République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A. Mira de Béjaïa

Faculté des Sciences Exactes

Département de Recherche Opérationnelle

# *Mémoire de Magister*

En

Mathématiques Appliquées

Option

Modélisation Mathématique et Techniques de Décision

# Thème

## *Méthode de Support pour la Résolution des Problèmes de Programmation Linéaire en Nombres Entiers et Mixtes*

Présenté par :

M^elle^ H. Boussouira

Devant le jury composé de :

| | | | | |
|---|---|---|---|---|
| Président | M$^r$ Djamil | Aïssani | Professeur | Université de Béjaïa |
| Rapporteur | M$^r$ Mohand Ouamer | Bibi | Professeur | Université de Béjaïa |
| Examinateur | M$^r$ Mohamed Saïd | Radjef | Professeur | Université de Béjaïa |
| Examinateur | M$^r$ Mohand | Ouanes | Maître de conférence | Université de Tizi Ouzou |

Béjaïa, 2009.

# Table des matières

# *Remerciements*

Je tiens à exprimer ici ma gratitude au professeur M.O. Bibi pour l'honneur qu'il m'a fait en assurant la direction du présent mémoire. Je le remercie pour ses précieux conseils et orientations.

Je suis heureuse de pouvoir remercier Mr. D. Aissani pour avoir accepté de présider le jury de soutenance.

Toute ma gratitude va également à Messieurs : Mr. M.S. Radjef et Mr. M. Ouanes pour avoir accepté de juger ce travail.

Enfin, du fond du coeur, je remercie mes amis et toute ma famille, en particulier mes parents.

# Introduction générale

La recherche opérationnelle s'occupe principalement à résoudre des problèmes d'optimisation au moyen de techniques souvent conjuguées à un formalisme précis du problème. En particulier, les techniques de programmation linéaire en nombres entiers et mixtes (PLE, PLM) sont étroitement liées à la modélisation des problèmes dont les variables de décision sont des vecteurs où certaines de ses composantes sont astreintes à prendre des valeurs entières. Malgré cette restriction, la programmation linéaire mixte permet de modéliser une grande variété de problèmes d'optimisation dans des domaines d'application extrêmement divers : logistique et transport, productique, réseaux de télécommunications ou de distribution d'énergie, automatique, ordonnancement de machines, problème de planification, problème de localisation etc...
Il s'agit là d'un des domaines les plus riches et les plus actifs de la programmation mathématique, et le volume de publications et de recherches qui lui a été consacré depuis les premiers travaux de Gomory en 1958 ([62], [65]) atteste de la difficulté du sujet et de l'importance de ses applications.

Les problèmes de programmation linéaire en nombres entiers et mixtes, à première vue très voisins des problèmes de programmation linéaire $(PL)$, sont d'une difficulté surprenante. Du point de vue de la complexité, les problèmes en nombres entiers et mixtes appartiennent à la classe des problèmes NP-difficiles (Karp 1978, Garey et Jehnson, 1979).

Les principales méthodes de résolution pour ce type de problèmes peuvent se ranger en quatre groupes : résolution comme un programme linéaire ordinaire, méthodes de coupes, méthodes arborescentes et méthodes de décomposition.

La résolution comme un $(PL)$ ordinaire est possible pour les $(PLE)$ équivalents à leurs $(PL)$ relaxés, c'est-à-dire que le sommet optimal du programme linéaire relaxé est à coordonnées entières. Il y a un cas particulier où, pour tout vecteur entier $b$, l'intégralité du polyèdre $X = \{x \in \mathbb{R}^n : Ax \leq b\}$ est assurée lorsque la matrice $A$ est totalement unimodulaire, c'est-à-dire quand les sous-déterminants de $A$ valent $-1,\ 0$ ou $1$. Durant les 40 dernières années, une profonde théorie sur les matrices totalement unimodulaires a immergé. Dans le cas où la matrice $A$ n'est pas totalement unimodulaire, l'intégralité de la solution du programme linéaire relaxé est très rare.

La faiblesse de la relaxation continue est dûe à la faiblesse de la formulation du problème, à savoir que $X$ est large par rapport à $convS$, qui est le plus petit polyèdre contenant toutes les solutions entières du $(PLE)$, $S$ étant l'ensemble des points entiers de $X$. Pour améliorer la relaxation continue, on peut ajouter au programme linéaire des inégalités linéaires qui coupent $X$. Cette technique permet ainsi de resserrer la formulation du problème ; on résout alors : $\max\{c'x :\ x \in X'\}$ avec $S \subseteq convS \subseteq X' \subseteq X$, en prenant implicitement en compte les contraintes d'intégralité. Idéalement, on cherche à déterminer des inégalités définissant des facettes de $convS$, c'est-à-dire des plans supports de $convS$ de dimension égale à $dim(convS) - 1$.
Un tel algorithme est appelé *algorithme de coupes ou de plans sécants*, puisqu'il s'agit de partir d'un point $x$ de l'espace, généralement une solution optimale du programme linéaire relaxé $\max\{c'x :\ x \in X\}$, puis, tant que $x$ n'appartient pas à $convS$, "couper" $X$ en ajoutant une ou plusieurs inégalités au programme linéaire qui ne sont pas vérifiées par la solution courante $x$. Les inégalités ajoutées sont appelées des *inégalités valides* (pour toute solution de $S$) ou, plus précisément, des *coupes*. Les plans engendrés séparent le point $x$ de $X$ du polyèdre $convS$ et la recherche de la nouvelle solution est effectivement restreinte à la partie de l'espace qu'ils définissent.

Les plans sécants générés par ces algorithmes peuvent êtres classés en deux grandes catégories : la première comprend les coupes générales qui sont valides pour tout problème $(PLM)$, elle inclut les coupes mixtes entières de Gomory (Balas et al., 1996 [13] ; Gomory, 1960 [63] ; Chvátal (1973)[31] et Schrijver (1980)[116]) et les coupes d'arrondis en nombres entiers et mixtes (MIR cuts) (Marchant and Wolsey, 2001 [95] ; Nemhauser et Wolsey, 1990 [106]). La seconde catégorie inclut des coupes particulières qui exploitent la structure spécifique du problème, telles que les coupes pour le problème du sac-à-dos (Balas, 1975 ; Crowder, Johnson et Padberg 1983 [36] ; Gu, Nemhauser et Savelsbergh, 1998 [69]).

Les méthodes arborescentes, quant à elles, sont largement utilisées, et ont permis de résoudre avec succès certains problèmes combinatoires difficiles tels que : le problème du voyageur de commerce (Held et Karp 1971, Hansen et Krarup 1974, Crowder et Padberg

1980), les problèmes d'ordonnancement (Fisher 1976), les problèmes de localisation et de classification (Mulvey et Crowder 1979, Arditi et Minoux 1981) et le problème d'affectation généralisée (Roos et Soland 1975, Legendre et Minoux 1977). Le principe de ces méthodes est de choisir une variable $x$ et de séparer le problème en sous-problèmes selon les valeurs de $x$, on peut dire qu'il s'agit d'une "exploration intelligente " du domaine des solutions réalisables du $(PLM)$ considéré.

Le principe des méthodes arborescentes est dû principalement à Land et Doig (1960) [85], Bertier et Roy (1964), Roy, Bertier et Nghiem (1965), Dakin (1965) [37] , Hervé (1967). Elles ont ensuite été utilisées et perfectionnées par de nombreux auteurs. On trouve également de bons exposés de synthèse sur la théorie et les applications dans Lawler et Wood (1966), Mitten (1970), Geoffrion et Marsten (1972), Hansen (1975), Linderoth et Savelsbergh (1999) et Achterberg 2007 [3].

L'inconvénient des méthodes de coupes est qu'elles fournissent souvent une solution réalisable uniquement à la fin, contrairement aux méthodes arborescentes qui livrent en cours de recherche une suite de bonnes solutions approchées.

Lorsque les classes de facettes connues ne décrivent pas de façon suffisamment fine le polyèdre des solutions entières, le processus ne se termine pas nécessairement par la mise en évidence d'une solution entière (optimale). Dans ce cas, on obtient des bornes (supérieures si l'on maximise) qui peuvent être utilisées pour guider une procédure de recherche arborescente (de type branch-and-bound). Cette combinaison des méthodes de coupes avec des méthodes arborescentes, dénommée " Branch and Cut ", est considérée aujourd'hui comme l'approche la plus efficace pour obtenir l'optimum exact de problèmes combinatoires difficiles. Elle a été utilisée avec succès, en particulier, sur les problèmes du voyageur de commerce de grande dimension.

Durant les quatre dernières décades, les problèmes de programmation en nombres entiers et mixtes ont été intensivement étudiés par de nombreux chercheurs : Hu 1969, Garfinkel et Nemhauser 1972, Zionts 1974, Salkin 1975, Taha 1975, Schrijver 1988, Nemhauser and Wolsey 1988, Parker et Rardin 1988.

Plusieurs livres et articles sur la programmation en nombres entiers et en optimisation combinatoire sont apparus dans les années 80 : $[108, 66, 116, 68, 105, 128]$.

Ce mémoire comporte une introduction, trois chapitres et une conclusion.

– Dans le premier chapitre, on présente certains fondements théoriques de la programmation linéaire. On y trouve, entre autres, les notions et résultats concernant les espaces vectoriels, les théorèmes de séparation des ensembles convexes et des résultats concernant les cônes polyédriques convexes. Aussi on passe en revue quelques résultats sur les polyèdres combinatoires et leurs applications en optimisation.

- Dans le second chapitre, on présente une synthèse des méthodes de résolution utilisées dans le domaine de la programmation linéaire mixte, en s'efforçant de faire ressortir les idées fondamentales qui sous-tendent la plupart des algorithmes actuellement connus. Nous commençons par étudier les deux principales familles de méthodes de résolution : les méthodes de recherche arborescente et les méthodes de coupes.
- Dans le dernier chapitre, on propose une méthode de plans sécants pour la résolution de programmes généraux en nombres entiers prenant mieux en compte les structures des problèmes que les méthodes actuelles, cette méthode proposée étant basée sur la méthode de support.
- Enfin, on clôture ce mémoire par une conclusion générale et quelques directions de recherche.

# Chapitre 1

# Rappels sur l'algèbre linéaire et la géométrie des polyèdres convexes

Dans ce chapitre, on présente certains fondements théoriques de la programmation linéaire. On y trouve, entre autres, les notions et résultats concernant les espaces vectoriels, les théorèmes de séparartion des ensembles convexes et des résultats concernant les cônes polyédriques convexes.

## 1.1 Rappels sur la géométrie de $\mathbb{R}^n$

### 1.1.1 Espaces vectoriels

Soit un espace vectoriel $E$, défini sur $\mathbb{R}$. Ses éléments sont des points de $E$.

- L'expression $\sum_{j=1}^{k} \lambda_j x^j$ est dite combinaison linéaire des vecteurs $x^1, \ ..., \ x^k$.
- Les vecteurs $x^1, \ ..., \ x^k$ sont dit ***linéairement dépendants*** s'il existe $k$ scalaires réels, $\lambda_1, \ ..., \ \lambda_k$, non tous nuls, tels que $\lambda_1 x^1 \ + \ ... \ + \ \lambda_k x^k = 0$. Autrement, ils sont dits ***linéairement indépendants***.
- La dimension de $E$ est le nombre maximal de vecteurs linéairement indépendants dans $E$. Si $E$ est de dimension $n$, une base de $E$ est un ensemble de $n$ vecteurs linéairement indépendants de $E$.

### 1.1.2 Distance, Boule, Voisinage et Ensemble Ouvert

**Définition 1.1.1.** ***Norme***

On appelle *norme* sur un espace vectoriel $E$ défini sur $\mathbb{R}$, une fonction de $E$ dans $\mathbb{R}$ qui, à tout $x \in E$, fait correspondre un scalaire non négatif, noté $\|x\|$, et satisfaisant :

1. $\|x\| = 0$ si et seulement si $x = 0$,
2. $\|x + y\| \leq \|x\| + \|y\|$ pour tout $x, \ y \in E$,
3. $\|\alpha x\| = |\alpha| \cdot \|x\|$, pour tout $\alpha \in \mathbb{R}$, $x \in E$.

Un espace vectoriel $E$ muni d'une norme est dit ***normé***.

**Définition 1.1.2.** ***Boule ouverte***

Dans un espace normé $E$, on appelle *boule ouverte* de centre $x^0$ et de rayon $r > 0$, l'ensemble $B_r(x^0) = \{x \in E : \| x - x^0 \| < r\}$. Dans $\mathbb{R}^2$ et $\mathbb{R}$, on emploie les mots ***disque ouvert*** et ***intervalle ouvert*** respectivement.

**Définition 1.1.3.** ***Voisinage***

Soit $E$ un espace normé et $x \in E$. On appelle ***voisinage*** de $x$ tout sous-ensemble, noté $V(x)$, contenant une boule ouverte $B_r(x)$.

**Définition 1.1.4.** ***Point intérieur***

Soit $S$ un sous-ensemble de $E$. Un point $p \in S$ est un ***point intérieur*** à $S$ si $S$ est un voisinage de $p$, c'est-à-dire, s'il existe une boule ouverte $B_r(x) \subset S$.

**Définition 1.1.5.** ***Intérieur***

L'***intérieur*** $S^0$ d'un ensemble $S$ est l'ensemble de tous les points qui sont intérieurs à $S$. On a évidemment $S^0 \subset S$.

**Définition 1.1.6.** ***Extérieur***

L'***extérieur*** d'un sous-ensemble $S \subset E$ est l'intérieur du complément de $S$.

**Définition 1.1.7.** ***Point frontière***

Soit $p \in E$ et $S \subset E$. On dit que $p$ est un ***point frontière*** de $S$ s'il n'est ni intérieur ni extérieur à $S$. Un point frontière de $S$ peut appartenir ou non à $S$.

**Définition 1.1.8.** ***Frontière***

La ***frontière*** d'un sous-ensemble $S \subset E$ est l'ensemble des points-frontières de $S$.

**Définition 1.1.9.** ***Point adhérent***

Soit $p \in E$ et $S \subset E$. On dit que $p$ est un ***point adhérent*** de $S$ si tout voisinage de $p$ rencontre $S$.

**Définition 1.1.10.** ***Adhérence***

L'***adhérence*** d'un sous-ensemble $S \subset E$ est l'ensemble des points adhérents de $S$, noté $\bar{S}$. On rencontre parfois le terme ***fermeture*** pour adhérence.

Dans ce qui suit, on prend $E = \mathbb{R}^n$.

**Définition 1.1.11.** Le produit scalaire de deux vecteurs $x$ et $y$ dans l'espace vectoriel $E = \mathbb{R}^n$ est un nombre réel noté $\langle x,\ y\rangle$, défini par :

$$\langle x,\ y\rangle = \sum_{i=1}^{n} x_i y_i.$$

## 1.2 Matrices

**Définition 1.2.1.** Soit $n, m \in \mathbb{N}^*$. Une matrice d'ordre $m \times n$ à coefficients dans $\mathbb{R}$ est un tableau à deux dimensions, ayant $m$ lignes et $n$ colonnes, représentée sous la forme suivante :

$$A \ = \ A(I,J) = (a_{ij},\ i \in I, j \in J) = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix},$$

où $I = \{1,\ 2,\ ...,\ m\}$ et $J = \{1,\ 2,\ ...,\ n\}$ représentent respectivement l'ensemble des indices des lignes et des colonnes de $A$.

Pour des calculs pratiques, la matrice $A$ se note aussi

$$A = (a_1,\ a_2,\ ...,\ a_j,\ ...,\ a_n) = \begin{pmatrix} A_1' \\ A_2' \\ \vdots \\ A_i' \\ \vdots \\ A_m' \end{pmatrix},$$

où

$$a_j = A(I,j) = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{pmatrix}$$

est un vecteur-colonne de dimension $m$, considéré comme une matrice d'ordre $m \times 1$, $A_i' = A(i,J) = (a_{i1},\ a_{i2},\ ...,\ a_{in})$ est un vecteur-ligne de dimension $n$, considéré comme une matrice d'ordre $1 \times n$.

Le symbole ($'$) est celui de la transposition. Chaque vecteur, noté $x = x(J) = (x_j, j \in J)$, sera ainsi considéré comme un vecteur-colonne tandis que le vecteur-ligne sera noté $x'$. La matrice transposée de $A$ sera notée

$$A' = A'(J, I) = (a_{ji}, j \in J, i \in I).$$

La matrice $A$ est dite carrée si on a $n = m$ ; de plus, si $A = A'$, la matrice est dite symétrique. La matrice identité d'ordre $n$ sera notée $I_n$.

## Rang d'une matrice

**Définition 1.2.2. Rang d'une matrice**
Soient $A \in \mathbb{R}^{m \times n}$ une matrice et

$$ImA = \{y \in \mathbb{R}^m : \ \exists \ x \in \mathbb{R}^n, \ y = Ax\}$$

le sous-espace engendré par les vecteurs colonnes de $A$. Le rang de $A$ est la dimension de ce sous-espace.

**Définition 1.2.3.** Le nombre maximum de colonnes (lignes) linéairement indépendantes de $A$ est le rang de $A$, noté $rangA$.

**Définition 1.2.4.** Une matrice carrée $B$ d'ordre $m$ est dite régulière si son rang est égal à $m$ ; dans ce cas, elle est inversible et la matrice inverse $B^{-1}$ vérifie $B^{-1}B = BB^{-1} = I_m$.

## Systèmes d'équations

Soit un système de $m$ équations à $n$ inconnues :

$$\sum_{j=1}^{n} a_{ij} x_j = b_i \quad i = \overline{1, \ m},$$

ou, en notation vectorielle, $Ax = b$, avec $b = (b_i, \ i \in I)$ et $m \leq n$. Si

- $rangA < rang(A, b)$ : le système est impossible et ne possède aucune solution.
- $rangA = rang(A, b) = n$ : le système possède une solution unique, cette solution est $x = A^{-1}b$.
- $rangA = rang(A, b) < n$ : le système possède une infinité de solutions.

## 1.3 Eléments d'analyse convexe

Par-delà la structure d'un ensemble, la notion d'ensemble convexe est de première importance. Dans cette section, on donne sa définition et certaines de ses caractérisations.

### Ensemble convexe

Un sous-ensemble $S$ d'un espace vectoriel est ***convexe*** si, quels que soient $x,\ y \in S$, l'intervalle $[x,\ y]$ est inclus dans $S$ ou, de façon équivalente :

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

Il est dit ***strictement convexe*** s'il est convexe et si sa frontière ne contient aucun segment de droite.

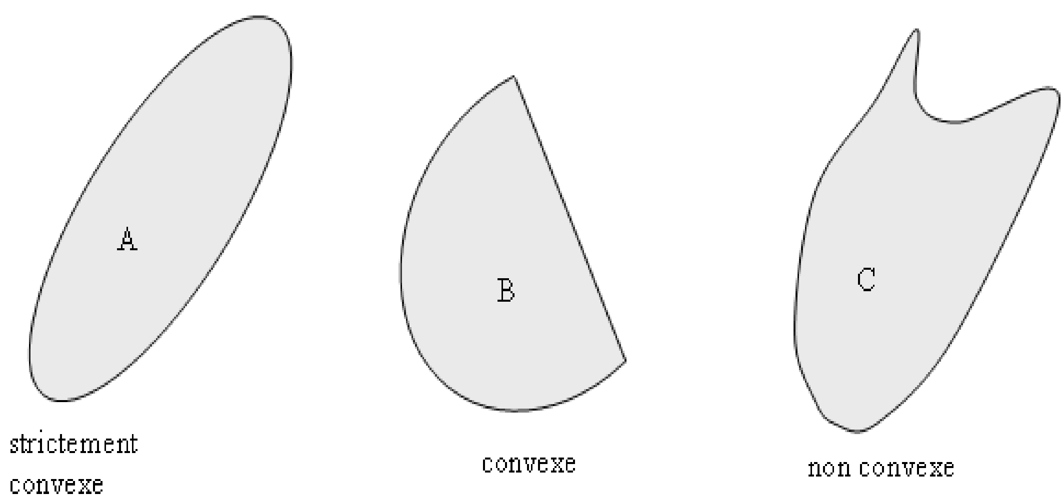


FIG. 1.1 – Exemples d'ensembles convexes et non convexes.

### Combinaison convexe

Soient $x^1,\ ...,\ x^k$ des vecteurs de $\mathbb{R}^n$. On dira qu'un vecteur $x$ *de* $\mathbb{R}^n$ est une combinaison convexe de $x^1,\ ...,\ x^k$ s'il existe $\lambda_1,\ ...,\ \lambda_k \in \mathbb{R}$ tels que

$$x = \sum_{j=1}^{k} \lambda_j x^j, \quad \sum_{j=1}^{k} \lambda_j = 1,\ \lambda_j \geq 0,\ j = \overline{1, k}.$$

**Théorème 1.3.1.** [122] *Un ensemble $S$ est convexe si et seulement si toute combinaison convexe finie de points de $S$ appartient à $S$.*

**Exemple 1.3.1.** Soit un ensemble $H = \{x : \; c'x = d\}$ de $\mathbb{R}^n$. Si $y \in H$ et $z \in H$, on vérifie que $x = \lambda y + (1-\lambda)z$ satisfait $c'x = d$ pour tout $\lambda$ tel que $0 \leq \lambda \leq 1$. Ainsi $H$ est convexe.

**Exemple 1.3.2.** Soit $H = \{x : \; c'x \leq d\}$ un demi-espace de $\mathbb{R}^n$. On vérifie de même que $H$ est convexe.

**Théorème 1.3.2.** [122] *Si $S_1$, ..., $S_r$ sont des ensembles convexes de $\mathbb{R}^n$, alors leur intersection est un ensemble convexe.*

**Exemple 1.3.3.** Soient les ensembles $S_i = \{x : \; \sum_{j=1}^{n} a_{ij}x_j \leq b_i\} \subset \mathbb{R}^n$, $i = \overline{1, \; m}$. Alors leur intersection :

$$S = \{x \; : \sum_{j=1}^{n} a_{ij}x_j \leq b_i; \; i = \overline{1, \; m}\} = \{x : Ax \leq b\},$$

est un ensemble convexe, où $A$ est une matrice d'ordre $m \times n$, formée des coefficients $a_{ij}$, $1 \leq i \leq m$, $1 \leq j \leq n$.
Ainsi, l'intersection d'une famille de demi-espaces de $\mathbb{R}^n$ est un ensemble convexe.

**Propriété 1.3.3.** [122] *Soient $S_1$ et $S_2$ deux ensembles convexes. Alors $S = \alpha_1 S_1 + \alpha_2 S_2$ est convexe pour tout $\alpha_1, \; \alpha_2 \in \mathbb{R}$.*

**Propriété 1.3.4.** [122] *Soit $S$ un sous-ensemble convexe d'un espace vectoriel. Alors, l'adhérence et l'intérieur de $S$, sont convexes.*

## Enveloppe convexe d'un ensemble

Si un ensemble $S$ n'est pas convexe, on peut alors définir un certain ensemble convexe qui le contient.
On appelle enveloppe convexe d'un ensemble $S$ l'intersection de tous les ensembles convexes qui contiennent $S$.
L'enveloppe convexe de $S$, notée ***convS***, est donc le plus petit ensemble convexe qui contient $S$.
Cette enveloppe convexe existe et elle est non vide pour tout ensemble non vide. On peut aussi caractériser l'enveloppe convexe par le théorème suivant :

**Théorème 1.3.5.** [23] *L'enveloppe convexe d'un ensemble $S$ de $\mathbb{R}^n$ est égale à l'ensemble de toutes les combinaisons convexes finies des points de $S$ :*

$$convS = \left\{ x \in \mathbb{R}^n : \ x = \sum_{i=1}^{m} \lambda_i x^i, \ x^i \in S, \ \sum_{i=1}^{m} \lambda_i = 1, \ \lambda_i \geq 0, \ i = \overline{1, \ m} \right\}.$$

**Théorème 1.3.6.** [23] *L'enveloppe convexe d'un ensemble compact (fermé et borné) est un compact.*

## Polyèdres convexes

**Définition 1.3.1.** Un ***polyèdre convexe*** $P$ est l'intersection (éventuellement vide) d'un nombre fini de demi-espaces fermés :

$$P = \{x \in \mathbb{R}^n : \ A_i' x \leq b_i, \ i = \overline{1, \ m}\}.$$

**Exemple 1.3.4.** L'ensemble $P = \{x \in \mathbb{R}^n : \ A_i' x = b_i, \ i = \overline{1, \ m}\}$ formé de l'intersection des hyperplans $A_i' x = b_i, \ i = \overline{1, \ m}$, est un polyèdre convexe, puisque, on a :

$$A_i' x = b_i \Leftrightarrow \left\{ \begin{array}{ll} & A_i' x \geq b_i, \\ et & \\ & A_i' x \leq b_i. \end{array} \right.$$

**Définition 1.3.2.** Un polyèdre convexe $P$ est borné s'il existe une valeur $u$ finie et positive telle que :

$$|x_j| \leq u \qquad \forall j = \overline{1, \ n}, \ \ \forall x \in P.$$

Un polyèdre convexe, borné et non vide, est dit ***polytope***.

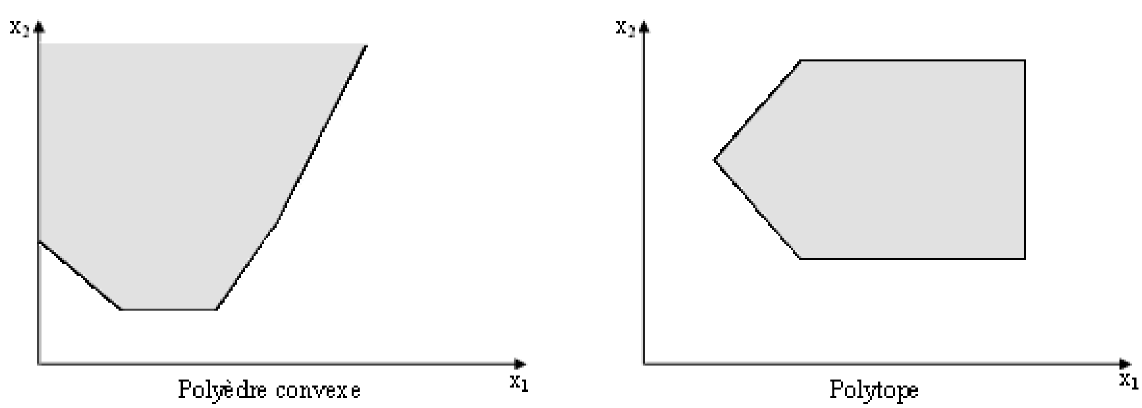


FIG. 1.2 – Polyèdre convexe et polytope dans $\mathbb{R}^2$.

**Définition 1.3.3.** Un polyèdre $P$ est dit ***rationnel*** s'il existe une matrice $(A, b)$ d'ordre $m \times (n+1)$ avec des coefficients rationnels, tel que : $P = \{x \in \mathbb{R}^n : \ Ax \leq b\}$.

