

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA
RECHERCHE SCIENTIFIQUE



UNIVERSITE MOULOU D MAMMERI - TIZI OUZOU
FACULTE DES SCIENCES
DEPARTEMENT DE MATHEMATIQUES

MEMOIRE DE FIN D'ETUDE

En vue de l'obtention du diplôme de MASTER en Mathématiques Appliquées à la
Gestion

Thème :

Optimisation discrète et application aux
problèmes stochastiques

Préparé par :

SFIHI Malika
MEGUELLATI Baya

Encadré par:

Mme RABIA Fatima
Maître de conférences A

Devant le jury composé de :

Mme LESLOUS Fadila M.A.A Président

Mme RABIA Fatima M.C.A Rapporteur

Mr KASDI Kamel M.A.A Examineur

Année universitaire 2017 - 2018

Remerciements

Nous remercions Dieu tout puissant qui nous a ouvert les portes du savoir et qui nous a éclairé le chemin et permis de mener à bien ce travail.

Nous tenons à remercier Mme RABIA de nous avoir fait l'honneur d'être promotrice de ce mémoire.

Nous remercions également les membres du jury pour avoir accepté d'évaluer notre travail.

Un grand merci à toutes les personnes qui nous ont soutenues de près ou de loin au cours de la réalisation de ce modeste travail.

Enfin, je remercie chaleureusement nos chers enseignants pour leur soutien et leurs encouragements.

Dédicaces

Je dédie ce modeste travail

À mes chers parents

qui m'ont toujours comblé de leur amour et leur grand affection

À mes chères sœurs

Hala, Assma, Rima,

À mes frères

Rabah et Nadji.

À mes très chers nièces et neveux

Sohaib, Romaïssa, Lydia

À mes chers cousins et cousines spécialement

Rima, Amira, Mehdi et Abdennour.

À mes chères amies spécialement

Ouafa, Hakima, Sabrina, Fahima, Sadia, Samia, Dihia et Sabrina.

À ma binome et sa famille

MEGUELLATI Baya

Dédicaces

Je dédie ce modeste travail

À mes chers parents

En premier lieu à ma chère mère pour son soutien et son dévouement tout au long de mes études et à mon cher père que DIEU le grand dans son vaste paradis .

À mes frères

Remdan, Slimane, Rachid, Houcine, Smail.

À ma belle sœur

Nora.

À mes très chers nièces et neveux

Lina, Ilyas, Sarah .

À mes chères amies spécialement

Ouafa, Hakima, Sabrina, Lamia, Sadia, Samia, Kamilia.

À ma binome et sa famille .

À mon mari et sa famille .

SFIHI Malika

Table des matières

Introduction générale	3
1 État de l'art de la programmation en nombres entiers	5
1.1 Introduction	5
1.2 Programmation linéaire en nombres entiers	5
1.3 La méthode résolution	6
1.3.1 Algorithme Branch and Bound	6
1.3.2 Exemple illustratif	7
1.4 Méthode des troncatures de Gomory	9
1.4.1 Processus itératif de l'algorithme Gomory	10
1.4.2 Règle pratique pour générer la coupe Gomory	10
1.4.3 Exemple illustratif	11
1.5 La méthode Poly-blocs	16
1.5.1 Rappels et notations	17
1.5.2 Algorithme de la méthode Poly-blocs	17
1.5.3 Exemple illustratif	19
1.6 Conclusion	21
2 Programmation linéaire stochastique	22
2.1 Introduction	22
2.2 Forme générale d'un programme linéaire stochastique	22
2.3 Approche passive ou "Wait and See"	22
2.4 L'approche active ou "Here and Now"	24
2.4.1 Cas des objectifs aléatoires	24
2.4.2 Cas des contraintes aléatoires	26
2.4.2.1 Le modèle avec seuil de probabilité sur les contraintes	26
2.4.2.2 Modèle avec recours	28
2.4.2.3 Exemple numérique	29
2.5 Conclusion	32
3 Application de la méthode Poly-blocs à un problème stochastique	33
3.1 Introduction	33
3.2 Position du problème	33
3.3 Application	35
3.3.1 Matlab	38
3.3.1.1 Introduction	38
3.3.2 Lingo	40
3.3.2.1 Introduction	40
Conclusion générale	42

Annexe1	43
Annexe2	47
Bibliographie	47

Introduction générale

La recherche opérationnelle est tout d'abord l'art de comprendre un problème réel et de pouvoir le transformer en un modèle mathématique. C'est l'art de modéliser ce problème afin d'extraire ses propriétés structurelles et de caractériser ses solutions. C'est l'art d'exploiter cette caractérisation afin de déterminer des algorithmes qui calculent ces solutions mais aussi de mettre en évidence des limites sur l'efficacité et l'efficacités de ces algorithmes. C'est enfin l'art d'enrichir ou d'abstraire les modèles construits afin d'augmenter leur puissances, leur probabilité et leur capacité à décrire mathématiquement (et informatiquement) d'autres problèmes similaires ou non aux problèmes qui ont inspiré les modèles initiaux.

Vue sous cet angle, on comprend aisément pourquoi la recherche opérationnelle se situe au cœur des autres disciplines scientifiques. C'est un carrefour où se rencontre l'économie, les mathématiques et l'informatique.

On peut dire aussi que la recherche opérationnelle est l'ensemble des méthodes mathématiques d'aide à la décision. Ces méthodes sont divisées en deux catégories : Les méthodes déterministes qui traitent des problèmes dans lesquelles toutes les données sont connues d'une façon précise et les méthodes stochastiques ou floues qui conviennent plus à la réalité par la non-négligence des risques et de l'incertitude. C'est à cette deuxième catégorie de problèmes que nous nous intéressons dans ce mémoire. Cependant, nous nous limiterons au cas stochastique.

L'optimisation stochastique a de nombreuses applications dans des problèmes de gestion, d'économie et de finance. C'est la situation où l'on fait face à des systèmes dynamiques évoluant dans des conditions d'incertitude et où il s'agit de prendre des décisions à chaque date afin d'optimiser un critère économique. Les données des problèmes traités, ou certaines d'entre elles, dépendent d'un paramètre extérieur non maîtrisé (niveau de croissance, prix du pétrole, taux d'inflation, degré de pluviométrie, ...etc. Elles sont considérées comme des variables aléatoires de distributions connues. L'optimisation stochastique est une discipline qui s'est développée dans les années cinquante. Elle a connu de nombreux développements et a donné lieu à de très nombreux ouvrages dont *Kall*[13], *Wets* [24].

Il est important de noter qu'un problème linéaire stochastique sera « mal posé » du point de vue mathématique et qu'il sera donc essentiel de lui donner un sens en lui associant un problème bien posé (déterministe) équivalent, mais qui ne sera pas nécessairement linéaire. L'efficacité de l'utilisation des équivalents déterministes est liée à leurs propriétés de linéarité ou de convexité. Ce point est important en pratique, c'est en effet pour les problèmes linéaires ou convexes que nous connaissons ou nous pouvons développer des algorithmes de résolution efficaces.

La nature d'une application impose parfois l'intégration d'une contrainte d'intégrité dans

son modèle. Nous sommes donc amenés à considérer des problèmes avec des variables de décision entières. Même si les méthodes de résolution des problèmes d'optimisation en nombres entiers n'est pas un sujet récent, celui-ci vit un renouveau depuis quelques années. Pour preuve, les publications sur ce sujet sont abondantes actuellement et de nombreux domaines scientifiques tentent d'appliquer ces méthodes à leurs problèmes spécifiques. On peut attribuer le regain d'intérêt pour ces techniques à l'augmentation de la puissance des ordinateurs qui permet de simuler, en temps interactif, des systèmes composés de plusieurs centaines d'actions.

En résumé, dans ce mémoire, nous traiterons des problèmes qui combinent le caractère probabiliste et le caractère discret.

Dans le premier chapitre, nous présentons les méthodes de résolution des problèmes linéaires en nombres entiers les plus utilisées, à savoir, la méthode *Branch and Bound*[3, 18, 19] et la méthode des troncatures de *Gomory*[12]. Puis nous exposons la méthode *poly-blocs*[22, 23] destinée à la résolution des programmes mathématiques monotones[16] car, certains problèmes stochastiques se transforment parfois en problèmes non linéaires dont les fonctions objectifs et/ou contraintes sont des fonctions monotones.

Le Deuxième chapitre traite de diverses techniques utilisée en programmation linéaire stochastique . Nous exposons les deux approches “*Wait and See*” et “*Here and Now*”.

Le troisième chapitre est consacré à l'application de la méthode *poly-blocs* à un problème réel de gestion. Nous terminons par la présentation de perspectives ouvertes par ce travail et les futures directions de recherche.

Chapitre 1

État de l'art de la programmation en nombres entiers

1.1 Introduction

De nombreux problèmes d'optimisation, issus de domaines d'applications très divers, peuvent être modélisés comme des programmes linéaires ou non linéaires dont les variables de décision sont contraintes à être entières. Les programmes ainsi obtenus sont souvent difficiles à résoudre du fait que l'espace de recherche n'est pas convexe mais discret. Toutefois, la programmation mathématique en nombres entiers est un outil très utile de modélisation et de résolution de nombreux problèmes d'optimisation.

1.2 Programmation linéaire en nombres entiers

La programmation linéaire en nombres entiers est un domaine très riche de la programmation mathématique. Les recherches dans ce domaine sont nombreuses et ont vu leurs naissances en 1958. Un problème de programmation linéaire en nombres entiers (*PLNE*) est un programme mathématique dont la fonction objectif et les contraintes sont des fonctions linéaires et dans lequel il y a une contrainte supplémentaire que les variables sont entières.

La forme générale d'un programme linéaire en nombres entiers se présente comme suit :

$$(PLNE) \begin{cases} Z = c'x \rightarrow \max(\min) \\ Ax(\leq, \geq, =)b \\ x_j \geq 0, j = 1, \dots, n \\ x_j \text{ entier}; j = 1, \dots, n \end{cases}$$

En général, on suppose que A et b sont composés uniquement de valeurs entières et $c \in R^n$. x_j entier ($j = 1 \dots, n$) désigne les contraintes d'intégrité.

On appelle relaxation continue, le problème obtenu de (*PLNE*), en supprimant les contraintes d'intégrité .

Le programme relaxé du problème (*PLNE*) est donné comme suit :

$$(Pr) \begin{cases} Z = c'x \rightarrow \max(\min) \\ Ax(\leq, \geq, =)b \\ x_j \geq 0, j = 1, \dots, n \end{cases}$$

1.3 La méthode résolution

La méthode *branch – and – bound* est une méthode basée sur une énumération "intelligente" des solutions admissibles d'un problème d'optimisation combinatoire. On représente l'exécution de la méthode branch-and-bound à travers une arborescence. La racine de cette arborescence représente l'ensemble de toutes les solutions du problème considéré. Dans ce qui suit, nous résumons la méthode branch-and-bound sur des problèmes de minimisation ou maximisation et pour l'appliquer, nous devons être en possession : [2, 18, 19, 20].

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

La méthode commence par considérer le problème de départ avec son ensemble de solutions, appelé la racine.

Des procédures de bornes inférieures et supérieures sont appliquées à la racine. Si ces deux bornes sont égales, alors une solution optimale est trouvée. Le processus s'arrête.

Sinon, l'ensemble des solutions est divisé en deux ou plusieurs sous-problèmes, devenant ainsi des enfants de la racine.

La méthode est ensuite appliquée récursivement à ces sous-problèmes, engendrant ainsi une arborescence. Si une solution optimale est trouvée pour un sous-problème, elle est réalisable, mais pas nécessairement optimale, pour le problème de départ. Comme elle est réalisable, elle peut être utilisée pour éliminer toute sa descendance : si la borne inférieure d'un nœud dépasse la valeur d'une solution déjà connue, alors on peut affirmer que la solution optimale globale ne peut être contenue dans le sous-ensemble de solution représenté par ce nœud. La recherche continue jusqu'à ce que tous les nœuds explorés ou éliminés.

1.3.1 Algorithme Branch and Bound

On procède par itérations successives [2, 18, 19, 20].

1. **Initialisation** : on résout le problème relaxé $\{Z_{max[ou min]}, x \in \Omega\}$, où Ω est un ensemble. Si la solution est entière alors *FIN*, sinon il faut trouver un majorant Z_u de la fonction économique : soit en calculant Z sur un point intérieur entier du polyèdre, soit en prenant $Z_u = +\infty$.

2. **Séparation(K^{ième} itération)** : on considère le sous-ensemble Ω_k de solution, avec $\Omega_1 = \Omega$. On choisit une composante non déjà entière (celle qui a la plus grande partie fractionnaire) de la solution optimale du sous-problème $\{Z_{max[ou min]}, x \in \Omega\}$, et on partitionne Ω_k en Ω'_k et Ω''_k , en ajoutant des contraintes qui excluent la partie fractionnaire de la composante choisie.

3. **Évaluation** : Pour chaque sous-problème $\{Z_{max[ou min]}, x \in \Omega'_k\}$, et $\{Z_{max[ou min]}, x \in \Omega''_k\}$, on calcule la solution optimale $Z_{L'k}$ et $Z_{L''k}$.

4. **Stérilisation** : on examine chaque sous-ensemble susceptible de contenir la solution optimale, et on stérilise un sous-ensemble si :

- $Z_L \leq Z_u$, [respectivement $Z_L \geq Z_u$], avec Z_u la solution courante et Z_L la solution père.

- Le sous-problème a un ensemble vide de solution.
- Z_L est obtenue avec une solution entière (évaluation exacte) et $Z_L > Z_u$, [respectivement $Z_L < Z_u$] .

5. **Test** : Si tous les sous-ensembles sont stériles, alors FIN, sinon on retourne en 1.

- Z_u : la solution courante
- Z_L : la solution père.

