Package inputenc Error: Keyboard character used is undefinedin inputencoding 'latin1'See the inputenc package documentation for explanation.You need to provide a definition with or before using this key.3.236

République Algérienne Démocratique et populaire
Ministère de l'enseignement Supérieur et de la Recherche Scientifique
Université M'hamed Bougara de BOUMERDES
*F*aculté des Sciences
*D*épartement de Mathématiques

Mémoire présenté le 16/09/2018,
Par :

M$^{\text{elle}}$ GUENOUN Nesrine
M$^{\text{elle}}$ BAKHALI Fatma Zohra

Pour l'obtention du diplôme de
Master en Recherche Opérationnelle
Option : Recherche Opérationnelle, Optimisation et Management Stratégique
Sous le thème :

Optimisation multi-objectif stochastique en nombre entier

Soutenu publiquement devant le jury composé de :

Présidente	**M$^{\text{me}}$**	**S.OUATIKI**	**M.C.B**	**U.M.B.B.**
Promotrice	**M$^{\text{me}}$**	**W. DRICI**	**M.A.B**	**U.M.B.B.**
Examinateur	**M$^{\text{me}}$**	**F.CHEURFA**	**M.A.A**	**U.M.B.B.**

Année Universitaire 2017/2018

Table des matières

Table des figures

Remerciements

Nous tenous tout d'abord à remercier ALLAH le Tout Puisant et miséricordieux, qui nous a donné le courage, la force et la patience pour la réalisation de ce mémoire.

Nous tenons à exprimer notre profonde gratitude à Madame **W.Drici**, notre promotrice de mémoire, pour la confiance qu'elle nous a faite en acceptant de diriger nos traveaux, et pour ses précieux conseils et orientations, ainsi que pour l'intérêt particulier qu'elle a accordé à ce travail. Nous la remercions pour sa grande contribution   l'aboutissement de ce travail.

Nous adressons nos remerciements à madame **S.OUATIKI** pour l'honneur qu'elle nous a fait en acceptant de présider le jury de ce mémoire.

Nous remercions également monsieur **F.CHERFA** pour avoir accepté de juger ce travail.

Nous remercions nos trés chers parents, qui ont toujours été là pour nous.

Nous remercions nos frères et soeurs pour leur encouragements, ainsi que toute la famille.

Sans oublier de remercier aussi tous nos collègues, nos amies et tous ceux qui ont contribué de prés ou de loin à la réalisation de ce travail.

Dédicaces

Aux plus beaux créatures que Dieu a crée sur terre,à ces sources de tendresse, de patience et de générosité.

À mes parents .

À mes deux frères Abd el nour et Amine .

À ma sœur Rima.

À mon binôme Bakhali Fatma Zohra et sa famille.

À tou(te)s mes ami(e)s et collègues.

À tous les étudiants de la promotion 2017/2018 Option :Recherche Opérationnelle Optimisation et Management Stratégique (ROOMS)

À tous ceux qui, par un mot, m'ont donné la force de continuer .

Je dédie ce modeste travail...Nesrine

Dédicaces

Aux plus beaux créatures que Dieu a crée sur terre, à ces sources de tendresse, de patience et de générositéâ .

À mes parents .

À mes deux frères Mouloud et Mohamed .

À ma sœur Assia .

À mon binôme Guenoun Nesrine et sa famille

À tou(te)s mes ami(e)s et collègues

À tous les étudiants de la promotion 2017/20118 option :Recherche Opérationnelle, Optimisation et Management Stratégique (ROOMS)

À tous ceux qui, par un mot, m'ont donné la force de continuer.

Je dédie ce modeste travail...Fatma Zohra

Introduction générale

Les problèmes d'optimisation déterministes sont utilisés pour modéliser et analyser un grand nombre de systèmes pour les quels on recherche une stratégie optimale ($vis - à - vis$ d'un certain critère appelé encore fonction objectif) satisfaisant un certain nombre de contraintes. Dans ces problèmes, les paramètres définissant les contraintes et la fonction objectif sont supposés connus. Cependant, un grand nombre de problèmes d'optimisation sont des problèmes d'optimisation stochastique dont les paramètres ne sont pas tous connus avec certitude lors que la décision doit être prise.
La notion d'incertitude dans la programmation mathématique est apparue pour la première fois dans les années 50 avec les travaux de Dantzig [25], Cooper et Charnes [15], Beale [5]. Au cours des dernières années, des ingénieurs, des mathématiciens, des scientifiques et des décideurs ont manifesté un fort intérêt pour la prise en compte de lincertitude.

La plupart des modèles développés dans ce cadre considéraient en général un critère unique. Cependant, dans de nombreux cas, cette modélisation des problèmes ne traduit pas exactement la réalité à appréhender. Une autre façon de modéliser les problèmes a vu le jour, il y a maintenant une trentaine dannées, permettant une représentation fidèle de la réalité. La nouveauté consiste à optimiser plusieurs critères éventuellement conflictuels simultanément.

Dans ce mémoire, nous nous intéressons plus précisément au problème de la résolution d'un programme linéaire stochastique multi-objectif en nombres entiers (MOILSP). Il sagit d'un domaine spécifique de l'optimisation stochastique multi-objectif, qui, à cause de son lien direct avec la programmation linéaire multi-objectif en nombres entiers, possède ses particularités propres et ses techniques appropriées.
Dans le premier chapitre, nous présentons l'essentiel des définitions et résultats reliés à l'optimisation multi-objectif et nous parlerons des différentes approches de résolution.
Le deuxième chapitre expose l'essentiel des définitions et des résultats liés à l'optimisation multi-objectif et nous parlerons des différentes approches de résolution.
Le troisième chapitre se focaliser sur la programmation stochastique. Différents modèles de prise en compte de l'incertitude seront présentés, dont les modèles de recours, les contraintes probabilistes, et on expose la méthode de décomposition de Benders ($L -$

shaped) concernant la résolution des problèmes pouvant être modélisés en tant que modèles de recours en deux étapes, et on définit quelques notions liées à la programmation stochastique $multi-objectif$.
Le quatrième chapitre finalise notre travail par la présentation d'une méthode exacte élaboré par S.AMROUCHE et M.MOULAÏ [53] pour déterminer toutes les solutions, dites efficaces, d'un problème de programmation stochastique linéaire multi-objectif en nombres entiers (MOILSP).
Enfin, le mémoire s'achèvera par une conclusion général et quelques perspectives de recherche.

Chapitre 1

Programmation linéaire en nombres entiers

1.1 Introduction

Dans ce chapitre, nous faisons de brefs rappels sur les principaux résultats de la programmation linéaire continue (LP) et de la programmation linéaire en variables entières (ILP). Nous abordons uniquements les résultats nécéssaires pour l'étude et la résolution du problème de la programmation linéaire stochastique multi-objectif en nombres entiers (MOISLP). Nous commançons par la formulation des problèmes (LP) et nous énonçons quelques proprietés et résultats fondamentaux de la programmation linéaire. Nous formulons ensuite les problèmes (ILP) et nous décrivons quelques méthodes de résolutions.

1.2 La programmation linéaire continue

Un programme linéaire (LP) est un problème d'optimisation consistant à maximiser (minimiser) une fonction objectif linéaire à n variable continue les soumises à un ensemble de contraintes exprimées sous la forme d'équations ou d'inéquations linéaire.

1.2.1 La formulation du problème

Un programme linéaire s'écrit sous la forme suivante :

$$(LP)\begin{cases} Max(resp : Min)Z = \sum_{j=1}^{n} c_j x_j & \\ s.c : & \\ \sum_{j=1}^{n} a_{ij}x_j \leq, (\geq\ ou =) b_j & \forall \text{i=1,...,m} \\ x_j \geq 0 & \forall \text{j=1,...,n.} \end{cases}$$

où :
Z :la fonction objectif qu'on cherche à optimiser.
x_j :variable de décision $j = 1, ..., n$.
a_{ij},b_j,c_j sont des constantes
Cette forme génrale peut être simplifie, en effet il est possible de ramener le problème à des formes plus compactes mais équivalentes en particulier aux formes dites "canonique" et "standard".

La forme canonique

Un problème de maximisation (resp :minimisation) peut être écrit sous la forme canonique comme suit :

$$(LP)\begin{cases} Max(resp : Min)Z = Cx \\ s.c : \\ Ax \\ x \geq 0 \end{cases}$$

où :
$x = (x_1, x_2, ..., x_n)^t$:vecteur des variables de décision.
$b = (b_1, b_2, ..., b_m)^t$:vecteur de second membre de taille m.
$C = (c_1, c_2, ..., c_n)^t$:vecteur coût de taille n.
A :la matrice des contraintes des taille $m \times n$.
De manière évidente, il est toujours possible de ramener l'optimisation à une maximisation (minimiser la fonction Z est équivalent à maximiser la fonction -Z) ; Toutes les inégalités $\geq$ à des inégalités de même type $\leq$ (il suffit de multiplier par (-1), le cas échéant) ;

La forme standard

On dit qu'un prorgramme linéaire est écrit sous forme standard si toutes les containtes sont des égalités d'où le (LP) devient :

$$(LP)\begin{cases} Max(resp:Min)Z = Cx \\ s.c: \\ Ax = b \\ x \geq 0 \end{cases}$$

Dans la suite nous utilisirons le plus souvent la forme canonique

1.3 Notion de dualité

La notion de dualité est un aspect très important de la programmation linéaire, elle fut dévloppée par John Newmann en 1947 où il a démontré qu'à tout modèle linéaire qu'on appelle programme primal (LP) correspond un autre appelé programme dual (D).
Problème dual d'un problème de programmation linéaire
Considérons le problème de LP suivant :

$$(LP)\begin{cases} Max\ Z = C^T x \\ s.c: \\ Ax \leq b \\ x \geq 0 \end{cases}$$

Son dual est le programme linéaire suivant :

$$(D)\begin{cases} Min\ W = b^T y \\ s.c: \\ yA^T \leq c \\ y \geq 0 \end{cases}$$

Evidemment, on peut définir le dual d'un programme linéaire quelconque.
Le tableau suivant résume les correspondances entre primal et dual et permet d'écrire directement de dual d'un programme linéaire quelconque.

$Max\ Z = C^T x$	$Min\ W = b^T y$
$i^{eme} contrainte \geq$	variable $y_i \leq$
$i^{eme} contrainte \leq$	variable $y_i \geq$
$i^{eme} contrainte =$	$y_i \in \mathbb{R}$
$x_j \geq 0$	$j^{eme} contrainte \geq$
$x_j \leq 0$	$j^{eme} contrainte \leq$
$x_j \in R$	$j^{eme} contrainte =$

Exemple
Soit le problème primale suivant :

$$(LP)\begin{cases} Max\ Z = 3x_1 + 5x_2 \\ s.c: \\ x_1 \leq 4 \\ 2x_2 \leq 12 \\ 3x_1 + 2x_2 \leq 18 \\ x_1, x_2 \geq 0 \end{cases}$$

Le problème dual qui lui est associé est :

$$(D)\begin{cases} Min\ W = 4y_1 + 12y_2 + 18y_3 \\ s.c: \\ y_1 + 3y_3 \geq 3 \\ 2y_2 + 2y_3 \geq 5 \\ y_1, y_2, y_3 \geq 0 \end{cases}$$

Théorème 1

Soient x et y des solutions réalisables réspectivement du primal et du dual. si $C^T x = b^T y$ alors,x et y sont des solutions optimales.

Remarque 1

Si un problème possède une valeur optimale infinie, son dual n'a pas de solutions.

1.4 Résultats fodamenteaux de la programmation linéaire

Définition 1

Un ensemble S non vide est dit convexe si et seulement si pour tout élément x et y de S et pour tout $\lambda \in [0, 1]$, $\lambda x + (1 - \lambda)y \in S$.

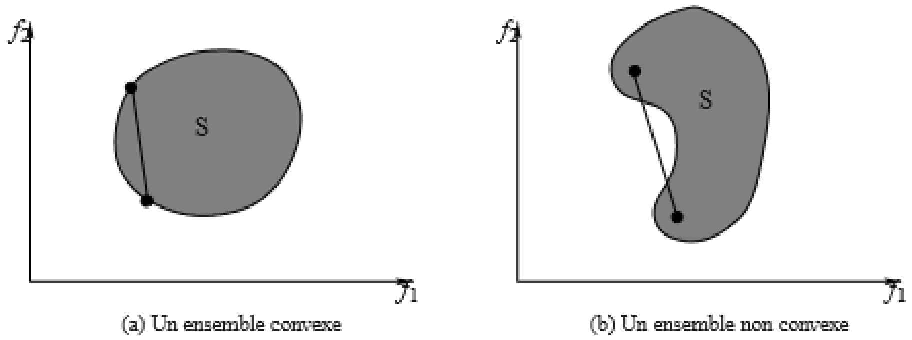

FIGURE 1.1: Ensemble convexe et ensemble non convexe

Définition 2

* L'ensemble $\{x \in R^n / Ax = b\}$ représente un hyperplan de R^n.
* L'ensemble $\{x \in R^n / Ax \leq b\}$ représente un demi-éspace fermé de R^n dont l'hyperplan correspondant constitue la frontière.

Définition 3 (Polyèdre)

* Un polyèdre S est l'intersection d'un nombre fini de demi-espace fermés et/ou d'hyper plans.
* Un polyèdre est un ensemble convexe fermé
* Un polyèdre S est borné s'il existe une valeur β finie et positive telle que $|x_j| \leq \beta \quad \forall j \in \{1, ..., n\}$ *et* $\forall x \in S$.
* Un polytope est un polyèdre borné et non vide.

1.4.1 Optimalité et point extrême

Définition 4

Soit S un ensemble convexe non vide de R^n. x est dit point extrême ou sommet de S si

$$x = \lambda x_1 + (1-\lambda)x_2, \forall x_1, x_2 \in S \text{ } et \text{ } \lambda \in]0,1[\text{ } alors \text{ } x = x_1 = x_2$$

Ou encore un point extrême est un point que ne peut pas s'écrire comme combinaison convexe sticte de deux points différentes de l'ensemble en question. Donc, tout les points extrêmes sont des points frontières.

Remarque 2

Le problème (LP) étant mis sous forme standard, chaque sommet de polyèdre $S = \{x \in R/Ax = b, x \geq 0\}$ corréspond à une seule solution réalisable et inversement, cette solution est appelée *solution réalisable de base* à laquelle corréspond une base réalisable notée B.

Théorème 2

L'ensemble des points extrêmes du polygone convexe $S = \{x \in R/Ax = b, x \geq 0\}$, est égale à l'ensemble des solutions de bases réalisables du problème (LP).

Théorème 3

La fonction objectif du problème (LP), atteit son maximum en un point extrême de l'ensemble des solutions réalisables.
Si la fonction objectif atteint son maximum en plusieurs points, elle prendra cette même valeur maximale en tout points qui est une combinaison linéaire convexe de ces points extrêmes.

1.5 L'algorithme du simplexe

Dévlopée en 1947 George Dantzig, la méthode du simplexe reste d'actualité pour résoudre des problèmes de grande taille. L'algorithme simplexe est une procédure itérative qui converge vers la(les) solution(s) de base réalisable(s) fournissant l'optimum de la fonction objectif, ou à défaut, mettra en évidence que l'ensemble des solutions réalisable est vide. L'algorithme du simplexe contient deux phases :
Phase 1 procedure d'initialisation
Déterminer une première solution réalisable de base, c'està dire les coordonnées d'un premier sommet de l'ensemble des solutions réalisables. Si cette procédure échoue, cela signifiera que l'ensemble des solutions réalisables est vide et que le problème est impossible(les contraintes ne peuvent pas être satisfaites simultanément).
Phase 2 procédure itérative
Calculer, à partir d'une solution de base réalisable, une autre solution de base donnant une valeur meilleure de la fonction objectif ; géométriquement, une itération consistra à passer d'un sommet de l'ensemble des solutions réalisables à un autre sommet sera adjacent au précédent en ce sens qu'ils seront les deux extrémités d'une arête de l'ensemble des solutions réalisables ; les coordonnées de ces deux sommets adjacents corréspondront

à deux solutions réalisables de base. Cette caractéristique permettra de déterminer relativement aisément la nouvelle solution réalisable de base à partir de l'ancienne.
Enfin, il existera une procédure d'arrêt de l'algorithme : deux test d'arrêt mettront en évidence, respectivement que :

— Une solution optimale est déterminée.
— Il n'existe pas de solution optimale finie : l'ensomble des solutions réalisables est non borné et Z peut y tendre vers l'infini.

1.6 L'algorithme dual du simplexe

Pour appliquer l'algorithme dual on suppose qu'une première base initiale est disponible. Si cette hypothèse n'est pas vérifiée, il faudrait alors appliquer la méthode des variables artificielles J.TEGHEM [67] et V. CHVATAL [24].
Le principe de l'algorithme dual va consister à appliquer l'algorithme simplexe au problème dual, mais à transporter les calculs au problème primal : autrement dit, le raisonnement se fera sur les solutions de base $Y^{(B)}$, mais les calculs s'effectueront sur les solutions $X^{(B)}$.
Comme pour l'algorithme simplexe, il faut distinguer deux phases dans l'algorithme dual.
Phase 1 : procédure d'initialisation
Déterminer une première base duale réalisable. Cette phase n'est donc pas nécéssaire que si la base initiale, supposée déterminée par hypothèse, n'est pas duale réalisable. Si cette procédure échoue, cela signifie qu'une telle base n'existe pas et donc que le polyèdre des solutions réalisables du dual est vide. Le problème dual n'admet pas donc de solution et il résulte que soit le problème primal n'admet pas de solution, soit il est non borné.
Phase 2 : procédure itérative
Déterminer à partir d'une base réalisable B, une autre base B, elle aussi duale réalisable, mais correspondant à une solution précédente $Y^{(B)}$, du point de vue de la fonction objectif w du dual. La nouvelle solution duale réalisable $X^{(B^*)}$ fournira une moins bonne valeur de la fonction objectif z, que celle fournie par la solution précédente $X^{(B)}$ car les solutions $X^{(B)}$ considérées ne sont pas (primales) réalisables et la décroissance de la valeur de la fonction objectif z vise à obtenir une solution $X^(B)$ primale réalisable ; dès qu'une telle solution $X^(B)$, à la fois duale et primale réalisable, sera obtenue, ce sera une solution optimale et la procédure itérative s'arrêtera.
Dans cette phase, une procéédure d'arrêt mettra en évidence

— Soit qu'une solution optimale finie est obtenue,
— Soit que le problème dual est non borné et que dès lors, le problème primal n'admet pas de solution.

La méthode duale simplexe est due à C.E LEMKE [49]. Pour une étude détaillée et complète des algorithmes simplexe et dual simplexe décrits brièvement ci-dessus, on pourra consulter l'ouvrage [67].

1.7 La 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 en vue leurs naissances en 1958. Un problème de programmation linéaire en nombres entiers(ILP) est un programme linéaire, sous des contraintes linéaires, dans lequel il y a la contrainte supplémentaire que les variables sont entières.