**Définition 1.3.4.** Un polyèdre $P \subseteq \mathbb{R}^n$ est de pleine dimension si $dimP = n$.

**Propriété 1.3.7.** [121] *Soit* $P = \{x \in \mathbb{R}^n : \ A_i'x \leq b_i, \ i = \overline{1, \ m}\}$.

1. *Un point $x$ de $P$ est un point* **intérieur** *de $P$ si :*

$$A_i'x < b_i \quad \forall i = \overline{1, \ k} = m,$$

*si $k < m$, on parle de point intérieur relatif.*

2. *Un point $x$ de $P$ est un* **point frontière** *de $P$ si :*

$$\exists \ i, \ 1 \leq i \leq m : \ A_i' \ x = b_i.$$

3. *Si les $m$ hyperplans, définissant un polyèdre convexe, passent par un même point, le polyèdre convexe est appelé* **cône polyédrique convexe** *; en particulier, l'ensemble $\{x \in \mathbb{R}^n : \quad A \ x \leq 0\}$, où $A$ est une matrice d'ordre $m \times n$ est un cône polyédrique convexe.*

## 1.3.1 Inégalités Valides, Faces et Facettes

Soit $P = \{x \in \mathbb{R}^n : \ A_i'x \leq b_i, \ i = \overline{1, \ m}\}$.

**Définition 1.3.5.** ***Inégalité valide***

Une inégalité $\pi'x \leq \pi_0$ est une inégalité valide pour $P \subseteq \mathbb{R}^n$ si $\pi'x \leq \pi_0$ pour tout $x \in P$. Si $S = \{x \in \mathbb{Z}_+^n : \ Ax \leq b\}$ et $convS = \{x \in \mathbb{R}_+^n : \ \tilde{A}x \leq \tilde{b}\}$, les contraintes $A_i'x \leq b_i$ et $\tilde{A}_i'x \leq \tilde{b}_i$ sont clairement des inégalités valides pour $S$.

**Définition 1.3.6.** ***Face***

Les ensembles non vides de points de $P$ vérifiant un sous-ensemble des $m$ hyperplans

$$A_i'x = b_i, \qquad 1 \leq i \leq m,$$

sont appelés des ***faces*** de $P$.

Les faces constituent des polyèdres convexes. Si la face est définie par l'intersection de $r$ hyperplans linéairement indépendants, elle est dite de dimension $n - r$.

**Définition 1.3.7.**

- $F$ définit une face du polyèdre $P$ si $F = \{x \in P : \ \pi' x = \pi_0\}$ pour une certaine inégalité valide $\pi' x \leq \pi_0$ pour $P$.
- Si $F$ est une face de $P$, avec $F = \{x \in P : \ \pi' x = \pi_0\}$, l'inégalité valide $\pi' x \leq \pi_0$ représente ou définit la face.

**Théorème 1.3.8.** [96] *Un polyèdre a un nombre fini de faces.*

**Définition 1.3.8.** ***Facette***

$F$ est une facette de $P$ si $F$ est une face de $P$ et $dimF = dimP - 1$.

**Proposition 1.3.9.** [128] *Si $P$ est de pleine dimension, une inégalité valide $\pi' x \leq \pi_0$ est nécessaire pour la description de $P$ si et seulement si elle définit une facette de $P$.*

**Exemple 1.3.5.** Considérons le polyèdre $P \subset \mathbb{R}^2$, représenté dans la figure 1.3, décrit par les inégalités suivantes :

$$\begin{cases} x_1 & \leq 2, \\ x_1 + \ \ x_2 & \leq 4, \\ x_1 + 2x_2 & \leq 10, \\ x_1 + 2x_2 & \leq 6, \\ x_1 + \ \ x_2 & \geq 2, \\ x_1 & \geq 0, \\ \qquad x_2 & \geq 0. \end{cases}$$

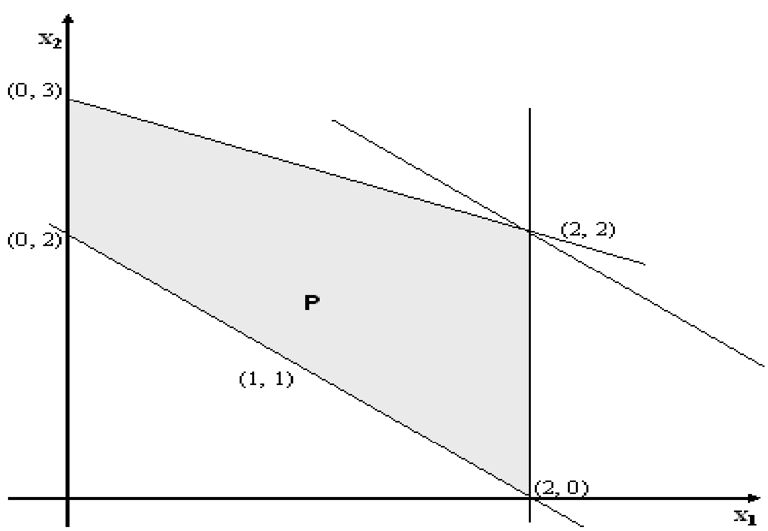


FIG. 1.3 – Faces et facettes d'un polyèdre.

L'inégalité $x_1 \leq 2$ définit une facette de $P$ ainsi que les inégalités $x_1 + 2x_2 \leq 6$, $x_1 + x_2 \geq 2$ et $x_1 \geq 0$. Par contre, l'inégalité $x_1 + x_2 \leq 4$ définit une face, constituée seulement d'un point $(2,\ 2)$ de $P$, et par conséquent elle est redondante. Alternativement, considérons les inégalités $x_1 \leq 2$ et $x_1+2x_2 \leq 6$ avec les poids $u = (\frac{1}{2},\ \frac{1}{2})$, aussi on montre que l'inégalité $x_1 + x_2 \leq 4$ est redondante. L'inégalité $x_2 \geq 0$ est la somme de l'inégalité $x_1 \leq 2$ et $-x_1 - x_2 \leq -2$ et ainsi elle est redondante.

La description minimale de $P$ est donnée par

$$\begin{cases} x_1 \qquad\quad \leq 2, \\ x_1 + 2x_2 \leq 6, \\ x_1 + \ \ x_2 \geq 2, \\ x_1 \qquad\quad \geq 0. \end{cases}$$

## 1.4 Fonction linéaire et fonction convexe

### Fonction linéaire

**Définition 1.4.1.** On dit qu'une fonction $f : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ est ***linéaire*** si, quel que soit $x,\ y \in \mathbb{R}^n$ et $\alpha,\ \beta \in \mathbb{R}$, on a $f(\alpha x + \beta y) = \alpha f(x) + \beta f(y)$.

**Propriété 1.4.1.** [122] *Soit $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ une fonction linéaire, $x^1,\ ...,\ x^m \in \mathbb{R}^n$ et $\alpha_1,\ ...,\ \alpha_m \in \mathbb{R}$. Alors*

$$f\left(\sum_{i=1}^{m} \alpha_i x^i\right) = \sum_{i=1}^{m} \alpha_i f(x^i).$$

**Définition 1.4.2.** On dit qu'une fonction $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ est ***affine*** si elle est la somme d'une fonction linéaire et d'une constante.

### Fonction convexe

**Définition 1.4.3.** Soit $S$ un sous-ensemble convexe de $\mathbb{R}^n$. Alors une fonction $f :\ S \longrightarrow \mathbb{R}$ est convexe si, $\forall\ x^1$ *et* $x^2 \in S$, $\forall \lambda \in [0,\ 1]$, l'inégalité suivante a lieu

$$f\left(\lambda x^1 + (1-\lambda)x^2\right) \leq \lambda f(x^1) + (1-\lambda) f(x^2).$$

D'autre part, une fonction $f$ est dite concave si $(-f)$ est convexe, autrement dit :

$$\forall x^1,\ x^2 \in S, \quad \forall \lambda \in [0,\ 1], \quad f\left(\lambda x^1 + (1-\lambda)x^2\right) \geq \lambda f(x^1) + (1-\lambda) f(x^2).$$

**Consequence 1.4.2.**

- *Toute fonction linéaire est à la fois convexe et concave.*
- *La somme de fonctions convexes est une fonction convexe.*
- *Une fonction vectorielle $F\ :\ S \longrightarrow \mathbb{R}^k$, où $F = (f_1,\ f_2,\ ...,\ f_k)$ est convexe si chacune des composantes $f_i$ est convexe.*

## 1.5 Séparation des ensembles convexes

### 1.5.1 Hyperplan dans l'espace $\mathbb{R}^n$

On appelle hyperplan dans l'espace $\mathbb{R}^n$ tout ensemble $H$ de dimension $(n-1)$ défini par :

$$H = \{x \in \mathbb{R}^n : a'x = \alpha\},$$

où $\alpha \in \mathbb{R}$ et $a$ un vecteur non nul appelé normale à $H$.

L'hyperplan $H$ sépare l'espace $\mathbb{R}^n$ en deux demi-espaces fermés $H^-$ et $H^+$ :

$$H^- = \{x \in \mathbb{R}^n : \ a'x \leq \alpha\}, \quad H^+ = \{x \in \mathbb{R}^n : \ a'x \geq \alpha\}.$$

Leurs demi-espaces ouverts correspondants sont :

$$H^{--} = \{x \in \mathbb{R}^n : \ a'x < \alpha\}, \quad H^{++} = \{x \in \mathbb{R}^n : \ a'x > \alpha\}.$$

**Remarque 1.5.1.** Ces différents demi-espaces sont des ensembles convexes.

### 1.5.2 Séparation de deux ensembles par un hyperplan

On dit qu'un hyperplan

$$H = \{x \in \mathbb{R}^n : a'x = \alpha\}$$

sépare deux ensembles non vides $S_1$ et $S_2$ si :

$$a'x \leq \alpha, \quad \forall x \in S_1; \ \ a'y \geq \alpha, \quad \forall y \in S_2.$$

Un tel hyperplan est alors dit ***Hyperplan séparateur*** de $S_1$ et $S_2$.

- Si un hyperplan séparateur $H$ ne contient pas simultanément les deux ensembles $S_1$ et $S_2$, on parle alors de séparation ***propre***. Dans le cas où $S_1 \cup S_2$ est inclus dans $H$, alors la séparation est dite ***singulière*** ou ***triviale***.

  La séparation de deux ensembles $S_1$ et $S_2$ veut dire que $S_1$ et $S_2$ se trouvent des deux cotés de l'hyperplan $H$, c'est-à-dire, $S_1 \subset H^-$, $S_2 \subset H^+$.
- L'hyperplan $H$ sépare strictement deux ensembles non vides $S_1$ et $S_2$ si :

$$a'x < \alpha, \quad \forall x \in S_1; \ \ a'y > \alpha, \quad \forall y \in S_2.$$

– La séparation de deux ensembles $S_1$ et $S_2$ par un hyperplan $H$ est dite ***forte***, s'il existe un nombre réel positif $\epsilon$ tel que :

$$a'x \leq \alpha, \quad \forall x \in S_1; \;\; a'y \geq \alpha + \epsilon, \quad \forall y \in S_2.$$

Il est clair qu'une séparation forte ou stricte induit toujours une séparation propre.

**Exemple 1.5.1.**

1. L'hyperplan $x = 1$ dans $\mathbb{R}$ sépare proprement les deux ensembles suivants :

$$S_1 = \{x \leq 1\}, \;\; S_2 = \{x \geq 1\}.$$

2. L'hyperplan donnée par $H = \{(x_1, x_2) \in \mathbb{R}^2 : \;\; x_2 = 0\}$ sépare fortement les deux ensembles suivant :

$$S_1 = \{(x_1,\ x_2) \in \mathbb{R}^2 :\ x_1 > 0,\ x_2 \leq 0\},\;\; S_2 = \{(x_1,\ x_2) \in \mathbb{R}^2 :\ x_1 > 0,\ x_2 \geq \frac{1}{x_1}+1\}.$$

3. L'hyperplan donné par $H = \{(x_1, x_2) \in \mathbb{R}^2 : \;\; x_2 = 0\}$ sépare strictement les deux ensembles suivant :

$$S_1 = \{(x_1,\ x_2) \in \mathbb{R}^2 :\ x_1 > 0,\ x_2 < 0\}, \;\; S_2 = \{(x_1,\ x_2) \in \mathbb{R}^2 :\ x_1 > 0,\ x_2 \geq \frac{1}{x_1}\}.$$

### 1.5.3 Hyperplan d'appui

$H$ est appelé ***hyperplan d'appui*** pour un ensemble $S$ au point $x^0$, où $x^0$ est un point frontière de $S$, si l'on a :

$$a'x^0 = \alpha \quad et \;\; a'x \leq \alpha, \;\; \forall x \in S.$$

**Théorème 1.5.1.** [125]

*Soit $S$ un ensemble convexe fermé et $x^0$ un point frontière de $S$, alors il existe un hyperplan d'appui de $S$ en $x^0$.*

**Remarque 1.5.2.**

– Un hyperplan $H$ est dit valide pour un polyèdre $P$ si $P \subseteq H^+$. Si $H$ est valide, alors $F = H \cap P$ est une face de $P$.

– Les facettes représentent les hyperplans d'appui d'un polyèdre.

### 1.5.4 Théorèmes de séparation des ensembles convexes

Lorsque deux ensembles $S_1$ et $S_2$ sont non vides, convexes et disjoints, alors il sont forcément séparables. Mais avant, considérons tout d'abord le problème de séparation d'un point $y$ avec un ensemble convexe $S$.

**Théorème 1.5.2.** [20] *Soit $S$ un ensemble convexe et $y$ un point de $\mathbb{R}^n$ n'appartenant pas à l'adhérance $\bar{S}$. Il existe alors un vecteur $a$, avec $\| a \|= 1$, et un nombre positif $\epsilon$ tel que :*

$$\langle a, x\rangle \leq \langle a, y\rangle - \epsilon, \quad \forall x \in S.$$

**Théorème 1.5.3.** [20] *Soit $x^0$ un point frontière d'un ensemble convexe $S$. Il existe alors un vecteur $a$, avec $\| a \|= 1$, tel que :*

$$\langle a, x\rangle \leq \langle a, x^0\rangle, \quad \forall x \in S.$$

**Remarque 1.5.3.** Dans le cas d'un ensemble ouvert, cette inégalité est stricte.

Passons maintenant au théorème de séparation de deux ensembles convexes.

**Théorème 1.5.4.** [20] *Soient $S_1$ et $S_2$ deux ensembles non vides convexes et disjoints. Il existe alors un vecteur $a$, $\| a \|= 1$, tel que :*

$$\langle a, x\rangle \leq \langle a, y\rangle, \qquad \forall x \in S_1 \text{ et } y \in S_2.$$

**Remarque 1.5.4.** Si l'un des deux ensembles $S_1$ ou $S_2$ est ouvert, alors la séparation est stricte :

$$\langle a, x\rangle < \langle a, y\rangle, \quad \forall x \in S_1 \text{ et } y \in S_2.$$

Si les ensembles convexes disjoints $S_1$ et $S_2$ sont tous les deux fermés et qu' au moins l'un d'entre eux est borné, alors le théorème suivant affirme que l'on peut les séparer fortement :

**Théorème 1.5.5.** [20] *Soit $S_1$ et $S_2$ deux ensembles convexes et fermés, dont l'intersection est vide. Supposons de plus que $S_1$ est borné, donc compact. Il existe alors un vecteur $a$, $\| a \|= 1$, et un nombre strictement positif $\epsilon$ tel que :*

$$\langle a, x\rangle \leq \langle a, y\rangle - \epsilon, \quad \forall x \in S_1 \text{ et } y \in S_2.$$

## 1.6 Cône polyédrique convexe et inéquations linéaires

**Définition 1.6.1. Cône**

Un ensemble $K$ de $R^n$ est un cône si $x \in K$ implique $\lambda x \in K$ pour tout $\lambda \geq 0$. Un cône est pointé s'il possède un point extrême.

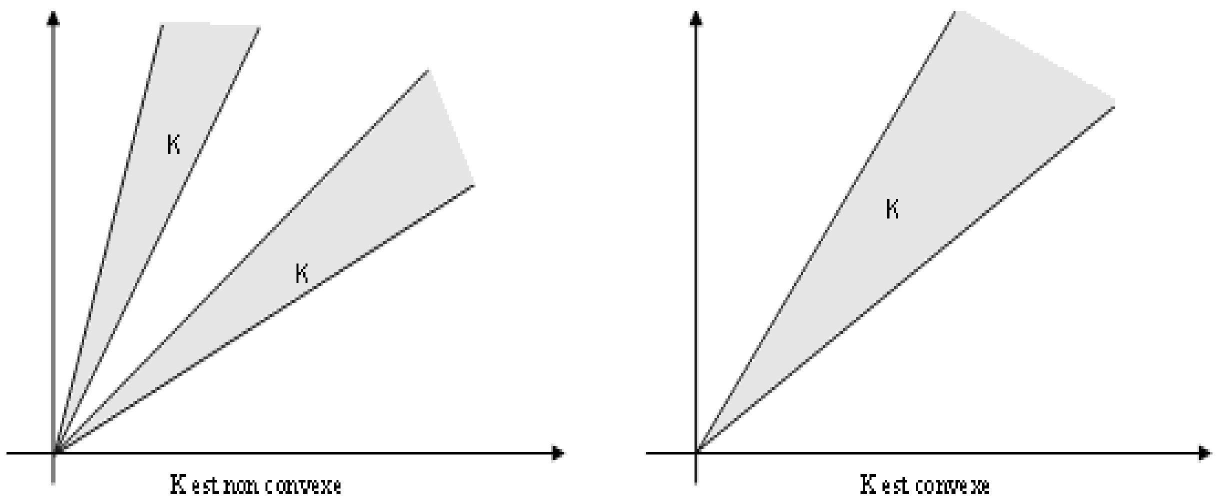


FIG. 1.4 – Exemple de cônes dans $\mathbb{R}^2$.

**Définition 1.6.2. Cône polyédrique convexe**

Soit $A_1, \ ..., \ A_m \in \mathbb{R}^n$. L'ensemble $K_t = \{x \in \mathbb{R}^n : \ A_i' x \leq 0, \ i = \overline{1, \ m}\}$ est un cône polyédrique convexe.

Autrement dit, l'intersection d'un nombre fini de demi-espaces fermés, dont les hyperplans générateurs passent par l'origine, est un cône polyédrique convexe.

En formant la matrice $A$ d'ordre $m \times n$, à l'aide des vecteurs lignes $A_1', \ ..., \ A_m'$, on peut aussi écrire $K_t = \{x \in \mathbb{R}^n : \ Ax \leq 0\}$.

Cette définition est dite ***tangentielle***. Il existe aussi une autre définition de cône polyédrique convexe qui est dite ***ponctuelle*** :

**Définition 1.6.3.** Soit $b^1, \ ..., \ b^k \in \mathbb{R}^n$. L'ensemble

$$K_p = \left\{ x \in \mathbb{R}^n : \ x = \sum_{i=1}^{k} \lambda_i b^i, \ \lambda_i \in \mathbb{R}, \ \lambda_i \geq 0, \ i = \overline{1, k} \right\},$$

est un cône polyédrique convexe. Autrement dit, un ensemble de combinaisons linéaires non négatives d'un nombre fini de points est un cône polyédrique convexe. En formant la matrice $B$ d'ordre $n \times k$ à l'aide des vecteurs colonnes $b^1$, ..., $b^k$ et le vecteur colonne $y$ avec les $\lambda_i$, on peut écrire

$$K_p = \{x \in \mathbb{R}^n : \ x = By, \ y \geq 0\}.$$

On dira alors que le cône $K_p$ est engendré par les rayons $b^i$, $i = \overline{1,k}$.

### Cône polaire

Soit $K$ un cône polyédrique convexe de $\mathbb{R}^n$. On appelle **cône polaire** de $K$, l'ensemble $K^+ = \{v \in \mathbb{R}^n : \ \langle v, x\rangle \leq 0, \ \forall x \in K\}$.

### Cône Dual. Théorème de Farkas [47]

On appelle dual du cône $K$, le cône $K^*$ constitué par l'ensemble des vecteurs qui font un angle aigu avec les vecteurs de $K$ :

$$K^* = \{v \in \mathbb{R}^n : \ \langle v, x\rangle \geq 0, \forall x \in K\}.$$

Le théorème de Farkas (1902) indique que le dual du dual d'un cône convexe est le cône inilial :

$$K^{**} = K.$$

## 1.7 Points extrêmes et rayons extrêmes

### Point extrême ou sommet

On dit que $x$ est un point extrême d'un ensemble convexe $P$, s'il n'existe pas de points $x^1, x^2$ de $P$, avec $x^1 \neq x^2$, tels que $x = \lambda x^1 + (1-\lambda)x^2$, où $0 < \lambda < 1$.

Autrement dit, $x$ est un point extrême de $P$ si on ne peut l'exprimer comme combinaison convexe de deux points différents de $P$. Un point extrême sera toujours un point frontière mais l'inverse n'est pas vrai.

**Proposition 1.7.1.** [96] *$x$ est un point extême de $P$ si et seulement si $x$ est une face de dimension 0 de $P$.*

Un ensemble convexe $P$ fermé peut n'avoir aucun point extrême ( par exemple si $P$ est un demi-espace fermé) ; il peut avoir un nombre fini ou infini de points extrêmes (un disque fermé de $\mathbb{R}^2$ a un nombre infini de points extrêmes).

**Théorème 1.7.2.** [23] **Existence d'un point extrême.** *Un ensemble non vide convexe $P$ de $\mathbb{R}^n$ fermé et borné a au moins un point extrême.*

**Définition 1.7.1. Solution de base.**

Soit $P = \{x \in \mathbb{R}^n : \ Ax = b, \ d^- \leq x \leq d^+\}$ un polytope, avec $A \in \mathbb{R}^{m\times n}$ et $b \in \mathbb{R}^m$ et $n \geq m$.

Soit un sous-ensemble d'indices $J_B \subset J$, $J$ étant l'ensemble des indices des colonnes de $A$, tel que $J = J_B \cup J_N, \ J_B \cap J_N = \emptyset$ *et* $|J_B| = m$.

- Un vecteur $x \in P$ est appelé solution réalisable de base s'il existe un sous-ensemble d'indices $J_B \subset J$ tels que : $detA_B = detA(I, J_B) \neq 0$ et $x_j = d_j^- \vee d_j^+ \ \forall \ j \in J_N$.
- Si de plus $d_j^- < x_j < d_j^+, \ \forall j \in J_B$, le vecteur $x$ est appelé solution réalisable de base non dégénérée.

**Théorème 1.7.3.** [23] **Identification des sommets**

*Soit $P = \{x \in \mathbb{R}^n : \ Ax = b, \ x \geq 0\}$ un polytope en forme standard, avec $A \in \mathbb{R}^{m\times n}$ de rang plein et $b \in \mathbb{R}^m$ et $n \geq m$. Soient $m$ colonnes de $A$ linéairement indépendantes. Appelons $A_B$ la matrice formée de ces $m$ colonnes (Sans perte de généralité, supposons que les $m$ colonnes choisies soient les $m$ premières. ) et $A_N$ la matrice formée des $n - m$ colonnes restantes, de telle sorte que*

$$A = (A_B|A_N).$$

*Soit le vecteur*

$$x = \begin{pmatrix} A_B^{-1}b \\ 0_{\mathbb{R}^{n-m}} \end{pmatrix}.$$

*Si $A_B^{-1}b \geq 0$, alors $x$ est un sommet de $P$.*

**Preuve.** Supposons par l'absurde qu'il existe $y, \ z \in P$, $y \neq x$, $z \neq x$ et $0 < \lambda < 1$ tels que

$$x = \lambda y + (1 - \lambda)z.$$

En décomposant, on obtient

$$x_B = \lambda y_B + (1 - \lambda)z_B$$

et

$$x_N = \lambda y_N + (1 - \lambda) z_N.$$

Comme $y$ et $z$ sont dans $P$, nous avons $y_N \geq 0$ et $z_N \geq 0$. Comme $0 < \lambda < 1$, la seule manière pour que $x_N = 0$ est que $y_N = z_N = 0$.

Donc,

$$y_B = A_B^{-1}(b - A_N y_N) = A_B^{-1} b = x_B$$

et

$$z_B = A_B^{-1}(b - A_N z_N) = A_B^{-1} b = x_B.$$

Nous obtenons $x = y = z$, ce qui contredit le fait que $y$ et $z$ sont différents de $x$, et ainsi le résultat est prouvé. $\square$

**Théorème 1.7.4.** [23] **Equivalence entre points extrêmes et solutions de base réalisables**

*Soit* $P = \{x \in \mathbb{R}^n : \; Ax = b, \; x \geq 0\}$ *un polytope en forme standard, avec* $A \in \mathbb{R}^{m \times n}$ *de rang plein et* $b \in \mathbb{R}^m$ *et* $n \geq m$.

*Le point* $x^* \in P$ *est un point extrême de* $P$ *si et seulement s'il est une solution de base réalisable.*

**Preuve.** *Implication directe.* Soit $x^*$ un sommet de $P$. Supposons que $x^*$ n'est pas une solution de base réalisable. Soit $m$ colonnes linéairement indépendantes de $A$, prises arbitrairement, de manière qu'elles forment une matrice inversible $A_B$, les $n - m$ colonnes qui restent forment une matrice $A_N$. Dès lors, $x^*$ peut être décomposé en une composante de base $x_B^*$ et une composante hors base $x_N^*$ telles que

$$x_B^* = A_B^{-1}(b - A_N x_N^*).$$

Comme $x^*$ n'est pas une solution de base réalisable, il existe au moins une composante de $x_N^*$ qui soit non nulle. Appelons la composante $x_k^*$. Construisons la direction $d = (d_B | d_N)$ telle que

$$d_B = -A_B^{-1} a_k,$$

où $a_k$ est la $k$ième colonne de $A$ et la composante hors base $d_N$ a toutes les coordonnées nulles, sauf la $k$ième qui vaut 1. Dès lors,

$$Ad = A_B d_B + A_N d_N = -A_B A_B^{-1} a_k + \sum_{j \in J_N} a_j d_j = -a_k + a_k = 0.$$

Ainsi, pour tout $\alpha$,

$$A(x^* + \alpha d) = Ax^* + \alpha Ad = Ax^* = b.$$

Comme $x_B^* > 0$ et $x_k^* > 0$, il est possible de choisir $\alpha_1 > 0$ et $\alpha_2 > 0$ suffisamment petits pour que $x^1 = x^* + \alpha_1 d$ et $x^2 = x^* - \alpha_2 d$ soient dans $P$. Soit

$$\lambda = \frac{\alpha_2}{\alpha_1 + \alpha_2}.$$

On a $0 < \lambda < 1$ et $x^* = \lambda x^1 + (1-\lambda)x^2$, ce qui contredit le fait que $x^*$ est un sommet du polytope. Donc $x_B^* = A_B^{-1}b \geq 0, \;\; x_N^* = 0$ et par conséquent $x^*$ est une solution réalisable de base réalisable.

*L'implication inverse* est prouvée par le théorème 1.7.3. $\square$

**Théorème 1.7.5.** [105] *Si* $P = \{x \in \mathbb{R}^n : \;Ax \leq b\} \neq \emptyset$, *rangA* $= n$, *et* $\max\{c'x : \; x \in P\}$ *est fini, alors il existe une solution optimale qui est un point extrême de* $P$.

