$\theta$	$\frac{13}{2}$	$\frac{29}{8}$	$-\frac{5}{4}$	-1
$x_{10}$	3	-2	-1	0
$x_4$	0	-1	1	0
$x_8$	3	-1	-1	0
$x_9$	1	-1	0	0
$x_{11}$	3	-3	-1	0
$x_7$	5	0	-1	0
$x_{13}$	8	11	-8	0
$x_{14}$	7	10	-8	0
$Z'_1$	-5	3	1	0
$Z'_2$	10	5	-2	0
$Z'_3$	5	-2	-1	0

La solution entière optimale  $x = (0, 5)$  du programme  $(P_{10})$  est obtenue dans le tableau 13.

- Pour tester la faisabilité du problème de deuxième étape, nous résolvons chaque sous problème (5.6) (correspondant aux scénarios  $\xi^1$  et  $\xi^2$ ) auquel on a passé la solution  $x = (0, 5)$ :

$$h(\xi^1) - T(\xi^1)x = \begin{pmatrix} 3 \\ 5 \end{pmatrix} - \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 5 \end{pmatrix} = \begin{pmatrix} -7 \\ 0 \end{pmatrix}$$

$$h(\xi^2) - T(\xi^2)x = \begin{pmatrix} 6 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 0 \\ 5 \end{pmatrix} = \begin{pmatrix} 6 \\ -19 \end{pmatrix}$$

$$\left\{ \begin{array}{l} \max -7\sigma_1^1 \\ -2\sigma_1^1 + 3\sigma_1^2 \leq 0 \\ -\sigma_1^1 + 2\sigma_1^2 \leq 0 \\ 2\sigma_1^1 - 5\sigma_1^2 \leq 0 \\ \sigma_1^1 - 6\sigma_1^2 \leq 0 \\ \sigma_1^1 + \sigma_1^2 \leq 1 \end{array} \right. , \text{ le maximum est atteint en : } \sigma_1^t = (\sigma_1^1, \sigma_1^2) = (0, 0)$$

$$\left\{ \begin{array}{l} \max 6\sigma_2^1 - 19\sigma_2^2 \\ -2\sigma_2^1 + 3\sigma_2^2 \leq 0 \\ -\sigma_2^1 + 2\sigma_2^2 \leq 0 \\ 2\sigma_2^1 - 5\sigma_2^2 \leq 0 \\ \sigma_2^1 - 6\sigma_2^2 \leq 0 \\ \sigma_2^1 + \sigma_2^2 \leq 1 \end{array} \right. , \text{ le maximum est atteint en : } \sigma_2^t = (\sigma_2^1, \sigma_2^2) = (0, 0)$$

$\sigma_1 = \sigma_2 = 0$ , ceci implique que la solution  $x = (0, 5)$  obtenue dans le tableau 13 est réalisable pour le problème de deuxième étape.

- Pour tester l'optimalité de  $x = (0, 5)$  pour le problème de deuxième étape, nous résolvons les sous problème (5.5) (correspondant aux scénarios  $\xi^1$  et  $\xi^2$ ) auquel on a passé la solution  $x = (0, 5)$  :

$$\left\{ \begin{array}{l} \max -7\pi_1^1 \\ -2\pi_1^1 + 3\pi_1^2 \leq 1 \\ -\pi_1^1 + 2\pi_1^2 \leq 0 \\ 2\pi_1^1 - 5\pi_1^2 \leq 6 \\ \pi_1^1 - 6\pi_1^2 \leq 2 \end{array} \right. , \text{ le maximum est atteint en : } \pi_1^t = (\pi_1^1, \pi_1^2) = \left(-1, -\frac{1}{2}\right)$$