1.7.1 Formulation du problème

Le programme linéaire en nombres entiers s'écrit sous la forme suivante :

$$(ILP)\begin{cases} Max(resp: Min)Z = \sum_{j=1}^{n} c_j x_j \\ s.c: \\ \sum_{j=1}^{n} a_{ij}x_j \leq, (\geq ou =) b_j & \forall \text{i=1,...,m} \\ x_j \in Z^n & \forall \text{j=1,...,n.} \end{cases}$$

1.7.2 Méthode de résolution

Il s'agit plusieurs méthodes de résolution des problèmes linéaires en nombres entiers, nous citons par exemple, la méthode des coupes fractionnaire et les coupes mixtes de Gomory exposées dans les années soixante [34] et [35], les coupes de Chvàtal-Gomory de Chvàtal [23]

Procedure de séparation et évaluation(Branch and Bound)

La méthode de *Branch and Bound* a été spécialement élaborée pour des problèmes à variables discrètes. Elle a été à l'origine par Land et Doig [48]
Soit le problème (ILP) suivant :

$$(ILP)\begin{cases} MaxZ(x) \\ s.c: \\ x \in D = \{x \in R^n / Ax \leq b; x \in Z^n\} \end{cases}$$

Le programme linéaire (LP) obtenu à partir de (ILP) en rechrchant les contraintes d'integrité des variables est appelé programme relaxé.
La méthode combine de façon éléante la méthode primale-dual du simplexe et le principe de séparation et évaluation dans une arborescence de recherche. Elle repose sur les étapes suivantes :
Au nœud 0 de l'arborescence :
On pose $U = \underline{z}$ un minorant de la valeur optimale de la fonction objectif et on résout le problème relaxé (PL) avec la méthode primale du simplexe.

Séparation : Si la solution optimale, disons $\hat{x}$, a ses coordonnées $\hat{x}_j \in N$, pour tout j = 1,...,n, terminer. Cette solution est optimale pour (PLNE). Sinon, choisir une variable x_j qui n'est pas entière. Partitionner l'ensemble des solutions en deux sous-ensembles en ajoutant l'une ou l'autre des contraintes :

$$x_j \leq \lfloor \hat{x}_j \rfloor \quad ou \quad x_j \geq \lceil \hat{x}_j \rceil$$

où $\lfloor a \rfloor$ indique la partie entière du nombre réel a, et $\lceil a \rceil = \lfloor a \rfloor + 1$
. Deux nouveaux nœuds sont alors crées. Notons que $\hat{x}$ viole à la fois ces deux contraintes et que le coût optimal de chacun des deux sous-problèmes sera inférieur à celui de (PL) :
Evaluation : Résoudre le programme (PL) avec la nouvelle contrainte en utilisant la méthode duale du simplexe.
Test de stérilisation :

1. Si la nouvelle solution optimale x est telle que $z_{PL}(x) \leq U$.
2. Si le dual du simplexe implique qu'il n'y a pas de solutions réalisables.
3. Si la solution optimale x est réalisable pour le problème initial (PLNE), c'est-à-dire entière. Si de plus,$z_{PL}(x) > U$, alors poser $U = z_{PL}(x)$ et garder la solution x comme meilleure solution de la branche courante.

A un nœud k :
Faire : l'évaluation, le test de stérilisation et eventuellement, la séparation.
Plusieurs choix sont possibles pour le choix d'un nœud à évaluer : profondeur d'abord, meilleur d'abord et largeur d'abord. Le plus utilisé est profondeur d'abord pour trouver rapidement une solution réalisable et ne pas trop encombrer la mémoire du calculateur.
La méthode s'arrête lorsque tous les nœuds pendants de l'arborescence (nœuds de degré 1) sont sondés (le test de stérilisation est positif).

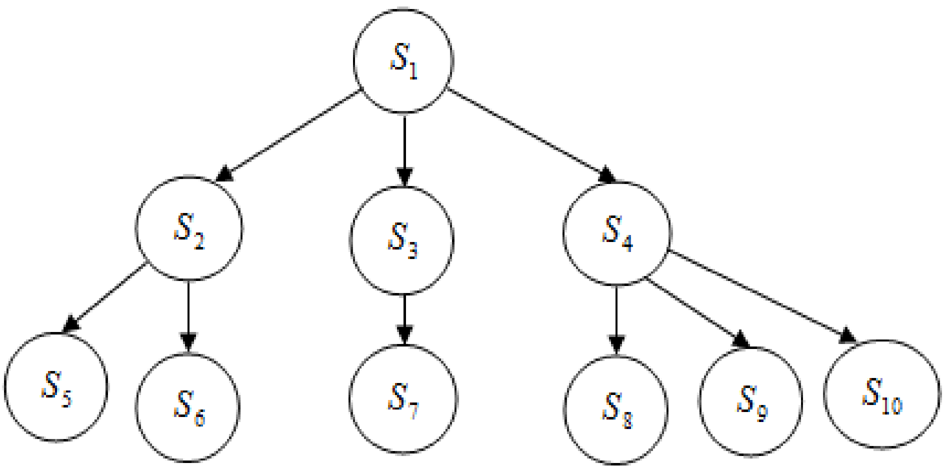

FIGURE 1.2: Principe de Branch end Bound

1.7.3 Méthodes des coupes de Gomory

L'algorithme utilisant les coupes de Gomory procéde comme suit :
Soit $\bar{x}$ une solution optimale du problème relaxé (LP) trouvée par la méthode du simplexe. Si $\bar{x}$ est une solution entière, elle est optimale pour (PLNE). Sinon, 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 :

$$x_j + \sum_{k \in N} \hat{a}_{ik} x_k = \bar{x}_j \tag{1.1}$$

où N est l'ensemble des indices des variables hors-base. La contrainte :

$$\sum_{k \in N} f(\hat{a}_{ik}) x_k \geq f(\bar{x}_j) \tag{1.2}$$

est alors déduite de l'expression précédente, où $f(r) = r - \lfloor r \rfloor$ désigne la partie fractionnaire du nombre réel r.
Cette coupe, appelée *coupe fractionnaire de Gomory*, ou coupe fondamentale, peut être rajouter au tableau courant du simplexe.
Dans la pratique, les coupes fractionnaires de Gomory convergent trés lentement.
Dautres coupes ont été introduites dont l'intention de les rendres plus performantes. En particulier, Gomory [] lui même a donné les coupes suivantes :

Proposition 1

Pour tout entier naturel t, l'inéquation

$$\sum_{k \in N} f(t\hat{a}_{ik}) x_k \geq f(t\bar{x}_j) \tag{1.3}$$

est une coupe. De plus, si $f(\hat{a}_{ik}) < \frac{1}{2}$ et $\frac{1}{2} f(t\hat{a}_{ik}) < 1$, alors la coupe 1.3 est plus profonde que la coupe 1.2

Proposition 2

L'inéquation

$$\sum_{k \in N} min\{f(\hat{a}_{ik}), f(\bar{x}_j) \frac{1 - f(\hat{a}_{ik})}{1 - f(\bar{x}_j)}\} x_k \geq f(\bar{x}_j) \tag{1.4}$$

est une coupe. De plus, elle est plus profonde que la coupe 1.2.
De même, pour tout entier naturel t, l'inéquation

$$\sum_{k \in N} min\{f(t\hat{a}_{ik}), f(t\bar{x}_j) \frac{1 - f(t\hat{a}_{ik})}{1 - f(t\bar{x}_j)}\} x_k \geq f(t\bar{x}_j) \tag{1.5}$$

est une coupe plus profonde que 1.3s.

1.8 Conclusion

Ce chapitre est consacré à l'introduction de la problématique linéaire en nombres entiers. Les définitions et les concepts de base sont présentés (convexité, dualité...). Nous avons décrit quelques méthodes exactes de résolution comme le simplexe, Branch and bound et la méthode de coupe de Gomory

Chapitre 2

L'optimisation multi-objectif

2.1 Introduction

Dans ce chapitre, nous présentons l'optimisation mathématique multi-objectif qui sera le cadre de travail de ce mémoire. Nous introduisons les concepts fondamentaux tels que la dominance et la surface de compromis. Nous décrivons aussi les principales approches de résolution pour ces problèmes.

2.2 Notions fondamentales

La plupart des problèmes d'optimisation réels sont d'écrits à l'aide de plusieurs objectifs ou critères souvent contradictoires ou conflictuels devant être optimisés simultanément. Il n'existe généralement pas de solution qui optimise tous les critères en même temps, le concept de solution optimale devient alors plus difficile à définir. En effet, en considérant deux critères contradictoires "a" et "b", améliorer "a" détériore forcement "b" et inversement, il faut donc trouver un compromis. Dans ce cas, la solution optimale cherchée n'est plus un point unique, mais un ensemble de compromis. Résoudre un problème comprenant plusieurs critères, consiste donc à calculer le meilleur ensemble de solutions compromis : *le front Pareto.*

Les premiers travaux dans ce domaine sont dus à Koopmans [46] en 1951 qui donna une condition nécessaire et suffisante d'efficacité d'une solution suivis par les travaux de Kuhn et Tucker [47] la même année qui formulèrent un problème de maximisation vectorielle.

Depuis, ce domaine a connu un développement fulgurant, traitant aussi bien du domaine de la programmation multi-objectif linéaire [52], [62] et non-linéaire [54], [55], [30], [19] que des problèmes multi-objectifs booléen ou en nombres entiers [66].

2.2.1 Position du problème

On définie un problème multi-objectif comme un problème de décision qui consiste à optimiser (maximiser ou minimiser) simultanément r fonctions réelles notées f_k, $k = 1, ...r$, appelées critères souvent contradictoires, sur un ensemble de solutions S.

Ce problème peut être formulé mathématiquement comme :

$$(M_{OP}) \begin{cases} "opt"[f_1(x),\ f_2(x),\ \dots\ , f_r(x)] \\ x \in S \end{cases}$$

Le symbole " " signifie qu'il n'est généralement pas possible de trouver dans S une solution qui optimise simultanément les r critères. Il est remarquable que de cette façon un problème multi-objectif est correctement formulé par rapport à la réalité concernée par le problème de décision. Il faut donc déterminer une solution $x^* \in S$ telle que, en regard des solutions de S, le vecteur $[f_1(x),\ f_2(x),\ \dots\ , f_r(x)]$ est bon, acceptable selon les préférences du décideur (DM : <u>D</u>écision <u>M</u>aker), voir optimal, à condition de se doter d'un cadre décisionnel donnant une signification à cette notion.

Dans tous ce qui suit, on considère que toutes les fonctions objectifs sont à "Maximiser".

— Chaque solution est caractérisée par un vecteur $x = (x_1, x_2, ..., x_n)$ représentant le vecteur de décision avec x_i, $i = 1, ..., n$, les variables de décision et n le nombre de ces variables,
— $f(x) = (f_1(x),\ f_2(x)\ , ...,\ f_r(x))$ est le vecteur de r critères f_k, $k = 1, ..., r$ et r le nombre de critères,
— L'ensemble S est un sous-ensemble de R^n décrit implicitement par des contraintes d'égalités et/ou d'inégalité,
— L'ensemble R^n qui contient S est dit *espace de décision* ,
— L'ensemble $F = f(S)$ qui est la projection de l'espace S sur l'espace des critéres.
— L'ensemble R^r qui contient F est dit *espace des critères*,

2.2.2 La relation de dominance

Lors de résolution d'un problème multi-objectif, toute solution admissible ($x \in S$), ne constitue pas évidemment un bon compromis à considérer ; pour qu'elle le soit, on impose généralement une propriété, basée sur une relation d'ordre partiel, notée "$\succ$" et appelée dominance :

Définition 5 (Dominance)

Soient deux vecteurs critères $f(x), f(y) \in F$. On dit que $f(x)$ domine $f(y)$, et on note $f(x) \succ f(y)$, si et seulement si $f(x) \geq f(y)$ et $f(x) \neq f(y)$
i.e.

$$f_i(x) \geq f_i(y)\ \forall\ i = \{1, ..., r\} \text{ et } \exists\ i \in \{1, ..., r\} \text{ tel que } f_i(x) > f_i(y).$$

Si $f(x)$ domine $f(y)$, alors $f(x)$ est au moins aussi bon que $f(y)$ sur tous les critères et meilleur que lui sur au moins un critère.

Nous illustrons la relation de dominance par un exemple en dimension 2 (Voir figure 2.1).

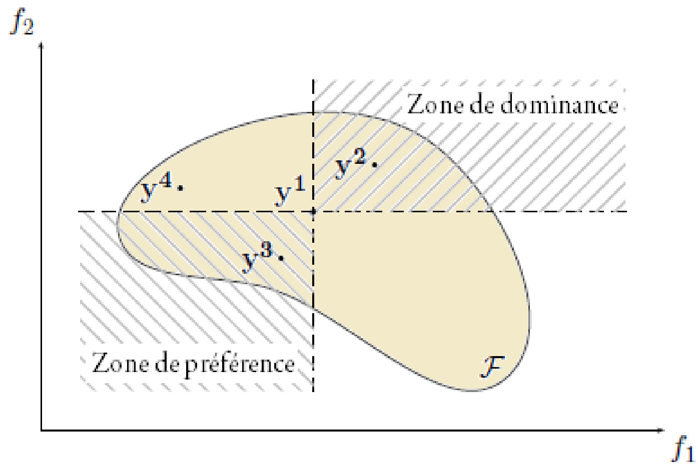

FIGURE 2.1: Exemple de dominance

Sur cette figure, F (l'espace réalisable dans l'espace des critères) est l'image de S. Ainsi chaque point y^i est l'image de x^i par $f : y^i = f(x^i)$. Prenons le point y^1 comme point de référence, nous pouvons distinguer trois zones :

— La zone de préférence est la zone contenant les points dominés par y^1.
— La zone de dominance est la zone contenant les points dominant y^1.
— La zone d'indifférence contient les points incomparables avec y^1.

Ainsi, il est claire que y^3 est dominée par y^1, que y^2 domine y^1 et que y^4 est non dominée par y^1 (incomparable avec y^1).

2.2.3 Optimalité de Pareto

Définition 6 (Efficacité)

Une solution $\bar{x} \in S$ est dite solution efficace (ou Pareto optimale) si et seulement s'il n'existe pas de solution $x \in S$ telle que $f_i(x)$ domine $f_i(\bar{x})$ $\forall\ i\ \in \{1, ..., r\}$.

Un point est efficace si son image par f est un vecteur critère non dominé. Une définition équivalente de l'efficacité est :

Définition 7

Une solution $\bar{x} \in S$ est dite solution efficace si et seulement s'il n'existe pas de solution $x \in S$ telle que

$$f_i(x) \geq f_i(\bar{x}),\ \forall\ i\ \in \{1, ..., r\}\ et\ \exists\ j \in \{1, ..., r\}\ avec\ f_j(x) > f_j(\bar{x}).$$

A partir d'un point efficace, il est impossible d'augmenter la valeur d'un des critères sans diminuer la valeur d'au moins un autre critère.

Définition 8

Une solution $\bar{x} \in S$ est dite solution faiblement efficace si et seulement s'il n'existe pas de solution $x \in S$ telle que :

$$f_i(x) > f_i(\bar{x}),\ \forall\ i\ \in \{1, ..., r\}.$$

Il est clair qu'une solution efficace est faiblement efficace, mais l'inverse est faux.

Nous pouvons définir l'optimalité locale et l'optimalité globale au sens de Pareto comme suit :

Définition 9 (Optimalité locale au sens de Pareto)

Un vecteur $x \in S$ optimal localement au sens de Pareto s'il existe un réel $\delta > 0$ tel qu'il n'y ait pas de vecteur x' dont le vecteur critère domine celui de x avec $x' \in S \cap B(x, \delta)$, ou $B(x, \delta)$ représente une boule de centre x et de rayon δ.

D'une manière équivalente un vecteur x est optimal localement au sens de Pareto s'il est optimal au sens de Pareto sur une restriction de l'ensemble S.

Définition 10 (Optimalité globale au sens de Pareto)

Un vecteur $x \in S$ optimal globalement au sens de Pareto (ou optimal au sens de Pareto) s'il n'existe pas de vecteur x' dont le vecteur critère domine celui de x.

Une version "*graphique*" de l'optimalité au sens de Pareto utilise le théorème du contact.

Définition 11

Soit $V \subset R^r$, $V \neq \emptyset$. V est un cône si $\alpha.v \in V$ pour tous scalaire $\alpha \geq 0$ et tout $v \in V$.

Définition 12 (Cône non négatif)

Un cône non négatif est défini dans R^r par $C^+ = \{x | f(x) \in^r \ \ et\ \ f(x) \geq 0\}$.

Définition 13 (Théorème du contact)

Un vecteur x est optimal au sens de Pareto pour un problème d'optimisation multi-objectif donné si $(C^+ + x) \cap F = \{x\}$.

L'utilisation de ce théorème est illustré par la figure 2.2

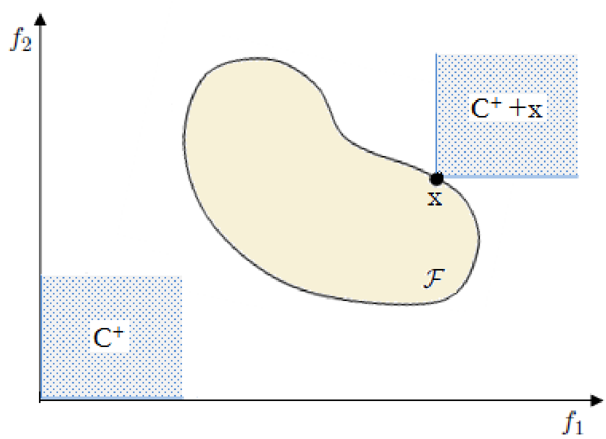

FIGURE 2.2: Le théorème du contact

Définition 14 (Ensemble Pareto optimal)

L'ensemble Pareto optimal de S (ou solutions efficients), est défini par l'ensemble Eff :

$$Eff = \{x \in S| \nexists x' \in S,\ f(x') \text{ domine } f(x)\}.$$

Définition 15 (Frontière Pareto de F)

Soit F l'image dans l'espace des critères de l'ensemble réalisable S. La frontière Pareto SND de F est définie par :

$$SND = \{y \in F| \nexists y' \in F,\ y' > y\}.$$

Elle est aussi définie comme l'image de l'ensemble Pareto optimal dans l'espace F.

Un exemple de surface de Pareto en dimension 2 est montré à la figure 2.3.

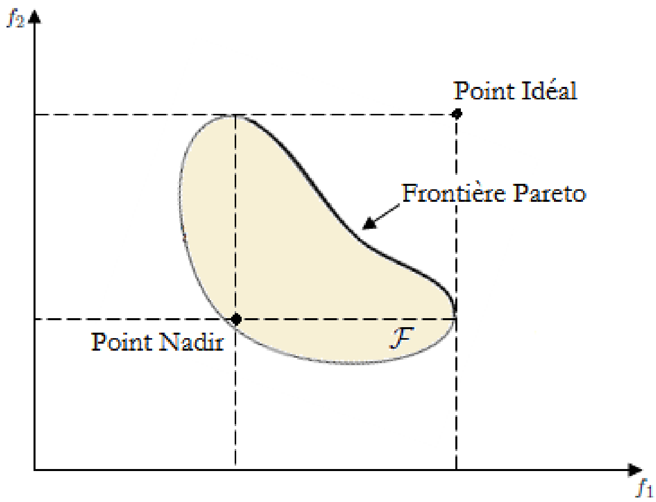

FIGURE 2.3: La Frontière Pareto

Dans cet exemple, le problème considéré est un problème de maximisation avec deux critères.