## Rayon extrême

**Définition 1.7.2.** Soit $r \in \mathbb{R}^n$. On appelle ***rayon*** engendré par $r$, l'ensemble $\{x \in \mathbb{R}^n : \; x = \lambda r, \; \lambda \geq 0\}$.

**Définition 1.7.3.** Soit $P^0 = \{r \in \mathbb{R}^n : \; Ar \leq 0\}$. Si $P = \{x \in \mathbb{R}^n : \; Ax \leq b\} \neq \emptyset$, alors $r \in P^0 \setminus \{0\}$ est dit rayon de $P$.

Un point $r \in \mathbb{R}^n$ est un rayon de $P$ si et seulement si pour tout point $x \in P$, l'ensemble $\{y \in \mathbb{R}^n : \; y = x + \lambda r, \; \lambda \in \mathbb{R}_+\} \subseteq P$.

**Définition 1.7.4.** Un rayon $r$ de $P$ est un ***rayon extrême*** s'il n'existe pas de rayons $r^1, \; r^2 \in P^0$, $r^1 \neq \lambda r^2$ pour tout $\lambda \in \mathbb{R}_+$, tel que $r = \frac{1}{2}r^1 + \frac{1}{2}r^2$.

Une autre manière de caractériser les rayons extrêmes d'un polyèdre est donnée par la proposition suivante :

**Proposition 1.7.6.** [105] *Soit* $P \neq \emptyset$ *;* $r$ *est un rayon extrême de* $P$ *si et seulement si* $\{\lambda r : \; \lambda \in \mathbb{R}_+\}$ *est une face de dimension un de* $P^0$.

**Corollaire 1.7.7.** [105] *Un polyèdre possède un nombre fini de points extrêmes et de rayons extrêmes.*

**Théorème 1.7.8.** [105] *Si* $P = \{x \in \mathbb{R}^n : \ Ax \leq b\} \neq \emptyset$, $rangA = n$, *et* $\max\{c'x : \ x \in P\}$ *est non borné, Alors* $P$ *possède un rayon extrême* $r^*$ *avec* $c'r^* > 0$.

**Lemme 1.7.9.** [96] *(Farkas)*
*Soit* $P = \{x \in \mathbb{R}^n : \ Ax \leq b, \ x \geq 0\}$ *un polyèdre dans* $\mathbb{R}^n$. *Exactement une des deux affirmations suivantes est vérifiée :*

**i.** $P$ *n'est pas vide ;*

**ii.** *il existe* $y \in \mathbb{R}^m$, *tel que* $y \geq 0$, $y'A \leq 0$ *et* $y'b > 0$.

Les deux résultats suivants sont des théorèmes fondamentaux pour la résolution de problèmes combinatoires par le biais de la programmation linéaire, approche aussi appelée combinatoire polyédrique.

**Théorème 1.7.10.** [105] *(Minkowski)*
*Soit* $P = \{x \in \mathbb{R}^n : \ A\ x \leq b \ \ x \geq 0\}$ *un polyèdre dans* $\mathbb{R}^n$. *Si* $P$ *n'est pas vide et si* $rangA = n$, *alors* $P$ *est l'ensemble des points pouvant être obtenus par la somme d'une combinaison convexe de ses points extrêmes et d'une combinaison conique de ses rayons extrêmes, c'est-à-dire :*

$$P = \left\{ x \in \mathbb{R}^n : \ x = \sum_{k \in K} \lambda_k x^k + \sum_{j \in J} \mu_j r^j, \ \ \sum_{k \in K} \lambda_k = 1, \ \lambda_k \geq 0 \textit{ pour } k \in K \textit{ et } \mu_j \geq 0 \textit{ pour } j \in J \right\},$$

*où* $\{x^k\}_{k \in K}$ *est l'ensemble des points extrêmes de* $P$ *et* $\{r^j\}_{j \in J}$ *est l'ensemble de ses rayons extrêmes.*

**Preuve.** Soit

$$Q = \left\{ x \in \mathbb{R}^n : \ x = \sum_{k \in K} \lambda_k x^k + \sum_{j \in J} \mu_j r^j, \ \ \sum_{k \in K} \lambda_k = 1, \ \lambda_k \geq 0 \textit{ pour } k \in K \textit{ et } \mu_j \geq 0 \textit{ pour } j \in J \right\}.$$

Comme $x^k \in P$ pour $k \in K$, et $P$ est convexe, alors $x' = \sum_{k \in K} \lambda_k x_k \in P$ pour tout $\lambda$ qui satisfait $\sum_{k \in K} \lambda_k = 1, \ \lambda_k \geq 0$ pour $k \in K$. De même, comme $r^j, \ j \in J$, sont des rayons, $x' + \sum_{j \in J} \mu_j r^j \in P$ pour tout $\mu_j \geq 0$ pour $j \in J$. Par conséquent $Q \subseteq P$.

Maintenant supposons que $Q \subsetneqq P$, donc il existe $y \in P \setminus Q$. En d'autres termes, il n'existe pas de $\lambda$ et $\mu$ qui satisfont

$$\sum_{k \in K} \lambda_k x^k + \sum_{j \in J} \mu_j r^j = y,$$

$$-\sum_{k\in K}\lambda_k = -1, \quad \lambda_k \geq 0 \ \textit{ pour } k \in K, \ \mu_j \geq 0 \textit{ pour } j \in J.$$

Alors d'après le lemme de Farkas, il existe $(\pi,\ \pi_0) \in \mathbb{R}^{n+1}$ tel que $\pi' x^k - \pi_0 \leq 0$ pour $k \in K$, $\pi' r^j \leq 0$ pour $j \in J$ et $\pi' y - \pi_0 > 0$. Maintenant considérons le programme linéaire $\max\{\pi' x :\ x \in P\}$. S'il possède une solution optimale finie, d'après le théorème 1.7.5 l'optimum est atteint sur un point extrême. Cependant, $y \in P$ et $\pi' y > \pi' x^k$ pour tout point extrême $\{x^k\}_{k\in K}$, qui est une contradiction. D'autre part, si le programme linéaire est non borné, par le théorème 1.7.8, il existe un rayon extrême $r^j$ avec $\pi' r^j > 0$. A nouveau c'est une contradiction. Par conséquent $Q = P$. $\square$

Ce théorème montre en particulier que tout polytope est l'enveloppe convexe de ses points extrêmes, puisqu'un polytope ne possède pas de rayons.

**Corollaire 1.7.11.** *Tout point d'un polytope $P$ peut s'exprimer comme combinaison convexe des points extrêmes de $P$.*

La réciproque de ce théorème est aussi vraie, c'est-à-dire qu'étant donné deux ensembles $\{x^k\}_{k\in K}$ et $\{r^j\}_{j\in J}$ de points de $\mathbb{R}^n$, l'ensemble des points pouvant être obtenus par la somme d'une combinaison convexe des points de $\{x^k\}_{k\in K}$ et d'une combinaison conique des points de $\{r^j\}_{j\in J}$ est un polyèdre.

**Théorème 1.7.12.** [96]*(Weyl)*
*Soit $\{x^k\}_{k\in K}$ et $\{r^j\}_{j\in J}$ deux ensembles finis de points de $\mathbb{R}^n$ à coordonnées rationnelles. Alors,*

$$P = \left\{ x \in \mathbb{R}^n :\ x = \sum_{k\in K}\lambda_k x^k + \sum_{j\in J}\mu_j r^j, \quad \sum_{k\in K}\lambda_k = 1,\ \lambda_k \geq 0 \textit{ pour } k \in K \textit{ et } \mu_j \geq 0 \textit{ pour } j \in J \right\},$$

*est un polyèdre rationnel.*

Ce dernier théorème est important, puisqu'une de ses conséquences est que l'enveloppe convexe de tout ensemble fini de points rationnels de $\mathbb{R}^n$ est un polytope rationnel.

**Théorème 1.7.13.** [125] *Soit $P$ un ensemble convexe et $H$ un hyperplan d'appui de $P$ ; alors tout point extrême de $T = H \cap P$ est un point extrême de $P$.*

**Preuve.** Soit $x^0$ un point extrême de $T = H \cap P$ et supposons que $x^0$ ne soit pas un point extrême de $P$ ; alors on a $x^0 = \lambda x^1 + (1-\lambda)x^2$ avec $x^1 \neq x^2$, $0 < \lambda < 1$ et $x^1,\ x^2 \in P$.

Représentons l'hyperplan d'appui $H$ par l'ensemble $\{x : \; a'x = \alpha\}$ avec $P$ contenu dans le demi-espace $\{x : \; a'x \leq \alpha\}$ ; on a alors $a'x^1 \leq \alpha$ et $a'x^2 \leq \alpha$. Puisque $x^0 \in H$, on a $\alpha = a'x^0 = \lambda a'x^1 + (1-\lambda)a'x^2$ ; les inégalités précédentes doivent donc être des égalités. Ainsi $x^1$ et $x^2$ sont dans $H$ et donc dans $T = H \cap P$, $x^0$ n'est pas un point extrême de $T$ ; ce qui est une contradiction. Ainsi tout point extrême de $T$ est un point extrême de $P$.□

**Théorème 1.7.14.** [125] *Soit $P \in \mathbb{R}^n$ un ensemble convexe fermé borné et $f : \; P \longrightarrow \mathbb{R}$ convexe sur $P$, si $f$ a un maximum sur $P$, il est atteint en un point extrême de $P$.*

**Preuve.**

1. Supposons qu'il existe un point intérieur $x^0$ où le maximum de $f$ est atteint ; ainsi $f(x^0) \geq f(x)$ pour tout $x$ de $P$. Choisissons un $\epsilon > 0$ tel que $B_\epsilon(x^0) = \{x : \; \|x - x^0\| \leq \epsilon\} \subset P$. Considérons un point extrême $y$ de $P$ ($y$ existe d'après le théorème 1.7.2). Soit $x' = x^0 + \frac{\epsilon(x^0-y)}{\|x^0-y\|}$, puisque $x^0 = \lambda y + (1-\lambda)x'$ avec $0 < \lambda = \frac{\epsilon}{(\epsilon+\|x^0-y\|)} < 1$, $f(x^0) \leq \lambda f(y) + (1-\lambda)f(x')$. Ceci implique soit que $f(x')$ ou $f(y)$ est supérieur à $f(x^0)$ (mais ceci est impossible), soit que $f(y) = f(x') = f(x^0)$. Le maximum est donc atteint en un point extrême de $P$.

2. Supposons maintenant que le maximum ne soit atteint qu'en des points frontières de $P$ ; soit $x^*$ un tel point. Si $x^*$ est un point extrême, on a terminé ; sinon, on considère un hyperplan d'appui $H_1$ en $x^*$ ($H_1$ existe d'après le théorème 1.5.1), $T_1 = H_1 \cap P$ est de dimension $n-1$ au plus. La valeur maximum de $f$ sur $T_1$ est $f(x^*)$ car $x^* \in T_1$. Si $x^*$ est un point intérieur de $T_1$, alors, d'après 1, il existe un point extrême $x^1$ de $T_1$ avec $f(x^1) = f(x^*)$. Si $x^1$ est un point extrême de $T_1$, c'est un point extrême de $P$ d'après le théorème 1.7.13. Si $x^*$ est un point frontière de $T_1$ (mais pas un point extrême), on prend un hyperplan d'appui $H_2$ de $T_1$ en $x^*$ et l'on pose $T_2 = H_2 \cap T_1$, $T_2$ est de dimension $n-2$ au plus et $\max\limits_{x \in T_2} f(x) = f(x^*)$. On continue ainsi en réduisant la dimension d'au moins une unité à chaque étape. On s'arrêtera dès que $x^*$ sera un point intérieur ou un point extrême d'un $T_k$. Ceci se produira nécessairement après un nombre fini d'étapes puisque un $T_k$ de dimension 0 est un point extrême de $P$ d'après le théorème 1.7.13. □

**Remarque 1.7.1.** Dans le cas où $P$ est un polytope, la démonstration du théorème 1.7.14 est simplifiée. D'après le corollaire 1.7.11, tout $x$ de $P$ est une combinaison convexe de points extrêmes $x^1,\ x^2,\ ..., x^r$ de $P$. Or $P$ a un nombre fini de points extrêmes (c'est l'intersection d'un nombre fini de demi-espaces), d'où

$$f(x) = f\left(\sum_{i=1}^{r} \lambda_i x^i\right) \leq \sum_{i=1}^{r} \lambda_i f(x^i) \leq \sum_{i=1}^{r} \lambda_i \max_k f(x^k) = \max_k f(x^k).$$

## 1.8 Polyèdres dont les points extrêmes sont entiers

**Définition 1.8.1.** Un polyèdre $P = \{x \in \mathbb{R}^n :\ Ax \leq b\}$ non vide est dit entier si chacune de ses faces (non vide) contient un point entier.

Dans cette section on va caractériser une classe importante de polyèdres avec cette propriété.

Considérons un polyèdre $P$ de la forme :

$$P = \left\{x \in \mathbb{R}^n :\ A\ x\ =\ b,\ d^- \leq\ x\ \leq\ d^+\right\},$$

où $A$ est une matrice d'ordre $m \times n$, $b$ est un vecteur de dimension $m$, $d^-$ et $d^+$ sont des vecteurs de dimension $n$. On suppose que toutes les composantes de la matrice $A$ et des vecteurs $b,\ d^-$ et $d^+$ sont entières. On veut déterminer une condition sous laquelle les points extrêmes de $P$ possèdent des composantes toutes entières.

**Définition 1.8.2. Matrice unimodulaire**

On dit qu'une matrice carrée $A$ d'ordre $n$ à coefficients entiers est ***unimodulaire*** si son déterminant vaut $0,\ 1,\ ou\ -1$.

On dit qu'une matrice rectangulaire avec des composantes entières est ***totalement unimodulaire*** $(TU)$ si chacune de ses sous-matrices carrées est unimodulaire.

Il découle de cette définition que les coefficients $a_{ij}$ de la matrice $A$ ne peuvent valoir que $0,\ 1\ ou\ -1$.

Cette dernière condition n'est pas suffisante. Tester si une matrice quelconque est unimodulaire n'est pas facile, mais certaines classes de matrices $TU$ peuvent être détectées par des propriétés. Heller et Tomkins [108] ont ainsi proposé une condition suffisante pour qu'une matrice d'éléments dans $\{-1,\ 0,\ 1\}$ soit $TU$ :

– Chaque colonne ne contient pas plus de deux éléments non nuls.

– On peut partitionner ses lignes en deux sous-ensembles $L_1$ et $L_2$ tels que :

*a*) Si une colonne possède deux éléments non nuls de même signe, l'un d'entre eux appartient à l'ensemble $L_1$ et l'autre à l'ensemble $L_2$.

*b*) Si une colonne possède deux éléments non nuls de signe contraire, alors ils appartiennent tous deux soit à $L_1$ ou à $L_2$.

**Théorème 1.8.1.** [94] ***Hofman, Kruskal*** (1956)**.**
*Une matrice $A$ d'ordre $m \times n$ est totalement unimodulaire si et seulement si pour tout vecteur entier $b \in \mathbb{R}^m$, le polyèdre $\{x \in \mathbb{R}^n,\ Ax \leq b,\ x \geq 0\}$ est entier.*

**Proposition 1.8.2.** [105] *Soit $P$ un polyèdre de la forme*

$$P = \left\{x \in \mathbb{R}^n :\ Ax = b,\ d^- \leq x \leq d^+\right\},$$

*où $A$ est une matrice d'ordre $m \times n$, $b$ est un vecteur de dimension $m$, $d^-$ et $d^+$ sont des vecteurs de dimension $n$. Supposons que toutes les composantes de la matrice $A$ et des vecteurs $b$, $d^-$ et $d^+$ sont entières, et que la matrice $A$ est totalement unimodulaire. Alors tous les points extrêmes de $P$ sont à composantes entières.*

**Preuve.** Soit $x^0$ un point extrême de $P$. Considérons le sous-ensemble d'indices

$$J_B = \{j :\ d_j^- < x_j^0 < d_j^+\},$$

et sans perte de généralité, supposons que

$$J_B = \{1,\ ...,\ \bar{m}\},\quad \bar{m} \leq m.$$

Soit $A_B$ la matrice constituée des $\bar{m}$ premières colonnes de $A$ et soit $x_B^0$ le vecteur constitué des $\bar{m}$ premières composantes de $x^0$.
Notons que les $n - \bar{m}$ dernières composantes de $x^0$ sont égales à l'une ou à l'autre des composantes correspondantes de $d^-$ ou $d^+$, qui sont entières. Ainsi, le point extrême $x^0$ possède des composantes toutes entières si et seulement si le sous-vecteur $x_B^0$ possède des composantes entières.

D'après le théorème 1.7.4, $A_B$ possède des colonnes linéairement indépendantes, alors $x_B^0$ est l'unique solution du système d'équations

$$A_B y = \tilde{b},$$

où $\tilde{b}$ est le vecteur $b$ à qui on a soustrait le produit des dernières $n - \bar{m}$ colonnes de $A$ par les composantes correspondantes de $x^0$, donc $\tilde{b}$ possède des composantes entières. D'une manière équivalente, il existe une sous-matrice inversible $\bar{A}_B$ d'ordre $\bar{m} \times \bar{m}$ de $A_B$ et un sous-vecteur $\bar{b}$ de $\tilde{b}$ avec $\bar{m}$ composantes tels que

$$x_B^0 = \bar{A}_B^{-1}\bar{b}.$$

Les composantes de $x_B^0$ seront entières si on peut garantir que les composantes de la matrice inverse $\bar{A}_B^{-1}$ sont entières. De la règle de Cramer pour la résolution de système d'équations linéaires, chacune des composantes de l'inverse d'une matrice est une fraction avec une somme de produits des composantes de la matrice au numérateur et le déterminant de la matrice au dominateur. Comme par hypothèse, $A$ est totalement unimodulaire, la sous-matrice inversible $\bar{A}_B$ est unimodulaire, et sont déterminant est égal à 1 ou $-1$. d'où $\bar{A}_B^{-1}$ possède des composantes entières, ce qui implique que $x_B^0$ et donc $x^0$ possède des composantes toutes entières. $\square$

### 1.8.1 Systèmes totalement duaux entiers (TDE)

Un système est TDE si pour tout vecteur entier $c \in \mathbb{Z}^n$ tel que le programme linéaire $max\{c'x : \; Ax \leq b\}$ admet une solution optimale, alors le programme dual correspondant possède une solution optimale entière.

**Théorème 1.8.3.** [93] ***Edmonds, Giles, (1977)***
*Si $Ax \leq b$ est TDE et $b$ est entier, alors le polyèdre $\{x \in \mathbb{R}^n : \; Ax \leq b\}$ est entier.*

## 1.9 Partie entière et partie fractionnaire d'un nombre

**Définition 1.9.1.** Si $a$ est un scalaire quelconque, on désigne par :

- $[a]$ le plus grand entier inférieur ou égal à $a$, appelé partie entière de $a$.
- La différence entre un nombre $a$ et sa partie entière est appelée partie fractionnaire et se note $\{a\}$ : $\{a\} = a - [a]$.

  La partie fractionnaire d'un nombre est un réel positif ou nul et strictement inférieur à 1.

**Propriété 1.9.1.** *Les propriétés suivantes sont vraies :*

**(i)** $\{a+b\} = \{a\} + \{b\} - [\{a\} + \{b\}], \quad \forall a,\ b \in \mathbb{R}^2,$

**(ii)** $\{-a\} = \begin{cases} 0, & \text{\textit{Si} } a \in \mathbb{Z}\,; \\ 1 - \{a\}, & \text{\textit{Sinon.}} \end{cases}$

**(iii)** $\{ak\} = \{a\}k - [\{a\}k], \quad \forall a \in \mathbb{R},\ \forall k \in \mathbb{Z},$

**(iv)** $\{a+k\} = \{a\}, \quad \forall a \in \mathbb{R},\ \forall k \in \mathbb{Z}.$

**Preuve.**

**(i)** On a $\forall a,\ b \in \mathbb{R}^2$ :

$$\begin{aligned} \{a\} + \{b\} &= [\{a\} + \{b\}] + \{\{a\} + \{b\}\}, \\ \Longleftrightarrow \{\{a\} + \{b\}\} &= \{a\} + \{b\} - [\{a\} + \{b\}]. \end{aligned}$$

Comme on a : $\{\{a\} + \{b\}\} = \{a+b\}$, alors

$$\{a+b\} = \{a\} + \{b\} - [\{a\} + \{b\}].$$

**(ii)** Si $a \in \mathbb{Z}$ : $\{-a\} = 0$.

Sinon, on a

$$\begin{aligned} -a &= [-a] + \{-a\} \qquad (1), \\ a &= [a] + \{a\} \qquad (2). \end{aligned}$$

(1) + (2) donne : $[a] + [-a] + \{a\} + \{-a\} = 0$.

Comme on a : $[a] + [-a] = -1$, alors $-1 + \{a\} + \{-a\} = 0 \Rightarrow \{-a\} = 1 - \{a\}$.

$$\{-a\} = \begin{cases} 0, & \text{Si } a \in \mathbb{Z}\,; \\ 1 - \{a\}, & \text{Sinon.} \end{cases}$$

**(iii)** On a : $\forall a \in \mathbb{R},\ \forall k \in \mathbb{Z}$ :

$$\begin{aligned} \{a\}k &= [\{a\}k] + \{\{a\}k\}, \\ \Rightarrow \{\{a\}k\} &= \{a\}k - [\{a\}k]. \end{aligned}$$

Comme on a : $\{\{a\}k\} = \{ak\}$, alors :

$$\{ak\} = \{a\}k - [\{a\}k].$$

**(iv)** On a $\forall a \in \mathbb{R},\ \forall k \in \mathbb{Z}$ :

$$\begin{aligned} a + k &= [a+k] + \{a+k\}, \\ \Rightarrow \{a+k\} &= a + k - [a+k], \\ &= a + k - [a] - k, \ \ ([a+k] = [a] + k), \\ &= \{a\}. \quad \square \end{aligned}$$

## 1.10 Lien polyédral entre un programme linéaire et un programme linéaire en nombres entiers

On montre dans cette section qu'un programme en nombres entiers peut, en théorie, être réduit en un programme linéaire.

**Théorème 1.10.1.** [105] *Soit* $P = \{x \in \mathbb{R}^n_+ : \ Ax \leq b\} \neq \emptyset$ *et* $S = P \cap \mathbb{Z}^n$, *où* $(A,\ b)$ *est une matrice d'ordre* $m \times (n+1)$ *d'entiers. Alors les propriétés suivantes sont vérifiées :*

**i.** *Il existe un ensemble fini de points* $\{q^l\}_{l\in L}$ *de* $S$ *et un ensemble fini de rayons* $\{r^j\}_{j\in J}$ *de* $P$ *tels que :*

$$\left\{x \in \mathbb{R}^n_+ : \ x = \sum_{l\in L} \alpha_l q^l + \sum_{j\in J} \beta_j r^j, \ \sum_{l\in L} \alpha_l = 1, \ \alpha \in \mathbb{Z}^{|L|}_+, \ \beta \in \mathbb{Z}^{|J|}_+\right\} = S.$$

**ii.** *Si* $P$ *est un cône* $(b = 0)$, *il existe un ensemble fini de rayons* $\{v^h\}_{h\in H}$ *de* $P$ *tel que :*

$$\left\{x \in \mathbb{R}^n_+ : \ x = \sum_{h\in H} \gamma_h v^h, \ \gamma \in Z^{|H|}_+\right\} = S.$$

**Preuve.**

**i.** Soit $\{x^k \in \mathbb{R}^n_+ : \ k \in K\}$ un ensemble fini de points extrêmes de $P$ et soit $\{r^j \in \mathbb{R}^n_+ : \ j \in J\}$ un ensemble fini de rayons de $P$. Comme $P$ est un polyèdre rationnel, tout vecteur extrême de $P$ possède des coordonnées rationnelles. On a

$$P = \left\{x \in \mathbb{R}^n_+ : \ x = \sum_{k\in K} \lambda_k x^k + \sum_{j\in J} \mu_j r^j, \ \sum_{k\in K} \lambda_k = 1, \ \lambda_k \geq 0, \ k \in K \ et \ \mu_j \geq 0, \ j \in J\right\}.$$

Sans perte de généralité, on peut supposer que $\{r^j\}$ pour $j \in J$ sont des vecteurs entiers.

Soit

$$Q = \left\{x \in \mathbb{Z}^n_+ : \ x = \sum_{k\in K} \lambda_k x^k + \sum_{j\in J} \mu_j r^j, \ \sum_{k\in K} \lambda_k = 1, \ \lambda_k \geq 0, \ k \in K \ et \ 0 \leq \mu_j < 1, \ j \in J\right\}.$$

$Q$ est un ensemble fini, disons $Q = \{q^l \in \mathbb{Z}^n_+ : \ l \in L\}$, et $Q \subseteq S$. Maintenant observons que $x^i \in S$ si et seulement si $x^i \in \mathbb{Z}^n_+$ et

$$x^i = \left(\sum_{k\in K} \lambda^i_k x^k + \sum_{j\in J} (\mu^i_j - [\mu^i_j]) r^j\right) + \left(\sum_{j\in J} [\mu^i_j] r^i\right),$$
$$\sum_{k\in K} \lambda_k = 1, \ \lambda_k, \ \mu_j \geq 0, \ pour \ k \in K \ et \ j \in J. \tag{1.1}$$

Le premier terme de (1.1) est un point de $Q$, donc il existe $l(i) \in L$ tel que

$$x^i = q^{l(i)} + \sum_{j \in J} \beta_j^i r^j, \quad \beta_j^i = [\mu_j^i] \quad \textit{pour tout } j \in J. \tag{1.2}$$

Le résultat s'ensuit.

**ii.** Observons que si $P$ est un cône, alors $q^l \in S$ implique $\gamma q^l \in S$ pour tout $\gamma \in \mathbb{Z}_+$. Donc il suffit de prendre

$$\{v^h : \ h \in H\} = \{q^l : \ l \in L\} \cup \{r^j : \ j \in J\}$$

dans la partie $i$. $\square$

**Théorème 1.10.2.** [105] *Soit $P = \{x \in \mathbb{R}_+^n : \ Ax \leq b\}$, où $(A, b)$ est une matrice d'entiers d'ordre $m \times (n+1)$ et $S = P \cap \mathbb{Z}^n$, alors $convS$ est un polyèdre rationnel.*

**Preuve.** Comme tout point $x^i \in S$ peut être écrit sous la forme (1.2), toute combinaison convexe des points $\{x^i \in S, \ i \in I\}$ peut être écrit comme

$$\begin{aligned} x &= \sum_{i \in I} \gamma_i x^i = \sum_{i \in I} \gamma_i \left( q^{l(i)} + \sum_{j \in J} \beta_j^i r^j \right) \\ &= \sum_{l \in L} \left( \sum_{\{i \in I : l(i) = l\}} \gamma_i \right) q^l + \sum_{j \in J} \left( \sum_{i \in I} \gamma_i \beta_j^i \right) r^j \\ &= \sum_{l \in L} \alpha_l q^l + \sum_{j \in J} \beta_j r^j, \end{aligned}$$

où $\alpha_l = \sum_{\{i \in I : l(i) = l\}} \gamma_i \geq 0$ pour $l \in L$, $\sum_{l \in L} \alpha_l = \sum_{i \in I} \gamma_i = 1$, et $\beta_j = \sum_{i \in I} \gamma_i \beta_j^i \geq 0$ pour $j \in J$. Maintenant il s'ensuit que

$$convS = \{x \in \mathbb{R}_+^n : \ x = \sum_{l \in L} \alpha_l q^l + \sum_{j \in J} \beta_j r^j, \ \sum_{l \in L} \alpha_l = 1, \ \alpha_l, \ \beta_j \geq 0 \ \textit{pour } l \in L \textit{ et } j \in J\},$$

avec $q^l, \ r^l \in \mathbb{Z}_+^n$ pour $l \in L$ et $j \in J$. par conséquent, d'après le théorème 1.7.12, $convS$ est un polyèdre rationnel. $\square$

La preuve ci-dessus s'étend simplement au cas d'ensemble en nombres entiers et mixtes avec des données rationnelles.