Représentation

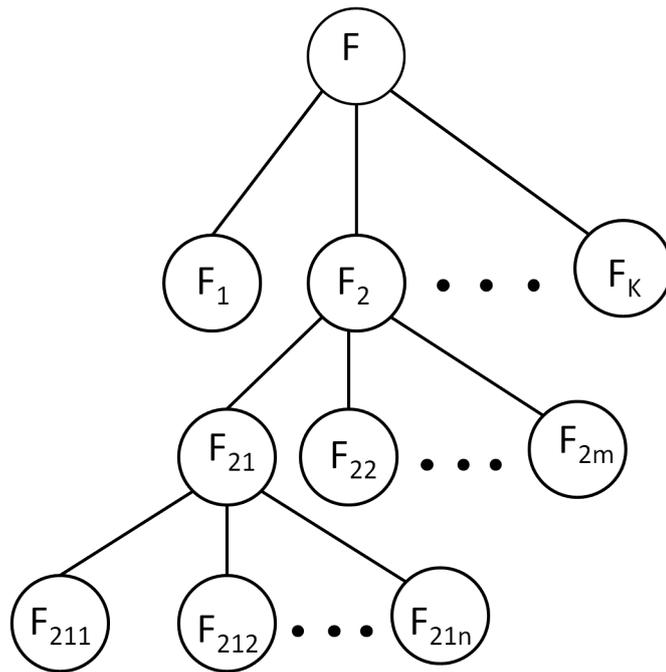
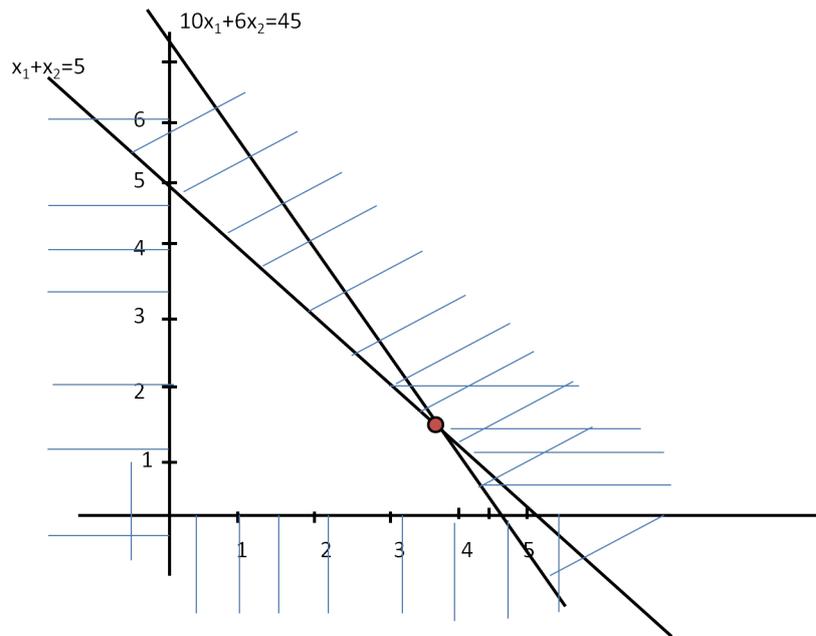


Figure1.2 : Représentation des sous – problèmes Branch and Bound

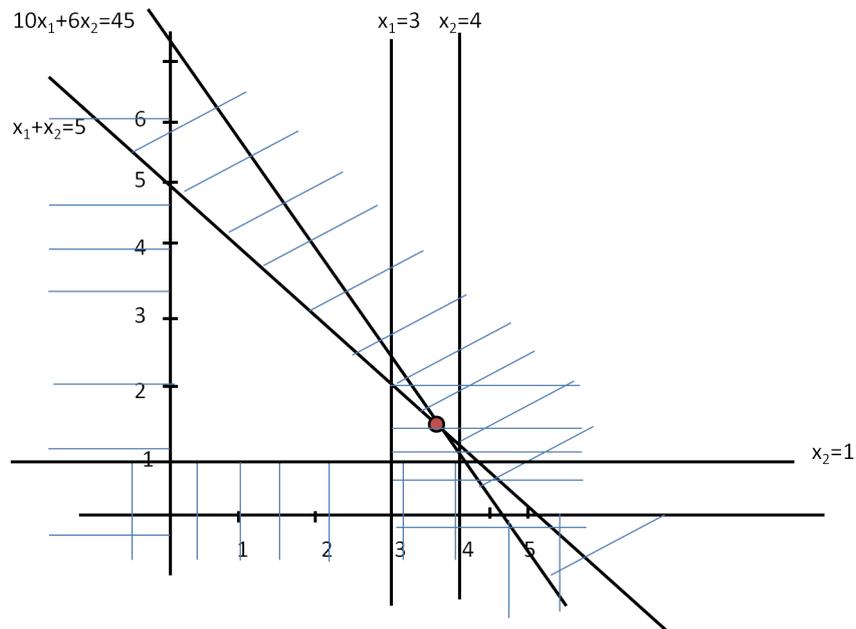
1.3.2 Exemple illustratif

$\max \rightarrow Z = 5x_1 + 4x_2$	relax : $\max \rightarrow Z = 5x_1 + 4x_2$
$x_1 + x_2 \leq 5$	$x_1 + x_2 \leq 5$
$10x_1 + 6x_2 \leq 45$	$10x_1 + 6x_2 \leq 45$
$x_1, x_2 \geq 0$	$x_1, x_2 \geq 0$
x_1, x_2 entiers	



- La solution optimale de la relaxation de F est $(3.75, 1.25)$ et $Z = 23.75$
- $U = -\infty$ valeur initiale
- Liste des sous problèmes actifs est $\{F\}$
- $b(F) = 23.75$
- Création de sous problème actifs
 $X_1 \leq \lfloor 3.75 \rfloor = 3$, $X_2 \geq \lceil 1.25 \rceil = 1$

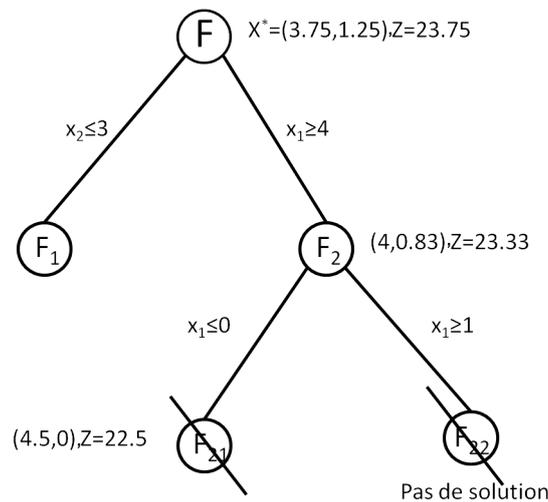
F_1	F_2
$\max \rightarrow Z = 5x_1 + 4x_2$	$\max \rightarrow Z = 5x_1 + 4x_2$
$x_1 + x_2 \leq 5$	$x_1 + x_2 \leq 5$
$10x_1 + 6x_2 \leq 45$	$10x_1 + 6x_2 \leq 45$
$x_1 \leq 3$	$x_2 \geq 1$
x_1, x_2 entiers	x_1, x_2 entiers



- Liste des sous problèmes actifs est $\{F, F_1, F_2\}$
- La solution de F_1 est $X^* = (3,2)$, $Z = 23$
- La solution de F_2 est $(4, 0.83)$, $Z = 23.33$
- Solution de F est $X^* = (3,2)$
- $b(F_1) = 23 > -\infty$
- $U = 23$
- Liste des sous problèmes actifs est $\{F, F_2\}$
- La solution de la relaxation de F_2 est $X^* = (4, 0.83)$
- $b(F_2) = 23.33$
- $b(F_2) > U$ on crée deux sous problèmes.

F_{22}	F_{21}
$\max \rightarrow Z = 5x_1 + 4x_2$	$\max \rightarrow Z = 5x_1 + 4x_2$
$x_1 + x_2 \leq 5$	$x_1 + x_2 \leq 5$
$10x_1 + 6x_2 \leq 45$	$10x_1 + 6x_2 \leq 45$
$x_2 \geq 4$	$x_2 \geq 4$
$x_2 \leq 0$	$x_2 \geq 1$
x_1, x_2 entiers	x_1, x_2 entiers

- Liste des sous problèmes actifs $\{F_1, F_2, F_{21}, F_{22}\}$
- F_{22} n'a pas de solution
- Solution de F_{21} est $(4.5, 0)$, $Z = 22.5$
- $b(F_{21}) = 22.5 < 23$ on supprime F_{21}
- La solution optimale du problème est donc $X^* = (3, 2)$ avec $Z = 23$



1.4 Méthode des troncatures de Gomory

La méthode des troncatures, dite également “ méthode des plans sécants ” ou “ méthode des coupes ” développée par *Ralph – Gomory* en 1958, est une méthode de résolution des programmes linéaires en nombres entiers (PLNE) . L’idée principale de cette méthode est d’ajouter des contraintes qui n’excluent aucun point entier réalisable[12].

La méthode consistera à ajouter de telles contraintes linéaires une par une, jusqu'à ce que la solution optimale du problème relaxé soit entière. Les contraintes ajoutées sont appelées troncatures ou coupes.

1.4.1 Processus itératif de l'algorithme Gomory

1. Étant donné un ($PLNE$), résoudre le (P_r) correspondant à l'aide de l'algorithme du simplexe. Si la solution optimale du ($PLNE$) est obtenue, la résolution est terminée.[12, 18, 19, 20]
2. Si une ou plusieurs variables de bases dans la solution optimale du (P_r) ne sont pas entières, on doit alors générer, à partir d'une des lignes du tableau (celle dont la partie fractionnaire pour la variable de base correspondante est la plus élevée), une contrainte supplémentaire dite coupe de *Gomory*. Cette contrainte est alors ajoutée au tableau optimal du (P_r) et on détermine le nouveau tableau optimal à l'aide de la méthode duale du simplexe.
3. Si les variables de base dans le nouveau tableau optimal sont entières, nous avons obtenu la solution optimale du($PLNE$). La résolution est terminée.
4. Sinon on doit générer (à partir du dernier tableau optimal) une nouvelle coupe de *Gomory*, l'ajouter au dernier tableau optimal et trouver la nouvelle solution optimale à l'aide de la méthode duale du simplexe.
 - Si la solution optimale obtenue est à valeurs entières, la résolution est terminée.
 - Sinon, on répète la procédure jusqu'à l'obtention d'une solution optimale entière.

1.4.2 Règle pratique pour générer la coupe Gomory

1. On choisira d'abord dans le tableau optimal, la variable de base qui présente la plus grande partie fractionnaire et on déduit l'équation (la contrainte) correspondante exprimant toute les variables du tableau en fonction de la valeur x_{B_i} de la variable base[12].

2. Pour générer la nouvelle contrainte (coupe de gomory), on remplace tout simplement tous les coefficients dans la contrainte obtenue du tableau optimal par sa partie fractionnaire correspondante et déclarons par la suite que l'expression résultante doit être plus grande ou égale à la partie fractionnaire du second membre c'est-à-dire la partie fractionnaire de la valeur x_{B_j} choisie en (1) dans l'algorithme, et la partie entière et partie fractionnaire d'un nombre est :

Pour un nombre réel "a" la somme de sa partie entière $[a]$ et de sa partie fractionnaire f s'écrit :

$$a = [a] + f$$

Où $N = [a]$ est le plus grand nombre entier inférieur ou égal au nombre a ($N \leq a$) et $0 < f < 1$. La partie fractionnaire d'un nombre réel "a" est donc la différence entre ce nombre et sa partie fractionnaire.

— L'organigramme de l'algorithme de Gomory

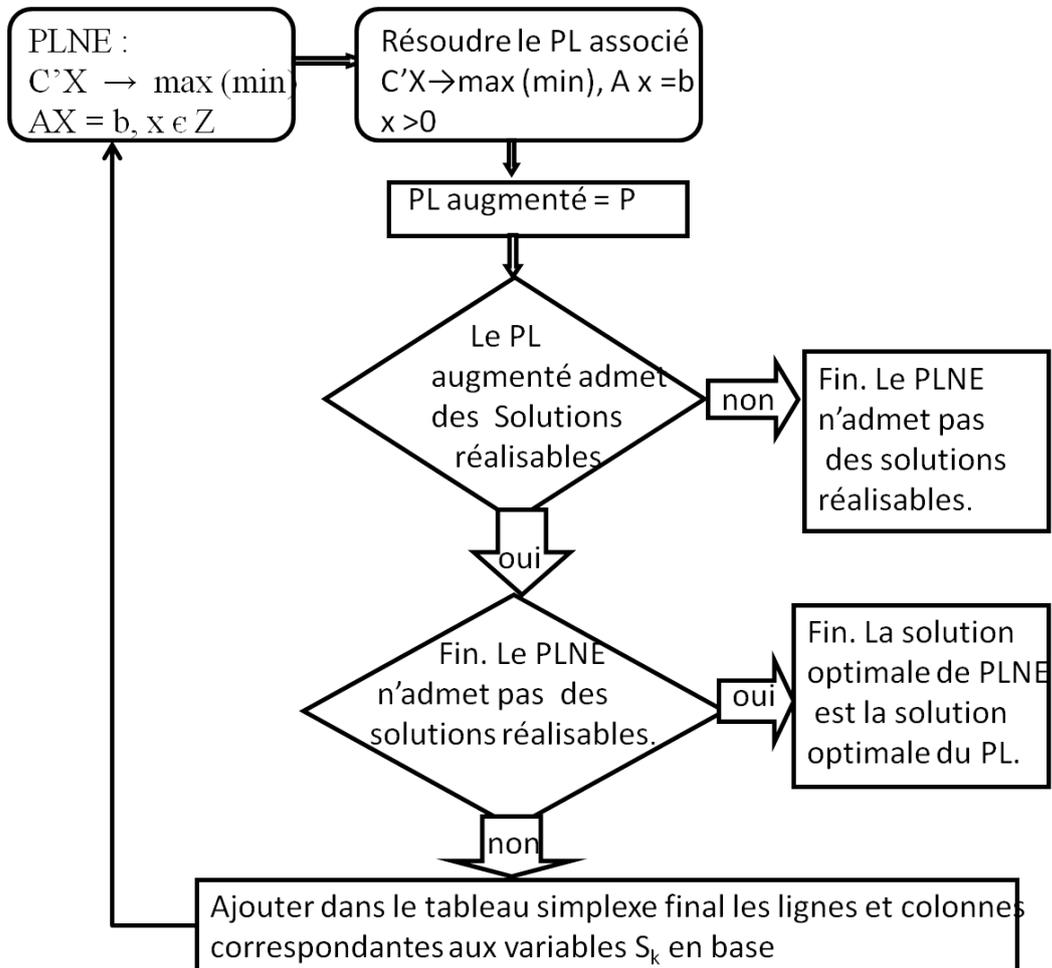


Figure 1.3 : L'organigramme de l'algorithme de Gomory

1.4.3 Exemple illustratif

Soit le programme linéaire en nombres entiers suivant

$$(PLNE) \begin{cases} Z = 100x_1 + 120x_2 \rightarrow \max \\ 3x_1 + 4x_2 \leq 4100 \\ x_1 + 3x_2 \leq 2400 \\ 2x_1 + 2x_2 \leq 2625 \\ x_1 \geq 0, x_2 \geq 0, \text{entier} \end{cases}$$

En ajoutant les variables d'écart respectivement dans chacune des contraintes du programme linéaire relaxé, on obtient le problème linéaire continue (P_r) suivant :

$$(P_r) \begin{cases} Z = 100x_1 + 120x_2 \rightarrow \max \\ 3x_1 + 4x_2 + x_3 = 4100 \\ x_1 + 3x_2 + x_4 = 2400 \\ 2x_1 + 2x_2 + x_5 = 2625 \\ x_j \geq 0, j = 1, \dots, 5 \end{cases}$$

En appliquant l'algorithme du simplexe, (d'après l'équation) on obtient le tableau optimale suivant :

c_j		100	120	0	0	0	
c_b	variables de bases	x_1	x_2	x_3	x_4	x_5	solution de base
100	x_1	1	0	-1	0	2	1150
120	x_2	0	1	1	0	$-\frac{3}{2}$	162.5
0	x_4	0	0	-2	1	$\frac{5}{2}$	762.5
z_j		100	120	20	0	20	
$c_j - z_j$		0	0	-20	0	-20	$z = 134500$

La solution optimale est $x = (x_1, x_2) = (1150, 162.5)$.