$$\left\{ \begin{array}{l} \max 6\pi_2^1 - 19\pi_2^2 \\ -2\pi_2^1 + 3\pi_2^2 \leq 5 \\ -\pi_2^1 + 2\pi_2^2 \leq 3 \\ 2\pi_2^1 - 5\pi_2^2 \leq 2 \\ \pi_2^1 - 6\pi_2^2 \leq 1 \end{array} \right. , \text{ le maximum est atteint en : } \pi_2^t = (\pi_2^1, \pi_2^2) = (1, 0)$$

$$Q(x, \xi^1) = \pi_1^t [h(\xi^1) - T(\xi^1)x] = 7$$

$$Q(x, \xi^2) = \pi_1^t [h(\xi^2) - T(\xi^2)x] = 6$$

$$Q(x) = \frac{1}{2}Q(x, \xi^1) + \frac{1}{2}Q(x, \xi^2) = \frac{13}{2}$$

$\theta = Q(x)$ , alors la solution courante  $x^5 = (0, 5)$  est une solution entière efficace du programme  $(P)$  avec la pénalité  $\theta^5 = \frac{13}{2}$ .

L'ensemble  $Eff = \{(2, 3), (1, 3), (1, 4), (0, 4), (0, 5)\}$  puisque  $\tilde{Z}(0, 5)$  n'est pas dominé par  $\{\tilde{Z}(2, 3), \tilde{Z}(1, 3), \tilde{Z}(1, 4), \tilde{Z}(0, 4)\}$ .

$$\bullet N_{10} = \{12, 15\}, H_{10} = \{12, 15\}$$

$$S_{12} = \left\{ x \in S_{11} : \sum_{j \in H_{10}} x_j \geq 1 \right\} = \{x \in S_{12} : x_{12} + x_{15} \geq 1\} \text{ et } x_{12} + x_{15} \geq 1 \Leftrightarrow -x_{12} - x_{15} + x_{16} = -1$$

Nous ajoutons cette coupe efficace (4.7)  $-x_{12} - x_{15} + x_{16} = -1$  et nous résolvons le nouveau programme de nœud correspondant.

Nous ajoutons cette coupe efficace et nous résolvons le nouveau programme mais le dual est non réalisable, alors le nœud correspondant est sondé.

L'algorithme SMOILP s'arrête puisque tous les nœuds sont sondés et l'ensemble de toutes les solutions efficaces et leur pénalités du programme  $(P)$  sont données par le tableau suivant:

Tableau 14

	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$
$x^i$	(2, 3)	(1, 3)	(1, 4)	(0, 4)	(0, 5)
$\theta^i$	3	$\frac{7}{2}$	$\frac{19}{4}$	$\frac{21}{4}$	$\frac{13}{2}$
$Z'_i$	(-3, 4, 1)	(0, -1, -1)	(-1, 3, 2)	(4, -8, -4)	(5, -10, -5)
$\tilde{Z}_i = Z'_i + \theta^i$	(0, 7, 4)	$\left(\frac{7}{2}, \frac{5}{2}, \frac{5}{2}\right)$	$\left(\frac{23}{4}, \frac{5}{4}, \frac{11}{4}\right)$	$\left(\frac{37}{4}, -\frac{11}{4}, \frac{5}{4}\right)$	$\left(\frac{23}{2}, -\frac{7}{2}, \frac{3}{2}\right)$

## 6.4 Expérimentation et résultats

L'algorithme décrit ci-dessus a été implémenté dans le programme MATLAB 2007, en utilisant un PC (Intel Core 2 duo 2.26 GHz processor), avec 3 Gb. avec RAM. L'algorithme est testé avec les problèmes SMOILP générés aléatoirement avec une distribution uniforme discrète. Le nombre  $k$  des fonctions objectifs a été pris comme 3, 5 et 10. Pour chaque dimension, dix instances sont résolus. La moyenne et le maximum du temps CPU (par second) et le nombre de solutions efficaces. Nous montrons les résultats des expériences de calcul dans le tableau suivant:

Tableau 15

R	K	m	n	Max Nbr of Sol	avg Nbr of Sol	Maximal Time	avg Time
2	3	2	3	5	4	1.3251	1.2486
2	3	5	10	7	3.2	2.3189	1.4785
2	3	10	20	4	2.4	3.8741	1.7273
2	3	25	50	8	5.1	43.2937	29.2338
3	5	5	10	4	3.1	2.1938	1.8412
3	5	10	20	7	5.7	8.7497	4.2785
3	5	20	50	5	2.3	231.8738	189.3721
3	10	20	50	12	7	56.1574	243.5691

**Max Nbr of Sol :** Le nombre maximal de solutions efficaces pour dix problèmes avec la même dimension.

**avg Nbr of Sol :** Le nombre moyen de solutions efficaces pour dix problèmes avec la même dimension.

**Maximal time :** Le temps maximal de dix instances de problèmes générées aléatoirement.

**Avg time :** Le temps moyen maximal de dix instances de problèmes générées aléatoirement.

# Conclusion

Dans la classe des problèmes déterministes, toute information est complètement définie. Par contre dans notre travail, l'information comporte des éléments indéterminés c'est-à-dire certains paramètres sont aléatoires mais définis par des caractéristiques probabilistes connues. Ce qui nous a conduit à l'étude du problème d'optimisation linéaire stochastique à objectifs multiples SMOLP. Pour ce faire, des données stochastiques sont traitées par un recours entier fixé au préalable pour obtenir un programme déterministe équivalent à deux étapes.

La méthode proposée peut identifier l'ensemble de toutes les solutions efficaces entières de ce problème équivalent si elles existent. Cet algorithme est approprié seulement pour des problèmes avec un nombre raisonnable (petit) de scénarios, mais il pourrait être appliqué pour un grand nombre d'objectifs. Basée sur la coupe de faisabilité, la coupe d'optimalité, l'exploration d'arborescence et la coupe efficace pour générer l'ensemble efficient discret, une nouvelle méthode de résolution est présentée dans ce document.

La méthode d'optimisation du problème stochastique linéaire en nombres entiers à objectifs multiples présentée dans cette thèse constitue un résultat important dans la littérature compte tenu des propriétés spécifiques du problème SMOILP, où l'essence du problème traite l'aspect aléatoire d'une certaine manière efficace.

Nous pensons qu'il serait toujours intéressant de voir l'impact de la méthode "*integer L-shaped*" à la place de L-shaped utilisée dans notre problème. Aussi, nous jugeons utile de voir le problème de recours avec plusieurs étapes qui correspond aux cas réels que nous pouvons rencontrer.



# Bibliographie

- [1] M. Abbas, M. Moulai (1999) Solving Multiple Objective Integer Linear Programming, *Ricerca Operativa* vol.29 n.89, 15–38.
- [2] M. Abbas, F. Bellahcene (2006) Cutting plane method for multiple objective stochastic integer linear programming. *European Journal of Operational Research* 168 967–984.
- [3] L. Amodeo, C. Prins et D. Sanchez (2009). Comparison of metaheuristic approaches for multi-objective simulation-based optimization in supply chain inventory management. In *Lecture notes in computer science. Applications of evolutionary computing*. Berlin: Springer.
- [4] S. Amrouche, M. Moulai (2006) Optimisation linéaire stochastique multi-objectifs en nombres entiers, Colloque sur l'Optimisation et les Systèmes d'Information (COSI'06), 11-13 Juin 2006, l'USTHB, Alger, Algérie. <http://www.isima.fr/cosi/cosi2006/>
- [5] S. Amrouche, M. Moulai (2008) Integer Linear Programming with Multiple Objectives under Uncertainty, *Proceedings of the International Symposium On Operational Research (ISOR)*. USTHB, Algérie.
- [6] S. Amrouche, M. Moulai (2009) Integer Linear Programming with Multiple objectives under uncertainty, *Recherche Opérationnelle et Aide à la Décision, ROAD*, n°49 novembre 2009.