Elle montre aussi que si $P \cap \mathbb{Z}^n \neq \emptyset$ alors les rayons extrêmes de $P = \{x \in \mathbb{R}_+^n : \ Ax \leq b\}$ et de $conv(P \cap \mathbb{Z}^n)$ coïncident.

Le théorème 1.10.2 suggère qu'on peut résoudre le programme en nombres entiers

$$(PLE) \qquad \max\{c'x : \ x \in S\} \ \text{ où } S = P \cap \mathbb{Z}^n,$$

en résolvant le programme linéaire

$$(CPLE) \qquad \max\{c'x : \ x \in convS\}.$$

Ce résultat important, mais élémentaire, est formulé par le théorème suivant.

**Théorème 1.10.3.** [105] *Etant donné* $S = P \cap \mathbb{Z}^n \neq \emptyset$, $P = \{x \in \mathbb{R}^n_+ : \ Ax \leq b\}$, *et* $c \in \mathbb{R}^n$, *il s'ensuit que :*

**a.** *La valeur de la fonction objectif de* $(PLE)$ *est non bornée si et seulement si la valeur de celle du* $(CPLE)$ *est non bornée.*

**b.** *Si* $(CPLE)$ *possède une valeur optimale finie, alors il possède une solution optimale (à savoir, un point extrême) qui est une solution optimale pour* $(PLE)$.

**c.** *Si* $x^0$ *est une solution optimale pour* $(PLE)$, *alors elle est aussi pour* $(CPLE)$.

**Preuve.** Soit $z^0$ et $z^*$ la valeur optimale de $(PLE)$ et $(CPLE)$, respectivement. Notons que $S \subseteq convS$ implique que

$$z^* \ \geq \ z^0. \tag{1.3}$$

**a.** L'inégalité (1.3) implique que si $z^0 = \infty$, alors $z^* = \infty$. D'autre part, si $z^* = \infty$, alors il existe un point extême entier $x^0 \in convS$ et un rayon $r \in \mathbb{Z}^n_+$ tels que $c'r > 0$ et $x^0 + \theta r \in convS$ pour tout $\theta \geq 0$. Mais $x^0 + \theta r \in S$ pour tout $\theta \in \mathbb{Z}_+$, ce qui implique que $z^0 = \infty$.

**b.** Comme $convS$ est un polyèdre, si $(CPLE)$ possède une solution optimale, celle-ci est atteinte en un point extrême, soit $x^0$. Comme $x^0 \in S$, alors $z^0 \geq c'x^0 = z^*$. d'après (1.3), $z^0 = z^*$.

**c.** S'ensuit de $a$ et $b$ et le fait que $x^0 \in convS$. $\square$

**Corollaire 1.10.4 (105).** $(PLE)$ *est soit non réalisable, non borné ou possède une solution optimale.*

Le théorème 1.10.3 montre qu'on peut résoudre le programme en nombres entiers $(PLE)$ en résolvant le programme linéaire $(CPLE)$. Mais, en général, on ne connait pas un ensemble d'inégalités linéaires définissant $convS$.

# Chapitre 2

# Programmation linéaire en nombres entiers et mixtes

## Introduction

Nous donnerons ici les idées de base de quelques techniques qui permettent de résoudre des problèmes de programmation linéaire où certaines variables sont astreintes à prendre des valeurs entières ; on parle alors de *programmation linéaire mixte*. Si toutes les variables sont à valeurs entières, on a un problème de *programmation linéaire en nombres entiers*. Il s'agit là d'un des domaines les plus riches et les plus actifs de la programmation mathématique, et le volume de publications et de recherches qui lui a été consacrées depuis les premiers travaux de Gomory (vers 1958) atteste de la difficulté du sujet et de l'importance de ses applications.

Les problèmes d'optimisation combinatoire surviennent typiquement sous la forme de programme linéaire en nombres entiers et mixtes :

$$(PLM) \qquad max\{c'x \; + \; d'y \; : \; Ax \; + \; Dy \leq b, \; x \geq 0 \; et \; entier, \; y \geq 0\},$$

où $A$ est une matrice réelle d'ordre $m \times n$, $D$ est une matrice réelle d'ordre $m \times p$, $b$ est un vecteur à $m$ composantes réelles, $c$ et $d$ sont des vecteurs à $n$ et $p$ composantes réelles respectivement. Si $n = 0$, alors on a un programme linéaire $(PL)$. Si $p = 0$, alors on a un programme linéaire en nombres entiers $(PLE)$. Fréquemment, les variables entières ou réelles sont bornées. Dans le cas où les variables entières sont astreintes à ne prendre que les valeurs 0 ou 1, on parle alors de programmation en variables bivalentes.

Les problèmes de programmation linéaire en nombres entiers et mixtes

$(PLE)$, $(PLM)$, à première vue très voisins des problèmes de programmation linéaire $(PL)$, sont d'une difficulté surprenante. Beaucoup de problèmes se formulent de manière naturelle sous forme de $(PLM)$. L'adjonction de la contrainte d'intégralité sur le vecteur $x$ accentue de manière fondamentale la difficulté du problème et aucune technique efficace n'est connue pour résoudre ce type de problèmes.

## 2.1 Particularité des modèles en variables entières et mixtes

La programmation linéaire en nombres mixtes permet de modéliser de très nombreuses contraintes complexes qui seraient intraitables sans variables entières. De nombreuses contraintes, en apparence non linéaires, peuvent être linéarisées grâce à des variables entières. Ces possibilités augmentent énormément le champ d'application de la programmation linéaire. Même si les programmes linéaires obtenus sont souvent difficiles à résoudre, la programmation linéaire en nombres mixtes est déjà très utile comme langage de modélisation : elle permet de décrire de façon concise des problèmes discrets d'optimisation.

### 2.1.1 Exemples de problèmes

**Exemple 2.1.1.** ***Problèmes avec quantités indivisibles***
C'est la première classe d'application qui vient à l'esprit. Il s'agit de problèmes où, soit les quantités produites, soit les quantités de facteurs à mettre en œuvre doivent être entières.

**Exemple 2.1.2.** ***Problèmes avec coûts fixes***
On veut représenter un coût de production qui est nul en l'absence de production et qui, dans le cas contraire, vaut la somme d'un coût fixe de production, noté $K$, et d'un coût proportionnel, le taux marginal étant $m$.
On veut donc pouvoir exprimer la fonction suivante :

$$c(x) = \begin{cases} 0, & \text{Si} \quad x = 0, \\ \text{K + mx}, & \text{Si} \quad x > 0, \end{cases}$$

où $x$ désigne le niveau de production.
La représentation mathématique de ce coût fixe nécessite l'ajout d'une variable indicatrice

d'une production positive :

$$y = \begin{cases} 1, & \text{si } x > 0, \\ 0, & \text{si } x = 0. \end{cases}$$

La modification de la fonction objectif en $c(x, y) = Ky + mx$ devient linéaire, avec l'ajout des contraintes suivantes : $x \leq My, \quad et \ y \in \{0, 1\}$, avec $M$ une borne supérieure sur la quantité produite $x$.

**Exemple 2.1.3.** ***Problèmes avec contraintes logiques***

Parfois des problèmes d'optimisation comportent une condition logique. Un exemple typique est celui des problèmes de gestion de projets avec contraintes disjonctives. Dans ces problèmes, on doit déterminer l'enchaînement des tâches d'un projet de manière à le réaliser dans le meilleur délai, et il se peut que deux tâches doivent être effectuées par la même équipe d'ouvriers, en mettant en œuvre la même machine. Les deux tâches ne peuvent donc avoir lieu simultanément, sans que l'on puisse dire laquelle doit être effectuée en premier lieu. Mathématiquement, on peut écrire ceci par la condition suivante :

$$\begin{cases} t_i + d_i \leq t_j, & \text{si i est réalisée avant j,} \\ t_j + d_j \leq t_i, & \text{si j est réalisée avant i,} \end{cases}$$

où $t_i$ est la variable indiquant le temps de début au plus tôt de la tâche $i$ et $d_i$ est sa durée.

Cette disjonction peut être résolue par la programmation mixte binaire. En effet, définissons la variable binaire $y_{ij}$, dont la valeur est 1 si la tache $i$ est réalisée avant la tâche $j$ et 0 si la tâche $j$ est réalisée avant la tâche $i$.

On remplace alors la condition de disjonction par les contraintes suivantes :

$$\begin{cases} t_i + d_i \leq t_j + M(1 - y_{ij}), \\ t_j + d_j \leq t_i + My_{ij}, \\ y_{ij} \in \{0, 1\}, \end{cases}$$

où M est une borne supérieure sur la date de fin des travaux.

**Exemple 2.1.4.** ***Respect d'un sous-ensemble de contraintes***

Supposons que dans une situation donnée, il est requis qu'uniquement $k$ contraintes parmi les $m$ contraintes d'un problème soient respectées. Supposons que les $m$ contraintes sont sous la forme :

$$g_i(x_1, \ ..., \ x_n) \leq b_i, \quad i = \overline{1, m}.$$

On définit la variable :

$$y_i = \begin{cases} 0, & \text{si la } i^{ème} \text{ contrainte est respectée,} \\ 1, & \text{si la } i^{ème} \text{ contrainte est violée.} \end{cases}$$

Pour que $k$ parmi $m$ contraintes soient respectées, il suffira de considérer :

$$\begin{cases} g_i(x_1,\ ...,\ x_n) \leq b_i\ +\ My_i,\ i = \overline{1, m},\ \text{ M suffisamment grand,} \\ \sum\limits_{i=1}^{m} y_i = m - k, \\ y \in \{0,\ 1\}^m. \end{cases}$$

### 2.1.2 Formulation des problèmes

Considérons la formulation mathématique initiale d'un problème $(PL)$ de programmation linéaire en variables continues :

$$(PL) \quad \begin{cases} z = c'x \longrightarrow \max, \\ x \in X, \end{cases}$$

où $X$ représente le polyèdre convexe des solutions admissibles dans $\mathbb{R}^n$ :

$$X = \{x :\ Ax \leq b,\ x \geq 0\}.$$

Pour que les calculs restent dans l'ensemble des rationnels, on suppose généralement les données $A$, $b$ et $c$ à valeur dans $\mathbb{Z}$.

Lorsque toutes les variables doivent être entières, le problème résultant, noté $(PLE)$ ou $(ILP)$ "Integer Linear Programming", est le problème général de la programmation linéaire totalement en variables entières :

$$(PLE) \quad \begin{cases} z = c'x \longrightarrow \max, \\ x \in S = X \cap \mathbb{Z}^n_+. \end{cases}$$

Si une partie seulement des variables doivent être entières, le problème résultant noté $(PLM)$ ou $(MILP)$ "Mixed Integer Linear Programming", est le problème général de la programmation linéaire mixte :

$$(PLM) \quad \begin{cases} z = c'x \longrightarrow \max, \\ x \in X, \\ x_j \in \mathbb{Z}_+,\ j = \overline{1,\ p},\ \text{ p<n.} \end{cases}$$

Ou, de manière équivalente, en faisant apparaître les deux types de variables dans la fonction objectif et les contraintes :

$$(PLM) \quad \begin{cases} z = c'x + d'y \longrightarrow \max, \\ Ax + Dy \leq b, \\ x \in \mathbb{Z}^p_+, \\ y \in \mathbb{R}^{n-p}_+. \end{cases}$$

**Illustration :**

Soit le problème $(PL)$ défini par

$$\left\{\begin{array}{rcrl} z = 3x_1 & + & 2x_2 \longrightarrow & \max, \\ -3x_1 & + & 4x_2 \leq 6 & (a), \\ 2x_1 & + & 2x_2 \leq 11 & (b), \\ 4x_1 & - & 5x_2 \leq 10 & (c), \\ x_1 & & \geq 0 & (d), \\ & & x_2 \geq 0 & (e), \end{array}\right.$$

et sa représentation dans $\mathbb{R}^2$ (voir figure 2.1). La solution optimale $x^*$ du problème (PL) est le point d'intersection $(\frac{25}{6},\ \frac{4}{3})$ des contraintes (b) et (c), correspondant à la valeur optimale $z^* = \frac{91}{6}$.

- Si les deux variables $x_1$ et $x_2$ doivent être entières, le problème $(PLE)$ correspond au domaine d'admissibilité formé des points de la figure 2.2. La solution optimale du problème $(PLE)$ est le point $(3,\ 2)$ correspondant à la valeur optimale $z^*_{PLE} = 13$.
- Si seule la variable $x_1$ doit être entière, le problème $(PLM)$ correspond au domaine d'admissibilité formé des cinq segments verticaux de la figure 2.3. La solution optimale du problème $(PLM)$ est le point $(4,\ \frac{3}{2})$ correspondant à la valeur optimale $z^*_{PLM} = 15$.

**Remarque 2.1.1.** Lorsque toutes les variables d'un problème doivent être entières, mais que tous les coefficients des contraintes ne le sont pas, les éventuelles variables d'écart ne sont pas soumises à être entières. L'introduction de ces variables d'écart fournit alors un problème mixte, dans le cas contraire on aura un problème en nombres entiers pur.

## 2.2 $(PLEs)$ avec matrices totalement unimodulaires

Un point de départ naturel dans la résolution d'un programme en nombres entiers

$$(PLE) \qquad \max\{c'x :\ Ax \leq b,\ x \in \mathbb{Z}^n_+\},$$

avec $(A,\ b)$ des données entières, est de se demander dans quels cas on a la chance d'avoir la solution optimale du programme relaxé entière

$$(PL) \quad \max\{c'x :\ Ax \leq b,\ x \in \mathbb{R}^n_+\}.$$

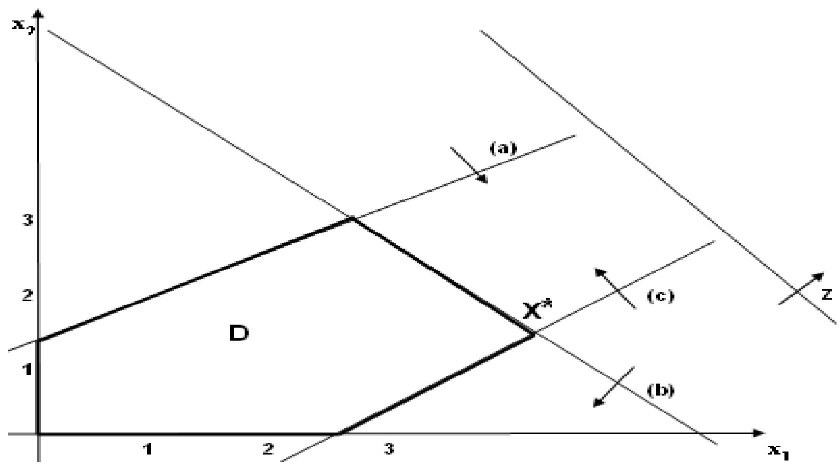


FIG. 2.1 – Domaine d'admissibilité du problème (PL).

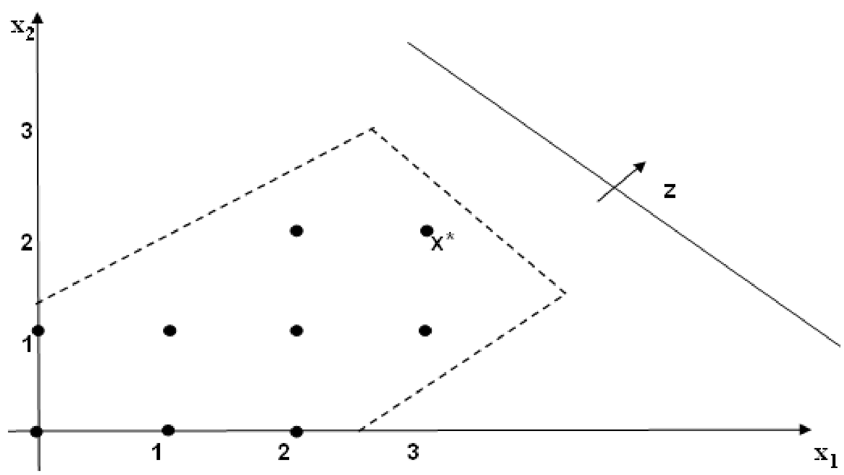


FIG. 2.2 – Domaine d'admissibilité du problème (PLE).

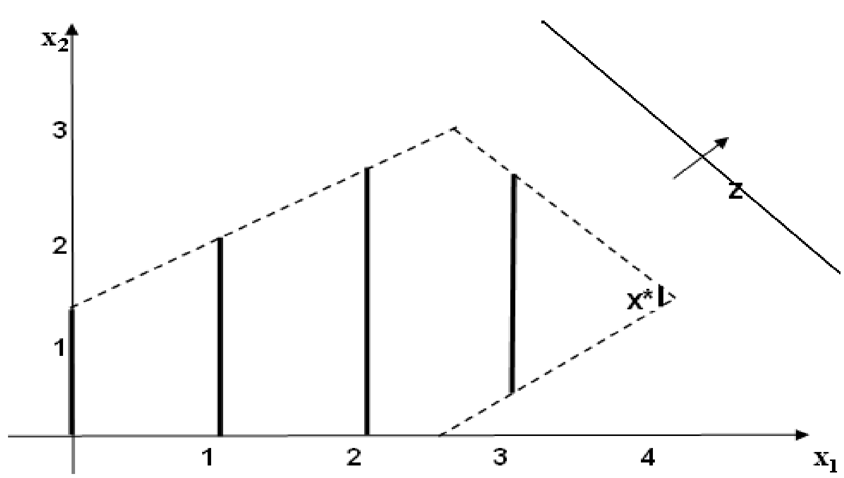


FIG. 2.3 – Domaine d'admissibilité du problème (PLM).

Dans la théorie de la programmation linéaire, on sait qu'une solution réalisable basique prend la forme : $x = (x_B, x_N) = (A_B^{-1}b, 0)$, où $A_B$ est une sous-matrice non singulière d'ordre $m$ de $A$.

**Observation 2.2.1.** [128] *(Condition suffisante). Si la base optimale $A_B$ est telle que $detA_B = \pm 1$, alors la résolution du programme linéaire relaxé résoud $(PLE)$.*

**Preuve.** De la règle de Cramer, $A_B^{-1} = A_B^{*}/detA_B$, où $A_B^{*}$ est la comatrice de $A_B$. Les entrées de $A_B^{*}$ sont tous des sommes de produits de termes de $A_B$. Par conséquent, $A_B^{*}$ est une matrice entière, et comme le $detA_B = \pm 1$, $A_B^{-1}$ est aussi entière. Par conséquent, $A_B^{-1}b$ est entier pour tout vecteur entier $b$. □

**Proposition 2.2.2.** [128] *Le programme linéaire* $\max\{c'x : \ Ax \leq b, \ x \in \mathbb{R}_{+}^{n}\}$ *possède une solution optimale entière pour tout vecteur entier b, pour lequel il possède une valeur optimale finie, si et seulement si A est totalement unimodulaire.*

Le cas le plus favorable est donc quand la matrice $A$ est *totalement unimodulaire* (tous les déterminants de ses sous-matrices sont égaux à 0, 1, ou $-1$), puisqu'alors les polyèdres $X$ et $convS$ coïncident. Certains problèmes se modélisent trivialement en $(PLE)$ avec une matrice des contraintes totalement unimodulaire, tels de nombreux problèmes de flots dans un graphe ou encore des problèmes d'ordonnancement simples. Ces problèmes peuvent donc être résolus en un temps polynomial.

## 2.3 Méthode d'arrondi, Optimalité, Relaxation et Dualité

### 2.3.1 Méthode d'arrondi

Hormis des cas très particuliers, une solution de $(PL)$ comportera généralement des composantes fractionnaires.
La première idée qui vient à l'esprit, lorsqu'on se trouve confronté à un problème en nombres entiers, est d'utiliser une méthode d'arrondi, par exemple en remplaçant, dans la solution optimale continue, chaque composante fractionnaire par l'entier le plus proche.
L'exemple suivant montre clairement l'insuffisance de telles méthodes et permet de mieux saisir la difficulté inhérente aux problèmes de programmation en nombres entiers.

**Exemple 2.3.1.** Considérons le problème (du type : "sac à dos") à deux variables et une seule contrainte :

$$\begin{cases} z = 10x_1 + 11x_2 \longrightarrow \max, \\ 10x_1 + 12x_2 \leq 59, \\ x_1,\ x_2 \geq 0, \text{ entiers.} \end{cases}$$

On peut représenter facilement l'ensemble des solutions continues d'une part, entières d'autre part, dans le plan (figure 2.4).

L'optimum continu est le point de coordonnées $x_1 = 5,9$ et $x_2 = 0$ pour lequel $z = 59$.

Une simple méthode d'arrondi conduirait à la solution $x' = (6,\ 0)$, laquelle ne satisfait pas les contraintes, et à la solution $x''(5,\ 0)$ qui n'est pas optimale parmi les solutions à valeurs entières.

En examinant maintenant les points à coordonnées entières à l'intérieur du polyèdre des solutions continues, on constate que l'optimum entier est le point de coordonnées : $x^* = (1,\ 4)$ pour lequel $z^* = 54$. On voit que ce point est très éloigné de l'optimum continu.

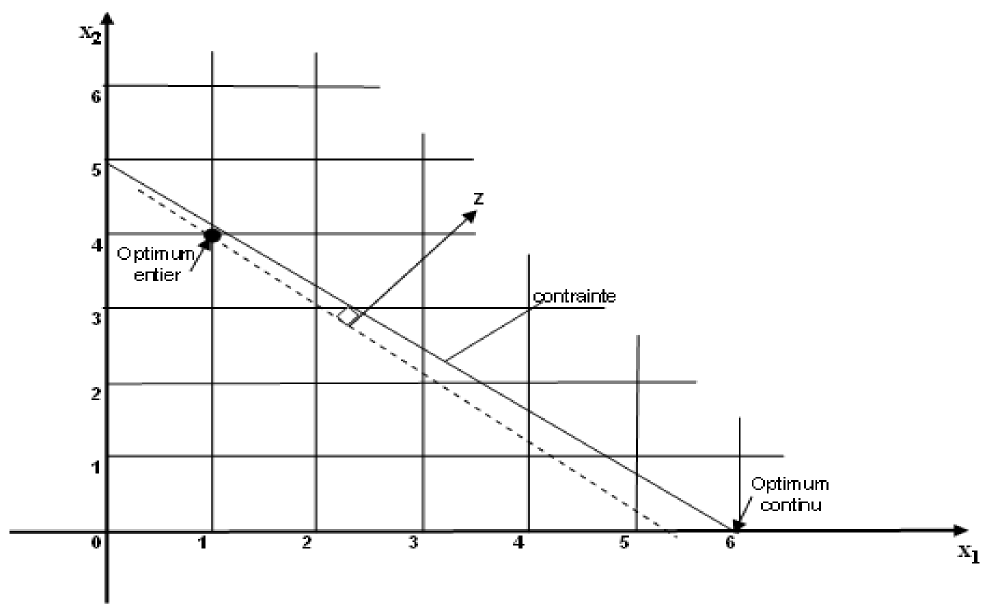


FIG. 2.4 – Exemple illustrant la différence entre la résolution d'un programme linéaire "en continu" et "en nombres entiers".

De fait, il est facile de construire des exemples de même type, pour lesquels l'optimum entier est aussi éloigné que l'on veut de l'optimum continu. Ceci explique pourquoi les méthodes d'arrondi sont en général inefficaces. Cependant, arrondir la solution du PL relaxé est parfois possible quand les variables représentent des quantités importantes.

### 2.3.2 Réduction à un problème en variables bivalentes (0 ou 1)

Considérons le $(PLE)$ et supposons que $X = \{x \in \mathbb{R}^n : \ Ax \leq b, \ \ x \geq 0\}$ est borné. Alors il est toujours possible d'associer à chaque variable $x_j$ des bornes de variation :

$$\alpha_j \leq x_j \leq \beta_j.$$

Pour déterminer $\beta_j$, par exemple, on pourra résoudre le programme linéaire :

$$\begin{cases} x_j \longrightarrow \max, \\ x \in X. \end{cases}$$

Ainsi on peut toujours se ramener au cas où chaque variable $x_j$ ne peut prendre qu'un nombre fini de valeurs.
On peut aller plus loin en remarquant que toute variable $x_j$ ne peut prendre que $\beta_j + 1$ valeurs entières $0,\ 1,\ 2,\ ...,\ \beta_j$ On peut alors lui substituer une combinaison linéaire :

$$x_j = y_0 + 2y_1 + 4y_2 + \ ... + \ 2^p y_p,$$

de $p+1$ variables : $y_0,\ ...,\ y_p$, chacune d'elles étant astreinte à ne prendre que deux valeurs 0 et 1 ($p$ est le plus petit entier tel que $\beta_j \leq 2^{p+1} - 1$).
On peut de la sorte toujours se ramener au cas où $(PLE)$ est un programme linéaire en variables bivalentes. Cependant, la transformation précédente, qui est toujours possible, n'est pas nécessairement toujours avantageuse en pratique.

### 2.3.3 Optimalité et relaxation

Soit le $(PLE)$ :

$$z = \max\{c'x \ : \ x \in S \subseteq \mathbb{Z}^n\},$$

comment peut-on prouver qu'un point donné $x^*$ est optimal ?
Ou posée d'une autre manière, on cherche quelques conditions d'optimalité qui nous permettra d'avoir un critère d'arrêt dans un algorithme de résolution d'un (PLE).
Une réponse naïve est de trouver une borne inférieure $\underline{z} \leq z$ et une borne supérieure $\overline{z} \geq z$ tel que $\underline{z} = \overline{z} = z$. Pratiquement, cela veut dire que l'algorithme trouvera une séquence décroissante : $\overline{z}_1 > \overline{z}_2 > ... > \overline{z}_s \geq z$, de la borne supérieure, et une séquence croissante : $\underline{z}_1 < \underline{z}_2 < ... < \underline{z}_t \leq z$, de la borne inférieure et arrêter le processus lorsque : $\overline{z}_s - \underline{z}_t \leq \epsilon$, où $\epsilon$ est un nombre non négatif choisi à l'avance. Cependant, on doit trouver un moyen pour déterminer de telles bornes inférieures et supérieures.