Puisque la solution optimale de (P_r) n'est pas à valeurs entières, on doit poursuivre en construisant la première contrainte additionnelle.

Construction de la première contrainte additionnelle

On peut déterminer, à partir du tableau optimal précédent, une contrainte (supplémentaire) qui réduit l'ensemble des solutions réalisables (on élimine la solution optimale à valeur non entières obtenue précédemment) sans supprimer de solutions entière réalisables.

Il s'agit de créer la contrainte supplémentaire à partir d'une des contraintes du tableau optimal dont la valeur x_{B_j} n'est pas entière.

La pratique consiste à choisir la contrainte (la ligne du tableau optimal) dont la variable de base a la plus grande partie fractionnaire. Ici, comme la partie fractionnaire est identique, nous choisissons arbitrairement la 2^{ème} ligne du tableau.

La variable de base, que l'on veut rendre "entière" est donc x_2 . Pour construire la contrainte additionnelle, on doit utiliser les variables : x_1, x_2, x_3, x_4, x_5 et b que l'on exprime sous forme algébrique de la façon suivante :

$$0x_1 + x_2 + x_3 + 0x_4 - \frac{3}{2}x_5 = 162.5$$

On écrit chaque coefficient des variables comme la somme de sa partie entière et de sa partie fractionnaire :

$$0x_1 + (1 + 0)x_2 + (1 + 0)x_3 + 0x_4 + \left(-2 + \frac{1}{2}\right)x_5 = \left(162 + \frac{1}{2}\right)$$

Supposons maintenant que toutes les variables sont des entiers non négatifs, alors le membre de gauche est une quantité non négative et le membre de droite doit être égal à $+\frac{1}{2}$ un entier :

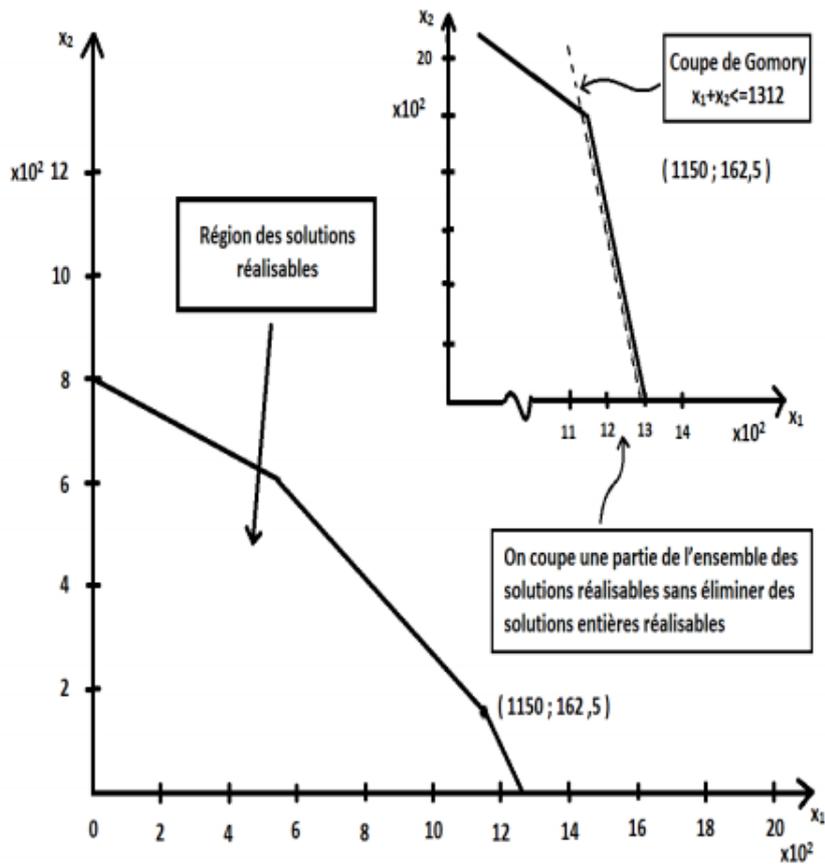
$a = [a] + f$		
a	$[a]$	$f = a - [a]$
1	1	0
$-\frac{3}{2}$	-2	$-\frac{3}{2} - (-2) = \frac{1}{2}$
162.5	162	$162.5 - 162 = \frac{1}{2}$

il s'ensuit que

$$\frac{1}{2}x_5 = \frac{1}{2} + \text{un entier négatif}$$

$$\frac{1}{2}x_5 \geq \frac{1}{2}$$

Cette inéquation constitue la coupe de Gomory que l'on doit ajouter au programme linéaire. Elle va nous permettre d'éliminer la valeur non entière [162.5] du tableau (on réduit la région des solutions réalisable comme nous allons le constater sur la figure suivante :



Ainsi, l'équation retenue du tableau optimal (pour la variable de base x_2) est :

$$0x_1 + x_2 + x_3 + 0x_4 + \left(-\frac{3}{2}\right)x_5 = 162.5$$

qui s'écrit aussi

$$0x_1 + (1 + 0)x_2 + (1 + 0)x_3 + 0x_4 + \left(-2 + \frac{1}{2}\right)x_5 = \left(162 + \frac{1}{2}\right)$$

$$\begin{array}{ccc} \downarrow & & \downarrow \\ \left[\frac{1}{2}x_5\right] & = & \frac{1}{2} \end{array}$$

La coupe de *Gomory* est donc

$$\frac{1}{2}x_5 \geq \frac{1}{2}$$

Déterminons le nouveau tableau optimal par la méthode duale du simplexe

Il n'est pas nécessaire, en ajoutant la contrainte additionnelle (*coupe de Gomory*)[12] au programme linéaire, de résoudre à nouveau tout le programme linéaire par la méthode du simplexe, il suffit d'ajouter la contrainte de *Gomory* (en incluant la variable d'écart appropriée) au dernier tableau optimal et de déterminer la nouvelle solution optimale à l'aide de l'algorithme dual du simplexe.[11]

Pour utiliser la méthode duale du simplexe, il faut modifier la contrainte de Gomory en multipliant les deux membres de l'équation $\frac{1}{2}x_5 \geq \frac{1}{2}$ par -1 , ce qui donne :

$$-\frac{1}{2}x_5 \leq -\frac{1}{2}$$

En ajoutant la variable d'écart x_6 , on obtient :

$$-\frac{1}{2}x_5 + x_6 = -\frac{1}{2}$$

Tableau avec l'ajout de la coupe de *Gomory*

c_j		100	120	0	0	0	0	
c_b	variables de base	x_1	x_2	x_3	x_4	x_5	x_6	solution de base
100	x_1	1	0	-1	0	2	0	1150
120	x_2	0	1	1	0	$-\frac{3}{2}$	0	162.5
0	x_4	0	0	-2	1	$\frac{5}{2}$	0	762.5
0	x_6	0	0	0	0	$-\frac{1}{2}$	1	$-\frac{1}{2}$
z_j		100	120	20	0	20	0	
$c_j - z_j$		0	0	-20	0	-20	0	$z=134500$

Le seul $x_{Bj} < 0$ correspond à x_6 , c'est la variable sortante ($i = k = 4$). La variable entrante est déterminée selon le critère suivant :

$$\min_j \left\{ \frac{c_j - z_j}{\mu_{kj}}, \mu_{kj} < 0 \right\} = \min_j \left\{ \frac{c_j - z_j}{\mu_{4j < 0}}, j = 3 \text{ et } 5 \right\}$$

Où

μ_{kj} = Le terme μ_{kj} est appelé le pivot et sert à effectuer l'opération de pivotage pour déterminer la nouvelle solution réalisable de base.

Il y a seulement μ_{45} qui est négatif, on obtient alors :

$$\frac{c_5 - z_5}{\mu_{45}} = \frac{-20}{-1/2} = 40$$

En introduisant la variable x_5 et en effectuant l'opération , on obtient le tableau optimal suivant :

Nouveau tableau optimal après la 1ère coupe de Gomory :

c_j		100	120	0	0	0	0	
c_b	variable de base	x_1	x_2	x_3	x_4	x_5	x_6	solution de base
100	x_1	1	0	-1	0	0	4	1146
120	x_2	0	1	1	0	0	-3	164
0	x_4	0	0	-2	1	0	5	760
0	x_5	0	0	0	0	1	-2	1
z_j		100	120	20	0	0	40	
$c_j - z_j$		0	0	-20	0	0	-40	$z=134\ 480$

La solution optimale est $x_1 = 1146$, $x_2 = 164$ et $z = 134\ 480$.

Puisque cette solution optimale est entière, il n'est pas nécessaire d'effectuer une autre coupe de *Gomory* . Nous avons obtenu la solution optimale entière du (*PLNE*).

1.5 La méthode Poly-blocs

La méthode "*Poly-blocs*" a été développée par *Tuy* [22, 23] pour résoudre les problèmes non linéaires en nombres entiers de la forme :

$$\begin{aligned} & \max f(x) \\ & \text{sc} \\ & g_i(x) \leq 0, \quad i = 1, \dots, m \\ & x \in X = \{x \in \mathbb{Z}_+^n \mid l_j \leq x_j \leq d_j, \quad j = 1, \dots, n\} \end{aligned} \tag{1.1}$$

où les deux fonctions f et les g_i sont des fonction monotones croissantes.

Le principe de la méthode "*poly-blocs*" est semblable à celui de la méthode "*Branch and Bound*". A chaque étape, le domaine réalisable est divisé en deux sous-domaines . Chacun deux est analysé afin de rejeter celui qui ne contient pas de solutions susceptibles d'être optimales pour le problème monotone .

1.5.1 Rappels et notations

Pour deux vecteurs $x, y \in \mathbb{R}^n$, $x \leq y$ veut dire $x_j \leq y_j$ pour chaque $j = 1 \dots n$. Si $l \leq d$ alors le box $[l, d]$ est l'ensemble de tous les $x \in \mathbb{R}^n$ satisfaisant $l \leq x \leq d$. Une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est dite croissante si $f(x) \leq f(y)$ quand $x \leq y$. [6]

$$G(x) = \max_{i=1, \dots, m} \{g_i(x)\} \quad (1.2)$$

La frontière de l'ensemble réalisable $S = \{x \in X \mid g_i(x) \leq 0, i = 1, \dots, m\}$ du problème (1.1) peut être exprimée comme l'ensemble $\Gamma = \{x \in X \mid G(x) = 0\}$

Soit $[\alpha, \beta]$ un box entier dans X avec $\alpha \in S$ et $\beta \notin S$. Puisque $G(\alpha) < 0$ et $G(\beta) > 0$ il doit exister un point x_b dans X qui satisfait

$G(x_b) = 0$ (i.e., $g_i(x_b) \leq 0, i = 1, \dots, m$ et il existe au moins un i telle que $g_i(x_b) = 0$), x_b est le point d'intersection de la ligne $x = \lambda^* \alpha + (1 - \lambda^*) \beta$, $0 \leq \lambda^* \leq 1$ et la frontière Γ .

Supposons maintenant que x_b n'est pas entier. Notons par $\lfloor x_b \rfloor$ le vecteur entier dont la $n^{i\text{ème}}$ composante est l'entier maximum inférieur ou égal à $x_{b,j}$, $j = 1, \dots, n$ et par $\lceil x_b \rceil$ le vecteur entier dont la $n^{i\text{ème}}$ composante est l'entier minimum ou égal à $x_{b,j}$, $j = 1, \dots, n$.

On a $x^F = \lfloor x_b \rfloor$ et $x^l = \lceil x_b \rceil$. Il est facile de voir que x^F est un point réalisable ($x^F \in S$) et x^l est irréalisable ($x^l \notin S$). La monotonie de f et g_i implique qu'il n'y a pas de points réalisables meilleurs que x^F dans $[\alpha, x^F]$ et il n'y a pas de points réalisables dans $[x^l, \beta]$. Par conséquent, nous pouvons supprimer les boxes entiers $[\alpha, x^F]$ et $[x^l, \beta]$ de $[\alpha, \beta]$.

Le théorème suivant montre comment diviser un domaine en sous-domaines.

Théorème 1(Sun et Al)[17]

Soit $A = [\alpha, \beta]$, $B = [\alpha, \gamma]$ et $C = [\gamma, \beta]$ des boxes entiers où $\alpha \leq \gamma \leq \beta$.

Alors $A \setminus B$ et $A \setminus C$ peuvent être partitionnés en au plus n nouveaux boxes entiers.

$$A \setminus B = \bigcup_{j=1}^n \left(\prod_{k=1}^{j-1} [\alpha_k, \gamma_k] \times [\gamma_j + 1, \beta_j] \times \prod_{k=j+1}^n [\alpha_k, \beta_k] \right) \quad (1.3)$$

$$A \setminus C = \bigcup_{j=1}^n \left(\prod_{k=1}^{j-1} [\gamma_k, \beta_k] \times [\alpha_j, \gamma_j - 1] \times \prod_{k=j+1}^n [\alpha_k, \beta_k] \right) \quad (1.4)$$

La méthode consiste à trouver un point réalisable x^F et un point irréalisable x^l et générer des boxes entiers en utilisant les formules (1.3) et (1.4). La meilleure solution obtenue lors de la génération de boxes entiers est conservée en tant que solution optimale courante. De plus, par la monotonie du problème, un box entier $[\alpha, \beta]$ avec $f(\beta)$ inférieur à la valeur de l'objectif $f(x^F)$ peut être rejeté.

1.5.2 Algorithme de la méthode Poly-blocs

L'algorithme de la méthode poly-blocs utilise la méthode de bisection pour trouver le point d'intersection x_b de la ligne $x = \lambda^* \alpha + (1 - \lambda) \beta$ avec la frontière Γ . A chaque itération

de l'algorithme, le domaine réalisable est divisé en deux sous-domaines [4, 6]. Chaque sous-domaine est analysé afin de rejeter celui qui ne contient pas un point réalisable qui donne une valeur de fonction objectif supérieure à la valeur fournie par la solution courante.

Initialisation

Soit $l = (l_1, l_2, \dots, l_n)$ et $d = (d_1, d_2, \dots, d_n)$

Si l est irréalisable, alors le problème (1.1) n'a pas de solution réalisable ;

Si d est réalisable, alors est la solution optimale à (1.1), le processus s'arrête

Autrement $x_{opt} = l$, $f_{opt} = f(x_{opt})$, $X^{(1)} = [l, d]$ on pose $k = 1$

Étape 1 : (sélection d'un box et recherche du point d'intersection)

Sélectionner un box $[\alpha, \beta] \in X^k$ avec la plus grande valeur $f(\beta)$ de objectif f . On pose $X^k = X^{(k)} \setminus [\alpha, \beta]$

Utiliser la méthode de bisection pour trouver la racine λ^* de l'équation suivante :

$$G[\lambda\alpha + (1 - \lambda)\beta] = 0, \quad \lambda \in [0, 1]$$

Où G est défini dans par l'expression (1.2).