- [7] S. Amrouche, M. Moulai (2010) Integer linear programming with multiple objective under uncertainty, Proceedings sur la cinquième conférence international en recherche opérationnelle (CIRO'10), Marrakech, Maroc.
- [8] S. Amrouche, M. Moulai (2012) Multi-objective stochastic integer linear programming with fixed recourse, International Journal of Multicriteria Decision Making (IJMCDM), Vol. 2, No. 4, 379–390.
- [9] S. Amrouche, M. Moulai (2012) Branch and Cut Method for Stochastic Integer Linear Problem with Multiple Objective, Proceedings du congrès des mathématiciens algériens CMA2012, Annaba, Algérie.
- [10] S. Amrouche, M. Moulai (2012) Solving two-stage stochastic integer linear programming with multiple objective, Proceedings du International Conference on Pure and Applied Mathematics ICPAM'12, Guelma, Algérie.
- [11] S. Amrouche, M. Moulai (2015) An exact method for solving Multi-Objective Stochastic Integer Linear Programming, In proceeding of the International Conference on Applied Mathematics and Computational Methods in Engineering (AM-CME 2015),(pp. 1405–1411). Barcelona, Spain.
- [12] S. Amrouche, M. Moulai (2017) Solving Biobjective Stochastic Integer Linear Optimization, In proceeding of the International Conference on Multiple Objective Programming and Goal Programming (MOPGP), Metz, France.
- [13] B. Armbruster et J. B. Luedtke (2011). Models and formulations for multivariate dominance constrained stochastic programs (Technical Report).
- [14] F. F. Baesler et J. A. Sepúlveda (2001). Multi-objective simulation optimization for a cancer treatment center. In Proc. winter simulation conference 2001 (pp. 36–38).
- [15] S. K. Bath, J. S. Dhillon et D. P. Kothari (2004). Stochastic multiobjective generation dispatch. Electric Power Components and Systems, 32, 1083–1103.

- [16] E.M.L. Beal (1955), On minimizing a convex function subject to linear inequalities, *J. Royal statistical society, series B17*. pp 173–184.
- [17] F. Ben Abdelaziz, P. Lang, et R. Nadeau (1994) Pointwise Efficiency in Multiobjective Stochastic Linear Programming, *Journal of the Operational Research Society*, Vol. 45, pp. 1324–1334.
- [18] F. Ben Abdelaziz, P. Lang, et R. Nadeau (1995) Distributional Efficiency in Multiobjective Linear Programming, *European Journal of Operational Research*, Vol. 85, pp. 399–415.
- [19] F. Ben Abdelaziz, P. Lang, R. Nadeau (1997) Distributional un animity multiobjective stochastic linear programming. In: Climaco, J. (Ed.), *Multicriteria Analysis: Proceedings of the XIth International Conference on MCDM*. Springer-Verlag, Berlin, pp. 225–236.
- [20] F. Ben Abdelaziz, P. Lang, R. Nadeau (1999) Dominance and efficiency in multicriteria decision under uncertainty. *Theory and Decision* 47, 191–211.
- [21] F. Ben Abdelaziz, B. Aouni et R. El Fayedh (2007). Multiobjective stochastic programming for portfolio selection. *European Journal of Operations Research*, 177, 1811–1823.
- [22] F. Ben Abdelaziz et H. Masri (2010) A compromise solution for the multiobjective stochastic linear programming under partial uncertainty. *European Journal of Operational Research* 202, 55–59.
- [23] F. Ben Abdelaziz (2012) Solution approaches for the multiobjective stochastic programming. *European Journal of Operational Research* 216, 1–16.
- [24] J. F. Benders (1962) Partitioning procedures for solving mixed-variables programming problems. *Numer., Math.*, 4 :238–252.
- [25] H. P. Benson (1978) Existence of efficient solutions for vector maximization problems, *Journal of Optimization Theory and Appl.*, 26, 569–580.