**Bornes primales (Bornes inférieures)**

Toute solution réalisable $x^* \in S$, nous fournit une borne inférieure $\underline{z} = c'x^* \leq z$. C'est là essentiellement le seul moyen qu'on connaît pour obtenir une borne inférieure. Pour certains $(PLE)$, trouver une solution réalisable est une tâche très difficile.

**Bornes duales (Bornes supérieures)**

L'approche la plus importante est par la relaxation.

**Définition 2.3.1.** Un problème $(LR)$ : $\; z^r = \max\{f(x) :\; x \in T \subseteq \mathbb{R}^n\}$ est une relaxation de $(PLE)$ : $\; z = \max\{c'x :\; x \in S \subseteq \mathbb{Z}^n\}$ si :

**(i)** $S \subseteq T$, et

**(ii)** $f(x) \geq c'x$ pour tout $x \in S$.

**Proposition 2.3.1.** [128] *Si $(LR)$ est une relaxation de $(PLE)$, alors $z^r \geq z$.*

**Preuve.** Si $x^*$ est une solution optimale de $(PLE)$, $x^* \in S \subseteq T$ et $z = c'x^* \leq f(x^*)$. Comme $x^* \in T$, $f(x^*)$ est une borne inférieure pour $z^r$, et donc $z \leq f(x^*) \leq z^r$. $\square$

La forme la plus utilisée est la programmation linéaire relaxée.

**Proposition 2.3.2.** [128]

**(i)** *Si le problème relaxé $(LR)$ est irréalisable, le problème original $(PLE)$ est aussi irréalisable.*

**(ii)** *Soit $x^*$ une solution optimale de $(LR)$. Si $x^* \in S$ et $f(x^*) = c'x^*$, alors $x^*$ est une solution optimale du $(PLE)$.*

**Preuve.**

**(i)** Comme $(LR)$ est irréalisable, $T = \emptyset$, cela implique que $S = \emptyset$.

**(ii)** Comme $x^* \in S$, $z \geq c'x^* = f(x^*) = z^r$. Comme $z \leq z^r$, on obtient $c'x^* = z = z^r$. $\square$

**Relaxation Lagrangienne**

Considérons le programme en nombres entiers

$$z = \max\{c'x :\; x \in S\}, \quad avec \;\; S = \{x \in \mathbb{Z}^n_+ :\; Ax \leq b\},$$

qui peut être écrit ainsi :

$$(PLE) \qquad \begin{cases} z = c'x \longrightarrow \max, \\ A^1x \leq b^1, \text{(contraintes compliquées)} \\ A^2x \leq b^2, \text{(contraintes faciles)} \\ x \in \mathbb{Z}^n_+, \end{cases}$$

où $A = \begin{pmatrix} A^1 \\ A^2 \end{pmatrix}$ et $b = \begin{pmatrix} b^1 \\ b^2 \end{pmatrix}$. On suppose que $A^2x \leq b^2$ sont les $m - m_1$ "contraintes faciles". En retirant les $m_1$ contraintes compliquées $A^1x \leq b^1$, on obtient une relaxation qui est plus facile à résoudre que le problème original. Il existe plusieurs problèmes qui peuvent être ainsi partitionnés.

Le problème précédent peut s'écrire :

$$(PLE_Q) \qquad \begin{cases} z = c'x \longrightarrow \max, \\ A^1x \leq b^1, \\ x \in Q, \end{cases}$$

où $Q = \{x \in \mathbb{Z}^n_+ : \ A^2x \leq b^2\}$.

Maintenant, pour tout $\lambda \in \mathbb{R}^{m_1}_+$ considérons le problème :

$$(LR_\lambda) \qquad z^{\,r}(\lambda) = \max\{z(\lambda, \ x) : \ x \in Q\},$$

où $z(\lambda, \ x) = c'x + \lambda(b^1 - A^1x)$.

Le problème $(LR_\lambda)$ est dit relaxation Lagrangienne de $(PLE_Q)$ liée aux contraintes $A^1x \leq b^1$.

$(LR_\lambda)$ ne contient pas les contraintes compliquées. Cependant, elles sont incluses dans la fonction objectif avec le terme "pénalité" : $\lambda(b^1 - A^1x)$. Comme $\lambda \geq 0$, violer les contraintes $A^1x \leq b^1$ met le terme pénalité négatif, et par conséquent $A^1x \leq b^1$ est nécessairement satisfaite si $\lambda$ est assez grand.

**Proposition 2.3.3.** [128] $(LR_\lambda)$ *est une relaxation de* $(PLE_Q)$ *pour tout* $\lambda \geq 0$.

**Preuve.** Si $x$ est réalisable pour $(PLE_Q)$, alors $x \in Q$ et par conséquent $x$ est réalisable pour $(LR_\lambda)$. Aussi, $z(\lambda, \ x) = c'x + \lambda(b^1 - A^1x) \geq c'x$ pour tout $x$ réalisable dans $(PLE_Q)$, puisque $A^1x \leq b^1$ et $\lambda \geq 0$. $\square$

Comme conséquence de la proposition 2.3.3, on a $z^r(\lambda) \geq z$ pour tout $\lambda \geq 0$. La borne minimale déduite de la famille infinie de la relaxation $(LR_\lambda)_{\lambda \geq 0}$ est $z^r(\lambda^*)$, où $\lambda^*$ est une solution optimale du problème :

$$(LD) \qquad z^{LD} = \min_{\lambda \geq 0} z^r(\lambda).$$

Le problème $(LD)$ est dit le dual Lagrangien de $(PLE)$ lié aux contraintes $A^1x \leq b^1$.

### 2.3.4 Dualité

La propriété importante de la dualité est que la valeur d'une solution quelconque réalisable du dual donne une borne supérieure sur la valeur optimale de la fonction objectif $z$ du primal. Ceci suggère la définition suivante :

**Définition 2.3.2.** Les deux problèmes

$$(PLE) \qquad z = \max\{c(x) : \ x \in S\},$$

$$(D) \qquad w = \min\{w(u) : \ u \in U\},$$

forment une paire duale faible si $c(x) \leq w(u)$ pour tout $x \in S$ et tout $u \in U$. Si $z = w$, ils forment alors une paire duale forte.
La question qui se pose alors est : est-ce que de telles formulations duales existent ?

**Proposition 2.3.4.** [128] *Le problème en nombres entiers* $z = \max\{c'x : \ Ax \leq b, \ x \in \mathbb{Z}^n_+\}$ *et le programme linéaire* $w^{LP} = \min\{u'b : \ u'A \geq c, \ u \in \mathbb{R}^m_+\}$ *forme une paire duale faible.*

**Proposition 2.3.5.** [128] *Supposons que* $(PLE)$ *et* $(D)$ *forment une paire duale faible. Alors,*

**(i)** *Si* $D$ *est non borné,* $(PLE)$ *est irréalisable.*

**(ii)** *Si* $x^* \in S$ *et* $u^* \in U$ *satisfont* $c(x^*) = w(u^*)$*, alors* $x^*$ *est optimale pour* $(PLE)$ *et* $u^*$ *est optimale pour* $D$.

**Proposition 2.3.6.** [105]*Si un problème est dual pour une relaxation du* $(PLE)$*, alors il est aussi dual pour* $(PLE)$.

**Preuve.** Supposons que $w = \min\{w(u) : \ u \in U\}$ est dual pour $(LR)$. Alors $z^r(x) \leq w(u)$ pour tout $x \in X$ et tout $u \in U$. Par relaxation, $c(x) \leq z^r(x)$ pour tout $x \in S \subseteq X$. Par conséquent, $c(x) \leq w(u)$ pour tout $x \in S$ et $u \in U$. $\square$

### 2.3.5 Complexité

Le problème général de la programmation linéaire en variables entières $(PLE)$ ou $(PLM)$ est d'une difficulté beaucoup plus grande, réellement d'une autre nature, que le problème de programmation linéaire en variables continues $(PL)$.

Même si le problème ($PLE$) ne contient qu'un nombre fini de solutions (si le polyèdre $S$ est borné), alors que le problème ($PL$) contient une infinité non dénombrable de solutions admissibles, le premier problème est beaucoup plus difficile que le second. En effet, l'étude initiale du problème (PL) nous a montré immédiatement :

- que l'obtention de la solution optimale ne nécessite de s'intéresser qu'à un nombre fini de solutions, à savoir les seuls sommets du polyèdre $X$ ;
- que ces sommets sont facilement caractérisables (ce sont des solutions de base admissibles) et qu'ils peuvent donc être aisément déterminés.

Par contre, dans les situations (PLE) ou (PLM), la solution optimale n'est, en général, pas un sommet de $S$ et peut donc être un point quelconque de $S$ (point intérieur, point frontière ou point extrême). On perd ainsi toute caractérisation particulière de la solution optimale qui est, pour cette raison, bien plus difficile à déterminer.

Cette approche intuitive est évidemment confirmée par l'approche rigoureuse de la complexité des algorithmes. Le problème de décision

$$\text{Existe-t-il une solution } x \in S = X \cap \mathbb{Z}_+^n ?$$

est un problème *NP-complet* (Garey 1979 ) de sorte que le problème ($PLE$) ou ($PLM$) est *NP-Difficile.*

Dans la suite de ce chapitre, nous commencerons par étudier les deux principales familles de méthodes actuellement connues pour résoudre les programmes linéaires en nombres entiers et mixtes : Les méthodes de coupes (ou de troncatures) dans la première section, et dans la deuxième section les méthodes de recherche arborescente (ou d'énumération partielle). La plupart des algorithmes connus dérive de l'une ou l'autre de ces deux familles.

## 2.4 Méthodes des plans sécants

L'idée de base sur laquelle se fondent ces méthodes est la suivante : on commence par résoudre le programme linéaire continu ($PL$) ; si la solution optimale obtenue est un point extrême à coordonnées entières, celle-ci est une solution optimale de ($PLE$). Dans le cas contraire, il est facile de voir que l'on peut toujours tronquer le domaine des

solutions (en rajoutant une contrainte supplémentaire au problème) de façon à éliminer ce point extrême sans exclure aucune solution réalisable entière. Une telle contrainte est appelée une *coupe* (on dit encore : une troncature).
Reprenons l'exemple précédent (exemple 2.3.1) : l'optimum continu était le point extrême (5.9, 0). La contrainte supplémentaire $x_1 \leq 5$ élimine ce point sans exclure aucune solution entière. De même, toute contrainte supplémentaire du type $x_1 + x_2 \leq \alpha$, avec $5 \leq \alpha < 5.9$ est une coupe.
Après avoir rajouté une coupe (ou éventuellement plusieurs), le programme linéaire augmenté des contraintes correspondantes est de nouveau résolu en continu. Si la solution optimale de ce nouveau problème est entière, on obtient alors une solution optimale du $(PLE)$. Sinon, on cherchera une nouvelle coupe que l'on rajoutera à l'ensemble des contraintes ; puis le programme linéaire ainsi augmenté sera optimisé de nouveau, etc.

Si les coupes sont correctement choisies à chaque étape, le polyèdre admissible initial $X$ sera ainsi progressivement réduit jusqu'à coïncider avec l'enveloppe convexe des solutions entières. La solution continue du problème augmenté deviendra alors entière et le problème sera résolu.
Il est clair, cependant, que le choix des coupes est déterminant pour la convergence de la méthode. Si, dans l'exemple précédent, on choisit à l'étape $k$ de rajouter une coupe du type :

$$x_1 + x_2 \leq \alpha_k, \qquad avec \quad \alpha_1 = 5.8 \; et \; \alpha_k = \alpha_{k-1} - \frac{1}{10^k},$$

il est sûr dans ces conditions que l'algorithme ne convergera pas vers une solution entière.
Si, par contre, on choisit comme coupes les deux contraintes :

$$x_1 + x_2 \leq 5 \quad et \quad x_2 \leq 4,$$

qui définissent (avec les contraintes de positivité $x_1 \geq 0,\ x_2 \geq 0$) l'enveloppe convexe des solutions entières du problème, on obtient directement l'optimum entier : $x_1 = 1$, $x_2 = 4$ comme solution continue du programme linéaire augmenté des deux contraintes.

Malheureusement, et c'est là la difficulté essentielle, on ne connaît pas de méthode systématique pour engendrer toutes les équations ou inéquations définissant l'enveloppe convexe des points entiers contenus dans un polyèdre convexe donné. Elles peuvent

d'ailleurs être énormément nombreuses comme l'indique clairement un résultat dû à Jeroslow (1969, 1971) et à Rubin (1970) : même pour un programme en nombres entiers à deux variables et une contrainte, on peut toujours choisir les coefficients de telle sorte que le nombre de faces de l'enveloppe convexe des points entiers soit aussi grand que l'on veut. Par conséquent, engendrer toutes les faces de l'enveloppe convexe des points entiers serait coûteux et superflu : pour la plupart, ces contraintes ne seraient pas actives et ne contribueraient en rien à définir l'optimum entier.
C'est pourquoi, un des résultats les plus importants en programmation en nombres entiers a été la mise en évidence par Gomory (1958) de coupes d'un type particulier permettant d'obtenir la convergence finie de la méthode.

Considérons le problème général de programmation linéaire en nombres entiers :

$$(PLE) \qquad \max\{c'x : \ x \in S\}, \ \textit{où} \ S = \{x \in \mathbb{Z}^n_+ : \ Ax \leq b\},$$

où $A$ et $b$ des données réelles.

**Proposition 2.4.1.** [128] *$convS = \{x \in \mathbb{R}^n_+ : \ \tilde{A}x \leq \tilde{b}\}$ est un polyèdre.*

Ce résultat nous montre qu'en théorie, on peut reformuler le problème $(PLE)$ comme un programme linéaire :

$$(PL) \qquad \max\{c'x : \ \tilde{A}x \leq \tilde{b}, \ x \geq 0\}.$$

Ainsi, pour toute valeur de $c$, une solution optimale de $(PL)$ est une solution optimale de $(PLE)$. Le même résultat reste valable pour un problème linéaire mixte, avec $S = \{(x, \ y) \in \mathbb{Z}^n_+ \times \mathbb{R}^p_+ : \quad Ax + Dy \leq b\}$, où $A, \ D \ et \ b$ sont des données rationnelles.
Pour des problèmes NP-difficiles, on a peu de chance de trouver une description exacte de $convS$. Le but de cette section est de pouvoir approximer $convS$.

On rappel que $[.]$ désigne la partie entière d'un nombre et $\{.\}$ sa partie fractionnaire.

### 2.4.1 Inégalités valides pour un programme linéaire

**Proposition 2.4.2.** [128] *$\pi' x \leq \pi_0$ est une inégalité valide pour $X = \{x \in \mathbb{R}^n_+ : \ Ax \leq b\} \neq \emptyset$ si et seulement si*
*Il existe $u \geq 0, \ v \geq 0$ tels que $u'A - v' = \pi'$ et $u'b \leq \pi_0$, ou alternativement il existe $u \geq 0$ tel que $u'A \geq \pi'$ et $u'b \leq \pi_0$.*

**Preuve.** Par la dualité, $\max\{\pi' x : x \in X\} \leq \pi_0$ si et seulement si $\min\{u'b : u'A - v' = \pi', u \geq 0, v \geq 0\} \leq \pi_0$. $\square$

**Définition 2.4.1.** ***Dominance***

Les inégalités valides $\pi' x \leq \pi_0$ et $u'x \leq u_0$ sont dite équivalente si $(\pi, \pi_0) = \lambda(u, u_0)$ pour un certain $\lambda > 0$.

Si $\pi' x \leq \pi_0$ et $u'x \leq u_0$ sont deux inégalités valides pour $X \subseteq \mathbb{R}^n_+$, $\pi' x \leq \pi_0$ domine $u'x \leq u_0$ s'il existe $\mu > 0$ tel que $\pi' \geq \mu u'$ et $\pi_0 \leq \mu u_0$, et $(\pi, \pi_0) \neq (\mu u, \mu u_0)$.

Observons que si $\pi' x \leq \pi_0$ domine $u'x \leq u_0$, alors $\{x \in \mathbb{R}^n_+ : \pi' x \leq \pi_0\} \subset \{x \in \mathbb{R}^n_+ : u'x \leq u_0\}$.

## 2.4.2 Inégalités valides pour un programme linéaire en nombres entiers

On considère le domaine réalisable d'un programme en nombres entiers :

$$S = \{x \in \mathbb{Z}^n_+ : Ax \leq b\}.$$

**Proposition 2.4.3.** [128] *Soit* $S^1 = \{x \in \mathbb{Z} : x \leq b\}$, *alors l'inégalité* $x \leq [b]$ *est valide pour* $S^1$.

### La procédure de Chvátal-Gomory (C-G) pour la construction d'inégalités valides

Soit l'ensemble $S = X \cap \mathbb{Z}^n$, où $X = \{x \in \mathbb{R}^n_+ : Ax \leq b\}$, $A$ est une matrice d'ordre $m \times n$ et $u \in \mathbb{R}^m_+$ :

**(i)** L'inégalité

$$\sum_{j=1}^{n} ua_j x_j \leq ub$$

est valide pour $X$ car $u \geq 0$ et $\sum_{j=1}^{n} a_j x_j \leq b$.

**(ii)** L'inégalité

$$\sum_{j=1}^{n} [ua_j] x_j \leq ub$$

est valide pour $X$ puisque $x \geq 0$.

**(iii)** L'inégalité

$$\sum_{j=1}^{n} [ua_j] x_j \leq [ub]$$

est valide pour $S$ car $x$ est entier, ainsi que $\sum_{j=1}^{n} [ua_j] x_j$.

L'effet surprenant est que cette simple procédure est suffisante pour générer toutes les inégalités valides pour un programme en nombres entiers.

**Théorème 2.4.4.** [128] *Toute inégalité valide pour $S$ peut être obtenue en appliquant la procédure de Chvátal-Gomory un nombre fini de fois.*

**Preuve.** *voir* [128, 105].

**Exemple 2.4.1.** Soit $S = X \cap \mathbb{Z}^n$ l'ensemble des points entiers dans $X$, où $X$ est donné par :

$$\begin{cases} 7x_1 - 2x_2 \leq 14, \\ \qquad\quad x_2 \leq 3, \\ 2x_1 - 2x_2 \leq 3, \\ x \geq 0. \end{cases}$$

**(i)** Premièrement, en combinant les contraintes avec des poids non négatifs $u = (\frac{2}{7},\ \frac{37}{63},\ 0)$, on obtient l'inégalité valide pour $X$

$$2x_1 + \frac{1}{63}x_2 \leq \frac{121}{21}.$$

**(ii)** En réduisant les coefficients du côté gauche de l'inégalité à l'entier le plus proche donne l'inégalité valide pour $X$

$$2x_1 + 0x_2 \leq \frac{121}{21}.$$

**(iii)** Maintenant, comme le côté gauche de l'inégalité est entier pour tout point de $S$, on peut réduire le côté droit de l'inégalité à l'entier le plus proche, et on obtient ainsi l'inégalité valide pour $S$ :

$$2x_1 \leq [\frac{121}{21}] = 5.$$

Observons que si on répète la procédure sur cette dernière inégalité avec un poids $\frac{1}{2}$, on obtient l'inégalité plus serrée $x_1 \leq [\frac{5}{2}] = 2$.

**Arithmétique modulaire** (*Modular Arithmetic*)

Ici on dérive une inégalité valide pour l'ensemble des solutions d'une équation linéaire en nombres entiers : $S = \{x \in \mathbb{Z}^n_+ :\ \sum_{j=1}^n \alpha_j x_j = \alpha_0\}$, où $\alpha_j \in \mathbb{R}$ pour tout $j = \overline{0,\ n}$. Soit $d$ un entier positif et

$$S_d = \{x \in \mathbb{Z}^n_+ :\ \sum_{j=1}^n \alpha_j x_j = \alpha_0 + kd,\ k \text{ entier}\}.$$

On va dériver une inégalité valide pour l'ensemble $S_d$, comme $S \subseteq S_d$, l'inégalité est aussi valide pour $S$.
Soit $\alpha_j = \beta_j + k_j d$ pour $j = \overline{0,\ n}$, où $0 \leq \beta_j < d$ et $k_j$ est un entier, ainsi $\beta_j$ est le reste de la division de $\alpha_j$ par $d$. Alors

$$S_d = \{x \in \mathbb{Z}_+^n : \sum_{j=1}^{n} \beta_j x_j = \beta_0 + kd,\ k \ entier\}.$$

Sachant que $\sum_{j=1}^{n} \beta_j x_j \geq 0$ et $\beta_0 < d$ implique $k \geq 0$ ; par conséquent on obtient l'inégalité valide

$$\sum_{j=1}^{n} \beta_j x_j \geq \beta_0. \tag{2.1}$$

Cette inégalité est non triviale uniquement si $d$ ne divise pas $\alpha_0$, c'est-à-dire si $\beta_0 > 0$.

Pour l'ensemble donné par

$$37x_1 - 68x_2 + 78x_3 + x_4 = 141, \qquad x \in \mathbb{Z}_+^4,$$

l'inégalité (2.1) avec $d = 12$ donne $x_1 + 4x_2 + 6x_3 + x_4 \geq 9$.

Une importante inégalité de ce type est dérivée quand $d = 1$ et $\alpha_0$ est non entier. Alors (2.1) produit l'inégalité valide

$$\sum_{j=1}^{n} (\alpha_j - [\alpha_j])x_j \geq \alpha_0 - [\alpha_0] \Longleftrightarrow \sum_{j=1}^{n} \{\alpha_j\} x_j \geq \{\alpha_0\}, \tag{2.2}$$

qui est dite coupe de Gomory.
Par exemple, supposons $3\frac{3}{4} - \frac{1}{2}x_1 + \frac{7}{4}x_2 - \frac{11}{4}x_3 = 0$ est une équation obtenue dans la résolution d'un programme linéaire et on requiert aussi que les variables soient entières non négatives. Alors (2.2) produit l'inégalité valide : $\frac{1}{2}x_1 + \frac{1}{4}x_2 + \frac{3}{4}x_3 \geq \frac{3}{4}$.

**Inégalités disjonctives** (*Disjunctive inequality*)

**Proposition 2.4.5.** [128] *Si* $\sum_{j=1}^{n} \pi_j^1 x_j \leq \pi_0^1$ *est valide pour* $S_1 \subset \mathbb{R}_+^n$ *et* $\sum_{j=1}^{n} \pi_j^2 x_j \leq \pi_0^2$ *est valide pour* $S_2 \subset \mathbb{R}_+^n$*, alors*

$$\sum_{j=1}^{n} \min(\pi_j^1,\ \pi_j^2) x_j \leq \max(\pi_0^1,\ \pi_0^2) \tag{2.3}$$

*est valide pour* $S_1 \cup S_2$.

**Preuve.** Si $x \in S_1 \cup S_2$, donc $x \in S_1$ ou $x \in S_2$. Comme $x \geq 0$, alors

$$\sum_{j=1}^{n} \min(\pi_j^1,\ \pi_j^2)x_j \leq \sum_{j=1}^{n} \pi_j^i x_j \leq \pi_0^i \leq \max(\pi_0^1,\ \pi_0^2), \quad pour\ i = \overline{1, 2}.$$

Ainsi, l'inégalité est valide pour tout $x \in S_1 \cup S_2$. $\square$

En d'autres termes, si on doit satisfaire un des ensembles de contraintes, mais pas nécessairement les deux, et qu'on connaît des inégalités valides pour chaque ensemble, alors (2.3) est une inégalité valide pour la disjonction des deux ensembles.

Cela donne un autre moyen, dit procédure disjonctive, pour la génération d'inégalités valides pour le domaine $S = \{x \in \mathbb{Z}_+^n :\ Ax \leq b\}$. Les deux étapes de la procédure sont :

**i.** $\sum_{j=1}^{n}(ua_j)x_j \leq ub$ pour tout $u \geq 0$.

**ii.** Etant donné $\delta \in \mathbb{Z}_+$, si

- **(a)** $\sum_{j=1}^{n} \pi_j x_j - \alpha(x_k - \delta) \leq \pi_0$

  est valide pour $S$ pour un certain $\alpha \geq 0$ et
- **(b)** $\sum_{j=1}^{n} \pi_j x_j + \beta(x_k - \delta - 1) \leq \pi_0$

  est valide pour $S$ pour un certain $\beta \geq 0$, alors
- **(c)** $\sum_{j=1}^{n} \pi_j x_j \leq \pi_0$

  est valide pour $S$.

Notons que $(a)$ montre que $(c)$ est valide pour $S_1 = S \cap \{x \in \mathbb{Z}_+^n :\ x_k \leq \delta\}$ et $(b)$ montre que $(c)$ est valide pour $S_2 = S \cap \{x \in \mathbb{Z}_+^n :\ x_k \geq \delta + 1\}$. Comme $S = S_1 \cup S_2$ la proposition 2.4.5 établit que $(c)$ est valide pour $S$.

Les inégalités générées par l'application répétée de la procédure disjonctive sont dites D-inégalités.

**Exemple 2.4.2.** Un exemple de D-inégalités est représenté dans la figure 2.5, où

$$X = \{x \in \mathbb{R}_+^2 :\ -x_1 + x_2 \leq \frac{1}{2},\ \frac{1}{2}x_1 + x_2 \leq \frac{5}{4},\ x_1 \leq 2\}.$$

Les deux premières inégalités peuvent être réécrites comme

$$-\frac{1}{4}x_1 + x_2 - \frac{3}{4}x_1 \leq \frac{1}{2}$$

et

$$-\frac{1}{4}x_1 + x_2 - \frac{3}{4}(1 - x_1) \leq \frac{1}{2}.$$

L'utilisation de la disjonction $x_1 \leq 0$ ou $x_1 \geq 1$ donne l'inégalité valide $-\frac{1}{4}x_1 + x_2 \leq \frac{1}{2}$ pour $S = X \cap \mathbb{Z}^2$.

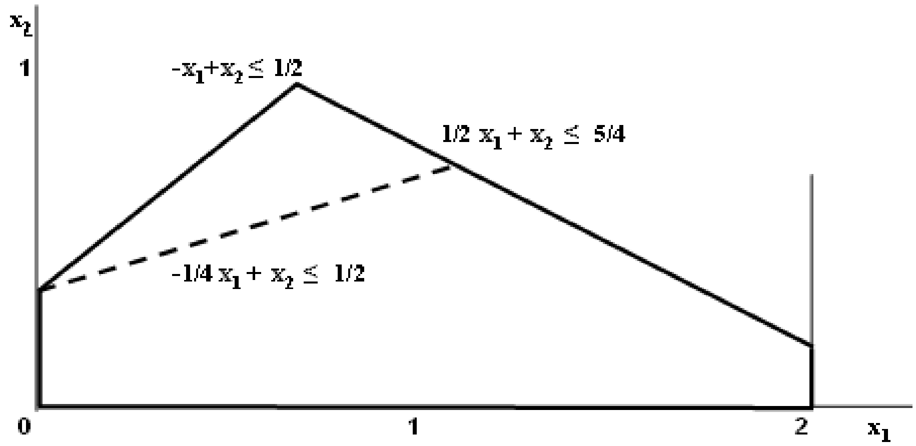


FIG. 2.5 –

### 2.4.3 Algorithme des plans sécants

On suppose que $S = X \cap \mathbb{Z}^n$ et que l'on connaît une famille $\mathcal{F}$ d'inégalités valides $\pi' x \leq \pi_0$, $(\pi,\ \pi_0) \in \mathcal{F}$ pour $S$.