Poser $x_b^{(k)} = \lambda^*\alpha + (1 - \lambda^*)\beta$, $x^{Fk} = \lfloor x_b^{(k)} \rfloor$ et $x^{Ik} = \lceil x_b^{(k)} \rceil$

Si $x^{Fk} = x_b^{(k)}$, poser $x^{lk} = x_b^{(k)} + e_j$, où e_j est la partie fractionnaire dans \mathbb{R}^n avec $x_b^{(k)} + e_j \leq \beta$

Sinon, poser $x^{lk} = \lceil x_b^{(k)} \rceil$.

Si $f(x^{Fk}) > f_{opt}$, poser $x_{opt} = x^{Fk}$ et $f_{opt} = f(x^{Fk})$

Étape 2 : (Partition et suppression de boxes)

1. Appliquez la formule (1.4) pour diviser l'ensemble $\Omega_1^{(k)} = [\alpha, \beta] \setminus [x^{lk}, \beta]$ en une union de boxes entiers.

Poser $[\hat{\alpha}, \hat{\beta}]$ le box entier contenant le point x^{Fk} . Poser $\Omega_1^{(k)} = \Omega_1^{(k)} \setminus [\hat{\alpha}, \hat{\beta}]$

1. Appliquez la formule (1.3) pour diviser l'ensemble $\Omega_2^{(k)} = [\hat{\alpha}, \hat{\beta}] \setminus [\hat{\alpha}, x^{Fk}]$
2. Poser $Y^{(k)} = \Omega_1^{(k)} \cup \Omega_2^{(k)}$
3. Effectuez les opérations suivantes pour chaque box entier $[\alpha, \beta]$ généré dans le processus de division ci-dessus :
 - Si β est réalisable, retirer $[\alpha, \beta]$ de Y^k . En outre si $f(\beta) > f_{opt}$, poser $x_{opt} = \beta$
 - Si α est irréalisable, retirer $[\alpha, \beta]$ de Y^k
 - Si $f(\alpha) < f_{opt}$, retirer $[\alpha, \beta]$ de Y^k
 - Si α est réalisable, β est irréalisable et $f(\alpha) > f_{opt}$ poser $x_{opt} = \alpha$ et $f_{opt} = f(\alpha)$

Noter par Z^k l'ensemble des boxes entiers après le processus de suppression ci-dessus.

Étape 3 : (Mise à jour des boxes entiers)

Supprimer toutes les boxes entiers $[\alpha, \beta]$ dans $X^{(k)}$ avec $f(\beta) < f_{opt}$

Poser $X^{(k+1)} = X^{(k)} \cup Z^k$

Si $X^{(k+1)} = \emptyset$, arrêtez. Sinon, poser $k = k + 1$ et allez à l'étape 1.

Deux stratégies de sélection de boxes peuvent être utilisées à l'étape 1. La première stratégie consiste à sélectionner le box entier dans $X^{(k)}$ avec la valeur maximale de la fonction objectif du point de vue limite supérieure. La deuxième stratégie consiste à sélectionner le dernier box entier incluse dans $X^{(k)}$.

La convergence finie de l'algorithme peut être facilement vue à partir de la finitude de X et le fait qu'à chaque itération, au moins les points entiers x^{Fk} et x^{lk} sont retirés de $X^{(k)}$. L'algorithme procède successivement en raffinant la partition et en supprimant les boxes entiers qui ne contiennent pas de solutions prometteuses et se termine finalement en un nombre fini d'itérations.

1.5.3 Exemple illustratif

$$\begin{cases} \max f(x) = 3x_1x_2 - x_1 + 6x_2 \\ sc \\ 5x_1x_2 - 4x_1 - 4.5x_2 \leq 32 \\ x \in X = \{x \in Z^2 / 1 \leq x_j \leq 5, j = 1, 2\} \end{cases}$$

Initialisation :

$$l = (1, 1), d = (5, 5)$$

$l = (1, 1)$ est réalisable

$d = (5, 5)$, n'est pas réalisable

$$x_{opt} = (1, 1), f_{opt} = 8$$

$$X^{(1)} = [l, d] = [(1, 1), (5, 5)], \quad k = 1$$

Itérations 1 :

Etape1 : sélectionnons $[\alpha, \beta] = [l, d] = [(1, 1), (5, 5)]$

$$g[\lambda(1, 1) + (1 - \lambda)(5, 5)] = 0$$

$$g(5 - 4\lambda, 5 - 4\lambda) = 0$$

$$-86\lambda + 50.5 = 0$$

La méthode de bisection nous donne $\lambda = 0.5872$

Dans ce cas $x_b = (5 - 4\lambda, 5 - 4\lambda)$

$$x_b = (2.6512, 2.6512)$$

$$x^F = (2, 2) \quad x^l = (3, 3)$$

$$f(x^F) = 22 > 8 = f_{opt}$$

Posons $x_{opt} = (2, 2), \quad f_{opt} = 22$

Etape2 : Partitionnons $\Omega_1 = [\alpha, \beta] \setminus [x^l, \beta]$ en deux boxes .

$$\begin{aligned} \Omega_1 &= [(1, 1), (5, 5)] \setminus [(3, 3), (5, 5)] \\ &= [(\alpha_1, \alpha_2)(\beta_1, \beta_2)] \setminus [(\gamma_1, \gamma_2)(\beta_1, \beta_2)] \\ &= \{(\alpha_1, \gamma_1 - 1) \times (\alpha_2, \beta_2)\} \cup \{(\gamma_1, \beta_1) \times (\alpha_2, \gamma_2 - 1)\} \\ &= \{[1, 2][1, 5]\} \cup \{[3, 5][1, 2]\} \\ &= [(1, 1), (2, 5)] \cup [(3, 1), (5, 2)] \end{aligned}$$

$$x^F = (2, 2) \in [(1, 1), (2, 5)] = [\hat{\alpha}, \hat{\beta}]$$

On pose $\Omega_1^{(1)} = [(3, 1), (5, 2)]$

$$\begin{aligned} \Omega_2^{(1)} &= [\hat{\alpha}, \hat{\beta}] \setminus [\hat{\alpha}, x^F] \\ &= [(1, 1), (2, 5)] \setminus [(1, 1), (2, 2)] \\ &= [(\hat{\alpha}_1, \hat{\alpha}_2), (\beta_1, \hat{\beta}_2)] \setminus [(\hat{\alpha}_1, \hat{\alpha}_2), (\gamma_1, \gamma_2)] \\ &= \{[\gamma_1 + 1, \hat{\beta}_1][\hat{\alpha}_2, \hat{\beta}_2]\} \cup \{[\hat{\alpha}_1, \gamma_1][\gamma_2, \hat{\beta}_2]\} \\ &= \{[3, 2][1, 5]\} \cup \{[1, 2][3, 5]\} \\ &= [(1, 3), (2, 5)] \end{aligned}$$

$$Y^{(1)} = \Omega_1^{(1)} \cup \Omega_2^{(1)} = [(3, 1), (5, 2)] \cup [(1, 3), (2, 5)]$$

$$Z^{(1)} = Y^{(1)}$$

$$f(5.2) = 37$$

$$f(2.5) = 58$$

(2, 5) réalisable et $f(2, 5) = 58 > 22 = f_{opt}$

On pose $x_{opt} = (2, 5) \quad f_{opt} = 58$

On retire le box $[(1, 3), (2, 5)]$ de $Y^{(1)}$

$f(5.2) = 37 < f_{opt} = 58$ on rejete le box $[(3, 1), (5, 2)]$ de $Y^{(1)}$

Donc $Y^{(1)} = \emptyset \quad Z^{(1)} = \emptyset$

Etape 3 : Pour le box $[(3, 1), (5, 2)] \in X^{(2)}$ puisque $f(2, 5) = 58 = f_{opt}$ on rejete ce box de $X^{(2)}$.

$$\begin{aligned} \text{On obtient } X^{(2)} &= X^{(1)} \cup Z^{(1)} \\ &= \emptyset \end{aligned}$$

Le processus de résolution prend fin

La solution optimale est $x_{opt} = (2, 5)$ avec $f_{opt} = 58$

1.6 Conclusion

Nous constatons que la méthode poly-blocs ne nécessite pas que les objectifs ou les contraintes non linéaires satisfassent certaines propriétés mathématiques spécifiques. Elle évite aussi l'application de la dualité et l'ajout des contraintes de coupes qui apparaissent dans la résolution des problèmes de programmation en nombres entiers par d'entiers par la méthode de Gomory ou Branch and Bound. De plus, aucune linéarisation ou convexification des fonctions non linéaires n'est nécessaire ce qui la distingue des autres méthodes. Son efficacité est due au fait qu'il suffit de résoudre des équations simples par la méthode de bisection plutôt que des problèmes d'optimisation complexes pour trouver la ou les solutions optimales du problème donné.

Chapitre 2

Programmation linéaire stochastique

2.1 Introduction

La programmation linéaire stochastique est une technique qui se caractérise par le souci d'une meilleure appréhension des problèmes réels. Son objectif est de prendre en compte les phénomènes aléatoires intervenant en pratique, exclus par les méthodes déterministes .[1, 5]

2.2 Forme générale d'un programme linéaire stochastique

Le modèle général de programmation linéaire stochastique dans le cas d'une minimisation est le suivant

$$\begin{aligned} \min_{sc} Z(x, w) &= c^t(w) x \\ x \in S(w) &= \{x \in \mathbb{R}^n \mid A(w) x \leq b(w), x \geq 0\} \end{aligned} \quad (2.1)$$

Où $x \in \mathbb{R}^n$ est le vecteur de décision, $A(w)$ est une matrice ($m \times n$), $C(w)$ et $b(w)$ sont des vecteurs ayant n respectivement m composantes. Les coefficients de $A(w)$ ainsi que les composantes de $C(w)$ et $b(w)$ sont des variables aléatoires définies sur un espace de probabilité (Ω, E, P) donné, de distributions connues.

Il y a essentiellement deux différents types d'approches en programmation linéaire stochastique qui sont [5] :

1. L'approche passive ou "wait and see"
2. L'approche active ou "here and now"

2.3 Approche passive ou "Wait and See"

Elle désigne la situation dans laquelle le décideur peut attendre la réalisation des variables aléatoires pour prendre une décision. Dans ce cas, on peut s'intéresser à la fonction de répartition de la valeur optimale de $Z(x, w)$. [1, 5]

$$\begin{aligned} \gamma(w) &= \min c^t(w) x \\ &Ax \leq b \\ &x \geq 0 \end{aligned} \quad (2.2)$$

Pour $w \in \Omega$. Dans cette approche $c(w)$ est de la forme :

$$c(w) = (\alpha_1 w_1 + \beta_1, \alpha_2 w_2 + \beta_2, \dots, \alpha_n w_n + \beta_n)$$

α_j et β_j sont des constantes.

Cependant, puisqu'il n'est pas possible de trouver un vecteur $x^0 \in S = \{x \in \mathbb{R}^n \mid Ax \leq b, x \geq 0\}$ qui soit optimal pour tout $w \in \Omega$, *Tinter* [21] a proposé de remplacer la notion classique d'optimalité utilisée en programmation linéaire déterministe par le concept « d'efficacité avec probabilité 1 » donné la définition suivante :

Définition 2.1(*Tinter* [21])

$x^0 \in S$ est efficace avec probabilité 1 s'il n'existe aucun $x^1 \in S$ tel que

$$P \{w \mid c^t(w) x^1 \leq c^t(w) x^0\} = 1 \text{ et } P \{w \mid c^t(w) x^1 < c^t(w) x^0\} > 0.$$

Autrement dit, un point $x^0 \in S$ est efficace avec probabilité 1, s'il n'existe aucun autre point x^1 de S presque sûrement aussi bon que x^0 , et qui soit meilleur que x^0 avec une probabilité positive.

Dans ce cas, si l'on désigne par $G = (A_B^l, 1 \leq l \leq q)$ l'ensemble des sous matrices carrées de A de rang m régulières, associées aux solutions de base réalisables $x_l, 1 \leq l \leq q$ de S , la *Définition 2.1* permet de caractériser les régions de décision

$$\begin{aligned} \Omega_l &= \{w \mid \gamma(w) = c^t(w) x_l = \min c^t(w) x, x \in S\} \\ &= \left\{ w \mid c_B^l(w) (A_B^l)^{-1} A_N^l - c_N^l(w) \leq 0 \right\} \end{aligned}$$

Où $c_B^l(w)$ et $c_N^l(w)$ sont, respectivement, les composantes de base et hors base du vecteur $c(w)$ et A_N^l est la matrice hors base associée à x_l .

$$P_l = P(\Omega_l) = \int_{\Omega_l} f(w) dw$$

Où $f(w)$ est la fonction de densité de w . ($f(w) = f(w_1, \dots, w_n) > 0$)

La fonction de répartition de la valeur optimale $\gamma(w)$ de $Z(x, w)$ est donnée par

$$F_{\gamma(w)}(z) = \sum_{l=1}^q \int_{\Omega_{l_z}} f(w) dw$$

avec

$$\Omega_{l_z} = \{w \mid \gamma_l(w) \leq z\}$$

Même si cette définition est intéressante du point de vue théorique par le problème de répartition qu'elle permet de poser, elle l'est moins du point de vue pratique, car son application nécessite la détermination de toutes les bases et solutions de base réalisables de l'ensemble des décisions S , ce qui est fastidieux et irréaliste pour les problèmes de grande taille.

Le problème de répartition en programmation stochastique a été introduit par *Tinter* [21] et a fait l'objet de nombreux travaux, notamment, ceux de *Bereanu* [7, 8], *Prekopa* [15],...etc.

2.4 L'approche active ou "Here and Now"

C'est une approche basée sur le principe de prise de décision et le choix d'une stratégie sans connaître au préalable la réalisation des variables aléatoires. Elle a été développée pour palier à la difficulté du calcul de la fonction de répartition de l'approche passive. L'idée de base de cette approche est la transformation systématique des problèmes stochastiques en programmes déterministes équivalents. La question posée est selon quel critère choisir une stratégie x de S , ou encore, comment devons nous interpréter l'objectif $Z(x, w)$ et les contraintes $A(w)x \leq b(w)$ pour obtenir le programme déterministe équivalent [1][5].