Deux points particuliers apparaissent clairement : le *point idéal* et le *point Nadir*, ces deux points sont calculés à partir de la frontière Pareto. Le point idéal (resp. le point Nadir) domine (resp. est dominé par) tous les autres points de la surface de Pareto. Bien que ces points ne soient pas forcément compris dans la zone réalisable, ils servent comme *point de référence* permettant de discuter de l'interêt des solutions trouvées. Les coordonnées de ces points sont définies comme suit :

Définition 16 (Point Idéal)

Les coordonnées du point idéal (Z^I) correspondent aux meilleurs valeurs de chaque objectif des points du front Pareto. Les coordonnées de ce point correspondent aussi aux valeurs obtenues en optimisant chaque fonction objectif séparément :

$$Z_i^I = \max\{y_i \mid y \in SND(F)\}.$$

Définition 17 (Point Nadir)

Les coordonnées du point Nadir (Z^{nad}) correspondent aux pires valeurs de chaque objectif des points du front Pareto.

$$Z_i^{nad} = \min\{y_i \mid y \in SND(F)\}.$$

2.3 Choix de la méthode d'aide à la décision

Dans la résolution d'un problème multi-objectif menant à la détermination d'un ensemble de solutions Pareto, il est nécessaire de faire intervenir l'humain à travers un décideur, pour le choix final de la solution à garder. Ainsi, avant de se lancer dans la résolution d'un problème multi-objectif, il faut se poser la question du type de méthode d'optimisation à utiliser. Elles peuvent être classées suivant trois catégories qui diffèrent selon les préférences du décideur pour la construction de sa fonction d'utilité.
Nous pouvons trouver les familles suivantes [28] :

— *Les méthodes d'optimisation à priori :* dans ce cas, le compromis que l'on désire faire entre les critères a été défini avant l'exécution de la méthode. Ainsi une seule exécution permettra d'obtenir la solution recherchée. Cette approche est donc rapide, mais il faut cependant prendre en compte le temps de modélisation du compromis et la possibilité pour le décideur de ne pas être satisfait de la solution trouvée et de relancer la recherche avec un autre compromis.

— *Les méthodes d'optimisation progressives :* ici, le décideur intervient dans le processus de recherche de solutions en répondant àdifférentes questions afin d'orienter la recherche. Cette approche permet donc de bien prendre en compte les préférences du décideur, mais nécessite sa présence tout au long du processus de recherche.

— *Les méthodes d'optimisation à posteriori :* Dans cette troisième famille de méthodes, on cherche à fournir au décideur un ensemble de bonnes solutions bien réparties. Il peut ensuite, au regard de l'ensemble des solutions, sélectionner celle qui lui semble la plus appropriée. Ainsi, il n'est plus nécessaire de modéliser les préférences du décideur (ce qui peut s'avérer être très difficile), mais il faut en contre-partie fournir un ensemble de solutions bien réparties, ce qui peut également être difficile et requérir un temps de calcul important (mais ne nécessite pas la présence du décideur).

On peut trouver des méthodes d'optimisation multi-objectif qui n'entrent pas exclusivement dans une famille. On peut utiliser, par exemple, une méthode a priori en lui fournissant des préférences choisies aléatoirement.

Nous nous placerons dans le cadre de cette troisième famille de méthodes où la modélisation des préférences n'est pas requise et où le procédé d'optimisation doit être puissant afin de fournir l'ensemble de solutions Pareto optimales ou à défaut une très bonne approximation de la frontière Pareto.

Dans ce type de méthode, deux phases importantes sont à considérer : la phase de recherche de l'ensemble des solutions Pareto optimales, que nous appellerons de façon abusive, *résolution du problème d'optimisation* et la phase de choix parmi ces solutions, qui relève de l'aide à la décision. Cette deuxième phase ne sera pas traitée dans notre travail.

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

Dans la littérature, plusieurs approches de résolution du problème de la programmation multi-objectif sont considérées. Des ouvrages ou articles de synthèse ont été rédigés, des états de l'art plus complets peuvent être consultés notamment dans (Ulungu et al.) [68], Miettinen [55], Ehrgott [30], (Ehrgott et al.) [29], Deb [27], (Collette et al.) [19].

Deux approches de résolution d'un probléme d'optimisation multi-objectif peuvent être distinguées dans la littérature voir Ehrgott [30], (Roy et al.) [58] et Roy [59].

— La première approche dite *approche mono-objectif* consiste à ramener le problème à un problème d'optimisation unicritère, au risque d'enlever toute signification au problème.

— La deuxième approche dite *approche multi-objectif* consiste à proposer des solutions en tenant compte de l'ensemble des critères.

2.4.1 Approche unicritère

Elle caractérise les méthodes "à priori" utilisés pour leur simplicité de mise en oeuvre. En effet, les critères du problème d'optimisation sont transformés en un seul critère. Dans ce cas, le décideur est supposé quantifier à priori l'importance de chaque critère afin de construire un critère unique. Le processus d'optimisation unicritère est ensuite lancé afin d'obtenir la solution "optimale". Plusieurs exécutions sont effectuées dans le but de trouver un ensemble de solutions qui approxime l'ensemble optimal de Pareto.

Nous donnons ci-dessous des méthodes a priori ou méthodes scalaires les plus connues, d'autres méthodes sont exposées dans Miettinen [55], Collette et al. [19].

Théorie de l'utilité multi-attributs

La théorie de l'utilité multi-attributs a fait l'objet de nombreux travaux de recherche dans les années 70 voir Fishburn [32], Keeney [43], Huber [36], Farquhar [31]. Elle repose sur l'axiome fondamental suivant : tout décideur essaie inconsciemment d'optimiser une fonction $U = U(f_1, ..., f_r)$ qui agrège tous les points de vue à prendre en compte, comme cela se présente habituellement dans la théorie du consommateur en économie.
En d'autres termes, si l'on interroge le décideur sur ses préférences,ses réponses seront en accord avec une certaine fonction U que l'on ne connaît pas. Le problème est donc d'essayer d'estimer cette fonction.

Définition 18

Étant donné un ordre de préférence noté " $\succ$ " sur l'ensemble F, une fonction U à valeurs réelles vérifiant $U(f_1) > U(f_2) \Leftrightarrow f_1 \succ f_2$; est appelée fonction de préférence ou d'utilité.

Deux problèmes essentiels se posent dans le cadre de cette théorie :

— Quelles propriétés doivent posséder les préférences du décideur pour être representable par une fonction U ayant une forme analytique donnée (additive, multiplicative, mixte,...).
— Comment construire ces fonctions et estimer les paramètres intervenant dans la forme analytique choisie.

Différentes formes de fonctions d'utilité conduisent à différentes solutions pour le problème multi-objectif. Les modèles les plus couramment utilisés pour la fonction d'utilité U sont le modèle additif et le modèle multiplicatif :

$$U(f(x)) = \sum_{i=1}^{r} U_i(f_i(x)) \; et \; U(f(x)) = \prod_{i=1}^{r} U_i(f_i(x))$$

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

Méthode des sommes pondérées

Cette méthode de résolution d'un problème d'optimisation multi-objectif est la plus évidente et, probablement, la plus largement utilisée en pratique. Elle consiste à ramener le problème multi-objectif au problème de l'optimisation d'une combinaison linéaire des objectifs initiaux. Ainsi, il s'agit d'associer à chaque critère un coefficient de pondération et à faire la somme des critères pondérées pour obtenir un nouveau et unique critère.

Le problème multi-objectif (MOP) se transforme alors de la manière suivante :

$$(MOP)\begin{cases} \max \sum_{i=1}^{r} \lambda_i f_i(x) \\ sc. \\ x \in S \ et \ \lambda_i \in \Lambda = \{\lambda_i : \sum_{i=1}^{r} \lambda_i = 1 \ et \ \lambda_i \geq 0, \forall i \in \{1,...,r\}\}. \end{cases}$$

— Les résultats obtenus avec de telles méthodes dépendent fortement des paramètres choisis pour le vecteur de poids. Les poids λ_i doivent également être choisis en fonction des préférences associées aux objectifs, ce qui est une tâche délicate. Une approche généralement utilisée consiste à répéter l'exécution de l'algorithme avec des vecteurs poids différents.

— Cette méthode est très efficace du point de vue algorithmique, mais son principal inconvénient est qu'elle ne permet pas de trouver les solutions enfermées dans des concavités (surface de compromis non-convexe).

Méthode Goal programming

Cette méthode, qui relève d'une théorie très avancée dans le domaine des problèmes multi-objectifs, a été initialement conçue par Charnes et Cooper [14] dans le cas linéaire ; elle a été prolongée par des travaux d'Ijiri [38] et d'Ignizio [37] dans le cas non linéaire. Cette théorie a fait l'objet d'un nombre important de travaux théoriques et pratiques voir Chankong [22], Martel [50], Spronk [64] et Steuer [62]. L'idée générale de la méthode est d'établir un but à atteindre pour chaque critère. Généralement, le point qui satisfait tous les buts n'est pas réalisable, la solution préférée serait donc celle qui se rapproche le plus possible de ces buts.
Soit $f^* = (f_1^*, ..., f_r^*)$ le vecteur de référence ou but fixé par le décideur relativement à tous les critères, le problème revient à considérer la relation suivante :

$$\begin{cases} \min(\sum_{i=1}^{r} |f_i(x) - f_i^*|^p)^{1/p} \\ sc. \\ x \in S. \end{cases}$$

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

La méthodologie du Goal programming repose sur les points suivants :

— Fixer les valeurs des cibles f_i^* que l'on désire atteindre sur chaque critère ;
— Définir des déviations positives d_i^+ et négatives d_i^- relativement à ces buts ;
— Minimiser la somme pondérée de ces déviations relativement à une norme $\|.\|_p$ définie dans ^{n}.

La formulation mathématique du Goal programming est la suivante :

$$\begin{cases} \min_{x\in E}(\sum_{i=1}^{r}(d_i^- + d_i^+)^p)^{1/p} \\ sc. \\ f_i(x) + d_i^+ d_i^+ = f_i^* \\ d_i^- \geq 0,\ d_i^+ \geq 0,\ i \in \{1, ..., r\}. \end{cases}$$

d_i^+ et d_i^- sont respectivement appelés la sur-réalisation et la sous-réalisation du $i^{\text{ème}}$ critère.

Le Goal programming se ramène toujours à un problème de minimisation d'une seule fonction. La solution qui en découle est efficace ou faiblement efficace selon la norme utilisée. Son avantage est que la solution qui en résulte satisfasse le décideur de la façon la plus proche possible lorsque les buts et les priorités de chaque but à atteindre sont bien définis.

La méthode ϵ-contrainte

Dans cette méthode, on n'optimise qu'un seul objectif jugé important par le décideur, les autres sont transformés en contraintes d'inégalités par rapport à un vecteur seuil ϵ. Le principe de la méthode ϵ-contrainte ou méthode de compromis est intéressant lorsque l'on cherche à énumérer toutes les solutions d'un front Pareto.
Le problème peut être reformulé de la manière suivante :

$$\begin{cases} \max f_i(x) \\ sc. \\ f_1(x) \leq \epsilon_1 \\ \vdots \\ f_{i-1}(x) \leq \epsilon_{i-1} \\ f_{i+1}(x) \leq \epsilon_{i+1} \\ \vdots \\ f_r(x) \leq \epsilon_r \\ x \in S \end{cases}$$

L'approche par ϵ-contrainte doit aussi être appliquée plusieurs fois en faisant varier le vecteur ϵ pour trouver un ensemble de points Pareto optimaux.

Cette approche a l'avantage par rapport à la précédentes de ne pas être trompée par les problèmes non convexes.

L'inconvénient de cette approche réside dans le fait qu'il faille lancer un grand nombre de fois le processus de résolution. De plus, pour obtenir des points intéressants et bien répartis sur la surface de compromis, le vecteur ϵ doit être choisi judicieusement. Il est clair qu'une bonne connaissance du problème a priori est requise.

2.4.2 Approche multi-objectif

A- Approche à posteriori

Dans ces approches l'expression des préférences et la pondération des objectifs se fait a posteriori, c'est à dire que le processus d'optimisation détermine un ensemble de solutions candidates et la sélection de la meilleure solution ne se fait qu'à la fin du processus.

Les méthodes hybrides

La méthode hybride la plus connue est la méthode de Corley [21] en 1980. Cette méthode utilise la méthode de pondération des fonctions objectifs et la méthode de ϵ-contraintes.
Le problème peut être reformulé de la manière suivante :

$$\begin{cases} \max \sum_{i=1}^{r} \lambda_i f_i(x) \\ sc. \\ f_j x) \leq \epsilon_j \text{ pour } j \in \{1, ..., r\} \\ x \in S \end{cases}$$

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

- Cette méthode permet de combiner les avantages de la méthode des sommes pondérées et la méthode ϵ-contrainte. Elle est ainsi efficace sur différents types de problèmes, qu'ils soient convexes ou non convexes.
- Le nombre de paramètres à déterminer a été multiplié par deux. Il sera beaucoup plus difficile à l'utilisateur d'exprimer sa préférence en jouant sur l'ensemble des paramètres.

D'autres approches à posteriori sont détaillées dans Miettinen [55], Collette et al. [19].

B- Approche interactive

Le principe des méthodes de l'approche progressive est de guider l'exploration tout au long du processus d'optimisation. Le but est d'alterner entre le processus de recherche et le processus de décision défini par un classement de solutions, ou bien un choix des poids de pondération des objectifs. A chaque étape, l'ensemble des solutions est analysé afin d'orienter les futures itérations vers les zones les plus intéressantes de l'espace de recherche. Plusieurs méthodes interactives sont traitées dans Miettinen [55], Collette et al. [19].

Méthode STEM

Dans cette méthode, les informations sur la préférence de l'utilisateur permettent de restreindre l'espace de recherche étape par étape en ajoutant des contraintes sur les valeurs des critères. A chaque étape interactive, un nouveau compromis est trouvé en optimisant une norme de type :

$$S(Z, \lambda) = \max_{1 \leq i \leq r} \{\lambda_i \left| Z_i - \bar{z}_i \right|\} + \rho \sum_{i=1}^{r} \lambda_i \left| Z_i - \bar{z}_i \right|, \ \rho > 0.$$

sur une partie de $f(S)$ selon une direction correspondant à la relaxation d'un critère fixé par les décideurs.

Les différentes étapes de cette méthodes sont décrites dans [11].

- Le probléme de cette méthode, commun à toutes les méthodes interactives, est que l'on ne peut trouver qu'un seul point solution, et non la totalité de la surface de compromis.
- Le mode d'interaction fait que, pour une bonne utilisation de la méthode, l'utilisateur doit avoir une bonne connaissance à priori du problème. Il doit être capable de trouver de bonnes bornes pour restreindre son domaine.

2.5 Programmation Linéaire Multi-objectif en Nombre Entiers MOILP

Un problème de la programmation linéaire multi-objectif en nombres entiers est défini comme suit :

$$(P)\begin{cases} \max\ Z_1(x) = c^1x \\ \max\ Z_2(x) = c^2x \\ \vdots \\ \max\ Z_r(x) = c^rx \\ sc \\ x \in S = \{x \in R^n | Ax \leq b,\ x \geq 0\} \\ x \text{ vecteur entier} \end{cases}$$

où $r \geq 2$, $c^i \in^n$ pour tout $i = \{1, ..., r\}$; A est une $m \times n$-matrice et b un m-vecteur à coefficients entiers.

2.5.1 Méthode de coupes multiples pour l'élimination d'un ensemble dominé

Proposition

Soient les ensembles suivants :

$$A = \mathbb{Z}^n\ \{x \in \mathbb{Z}^n / \sum_{j=1}^{n} c_{ij}x_j \leq 0, i = 1, ..., k'\}$$

$$B_1 = \{x \in \mathbb{Z}^n / \sum_{j=1}^{n} c_{1j}x_j > 0\}$$

$$B_p = \{x \in \mathbb{Z}^n / \sum_{j=1}^{n} c_{pj}x_j > 0, \sum_{j=1}^{n} c_{ij}x_j \leq 0, i = 1, ..., (p-1)\}$$

$$B = (\bigcup_{p=1}^{k'} B_p)$$

Nous avons alors l'égalité : A = B

A partir de cette égalité densembles nous pouvons définir un algorithme de coupes multiples en un point $\hat{x}$ pouvant isoler l'ensemble dominé $D'_{\bar{x}}$ de l'ensemble S, et donner une partition de l'ensemble de solutions $(S'_{\bar{x}} \bigcap U_{\bar{x}})$; Cet algorithme estle suivant :

Algorithme de coupes multiples en un point $\hat{x}$

1. Sélectionner une solution $\hat{x}$ de S
2. Former les problèmes $(P_1); (P_2), ..., (P_k)$, identiques au problème suivant :

$$(P)\begin{cases} \min Z_i(x) = C_i(x) \quad i = 1, ..., k' \\ sc : Ax = b \\ x \geq 0 \; x \; entier. \end{cases}$$

3. Pour $i_0 = 1, ..., k'$, faire
   (a) Pour $i_1 = 1, ..., (i_0 - 1)$, faire introduire la coupe $C_{i1}x \leq C_{i1}\hat{x}$ dans le système de contraintes de (P_{i0})
   (b) Introduire la coupe $C_{i_0}x \geq C_{i_0}\hat{x} + 1$ dans le système de contraintes de (P_{i0}).

   Dans le cas de variables continues (remplacer dans l'instruction b l'inéquation $C_{i_0}x \geq C_{i_0}\hat{x}+1$ par $C_{i_0}x > C_{i_0}\hat{x}$), pour introduire les coupes qui isolent l'ensemble de solutions.
   Fin de l'algorithme

2.6 Conclusion

Nous avons présenté dans ce chapitre les principaux concepts de l'optimisation multi-objectif tels que la modélisation d'un problème multi-objectif, la notion de la dominance de Pareto et la structure de la surface de compromis. Une classification des méthodes de résolution a été introduite tout en essayant de présenter plusieurs approches utilisées pour aborder les problèmes de l'optimisation multi-objectif, ces méthodes sont passées en revue et les avantages et les inconvénients de chacune sont mentionnées.

Chapitre 3

L'optimisation stochastique

3.1 Introduction

Pour beaucoup de problèmes réels, les données de problème ne peuvent pas être connues exactement pour une variété de raisons. La première raison est due à l'erreur de mesure simple, la deuxième raison plus fondamentale est que quelques données représentent des informations sur le futur (par exemple, demande ou prix de produit d.une future période de temps) et simplement ne peuvent pas être connues avec certitude. Nous discutons quelques manières de tenir compte de cette incertitude et, spécifiquement, pour illustrer comment la programmation stochastique peut être employée pour prendre quelques décisions optimales.