**Définition 2.4.2.** ***Problème de séparation***
Le problème de séparation associé à un problème d'optimisation combinatoire est le problème suivant : Etant donné $x^* \in \mathbb{R}^n$, tester si $x^* \in convS$ ? Sinon, trouver une inégalité $\pi' x \leq \pi_0$ satisfaite par tous les points dans $S$, mais qui est violée par le point $x^*$.

On décrit maintenant un algorithme de base de plans sécants pour $(PLE)$ : $\max\{c'x : \ x \in S\}$, qui génére des inégalités valides à partir de $\mathcal{F}$.

**Algorithme**
*Initialisation.* Poser $t = 0$ et $X^0 = X$.
*Itération t :* Résoudre le programme linéaire

$$\overline{z}^t = \max\{c'x : \ x \in X^t\}.$$

Soit $x^t$ la solution optimale.
Si $x^t \in \mathbb{Z}^n$, on arrête : $x^t$ est une solution optimale pour $(PLE)$.
Si $x^t \notin \mathbb{Z}^n$, résoudre le problème de séparation pour $x^t$ et la famille $\mathcal{F}$.
Si une inégalité $(\pi^t, \pi_0^t) \in \mathcal{F}$ est trouvée avec $(\pi^t)' x^t > \pi_0^t$ de sorte qu'elle coupe $x^t$, poser $X^{t+1} = X^t \cap \{x : \ (\pi^t)' x \leq \pi_0^t\}$, et incrémenter $t$. Sinon arrêter.

Si l'algorithme se termine sans trouver une solution entière pour $(PLE)$,

$$X^t = X \cap \{x : \ (\pi^i)' x \leq \pi_0^i, \ i = \overline{1, t}\}$$

est une formulation améliorée qui peut être introduite dans un algorithme de branch and bound.

Le problème de séparation des inégalités de Chv$\acute{a}$tal-Gomory a été récemment démontré comme étant NP-complet par Eisenbrand [45]. Cependant, Gomory [62, 64] a donné une méthode pour déterminer une coupe séparant un point extrême du polyèdre de la relaxation continue de $S$ qui n'est pas un point entier.

### Algorithme des coupes fractionnaires de Gomory

Ici on considère le programme en nombres entiers :

$$\max\{c'x : \ Ax = b, \ x \geq 0 \ et \ entier\}.$$

L'idée est de résoudre en premier lieu le programme linéaire relaxé associé et trouver une solution optimale basique, choisir une variable qui n'est pas entière, et générer alors l'inégalité de Chv$\acute{a}$tal-Gomory sur la contrainte associée à cette variable de base de telle sorte à couper la solution du programme linéaire.

Etant donnée une solution optimale de base, le problème s'écrit alors sous la forme :

$$\begin{cases} \overline{a}_{00} + \sum\limits_{j \in J_N} \overline{a}_{0j} x_j \longrightarrow \max, \\ \\ x_i + \sum\limits_{j \in J_N} \overline{a}_{ij} x_j = \overline{a}_{i0} \ \ pour \ i = \overline{1, \ m}, \\ \\ x \geq 0 \ et \ entier, \end{cases}$$

où $\overline{a}_{0j} \leq 0$ pour $j \in J_N$ et $\overline{a}_{i0} \geq 0$ pour $i = \overline{1, \ m}$, où $J_N$ est l'ensemble des indices des variables non basiques.

Si la solution de base optimale est non entière, il existe une certaine ligne $i$ avec $\overline{a}_{i0} \notin \mathbb{Z}$.

En choisissant une telle ligne, la coupe de Chv$\acute{a}$tal-Gomory associée est alors :

$$x_i + \sum_{j \in J_N} [\overline{a}_{ij}] x_j \leq [\overline{a}_{i0}]. \tag{2.4}$$

Après élimination de $x_i$ l'inégalité devient

$$\sum_{j\in J_N}(\overline{a}_{ij}-[\overline{a}_{ij}])x_j \geq \overline{a}_{i0}-[\overline{a}_{i0}] \Longleftrightarrow \sum_{j\in J_N}\{\overline{a}_{ij}\}x_j \geq \{\overline{a}_{i0}\}. \qquad (2.5)$$

Par définition et par le choix de la ligne $i$, $0 \leq \{\overline{a}_{ij}\} < 1$ et $0 < \{\overline{a}_{i0}\} < 1$. Comme $x_j^* = 0$ pour toute variable non basique $j \in J_N$ dans la solution optimale du programme linéaire, cette inégalité coupe $x^*$. Il est aussi important d'observer que la différence entre le terme du coté gauche et celui du coté droit de l'inégalité de Chv$\acute{a}$tal-Gomory (2.4) est entière quand $x$ est entier, donc quand on réécrit (2.5) comme une égalité :

$$s = -\{\overline{a}_{i0}\} + \sum_{j\in J_N}\{\overline{a}_{ij}\}x_j,$$

la variable d'écart $s$ est une variable entière non négative.

L'algorithme pour la programmation entière de Gomory [65] permet de trouver la solution optimale d'un programme linéaire en nombres entiers en un nombre fini d'application de la méthode de séparation décrite ci-dessus. En contrepartie, la convergence à l'optimum est extraordinairement lente, due au fait que ces coupes sont "faibles" au sens que fréquemment elles ne définissent pas des hyperplans d'appui de l'enveloppe convexe des points réalisables.

**Exemple 2.4.3.** Considérons le programme en nombres entiers

$$\begin{cases} z = \ 4x_1 - x_2 \longrightarrow \max, \\ 7x_1 - 2x_2 \leq 14, \\ \qquad\quad x_2 \leq 3, \\ 2x_1 - 2x_2 \leq 3, \\ x_1,\ x_2 \geq 0 \ \textit{et entiers}. \end{cases}$$

Ajoutons les variables d'écarts $x_3,\ x_4,\ x_5$, et remarquons que les données sont entières. Les variables d'écart doivent donc prendre des valeurs entières. La résolution comme un programme linéaire donne

| | | **c** | 4 | $-1$ | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| $c_B$ | $x_B$ | $b$ | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ |
| 4 | $x_1$ | $\frac{20}{7}$ | 1 | 0 | $\frac{1}{7}$ | $\frac{2}{7}$ | 0 |
| $-1$ | $x_2$ | 3 | 0 | 1 | 0 | 1 | 0 |
| 0 | $x_5$ | $\frac{23}{7}$ | 0 | 0 | $\frac{-2}{7}$ | $\frac{10}{7}$ | 1 |
| $z = \frac{59}{7}$ | | $E$ | 0 | 0 | $\frac{-4}{7}$ | $\frac{-1}{7}$ | 0 |

La solution du programme linéaire est $x = (\frac{20}{7},\ 3,\ 0,\ 0,\ \frac{23}{7}) \notin \mathbb{Z}^5_+$. Ainsi, on utilise la première ligne dans laquelle la variable $x_1$ est fractionnaire, pour générer la coupe :

$$\frac{1}{7}x_3 + \frac{2}{7}x_4 \geq \frac{6}{7}.$$

ou encore

$$s = -\frac{6}{7} + \frac{1}{7}x_3 + \frac{2}{7}x_4,$$

avec $s,\ x_3,\ x_4 \geq 0$ et entiers.
En additionnant cette coupe, et en réoptimisant de nouveau, on obtient le nouveau tableau optimal suivant :

| | | c | 4 | $-1$ | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|
| $c_B$ | $x_B$ | $b$ | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $s$ |
| 4 | $x_1$ | 2 | 1 | 0 | 0 | 0 | 0 | 1 |
| $-1$ | $x_2$ | $\frac{1}{2}$ | 0 | 1 | 0 | 0 | $\frac{-1}{2}$ | 1 |
| 0 | $x_3$ | 1 | 0 | 0 | 1 | 0 | $-1$ | $-5$ |
| 0 | $x_4$ | $\frac{5}{2}$ | 0 | 0 | 0 | 1 | $\frac{1}{2}$ | 6 |
| $z = \frac{15}{2}$ | | $E$ | 0 | 0 | 0 | 0 | $\frac{-1}{2}$ | $-3$ |

Maintenant la nouvelle solution optimale du programme linéaire est $x = (2,\ \frac{1}{2},\ 1,\ \frac{5}{2},\ 0)$ qui est toujours non entière, car la variable $x_2$ est fractionnaire. La coupe de Gomory sur la ligne 2, dans laquelle $x_2$ est basique, est : $\frac{1}{2}x_5 \geq \frac{1}{2}$ ou $-\frac{1}{2}x_5 + t = -\frac{1}{2}$ avec $t \geq 0$ et entier. En additionnant cette contrainte et en réoptimisant, on obtient

| | | c | 4 | $-1$ | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| $c_B$ | $x_B$ | $b$ | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $s$ | $t$ |
| 4 | $x_1$ | 2 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| $-1$ | $x_2$ | 1 | 0 | 1 | 0 | 0 | 0 | 1 | $-1$ |
| 0 | $x_3$ | 2 | 0 | 0 | 1 | 0 | 0 | $-5$ | $-2$ |
| 0 | $x_4$ | 2 | 0 | 0 | 0 | 1 | 0 | 6 | 1 |
| 0 | $x_5$ | 1 | 0 | 0 | 0 | 0 | 1 | 0 | $-1$ |
| $z = 7$ | | $E$ | 0 | 0 | 0 | 0 | 0 | $-3$ | $-1$ |

Maintenant la solution du programme linéaire optimale est entière, et ainsi le vecteur $(x_1, x_2) = (2,\ 1)$ résout le programme original en nombres entiers.

Considérons maintenant la première coupe, en substituant $x_3$ et $x_4$, on obtient :

$$\frac{1}{7}(14 - 7x_1 + 2x_2) + \frac{2}{7}(3 - x_2) \geq \frac{6}{7}, \quad ou\ x_1 \leq 2.$$

Dans la figure 2.6, on peut vérifier que cette inégalité est valide et coupe la solution fractionnaire $(\frac{20}{7},\ 3)$. D'une manière similaire, en substituant $x_5$ dans la seconde coupe on obtient l'inégalité valide $x_1 - x_2 \leq 1$, écrite en termes des variables originales.

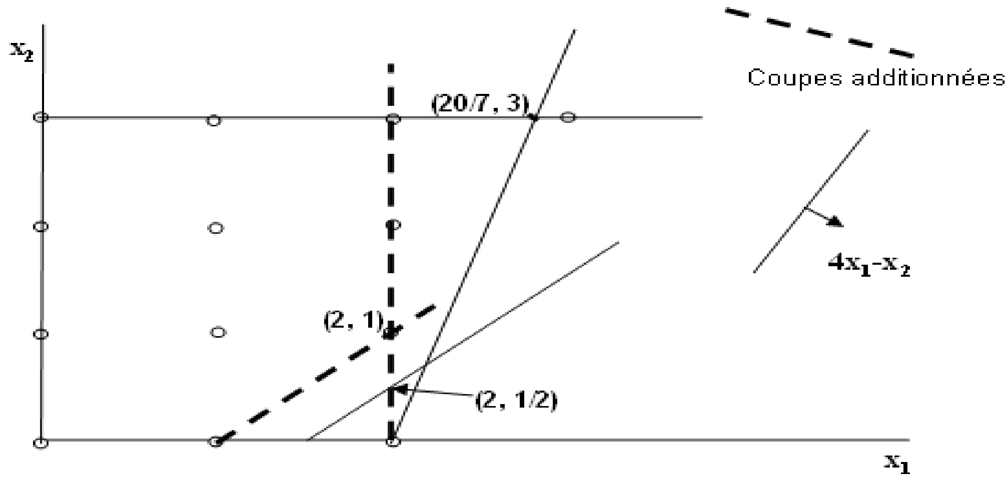


FIG. 2.6 – Les plans sécants de Gomory

Malgré le grand nombre de résultats théoriques qui ont pu être dérivés grâce aux coupes de Chv$\acute{a}$tal-Gomory, celles-ci sont peu utilisées pour la résolution algorithmique de programmes en nombres entiers. Une des causes en est que ces coupes ne sont pas applicables aux programmes en nombres mixtes. Les coupes mixtes de Gomory que nous allons maintenant étudier constituent une famille de coupes plus fortes et qui permettent de résoudre ce problème.

### 2.4.4 Inégalités valides pour un programme linéaire en nombres entiers et mixtes

**L'inégalité mixte basique** (*The Basic Mixed Integer Inequality*)

On a vu que quand $x \leq b,\ x \in \mathbb{Z}$, l'inégalité arrondie $x \leq [b]$ suffit pour générer toutes les inégalités pour un problème totalement en nombres entiers. Ici on examine le cas d'un programme en nombres entiers et mixtes.

**Proposition 2.4.6.** [128] *Soit* $X^{\geq} = \{(x,\ y) \in \mathbb{Z} \times \mathbb{R}_+ : x + y \geq b\}$. *L'inégalité*

$$y \geq \{b\}([b] + 1 - x) \quad ou \quad \frac{y}{\{b\}} + x \geq [b] + 1$$

*est valide pour* $X^{\geq}$.

**Preuve.** Si $x \geq [b] + 1$, alors $y \geq 0 \geq \{b\}([b] + 1 - x)$. Si $x < [b] + 1$, alors

$$\begin{aligned} y &\geq b - x = \{b\} + ([b] - x) \\ &\geq \{b\} + \{b\}([b] - x), \textit{ comme } [b] - x \geq 0 \textit{ et } \{b\} < 1, \\ &= \{b\}([b] + 1 - x). \quad \square \end{aligned}$$

**Corollaire 2.4.7.** [128] *Si* $X^{\leq} = \{(x,\ y) \in \mathbb{Z} \times \mathbb{R}_+ :\ x \leq b + y\}$. *L'inégalité*

$$x \leq [b] + \frac{y}{1 - \{b\}}$$

*est valide pour* $X^{\leq}$.

**Preuve.** En réécrivant l'inégalité $x \leq b + y$ comme $y - x \geq -b$ et en observant que $-b - [-b] = 1 - \{b\}$, on obtient d'après la proposition 2.4.6 que $\frac{y}{1-\{b\}} - x \geq [-b] + 1 = -[b]$.

Ainsi, on voit que quand la variable continue $y = 0$, on obtient l'inégalité arrondie entière.

**Exemple 2.4.4.** Soit l'ensemble $\{(x, y) \in \mathbb{Z}^4_+ \times \mathbb{R}_+ : 2x_1 + 2x_2 + x_3 + x_4 + \frac{y}{11} \geq \frac{72}{11}\}$. En utilisant la proposition 2.4.6 avec $[b] = 6$ et $\{b\} = \frac{6}{11}$, on obtient l'inégalité valide

$$\frac{y}{11} \geq \frac{6}{11}(7 - 2x_1 - 2x_2 - x_3 - x_4).$$

**L'inégalité d' arrondi en nombres entiers et mixtes** (*The Mixed Integer Rounding* (*MIR*) *Inequality*)

Considérons l'ensemble

$$X^{MIR} = \{(x, y) \in \mathbb{Z}^2_+ \times \mathbb{R}_+ :\ a_1x_1 + a_2x_2 \leq b + y\},$$

où $a_1,\ a_2$ *et* $b$ sont des réels et $b \notin \mathbb{Z}$.

**Proposition 2.4.8.** [128] *Supposons que* $\{a_1\} \leq \{b\} \leq \{a_2\}$, *alors*

$$[a_1]x_1 + ([a_2] + \frac{\{a_2\} - \{b\}}{1 - \{b\}})x_2 \leq [b] + \frac{y}{1 - \{b\}} \tag{2.6}$$

*est valide pour* $X^{MIR}$.

**Preuve.** $(x,\ y) \in X^{MIR}$ satisfait $[a_1]x_1 + ([a_2] + 1)x_2 \leq b + y + (1 - \{a_2\})x_2$ car $x_1 \geq 0$, et $a_2 = [a_2] + 1 - (1 - \{a_2\})$. Maintenant le corollaire de la proposition 2.4.76 donne

$$[a_1]x_1 + ([a_2] + 1)x_2 \leq [b] + (y + (1 - \{a_2\})x_2)/(1 - \{b\}),$$

ce qui prouve l'inégalité. $\square$

**Exemple 2.4.5.** Considérons l'ensemble $X = \{(x,\ y) \in \mathbb{Z}^3_+ \times \mathbb{R}^1_+ :\ \frac{10}{3}x_1 + 1x_2 + \frac{11}{4}x_3 \leq \frac{21}{2} + y\}$. On a $\{b\} = \frac{1}{2},\ \{a_1\} = \frac{1}{3},\ \{a_2\} = 0,\ \{a_3\} = \frac{3}{4}$, et d'après la proposition 2.4.8, on déduit que l'inégalité

$$3x_1 + x_2 + \frac{5}{2}x_3 \leq 10 + 2y$$

est valide pour $X$.

**Inégalités de Chv$\acute{a}$tal-Gomory pour ensembles en nombres entiers et mixtes**

Supposons qu'on a la région en nombres entiers et mixtes

$$X = \{(x,\ y) \in \mathbb{Z}^n_+ \times \mathbb{R}^p_+ :\ Ax + Dy \leq b\},$$

où $(A|D|b)$ est une matrice rationnelle d'ordre $m \times (n + p + 1)$. Notre objectif dans cette section est de développer une procédure pour générer des inégalités valides pour $X$. Notons que la procédure $C-G$ ne fonctionne pas dans le cas de variables mixtes. En particulier, on ne peut arrondir le coté droit de l'inégalité à sa partie entière lorsque tous les coefficients dans le coté gauche sont entiers. Cependant, on sera capable d'obtenir une procédure, reliée à la procédure disjonctive, qui généralise la procédure C-G.
Pour commencer, considérons l'exemple avec $X$ défini comme suit :

$$3x_1 - 7x_2 + 2x_3 + y_1 - y_2 \leq 4\frac{1}{3}, \quad x \in \mathbb{Z}^3_+,\ y \in \mathbb{R}^2_+.$$

En l'absence de la variable $y$, on obtient l'inégalité valide $3x_1 - 7x_2 + 2x_3 \leq 4$. Peut-on trouver une inégalité valide pour $X$ de la forme

$$3x_1 - 7x_2 + 2x_3 + \mu^+ y_1 - \mu^- y_2 \leq 4? \tag{2.7}$$

**i.** Une borne sur $\mu^+$. Supposons qu'il y a une solution avec $3x_1 - 7x_2 + 2x_3 = 4$, $y_2 = 0$, et $y_1 > 0$. L'inégalité (2.7) peut être valide si $4 + \mu^+ y_1 - 0 \leq 4$ ou $\mu^+ \leq 0$.

**ii.** Une borne sur $\mu^-$. Supposons qu'il y a une solution réalisable avec $3x_1 - 7x_2 + 2x_3 = 5$, $y_1 = 0$, et $y_2 = \frac{2}{3}$. La validité de (2.7) implique que $5 - \frac{2}{3}\mu^- \leq 4$ ou $\mu^- \geq \frac{3}{2}$.

L'exemple indique que $\mu^+ \leq 0,\ \mu^- \geq 1/(1 - \{b\})$.

**Proposition 2.4.9.** [105] *Soit* $X = \{(x,\ y) \in \mathbb{Z}^n_+ \times \mathbb{R}^p_+ :\ \sum_{j \in J} a_j x_j + \sum_{j \in K} d_j y_j \leq b\}$, *où* $J = \{1,\ ...,\ n\}$, $K = \{1,\ ...,\ p\}$, *et* $a_j,\ d_j,\ b \in \mathbb{R}$ *pour tout* $j$. *L'inégalité*

$$\sum_{j \in J} [a_j] x_j + \frac{1}{1 - \{b\}} \sum_{j \in K^-} d_j y_j \leq [b], \tag{2.8}$$

*où* $K^- = \{j \in K :\ d_j < 0\}$, *est valide pour* $X$.

**Preuve.** Supposons $\sum_{j\in K} d_j y_j > \{b\} - 1$. Alors

$$\sum_{j\in J}[a_j]x_j \leq \sum_{j\in J} a_j x_j \leq b - \sum_{j\in K} d_j y_j < b - (\{b\} - 1) = [b] + 1.$$

Puisque $\sum_{j\in J}[a_j]x_j$ est entier, on a $\sum_{j\in J}[a_j]x_j \leq [b]$. En ajoutant cette inégalité à $\frac{1}{1-\{b\}} \sum_{j\in K^-} d_j y_j \leq 0$, cela donne (2.8).
Maintenant supposons que $\sum_{j\in K} d_j y_j \leq \{b\} - 1$, ainsi $\sum_{j\in K^-} d_j y_j \leq \{b\} - 1$. Par conséquent, on aura

$$\begin{aligned}
\sum_{j\in J}[a_j]x_j + \frac{1}{1-\{b\}} \sum_{j\in K^-} d_j y_j &\leq \sum_{j\in J} a_j x_j + \frac{1}{1-\{b\}} \sum_{j\in K^-} d_j y_j \\
\leq \ b - \sum_{j\in K} d_j y_j + \frac{1}{1-\{b\}} \sum_{j\in K^-} d_j y_j &\leq b + (\sum_{j\in K^-} d_j y_j)(\frac{1}{1-\{b\}} - 1) \\
= \ b + \frac{\{b\}}{1-\{b\}}(\sum_{j\in K^-} d_j y_j) &\leq b - \{b\} = [b]. \quad \square
\end{aligned}$$

**Exemple 2.4.6.** Soit l'ensemble $X = \{(x_1, y_1) \in \mathbb{Z}_+ \times \mathbb{R}_+ : \ x_1 + y_1 \leq \frac{5}{2}\}$ . De (2.8), on obtient l'inégalité valide $x_1 \leq 2$ (voir figure 2.7). Notons que
$\{(x_1, y_1) \in \mathbb{R}_+^2 : \ x_1 + y_1 \leq \frac{5}{2}, \ x_1 \leq 2\} = conv\{(x_1, y_1) \in \mathbb{Z}_+ \times \mathbb{R}_+ : \ x_1 + y_1 \leq \frac{5}{2}\}$.
Maintenant, soit $X = \{(x_1, y_1) \in \mathbb{Z}_+ \times \mathbb{R}_+ : \ x_1 - y_1 \leq \frac{5}{2}\}$ . De (2.8), on obtient l'inégalité valide $x_1 - 2y_1 \leq 2$ (voir figure 2.7). Notons que
$\{(x_1, y_1) \in \mathbb{R}_+^2 : \ x_1 - y_1 \leq \frac{5}{2}, \ x_1 - 2y_1 \leq 2\} = conv\{(x_1, y_1) \in \mathbb{Z}_+ \times \mathbb{R}_+ : \ x_1 - y_1 \leq \frac{5}{2}\}$.

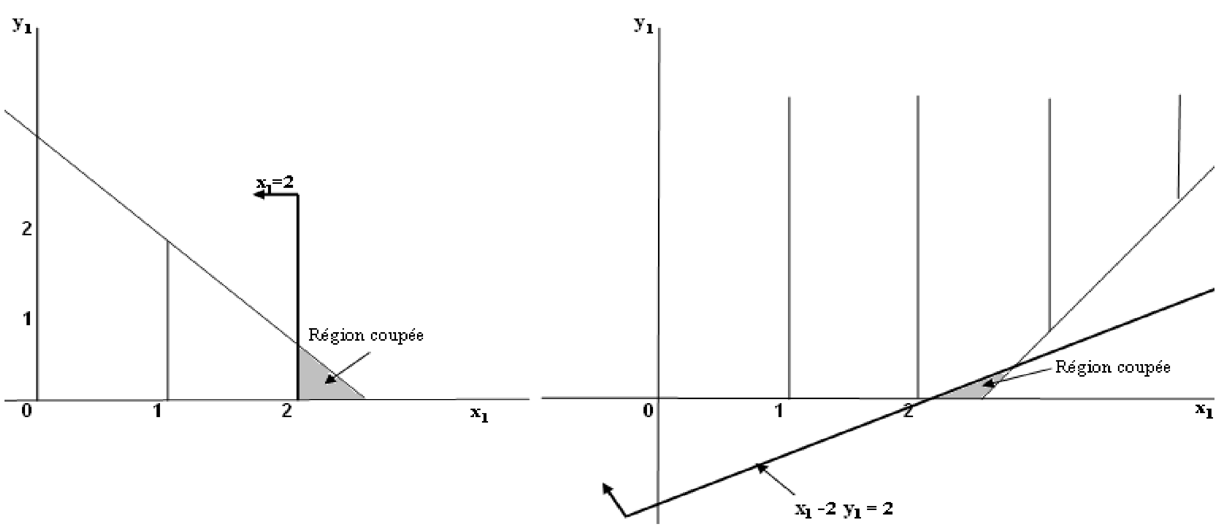


FIG. 2.7 –

L'exemple 2.4.6 illustre la proposition suivante.