2.4.1 Cas des objectifs aléatoires

Plusieurs critères d'optimisation sont utilisés dans la littérature pour transformer le problème stochastique en programme déterministe équivalent. On distingue :

a. Le critère espérance (E) : le E-modèle est le plus utilisé, il consiste à remplacer la variable aléatoire de l'objectif par son espérance mathématique pour obtenir le programme linéaire déterministe suivant :

$$\begin{cases} \min E(c(w))x \\ x \in S = \{x / Ax \leq b, x \geq 0\} \end{cases} \quad (2.3)$$

b. Le critère variance (V) : dans le cas où c est un vecteur aléatoire d'espérance \bar{c} , de matrice de covariance V . La variance de $c(w)x$ est $x^t V x$. De la minimisation de la variance résulte le V-modèle suivant :

$$\begin{cases} \min \sigma^2(x) = x^t V x \\ \text{sc} \\ x \in S = \{x / Ax \leq b, x \geq 0\} \end{cases} \quad (2.4)$$

c. Le critère espérance – variance (E – V modèle) : ce modèle consiste à minimiser la variance de $Z(x, w)$ tout en réalisant un niveau de rendement minimum Z_0 fixé préalablement par le décideur :

$$\begin{aligned} \min \sigma^2(x) &= x^t V x \\ \text{sc} \\ \bar{c}x &\geq Z_0, x \in S \end{aligned} \quad (2.5)$$

Où \bar{c} est l'espérance mathématique de $c(w)$: $\bar{c} = E[c(w)]$

d. Le critère de risque minimal (P – modèle) : la maximisation de la probabilité que la valeur de l'objectif est au moins égale à un certain niveau u choisi par le décideur est appelée P-modèle ou modèle de risque minimal :

$$\begin{cases} \max P(w/c(w)x \leq u) \\ \text{sc} \\ x \in S = \{x/Ax \leq b, x \geq 0\} \end{cases} \quad (2.6)$$

La solution de ce problème, dans le cas gaussien, est donnée par le programme fractionnaire suivant :

$$\begin{cases} \min \frac{\bar{c}x - u}{\sqrt{x^t V x}} \\ \text{sc} \\ x \in S \end{cases}$$

où \bar{c} est l'espérance mathématique de $c(w)$, V sa matrice de covariance et $x^t V x$ la variance de $c(w)x$:

e. Le critère de type K (Kataoka) [14] : la maximisation de α - fractile de la fonction de distribution de l'objectif où α est choisi par le décideur[14] :

$$\begin{cases} \min u \\ P(w/c(w)x \leq u) = \alpha \quad 0 < \alpha < 1 \\ x \in S \end{cases} \quad (2.7)$$

Dans le cas gaussien on a :

$$P(w/x(w)x \leq u) = P\left(w/\frac{c(w)x - \bar{c}x}{\sqrt{x^t V x}} \leq \frac{\bar{c}x - u}{\sqrt{x^t V x}}\right) = \phi\left(\frac{\bar{c}x - u}{\sqrt{x^t V x}}\right) \quad \text{pour } x \neq 0$$

Où ϕ est la fonction de répartition de la variable aléatoire normale centrée réduite. Donc :

$$u = \bar{c}x + \phi^{-1}(\alpha) \sqrt{x^t V x}$$

Par conséquent résoudre le problème (2,7), dans ce cas gaussien revient à résoudre le problème suivant :

$$\begin{cases} \min \bar{c}x + \phi^{-1}(\alpha) \sqrt{x^t V x} \\ \text{sc} \\ x \in S \end{cases} \quad (2.8)$$

La fonction $\bar{c}x + \phi^{-1}(\alpha) \sqrt{x^t V x}$ est convexe si $\phi^{-1}(\alpha) \geq 0$ ce qui revient à dire, que $\alpha \geq \frac{1}{2}$ ou bien $P(w/c(w)x \leq u) = \alpha \geq \frac{1}{2}$, Ceci nous garantit le minimum de l'objectif avec une probabilité supérieure ou égale à $\frac{1}{2}$. Le terme $\Phi^{-1}(\alpha) \sqrt{x^t V x}$ représente une pénalité pour l'acceptation d'un risque.

2.4.2 Cas des contraintes aléatoires

Dans ce paragraphe, nous supposons que l'objectif est déterministe ou qu'il a été rendu déterministe en appliquant l'un des critères précédents. Les contraintes stochastiques :

$$A(w)x \leq b(w)$$

Seront traitées suivant deux modèles différents : le modèle avec seuil de probabilité sur les contraintes « chance constrained programming » et le modèle avec recours.

2.4.2.1 Le modèle avec seuil de probabilité sur les contraintes

Cette technique a été introduite en programmation stochastique depuis les années soixante par *Charnes et Cooper*[10].

L'idée de la modélisation "chance-constrained" consiste à imposer que la violation des contraintes ne se produise qu'avec une probabilité fixée

Soit l'on impose un seuil de probabilité individuel α_i , avec $0 \leq \alpha_i \leq 1$ pour chaque contrainte i ($i = 1, \dots, m$) : il représente la probabilité avec laquelle la contrainte i doit être vérifiée.

Le problème déterministe équivalent est donc

$$\begin{cases} \min E(c(w))x \\ sc \\ P(A_i(w)x \leq b_i(w)) \geq \alpha_i, \quad i = 1, \dots, m \\ x \geq 0 \end{cases} \quad (2.9)$$

Soit l'on impose un seuil de probabilité global α , avec $0 < \alpha < 1$, pour l'ensemble des contraintes : il représente la probabilité avec laquelle les contraintes doivent être globalement vérifiées.

Le problème déterministe équivalent est donc :

$$\begin{cases} \min E(c(w))x \\ sc \\ P(A(w)x \leq b(w)) \geq \alpha \\ x \geq 0 \end{cases} \quad (2.10)$$

Dans les deux cas, la question est de savoir si les ensembles

$$S(\alpha_i) = \{x \in R^n \mid P[A_i(w)x \leq b_i(w)] \geq \alpha_i, \quad i = 1, \dots, m, \quad x \geq 0\}$$

et

$$S(\alpha) = \{x \in R^n \mid P[A(w)x \leq b(w)] \geq \alpha, \quad x \geq 0\}$$

Sont convexes.

La convexité des ensembles $S(\alpha)$ et $S(\alpha_i)$ dépend non seulement de la distribution de A et b mais aussi des seuils α et α_i .

Le théorème suivant est le seul établissant la convexité de $S(\alpha)$ ou $S(\alpha_i)$ sans prendre en compte la distribution de probabilité des variables aléatoires.

Théorème 2.4.2.1 (Kall [13])

$S(0)$ et $S(1)$ sont convexes.

Citons, à titre d'exemple, quelques conditions de convexité de $S(\alpha_i)$ ou $S(\alpha)$.

— *Cas où A est déterministe et b est aléatoire*

Pour ce cas spécial, la réponse à la question posée est très simple.

Si F_i est la fonction de répartition de b_i , alors :

$$\begin{aligned} S(\alpha_i) &= \{x \in R^n \mid P(A_i x \leq b_i(w)) \geq \alpha_i\} \\ &= \{x \in R^n \mid P(b(w) \leq A_i x) \leq 1 - \alpha_i\} \\ &= \{x \in R^n \mid A_i x \leq F_i^{-1}(1 - \alpha_i)\} \end{aligned}$$

$S(\alpha_i)$ est un ensemble de contraintes linéaires en x , donc il est convexe.

— *Cas où A et b sont des variables aléatoires normales :*

Cas 1 : A et b non indépendantes

Pour ce cas général, on suppose que (A_i, b_i) est un vecteur aléatoire normalement distribué de moyenne $\mu_i \in R^{n+1}$ et de matrice de covariance V_i . En vertu de la théorie des probabilités, la variable aléatoire $t_i(x) = A_i x - b_i$ a une distribution normale de

Moyenne $m_i(x) = \sum_{j=1}^n \mu_{ij} x_j - \mu_{i,n+1}$

et de

Variance $\sigma_i^2(x) = z^t V_i z$ avec $z = (x_1, x_2, \dots, x_n, -1)^t$ et $\sigma_i(x) > 0, \forall x$ car $x_{n+1} = -1$

$$\begin{aligned} S(\alpha_i) &= \{x \in R^n \mid P(t_i(x) \leq 0) \geq \alpha_i\} \\ &= \left\{ x \in R^n \mid P\left(\frac{t_i(x) - m_i(x)}{\sigma_i(x)} \leq \frac{-m_i(x)}{\sigma_i(x)}\right) \geq \alpha_i \right\} \\ &= \left\{ x \in R^n \mid \Phi\left(\frac{-m_i(x)}{\sigma_i(x)}\right) \geq \alpha_i \right\} \\ &= \{x \in R^n \mid m_i(x) + \Phi^{-1}(\alpha_i) \sigma_i(x) \leq 0\} \end{aligned}$$

Puisque $m_i(x)$ est affine en x et $\sigma_i(x)$ est convexe en x , la contrainte $(m_i(x) + \Phi^{-1}(\alpha_i) \sigma_i(x) \leq 0)$ est convexe si et seulement si $\Phi^{-1}(\alpha_i) \geq 0$, c'est-à-dire si et seulement si $\alpha_i \geq \frac{1}{2}$.

Cas 2 : A et b indépendantes

$$a_{ij} \sim N(\mu_{ij}, v_{ij}^2), \quad b_i \sim N(m_i, \sigma_i^2)$$

Alors la variable $A_i x - b_i$ a pour distribution :

$$N\left(\sum_{j=1}^n \mu_{ij} x_j - m_i, \sum_{j=1}^n v_{ij}^2 x_j^2 + \sigma_i^2\right)$$

$$S(\alpha_i) = \left\{ x \in R^n \mid \sum_{j=1}^n \mu_{ij} x_j - m_i + \Phi^{-1}(\alpha_i) \sqrt{\sum_{j=1}^n v_{ij}^2 x_j^2 + \sigma_i^2} \leq 0 \right\}$$

2.4.2.2 Modèle avec recours

Pour construire ce modèle, on procède en deux étapes.

Une décision x est choisie, préalablement à toute réalisation de l'aléatoire. Une fois qu'une réalisation $w \in \Omega$ est observée, une décision corrective représentée par un vecteur y ($k+1$) appelé recours est prise pour compenser la violation des contraintes qui y correspond. Cette compensation se fait par l'introduction d'une fonction de pénalité $q(w) y$ qui est en général linéaire. La minimisation de cette pénalité correspond au problème de recours ou de second niveau suivant :

$$\begin{cases} Q(x, w) = \min q^t(w) y \\ sc \\ W(w) y = b(w) - A(w) x \\ y \geq 0 \end{cases} \quad (2.11)$$

W ($m \times k$) étant la matrice de recours.

Considérées globalement, ces deux étapes fournissent le problème :

$$\begin{cases} \min_x (E(c^t(w) x + \min q^t(w) y)) \\ A(w) x + W(w) y = b(w) \\ y \geq 0 \end{cases} \quad (2.12)$$

Qui s'écrit de manière équivalente

$$\min_{x \in S} E(c^t(w) x + E(Q(w, x))) \quad (2.13)$$

Où

$$S = \{x \in R^n \mid \forall w \in \Omega, \exists y \geq 0 \mid A(w) x + W(w) y = b(w)\}$$

Pour que le problème (2.13) ait un sens, il faut que l'ensemble S soit non vide, autrement dit, il faut qu'il existe toujours, quelque soit la réalisation de l'aléatoire, un recours y possible (i.e. $Q(x, w) < \infty, \forall w \in \Omega$)

Wets[24] a montré que S est un ensemble convexe et la fonction $Q(x, w)$ est convexe.

Nous parlerons :

- De recours fixe, si la matrice de recours $W(w)$ est fixe i.e. $W(w) = W$.
- De recours (fixe) simple, si $W(I, -I)$ ou I est la matrice identité d'ordre m .

Dans ce cas, le vecteur y est décomposé en deux parties :

$y^+ (m \times 1)$: variable d'écart par excès ;

$y^- (m \times 1)$: variable d'écart par défaut.

Parallèlement le vecteur de pénalisation s'écrit $q(w) = (q^+(w), q^-(w))$ avec :

$$q^+(w) [b(w) - A(w)x] \text{ si } b(w) - A(w)x \geq 0$$

$$q^-(w) [A(w)x - b(w)] \text{ si } b(w) - A(w)x \leq 0$$

Le programme (2, 13) se met alors sous la forme

$$\begin{cases} \min_x E(c^t(w))x + E(\min_{y^+, y^-} q^+(w)y^+ + q^-(w)y^-) \\ y^+ - y^- = b(w) - A(w)x \\ y^+ \geq 0, y^- \geq 0 \end{cases}$$

2.4.2.3 Exemple numérique

$$\begin{cases} \min Z(x) = 2x_1 + x_2 \\ sc \\ x_1 + x_2 \leq 4 \\ 0.5x_1 + 0.3x_2 \geq b(w) \\ x_1 \geq 0, \quad x_2 \geq 0 \end{cases}$$

Où b suit une loi uniforme sur l'intervalle $[1.2, 1.6]$.

Ce problème correspond à la recherche du cout minimal pour une opération de fusion de deux types de minerai. La demande est aléatoire uniforme et un problème de capacité limite l'opération à 4 unités.

- Remplaçons en premier lieu la demande par son espérance $E(b(w)) = 1.4$
- Le problème :

$$\begin{cases} \min Z(x) = 2x_1 + x_2 \\ \text{sc} \\ x_1 + x_2 \leq 4 \\ 0.5x_1 + 0.3x_2 \geq 1.4 \\ x_1 \geq 0, \quad x_2 \geq 0 \end{cases}$$

a pour solution $x_1^* = 1$, $x_2^* = 3$ et pour valeur optimale $Z^* = 5$

La probabilité pour que cette solution soit admissible est :

$$P(b \mid 1 + 3 \leq 4; 0.5 \times 1 + 0.3 \times 3 \geq b(w)) = P(1.4 \geq b(w)) = \frac{1}{2}$$

— Pour l'interprétation avec seuils de probabilités sur les contraintes, choisissons $\alpha = 0.9$. Cette interprétation peut être utilisée par la firme si elle n'a pas de capacité de stockage et souhaite maintenir le nombre de clients satisfaits. Elle doit être en mesure d'assurer les livraisons à 90%. Dans ce cas, la contrainte devient :

$$P(0.5x_1 + 0.3x_2 \geq b(w)) \geq 0.9$$

Soit F la fonction de répartition de $b(w)$, alors :

$$F(0.5x_1 + 0.3x_2) \geq 0.9 \iff 0.5x_1 + 0.3x_2 \geq F^{-1}(0.9)$$

Or

$$F^{-1}(0.9) = 1.56$$

Le problème avec seuils de probabilités sur les contraintes stochastiques s'écrit :

$$\begin{cases} \min Z(x) = 2x_1 + x_2 \\ \text{sc} \\ x_1 + x_2 \leq 4 \\ 0.5x_1 + 0.3x_2 \geq 1.56 \\ x_1 \geq 0, \quad x_2 \geq 0 \end{cases}$$

La solution est : $x_1^* = 1.8$ $x_2 = 2.2$ $Z^* = 5.8$

— Considérons maintenant un problème avec recours.

Supposons que la firme ait un contrat stipulant que la demande doit être satisfaite, et qu'elle doit commander le minerai à l'avance. Si elle produit trop, elle peut écouler l'excédent chez d'autres clients à 2 unités monétaires au dessous du taux fixé. Si elle produit trop peu, elle peut acheter sur le marché le complément à 4 unités monétaires au dessus du taux fixé. Les couts supplémentaires sont :

$$2(0.5x_1 + 0.3x_2 - b(w)) \quad \text{si} \quad 0.5x_1 + 0.3x_2 - b(w) \geq 0$$

$$4(b(w) - 0.5x_1 + 0.3x_2) \quad \text{si} \quad 0.5x_1 + 0.3x_2 - b(w) \leq 0$$

Soit $Q(x_1, x_2, w)$ ces couts supplémentaires, c'est aussi la pénalité que l'on doit ajouter à la fonction économique d'origine.