- [26] J. R. Birge, R. J.-B. Wets (1986) Designing approximation schemes for stochastic optimization problems, in particular for stochastic programs with recourse, *Math. Programming Stud.* 27, 54–102.
- [27] J. R. Birge et F. Louveaux (1997) *Introduction to stochastic programming*. Springer-Verlag, New York.
- [28] John R. Birge. *Stochastic programming computation and applications*. *INFORMS, J. Comput.*, 9(2) :111–133.
- [29] J. R. Birge et F. Louveaux (2011) *Introduction to Stochastic Programming*, Springer Series in Operations Research and Financial Engineering.
- [30] C. Caballero, E. Cerda, M.M. Muñoz, L. Rey (2000) Relations among several efficiency concepts in stochastic multiple objective programming. In: *Research and Practice in Multiple Criteria Decision Making* Haimes, Y.Y., Steuer, R. (Eds.), *Lectures Notes in Economics and Mathematical Systems*. Springer-Verlag, Berlin, Germany, pp. 57–68.
- [31] R. Caballero, E. Cerdá, M.M. Muñoz, L. Rey, I.M. Stancu-Minasian (2001) Efficient solution concepts and their relations in stochastic multiobjective programming. *Journal of Optimization, Theory and Applications* 110 (1), 53–74.
- [32] Caroe, Tind (1998) L-Shaped Decomposition of Two-Stage Stochastic Programs with Integer Recourse, *European Journal of Operations Research*, 101, 306–316.
- [33] D. Chaabane (2005) *Contribution à l’optimisation multicritère en variables discrètes*, thèse de doctorat d’Etat en mathématiques, option recherche opérationnelle, USTHB, Algérie.
- [34] A. Charnes, W.W. Cooper, et G.H. Symonds (1958). Cost horizons and certainty equivalents : An approach to stochastic programming of heating oil. *Management Science*, 4 :183–195.

- [35] A. Charnes et W. W. Cooper (1959) Chance-constrained programming, *Management Sci.* 5 , 73–79.
- [36] M. E-A. Chergui et M. Moulay (2008) An exact method for a discrete multiobjective linear fractional optimization. *Journal Applied Mathematical Decision VS*; 2008 (ID 760191) :12 pages.
- [37] M. E-A. Chergui (2010) Contribution à la programmation non linéaire multicritère, thèse de doctorat d'Etat en mathématiques, spécialité recherche opérationnelle, USTHB, Algérie.
- [38] K. L. Chung (1974) A course in probability theory, third edition, Academic press, A harcourt science and technology company.
- [39] V. Chvatal (1999) Linear programming, MC Gill University. p115 and p28–29.
- [40] J. Clímaco, C. Ferreira et M. Captivo (1997) Multicriteria integer programming : An overview of the different algorithmic approaches, *Multicriteria Analysis* 2, 248–258.
- [41] Y.Collette et P. Siarry (1992) optimisation multiobjectif, Eyrolles, ISBN 2-212-11168-1.
- [42] G. B. Dantzig, W.Orchard-Hays (1954) The product form of the inverse in the simplexe method. *Mathematical tables and other aids to computation*, p. 64–67
- [43] G. B. Dantzig (1955) Linear programming under uncertainty. *Management Science*, 1 :197–206.
- [44] G.B. Dantzig and A. Madansky (1961) On the solution of two-stage linear programs under uncertainty. In I.J. Neyman, editor, *Proc. 4th Berkeley Symp. Math. Stat. Prob*, pages 165-176, Berkeley.
- [45] G. B. Dantzig (1963) *Linear programming and extensions*. Princeton University Press. Princeton, New Jersey,