3.2 L'incertitude dans la programmation mathématique

La notion d'incertitude dans la programmation mathématique est apparue pour la première fois dans les années 50 avec les travaux de Dantzig [25], Cooper et Charnes [15], Beale [5]. 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 de clients, les délais de livraison, les temps de traitement, la disponibilité des machines et d'autre 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 crées à partir des historiques, (comme par exemple les ventes des derniers années).

3.2.1 Exemple

Considérons le problème linéaire stochastique suivant :

$$(P)\begin{cases} \min Z = -x_2 \\ s.c: \\ x_1 + x_2 + x_3 = 2 \\ \tilde{a}_{21}x_1 + \tilde{a}_{22}x_2 + x_4 = 2 \\ -1 \leq x_1 \leq 1 \\ x_j \geq 0; j = 2,3,4. \end{cases} \tag{3.1}$$

$\tilde{a}_{21}$,$\tilde{a}_{22}$ sont les coefficients de x_1 et de x_2 respectivement, ils sont supposés pas connu avec certitude, et qu'on connaît leur distribution jointe :

$$(\tilde{a}_{21}, \tilde{a}_{22}) = \begin{cases} (1, \frac{3}{4}) \text{ avec une probabilité } 0.5 \\ (-3, \frac{5}{4}) \text{ avec une probabilité } 0.5. \end{cases}$$

EXPECTED VALUE

Dans ce cas : les espérances $E(\tilde{a}_{21}) = -1$ et $E(\tilde{a}_{22}) = 1$.
En remplacent les coefficients $\tilde{a}_{21}$ et $\tilde{a}_{22}$ par leurs valeurs estimées on trouvera la solution faisable $(x_1, x_2, x_{3,4}) = (0, 2, 0, 0)$.
On veut examiner sa faisabilité sous incertitude : la deuxième contrainte du problème (P) peut être équitablement $x_1 + \frac{3}{4}x_2 + x_4 = 2$ ou $-3x_1 + \frac{5}{4}x_2 + x_4 = 2$.

La solution $(0, 2, 0, 0)$ ne satisfait aucune des contraintes et elle est de ce fait infaisable sous incertitude. Remplacer les variables aléatoires par leurs estimations ne fournit pas toujours une solution faisable à l'égard des variables aléatoires.

Approche d'analyse de scénarios

Cette approche limite le processus de reporter toutes les décisions jusqu'au dernier moment où toutes les incertitudes seront levées. Par conséquence, les problèmes associés à toutes les réalisations possibles des variables aléatoires. Par exemple :

— La solution associée à $(\tilde{a}_{21}, \tilde{a}_{22}) = (1, \frac{3}{4})$ est $(-1, 3, 0, 0.75)$.

— La solution associée à $(\tilde{a}_{21}, \tilde{a}_{22}) = (-3, \frac{5}{4})$ est $(0.11, 1.88, 0, 0)$.

Comme pour l'approche précédente, aucune solution n'est faisable pour la réalisation alternative. Autrement dit, chaque solution a une probabilité 0.5 d'échouer à satisfaite la contrainte.

3.3 Notions Fondamentales

3.3.1 Scénario

Un scénario est une suite d'évènements possibles. Pour fixer les idées, s désignera un scénario. Par ailleurs supposons également que, même si les évènements nont pas encore eu lieu, il faut quand même prendre une décision.

3.3.2 Recours

Le recours est la capacité de prendre la modalité de reprise après qu'un évènement aléatoire a eu lieu.

3.4 Formulation générale

On parle d'un problème de programmation linéaire stochastique (SLP : Stochastic Linear Programming) dans le cas où les contraintes ainsi que la fonction objectif dependent linéairement de x et S un polyèdre convexe, il est définit par :

$$(P) \begin{cases} minZ = C_{\xi}x \\ s.c : \\ T(\xi) = h(\xi) \\ x \in S \end{cases} \tag{3.2}$$

où $S = \{Ax = b; x \geq 0\}$ un polyèdre convexe avec x,A,b des vecteurs déterministes de dimensions $(n_1, 1)$, (m_1, n_1) et $(m_1, 1)$.
C_{ξ}, $T(\xi)$, $h(\xi)$ sont des vecteurs aléatoire de dimensions respectives $(1, n_1)$, (m_2, n_1) et $(m_2, 1)$, définies sur l'éspace de probabilités (Ξ, F, P) ; Les travaux de Sahinidis et Birge [39] présentent l'état de l'art de la programmation stochastique, alors que Willem Haneveld and Maarten [45] se focalise aux 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 Birge et Louveaux[40], Kall et Mayer [41] et Kall et Wallace [42].

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

— Modèle avec des contraintes probabilistes.
— Modèle de recours.

3.5 Modèle avec des contraintes probabilistes

Des problèmes dont la modélisation appartient à la catégorie des modèles probabilistes sont apparus pour la première fois dans Charnes [14] sous le nom de programmation sous contraintes probabilistes (**"chance constrained programming"**). L'ensemble réalisable du problème (3.3) peut être aussi définit des contraintres définissant des évènements qui doivent être réalisables avec une certaine probabilité noté $\alpha \in [0,1]$(chaque contrainte est considérée comme étant un évènement).
Sous cette supposition, le problème déterministe équivalent de (3.3) peut être définit comme suit [42] :

$$(PD)\begin{cases} minE[C^T(\xi)x] \\ s.c: \\ P(\{\xi \ tq \ T_i(\xi)_i(\xi) \geq \alpha_i\} \\ x \in D \end{cases}$$

α étant une probabilité réglée par l'utilisateur qui reflète par exemple la fiabilité d'un système ou une tolérence acceptable.
T_i,h_i représentant la ligne i de la matrice T,h resp et α_i la probabilité que cette contrainte soit satisfaite.

3.6 Modèle de Recours

La programmation stochastique s'intéresse essentiellement à des problèmes où on doit prendre une décision sur le champ sous présence d'incertitude, sans attendre la réalisation des certaines variables aléatoires ; on fait souvent appel à ce type de problèmes sous le nom de "here and now problems".
Dans la famille "here and now" on est obligé de déterminer au moment t_0 certaines valeurs x_j^0 qui ne peuvent plus changer dans la suite, quoi qu'il arrive aux instants ultérieurs. Il se peut qu'une décision à l'instant initial t_0 coûte trop cher dans le futur, cet effet faisant émerger la nécéssité d'une approche systématique de la prise de telles décision.
L'idée fondamentale de la programmation linéaire stochastique est le concept du recours. Dans la suite on étudiera plus particulièrement la programmation stochastique linéaire avec recours. L'idée est d'optimiser les décisions que l'on doit prendre au moment t_0 tout en tenant compte des conséquences de ces décisions pour toute réalisation des aléas qui pourrait se produire aux instants ultérieurs. Le mot "recours" révèle la possibilité dont on dispose de remédier à une décision éventuellement trop optimiste à l'instant t_0, en mettant en oeuvre des actions correctives, évidemment plus chères, qui satisfont pourtant les contraintes posées aux instants ultérieurs t_i, $i > 0$. On précise qu'après avoir pris la décision au moment t_0, on passe au moment t_1 où tous les évènements inconnus jusqu'à cet instant se révèlent. Par conséquent, l'étude de tels cas a un intérêt seulement si on dispose

de quelques informations sur les réalisations des évènements qui nous sont inconnus au moment t_0 (information comme par exemple la distribution de probabilité, des scénarios, la moyenne, la variance etc).

3.6.1 Modèles de recours à 2-niveaux(en 2-étapes)

Les variables de la première étape sont considérées comme étant des variables structurantes, stratégiques qui ne peuvent plus changer à la seconde étape. La décision à la première étape doit être prise en respectant des contraintes propres à cette étape. La seconde étape comprend des actions (pénalisantes) qui peuvent être entreprises afin de satisfaire des contraintes qui font intervenir des variables de la première étape.
Toutes ces idées sont résumées dans le programme suivant :

$$(PS)\begin{cases} min\ C^T x + \psi(x) \\ s.c : \\ Ax = b, x \geq 0 \\ avec\ \psi(x) = E_\xi[Q(x,\xi)] \\ et\ Q(x,\xi^j) = min\ \{q^t(\xi)^T y\} \quad (problème\ de\ seconde\ étape) \\ s.c : \\ W(\xi)y = h(\xi) - T(\xi)x, y \geq 0 \end{cases}$$

Notons que

A	Matrice déterministe ; $A \in \mathbb{R}^{m_1 \times n_1}$
b	Vecteur déterministe ; $b \in \mathbb{R}^{m_1}$
c	Vecteur déterministe ; $c \in \mathbb{R}^{n_1}$
x	Les variables x représentent les décisions de la première étape qui doivent être prises avant que toute réalisation de la variable aléatoire ξ ne soit connu.
y	Les variables y représentent les décisions de la deuxième étape, prises une fois que les réalisations de la variable aléatoire ξ seront observées.
Fonction objectif	On cherche à minimiser sur les variables de la première étape x , mais on veille en même temps à ce que les coûts au deuxième étape, après la réalisation de la variable aléatoire ξ, soient minimaux en moyenne.
$Q(x, \xi)$	Fonction de recours qui représente le coût minimum pour corriger la décision de la première étape et satisfaire les contraintes.
$\psi(x)$	Espérance de la fonction de recours.
$q(\xi) = q + \sum_i q_i \xi_i$	Vecteur stochastique des coûts unitaires de pénalités ; $q(\xi) \in \mathbb{R}^{n_2}$; q, q_i : vecteurs déterministes.
$W(\xi)$	Matrice de recours stochastique ; $W \in \mathbb{R}^{m_2 \times n_2}$
$h(\xi) - T(\xi)x$ Où : $h(\xi) = h + \sum_i h_i \xi_i$ $T(\xi) = T + \sum_i T_i \xi_i$	Mesurela violation des contraintes. $T(\xi) \in \mathbb{R}^{m_2 \times n_1}$ $h(\xi) \in \mathbb{R}^{m_2}$ h,h_i, T, T_i :déterministes.

Dans de nombreuses applications on pourra supposer qu'il existe une mesure de probabilité P tel que $P\{\xi = \xi_s\} = p_s$ avec $\xi_1, \xi_2, ..., \xi_s \in \mathbb{R}^r$.
L'ensemble des réalisations possibles est fini, et les réalisations ξ_i s'appellent souvent des "scénarios", dont p est la probabilité d'occurrence. Cette hypothèse va nous permettre de passer de (3.8)au programme suivant :

$$(P) \begin{cases} minC^T x + p^1 q^1 y^1 + p^2 q^2 y^2 + \ldots + p^s q^s y^s \\ s.c \\ Ax = b \\ T^1 x + W^1 y^1 = h^1 \\ T^2 x + W^2 y^2 = h^2 \\ \vdots \ddots \\ T^s x + W^s y^s = h^s \\ x \geq 0, y^1, y^2, ..., y^s \geq 0 \end{cases}$$

Où les indices $1, 2, ...s$ font référence aux scénarios correspondants.

Exemple introductif à la programmation linéaire stochastique avec recours (Planification de production)[42]

Problématique :

Il s'agit de minimiser les coûts de production tout en satisfaisant les demandes des clients :

- **Le cas le plus trivial (irréaliste) :** un seul produit, sans contraintes techniques :

$$(Pc)\begin{cases} mincx \\ s.c: \\ x \geq w \\ x \geq 0 \end{cases}$$

  où :
  x :production
  w :demande

- **Modèle plus réaliste :** plusieurs produits, contraintes sur la productivité et sur la capacité de la production.
  avec :

  * b : capacité de production (quantité maximale des matières premières qui peut être traitée).
  * γ : Coût de production.
  * h : demandes des clients

  Le programme linéaire suivant décrit le problème de production :

$$\begin{cases} min(2x_{mat1} + 3x_{mat2}) \\ s.c: \\ x_{mat1} + x_{mat2} \leq 100 \\ 2x_{mat1} + 6x_{mat2} \geq 180 \\ 3x_{mat1} + 3x_{mat2} \geq 162 \\ x_{mat1} \geq 0 \\ x_{mat2} \geq 0 \end{cases}$$

- Pratiquement, il peut arriver que les productivités ou/et les demandes varient aléatoirement, et qu'on doit prendre notre décision **"plan de production"** avant de connaitre leurs valeurs exactes.

Supposons que :

— Un plan de production est à décider chaque semaine et ne peut être changé durant la semaine.

— Les productivités $\pi(mat1, prod1)$ et $\pi(mat2, prod2)$ ainsi que les demandes sont aléatoires (les autres productivités sont déterministe mat2/prod1 et mat1/prod2), où :

$$\begin{cases} h_{prod1} = 180 + \tilde{\xi}_1 \\ h_{prod2} = 162 + \tilde{\xi}_2 \\ \pi(mat1, prod1) = 2 + \tilde{\eta}_1 \\ \pi(mat2, prod2) = 3.4 + \tilde{\eta}_2 \end{cases}$$

Et que les réalisations des 4 Variables Aléatoires sont comme suit :

$$\begin{cases} \xi_1 \in [-30.91, 30.91] \\ \xi_2 \in [-23.18, 23.18] \\ \eta_1 \in [-0.8, 0.8] \\ \eta_2 \in [0.0, 1.84] \end{cases}$$

De ce fait, on aura le programme stochastique suivant :

$$\begin{cases} min(2x_{mat1} + 3x_{mat2}) \\ s.c : \\ x_{mat1} + x_{mat2} \leq 100 \\ (2 + \tilde{\eta}_1)x_{mat1} + 6x_{mat2} \geq 180 + \tilde{\xi}_1 \\ 3x_{mat1} + (3.4 - \tilde{\eta}_2)x_{mat2} \geq 162 + \tilde{\xi}_2 \\ x_{mat1} \geq 0 \\ x_{mat2} \geq 0 \end{cases}$$

En fin, ce problème de décision n'est pas bien défini, car ce n'est pas encore claire le sens de "min" quand les réalisations des variables aléatoires ne sont pas connu ! **Géométriquement**

— Le changement dans les productivités correspond à des translations parallèles des facettes de l'ensemble des solutions admissibles.

— Le changement dans les demandes résulte en des rotations des facettes correspondantes.

— Les changements dans les productivités et les demandes résultent en une superposition de deux motions géométriques

On parle de la résolution graphique de ce problème stochastique

Les clients considèrent que leurs demandes seront satisfaites durant la même semaine. Pour cela, considérons que l'usine a établit un arrangement avec ses clients qui consiste à satisfaire les demandes des clients en cas de manque, en achetant la quantité manquante directement du marché. Cela va certainement causer des coûts supplémentaires à l'usine.

On considère les couts unitaires suivants :

$q_{prod1}7, q_{prod2} = 12$

Dans ce cas, l'objectif est de trouver un plan de production qui minimise la somme des coûts du premier stage (i.e. de production) et de l'estimation des coûts de recours (i.e. coût des produits achetés du marché en cas de manque).

Pour formaliser cette approche, nous résumons notre notation :

— Un seul vecteur aléatoire :$\tilde{\xi} = (\tilde{\xi}_1, \tilde{\xi}_2, \tilde{\eta}_1, \tilde{\eta}_2)$
— Pour chaque contrainte stochastique on introduit une variable $Y_i(\tilde{\xi})$ qui mesure le manque dans la production correspondante.

Le programme stochastique avec recours de notre exemple est le suivant :

$$\begin{cases} min 2x_{mat1} + 3x_{mat2} + E_{\tilde{\xi}}[7y_1(\tilde{\xi}) + 12y_2(\tilde{\xi})] \\ s.c: \\ x_{mat1} + x_{mat2} \leq 100 \\ \alpha(\tilde{\xi})x_{mat1} + 6x_{mat2} + y_1(\tilde{\xi}) \geq h_1(\tilde{\xi}) \\ 3x_{mat1} + \beta(\tilde{\xi})x_{mat2} + y_2(\tilde{\xi}) \geq h_2(\tilde{\xi}) \\ x_{mat1} \geq 0 \\ x_{mat2} \geq 0 \\ y_1(\tilde{\xi}) \geq 0 \\ y_2(\tilde{\xi}) \geq 0 \end{cases}$$

Où :

$h_1(\tilde{\xi}) = h_{prod1} = 180 + \tilde{\xi}_1$ $h_2(\tilde{\xi}) = h_{prod2} = 162 + \tilde{\xi}_2$ $\alpha(\tilde{\xi}) = \pi(mat1, prod1) = 2 + \tilde{\eta}_1$ $\beta(\tilde{\xi}) = \pi(mat2, prod2) = 3.4 - \tilde{\eta}_2$

3.7 Différents cas de recours

En observant la formulation (3.8) on peut distinguer différents cas de recours en fonction des types de (q, y, W) ou plus généralement des $q(\xi)$, $y(\xi)$, et $W(\xi)$. Les cas qui méritent une étude plus spéciale et qui sont cités dans la littérature sont le recours fixe, le recours complet et le recours simple.

3.7.1 Recours fixe

Définition 19

Le recours est dit fixe $W(\xi) = W$ (i.e déterministe donc indépendant de ξ).

3.7.2 Recours complet

Définition 20

Le recours fixe est complet si la matrice de recours W de dimension $m \times n$ satisfait :

$$\{t|t = Wy, y \geq 0\} = \mathbb{R}^m$$

Remarque
$\{t|t = Wy, y \geq 0\} = \mathbb{R}^m \Rightarrow \forall x \in \forall \mathbb{R}^{n_1}\ \forall \xi \in \Xi$: le problème de seconde étape est faisable

3.7.3 Recours relativement complet

Définition 21

Si le problème de seconde étape est faisable pour tout $x \in K_1 = \{x \in \mathbb{R}_+^{n_1} | Ax = b\}$ alors le problème (PS) est un problème à recours relativement complet.

3.7.4 Recours simple

Définition 22

Le recours fixe est simple si sa forme canonique est $celle - ci$:

$$\begin{pmatrix} q \\ W \end{pmatrix} = \begin{pmatrix} q^+ & q^- \\ I & -I \end{pmatrix}$$

Où ; q^+, q^- sont les $1 \times n_2$ vecteurs de pénalités et I est la $n_2 \times n_2$ matrice identité.

3.7.5 Modèles de recours à niveaux multiples

Cette approche modélise les situations où les décisions à prendre sont déterminées périodiquement à base de certaines réalisations de certaines variables aléatoires. Chaque niveau (le premier n'y pas compris) correspond au moment où certaines informations sont

disponibles et on doit prendre une décision (sur la période qui suit).
Un problème linéaire à $K - niveaux$ avec recours peut être écrit sous la forme :

$$(P_{rm}) \begin{cases} s.c: \quad Ax_1 = b \\ avec \;\; T_2x_1 + W_2x_2 = h_2 \\ \qquad \vdots \\ \qquad T_Kx_{K-1} + W_Kx_K = h_k \\ \qquad x_1, x_2, ..., x_K \geq 0 \end{cases}$$

Donc là, au lieu des deux décisions x et y qu'on doit prendre aux niveaux 1 et 2, on est face à K décisions séquentielles $x_1, x_2, ..., x_K$, qui correspondent aux niveaux 1,2,...,K.
Au niveau $(\tau(2 \leq \tau \leq K)$, les réalisations des variables aléatoires $(\xi_1, \xi_2, ..., \xi_K)$ sont disponibles ainsi que les décisions $x_1, x_2, ..., x_{-1}$ et on doit se décideur sur x d'une façon que les contraintes à ce niveau soient satisfaites.