Le problème avec recours revient à résoudre :

$$(P) \begin{cases} \min & Z'(x) = 2x_1 + x_2 + E(Q(x_1, x_2, w)) \\ & sc \\ & x_1 + x_2 \leq 4 \\ & x_1 \geq 0, \quad x_2 \geq 0 \end{cases}$$

Soit

$$Q(x_1, x_2) = E[Q(x_1, x_2, w)] = \frac{1}{0.4} \int_{1.2}^{0.5x_1 + 0.3x_2} 2(0.5x_1 + 0.3x_2 - t) dt$$

$$+ \frac{1}{0.4} \int_{0.5x_1 + 0.3x_2}^{1.6} 4(t - 0.5x_1 - 0.3x_2) dt$$

$$Q(x_1, x_2) = \frac{15}{2} (0.5x_1 + 0.3x_2)^2 - 22(0.5x_1 + 0.3x_2) + \frac{82}{5}$$

(P) devient un problème quadratique :

$$\begin{cases} \min & \frac{15}{2} (0.5x_1 + 0.3x_2)^2 - 22(0.5x_1 + 0.3x_2) + \frac{82}{5} \\ & sc \\ & x_1 + x_2 \leq 4 \\ & x_1 \geq 0, \quad x_2 \geq 0 \end{cases}$$

Sa solution est : $x_1^* = 0 \quad x_2^* = 4 \quad Z^* = 4.8$

2.5 Conclusion

Ce chapitre a balayé les différents modèles de l'approche *Here and Now*. Ceci est important pour la suite de notre travail car le modèle avec seuil de probabilité sur les contraintes sera utilisé dans le troisième chapitre pour modéliser un problème de décision réel.

Chapitre 3

Application de la méthode Poly-blocs à un problème stochastique

3.1 Introduction

Dans ce chapitre, nous montrerons comment appliquer la méthode *poly – blocs* à un programme stochastique avec des variables de décision entières. Le programme stochastique sera d'abord réduit à un programme non linéaire déterministe équivalent avec des fonctions objectifs et contraintes monotones. Ce dernier sera résolu par la méthode *poly – blocs* qui exploite sa structure particulière.

3.2 Position du problème

Nous considérons le problème stochastique où l'espérance mathématique d'une fonction linéaire de variables aléatoires $Y_j(x)$ est maximisée sous une contrainte probabiliste dans laquelle, avec un niveau de probabilité α donné, $\sum Y_j(x)$ est limitée supérieurement. Le problème se formule comme suit :

$$\begin{aligned} \max E \left(\sum_{j=1}^n B_j Y_j(x) \right) \\ \text{s.c.} \\ Pr(\sum_{j=1}^n Y_j(x) \leq b) \geq \alpha \\ x \in X = \{x \in Z_+^n \mid l \leq x \leq d\} \end{aligned} \tag{3.1}$$

Où $l = (l_1, l_2, \dots, l_n)$ et $d = (d_1, d_2, \dots, d_n)$ sont des vecteurs de nombres entiers avec $l_j \leq d_j$ pour $j = 1, \dots, n$. $B_j, j = 1, \dots, n$. Sont des nombres positifs.

Un tel problème de décision peut survenir dans les applications d'enseignement supérieur lorsqu'une institution souhaite maximiser l'attrait des d'étudiants pour une spécialité tout en satisfaisant une contrainte sur le nombre d'étudiants qui s'inscrivent en raison des places limitées. Les étudiants sont classés en différents types, $j = 1, \dots, n$. Les variables de décision, x_j représentent le nombre de candidats de type j admis, et les variables aléatoires $Y_j(x)$ représentent le nombre de candidats de type j qui s'inscrivent. Il est supposé que l'institution impose une limite supérieure b sur le nombre inscriptions qui doit être remplie avec un degré de probabilité α élevé. Si un étudiant admis de type j s'inscrit avec une probabilité p_j indépendante des autres étudiants, alors chaque variable aléatoire $Y_j(x)$ a une distribution binomiale avec les paramètres p_j et x_j et a donc une moyenne $\mu_j = p_j x_j$ et une variance $\sigma_j^2 = p_j (1 - p_j) x_j = v_j x_j$.

On peut aussi trouver le même problème dans d'autres applications telles que les : lignes aériennes ou l'hôtellerie où , x_j est le nombre de réservations acceptées et $Y_j(x)$ est le nombre de passagers ou d'invités qui se présentent.

Dans chacune de ces applications, le nombre d'éléments est généralement très grand. Par conséquent , les variables aléatoires binomiales sont approximées par des variables aléatoires normales. Cependant, les limites inférieures l_j des variables x_j doivent être suffisamment grandes pour permettre cette approximation, car nous approximations habituellement une variable aléatoire binomiale $B(p_j, x_j)$ par une distribution normale à partir du moment où $p_j x_j > 15$ ce qui implique que $x_j > 30$.

Sous l'hypothèse que les types d'éléments agissent indépendamment, $\sum_{j=1}^n Y_j(x)$ est approximativement normal, avec la moyenne μ et la variance σ^2 où :

$$\mu = \sum_{j=1}^n \mu_j = \sum_{j=1}^n p_j x_j \quad \text{et} \quad \sigma^2 = \sum_{j=1}^n \sigma_j^2 = \sum_{j=1}^n v_j x_j \quad (3.2)$$

La fonction objectif peut être écrite comme :

$$E \left(\sum_{j=1}^n \beta_j Y_j(x) \right) = \sum_{j=1}^n \beta_j p_j x_j = \sum_{j=1}^n c_j x_j \quad (3.3)$$

En outre, comme introduit par *Charnes et Cooper* [10], la contrainte de probabiliste dans (3.1) peut être écrite comme :

$$Pr \left(\sum_{j=1}^n Y_j(x) \leq b \right) \geq \alpha \iff \Phi \left(\frac{b-\mu}{\sigma} \right) \geq \alpha \iff -b + \mu + \sigma K_\alpha \leq 0$$

Où Φ est la fonction cumulative de la distribution normale standard et K_α est son $-\alpha$ quantile.

En utilisant cette approximation et en tenant compte des expressions (3.2) et (3.3), le programme stochastique (3.1) devient :

$$\begin{aligned} & \text{Max} \sum_{j=1}^n c_j x_j \\ & \text{su} \text{jet à (3.4)} \\ g_0(x) &= -b + \sum_{j=1}^n p_j x_j + K_\alpha \sqrt{\sum_{j=1}^n v_j x_j} \leq 0 \\ x \in X &= \{x \in \mathbb{Z}_+^n \mid l_j \leq x_j \leq d_j, j = 1, \dots, n\} \end{aligned} \quad (3.4)$$

Ici, g_0 est une fonction convexe et croissante de x_j sur $[l_j, d_j]$ pour $\alpha \geq \frac{1}{2}$. Nous supposons que $\alpha \geq \frac{1}{2}$, ce qui est justifié par le fait que nous sommes intéressés par la politique qui garantit la réalisation des contraintes avec une forte probabilité.

Notons que le problème (3.1) n'est pas forcément limité à une contrainte de stochastique . D'autres contraintes techniques $g_i \leq 0, i = 1, \dots, m$ avec des fonctions linéaires ou non

linéaires g_i peut être ajoutées à l'ensemble des contraintes sans rien changer dans l'algorithme que nous avons présentés dans le *chapitre 1*. En résumé, nous résolvons le programme monotone entier suivant :

3.3 Application

Prenons l'exemple suivant avec une structure semblable au problème (3.1) avec :

$$\alpha = 0.998 \quad \beta_1 = 400; \quad \beta_2 = 200; \quad b = 125; \quad p_1 = 0.80 \quad p_2 = 0.90;$$

$$l_1 = 55; \quad l_2 = 70; \quad d_1 = 60; \quad d_2 = 76$$

Le pas de la terminaison dans la méthode de bisection est mis à $\varepsilon = 0.0002$. Les deux dernières contraintes sont ajoutées pour montrer que la méthode *Poly – blocs* applique aussi pour les contraintes multiples.

$$\left\{ \begin{array}{l} \max f(x) = 320x_1 + 180x_2 \\ \text{su jet à} \\ g_0(x_1, x_2) = 0,80x_1 + 0,90x_2 + 2,9\sqrt{0,16x_1 + 0,09x_2} - 125 \leq 0 \\ g_1(x_1, x_2) = x_1x_2 + 2x_1 - x_2 - 4380 \leq 0 \\ g_2(x_1, x_2) = x_1 + 5x_2 - 435 \leq 0 \\ x \in X = \{x \in \mathbb{Z}_+^2 \mid 55 \leq x_1 \leq 60, 70 \leq x_2 \leq 76\} \end{array} \right.$$

Initialisation

On pose $l = (55, 70)$, $d = (60, 76)$

$l = (55, 70)$ est réalisable, $d = (60, 76)$ est irréalisable

$x_{opt} = (55, 70)$, $f_{opt} = 30200$

$X^{(1)} = [l, d] = [(55, 70), (60, 76)]$; $k = 1$.

Itération 1 :

Etape1 : Sélectionnons $[\alpha, \beta] = [l, d] = [(55, 70), (60, 76)]$

Dans ce cas $X^{(1)} \setminus [\alpha, \beta] = \emptyset$

Nous utilisons la procédure de bisection pour trouver la racine de l'équation

$$G[\lambda(55, 70) + (1 - \lambda)(60, 76)] = 0$$

Cette méthode nous donne $\lambda = 0,31960$. Le point d'intersection est donné par :

$$x_b^{(1)} = \lambda^*(55, 70) + (1 - \lambda^*)(60, 76) = (58,4020, 74,0824)$$

$$x^{F1} = \lfloor x_b^{(1)} \rfloor = (58, 74) \text{ et } x^{l1} = \lceil x_b^{(1)} \rceil = (59, 75)$$

$$f(x^{F1}) = 31880 > 30200 = f_{opt}$$

On pose $x_{opt} = (58, 74)$ et $f_{opt} = 31880$

La croissance de f implique qu'il n'y a pas de points réalisables dans

$[x^{l1}, \beta] = [(59, 75), (60, 76)]$ par conséquent, nous pouvons retirer ce box du box $[\alpha, \beta] = [(55, 70), (60, 76)]$

Etape2 : maintenant, nous appliquons la formule (1.4) pour partitionner l'ensemble

$$\begin{aligned}\Omega_1^{(1)} &= [\alpha, \beta] \setminus [x^{l1}, \beta] \text{ en deux boxes entiers.} \\ \Omega_1^{(1)} &= [(55, 70), (60, 76)] \setminus [(59, 75), (60, 76)] \\ &= \{[\alpha_1, \gamma_1 - 1] \times [\alpha_2, \beta_2]\} \cup \{[\gamma_1, \beta_1] \times [\alpha_2, \gamma_2 - 1]\} \\ &= \{[55, 58] \times [70, 76]\} \cup \{[59, 60] \times [70, 74]\} \\ &= [(55, 70), (58, 76)] \cup [(59, 70), (60, 74)] \\ x^{F1} &\in [\hat{\alpha}, \hat{\beta}] = [(55, 70), (58, 76)]. \\ \Omega_1^{(1)} &= \Omega_1^{(1)} \setminus [\hat{\alpha}, \hat{\beta}] = [(59, 70), (60, 74)]\end{aligned}$$

f étant une fonction croissante, il n'y a pas de points réalisables meilleurs que x^{F1} dans $[\hat{\alpha}, x^{F1}] = [(55, 70), (58, 74)]$. Nous pouvons donc retirer le box $[\hat{\alpha}, x^{F1}]$ du box $[\hat{\alpha}, \hat{\beta}]$ en appliquant la formule (1.3)

$$\begin{aligned}\Omega_2^{(1)} &= [\hat{\alpha}, \hat{\beta}] \setminus [\hat{\alpha}, x^{F1}] \\ &= [(55, 70), (58, 76)] \setminus [(55, 70), x^{F1}] \\ &= [(55, 70), (58, 76)] \setminus [(55, 70), (58, 74)] \\ &= \{[\gamma_1 + 1, \beta_1] \times [\alpha_2, \beta_2]\} \cup \{[\alpha_1, \gamma_1] \times [\gamma_2 + 1, \beta_2]\} \\ &= \{[59, 58] [70, 76]\} \cup \{[55, 58] [75, 76]\} \\ &= \emptyset \cup \{[55, 58] \times [75, 76]\} \\ &= [(55, 75), (58, 76)]\end{aligned}$$

$$Y^{(1)} = \Omega_1^{(1)} \cup \Omega_2^{(1)} = [(59, 70), (60, 74)] \cup [(55, 75), (58, 76)]$$

Aucun des boxes entiers dans $Y^{(1)}$ n'est rejeté, alors $Z^{(1)} = Y^{(1)}$

Etape3 : On pose $X^{(2)} = X^{(1)} \cup Z^{(1)} = [(59, 70), (60, 74)] \cup [(55, 75), (58, 76)]$

Itération 2 :

Etape1 : Nous utilisons la stratégie qui sélectionne le box entier dans $X^{(1)}$ avec la valeur de la fonction objectif maximale du point de vue limite supérieure

$$f(60, 74) = 32520 > 32240 = f(58, 76) \text{ on pose } [\alpha, \beta] = [(59, 70), (60, 74)]$$

$$X^{(2)} = X^{(2)} \setminus [\alpha, \beta] = [(55, 75), (58, 76)]$$

La procédure de bisection appliquée à l'équation

$$G[\lambda(59, 70) + (1 - \lambda)(60, 74)] = 0$$

$$\text{Nous donne } \lambda^* = 0,34123 \text{ et } x_b^{(2)} = \lambda^*(59, 70) + (1 - \lambda^*)(60, 74) = (59.65877, 72.63508)$$

$$x^{F2} = \lfloor x_b^{(2)} \rfloor = (59, 72) \text{ et } x^{l2} = \lceil x_b^{(2)} \rceil = (60, 73)$$

$$f(x^{F2}) = 31840 < 31880 \text{ on pose } x_{opt} = (58, 74) \text{ et } f_{opt} = 31880$$

Etape2 : La formule (1.4) nous donne

$$\begin{aligned}\Omega_1^{(2)} &= [\alpha, \beta] \setminus [x^{l2}, \beta] \\ &= [(59, 70), (60, 74)] \setminus [(60, 73), (60, 74)] \\ &= \{[\alpha_1, \gamma_1 - 1] \times [\alpha_2, \beta_2]\} \cup \{[\gamma_1, \beta_1] \times [\alpha_2, \gamma_2 - 1]\} \\ &= \{[59, 59] \times [70, 74]\} \cup \{[60, 60] \times [70, 72]\} \\ &= [(59, 70), (59, 74)] \cup [(60, 70), (60, 72)] \\ x^{F2} &= (59, 72) \in [(59, 70), (59, 74)]. \text{ On pose } \Omega_1^{(2)} = [(60, 70), (60, 72)]\end{aligned}$$

$$\begin{aligned}
\Omega_2^{(2)} &= [(59, 70), (59, 74)] \setminus [(59, 70), x^{F2}] \\
&= [(59, 70), (59, 74)] \setminus [(59, 70), (59, 72)] \\
&= \{[\gamma_1 + 1, \beta_1] \times [\alpha_2, \beta_2]\} \cup \{[\alpha_1, \gamma_1] \times [\gamma_2 + 1, \beta_2]\} \\
&= \{[60, 59] \times [70, 74]\} \cup \{[59, 59] \times [73, 74]\} \\
&= \emptyset \cup [59, 59] \times [73, 74] \\
&= [(59, 73), (59, 74)]
\end{aligned}$$

$$Y^{(2)} = \Omega_1^{(2)} \cup \Omega_2^{(2)} = [(60, 70), (60, 72)] \cup [(59, 73), (59, 74)]$$