**Proposition 2.4.10.** [105] *Soit* $X = \{(x, y) \in \mathbb{Z}_+^n \times \mathbb{R}_+^p : \sum_{j \in J} a_j x_j + \sum_{j \in K} d_j y_j \leq b\}$*, où* $a_j \in \mathbb{Z}$ *pour* $j \in J$*, pgcd*$\{a_1, \ ..., \ a_n\} = 1$*, et* $b \notin \mathbb{Z}$*. Alors* (2.8) *est une facette de conv*$X$*.*

### 2.4.5 Les coupes de Gomory en nombres entiers et mixtes

Comme dans le cas d'un programme en nombres entiers, chaque ligne du tableau optimal associé au programme linéaire, dans laquelle une variable est basique mais fractionnaire, peut être utilisée pour générer une coupe éliminant la solution optimale du programme linéaire. Spécifiquement, chaque ligne produit un ensemble de la forme :

$$X^i = \{(x_i, \ x, \ y) \in \mathbb{Z} \times \mathbb{Z}_+^{n_1} \times \mathbb{R}_+^{n_2} : \ x_i + \sum_{j \in N_1} \overline{a}_{ij} x_j + \sum_{j \in N_2} \overline{a}_{ij} y_j = \overline{a}_{i0}\}.$$

**Proposition 2.4.11.** *Si* $\overline{a}_{i0} \notin \mathbb{Z}$*, la coupe de Gomory en nombres entiers et mixtes :*

$$\sum_{\{\overline{a}_{ij}\} \leq \{\overline{a}_{i0}\}} \{\overline{a}_{ij}\} x_j + \sum_{\{\overline{a}_{ij}\} > \{\overline{a}_{i0}\}} \frac{\{\overline{a}_{i0}\}(1 - \{\overline{a}_{ij}\})}{1 - \{\overline{a}_{i0}\}} x_j + \sum_{\overline{a}_{ij} > 0} \overline{a}_{ij} y_j + \sum_{\overline{a}_{ij} < 0} \frac{\{\overline{a}_{i0}\}}{1 - \{\overline{a}_{i0}\}} \overline{a}_{ij} y_j \geq \{\overline{a}_{i0}\}$$

*est valide pour* $X^i$*.*

**Preuve.** L'inégalité d'arrondi en nombres entiers et mixtes (2.6) pour $X^i$ est

$$x_i + \sum_{\{\overline{a}_{ij}\} \leq \{\overline{a}_{i0}\}} [\overline{a}_{ij}] x_j + \sum_{\{\overline{a}_{ij}\} > \{\overline{a}_{i0}\}} ([\overline{a}_{ij})] + \frac{\{\overline{a}_{ij}\} - \{\overline{a}_{i0}\}}{1 - \{\overline{a}_{i0}\}}) x_j + \sum_{\overline{a}_{ij} < 0} \frac{\overline{a}_{ij}}{1 - \{\overline{a}_{i0}\}} y_j \leq [\overline{a}_{i0}].$$

Après la substitution de $x_i$, on obtient le résultat demandé. □

**Exemple 2.4.7.** Considérons le programme en nombres entiers et mixtes :

$$\begin{cases} z = 4x_1 - x_2 \longrightarrow \max, \\ 7x_1 - 2x_2 \leq 14, \\ \qquad\quad x_2 \leq 3, \\ 2x_1 - 2x_2 \leq 3, \\ x_1 \in \mathbb{Z}_+, \ x_2 \geq 0. \end{cases}$$

En le résolvant comme un programme linéaire, cela donne :

| | | c | 4 | −1 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| $c_B$ | $x_B$ | $b$ | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ |
| 4 | $x_1$ | $\frac{20}{7}$ | 1 | 0 | $\frac{1}{7}$ | $\frac{2}{7}$ | 0 |
| −1 | $x_2$ | 3 | 0 | 1 | 0 | 1 | 0 |
| 0 | $x_5$ | $\frac{23}{7}$ | 0 | 0 | $\frac{-2}{7}$ | $\frac{10}{7}$ | 1 |
| $z = \frac{59}{7}$ | | $E$ | 0 | 0 | $\frac{-4}{7}$ | $\frac{-1}{7}$ | 0 |

La variable de base $x_1$ est fractionnaire et la première ligne donne la coupe $(MIR)$ : $x_1 \leq 2$, qui après élimination de $x_1$ devient la coupe de Gomory en nombres entiers et mixtes :

$$\frac{1}{7}x_3 + \frac{2}{7}x_4 \geq \frac{6}{7}.$$

En ajoutant cette coupe et en réoptimisant, on aura la solution $x = (2,\ \frac{1}{2})$, qui est par conséquent optimale pour le programme en nombres entiers et mixtes.

### 2.4.6 Procédure d'arrondi en nombres entiers et mixtes (MIR)

Maintenant on donne une procédure basée sur (2.8) pour la génération des inégalités valides pour l'ensemble $X = \{(x, y) \in \mathbb{Z}_+^n \times \mathbb{R}_+^p :\ Ax + Dy \leq b\}$.

*Etape1 :* Les inégalités

$$\sum_{j \in J}(ua_j)x_j + \sum_{j \in K}(ud_j)y_j \leq ub \ \ \textit{sont valides} \ \ \textit{pour} \ \ \textit{tout}\ u \in \mathbb{R}_+^m.$$

*Etape 2 :* Etant données les deux inégalités valides

$$\sum_{j \in J}\pi_j^i x_j + \sum_{j \in K}\mu_j^i y_j \leq \pi_0^i \ \ \textit{pour} \ \ i = 1,\ 2, \tag{2.9}$$

on construit la troisième inégalité valide

$$\sum_{j \in J}[\pi_j^2 - \pi_j^1]x_j + \frac{1}{1 - \{\pi_0^2 - \pi_0^1\}}\left(\sum_{j \in J}\pi_j^1 x_j + \sum_{j \in K}\min(\mu_j^1,\ \mu_j^2)y_j - \pi_0^1\right) \leq [\pi_0^2 - \pi_0^1]. \tag{2.10}$$

**Proposition 2.4.12.** [105] *Etant données les deux inégalités valides* (2.9) *pour* $X$*, il s'ensuit que* (2.10) *est aussi valide pour* $X$*.*

**Preuve.** Comme (2.9) est valide pour $X$ et $y \geq 0$, il s'ensuit que

$$\sum_{j \in J}\pi_j^i x_j + \sum_{j \in K}\min(\mu_j^1,\ \mu_j^2)y_j \leq \pi_0^i \ \ \textit{pour} \ \ i = 1,\ 2 \tag{2.11}$$

est valide pour $X$. Réécrivons (2.11) pour $i = 2$ comme

$$\sum_{j \in J}(\pi_j^2 - \pi_j^1)x_j - \left(\pi_0^1 - \sum_{j \in J}\pi_j^1 x_j - \sum_{j \in K}\min(\mu_j^1,\ \mu_j^2)y_j\right) \leq \pi_0^2 - \pi_0^1.$$

Maintenant (2.11) avec i=1 implique

$$s = \pi_0^1 - \sum_{j \in J} \pi_j^1 x_j - \sum_{j \in K} \min(\mu_j^1,\ \mu_j^2) y_j \geq 0.$$

Ainsi on peut appliquer la proposition 2.4.9 à

$$\sum_{j \in J} (\pi_j^2 - \pi_j^1) x_j - s \leq \pi_0^2 - \pi_0^1$$

pour obtenir (2.10). $\square$

**Exemple 2.4.8.** $X = \{x \in \{0,\ 1\}^2,\ y \in \mathbb{R}_+^2 :\ y_1 + y_2 \leq 7,\ y_i \leq 5x_i,\ pour\ i = 1,\ 2\}$. En utilisant l'étape 1, on obtient les deux inégalités valides

$$\frac{1}{3}(y_1 + y_2) \leq \frac{7}{3},$$

$$-\frac{5}{3}(x_1 + x_2) + \frac{1}{3}(y_1 + y_2) \leq 0.$$

En prenant l'une d'entre elles comme la 1ère (i=1) inégalité, et l'autre comme la 2ème (i=2) inégalité, on obtient à partir de (2.10) l'inégalité valide, donnée par :

$$[-\frac{5}{3}](x_1 + x_2) + \frac{1}{1 - \frac{2}{3}}(\frac{1}{3}(y_1 + y_2) - \frac{7}{3}) \leq [-\frac{7}{3}],$$

c'est-à-dire

$$-2(x_1 + x_2) + (y_1 + y_2) \leq 4.$$

Les coupes mixtes de Gomory sont plus fortes et contiennent les coupes de Chvátal-Gomory dans le cas des problèmes en nombres entiers. Comme pour celles-ci le problème général de séparation est NP-complet, mais est soluble en temps polynomial pour les solutions de base de la relaxation linéaire.

Comme pour le cas entier, Gomory a pu établir un algorithme de résolution exacte des problèmes mixtes basé sur la séparation de ces coupes dans les années 60. Cependant, c'est surtout depuis une dizaine d'années que ces coupes ont été utilisées pour la résolution de problèmes par branch-and-cut. Suite aux premiers travaux de Balas et al., étudiant l'utilisation de ces coupes dans un tel cadre ([13]), de nombreux travaux théoriques et pratiques ont été conduits pour améliorer qualitativement l'efficacité des algorithmes de séparation ([90, 34]).

## 2.5 Les méthodes de recherche arborescente par séparation et évaluation

Cette approche a été proposée par Little et al. pour résoudre le problème du voyageur de commerce puis, repris par d'autres sous différentes variantes.
Essentiellement, il s'agit de diviser, c'est-à-dire séparer l'ensemble de toutes les solutions réalisables en sous-ensembles plus petits (régions réalisables plus restreintes) et mutuellement exclusifs et en tentant d'identifier et d'éliminer de l'ensemble des solutions réalisables du programme linéaire, certaines parties qui ne contiennent pas des solutions entières réalisables.
L'implémentation de l'algorithme de Branch and Bound peut être vu comme un arbre avec au nœud de la racine le problème original $(PLE)$. L'arbre est construit d'une manière itérative avec de nouveaux nœuds formés par la séparation d'un nœud existant dont lequel la solution optimale du problème relaxé associé n'est pas entière.

#### Choix de la variable de séparation

Durant le processus de partitionnement, une variable doit être choisie pour la séparation, il est clair que le choix de cette variable influe sur le temps d'exécution de l'algorithme. Plusieurs approches ont été développées, les plus courantes sont :

- La variable qui possède la partie fractionnaire la plus grande par rapport à l'entier le plus proche est choisie.
- *Pénalités de Driebck-Tomlin* : Les pénalités donnent une borne inférieure sur la dégradation de la valeur de la fonction objectif pour chaque branchement d'une variable donnée. La pénalité est le coût du pivot dual dont on a besoin pour éliminer la variable à valeur fractionnaire de la base. La pénalité supérieure $u_p$ lorsqu'on force la valeur de la $k$ème variable de base à une valeur supérieure, après l'ajout de la contrainte $x_k \geq [x_k] + 1$, est égale à :

$$u_k = \min_{j:\ a_{kj}<0} \frac{(1 - \{x_k\})\overline{c}_j}{-a_{kj}},$$

où $\overline{c}_j$ est le coût réduit de la variable $x_j$ et les $a_{kj}$ sont les coefficients de la matrice transformée de la $k$ème ligne du tableau optimal du problème relaxé $(PL)$. La pénalité inférieure $d_k$, obtenue après l'ajout de la contrainte $x_k \leq [x_k]$, est égale à :

$$d_k = \min_{j:\ a_{kj}>0} \frac{\{x_k\}\bar{c}_j}{a_{kj}}.$$

Une fois les pénalités sont calculées, une variété de règles peuvent être utilisées pour sélectionner la variable de branchement.

### Principe d'évaluation

L'évaluation est effectuée en résolvant le programme linéaire relaxé associé à $(PLE)$. Cependant, il serait très inefficace de recommencer à chaque étape la résolution d'un nouveau programme linéaire sans tirer parti des résultats obtenus jusqu'alors. Quand on a à résoudre les programmes linéaires issus de la séparation d'un nœud donné on connaît la solution optimale $x^r$ associée au $(PL)$ en ce nœud, cette solution serait encore optimale pour $(PL^1)$ et $(PL^2)$ si seulement elle était réalisable : une contrainte est violée dans chaque programme linéaire. On est donc dans un cas d'application d'un algorithme dual.
A l'étape $i$, on résout le programme linéaire associe à $(PLE^i)$. Deux cas se présentent :

- La solution obtenue est entière. Elle est une borne inférieure au programme $(PLE)$ et en même temps un majorant à toutes les solutions issues de la branche $(PLE^i)$. On coupe sa branche.
- La solution obtenue n'est pas entière. Alors, $(PLE^i)$ est candidat à la séparation.

### Choix d'une stratégie d'exploration des nœuds

L'exploration des nœuds de l'arborescence des solutions obéit à une stratégie donnée. Dans la littérature, on distingue trois types de stratégies : Profondeur d'abord, largeur d'abord et meilleur d'abord.
Dans la stratégie *profondeur d'abord*, on fait la séparation d'un nœud tant que celle-ci peut se faire, c'est-à-dire, jusqu'au moment où l'on coupe sa branche, pour ensuite revenir à une autre branche. A chaque étape on change de niveau.
Dans la stratégie *largeur d'abord*, on résoud tous les programmes d'un même niveau, puis on sépare tous les nœuds séparables pour ensuite résoudre les programmes du niveau suivant.
Dans le cas des deux stratégies précédentes, l'ordre d'exploration est défini au départ. Par contre, dans la stratégie *meilleur d'abord*, il ne l'est pas. On sépare le nœud ayant la

meilleure valeur de la fonction objectif.

L'inconvénient de la stratégie d'exploration profondeur d'abord est de ne pas tenir compte de la fonction d'évaluation. C'est-à-dire qu'on peut être amené à traiter des sous-ensembles dont l'évaluation est très médiocre et qui ont peu de chance de contenir une solution optimale. En revanche elle possède l'avantage d'obtenir rapidement une solution réalisable.

On trouve aussi dans la littérature d'autres stratégies d'exploration, comme par exemple, choisir le nœud qui possède la somme des parties fractionnaires la plus élevée, cette dernière est calculée comme suit : $f = \sum_{j=1}^{n} \min(\{x_j\}, 1 - \{x_j\})$.

Considérons le problème en nombres entiers suivant :

$$(PLE) \qquad z = \max\{c'x : \; x \in S\}, \quad \text{avec } S = \{x \in \mathbb{Z}_+^n : \; Ax \leq b\}.$$

**Proposition 2.5.1.** [128] *Soit $(PLE^i)$ le programme en nombres entiers associé au nœud $i$ :*

$$(PLE^i) \qquad z^i = \max\{c'x : \; x \in S^i\},$$

*où $(S^i)_{i=1}^k$ est une division de $S$. Alors $z = \max\limits_{i=\overline{1,k}} z^i$.*

Soit $(LR^i)$ une relaxation de $(PLE^i)$ :

$$(LR^i) \qquad z_X^i = \max\{c'x : \; x \in X^i\},$$

on a alors $S^i \subseteq X^i$ et $z_X^i \geq c'x$ pour tout $x \in S^i$.

**Proposition 2.5.2.** [128] *Le nœud correspondant à $S^i$ peut être coupé si une des trois conditions suivantes est vérifiée.*

1. *$(LR^i)$ est irréalisable.*
2. *La solution optimale $x^i$ de $(LR^i)$ satisfait $x^i \in S^i$ et $z_X^i = c'x^i$.*
3. *$z_X^i \leq \underline{z}$, où $\underline{z}$ est la valeur de la meilleure solution entière réalisable trouvée jusqu'alors pour $(PLE)$.*

**Preuve.** La condition 1 implique $S^i = \emptyset$. La condition 2 implique que $x^i$ est une solution optimale de $(PLE^i)$. La condition 3 implique $z^i \leq \underline{z}$. $\square$

Soit $(DP^i)$ un dual de $(PLE^i)$.

**Proposition 2.5.3.** [128] *Le nœud correspondant à $S^i$ peut être coupé si une des deux conditions suivantes est vérifiée :*

1. *La valeur de la fonction objectif de $(DP^i)$ et non bornée.*
2. *$(DP^i)$ possède une solution réalisable de valeur inférieure ou égale à $\underline{z}$.*

**Preuve.** La condition 1 implique $S^i = \emptyset$. La condition 2 implique $z^i \leq \underline{z}$. $\square$

Considérons maintenant le problème en nombres mixtes suivant :

$$(PLM) \qquad z = \max\{c'x + d'y : (x, y) \in S\}, \quad avec\ S = \{(x, y) \in \mathbb{Z}^n_+ \times \mathbb{R}^p_+ : Ax + Dy \leq b\},$$

ainsi que le programme linéaire relaxé associé $(LR)$ et sa valeur optimale $z_X$ sur l'ensemble $X = \{(x, y) \in \mathbb{R}^n_+ \times \mathbb{R}^p_+ : Ax + Dy \leq b\}$.

Premièrement, on décrit le principe général de "diviser et conquérir" de l'algorithme de branch and bound pour la résolution d'un $(PLM)$.

### Algorithme général de Branch and Bound

**(i)** La borne supérieure initiale sur $z$ est obtenue par la valeur de la solution optimale $z_X$ de la relaxation linéaire $(LR)$. Ce programme est facile à résoudre (en terme de complexité et dans la pratique). Soit $(x^*,\ y^*)$ la solution optimale de $(LR)$.
On suppose ici sans perte de généralité que $(LR)$ est borné.
On essaye de résoudre $(PLM)$ en résolvant une séquence de programmes linéaires.

**(ii)** Si $x^* \in \mathbb{Z}^n_+$, alors elle est réalisable pour $(PLM)$ comme $(x^*,\ y^*) \in S$, elle est aussi une borne inférieure sur $z$. Donc $(x^*, y^*)$ est une solution optimale pour $(PLM)$, car dans ce cas la borne inférieure et supérieure sont égales.

**(iii)** Autrement, $x^* \notin \mathbb{Z}^n_+$, alors la solution $(x^*,\ y^*)$ n'est pas réalisable pour $(PLM)$. On essaye d'éliminer cette solution de $(LR)$ en additionnant des contraintes linéaires.
Soit $x_j^*$ avec $j \in \{1,\ ...,\ n\}$ une variable à valeur fractionnaire dans la solution $(x^*,\ y^*)$ de $(LR)$.
***Etape de branchement.*** Pour éliminer la solution $(x^*,\ y^*)$, aussi bien que toutes solutions $[x_j^*] < x_j < [x_j^*] + 1$, on remplace l'ensemble $X$ par l'union de deux ensembles disjoints $X^0$ et $X^1$, où

$$X^0 = X \cap \{(x,\ y) \in \mathbb{R}^n_+ \times \mathbb{R}^p_+ :\ x_j \leq [x_j^*]\}$$

et

$$X^1 = X \cap \{(x,\ y) \in \mathbb{R}_+^n \times \mathbb{R}_+^p :\ x_j \geq [x_j^*] + 1\}.$$

La variable $x_j^*$ est dite la variable de branchement, et les contraintes $x_j \leq [x_j^*]$ et $x_j \geq [x_j^*] + 1$ sont dites les contraintes de branchement. On peut maintenant remplacer la recherche de la meilleure solution mixte dans $X$ par la recherche de la meilleure solution dans $X^0 \cup X^1$.

**(iv)** On cherche maintenant la meilleure solution dans une des formulations de la liste $L = \{X^0,\ X^1\}$. On continue l'approche de décomposition de la même façon. Cela exige d'analyser chaque formulation de la liste $L$ séparément.

***Itération principale.*** Soit $L$ la liste des formulations, et soit $\underline{z}$ la valeur de la fonction objectif de la meilleure solution mixte trouvée. Tant qu'on ne connaît pas de solution réalisable pour $(PLM)$, poser $\underline{z} = -\infty$.

***Etape de sélection et de résolution.*** On choisit une formulation $V$ de la liste $L$, et on résout le programme linéaire correspondant pour obtenir $z_V$ correspondant à la solution optimale $(x^V,\ y^V)$. Cette valeur $z_V$ est une borne supérieure sur la valeur de la meilleure solution mixte dans l'ensemble $V$.

***Etape d'élimination.*** Plusieurs cas peuvent se produire en examinant l'ensemble $V$ :

**a.** Si $z_V \leq \underline{z}$, alors la meilleure solution dans $V$ ne peut être strictement meilleure que $\underline{z}$, puisque $z_V$ est une borne supérieure sur la valeur de la meilleure solution dans $V$. Cependant, on a pas besoin de considérer les solutions mixte dans $V$, et on élimine simplement $V$ de la liste $L$.

**b.** Comme cas spécial, du précédant, quand $V$ est vide, on élimine aussi $V$ de la liste.

**c.** Si $z_V > \underline{z}$ et $x^V \in \mathbb{Z}_+^n$, ainsi cette solution améliore la meilleure solution connue jusqu'à présent. Alors, on a pas besoin de décomposer $V$. On met à jour la meilleure solution on posant $\underline{z} = z_V$, et on retire $V$ de la liste $L$.

**d.** Si $z_V > \overline{z}$ et $x^V \notin \mathbb{Z}_+^n$, alors la meilleure solution mixte dans $V$ peut améliorer la borne inférieure $\underline{z}$. Cependant, on a besoin de décomposer le problème, retirer $V$ de la liste $L$, et rajouter les deux ensembles $V^0$et $V^1$ à la liste $L$, obtenus par branchement comme décrit précédemment.

**(V) Arrêt.** L'algorithme s'arrête quand la liste $L$ des problèmes est vide. Cela est garantie de se produire en un nombre fini d'étapes, si les variables entières sont bornées. **Temps d'exécution.** Théoriquement, l'algorithme de branch and bound requiert un nombre d'itérations qui est exponentielle en le nombre de variables entières ($n$). Chaque itération consiste à la résolution d'un programme linéaire de la liste $L$ qui peut se faire en un temps polynomial avec un algorithme approprié.

**Exemple**

Soit à résoudre le problème en nombres entiers suivant :

$$(PLE) \quad z = \max\{3x + 4y : \quad (x, y) \in S\},$$

avec $S = \{(x, y) \in \mathbb{Z}^2_+ \ : \quad 2x + y \leq 6,\ 2x + 3y \leq 9\}$.
Le programme linéaire associé est :

$$(LR) \quad z = \max\{3x + 4y : \quad (x, y) \in X\},$$

avec $X = \{(x, y) \in \mathbb{R}^2_+ \ : \quad 2x + y \leq 6,\ 2x + 3y \leq 9\}$.
Le domaine associé est délimité par $OABC$, où $O = (0, 0),\ A = (0, 3),\ B = (\frac{9}{4}, \frac{3}{2})$ *et* $C = (3, 0)$. La solution optimale de $(LR)$ est atteinte au point $B = (\frac{9}{4}, \frac{3}{2})$ avec $z_B = \frac{51}{4}$. Cette solution n'est pas entière. Posons $\underline{z} = -\infty$.
a) On choisit de séparer sur la variable $y$. On crée alors deux nouveaux sommets :

$$(LR^0) = \{LR,\ y \leq 1\} \ \ et \ \ (LR^1) = \{LR,\ y \geq 2\}.$$

On résout $(LR^1)$ et on obtient le domaine délimité par $DAB$, où $D = (0, 2)$ *et* $E = (\frac{3}{2}, 2)$. La solution optimale est atteint au point $E = (\frac{3}{2}, 2)$ avec $z_E = \frac{25}{2}$.
Cette solution n'est pas entière. On explore $(LR^1)$ en le séparant sur la variable $x$ qui n'est pas entière. Cette séparation crée deux nouveaux sommets représentant les programmes suivants :

$$(LR^2) = \{LR^1, x \leq 1\} \ \ et \quad (LR^3) = \{LR^1,\ x \geq 2\}.$$

On résout $(LR^2)$ qui donne le domaine délimité par $DAFG$, où $F = (1, 2)$ *et* $G = (1, \frac{7}{3})$. La solution optimale est atteinte au point $G = (1, \frac{7}{3})$ avec $z_G = \frac{37}{3}$, qui est non entière. On sépare sur la variable $y$ en créant deux nouveaux sommets associés aux programmes :

$$(LR^4) = \{LR^2, y \leq 2\} \ \ et \ \ (LR^5) = \{LR^2,\ y \geq 3\}.$$

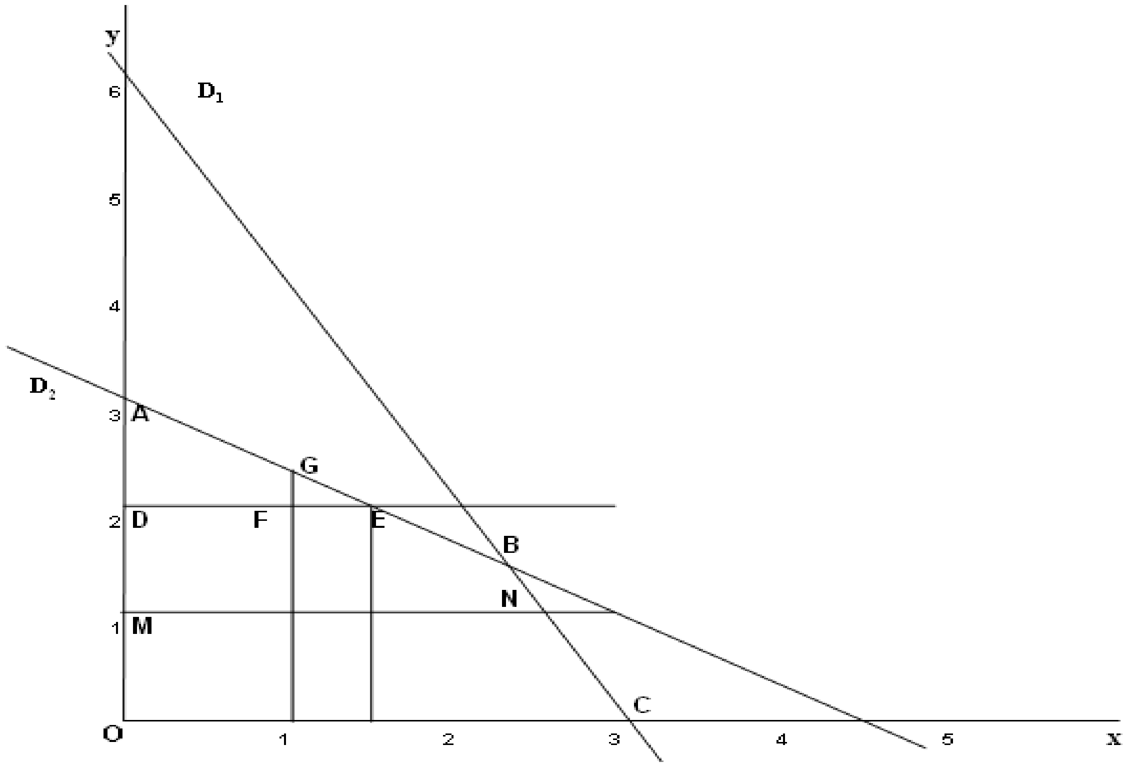


FIG. 2.8 – Domaine des solutions

On résout $(LR^4)$, le domaine est réduit au segment $[D,\ F]$. La solution optimale est atteinte au point $F = (1, 2)$. Cette solution est entière, on coupe le nœud correspondant. Comme $z_F > \underline{z}$, alors $z_F = 11$ est une borne inférieure pour le programme $(PLE)$, on pose alors $\underline{z} = 11$. On remonte au niveau 2, on explore à nouveau $(LR^2)$ en résolvant le programme $(LR^5)$ qui est réduit au point $A = (0, 3)$, cette solution étant entière, on coupe ce nœud. Comme $z_A = 12 > \underline{z} = 11$, elle constitue une nouvelle borne inférieure au programme $(PLE)$. On remonte au niveau 1 et on résout $(LR^3)$ qui est vide, on coupe ce noeud. On remonte au niveau 0 et on résout $(LR^0)$, le domaine correspondant à ce nœud est $OMNC$, où $M = (0, 1)$ *et* $N = (\frac{5}{2}, 1)$. La solution n'étant pas entière et comme $z_N = 11.5 < \underline{z} = 12$, on coupe cette branche.

L'arborescence des solutions est représentée sur la figure 2.9.

Lorsque le programme à résoudre est de grand taille, il est très important de chercher des heuristiques permettant de réduire rapidement l'arborescence. Autrement, non seulement le temps d'exécution devient important mais aussi l'espace mémoire devient insuffisant pour terminer la résolution.