3.8 La méthode L-Shaped

3.8.1 Principe

Basée sur la décomposition de Benders (1962)[4], la méthode L-shaped est une méthode permettant de résoudre des problèmes stochastiques à deux étapes, dans le cas où ξ est une variable aléatoire discréte avec $|\Xi| < \infty$ et lorsqu'il n'y a pas de contrainte d'intégrité. Elle a été introduite par Wets (1966) [71] et Van Slyke (1969)[70] .
Même si elle n'est pas applicable en l'état pour notre problème (qui est en nombres entiers), son importance théorique fait que nous la présentons ici.
Puisque nous supposons que $|\Xi|$ est fini, nous allons indicer par $k \in 1, ..., K$ chacune de ses réalisations, de probabilité respective p_k. Nous notons (q_k, T_k, h_k) les paramètres correspondant à la réalisation k, et y_k le recours associé.
Dans l'algorithme $L - Shaped$ deux classes de problèmes sont crées : le problème maître et les problèmes esclaves.

- Les problèmes esclaves (un pour chaque scénario) reçoivent comme entrée les valeurs des variables du premier niveau et calculent les variables de recours propres à chaque scénario ainsi que le coût associé.
- Le problème maître comporte les variables du premier niveau et une description approximée de la fonction de recours ; il est mis initialement sous la forme :

$$(M)\begin{cases} min\ Z = c^T x + \theta & (1) \\ s.c \\ Ax = b \\ D_l x \geq d_l \quad l = 1, ..., r & (2) \\ E_l + \theta \geq e_l \quad l = 1, ..., s & (3) \\ \\ x \geq 0, \theta \in \mathbb{R} \end{cases}$$

Où :
θ est la variable qui donne l'approximation de la fonction réelle de recours.
L'équation (2) : est une coupe de faisabilité
L'équation (3) : est une coupe d'optimalité
Au départ on résout le problème maître (M) on récupère la solution (x^v, θ^v) et on fait passer cette solution à chaque $sous - problème$:

$$\textbf{Primal}\begin{cases} min\ w = q_k^T(\xi)y \\ s.c : \\ Wy \geq h_k(\xi) - T_k(\xi)x \\ y \geq 0 \end{cases} \qquad \textbf{Dual}\begin{cases} max\ \pi_k^{vT}(h_k(\xi) - T_k(\xi)x) \\ s.c : \\ \pi_k^v W \leq q_k^T(\xi) \end{cases}$$

Le but est de construire la fonction de recours à partir des solutions proposées au fur et à mesure par le problème maître et en ajoutant des coupes déduites de la résolution des problèmes esclaves. Plus précisément :

- La résolution des problèmes esclaves nous permet de définir une borne supérieure de la fonction objectif.
- La résolution du problème maître nous permet de définir une borne inférieure de la fonction objectif. Quand les deux bornes sont suffisamment proches on s'arrête et on retient la solution optimale x^* correspondante.

3.8.2 Description de l'algorithme L-shaped

Etape 0 :
initialiser s=r=v=0 , $\theta^0 = -\infty$;
Etape 1 :
Si s=0 (n'existe pas une coupe de faisabilité) retirer θ de (M)
Trouver la solution de problème maître (M), soit (x^v, θ^v) une solution optimale.
Etape 2 :
Si $x \notin K_2$ ajouter une coupe de faisabilité (2) tel que :

$$K_2 = \{x | \exists y : Wy = h_k - T_k x, y \geq 0, k = 1, ..., K\}$$

on calcule D_{r+1}, d_{r+1} :

$$D_{r+1} = (\sigma^v)^T.T_k$$
$$d_{r+1} = (\sigma^v)^T.h_k$$

et aller à l'étape 1
r=r+1 ;
Sinon aller à l'étape 3.

Etape 3 :
Calculer E_{s+1}, e_{s+1} où :

$$E_{s+1} = \sum_{k=1}^{K} p_k(\pi_k^v)^T.T_k$$
$$e_{s+1} = \sum_{k=1}^{K} p_k(\pi_k^v)^T.h_k$$

Soit $w^v = e_{s+1} - E_{s+1}x^v$
Si $\theta^v \geq w^v$
Terminer avec x^v une solution optimale.
sinon
v=v+1 ;
s=s+1 ;
et ajouter une coupe d'optimalité (3) et aller à l'étape 1 ;
Fin de l'algorithme.

Exemple illustratif[3]

$$\begin{cases} Z = min\ 100x_1 + 150x_2 + E(q_1y_1 + q_2y_2) \\ s.c: \\ x_1 + x_2 \leq 120 \\ 6y_1 + 10y_2 \leq 60x_1 \\ 8y_1 + 5y_2 \leq 80x_2 \\ y_1 \leq d_1, y_2 \leq d_2 \\ x_1 \geq 40, x_2 \geq 20 \\ y_1, y_2 \geq 0 \end{cases}$$

Avec :
$c = (100, 150)\,;\ A = \begin{pmatrix} 1 & 1 \\ -1 & 0 \\ 0 & -1 \end{pmatrix}; b = \begin{pmatrix} 120 \\ -40 \\ -20 \end{pmatrix}$

$\xi = (d_1, d_2, q_1, q_2) = \begin{cases} (500, 100, -24, -28); & p_1 = 0.4 \\ (300, 300, -28, -32); & p_2 = 0.6 \end{cases}$;

W=$\begin{pmatrix} 6 & 10 \\ 8 & 5 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}h_1 = (0, 0, 500, 100)^T$; $h_2 = (0, 0, 300, 300)^T$

$T = \begin{pmatrix} -60 & 0 \\ 0 & -80 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$ tel que : $T_{.,1} = (-60, 0, 0, 0)^T$; $T_{.,1} = (0, -80, 0, 0)^T$

Iteration 1 :

Etape 0

Initialisation s=v=r=0 ;

Etape 1

Résoudre le problème maître suivant :

$min \ \ \{100x_1 + 150x_2 | x_1 + x_2 \leq 120, x_1 \geq 40, x_2 \geq 20\}$

$x^1 = (40, 20)^T$, $\theta^1 = -\infty$

Etape 2

$x^1 \in K_2$ tel que : $K_2 = \{x | \exists y : Wy = h_k - T_k x^1, y \geq 0, k = 1, ..., K\}$

Etape 3

Pour $\xi = \xi_1$

Résoudre le problème suivant :

$min\{-24y_1 - 28y_2 | 6y_1 + 10y_2 \leq 2400, 8y_1 + 5y_2 \leq 1600, 0 \leq y_1 \leq 500, 0 \leq y_1 \leq 100\}$

$y^T = (137.5, 100)$

$w_1 = (-24) * (137.5) - (28) * (100) = -6100$

résoudre le dual por trouver π_1

$max \ \ \{2400\pi_1^1 + 1600\pi_1^2 + 500\pi_1^3 + 100\pi_1^4 | 6\pi_1^1 + 10\pi_1^2 + \pi_1^3 \leq -24, 8\pi_1^1 + 5\pi_1^2 + \pi_1^4 \leq -28\}$

$\pi_1^T = (0, -3, 0, -13)$

Pour $\xi = \xi_2$ résoudre le problème :

$min\{-28y_1 - 32y_2 | 6y_1 + 10y_2 \leq 2400, 8y_1 + 5y_2 \leq 1600, 0 \leq y_1 \leq 300, 0 \leq y_1 \leq 300\}$

$y^T = (80, 192)$

$w_2 = (-28) * (80) - (32) * (192) = -8384$

résooudre le dual por trouver π_1

$max \ \ \{2400\pi_1^1 + 1600\pi_1^2 + 300\pi_1^3 + 300\pi_1^4 | 6\pi_1^1 + 10\pi_1^2 + \pi_1^3 \leq -28, 8\pi_1^1 + 5\pi_1^2 + \pi_1^4 \leq -32\}$

$\pi_2^T = (-2.32, -1.76, 0, 0)$

$e_1 = 0.4 * \pi_1^T * h_1 + 0.6 * \pi_2^T * h_2$

$e_1 = 0.4 * (0, -3, 0, -13)^T * (0, 0, 500, 100) + 0.6 * (-2.32, -1.76, 0, 0)^T * (0, 0, 300, 300)$

$e_1 = 0.4 * (-1300) + 0.6 * (0) = -520$

$E_1 = 0.4 * \pi_1^T * T + 0.6 * \pi_2^T * T$

$E_1 = 0.4 * (0, -3, 0, -13)^T * \begin{pmatrix} -60 & 0 \\ 0 & -80 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} + 0.6 * (-2.32, -1.76, 0, 0) * \begin{pmatrix} -60 & 0 \\ 0 & -80 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$

E_1 =0.4(0,-240)+0.6(139.2,140.8)=(83.52,180.48)

w$^1 = e_1 - E_1 * x^1 = -520 - (83.52, 180.48) * \begin{pmatrix} 40 \\ 20 \end{pmatrix} = -7470.4$

On a : $w^1 - 7470.4 > \theta^1 = -\infty$

On ajoute la coupe d'optimalité au problème maître

$E_1 + \theta \geq e_1$

$83.52x_1 + 150x_2 + \theta \geq -520$

Iteration 2 :

Etape 1

Résoudre le problème maître suivant :

$min \quad \{100x_1 + 150x_2 + \theta | x_1 + x_2 \leq 120, x_1 \geq 40, x_2 \geq 20, 83.52x_1 + 150x_2 + \theta \geq -520\}$

$x^2 = (40, 80)^T$, $Z = -2299.2$, $\theta^2 = -182992$

Etape 3

on ajoute la coupe d'optimalité suivante :

$211.2x_1 + \theta \geq -1584$

Iteration 3 :

Etape 1

Résoudre le problème maître suivant :

$min \quad \{100x_1 + 150x_2 + \theta | x_1 + x_2 \leq 120, x_1 \geq 40, x_2 \geq 20, 83.52x_1 + 150x_2 + \theta \geq -520, 211.2x_1 + \theta \geq -1584\}$

$x^3 = (66.28, 63.172)^T$, $Z = -1039.375$, $\theta^3 = -15697.994$

Etape 3

on ajoute la coupe d'optimalité suivante :

$115.2x_1 + 96x_2 + \theta \geq -2104$

Iteration 4 :

Résoudre le problème maître suivant :

$min \quad \{100x_1 + 150x_2 + \theta | x_1 + x_2 \leq 120, x_1 \geq 40, x_2 \geq 20, 83.52x_1 + 150x_2 + \theta \geq -520, 211.2x_1 + \theta \geq -1584, 115.2x_1 + 96x_2 + \theta \geq -2104\}$

$x^4 = (40, 33.75)^T$, $Z = -889.5$, $\theta^4 = -9952$

Etape 3

on ajoute la coupe d'optimalité suivante :

$133.44x_1 + 130.56x_2 + \theta \geq 0$

Iteration 5 :

Etape 1

Résoudre le problème maître suivant :

$min \quad \{100x_1 + 150x_2 + \theta | x_1 + x_2 \leq 120, x_1 \geq 40, x_2 \geq 20, 83.52x_1 + 150x_2 + \theta \geq -520, 211.2x_1 + \theta \geq -1584, 115.2x_1 + 96x_2 + \theta \geq -2104, 133.44x_1 + 130.56x_2 + \theta \geq 0\}$

$x^5 = (46.667, 36.25)^T$, $Z = -855.833$, $\theta^5 = -10960$

Etape 3

$w_5 = -520 - (83.52, 180.48) * x^5 = -10960 = \theta^5$

Terminer

$x = x^5 = x^5 = (46.667, 36.25)^T$ est la solution optimale.

Chapitre 4

Optimisation linéaire stochastique multi-objectif

4.1 Introduction

Le bute de ce chapitre est des présenter une nouvelle procédure. Pour la résolution du programme de la programmation linéaire stochastique en nombres entiers à objectifs multiples, devloppée par M.Moulaï et S.Amrouche [53], basée sur la méthodes à 2 niveaux avec recours entier et la méthode de séparation et évaluation détaillée dans les chapitres précédents.

4.2 Définitions et notations

Considérons le problème linéaire stochastique $multi - objectif$ en nombres entiers (MOSILP) donné sous la forme mathématique suivante :

$$(MOSILP) \begin{cases} minZ = C_K(\xi)x \quad K = 1, ..., p \\ s.c : T(\xi)x = h(\xi) \\ x \in D \end{cases}$$

Où $D = S \bigcap z^n$ avec $C_k(\xi)$, $T(\xi)$, $h(\xi)$ sont des vecteurs aléatoires de dimensions respectives $(1 \times n_1)$,$(m_2 \times n_1)$,$(m_2 \times 1)$, définis sur l'espace de probabilités (Ξ, F, P), $D = \{Ax = b : x \geq 0 \quad entiers\}$, un polyèdre convexe et déterministe des décisions x, A et b sont des vecteurs déterministes de dimensions $(m_1 \times n_1)$,$(m_1 \times 1)$ respectivement.
La plupart des méthodes de résolution des MOSILP transforment d'abord le problème (4.1) en un problème déterministe et puis le résoudre par une méthode interactive, dans ce contexte, nous citons la méthode PROTRADE,STRANGE [66] et PROMISE [69].

4.2.1 Le problème "Déterministe Equivalent"

Supposons que nous avons une distribution discrète et finie { $(\xi^s; p_r)$; s = 1,...,S} des données aléatoires. Dans le premier pas, pour chaque réalisation ξ^s de ξ nous associons une fonction objective $Z_{ks} = C_k(\xi^s)x$; une matrice $T(\xi^s)$ et un vecteur $h(\xi^s)$ en prenant en considération les scénarios différents qui affectent les K objectifs et les contraintes stochastiques.
Le deuxième pas est de revenir lidée du recours utilisée dans la programmation stochastique $mono - objectif$. nous supposons que le décideur peut indiquer d'une manière satisfaisante les pénalités $q^s = q(\xi^s)$ des variables violentes z_s ; s = 1,...,S des contraintes et la dimension du problème déterministe associé reste raisonnable. Contrairement la méthode Strange où une fonction objective supplémentaire est crée pour pénaliser les contraintes violentes, la fonction du recours $Q(x, \xi^s)$ est ajoutée chaque fonction objective. Cette pénalité est donnée par :

$$\text{Q(x,}\xi^s) = min\{(q^s)z/W(\xi^s)z = h(\xi^s) - T(\xi^s)x; z \geq 0\} \qquad ((1))$$

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

$$\widetilde{Z_k} = E[Z_k + Q(x, \xi)], k = 1, ..., K$$

Nous résultons le problème déterministe suivant :

$$(P_2)\begin{cases} min\widetilde{Z_k} = Z'_k + Q(x), \quad K = 1, ..., p \\ s.c: \\ Ax = b \\ x \geq 0, xentier \end{cases}$$

Où :

$$Z'_k = E[Z_k] = \sum_{s=1}^{S} p^s Z_{ks} = \sum_{s=1}^{S} p^s C_k x = E[C_k(\xi)x]$$

et

$$Q(x) = E[Q(x,\xi)] = \sum_{s=1}^{S} p^s C_k(x,\xi^s) = \sum_{s=1}^{S} p^s (q_s)^T z^s$$

sont respectivement les valeurs d'espérance de Z_k et de la fonction du recours $Q(x,\xi)$. Nous attendons le deuxième étage $Q(x,\xi^s)$ de programme pour être réalisable pour toute réalisation de s ; s = 1,...,S de ξ selon $(m_0 \times n_0) - matrice\ de\ recours\ W(\xi^s)$; ceci ne doit pas être vrai pour toutes les décisions $x \in \{x \in \mathbb{R}/Ax = b; x \geq 0\}$ de premier étage.
Un cas spécial de matrice du recours fixe et complète est du recours simple avec la matrice didentité I d'ordre m_0 ; $W = (I; -I)$

4.2.2 Réalisabilité

Supposons que la matrice du recours W est fixé. Nous clarions la question de comment décider si une donnée $x = x^0$ est réalisable pour le deuxième étage de problème pour toutes les réalisations possibles de ξ. Alors, il est plus simple de travailler avec le dual de programme de deuxième étage.

$$max\{\pi^T[h(\xi^s) - T(\xi^s)x^0]/\pi^T W \leq (q^s)^T\} \qquad ((2))$$

L'ensemble des contraintes $P = \{\pi/\pi^T W \leq (q^s)^T\}$ est indépendante de x.
Soit $\{\pi^T/t \in I\}$ l'ensemble des points extrêmes de P et $\{\sigma^\delta/\delta \in \Delta\}$ l'ensemble des arêtes extrêmes.

1. Si $P = \varnothing$ alors $Q(x^0,\xi^s)$ est illimité ($Q(x^0,\xi^s) = -\infty$) ou non réalisable ($Q(x^0,\xi^s) = +\infty$)
2. Si $P \neq \varnothing$ alors $Q(x^0,\xi^s)$ 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 réalisabilité d'un système des contraints linéaires et peut être énoncé comme la proposition suivante :

Proposition 3

l'ensemble

$$\{z/Wz = h(\xi^s) - T(\xi^s)x^0, z \geq 0\} \neq \varnothing$$

si et seulement si

$$\sigma^T W \leq 0 \Longrightarrow \sigma^T[h(\xi^s) - T(\xi^s)x^0] \leq 0$$

Nous concluons que $Q(x^0, \xi^s)$ est non réalisable si et seulement si P admet une arête extrême σ telle que : $\sigma^T[h(\xi^s) - T(\xi^s)x^0] > 0$; Autrement la valeur optimale de $Q(x^0, \xi^s)$ est obtenu par $\pi^T[h(\xi^s) - T(\xi^s)x^0]$, où π est point extrême de P. Puis nous vérifions la réalisabilité pour le problème du deuxième étage, nous devons trouver une direction σ en résultant le programme suivant :

$$max\{\sigma^T[h(\xi^s) - T(\xi^s)x^0]/\sigma^T W \leq 0, \|\sigma\|_1 \leq 1\} \qquad ((3))$$

Tel que la dernière contrainte est ajoutée pour limitée (bornée)σ ; si non la valeur maximale sera $(+\infty)$ et nous ne l'intéresse pas.
Si pour un certain ξ^s, $s \in \{1, ..., S\}$, $\sigma_s^T[h(\xi^s) - T(\xi^s)x^0] > 0$, où σ_s est la solution optimale de (3), nous avons trouvé ξ^s pour lequel x^0 n'est pas une solution réalisable de problème du dexième étage. Dans ce $cas - ci$, nous ajoutons au problème (P_2) **la coupe de réalisabilité**

$$\sigma_s^T[h(\xi^s) - T(\xi^s)x^0] \leq 0 \qquad ((4))$$

4.2.3 Efficacité

Supposons que toutes les coupes de réalisabilité déterminées sont rajoutées au problème (P_2), nous pouvons le reformulé en présentant une nouvelle variable tel que nous résultons le problème suivant :

$$(P_3)\begin{cases} min\ \tilde{Z}_k = Z'_k + \theta, \quad k = 1, ..., K \\ s.c : \\ x \in S_3 \\ \theta \geq Q(x), \\ x\ entier \end{cases}$$

avec :

$$S_3 = \{x \in R^n/Ax = b, \sigma_s^T h(\xi^s)x \geq \sigma_s^T T(\xi^s), \in \{1, ..., S\}, x \geq 0\}$$

S_3 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 (P_3) qui sont efficaces dans sens de la définition suivante :

Définition 23

Une solution $x^0 \in S_3$ est une solution efficace pour (P_3) s'il n'existe pas de solution $x^1 \in S_3$ tel que $\tilde{Z}_i(x^1) \leq \tilde{Z}_i(x^0)$, $i \in \{1, ..., K\}$ et $\tilde{Z}_i(x^1) < \tilde{Z}_i(x^0)$ pour au moins une valeur de $i \in \{1, ..., K\}$ et pour toute réalisation ξ^s, s=1,...,S.