- [46] M. Ehrgott (2005) Multicriteria optimization, Second edition, lecture notes in economics and mathematical systems.
- [47] M. Ehrgott , J. Figueira et X. Gandibleux (2006) Multiple Objective Discrete and Combinatorial Optimization, Annals of Operations Research, Vol. 147.
- [48] Evans J, Steuer R (1973) A revised simplex method for multiple objective programs. Math Program 5(1):54–72
- [49] M. Fonseca, A. Garcia-Sánchez, M. Ortega-Mier et F. Saldanha da Gama (2010). A stochastic bi-objective location model for strategic reverse logistics. TOP, 18, 158–184.
- [50] A. M. Geoffrion (1968). Proper efficiency and the theory of vector maximization. Journal of Mathematical Analysis and Applications, 22:618–630.
- [51] P. Girardeau (2010) Résolution de grands problèmes en optimisation stochastique dynamique et synthèse de lois de commande, thèse de doctorat d’Etat en mathématiques, université Paris-Est.
- [52] R. E. Gomory (1958) Outline of an algorithm for integer solutions to linear programs, Bulletin of the AMS 64, 275–278.
- [53] R. E. Gomory (1960) An algorithm for the mixed-integer problem. Report RM-2597, Rand Corporation.
- [54] A. Goicoechea, L. Dukstein, R.L. Bulfin, Multiobjective stochastic programming the PROTRADE-method. Operation Research Society of America, 1976.
- [55] W. J. Gutjahr et P. Reiter (2010). Bi-objective project portfolio selection and staff assignment under uncertainty. Optimization Methods & Software, 59, 417–445.
- [56] H. A. Hassan-Pour, M. Mosadegh-Khah et R. Tavakkoli-Moghaddam (2009). Solving a multi-objective multi-depot stochastic location-routing problem by a hybrid simulated annealing algorithm. Proceedings of the Institution of Mechanical Engineers. Part B, Journal of Engineering Manufacture, 1045–1054.

- [57] C. Henggeler Antunes et al.(2016) Multiobjective Linear and Integer Programming, EURO Advanced Tutorials on Operational Research, DOI 10.1007/978-3-319-28746-1\_2
- [58] F. Hnaien, X. Delorme et A. Dolgui (2010). Multi-objective optimization for inventory control in two-level assembly systems under uncertainty of lead times. *Computers and Operations Research*, 20, 1835–1843.
- [59] Higle et Sen (1991) Stochastic Decomposition : An Algorithm for Two-Stage Linear Programs with Recourse, *Mathematics of Operations Research*, 16(3), 650–669.
- [60] P. Kall, S. W. Wallace (1994) *Stochastic Programming*. John Wileyet Sons, Chichester.
- [61] J. Mayer et P.Kall (2005) *Stochastic linear programming, Models, Theory, and Computation*, International series in operations research & Management science.
- [62] S. Kataoka (1963) A stochastic programming model. *Econometrica*, 31 :181–196.
- [63] J.E. Kelley (1960) The cutting plane method for solving convex programs. *SIAM J. Appl. Math.*, 8:703-712.
- [64] G. Kolomvos : Résolution de grands problèmes stochastiques multi-étapes : application à un problème de dimensionnement de capacités et de gestion de flux et de stocks. Thèse de doctorat, École Central Paris, 2007
- [65] T. C. Koopmans (1951). An analysis of production as an efficient combination of activities. In *Activity analysis of production and allocation*. New York: Wiley.
- [66] M. Köksalan, J. Wallenius et S. Zionts (2011). *Multiple criteria decision making: from early history to the 21st century*. Singapore: World Scientific.
- [67] H. Kuhn , A. Tucker (1951) Nonlinear programming. In: Neyman J (ed) *Proceedings of the 2nd Berkley symposium on mathematics statistics and probability*. University of California Press, Berkeley, CA, pp 481–492