Le point $(60, 72)$ est réalisable et $f(60, 72) = 32160 > 31880$

On pose $x_{opt} = (60, 72)$ et $f_{opt} = 32160$ on retire le box $[(60, 70), (60, 72)]$ de $Y^{(2)}$
 $f(59, 74) = 32200 > 32160 = f_{opt}$, le box $[(59, 73), (59, 74)]$ doit être scanné

Par conséquent, nous ne le rejetons pas de $Y^{(2)}$

On a $Z^{(2)} = [(59, 73), (59, 74)]$

Etape3 : Pour le box $[(55, 75), (58, 76)] \in X^{(2)}$,

nous avons $f(58, 76) = 32240 > 32160$, donc ce box doit être exploré

Donc $X^{(3)} = X^{(2)} \cup Z^{(2)} = [(55, 75), (58, 76)] \cup [(59, 73), (59, 74)]$

Itération 3 :

Etape1 : $f(58, 76) > f(59, 74)$, sélectionnons $[\alpha, \beta] = [(55, 75), (58, 76)]$ et posons

$$X^{(3)} = X^{(3)} \setminus [\alpha, \beta] = [(59, 73), (59, 74)]$$

La procédure de bisection appliquée à l'équation

$$G[\lambda(55, 75) + (1 - \lambda)(58, 76)] = 0 \text{ devenir } \lambda^* = 0,41162$$

$$\text{Et } x_b^{(3)} = \lambda^*(55, 75) + (1 - \lambda^*)(58, 76) = (56,76514.75, 58838)$$

$$x^{F3} = \lfloor x_b^{(3)} \rfloor = (56,75) \text{ et } x^{l3} = \lceil x_b^{(3)} \rceil = (57,76)$$

$$f(x^{F3}) = 31420 < 32160 = f_{opt}, \text{ par conséquent, } x_{opt} = (60, 72) \text{ et } f_{opt} = 32160$$

$$\begin{aligned}
\Omega_1^{(3)} &= [\alpha, \beta] \setminus [x^{l3}, \beta] \\
&= [(55, 75), (58, 76)] \setminus [(57, 76), (58, 76)] \\
&= \{[\alpha_1, \gamma_1 - 1] \times [\alpha_2, \beta_2]\} \cup \{[\gamma_1, \beta_1] \times [\alpha_2, \gamma_2 - 1]\} \\
&= \{[55, 56] \times [75, 76]\} \cup \{[57, 58] \times [75, 75]\} \\
&= \{(55, 75), (56, 76)\} \cup \{(57, 75), (58, 75)\}
\end{aligned}$$

$$x^{F3} \in [(55, 75), (56, 76)], \text{ posons } \Omega_1^{(3)} = [(57, 75), (58, 75)]$$

$$\begin{aligned}
\Omega_2^{(3)} &= [(55, 75), (56, 76)] \setminus [(55, 75), x^{F3}] \\
&= [(55, 75), (56, 76)] \setminus [(55, 75), (56, 75)] \\
&= \{[\gamma_1 + 1, \beta_1] \times [\alpha_2, \beta_2]\} \cup \{[\alpha_1, \gamma_1] \times [\gamma_2 + 1, \beta_2]\} \\
&= \{[57, 56] \times [75, 76]\} \cup \{[55, 56] \times [76, 76]\} \\
&= [(55, 76), (56, 76)]
\end{aligned}$$

$$Y^{(3)} = \Omega_1^{(3)} \cup \Omega_2^{(3)} = \{(57, 75), (58, 75)\} \cup \{(55, 76), (56, 76)\}$$

$$f(58, 75) = 32060 < 32160 = f_{opt} \text{ et } f(56, 76) = 31600 < 32160 = f_{opt}$$

On va supprimer les deux boxes $[(57, 75), (58, 75)]$ et $[(55, 76), (56, 76)]$ de $Y^{(3)}$

Par conséquent, $Z^{(3)} = \emptyset$

Etape3 : Pour le box $[(59, 73), (59, 74)] \in X^{(3)}$, nous avons

$f(59, 74) = 32200 > 32160 = f_{opt}$, ce box doit être exploré

alors, $X^{(4)} = X^{(3)} \cup Z^{(3)} = [(59, 73), (59, 74)]$

Itération 4 :

Etape1 : Sélectionnons $[\alpha, \beta] = [(59, 73), (59, 74)]$ et posons $X^{(4)} = X^{(4)} \setminus [\alpha, \beta] = \emptyset$

La procédure de bisection appliquée à l'équation

$G[\lambda(59, 73) + (1 - \lambda)(59, 74)] = 0$ nous donne $\lambda^* = 0,51724$ et

$$x_b^{(4)} = \lambda^*(59, 73) + (1 - \lambda^*)(59, 74) = (59, 73.48276)$$

$$x^{F4} = \lfloor x_b^{(4)} \rfloor = (59, 73) \text{ et } x^{L4} = \lceil x_b^{(4)} \rceil = (59, 74)$$

$$f(x^{F4}) = 32020 < 32160 = f_{opt}$$

Alors le point $(60, 72)$ est toujours la solution optimale

$$\Omega_1^{(4)} = [\alpha, \beta] \setminus [x^{L4}, \beta]$$

$$= [(59, 73), (59, 74)] \setminus [(59, 74), (59, 74)]$$

$$= \{[\alpha_1, \gamma_1 - 1] \times [\alpha_2, \beta_2]\} \cup \{[\gamma_1, \beta_2] \times [\alpha_2, \gamma_2 - 1]\}$$

$$= \{[59, 58] \times [73, 74]\} \cup \{[59, 59] \times [73, 73]\}$$

$$= \emptyset \cup [(59, 73), (59, 73)]$$

$$= [(59, 73), (59, 73)]$$

$$x^{F4} \in [(59, 73), (59, 73)], \text{ posons } \Omega_1^{(4)} = \emptyset$$

$$\Omega_2^{(4)} = [(59, 73), (59, 73)] \setminus [(59, 73), x^{F4}]$$

$$= [(59, 73), (59, 73)] \setminus [(59, 73), (59, 73)] = \emptyset$$

$$Y^{(4)} = \Omega_1^{(4)} \cup \Omega_2^{(4)} = \emptyset$$

$$\text{Par conséquent, } Z^{(4)} = \emptyset \text{ et } X^{(5)} = X_1^{(4)} \cup X_2^{(4)} = \emptyset$$

La solution optimale du problème est donc $x_{opt} = (60, 72)$, avec la valeur de la fonction objectif $f_{opt} = 32160$

3.3.1 Matlab

3.3.1.1 Introduction

Matlab est un puissant outil de calcul numérique, de programmation et de visualisation graphique. Son nom signifie *Matrix Laboratory*, c'est à dire un environnement interactif de travail avec des matrices. La facilité de développement des applications dans son langage fait qu'il est pratiquement devenu le standard dans son domaine. Actuellement, on trouve de nombreuses boites à outils (*Toolboxes*) qui contiennent des fonctions spécialisées permettant d'utiliser l'environnement Matlab pour résoudre des classes spécifiques de problèmes.

Avec ses fonctions spécialisées, *MATLAB* peut être aussi considéré comme un langage de programmation adapté pour les problèmes scientifiques. *MATLAB* est un interpréteur : les instructions sont interprétées et exécutées ligne par ligne. *MATLAB* fonctionne dans plusieurs environnements tels que X-Windows, Windows, Macintosh. Il existe deux modes de fonctionnement :

Mode interactif : *MATLAB* exécute les instructions au fur et à mesure qu'elles sont données par l'utilisateur.

Mode exécutif : *MATLAB* exécute ligne par ligne un "fichier M" (programme en langage *MATLAB*).

Exécution du problème (3.1) sur *Matlab*

```

MATLAB 7.8.0 (R2009a)
File Edit Debug Parallel Desktop Window Help
Current Directory: C:\Users\USER\Downloads
Shortcuts How to Add What's New
Command Window

g10 =
    -6.7310 -490.0000 -30.0000

g20 =
     3.1584 224.0000     5.0000

x0 =
     55     70

x0 =
     55     70

x0 =
     55     70

ub =
     60     76

Warning: Trust-region-reflective method does not currently solve this type of

```

```

MATLAB 7.8.0 (R2009a)
File Edit Debug Parallel Desktop Window Help
Current Directory: C:\Users\USER\Downloads
Shortcuts How to Add What's New
Command Window

     60     76

Warning: Trust-region-reflective method does not currently solve this type of
problem,
using active-set (line search) instead.
> In fmincon at 439
   In programe_final at 38

Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the default value of the function tolerance,
and constraints were satisfied to within the default value of the constraint tol

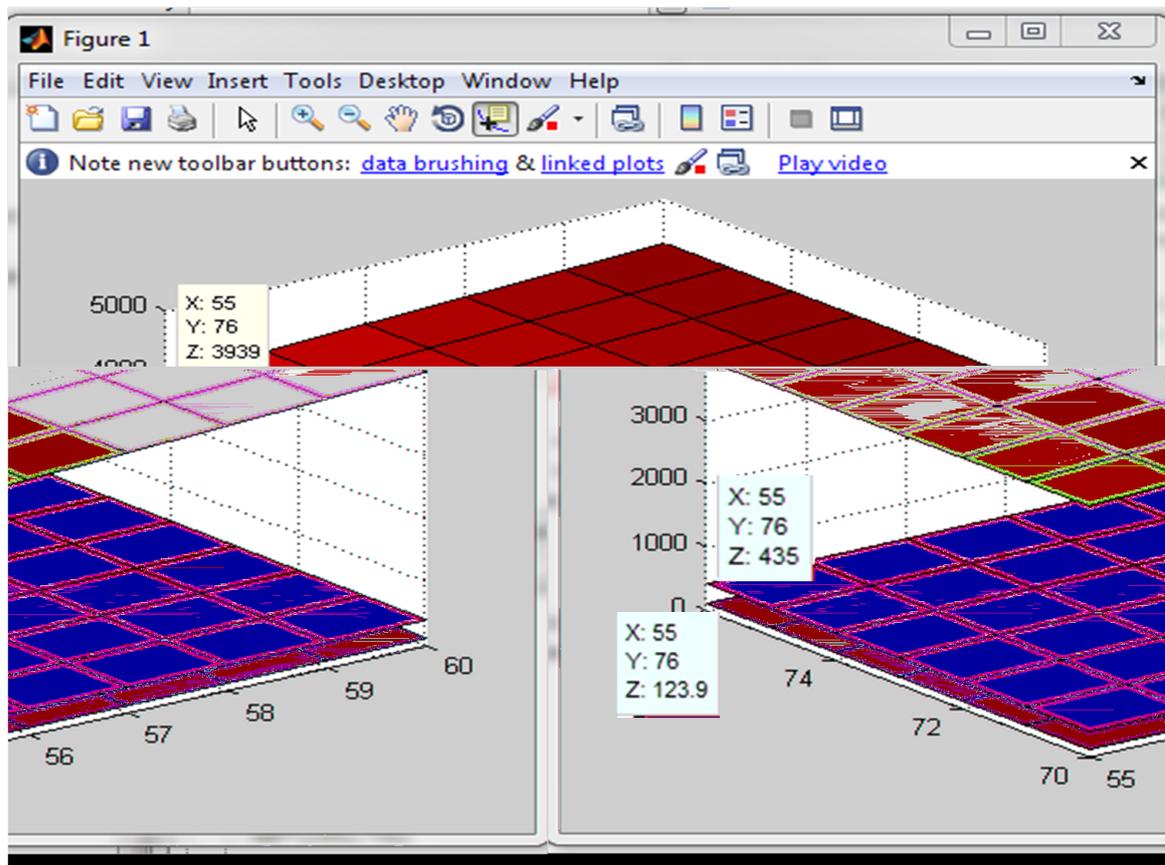
<stopping criteria details>

Active inequalities (to within options.TolCon = 1e-006):
   lower      upper   ineqlin   ineqnonlin
         1             2

xopt =
     60     72

Fopt =
    3.2197e+004
fx \

```



3.3.2 Lingo

3.3.2.1 Introduction

Lingo est un logiciel utilisé pour résoudre les modèles linéaires, entiers et quadratiques, ainsi que les modèles non linéaires d'optimisation globale.

Programmation du problème Polyblocs sur Lingo

Exécution du problème (3.1) sur *Lingo*

```

! la solution optimale de la fonction objectif ;
max =320*x1+180*x2;
0.8*x1+0.9*x2+2.9*(0.16*x1+0.09*x2)^(0.5)-125<=0;
x1*x2+2*x1-x2-4380<=0;
x1+5*x2-435<=0;
!les valeur optimale du problème ;
x1<=60;
x1>=55;

```

Solution Report - sfih

Global optimal solution found.
Objective value: 32160.00
Objective bound: 32160.00
Infeasibilities: 0.000000
Extended solver steps: 0
Total solver iterations: 0

Model Class: PINLP

Total variables: 2
Nonlinear variables: 2
Integer variables: 2

Total constraints: 8
Nonlinear constraints: 2

Total nonzeros: 12
Nonlinear nonzeros: 4

Variable	Value	Reduced Cost
X1	60.00000	-320.0000
X2	72.00000	-180.0000

Lingo 13.0 Solver Status [sfih]

Solver Status: Model Class: PINLP, State: Global Opt, Objective: 32160, Infeasibility: 0, Iterations: 0

Extended Solver Status: Solver Type: Global, Best Obj: 32160, Obj Bound: 32160, Steps: 0, Active: 0

Variables: Total: 2, Nonlinear: 2, Integers: 2

Constraints: Total: 8, Nonlinear: 2

Nonzeros: Total: 12, Nonlinear: 4

Generator Memory Used (K): 19

Elapsed Runtime (hh:mm:ss): 00:00:02

Update Interval: 2 | Interrupt Solver | Close

La fenêtre de commande Lingo

Exemple 2

Programmation du 2ème problème Polyblocs susLingo

Exécution du 2eme problème sur *Lingo*

```

max = 3*x1*x2 -x1+ 6*x2;
5*x1*x2-4*x1-4.5*x2 -32 <=0;
x1<=5;
x1>=1;
x2<=5;
x2>=1;
@gin (x1) ;
@gin (x2) ;
end

```

Solution Report - 11111

Global optimal solution found.
Objective value: 58.00000
Objective bound: 58.00000
Infeasibilities: 0.000000
Extended solver steps: 0
Total solver iterations: 0

Model Class: PINLP

Total variables: 2
Nonlinear variables: 2
Integer variables: 2

Total constraints: 6
Nonlinear constraints: 2

Total nonzeros: 8
Nonlinear nonzeros: 4

Generator Memory Used (K): 18

Elapsed Runtime (hh:mm:ss): 00:00:01

Variable	Value	Reduced Cost
X1	2.00000	0.00000
X2	5.00000	0.00000

Lingo 13.0 Solver Status [11111]

Solver Status: Model Class: PINLP, State: Global Opt, Objective: 58, Infeasibility: 0, Iterations: 0

Extended Solver Status: Solver Type: Global, Best Obj: 58, Obj Bound: 58, Steps: 0, Active: 0

Variables: Total: 2, Nonlinear: 2, Integers: 2

Constraints: Total: 6, Nonlinear: 2

Nonzeros: Total: 8, Nonlinear: 4

Generator Memory Used (K): 18

Elapsed Runtime (hh:mm:ss): 00:00:01

Update Interval: 2 | Interrupt Solver | Close

Conclusion

Dans ce mémoire, nous avons considéré des programmes linéaires où les données sont imprécises et les variables de décision sont des nombres entiers. Nous nous sommes intéressés particulièrement à un *E – modèle* avec seuil de probabilité sur les contraintes qui peut résulter d'un problème réel, à savoir, la gestion des inscriptions dans un établissement et des réservations de places dans les compagnies aériennes ou dans les hôtels.