En mettant en considération le programme $mono - objectif$ (P_3), la première étape est de déterminer une solution réalisable entière de ce problème.

$$(P_4) \begin{cases} min \ \sum_{k=1}^{K} \tilde{Z}_k = \sum_{k=1}^{K} Z'_k + K\theta \\ s.c : \\ x \in S_3 \\ \theta \geq Q(x), \\ x \ entier \end{cases}$$

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 (P_4) sans $\theta \geq Q(x)$ et obtenons une solution faisable (x^0,θ^0) (initialiser :$\theta = -\infty$).
Les solutions optimales (π_s ; s = 1,. . .,S) de dual(2) sont utilisées pour calculer l'esperance de la fonction du recours Q(x^0) donnée par :

$$Q(x^0) = \sum_{s=1}^{S} p^s Q(x^0, \xi^s) = \sum_{s=1}^{S} p^s (\pi_s)^T [h(\xi^s) - T(\xi^s)x^0]$$

Si $\theta^0 \geq Q(x^0)$, x^0 est optimale pour (P_4), sinon **la coupe d'efficacité**

$$\theta \geq \sum_{s=1}^{S} p^s (\pi_s)^T [h(\xi^s) - T(\xi^s)x^0] \qquad ((5))$$

est ajoutée au (P_4) qui est optimiser.

4.3 Méthode de résolution d'un problème MOSILP

[53]

4.3.1 Organigramme de l'algorithme

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

— Principe 2 – *étges* avec recours entier ;

— Principe de séparation et évaluation en utilisant le concept d'ensembles dominés.

Désignons par :

Eff :ensemble des solutions entières et efficaces de (P_3).

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

F : une liste contenant les $sous - problèmes$ non sondés.

e, α : entiers positifs.

Procédure d'initialisation

1. Pour q=1,...,K appliquer la première phase de l'algorithme simplexe pour résoudre le programme linéaire :

$$\begin{cases} min\ \varphi_q = \sum_{j=1}^{n} v_j \\ s.c : \\ \sum_{i=1_{i\neq q}}^{k} c_{ij}y_i + v_j = c_{qj}, \quad j = 1, ..., n \\ y_i \geq 0, \quad i = 1, ..., k \end{cases}$$

1.1 Si $\varphi_q = 0$, supprimer la fonction objective Z_q du problème (P_1), **faire** $Z_t = Z_{t+1}, t = q, ..., (k-1)$ et $k = k - 1$.

2. Faire k'=k ;
3. Considérer le problème suivant :

$$\begin{cases} min\ \ \sum_{i=1}^{k'} E[C_i(\xi)x] \\ s.c : \\ Ax = b \\ \sigma_r^T[h(\xi^r) - T(\xi^r)x] \leq 0, \quad r = 1, ..., \alpha \\ \theta^e \geq Q(x) \\ x \geq 0, xentier. \end{cases}$$

3.1 Initialiser : $\theta = -\infty$,$e = \alpha = 0$,
$F = \{P_0\}$, $Eff = \varnothing$ et $\widetilde{Eff} = \varnothing$;
3.2 Aller à la procédure de choix ;

Procédure de choix

1. Si $F = \varnothing$, afficher Eff, $\widetilde{Eff}$, terminer ;
2. Sinon, sélectionner le problème (P_e) de plus fort indice e dans la liste F, aller à la procédure d'évaluation ;

Procédure d'évaluation

1. Appliquer la méthode duale fractionnaire en utilisant la coupe de Gomory si nécéssaire pour trouver une solution optimale entière x^e pour le problème (P_e) sans aucune coupe d'optimalité ou de réalisabilité.

(minimiser la fonction objective $\sum_{i=1}^{k'} E[C_i(\xi)x]$ sous les contraintes déterministes).
1.1 S'il n'existe pas une telle solution x^e, faire $F = F$
$(P_e)\}$, aller la procédure de choix ;
1.1 Sinon, aller la procédure de réalisation ;

Procédure de réalisabilition

1. Résoudre le problème (3)

$$max\{\sigma^T[h(\xi^s) - T(\xi^s)x^e]/\sigma^T W \leq 0, \|\sigma\|_1 \leq 1\}, s \in \{1, ..., S\}$$

1.1 Si $\sigma_s^T[h(\xi^s) - T(\xi^s)x^e] > 0$, rajouter au problème (P_e) la coupe de réalisabilité (4)

$\sigma_s^T[h(\xi^s) - T(\xi^s)x] \leq 0$

1.2 $\sigma_s^T[h(\xi^s) - T(\xi^s)x] \leq 0$, aller à la procédure d'optimisation ;

Procédure d'efficacité

1. Résoudre le problème (2)

$$max\{\pi^T[h(\xi^s) - T(\xi^s)x^e]/\pi^T W \leq (q^s)^T\}, s \in \{1, ..., s\}$$

1.1 Si $Q(x^e) > \theta$; Rajouter au problème (P_e) la coupe d'efficacité (5)

$$\theta \geq \sum_{s=1}^{S} p^s(\pi_s)^T[h(\xi^s) - T(\xi^s)x]$$

1.2 Si $Q(x^e) \leq \theta$; faire $Eff = Eff \bigcup U_{x^e}$, $\widetilde{Eff} = \widetilde{Eff} \bigcup \{Z(x^e)\}$, $s \in \{1, ..., S\}$, aller à la procédure de séparation ;

Procédure de séparation

1. Appliquer l'algorithme de coupes multiples la solution x^e pour éliminer l'ensemble dominé $(D'_{x^e} \bigcup U_{x^e})$ et obtenir les sous problèmes $(P_{e+1}), (P_{e+2}), ..., (P_{e+k'})$.
2. Faire $F = F \bigcup\{(P_{e+1}), (P_{e+2}), ..., (P_{e+k'})\ (P_e)\}$, aller à la procédure de choix ;
   Fin de l'algorithme.

4.3.2 Exemple Numerique[53]

Comme illustration de la méthrode proposée, nous présentons un problème de programmation linéaire stochastique en nombres entiers $multi-objectif$ avec une structure s'emblable celle du problème (P_1) ; K = 3 ; n_0=4 ; m_0=m=n= 2 :

— Contrainte déterministe :

$$-4x_1 + 2x_2 \geq -8$$

$$x_1 + x_2 \leq 5$$

— Deux scénarios (R=2) affectent les trois objectifs et les contraintes stochastiques
$C_1(\xi^1) = (-9, 4)$, $C_2(\xi^1) = (3, -5)$, $C_3(\xi^1) = (8, -11)$;
$C_1(\xi^2) = (3, -2)$, $C_2(\xi^2) = (7, 1)$, $C_3(\xi^2) = (-4, 9)$;

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

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

$\acute{Z}_1 = E[C_1(\xi)] = p(\xi^1)C_1(\xi^1) + p(\xi^2)C_1(\xi^2) = \frac{1}{2}(-9, 4) + \frac{1}{2}(3, -2) = (-3, 1)$
$\acute{Z}_2 = E[C_2(\xi)] = p(\xi^1)C_2(\xi^1) + p(\xi^2)C_2(\xi^2) = \frac{1}{2}(3, -5) + \frac{1}{2}(7, 1) = (5, -2)$
$\acute{Z}_3 = E[C_3(\xi)] = p(\xi^1)C_3(\xi^1) + p(\xi^2)C_3(\xi^2) = \frac{1}{2}(8, -11) + \frac{1}{2}(-4, 9) = (2, -1)$

Initialisation

$$(P_0)\begin{cases} min\ E[C_1(\xi)] + E[C_3(\xi)] \\ s.c: \\ 4x_1 - 2x_2 + x_3 = 8 \\ x_1 + x_2 + x_4 = 5 \\ x_1, x_2, x_3, x_4 \geq 0 \\ x_1, x_2 entiers \end{cases}$$

$F = \{(P_0)\}$; $Eff_0 = \varnothing$; $\widetilde{Eff_0} = \varnothing$;

— **Première itération**
Procédure de choix Nous sélectionnons le problème (P_0).
Procédure d'évaluation La résolution du problème (P_0) donne le tableau 1 $ci-dessous$

B	x_1	x_2	x_3	x_4	x_B
x_1	1	0	$\frac{1}{6}$	$\frac{1}{3}$	3
x_2	0	1	$\frac{-1}{6}$	$\frac{2}{3}$	2
$\tilde{C}_1^+\tilde{C}_3^B$	0	0	$\frac{-1}{6}$	$\frac{-1}{3}$	-3

FIGURE 4.1: Tableau1

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

Procédure de réalisation Pour examiner (tester) la réalisabilité du problème (1) du deuxième étage, nous résolvons le programme (3) avec

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

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

$$\begin{bmatrix} max \ -4\sigma_1^1 + 9\sigma_1^2 \\ -2\sigma_1^1 + 3\sigma_1^2 \leq 0 \\ -\sigma_1^1 + 2\sigma_1^2 \leq 0 \\ 2\sigma_1^1 - 5\sigma_1^2 \leq 0 \\ \sigma_1^1 - 6\sigma_1^2 \leq 0 \\ \sigma_1^1 + \sigma_1^2 \leq 1 \end{bmatrix}$$

Le maximum est à $\sigma_1^T = (\sigma_1^1, \sigma_1^2) = (\frac{2}{3}, \frac{1}{3})$

$$\begin{bmatrix} max \ 3\sigma_2^1 - 16\sigma_1^2 \\ -2\sigma_2^1 + 3\sigma_2^2 \leq 0 \\ -\sigma_2^1 + 2\sigma_2^2 \leq 0 \\ 2\sigma_2^1 - 5\sigma_2^2 \leq 0 \\ \sigma_2^1 - 6\sigma_2^2 \leq 0 \\ \sigma_2^1 + \sigma_2^2 \leq 1 \end{bmatrix}$$

Le maximum est à $\sigma_2^T = (\sigma_2^1, \sigma_2^2) = (0, 0)$

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

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

$\sigma_1^T[h(\xi^1) - T(\xi^1)x] > 0$, ceci signifie que le problème du second étage(1) n'est pas réalisable pour ξ^1. Alors nous créons une coupe de réalisabilité de la forme(4) :

$$\frac{5}{3}x_2 \geq \frac{11}{3} ou \frac{-5}{18}x_3 + \frac{10}{9}x_4 + x_5 = \frac{-1}{3}$$

Nous ajoutons cette coupe et la coupe de Gomory $\frac{-3}{5}x_5 + x_6 = \frac{-4}{5}$ à la première contrainte, nous obtenons un autre point minimum et entier $x = (2, 3)$ qui est donné par le tableau 2 suivant :

B	x_1	x_2	x_3	x_4	x_5	x_6	x_B
x_1	1	0	0	1	0	1	2
x_2	0	1	0	0	0	-1	3
x_3	0	0	1	-4	0	-6	6
x_4	0	0	0	0	1	$\frac{-5}{3}$	$\frac{4}{3}$
$\tilde{C}_1^+\tilde{C}_3^B$	0	0	0	-1	0	-1	-2

FIGURE 4.2: Tableau 2

Procédure de réalisation : Pour tester la réalisabilité du problème (1) du deuxième étage, nous résolvons le programme (3) avec

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

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

$$\begin{bmatrix} max \ -5\sigma_1^1 + 6\sigma_1^2 \\ -2\sigma_1^1 + 3\sigma_1^2 \leq 0 \\ -\sigma_1^1 + 2\sigma_1^2 \leq 0 \\ 2\sigma_1^1 - 5\sigma_1^2 \leq 0 \\ \sigma_1^1 - 6\sigma_1^2 \leq 0 \\ \sigma_1^1 + \sigma_1^2 \leq 1 \end{bmatrix}$$

Le maximum est à $\sigma_1^T = (\sigma_1^1, \sigma_1^2) = (0, 0)$

$$\begin{bmatrix} max \ 4\sigma_2^1 - 17\sigma_1^2 \\ -2\sigma_2^1 + 3\sigma_2^2 \leq 0 \\ -\sigma_2^1 + 2\sigma_2^2 \leq 0 \\ 2\sigma_2^1 - 5\sigma_2^2 \leq 0 \\ \sigma_2^1 - 6\sigma_2^2 \leq 0 \\ \sigma_2^1 + \sigma_2^2 \leq 1 \end{bmatrix}.$$

Le maximum est à $\sigma_2^T = (\sigma_2^1, \sigma_2^2) = (0, 0)$
$\sigma_1 = \sigma_2 = 0$, alors ceci implique que la solution $x = (2, 3)$ obtenue dans le tableau

2 est réalisable pour le problème du deuxième étage.

Procédure d'optimisation Pour tester l'optimalité de x ; le dual (2) est résolu pour ξ^1 et ξ^2

$$\begin{bmatrix} max \ -5\pi_1^1 + 6\pi_1^2 \\ -2\pi_1^1 + 3\pi_1^2 \leq 1 \\ -\pi_1^1 + 2\pi_1^2 \leq 0 \\ 2\pi_1^1 - 5\pi_1^2 \leq 6 \\ \pi_1^1 - 6\pi_1^2 \leq 2 \end{bmatrix}$$

Le maximum est à $\pi_1^T = (\pi_1^1, \pi_1^2) = (-1, -\frac{1}{2})$

$$\begin{bmatrix} max \ 4\pi_2^1 - 17\pi_1^2 \\ -2\pi_1^1 + 3\pi_1^2 \leq 5 \\ -\pi_1^1 + 2\pi_1^2 \leq 3 \\ 2\pi_1^1 - 5\pi_1^2 \leq 2 \\ \pi_1^1 - 6\pi_1^2 \leq 1 \end{bmatrix}$$

Le maximum est à $\pi_2^T = (\pi_2^1, \pi_2^2) = (1, 0)$

$$Q(x, \xi^1) = \pi_1^T[h(\xi^1) - T(\xi^1)x] = 2$$
$$Q(x, \xi^2) = \pi_2^T[h(\xi^2) - T(\xi^2)x] = 4$$
$$Q(x) = \frac{1}{2}Q(x, \xi^1) + \frac{1}{2}Q(x, \xi^2) = 3$$

$\theta = -\infty < Q(x)$, nous introduisons la coupe d'optimalité de la forme (5) :
$\theta \geq \frac{1}{4} - \frac{1}{2}x_1 + \frac{5}{4}x_2$ ou $\frac{1}{2}x_4 + \frac{7}{4}x_6 - \theta + s_1 = -3$ et nous réoptimisons le programme précident(voir tableau 3)

B	x_1	x_2	x_3	x_4	x_5	x_6	θ	s_1	x_B
x_1	1	0	0	1	0	1	0	0	2
x_2	0	1	0	0	0	-1	0	0	3
x_3	0	0	1	-4	0	-6	0	0	6
x_5	0	0	0	0	1	$\frac{-5}{3}$	0	0	$\frac{4}{3}$
θ	0	0	0	$\frac{-1}{2}$	0	$\frac{-7}{4}$	1	3	3
$\tilde{C}_1^+\tilde{C}_3^B$	0	0	0	-1	0	-1	0	0	-2

FIGURE 4.3: Tableau 3

$\theta = Q(x) = 3$ alors, $x^1 = (2,3)$ est une solution du base optimale avec une pénalité $\theta^1 = 3$.
$\underline{Eff_1} = (2,3)$;
$\widetilde{Eff_1} = (0,7,4)$ où $\tilde{Z}^1 = \tilde{Z}(x^1) = (\tilde{Z}_1(x^1), \tilde{Z}_2(x^1), \tilde{Z}_3(x^1)) = (0,7,4)$
Procédure de séparation : On génère deux *sous − problèmes* (P_1) et (P_2) en ajoutant au tableau 3 respectivement les ensembles de coupes suivants :

$$-3x_1 + x_2 \geq -2 et \left\{ \begin{array}{l} -3x_1 + x_2 \geq -3 \\ 2x_1 - x_2 \leq 2 \end{array} \right.$$

$F = (P_1), (P_2)$

— **Deuxième itération**
Procédure de choix Nous sélectionons le problème (P_2).
Procédure d'évaluation IL n'existe pas de solution réalisable , $F = (P_1)$
Procédure de choix Nous sélectionons le problème (P_1).
Procédure d'évaluation La résolution du problème (P_1) tel que nous ajoutons la coupe de Gomory $\frac{-1}{4}x_6 - \frac{1}{4}x_7 + x_8 = \frac{-3}{4}$ à la première contrainte, nous obtenons un autre minimum entier $x = (1,4)$ qui est donné par le tableau 4 suivant :

B	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	θ	s_1	x_B
x_1	1	0	0	0	0	0	0	1	0	0	1
x_2	0	1	0	1	0	0	0	-1	0	0	4
x_3	0	0	1	2	0	0	0	-6	0	0	12
x_5	0	0	0	$\frac{5}{3}$	1	0	0	$\frac{-5}{3}$	0	0	3
x_6	0	0	0	1	0	1	0	-1	0	0	1
x_7	0	0	0	1	0	0	1	-4	0	0	3
θ	0	0	0	$\frac{5}{4}$	0	0	0	$\frac{-7}{4}$	1	-1	$\frac{19}{4}$
$\tilde{C}_1^+ \tilde{C}_3^B$	0	0	0	0	0	0	0	-1	0	0	-1

FIGURE 4.4: Tableau 4

Procédure de réalisation : Pour tester la réalisabilité du second étage du problème(1), nous résolvons le programme (3) avec

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

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

$$\begin{bmatrix} max \ -6\sigma_1^1 + 3\sigma_1^2 \\ -2\sigma_1^1 + 3\sigma_1^2 \leq 0 \\ -\sigma_1^1 + 2\sigma_1^2 \leq 0 \\ 2\sigma_1^1 - 5\sigma_1^2 \leq 0 \\ \sigma_1^1 - 6\sigma_1^2 \leq 0 \\ \sigma_1^1 + \sigma_1^2 \leq 1 \end{bmatrix}$$

Le maximum est à $\sigma_1^T = (\sigma_1^1, \sigma_1^2) = (0, 0)$

$$\begin{bmatrix} max \ 5\sigma_2^1 - 18\sigma_1^2 \\ -2\sigma_2^1 + 3\sigma_2^2 \leq 0 \\ -\sigma_2^1 + 2\sigma_2^2 \leq 0 \\ 2\sigma_2^1 - 5\sigma_2^2 \leq 0 \\ \sigma_2^1 - 6\sigma_2^2 \leq 0 \\ \sigma_2^1 + \sigma_2^2 \leq 1 \end{bmatrix}$$

Le maximum est à $\sigma_2^T = (\sigma_2^1, \sigma_2^2) = (0, 0)$
$\sigma_1 = \sigma_2 = 0$, alors ceci implique que la solution $x = (1, 4)$ obtenue dans le tableau 4 est réalisable pour le problème du second étage du problème(1).
Procédure d'optimisation : Pour tester l'optimalité de x ; le dual (2) est résolu pour ξ^1 et ξ^2

$$\begin{bmatrix} max \ -6\pi_1^1 + 3\pi_1^2 \\ -2\pi_1^1 + 3\pi_1^2 \leq 1 \\ -\pi_1^1 + 2\pi_1^2 \leq 0 \\ 2\pi_1^1 - 5\pi_1^2 \leq 6 \\ \pi_1^1 - 6\pi_1^2 \leq 2 \end{bmatrix}$$

Le maximum est à $\pi_1^T = (\pi_1^1, \pi_1^2) = (-1, -\frac{1}{2})$

$$\begin{bmatrix} max \ 5\pi_2^1 - 18\pi_1^2 \\ -2\pi_1^1 + 3\pi_1^2 \leq 5 \\ -\pi_1^1 + 2\pi_1^2 \leq 3 \\ 2\pi_1^1 - 5\pi_1^2 \leq 2 \\ \pi_1^1 - 6\pi_1^2 \leq 1 \end{bmatrix}$$

Le maximum est à $\pi_2^T = (\pi_2^1, \pi_2^2) = (1, 0)$

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

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

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

$\theta = Q(x)$, alors $x^2 = (1,4)$ est une solution entière et optimale avec $\theta^2 = \frac{19}{4}$.
$\underline{Eff}_2 = (2,3),(1,4)$;
$\widetilde{Eff}_2 = (0,7,4),(\frac{23}{4},\frac{5}{4},\frac{11}{4})$ où $\tilde{Z}^2 = \tilde{Z}(x^2) = (\tilde{Z}_1(x^2),\tilde{Z}_2(x^2),\tilde{Z}_3(x^2)) = (\frac{23}{4},\frac{5}{4},\frac{11}{4})$
Procédure de séparation : On génère deux $sous-problèmes$ (P_3) et (P_4) en ajoutant au tableau 3 respectivement les ensembles de coupes suivants :

$$-3x_1 + x_2 \geq -2 \quad et \quad \begin{cases} -3x_1 + x_2 \leq 1 \\ 2x_1 - x_2 \geq -1 \end{cases}$$

$F = (P_3),(P_4)$.