- 
- [68] H. Land et A. G. Doig (1960) An automatic method for solving discrete programming problems. *Econometrica*, 28, 497–520.
- [69] G. Laporte, F V. Louveaux (1992) The integer L-shaped method for stochastic integer programs with complete recourse, *Operations Research Letters* 13. 133–142, North-Hollet.
- [70] G. Laporte and F.V Louveaux (1993) The integer l-shaped method for stochastic integer programs. *Operations Research Letters*, 13 : 48–62.
- [71] F.V. Louveaux (1986) Multistage stochastic programs with block-separable recourse, *Math. Programming Stud.* 28, 48–62.
- [72] A. Lucena, J .E. Beasley (1996) Banch and cut algorithms, ch.5 in *advances in linear and integer programming* , J.Beasley Ed, Oxford, university press.
- [73] Maarten H. van der Vlerk (1995) *Stochastic Programming with Integer Recourse*, PhD thesis, University of Groningen.
- [74] H. Markowitz (1952) Portofolio secection. *Journal of Finance*, 7 :77–91.
- [75] M. Minoux (2008) *Programmation mathématique, Théorie et algorithmes*, 2ème édition. Lavoisier.
- [76] M. Moulai (2002) *Optimisation multicritère fractionnaire linéaire en nombres entiers*, thèse de doctorat d’Etat en mathématiques, option recherche opérationnelle, USTHB, Algérie.
- [77] G. L. Nemhauser, L. A. Wolsey (1988) *Integer and combinatorial optimization*, Wiley Interscience Series in Discrete Mathematics and Optimisation.
- [78] A. Prekopa (1995) *Stochastic Programming*. Kluwer Academic Publishers, Dordrecht.
- [79] Rasmussen LM (1986) Zero–one programming with multiple criteria. *Eur J Oper Res* 26:83–95



- [80] D. Roman, K. Darby-Dowman et G. Mitra (2007). Mean-risk models using two risk measures: a multiobjective approach. *Quantitative Finance*, 7, 443–458.
- [81] A. Ruszczyński and A. Shapiro (2006) Optimization of convex risk functions. *Mathematics of Operations Research*, 31 :433–452.
- [82] Th. Saaty et S. Gass (1954). Parametric objective function (part i). *Operations Research*, 316–319.
- [83] M. Sakarovitch (1983). *Techniques mathématique de la recherche opérationnelle, IV Optimisation combinatoire*, ENSIMAG, Université Scientifique et Médicale, Institut National Polytechnique de Grenoble.
- [84] N. Sahinidis (2003) Optimization under uncertainty : state-of-the-art and opportunities. Technical report, Computers and Chemical Engineering.
- [85] M. Sakarovitch (1984). *Optimisation combinatoire, méthodes mathématiques algorithmiques, programmation discrète*, tomes 1 et 2, ed. Hermann.
- [86] P. Schutz, A. Tomasgard et S. Ahmed (2009) Supply chain design under uncertainty using sample average approximation and dual decomposition. *European Journal of Operational Research*, 199, 409–419.
- [87] A. Shapiro, D. Dentcheva, A. Ruszczyński (2009) *Lectures on stochastic programming, Modeling and theory*, Society for Industrial and Applied Mathematics and the Mathematical Programming Society MPS-SIAM series on optimization; 9.
- [88] Sherali et raticelli (2002). A Modification of Benders Decomposition Algorithm for Discrete Subproblems: An Approach for Stochastic Programs with Integer Recourse, *Journal of Global Optimization*, 22(1-4), 319–342.
- [89] R. Slowinski, J. Teghem (1988) Fuzzy versus Stochastic Approaches to multicriteria Linear programming under uncertainty, *Naval Research Logistics*, 35, 673–695.