Pour aborder notre problème, nous avons exploité les techniques de la recherche opérationnelle qui ont déjà prouvé leur efficacité comme, la programmation linéaire stochastique, l'optimisation monotone et l'optimisation discrète.

Nous avons d'abord montré comment transformer le modèle stochastique discret en problème non linéaire monotone discret. Ce dernier est ensuite résolu par une méthode exacte dite «*méthode poly – blocs*». Cette méthode dont le principe est semblable à celui de la méthode *Branch and Bound* est basée sur la méthode de *bissection*.

Cependant, même si le modèle que nous avons considéré n'a pas pu être testé en situation réelle, les tests que nous avons effectués sont très prometteurs. De plus, l'algorithme présenté peut être étendu sans difficultés au *modèle Kataoka*.

En perspectives, il serait intéressant d'étendre la méthode *poly – blocs* aux modèles multi-objectifs stochastiques en nombres entiers. D'autres part, les méthodes exactes ne sont pas utilisables pour les problèmes de grande taille. Ainsi, Il paraît judicieux de faire appel aux méthodes heuristiques pour la résolution du problème étudié dans ce manuscrit.

Annexe 1

Rappel sur les variables aléatoires[9]

Une variable aléatoire (*v.a*) est une application de l'ensemble des épreuves dans le corps des réels. Elle est caractérisée par l'ensemble des probabilités associées à tous ses états possibles.

Définition 1.1

Soient (Ω, A) et (E, F) deux espaces probabilisables et X une application

$$\begin{aligned} X : \Omega &\rightarrow E \\ \omega &\rightarrow x(\omega) \end{aligned}$$

X est dite (*v.a*) si :

$$\begin{aligned} \forall B \in F, X^{-1}(B) &\in A \\ X^{-1}(B) &= \{\omega \in \Omega / X(\omega) \in B\} \end{aligned}$$

Ce qui signifie l'image réciproque de toute événement par X est aussi un événement, en d'autre terme une variable est dite aléatoire si l'ensemble de ses valeurs dépend du hasard

Propriétés

Si X, Y sont deux v.a alors :

- $aX + bY$ est aussi une v.a $\forall a, b \in \mathbb{R}$
- $XY; \frac{X}{Y}$ ($Y \neq 0$) sont aussi des v.a

$\forall f$ continue, $f(x)$ est aussi une v.a

— Fonction de densité

Soit[6] f une application de \mathbb{R} dans \mathbb{R}
 f est dite fonction de densité si :

1. $f(x) \geq 0 \forall x \in \mathbb{R}$
2. f est intégrable
3. $\int_{\mathbb{R}} f(x) d(x) = 1$

— Fonction de répartition

Soit X une v.a quelconque, on appelle fonction de répartition de x , la fonction notée F définie par

$$\begin{aligned} F : \mathbb{R} &\rightarrow [0,1] \\ x &\rightarrow F(x) = P(X \leq x) \end{aligned}$$

P est une probabilité.

A. Variable discrète :

une v.a est dite discrète si l'ensemble des valeurs qu'elle prend est soit fini soit infini dénombrable

— Loi uniforme discrète

$$\begin{aligned} X(\Omega) &= \{1, 2, \dots, n\} \\ P(X = x) &= \frac{1}{n} \quad x \in [1, n] \\ E(X) &= \frac{n+1}{2} \quad V(X) = \frac{n^2-1}{12} \end{aligned}$$

— Loi de Bernoulli

On dit qu'une v.a X à valeur dans $\{0, 1\}$ suit une loi de Bernoulli de paramètre $p \in]0, 1[$ notée $B(p)$:

$$\begin{aligned} f(x, p) &= p^x (1-p)^{1-x} \quad x = 0, 1 \\ E(X) &= p \quad V(X) = p(1-p) \end{aligned}$$

— Loi de Binomiale

On dit qu'une v.a X à valeurs dans $\{0, 1, \dots, n\}$ suit une loi Binomiale de paramètres (n, p) notée $B(n, p)$

$$\begin{aligned} P(X = x) &= C_n^x p^x (1-p)^{n-x} \text{ pour } x \in [0, n] \text{ et } C_n^x = \frac{n!}{x!(n-x)!} \\ E(X) &= np \quad V(X) = np(1-p) \end{aligned}$$

— Loi de poisson

On dit qu'une v.a X à valeur dans \mathbb{N} suit une loi de poisson de paramètre $\lambda > 0$ notée :

$$\begin{aligned} P(X = x) &= e^{-\lambda} \frac{\lambda^x}{x!} \quad X \in \{0, 1, \dots, n\} \\ E(X) &= \lambda \quad V(X) = \lambda \end{aligned}$$

Tableau des lois discrètes

la loi	Probabilité	Moy	Var	L'écart type
Bernoulli	$P(X = 0) = 1 - p$ et $P(X = 1) = p$	p	$p(1-p)$	$\sqrt{p(1-p)}$
Binomiale	$P(X = x) = C_n^x p^x (1-p)^{n-x}$ pour $x \in [0, n]$	np	$np(1-p)$	$\sqrt{np(1-p)}$
Uniforme	$P(X = x) = \frac{1}{n}$, $x \in [1, n]$	$\frac{n+1}{2}$	$\frac{n^2-1}{12}$	$\sqrt{\frac{n^2-1}{12}}$
Poisson	$P(X = x) = e^{-\lambda} \frac{\lambda^x}{x!}$ pour $\lambda > 0$ et $x = 0, 1, \dots, n$	λ	λ	$\sqrt{\lambda}$

B. Variables continues

Soit X une v.a continue. Alors la loi de X est caractérisée par l'ensemble des probabilités

$$P(a < X < b) = \int_a^b f_X(x) dx$$

Où f_X est la densité de probabilité de (X) et a et b sont deux nombres réels, éventuellement infinis. Comme nous l'avons vu plus haut, il suffit de connaître cette densité pour connaître la loi de X .

— **Loi uniforme**

La loi uniforme sur un intervalle est la loi des "tirages au hasard" dans cet intervalle. Si $a < b$ sont deux réels, la loi uniforme sur l'intervalle $[a, b]$ est notée $U(a, b)$. Elle a pour densité :

$$f(x) = \begin{cases} \frac{1}{b-a} & x \in [a, b] \\ 0 & \text{sinon} \end{cases}$$

$$E(X) = \frac{a+b}{2} \quad \text{Var}(X) = \frac{(b-a)^2}{12}$$

— **Loi exponentielle**

On dit que X suit une loi exponentielle de paramètre $\lambda > 0$ notée $\varepsilon(\lambda)$, si la loi de X a pour densité

$$f_X(x) = \begin{cases} \lambda \exp(-\lambda x) & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases}$$

$$E(x) = \frac{1}{\lambda} \quad \text{Var}(X) = \frac{1}{\lambda^2}$$

— **Loi normal**

Soient $\mu \in \mathbb{R}$ et $\sigma > 0$. On dit que X suit une loi normale de paramètres (μ, σ^2) , notée $N(\mu, \sigma^2)$, si la loi de X a pour densité

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}, \quad x \in \mathbb{R}$$

$$E(X) = \mu \quad \text{Var}(X) = \sigma^2$$

Tableau des lois continues

Loi	Probabilité	Moyenne	Variance	L'écart type
Uniforme	$f(x) = \frac{1}{b-a}$ avec $a \leq x \leq b$	$\frac{a+b}{2}$	$\frac{(a+b)^2}{12}$	$\frac{a+b}{\sqrt{12}}$
Normale	$P(X) = 0 = 1 - p$ et $P(X = 1) = 1$	p	$p(1-p)$	$\sqrt{p(1-p)}$
Exponentielle	$f(x) = \frac{1}{a} e^{-\frac{x}{a}}$ pour $x > 0$ et $a > 0$	a	a^2	a

a. Espérance

Définition

— Si X est une v.a discrète, qui prend les valeurs $\{X_1, \dots, X_K\}$ l'espérance de X est définie par :

$$E(X) = \sum_{i=1}^k x_i p(X = x_i)$$

— Si X est une v.a absolument continue de densité f , l'espérance de X est définie par :

$$E(X) = \int_{\mathbb{R}} x f(x)$$

— Linéarité :

$$E(X + Y) = E(X) + E(Y)$$

— Cas de deux v.a X et Y indépendantes :

$$E(XY) = E(X) E(Y)$$

b. Variance

Définition 1.2 Pour calculer la variance d'une série statistique ou d'une variable aléatoire, on calcule les écarts entre la série (ou la variable) et sa moyenne (ou espérance) puis on prend la moyenne (ou l'espérance) de ces écarts élevés au carré. Synthétiquement, on écrit :

$$V(X) = E[(X - E(X))^2]$$

c. L'écart type

$$\sigma(X) = \sqrt{V(X)}$$

d. La covariance

La covariance de deux variables aléatoires réelles X et Y ayant chacune une variance (finie¹), notée $cov(X, Y)$ ou parfois σ_{XY} , est la valeur :

$$cov(X, Y) = E[(X - E[X])(Y - E[Y])]$$

Annexe 2

Méthode de Dichotomie ou Bissection

Dans ce paragraphes[11] nous rappelons la méthode de dichotomie qui permet d'approcher le zéro de f et il basée sur le théorème suivant :

Théorème

Soit $[a, b]$ un intervalle fermé de \mathbb{R} et $f : [a, b] \rightarrow \mathbb{R}$ une fonction continue

Si $f(a).f(b) < 0$ alors $\exists \alpha \in]a, b[$ tel que $f(\alpha) = 0$

On se donne un intervalle $I_0 = [a, b]$ contenant le zéro α que l'on veut approcher. La méthode de dichotomie produit une suite de sous-intervalles $I_n = [a_n, b_n]$, $n \geq 0$ avec $I_{n+1} \subset I_n$ et $f(a_n).f(b_n) < 0$. En particulier, on prend $a_0 = a, b_0 = b$ et $x_0 = \frac{a_0 + b_0}{2}$ et pour $n \geq 0$:

On pose $a_{n+1} = a_n$, et $b_{n+1} = x_n$ Si $f(a_n).f(x_n) < 0$

Ou $a_{n+1} = x_n$, $b_{n+1} = b_n$ Si $f(x_n).f(b_n) < 0$

et $x_{n+1} = \frac{a_{n+1} + b_{n+1}}{2}$

Fonction convexe

Définition

Soit f une fonction définie sur un convexe $K \in \mathbb{R}^n$, f est dite convexe si :

$\forall x_1, x_2 \in K, \forall \lambda \in [0, 1]$:

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2)$$

Bibliographie

- [1] Aïche F., Sur la programmation linéaire multi-objectifs floue stochastique, Thèse de Doctorat, Université de Tizi Ouzou (2013)
- [2] Arnal J., Migallen V. and Penadés J., Programmation linéaire appliquée. La programmation en nombres entiers, (1996).
- [3] Baillargeon G., Programmation linéaire appliquée. Les éditions SMG, Québec,(1996).
- [4] Bazaraa M.,Sherali H. and Shetty C., Theory and algorithms, Second edition, Wily, New York, (1993).
- [5] Bellahcene F., sur la programmation lineaire stochastique multiobjectifs en nombres entiers,Thèse de Doctorat, Université de Tizi Ouzou (2006)
- [6] Bellahcene F., Applied polyblock method for a special integer chance constrained problem, Actes du colloque international sur l'optimisation et les système d'inforamation COSI'13, Alger, (2013).
- [7] Bereanu B., On stochastic linear programming I : Distribution problems : A single random variable, Revue de Mathématiques Pures et Appliquées, 8, 683-697, (1963).
- [8] Bereanu B., Programme de risque minimal en programmation linéaire stochastique, C.R. Acad. Sci. Paris, 259, 1383-1386, (1964).
- [9] Berkoune, cours de statistique inférantielle,(*variable aléatoire*), Université de Tizi Ouzou ,(2014)
- [10] Charnes A. and Cooper W.W., Chance constrained programming, Management Science, 6, 73-79, (1959).
- [11] Gérald Baillargeon, Programmation linéaire appliquée. Les éditions SMG, Québec,(1996).
- [12] Gomory R. E., An algorithm for integer solutions to linear programs, In : Graves R. L., Wolfe P. (Eds), Recent Advances in Mathematical Programming, McGraw-Hill, New York, 263-302, (1963).
- [13] Kall P., Stochastic linear programming, Econometrics and Operations Research XXI, Springer Verlag , (1976).
- [14] Kataoka S., A stochastic programming model, Econometrics, 31, 1-2, 181-196, (1963).
- [15] Prekopa A., On the probability distribution of the optimum of a random linear program, Siam Journal Control, 1, 211-222, (1966).
- [16] Rubinov A., Tuy H. and Mays H. An algorithm for monotonic global optimization problems, Optimization 49, 205-221, (2006).
- [17] Sun X.L. and Li J.L., Nonlinear integer programming. Springer 1-451, (2006).
- [18] Tawarmalani M. and Sahinidis N.V., Global optimization of mixed integer nonlinear programs : A theoretical and computational study. Mathematical Programming 99, 563-591, (2004).

- [19] Teghem J., Recherche Opérationnelle : méthodes d'optimisation, Tome 1,2 ,Édition ellipses Marketing S.A, (2012)
- [20] Teghem J., Programmation linéaire, Edition ellipses SMA , Bruxelles ,(1996).
- [21] Tinter G., A note on stochastic linear programming, *Econometrica*, 28, 490- 495, (1960).
- [22] Tuy H., Monotonic optimization : Problems and solution approaches. *SIAM Journal of Optimization* 11 (2) ,464-494, (2000) .
- [23] Tuy H., Thach P.T. and H. Konno, Optimization of polynomial fractional functions. *Journal of Global Optimization* 29 ,19-44,(2004) .
- [24] Wets R., Stochastic programming : Solution techniques and approximation schemes, in Berchem, Grötschel, Korte (Eds), *Mathematical Programming*, Springer Verlag, 566-603, (1983).