— **troisième itération**
Procédure de choix : Nous sélectionons le problème (P_4).
Procédure d'évaluation : La résolution du problème (P_4) donne le tableau 5 $ci-dessous$

B	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	θ	s_1	x_B
x_1	1	0	0	0	0	0	0	1	0	0	0	0	1
x_2	0	1	0	0	0	0	0	2	0	1	0	0	3
x_3	0	0	1	0	0	0	0	0	0	2	0	0	10
x_5	0	0	0	0	1	0	0	$\frac{10}{3}$	0	$\frac{5}{3}$	0	0	$\frac{4}{3}$
x_6	0	0	0	0	0	1	0	2	0	1	0	0	0
x_7	0	0	0	0	0	0	1	-1	0	1	0	0	2
x_9	0	0	0	0	0	0	0	1	1	-1	0	0	1
x_4	0	0	0	1	0	0	0	-3	0	-1	0	0	1
θ	0	0	0	0	0	0	0	2	0	$\frac{5}{4}$	1	-1	$\frac{7}{2}$
$\tilde{C}_1^+\tilde{C}_3^B$	0	0	0	0	0	0	0	-1	0	0	0	0	-1

FIGURE 4.5: Tableau 5

le minimum est atteint en $x = (1,3)$
Procédure de réalisation : Pour tester la réalisabilité du second étage du problème(1), nous résolvons le programme (3) avec

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

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

$$\begin{bmatrix} max \ -4\sigma_1^1+4\sigma_1^2 \\ -2\sigma_1^1+3\sigma_1^2 \leq 0 \\ -\sigma_1^1+2\sigma_1^2 \leq 0 \\ 2\sigma_1^1-5\sigma_1^2 \leq 0 \\ \sigma_1^1-6\sigma_1^2 \leq 0 \\ \sigma_1^1+\sigma_1^2 \leq 1 \end{bmatrix}$$

Le maximum est à $\sigma_1^T = (\sigma_1^1, \sigma_1^2) = (0, 0)$

$$\begin{bmatrix} max \ 5\sigma_2^1-14\sigma_1^2 \\ -2\sigma_2^1+3\sigma_2^2 \leq 0 \\ -\sigma_2^1+2\sigma_2^2 \leq 0 \\ 2\sigma_2^1-5\sigma_2^2 \leq 0 \\ \sigma_2^1-6\sigma_2^2 \leq 0 \\ \sigma_2^1+\sigma_2^2 \leq 1 \end{bmatrix}$$

Le maximum est à $\sigma_2^T = (\sigma_2^1, \sigma_2^2) = (0, 0)$
$\sigma_1 = \sigma_2 = 0$, alors ceci implique que la solution $x = (2, 3)$ obtenue dans le tableau 2 est réalisable pour le problème du second étage.
Procédure d'optimisation : Pour tester l'optimalité de x ; le dual (2) est résolu pour ξ^1 et ξ^2

$$\begin{bmatrix} max \ -5\pi_1^1+6\pi_1^2 \\ -2\pi_1^1+3\pi_1^2 \leq 1 \\ -\pi_1^1+2\pi_1^2 \leq 0 \\ 2\pi_1^1-5\pi_1^2 \leq 6 \\ \pi_1^1-6\pi_1^2 \leq 2 \end{bmatrix}$$

Le maximum est à $\pi_1^T = (\pi_1^1, \pi_1^2) = (-1, -\frac{1}{2})$

$$\begin{bmatrix} max \ 5\pi_2^1-14\pi_1^2 \\ -2\pi_1^1+3\pi_1^2 \leq 5 \\ -\pi_1^1+2\pi_1^2 \leq 3 \\ 2\pi_1^1-5\pi_1^2 \leq 2 \\ \pi_1^1-6\pi_1^2 \leq 1 \end{bmatrix}$$

Le maximum est à $\pi_2^T = (\pi_2^1, \pi_2^2) = (1, 0)$

$$Q(x, \xi^1) = \pi_1^T[h(\xi^1) - T(\xi^1)x] = 2$$
$$Q(x, \xi^2) = \pi_2^T[h(\xi^2) - T(\xi^2)x] = 5$$
$$Q(x) = \frac{1}{2}Q(x, \xi^1) + \frac{1}{2}Q(x, \xi^2) = \frac{7}{2}$$

$\theta = Q(x)$, alors $x^3 = (1,3)$ est une solution entière et optimale avec $\theta^3 = \frac{7}{2}$.
$Eff_3 = (2,3),(1,4),(1,3)$;
$\widetilde{Eff}_3 = (0,7,4),(\frac{23}{4},\frac{5}{4},\frac{11}{4}),(\frac{7}{2},\frac{5}{2},\frac{5}{2}$ où $\tilde{Z}^3 = \tilde{Z}(x^3) = (\tilde{Z}_1(x^3),\tilde{Z}_2(x^3),\tilde{Z}_3(x^3)) = (\frac{7}{2},\frac{5}{2},\frac{5}{2})$
Procédure de séparation : On génère deux $sous - problèmes$ (P_5) et (P_6) en ajoutant au tableau 3 respectivement les ensembles de coupes suivants :

$$-3x_1 + x_2 \geq -1 \quad et \quad \begin{cases} -3x_1 + x_2 \leq 0 \\ 2x_1 - x_2 \geq 0 \end{cases}$$

$F = (P_3),(P_5),(P_6)$.

— **Quatrième itération**
Procédure de choix Nous sélectionons le problème (P_6).
Procédure d'évaluationIL n'existe pas de solution réalisable, $F = (P_3),(P_5)$

— **Cinquième itération**
Procédure de choix Nous sélectionons le problème (P_5).
Procédure d'évaluationIL n'existe pas de solution réalisable, $F = (P_3)$

— **Sixième itération**
Procédure de choix Nous sélectionons le problème (P_3), nous ajoutons la coupe de Gomory $\frac{-1}{4}x_4 - \frac{1}{4}x_9 + x_{10} = \frac{-3}{4}$ à la première contrainte du problème (P_3), nous obtenons un autre minimum $x = (0,5)$ qui est donné par le tableau 6 $ci - dessous$

B	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	θ	s_1	x_B
x_1	1	0	0	0	0	0	0	0	0	1	0	0	0
x_2	0	1	0	1	0	0	0	0	0	-1	0	0	5
x_3	0	0	1	2	0	0	0	0	0	-6	0	0	18
x_5	0	0	0	$\frac{5}{3}$	1	0	0	0	0	$\frac{-5}{3}$	0	0	$\frac{14}{3}$
x_6	0	0	0	1	0	1	0	0	0	-1	0	0	2
x_7	0	0	0	1	0	0	1	0	0	-4	0	0	7
x_8	0	0	0	0	0	0	0	1	1	-1	0	0	3
x_9	0	0	0	1	0	0	0	0	0	-4	0	0	1
θ	0	0	0	$\frac{5}{4}$	0	0	0	0	0	$\frac{-7}{4}$	1	-1	$\frac{13}{2}$
$\tilde{C}_1^+\tilde{C}_3^B$	0	0	0	0	0	0	0	0	0	-1	0	0	0

FIGURE 4.6: Tableau 6

Procédure de réalisation : Pour tester la réalisabilité du second étage du problème(1), nous résolvons le programme (3) avec

$$h(\xi^1) - T(\xi^1)x = \begin{pmatrix} 3 \\ 5 \end{pmatrix} - \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 5 \end{pmatrix} = \begin{pmatrix} -7 \\ 0 \end{pmatrix}$$

$$h(\xi^2) - T(\xi^2)x = \begin{pmatrix} 6 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 0 \\ 5 \end{pmatrix} = \begin{pmatrix} 6 \\ -19 \end{pmatrix}$$

$$\begin{bmatrix} max\ -7\sigma_1^1 \\ -2\sigma_1^1 + 3\sigma_1^2 \leq 0 \\ -\sigma_1^1 + 2\sigma_1^2 \leq 0 \\ 2\sigma_1^1 - 5\sigma_1^2 \leq 0 \\ \sigma_1^1 - 6\sigma_1^2 \leq 0 \\ \sigma_1^1 + \sigma_1^2 \leq 1 \end{bmatrix}$$

Le maximum est à $\sigma_1^T = (\sigma_1^1, \sigma_1^2) = (0, 0)$

$$\begin{bmatrix} max\ 6\sigma_2^1 - 19\sigma_1^2 \\ -2\sigma_2^1 + 3\sigma_2^2 \leq 0 \\ -\sigma_2^1 + 2\sigma_2^2 \leq 0 \\ 2\sigma_2^1 - 5\sigma_2^2 \leq 0 \\ \sigma_2^1 - 6\sigma_2^2 \leq 0 \\ \sigma_2^1 + \sigma_2^2 \leq 1 \end{bmatrix}$$

Le maximum est à $\sigma_2^T = (\sigma_2^1, \sigma_2^2) = (0, 0)$

$\sigma_1 = \sigma_2 = 0$, alors ceci implique que la solution $x = (0, 5)$ obtenue dans le tableau 6 est réalisable pour le problème du second étage du problème(1).
Procédure d'optimisation : Pour tester l'optimalité de x ; le dual (2) est résolu pour ξ^1 et ξ^2

$$\begin{bmatrix} max \ -7\pi_1^1 \\ -2\pi_1^1 + 3\pi_1^2 \leq 1 \\ -\pi_1^1 + 2\pi_1^2 \leq 0 \\ 2\pi_1^1 - 5\pi_1^2 \leq 6 \\ \pi_1^1 - 6\pi_1^2 \leq 2 \end{bmatrix}$$

Le maximum est à $\pi_1^T = (\pi_1^1, \pi_1^2) = (-1, -\frac{1}{2})$

$$\begin{bmatrix} max \ 6\pi_2^1 - 19\pi_1^2 \\ -2\pi_1^1 + 3\pi_1^2 \leq 5 \\ -\pi_1^1 + 2\pi_1^2 \leq 3 \\ 2\pi_1^1 - 5\pi_1^2 \leq 2 \\ \pi_1^1 - 6\pi_1^2 \leq 1 \end{bmatrix}$$

Le maximum est à $\pi_2^T = (\pi_2^1, \pi_2^2) = (1, 0)$

$$Q(x, \xi^1) = \pi_1^T[h(\xi^1) - T(\xi^1)x] = 7$$
$$Q(x, \xi^2) = \pi_2^T[h(\xi^2) - T(\xi^2)x] = 6$$
$$Q(x) = \frac{1}{2}Q(x, \xi^1) + \frac{1}{2}Q(x, \xi^2) = \frac{13}{2}$$

$\theta = Q(x)$, alors $x^4 = (0, 5)$ est une solution entière et optimale avec $\theta^4 = \frac{13}{2}$.
$Eff_4 = (2, 3), (1, 4), (1, 3), (0, 5)$;
$\widetilde{Eff}_4 = (0, 7, 4), (\frac{23}{4}, \frac{5}{4}, \frac{11}{4}), (\frac{7}{2}, \frac{5}{2}, \frac{5}{2}), (\frac{23}{2}, \frac{-7}{2}, \frac{3}{2})$ où $\tilde{Z}^4 = \tilde{Z}(x^4) = (\tilde{Z}_1(x^4), \tilde{Z}_2(x^4), \tilde{Z}_3(x^4)) = (\frac{23}{2}, \frac{-7}{2}, \frac{3}{2})$
Procédure de séparation : On génère deux $sous-problèmes$ (P_7) et (P_8) en ajoutant au tableau 3 respectivement les ensembles de coupes suivants :

$$-3x_1 + x_2 \geq 6 \quad et \quad \begin{cases} -3x_1 + x_2 \leq 5 \\ 2x_1 - x_2 \geq -4 \end{cases}$$

$F = (P_7), (P_8)$.

— **Septième itération**
Procédure de choix : Nous sélectionons le problème (P_8)

Procédure d'évaluation : La résolution du problème (P_8) donne le tableau 7 $ci - dessous$

B	x_1	x_2	x_3	x_4	...	x_{10}	x_{11}	x_{12}	θ	s_1	x_B
x_1	1	0	0	0	...	1	0	0	0	0	0
x_2	0	1	0	0	...	2	0	1	0	0	4
x_3	0	0	1	0	...	0	0	2	0	0	16
x_5	0	0	0	0	...	$\frac{10}{3}$	0	$\frac{5}{3}$	0	0	3
x_6	0	0	0	0	...	2	0	1	0	0	1
x_7	0	0	0	0	...	-1	0	1	0	0	6
x_8	0	0	0	0	...	-1	0	0	0	0	1
x_9	0	0	0	0	...	-1	0	1	0	0	2
x_{11}	0	0	0	0	...	1	1	-1	0	0	1
x_4	0	0	0	1	...	2	0	$\frac{5}{4}$	1	-1	1
θ	0	0	0	0	...	2	0	$\frac{5}{4}$	1	-1	$\frac{21}{4}$
$\tilde{C}_1^+\tilde{C}_3^B$	0	0	0	0	...	1	0	0	0	0	0

FIGURE 4.7: Tableau 7

Le minimum est atteint en $x = (0, 4)$

Procédure de réalisation :Pour tester la réalisabilité du second étage du problème(1), nous résolvons le programme (3) avec

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

$$h(\xi^2) - T(\xi^2)x = \begin{pmatrix} 6 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 0 \\ 4 \end{pmatrix} = \begin{pmatrix} 6 \\ -15 \end{pmatrix}$$

$$\begin{bmatrix} max \ -5\sigma_1^1 + \sigma_1^2 \\ -2\sigma_1^1 + 3\sigma_1^2 \leq 0 \\ -\sigma_1^1 + 2\sigma_1^2 \leq 0 \\ 2\sigma_1^1 - 5\sigma_1^2 \leq 0 \\ \sigma_1^1 - 6\sigma_1^2 \leq 0 \\ \sigma_1^1 + \sigma_1^2 \leq 1 \end{bmatrix}$$

Le maximum est à $\sigma_1^T = (\sigma_1^1, \sigma_1^2) = (0, 0)$

$$\begin{bmatrix} max\ 6\sigma_2^1 - 15\sigma_1^2 \\ -2\sigma_2^1 + 3\sigma_2^2 \leq 0 \\ -\sigma_2^1 + 2\sigma_2^2 \leq 0 \\ 2\sigma_2^1 - 5\sigma_2^2 \leq 0 \\ \sigma_2^1 - 6\sigma_2^2 \leq 0 \\ \sigma_2^1 + \sigma_2^2 \leq 1 \end{bmatrix}$$

Le maximum est à $\sigma_2^T = (\sigma_2^1, \sigma_2^2) = (0, 0)$
$\sigma_1 = \sigma_2 = 0$, alors ceci implique que la solution $x = (0, 4)$ obtenue dans le tableau 7 est réalisable pour le problème du second étage du problème(1).
Procédure d'optimisation : Pour tester l'optimalité de x ; le dual (2) est résolu pour ξ^1 et ξ^2

$$\begin{bmatrix} max\ \ -5\pi_1^1 + 15\pi_1^2 \\ -2\pi_1^1 + 3\pi_1^2 \leq 1 \\ -\pi_1^1 + 2\pi_1^2 \leq 0 \\ 2\pi_1^1 - 5\pi_1^2 \leq 6 \\ \pi_1^1 - 6\pi_1^2 \leq 2 \end{bmatrix}$$

Le maximum est à $\pi_1^T = (\pi_1^1, \pi_1^2) = (-1, -\frac{1}{2})$

$$\begin{bmatrix} max\ 6\pi_2^1 - 15\pi_1^2 \\ -2\pi_1^1 + 3\pi_1^2 \leq 5 \\ -\pi_1^1 + 2\pi_1^2 \leq 3 \\ 2\pi_1^1 - 5\pi_1^2 \leq 2 \\ \pi_1^1 - 6\pi_1^2 \leq 1 \end{bmatrix}$$

Le maximum est à $\pi_2^T = (\pi_2^1, \pi_2^2) = (1, 0)$

$$Q(x, \xi^1) = \pi_1^T[h(\xi^1) - T(\xi^1)x] = \frac{9}{2}$$
$$Q(x, \xi^2) = \pi_2^T[h(\xi^2) - T(\xi^2)x] = 6$$
$$Q(x) = \frac{1}{2}Q(x, \xi^1) + \frac{1}{2}Q(x, \xi^2) = \frac{21}{4}$$

$\theta = Q(x)$, alors $x^5 = (0, 4)$ est une solution entière et optimale avec $\theta^5 = \frac{21}{2}$.
$Eff_5 = (2, 3), (1, 4), (1, 3), (0, 5), (0, 4)$;
$\widehat{Eff_5} = (0, 7, 4), (\frac{23}{4}, \frac{5}{4}, \frac{11}{4}), (\frac{7}{2}, \frac{5}{2}, \frac{5}{2}), (\frac{23}{2}, \frac{-7}{2}, \frac{3}{2}), \frac{5}{2}, \frac{5}{2}), (\frac{37}{4}, \frac{-11}{4}, \frac{5}{4})$
où $\tilde{Z}^5 = \tilde{Z}(x^5) = (\tilde{Z}_1(x^4), \tilde{Z}_2(x^5), \tilde{Z}_3(x^5)) = ($
Procédure de séparation On génère deux $sous-problèmes$ (P_9) et (P_{10}) en ajoutant au tableau 3 respectivement les ensembles de coupes suivants :

$$-3x_1 + x_2 \geq 5 \quad et \quad \begin{cases} -3x_1 + x_2 \leq 4 \\ 2x_1 - x_2 \geq -3 \end{cases}$$

$F = (P_7), (P_9), (P_{10})$.