- [90] I. M. Stancu-Minasian (1982) Recent results in stochastic programming with multiple objective function, in M. Grauer, A. Lewetowski et A. P. Wierzbicki (eds.), Multi-Objective and Stochastic optimisation, IIASA Collaborative Proceedings Series CP-S12.
- [91] I.M. Stancu-Minasian (1984) Stochastic Programming with Multiple Objective Functions. D. Reidel Publishing Company, Dordrecht. Optimization, Cluj-Napoca, pp. 321–328.
- [92] R. E. Steuer (1986) Multiple Criteria Optimisation : Theory, Computation and Application, Wiley.
- [93] A. Syberfeldt, A. Ng, R. I. John et P. Moore (2009). Multi-objective evolutionary simulation-optimisation of a real-world manufacturing problem. Robotics and Computer-Integrated Manufacturing, 25, 926–931.
- [94] J. Teghem (1983) Multiobjective and stochastic Linear programming, Foundations of Control Engineering, 8, 3-4, 225–232.
- [95] J. Teghem, D. Dufrane, M. Thauvoys, P. Kunsch (1986) STRANGE: an interactive method for multi-objective linear programming under uncertainty. Eur J Oper Res 26:65–82
- [96] Teghem J., Kunsch P. (1986) A survey of techniques to determine the efficient solutions to multi-objective integer linear programming, Asia Pacific Journal of Oper. Res. 3, 95–108.
- [97] Teghem J, Kunsch PL (1986) Interactive methods for multi-objective integer linear programming. In: Fandel G, Grauer M, Kurzhanski A, Wierzbicki AP (eds) Large-scale modelling and interactive decision analysis, lecture notes in economics and mathematical systems, vol 273. Springer, Berlin, pp 75–87

- [98] J. Teghem, (1990). Stochastic versus Fuzzy Approches to Multiobjective Mathematical Programming under Uncertainty, *An Interactive Method for Multiobjective Stochastic Linear Programming* Dordrech : Kluwer Academic Publishers, 103–115.
- [99] J. Teghem (1990) Strange-Momix: An interactive method for mixed integer linear programming, in: R. Slowinski, J. Teghem (Eds.), *Stochastic Versus Fuzzy Approaches to Multiobjective Mathematical Programming Under Uncertainty*, Kluwer Academic Publishers, Dordrecht, pp. 101–115.
- [100] J. Teghem (2012) *Recherche Opérationnelle - Tome 1, méthodes d'optimisation, ellipses marketing*
- [101] F. Tricoire, A. Graf et W. J. Gutjahr (2012). The bi-objective stochastic covering tour problem. *Computers et Operations Research*, 39, 1582–1592.
- [102] B. Urli, R. Nadeau (1990) Multiobjective stochastic linear programming with incomplete information: A general methodology, In: R. Slowinski, J. Teghem (Eds.), pp. 131–161.
- [103] R. Van Slyke, R.J.-B. Wets (1969) L-shaped linear programs with applications to optimal control and stochastic programming, *SIAM J. Appl. Math.* 17 : 638–663.
- [104] D. Walkup, R.J.-B. Wets (1967) Stochastic programs with recourse. *SIAM, Journal on Applied Mathematics*, 15 : 1299–1314.
- [105] R. J. B Wets (1966) Programming under uncertainty : the equivalent convex program. *SIAM Journal on Applied Mathematic*, 14 : 89–105.
- [106] R.J.-B. Wets (1974) Stochastic programs with fixed recourse: the equivalent deterministic program. *SIAM Review*, 16:309–339.
- [107] Willem K. Klein Haneveld et Maarten H. van der Vlerk (1999) Stochastic integer programming : general models and algorithms. *Ann. Oper. Res.*, 85 : 39–57.
- [108] R. D. Wollmer (1980) Two-stage linear programming under uncertainty with 0-1 integer first-stage variables, *Math Programming* 19, 279–288.

- [109] Zhi-Long Chen et Warren B. Powell (1998) A convergent cutting-plane and partial-sampling algorithm for multistage stochastic linear programs with recourse. Technical Report SOR-97-11, Department of Civil Engineering and Operations Research, Princeton University, Princeton, NJ 08544.
- [110] G. Zoutendijk (1960) *Methods of Feasible Directions*. Elsevier, Amsterdam/D. Van Nostrand, Princeton, New Jersey.