— **Huitième itération**

Procédure de choix : Nous sélectionons le problème (P_{10}),

Procédure d'évaluation : La résolution du problème (P_{10}) donne le tableau 8 $ci - dessous$

B	x_1	x_2	x_3	x_4	...	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	θ	s_1	x_B
x_1	1	0	0	0	...	1	0	0	0	0	0	0	0
x_2	0	1	0	0	...	2	0	0	0	1	0	0	3
x_3	0	0	1	0	...	0	0	0	0	2	0	0	14
x_5	0	0	0	0	...	$\frac{10}{3}$	0	0	0	$\frac{5}{3}$	0	0	$\frac{4}{3}$
x_6	0	0	0	0	...	2	0	0	0	1	0	0	0
x_7	0	0	0	0	...	-1	0	0	0	1	0	0	5
x_8	0	0	0	0	...	-1	0	0	0	0	0	0	1
x_9	0	0	0	0	...	-1	0	0	0	1	0	0	1
x_{11}	0	0	0	0	...	1	1	0	0	-1	0	0	2
x_4	0	0	0	1	...	-3	0	0	0	-1	0	0	2
x_{13}	0	0	0	0	...	0	0	1	0	-1	0	0	1
x_{12}	0	0	0	0	...	0	0	1	0	-1	0	0	1
θ	0	0	0	0	...	2	0	0	0	$\frac{5}{4}$	1	-1	4
$\tilde{C}_1^+ \tilde{C}_3^B$	0	0	0	0	...	-1	0	0	0	0	0	0	0

FIGURE 4.8: Tableau 8

Le minimum est atteint en $x = (0, 4)$

Procédure de réalisation : Pour tester la réalisabilité du second étage du problème(1), nous résolvons le programme (3) avec

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

$$h(\xi^2) - T(\xi^2)x = \begin{pmatrix} 6 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 0 \\ 3 \end{pmatrix} = \begin{pmatrix} 6 \\ -11 \end{pmatrix}$$

$$\begin{bmatrix} max - 3\sigma_1^1 + 2\sigma_1^2 \\ -2\sigma_1^1 + 3\sigma_1^2 \leq 0 \\ -\sigma_1^1 + 2\sigma_1^2 \leq 0 \\ 2\sigma_1^1 - 5\sigma_1^2 \leq 0 \\ \sigma_1^1 - 6\sigma_1^2 \leq 0 \\ \sigma_1^1 + \sigma_1^2 \leq 1 \end{bmatrix}$$

Le maximum est à $\sigma_1^T = (\sigma_1^1, \sigma_1^2) = (0,0)$

$$\begin{bmatrix} max\ 6\sigma_2^1 - 11\sigma_1^2 \\ -2\sigma_2^1 + 3\sigma_2^2 \leq 0 \\ -\sigma_2^1 + 2\sigma_2^2 \leq 0 \\ 2\sigma_2^1 - 5\sigma_2^2 \leq 0 \\ \sigma_2^1 - 6\sigma_2^2 \leq 0 \\ \sigma_2^1 + \sigma_2^2 \leq 1 \end{bmatrix}$$

$\sigma_2^T[h(\xi^2) - T(\xi^2)x] = \frac{8}{7}$
$\sigma_1^T[h(\xi^1) - T(\xi^1)x] > 0$, ceci signifie que le problème du second étage(1) n'est pas réalisable pour ξ^2. Alors nous créons une coupe de réalisabilité de la forme (4) :
$\frac{11}{7}x_1 + \frac{8}{7}x_2 \geq \frac{32}{7}$ et réoptimisons le programme précident
Il n'existe pas de solution réalisable,$F = (P_7), (P_9)$

— **Neuvième itération**
Procédure de choix Nous sélectionons le problème (P_9),
Procédure d'évaluation IL n'existe pas de solution réalisable, $F = (P_7)$
Dixième itération
Procédure de choix$Noussélectiononsleproblème$($\mathrm{P}_7$),
Procédure d'évaluation IL n'existe pas de solution réalisable, $F = \varnothing$

— **Onzième itération**
Procédure de choix $F = \varnothing$, afficher $Eff, \widetilde{Eff}, terminer.$

	$i = 1$	i=2	i=3	i=4	i=5
x^i	(2,3)	(1,4)	(1,3)	(0,5)	(0,4)
θ^i	3	$\frac{19}{4}$	$\frac{7}{2}$	$\frac{13}{4}$	$\frac{21}{4}$
$\tilde{Z}^i$	(0,7,4)	$(\frac{23}{4},\frac{5}{4},\frac{11}{4})$	$(\frac{7}{2},\frac{5}{2},\frac{5}{2})$	$(\frac{23}{2},\frac{-7}{2},\frac{3}{2})$	$(\frac{37}{4},\frac{-11}{4},\frac{5}{4})$

4.4 Conclusion

Le problème de la programmation linéaire stochastique $multi - objectif$ en nombres entiers a été appliquée une grande variété de secteurs comme l'agriculture, finances, militaire, le contrôle de production, transport.

Chapitre 4

Conclusion générale

Dans ce travail nous avons considéré un problème d'optimisation linéaire stochastique objectifs multiples dans lesquels l'information comporte des éléments indéterminés, ou bien les problèmes dont certains paramètres sont aléatoires mais définis par des caractéristiques probabilistes connues. Les données stochastiques sont traitées par un recours approché pour obtenir un programme déterministe équivalent à deux étages.
L'algorithme proposé par M.Moulaï et S.Amrouche [53] peut déterminer l'ensemble de toutes les solutions efficaces entières de ce problème équivalent (MOSILP), si elles existent, ou alors un sous ensemble de solutions efficaces si leur nombre est fini ou très grand. 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 ; puisque pour construire l'ensemble des solutions efficaces, le problème $mono-objectif$ d'une fonction scalaraisante est résolu chaque itération et les valeurs des fonctions objectives d'une solution efficace sont facilement evaluées.
Puisque les coupes de réalisabilité éliminent plusieurs parties de décision et aussi les coupes multiples une solution éliminent l'ensemble dominé, le nombre d'itérations diminue pour obtenir les solutions efficaces.
Il reste beaucoup de questions en suspens de recherches dans le domaine stochastique, en particulier dans les secteurs de la programmation stochastique multi-objectif. Nous proposons comme perspectifs de résoudre le problème (MOILSP) en utilisant d'autres techniques de résolutions de problèmes multi-objectif autre que la méthode de coupe multiple.
Aussi dans certaines applications, le décideur n'a pas souvent la possibilité de faire recours dans le futur, après l'occurrence d'un scénario. Dans ce cas, nous devons faire recours à une autre approche de la programmation stochastique pour convertir le problème stochas-

tique à un problème déterministe.
Nous suggérons dans l'avenir d'introduire cette approche à la place du modèle de recours utilisé pour résoudre le problème.

Chapitre 4

Bibliographie

[1] M.Abbas, F.Bellahcene ; *Cutting plane method for multiple objective stochastic integer linear programming* ; European Journal of Operational Research 168, $967-984$; 2006..

[2] M. Abbas and M.Moulaï ; *Solving multiple objective integer linear programming* ; Journal of the Italian Operations Research Society (Ricerca Operativa)29,pp. $15-38$, (1999).

[3] Anthony Papavasiliou, *The L-Shaped Method*

[4] J.F.Benders ; *Partitioning procedures for solving mixedvariables programming problems.* Numerische Mathematik, 4 :$238-252$, 1962.

[5] E.M.L.Beale.*On minimizing a convex functions ubject to linear inequalities* ; J.Roy.Statist.Soc., 17, 1955.

[6] E.M.L.Beale.*On minimizing a convex functions ubject to linear inequalities* ; J.Roy.Statist.Soc., 17, 1955.

[7] F.Ben Abdelaziz ; *L'efficacité en programmation multiobjectifs stochastique* ; Ph.D. Thesis, Université de Laval, Québec, 1992.

[8] F.Ben Abdelaziz, P.Lang, R.Nadeau ; *Distributional unanimity multiobjective stochastic linear programming, Multicriteria Analysis : Proceedings of the XIth International Conference on MCDM, $Springer-Verlag$*, Berlin, $225-236$, 1997.

[9] F.Ben Abdelaziz, P.Lang, R.Nadeau ; *Dominance and efficiency in multicriteria decision under ncertainty* ; Theory and Decision 47, $191-211$, 1999.

[10] A.$Ben-Tal$, J. Tind ; *Duality with Multiple Criteria and Multiple Resources* ; 1996.

[11] R. Benayoun, J. De Montgolfier, J. Tergny and O. Laritchev, *Linear Programming with Multiple Objective Functions : Step Method (STEM)*, Mathematical Programming 1, No-3, 366-375, 1971.

[12] Benki ; *A.Méthodes efficaces de capture de front de pareto en conception mécanique multicritère* ; applications industrielles ;2014.

[13] Bérubé $Jean - François$, Gendreau Michel, et Potvin $Jean - Yves$. *An exact ϵ − constraint method for bi − objective combinatorial optimization problems : Application to the traveling salesman problem with profits. European Journal of Operational Research* ;194 (1) :39 − 50, 2009.

[14] A. Charnes and W.W. Cooper, *Management models and industrial applications of linear programming*, John Wiley, New York, 1961.

[15] A.Charnes,W.W.CooperandG.H.Symonds.*Cost horizons and certainty equivalents : An approach to stochastic programming of heatingoil* ; Management Science, 4 :183-195, 1958.

[16] CHANKONG, V., and Y.Y. Haimes ; *Multiobjective Decision − Making : Theory and Methodology, System Science and Engineering* ; North Holland, 1983.

[17] CHERGUI, M.A and MOULAï ; *An Exact Method for a Discrete Multiobjective Linear Fractional Optimization, Applied Mathematics and Decision Sciences* ; 2008.

[18] COELLO COELLO. C.A ; *An updated survuey of g.a based multiobjective optimization techniques Technical report Lania − RI* − 98 − 08 Laboratorio Nacional Avanzada,Xalapa, Veracruz, Mexico Deceumber 1998.

[19] Y. Collette et P. Siarry, *Optimisation multiobjectif*, Editions Eyrolles, 2002.

[20] Crowder, H., Johnson, E.L and Padberg, M ;*Solving largescale zero − one linear programming problems* ; 803-834.

[21] H.W. Corley, *A New Scalar Equivalence for Pareto Optimization*, IEEE Transactions on Automatic Control 25, No-4, 829-830, 1980.

[22] V. Chankong and Y.Y. Hamies, *Multiobjective Decision Making : Theory and Methodology*, New York : Elsevier-North-Holland, 1983.

[23] Chvátal V. *Edmonds polytopes and a hierarchy of combinatorial problems*, Discrete Math.4, 305-337(1973).

[24] V. CHVATAL, *Linear programming* , MC Gill University. p115 et p28-29.(1999)

[25] G.B.Dantzig. *Linear programming under uncertainty. Management Science*, 1 :197-206, 1955.

[26] Dantzing, G. B ;*Discrete-variable extremum problems* ; 266277. Operations Research.

[27] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, New-York : John Wiley, 2001.

[28] C. Dhaenens-Flipo, *Optimisation Combinatoire Multi-Objectif : Apport des Méthodes Coopératives et Contribution à l'Extraction de Connaissances*, thèse d'Habilitation à diriger des Recherches de l'U.S.T.L, Lille, 2005.

[29] M. Ehrgott and X. Gandibleux, *A survey and annotated bibliography of multiobjective combinatorial optimization*, OR Spektrum, 22 :425-460, 2000.

[30] M. Ehrgott, *Multicriteria Optimization*, Lecture Notes in Economics and Mathematical Systems, numéro 491, Springer Verlag, Berlin, Heidelberg, 2000.

[31] P.H. Farquhar, *Utility assesment methods*, Management Science 30, 1283-1300, 1984.

[32] P.C. Fishburn, *Utility theory*, Managment science, vol. 14, no. 4, 355-378, 1968.

[33] C.GUERARD. *Programmation Linéaire* Paris, Eyrolles 1976.

[34] Gomory R.E. *Outline of an algorithm for intger solutions to linear programs*, Bull.Amer.Math .Soc 64, 275-278(1958)

[35] Gomory R.E. *An algorithm for the mixed-integer problem*, Report RM-2597, Rand corporation(1960).

[36] G.P. Huber, *Multiattribute utility models, A review of field and fieldlike studies*, Managament Science 20, No-10, 1393-1402, 1974.

[37] J.P. Ignizio, *Goal Programming and Its Extensions*, D.C. Heath, Lexington, MA, 1976.

[38] J. Ijiri, *Management Goals and Accounting for Control*, American Elsevier, New York, 1965.

[39] John R.Birge ; *Stochastic programming computation and applications*.INFORMSJ.Comput.,9(2) :111 − 133,1997.

[40] John R.Birge and François Louveaux.*Introduction to stochastic programming.* $Springer - Verlag$, New York, 1997.

[41] P.Kall and J.Mayer ; *Stochastic Linear Programming.* Springer, New York, 2005.

[42] P.Kall and S.W.Wallace ; *Stochastic Programming* ;Wiley, Chichester etc. 1994.

[43] R.L. Keeney and H. Raiffa, *Decisions with multiple objectives : Preferences and value tradeoff*, éditions Cambridge University Press, 1993.

[44] W.K.Klein Haneveld,L.Stougie,and M.H.vander Vlerk ; *Stochastic integer programming with simple recourse* ; Research Memorandum 455, Institute of Economic Research, University of Groningen, 1991

[45] Willem K.Klein Haneveld and Maarten H.van der Vlerk.*Stochastic integer programming : general models and algorithms* ; Ann. Oper. Res., 85 :39 − 57, 1999. Stochastic programming. State of the art, 1998 (Vancouver, BC).

[46] T.C. Koopmans, *Analysis and Production as an Efficient Combination of Activities*, Activity Analysis of Production and Allocation, Yale University Press, New Haven, London, 33-97, 1971 (originally published in 1951).

[47] H.W. Kuhn and A.W. Tucker, *Nonlinear Programming*, Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, Los Angeles, 481-492, 1951.

[48] Land A.H and Doig A.G, *An automatic method for solving discrete programming problems*. Econometrica 28(1960), 497-520.

[49] C.E LEMKE, *The dual method of solving the linear programming problem* , Naval Research Logistics Quarterly 1 : 36-47.(1954)

[50] J.M. Martel and B. Aouni, *Diverse Imprecise Goal Programming Model Formulations*, Journal of Global Optimization, 12, 1998, 127-138.

[51] M.Moulaï, S.Amrouche ; *Optimisation linéaire stochastique multiobjectifs en nombres entiers* ; Actes COSI'06, $404 - 420$, 2006.

[52] M. Zeleny, *Multiple Criteria Decision Making*, McGraw-Hill, New York, 1982.

[53] M.Moulaï et S.Amrouche, *Optimisation linéaire stochastique multi-objectifs en nombres entiers*, Université des Sciences et de la Technologie Houari Boumédiène en 2005

[54] K. Miettinen, *On the Methodology of Multiobjective Optimization Problems*, Journal of Optimization Theory and Applications 42, No-4, 499-524, 1984.

[55] K. Miettinen, *Nonlinear Multiobjective Optimization*, Kluwer Academic Publishers, Boston, 1999.

[56] G.L.NEMHAUSER, L.A.WOLSEY, 1988 ; *Integer and combinatorial optimization* ; Willey Interscience Serirs in Discrete Mathematics and Optimisation.

[57] A.PREKOPA, 1995. *Stochastic Programming*. Kluwer Academic Publishers, Dordrecht.

[58] B. Roy et D. Bouyssou, *Aide multicritère à la décision : méthodes et cas*, éditions Economica, 1993.

[59] B. Roy, *Paradigms and challenges*, Multiple Criteria Decision Analysis-State of the Art Surveys, Springer, 3-24, 2005.

[60] Roger, J-B. Wets. *Solving stochastic programs with simple recourse*. Stochastics, $10(3-4)$:$219-242$, 1983.

[61] R.T.Rockafellar and Roger.$J-B$.Wets ; *A dual solution procedure for quadratic stochastic programs with simple recourse* ; In Numerical methods (Caracas, 1982), volume 1005 of Lecture Notes in Math.,pages $252-265$.Springer, Berlin, 1983.

[62] R.E. Steuer, *Multiple Criteria Optimization : Theory, Computation and Applications*, John Wiley and Sons, New-York, 1985.

[63] I.M. $STANCU-MINASIAN$; 1984 ; *Stochastic Programming with Multiple Objective Functions* ; D.Reidel Publishing Company, Dordrecht.Optimization, $Cluj-Napoca$, pp. 321328.

[64] J. Spronk, *Interactive Multiple Goal Programming : Applications to Financial Planning*, Martinus Nijhoff Publishing, Boston, 1981.

[65] J.Teghem, D.Dufrane, M.Thauvoye, P.L.Kunsch, *STRANGE : Interactive method for multiobjective linear programming under uncertainty* ; European Journal of Operational Research 26 (1), $65 - 82$, 1986.

[66] J. Teghem and P.L. Kunsh, *A Survey Of Techniques For Finding Efficient Solutions To Multi-Objective Integer Linear Programming*, Asia-Pacific Journal of Operational Research 3 95-108, 1986.

[67] J.Teghem, $STRANGE-MOMIX$; *An interactive method for mixed integer linear programming* ; in : R. Slowinski, J.Teghem(Eds.),Stochastic Versus Fuzzy Apprroaches to Multiobjective Mathematical Programming Under Uncertainty,Kluwer Academic Publishers,Dordrecht,$101 - 115$; 1990.

[68] E.L. Ulungu and J. Teghem, *Multi-objective combinatorial optimization : a survey*, Journal of Multi-Criteria Decision Analysis, 3,83-104, 1994.

[69] B. Urli, R. Nadeau, *Multiobjective stochastic linear programming with incomplete information : A general methodology*, In : R. Slowinski, J. Teghem (Eds.), 131161, 1990.

[70] R.Van Slyke, R.J-B. Wets, *L-shaped linear programs with applicatios to optimal control and stochastic programming*, SIAM Journal on Applid Mathematics 17,638-663, 1969

[71] R.J.B Wets. *Programming under uncertainty : the equivalent convex program.* SIAM Journal on Applied Mathematic, 14 :$89 - 105$, 1966.

[72] W.T.Ziemba. *Stochastic programs with simple recourse.* In P.L.Hammer and G.Zoutendijk, editors, Mathematical Programming in Theory and Practice, pages $213 - 273$. $North - Holland$ Publishing Company, Amsterdam, 1974.