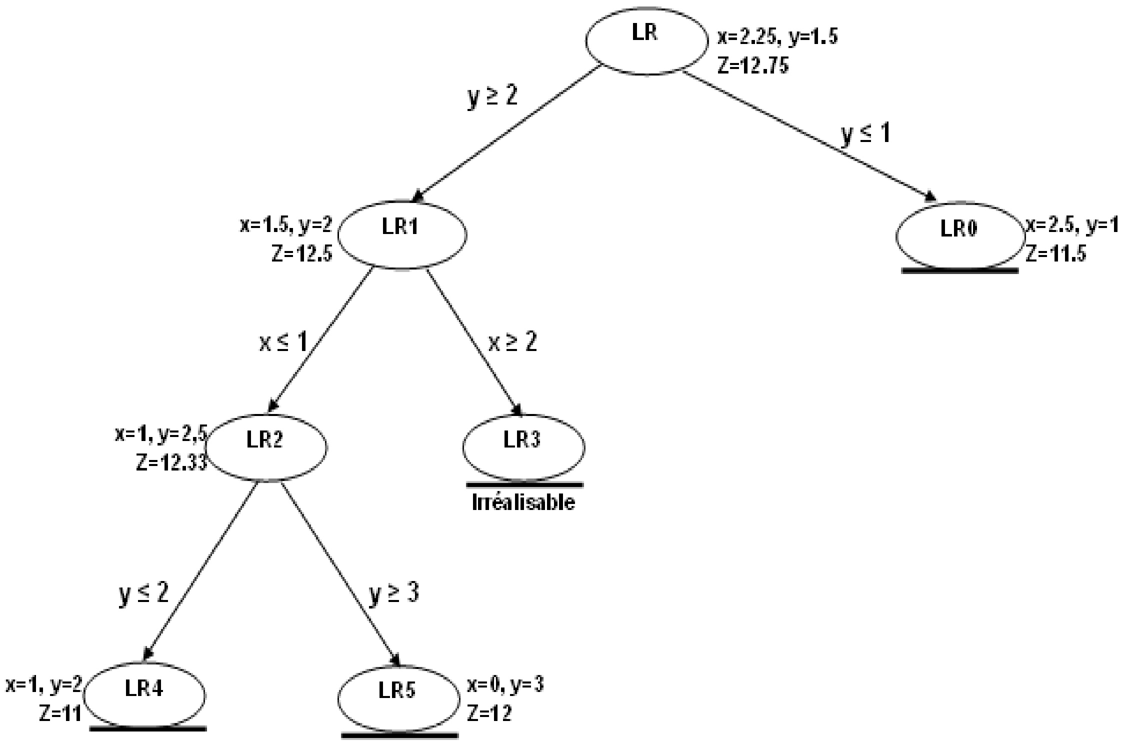


FIG. 2.9 – Arborescence des solutions.

Ces heuristiques interviennent surtout dans la recherche d'une borne inférieure. Cette dernière sert à couper assez rapidement le plus de branches possibles.
Une manière d'opérer est de prospecter le voisinage de la solution optimale initiale du programme linéaire associé. N'importe qu'elle solution entière réalisable voisine de la solution optimale réelle peut être retenue comme borne inférieure. La meilleure solution voisine sera plus intéressante.
Dans l'exemple précédent $x^* = (\frac{9}{4}, \frac{3}{2})$, les solutions entières voisines sont : $x_1 = (2,1)$, $x_2 = (2,2)$, $x_3 = (3,1)$, $x_4 = (3,2)$, la seule solution entière réalisable est $x_1$.

Dans le cas général, considérons le programme linéaire suivant :

$$(LR) \quad \left\{ \begin{array}{l} z \ = c'x = \sum_{j=1}^{n} c_j x_j \longrightarrow \max, \\ \sum_{j=1}^{n} a_{ij} x_j \leq b_i, \ i = \overline{1,\ m}, \\ x_j \geq 0, \ j = \overline{1,\ n}. \end{array} \right.$$

Soit $x^* = (x_1^*,\ \ldots, x_n^*)$ la solution optimale de ce programme. Posons $x_j = [x_j^*] + y_j$, où $y_j = 0$ *ou* 1. Le vecteur $x = (x_1,\ ...,\ x_n)$ est entier. Si, de plus, la solution $x$ est réalisable, alors elle définit une borne inférieure au problème initial. La meilleure borne inférieure est celle qui optimise le programme suivant :

$$(PB) \quad \begin{cases} \sum\limits_{j=1}^{n} c_j y_j \longrightarrow \max, \\ \sum\limits_{j=1}^{n} a_{ij} y_j \leq b_i - \sum_{j=1}^{n} a_{ij}[x_j^*], \ i = \overline{1,\ m}, \\ y_j = 0 \ ou \ 1 \ j = \overline{1,\ n}. \end{cases}$$

En effet, ce problème est déduit en remplaçant $x_j$ dans $(LR)$ et en utilsant le fait que les quantités $c_j[x_j^*]$ et $a_{ij}[x_j^*]$ sont des constantes, pour $j = \overline{1,\ n}$.
L'inconvénient est la résolution pour chaque noeud d'un programme linéaire bivalent en 0 *ou* 1. Ainsi, la recherche de la solution de $(PB)$ est intéressante au début de la résolution (vu que nous n'avons pas encore de solution entière). Il n'est pas nécessaire de la rechercher en tout nœud de l'arborescence, car elle est coûteuse sans que la borne inférieure ne soit améliorée.

## 2.6 Branch and cut

Nous avons vu dans l'algorithme de Branch and Bound que la recherche des solutions consiste, en chaque étape, à éclater le domaine admissible du programme linéaire associé en trois parties, une partie est éliminée puisque elle ne contient pas de solution entière, les deux autres forment les domaines des nouveaux programmes à explorer. Dans les méthodes des coupes, on réduit le domaine en ajoutant à chaque fois une contrainte supplémentaire de manière à ce que tous les points extrêmes du nouveau polyèdre devient entiers.

Cependant, il est possible de combiner approche polyédrique (génération de coupes) et branch and bound, on parle alors d'une méthode de branch and cut. A chaque itération de la recherche arborescente, l'évaluation d'un nœud est calculée par relaxation augmentée de coupes. L'introduction des coupes artificielles (les branchements) permet généralement de déterminer de nouvelles inégalités valides plus profondes pour le sous-espace considéré à chaque nœud . Quoique dans la pratique, rechercher de nouvelles coupes à chaque nœud peut ralentir considérablement la procédure. Un compromis consiste à générer préalablement, à la racine, un certain nombre d'inégalités valides et à un nœud donné qui n'est pas invalidé par la relaxation continue seule, d'ajouter des inégalités qui coupent la solution optimale de ce nœud. Eventuellement, le $(PL)$ peut être réoptimisé et ce, tant qu'il existe des coupes pour la solution fractionnaire courante.

## 2.7 Décomposition de Benders

La décomposition de Benders présente une méthode de résolution des problèmes pour lesquels on peut isoler des variables qui, une fois fixées, simplifient le problème. Par exemple, dans un programme en variables mixtes, on considère généralement les variables entières comme difficiles.
Cette méthode à été initialement développée par Benders en 1962 [18] et par Geoffrion en 1974 [59].

### Reformulation de Benders

Considérons le programme en nombres mixtes suivant :

$$(PLM) \quad z = \max\{c'x \ + \ d'y : \ Ax \ + \ Dy \leq b, \ x \in S \subseteq \mathbb{Z}_+^n, \ y \in \mathbb{R}_+^p\}.$$

La procédure décrite ci dessous montre comment un $(PLM)$ peut être reformulé comme un problème dans $S \times \mathbb{R}_+$ ; Ainsi, il y a une seule variable continue. Cependant, cette formulation contient généralement un nombre énorme de contraintes linéaires.

Si on fixe dans ce programme les variables $x$, on obtient un programme linéaire (en $y$) :

$$(PL_x) \qquad z_x = \max\{d'y : \ \ Dy \leq b - Ax, \ \ y \in \mathbb{R}_+^p\},$$

dont le dual s'écrit :

$$(D_x) \qquad z_D = \min\{u'(b - Ax) : \ \ u \in Q\} \ \ avec\ Q = \{u \in \mathbb{R}_+^m : \ \ Du \geq d\}.$$

D'après la décomposition de Minkowski, un point $u$ appartient au polyèdre $Q$ si et seulement s'il s'écrit comme combinaison linéaire des points extrêmes $u^k$, $k \in \{1, ..., K\}$ et des rayons extrêmes $v^j, \ j \in \{1, ..., J\}$ de $Q$ : $u = \sum_{k=1}^K \lambda_k u^k + \sum\limits_{j=1}^J \mu_j v^j$, avec $\sum\limits_{k=1}^K \lambda_k = 1, \ \lambda_k \geq 0, \ k \in K$ et $\mu_j \geq 0, \ j \in J$.
En supposons que le $(PLM)$ est borné alors $Q$ est non vide et on voit facilement que pour un $x$ donné, s'il existe un rayon extême $(v^j)'(b \ - \ Ax) < 0$ alors $z_D = -\infty$, sinon $D_x$ atteint son minimum en un point extême $u^k$.

**Proposition 2.7.1.** [105] *La fonction $z_x$ est caractérisée comme suit :*

**i.** *Si $Q = \emptyset$, alors $z_x = +\infty$ si $(v^j)'(b - Ax) \geq 0$ pour tout $j \in J$, et $z_x = -\infty$ autrement.*

**ii.** *Si* $Q \neq \emptyset$*, alors* $z_x = \min\limits_{k\in K}(u^k)'(b - Ax) < +\infty$ *si* $(v^j)'(b - Ax) \geq 0$ *pour tout* $j \in J$*, et* $z_x = -\infty$ *autrement.*

Une conséquence immédiate de la proposition (2.7.1) est que quand $Q \neq \emptyset$, le $(PLM)$ peut être equivalent à :

$$\begin{cases} z = c'x + \min\limits_{k\in K}(u^k)'(b - Ax) \longrightarrow \max\limits_{x}, \\ (v^j)'(b - Ax) \geq 0, \quad j \in J, \\ x \in S. \end{cases}$$

Le $(PLM)$ qui, par dualité, correspond à $\max\{c'x + z_D : x \in S\}$, peut donc se reformuler par le programme en variables mixtes donnée par le théorème suivant :

**Théorème 2.7.2.** [105] $(PLM)$ *peut être reformulé comme suit :*

$$(PLM') \quad \begin{cases} z = \eta \longrightarrow \max, \\ \eta \leq c'x + (u^k)'(b - Ax), \quad k \in K, \\ (v^j)'(b - Ax) \geq 0, \quad j \in J, \\ x \in S, \ \eta \in \mathbb{R}. \end{cases}$$

**Preuve.** S'il n'existe pas $x \in S$ tel que $(v^j)'(b - Ax) \geq 0$ pour tout $j \in J$, alors $z_x = -\infty$ pour tout $x \in S$ et $z = -\infty$. S'il existe $x \in S$ tel que $(v^j)'(b - Ax) \geq 0$ pour tout $j \in J$ et $Q = \emptyset$, alors $K = \emptyset$, donc $z = +\infty$ ; autrement $(PLM')$ est équivalent à (2.12). $\square$

$(PLM')$ est la reformulation de Benders. Comme il possède typiquement un nombre énorme de contraintes, une approche naturelle est de considérer les relaxations obtenus en générant seulement les contraintes correspondantes à un nombre réduit de points extrêmes et de rayons extrêmes.

**Exemple 2.7.1.**

$$\begin{cases} z = 5x_1 - 2x_2 + 9x_3 + 2y_1 - 3y_2 + 4y_3 \longrightarrow \max, \\ 5x_1 - 3x_2 + 7x_3 + 2y_1 + 3y_2 + 6y_3 \leq -2, \\ 4x_1 + 2x_2 + 4x_3 + 3y_1 - y_2 + 3y_3 \leq 10, \\ x_j \leq 5, \quad j = \overline{1,3}, \\ x \in \mathbb{Z}^3_+, \ y \in \mathbb{R}^3_+. \end{cases}$$

Ici on suppose que $S = \{x \in \mathbb{Z}^3_+ : \ x_j \leq 5, \quad j = \overline{1,3}\}$.

Dans la figure 2.10 on représente le polyèdre $\{u \in \mathbb{R}^2_+ : \ Du \geq d\}$ :

$$\begin{cases} 2u_1 + 3u_2 \geq 2, \\ 3u_1 - u_2 \geq -3, \\ 6u_1 + 3u_2 \geq 4, \\ u \in \mathbb{R}^2_+. \end{cases}$$

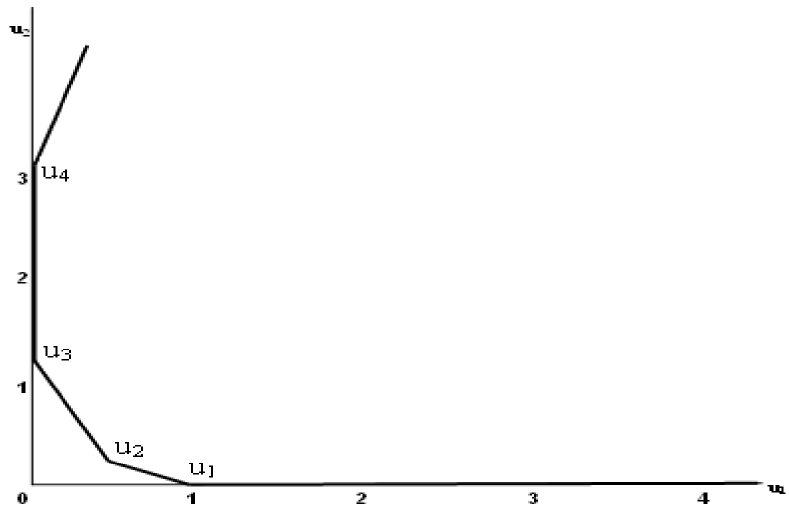


FIG. 2.10 –

Les points extrêmes de ce polyèdre sont $u^1 = (1,\ 0),\ u^2 = (\frac{1}{2},\ \frac{1}{3}),\ u^3 = (0,\ \frac{4}{3})\ et\ u^4 = (0,\ 3)$, ses rayons extrêmes sont $v^1 = (1,\ 0)$ et $v^2 = (1,\ 3)$.
La reformulation résultante du programme en nombres mixtes est :

$$\begin{cases} z = \eta \longrightarrow \max, \\ \eta \leq 5x_1 - 2x_2 + 9x_3 + (-2 - 5x_1 + 3x_2 - 7x_3), \\ \eta \leq 5x_1 - 2x_2 + 9x_3 + \frac{1}{2}(-2 - 5x_1 + 3x_2 - 7x_3) + \frac{1}{5}(10 - 4x_1 - 2x_2 - 4x_3), \\ \eta \leq 5x_1 - 2x_2 + 9x_3 + \frac{4}{3}(10 - 4x_1 - 2x_2 - 4x_3), \\ \eta \leq 5x_1 - 2x_2 + 9x_3 + 3(10 - 4x_1 - 2x_2 - 4x_3), \\ (-2 - 5x_1 + 3x_2 - 7x_3) \geq 0, \\ (-2 - 5x_1 + 3x_2 - 7x_3) + 3(10 - 4x_1 - 2x_2 - 4x_3) \geq 0, \\ x_j \leq 5\ pour\ j = \overline{1,3}, \quad x \in \mathbb{Z}_+^3, \eta \in \mathbb{R}. \end{cases}$$

La solution optimale étant $x = (0,\ 3,\ 1)$ et $\eta = 3$.

Etant donné le nombre de contraintes de $(PLM')$, on optimise itérativement une relaxation $(PL)$ en générant une à une les contraintes à la manière de coupes (coupes de Benders).

## Algorithme de génération de contraintes pour $(PLM')$

**Initialisation** Trouver les ensembles $K^1 \subseteq K$, $J^1 \subseteq J$ (éventuellement vide). Poser $S_R^1 = \{\eta \in \mathbb{R},\ x \in S :\ \eta \leq c'x + (u^k)'(b - Ax)\}\ pour\ k \in K^1,\ \ (v^j)'(b - Ax) \geq 0$ pour $j \in J^1\}$, Poser $t = 1$.

**Itération** $t$ **: Etape** 1 **:** Résoudre la relaxation de $(PLM')$ :

$$(PLM^t) \qquad z^t = \max\{\eta : (\eta,\ x) \in S_R^t,\ x \in S\}.$$

- **a.** Si $(PLM^t)$ est irréalisable, arrêter. $(PLM')$ est irréalisable.
- **b.** Si $(PLM^t)$ est non borné, trouver une solution réalisable $(\eta^t,\ x^t)$ avec $\eta^t > \omega$ pour une certaine valeur grande $\omega$.
- **c.** Autrement poser la solution optimale est $(\eta^t,\ x^t)$.

**Etape** 2 **Séparation :** Résoudre le programme linéaire

$$(PL_{x^t}) \qquad \begin{cases} z_{x^t} = d'y \longrightarrow \max, \\ Dy \leq b - Ax^t, \\ y \in \mathbb{R}_+^p, \end{cases}$$

ou son dual.

- **a.** Si $z_{x^t} \to +\infty$, arrêter. $(PLM')$ est non borné.
- **b.** Si $z_{x^t}$ est fini, poser la solution primale $y^t$, et la solution duale $u^t$.
- **c.** Si $(PL_{x^t})$ est irréalisable, poser $v^t$ le rayon dual avec $(v^t)'(b - Ax^t) < 0$.
- **d.** *Test d'optimalité :* Si $c'x^t + d'y^t \geq \eta^t$, arrêter. $(x^t, y^t)$ est une solution optimale pour $(PLM')$.
- **e.** *Violation.* Si $c'x^t + d'y^t < \eta^t$ ou $(PL_{x^t})$ est irréalisable, au moins une contrainte de $(PLM')$ est violée.
  - **i.** Si $z_{x^t}$ est fini, $\eta \leq c'x + (u^t)'(b - Ax)$ est violée. Poser $K^{t+1} = K^t \cup \{t\}$, ainsi
  $$S_R^{t+1} = S_R^t \cap \{(\eta,\ x) :\ \eta \leq c'x + (u^t)'(b - Ax)\}.$$
  - **ii.** Si $(PL_{x^t})$ est irréalisable, $(v^t)'(b - Ax) \geq 0$ est violée. Poser $J^{t+1} = J^t \cup (t)$, ainsi
  $$S_R^{t+1} = S_R^t \cap \{(\eta,\ x) :\ (v^t)'(b - Ax) \geq 0\}.$$
- **f.** $t \longleftarrow t + 1$.

Il y a plusieurs difficultés dans l'implémentation de l'algorithme de décomposition de Benders pour la résolution de la relaxation

$$(PLM^t) \qquad \begin{cases} z^t = \eta \longrightarrow \max, \\ \eta \leq c'x + (u^k)'(b - Ax), \quad k \in K^t, \\ (v^j)'(b - Ax) \geq 0, \quad j \in J^t, \\ x \in S \subseteq \mathbb{Z}_+^n,\ \eta \in \mathbb{R}. \end{cases}$$

où $K^t$ et $J^t$ sont les ensembles d'indices des inégalités valides après les $t$ premières itérations.

Une des difficultés est que $(PLM^t)$ est un programme en nombres mixtes. Une seconde difficulté est qu'il y a souvent dégénérescence du problème $(PL_{x^t})$, ainsi il n'y a pas de solution duale unique $u^t$. Une troisième difficulté est liée au choix des ensembles initiaux $K^t$ et $J^t$.

**Exemple 2.7.2.** Reprenons l'exemple 2.7.1

*Initialisation :* $K^1 = J^1 = \emptyset$. $t = 1$.

*Itération 1*

*Etape 1 :* $(PLM)^1 \qquad z^1 = \max\{(\eta, x) \in \mathbb{Z}_+^3 \times \mathbb{R},\ x_j \leq 5 \text{ } pour \text{ } j = \overline{1,3}\}$.

$\eta^1 \to +\infty,\ x^1 = (0,\ 0,\ 0)$ est réalisable.

*Etape 2 :* Séparation : Résoudre le programme linéaire

$$(PL_{x^1}) \qquad \begin{cases} z_{x^1} = 2y_1 - 3y_2 + 4y_3 \longrightarrow \max, \\ 2y_1 + 3y_2 + 6y_3 \leq -2, \\ 3y_1 - y_2 + 3y_3 \leq 10, \\ y \in \mathbb{R}_+^3. \end{cases}$$

$(PL_{x^1})$ est irréalisable puisque son dual est non borné. qui peut être vérifié par le point extrême dual $u^1 = (1,\ 0)$ et le rayon extrême $v^1 = (1,\ 0)$ (voir la figure 2.10).

$K^2 = K^1 \cup \{1\},\ J^2 = J^1 \cup \{1\}$.

*Itération 2*

*Etape 1 :*

$$(PLM^2) \qquad \begin{cases} z^2 = \eta \longrightarrow \max, \\ \eta \leq -2 + x_2 + 2x_3, \\ -2 - 5x_1 + 3x_2 - 7x_3 \geq 0, \\ x_j \leq 5, \quad j = \overline{1,3}, \\ x \subseteq \mathbb{Z}_+^3, \quad \eta \in \mathbb{R}. \end{cases}$$

Une solution optimale est $z^2 = 5, \quad x^2 = (0,\ 5,\ 1)$.

*Etape 2 :*

$$(Dual\ de\ (PL_{x^2})) \qquad \begin{cases} 6u_1 - 4u_2 \longrightarrow \min, \\ 2u_1 + 3u_2 \geq 2, \\ 3u_1 - u_2 \geq -3, \\ 6u_1 + 3u_2 \geq 4, \\ u \in \mathbb{R}_+^2. \end{cases}$$

Le dual est non borné, qui peut être vérifié par le point extrême $u^2 = (0,\ 3)$ et le rayon extrême $v^2 = (1,\ 3)$.

$K^3 = K^2 \cup \{2\},\ J^3 = J^2 \cup \{2\}.$

*Itération 3*

*Etape1 :*

$$(PLM^3) \quad \begin{cases} z^3 = \eta \longrightarrow \max, \\ \eta \leq -2 + 1x_2 + 2x_3, \\ \eta \leq 30 - 7x_1 - 8x_2 - 3x_3, \\ -2 - 5x_1 + 3x_2 - 7x_3 \geq 0, \\ 28 - 17x_1 - 3x_2 - 19x_3 \geq 0, \\ x_j \leq 5, \quad j = \overline{1,3}, \\ x \in \mathbb{Z}^3_+,\ \eta \in \mathbb{R}. \end{cases}$$

Une solution optimale est $z^3 = 3,\ x^3 = (0,\ 3,\ 1)$.

*Etape 2 :*

$$(PL_{x^3}) \quad \begin{cases} z_{x^3} = 2y_1 - 3y_2 + 4y_3 \longrightarrow \max, \\ 2y_1 + 3y_2 + 6y_3 \leq 0, \\ 3y_1 - y_2 + 3y_3 \leq 0, \\ y \in \mathbb{R}^3_+. \end{cases}$$

La solution optimale est $z_{x^3} = 0,\ y^3 = (0,\ 0,\ 0)$ et la solution duale optimale est $u^3 = (0,\ \frac{4}{3})$, $c'x^3 + z_{x^3} = 3 = \eta^3$. Par conséquent $(x^3,\ y^3) = (0,\ 3,\ 1,\ 0,\ 0,\ 0)$ est une solution optimale.

### Conclusion

Les méthodes de coupes ont permis, de résoudre des problèmes particuliers de type "partitionnement" ou "recouvrement" (Delorme 1974, par exemple). La motivation des méthodes de *coupes* ou de troncature est de tenter de trouver l'enveloppe convexe des solutions entières, c'est-à-dire le plus petit polyèdre contenant toutes les solutions entières du $(PLE)$. L'inconvénient de ces dernières est qu'elles fournissent souvent une solution réalisable uniquement à la fin, contrairement aux méthodes arborescentes qui livrent en cours de recherche une suite de bonnes solutions. Les méthodes de coupes sont complexes et affichent des performances modestes en pratique. Cependant, les coupes sont utiles comme techniques d'appoint dans les méthodes arborescentes. Plus récemment, l'étude polyédrale des problèmes d'optimisation combinatoire, introduite par J. Edmonds, a donné un regain d'intérêt pour les méthodes de coupes utilisées conjointement avec des procédures par séparation et évaluation. Il s'agit d'étudier les spécificités de chaque problème particulier et d'en déduire, sinon le polyèdre enveloppe convexe des solutions réalisables entières du problème, du moins un certain nombre de ses faces.

# Chapitre 3

# Méthode de support pour la résolution d'un problème de programmation linéaire en nombres entiers

## Introduction

On se propose de développer une nouvelle méthode pour la résolution d'un programme linéaire en nombres entiers à variables bornées. A chaque itération de cet algorithme on rajoute une nouvelle coupe et on résout le programme linéaire résultant par la méthode de support. Après avoir démontré que les coupes rajoutées sont valides, on a mis au point un algorithme pour sa construction, que l'on a illustré sur un exemple numérique.

## 3.1 Position du problème

Considérons le problème de programmation linéaire en nombres entiers et à variables bornées, s'écrivant sous la forme canonique suivante :

$$(PLE) \quad \left\{ \begin{array}{l} z = c'x \longrightarrow \max, \\ Ax \ = \ b, \\ d^- \leq x \leq d^+, \ x \text{ entier}, \end{array} \right. \tag{3.1}$$

où $A$ est une matrice rationnelle d'ordre $m \times n$, $rangA = m < n$ ; $b$ est un vecteur rationnel de dimension $m$ ; $c, \ x, \ d^- \ et \ d^+$ sont des vecteurs de dimension $n$ dans $\mathbb{Z}^n$.

On considère alors le problème relaxé associé de programmation linéaire suivant :

$$(PL) \quad \left\{ \begin{array}{l} z = c'x \longrightarrow \max, \\ Ax \ = \ b, \\ d^- \leq x \leq d^+. \end{array} \right. \tag{3.2}$$

Notons

$I = \{1,\ 2,\ ...,\ m\}$ : l'ensemble d'indices des lignes de $A$,

$J = \{1,\ 2,\ ...,\ n\}$ : l'ensemble d'indices des colonnes de $A$.

Donnons les définitions suivantes :

- Un vecteur $x$ vérifiant les contraintes $Ax = b$, $d^- \leq x \leq d^+$, $x$ entier, est une solution réalisable (ou plan) du problème (3.1). L'ensemble des solutions réalisables est alors donné par :

$$S = \{x \in \mathbb{Z}^n : \quad Ax = b, \quad d^- \leq x \leq d^+\}.$$

  De même, on définit par $X = \{x \in \mathbb{R}^n : \quad Ax = b, \quad d^- \leq x \leq d^+\}$, l'ensemble des solutions réalisables du problème (3.2).
- Un plan $x^0$ est dit optimal pour le problème (3.2) si $z(x^0) = \max\limits_{x \in X} c'x$.
- Un plan $x^\epsilon$ est appelé $\epsilon-$optimal ou suboptimal si :

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

  où $x^0$ est une solution optimale du problème (3.2), $x^\epsilon$ est une solution réalisable du problème (3.2) et $\epsilon$ un nombre supérieur ou égal à zéro choisi à l'avance.
- Soit un sous-ensemble d'indices $J_B \subset J$ tel que $J = J_B \cup J_N$, $J_B \cap J_N = \emptyset$, $|J_B| = m$. L'ensemble $J_B$ est alors appelé support si :

$$detA_B = detA(I, J_B) \neq 0.$$

- Le couple $\{x,\ J_B\}$ formé du plan $x$ et du support $J_B$ est appelé plan de support. Un plan de support $\{x,\ J_B\}$ est dit basique si

$$\forall j \in J_N, \quad x_j = d_j^- \vee d_j^+.$$

- Le plan de support $\{x,\ J_B\}$ est non dégénéré si :

$$d_j^- < x_j < d_j^+, \qquad \forall j \in J_B.$$

- En vertu de la partition de $J = J_B \cup J_N$, on peut alors écrire et fractionner les vecteurs et les matrices de la manière suivante : $x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$, $x_B = (x_j,\ j \in J_B)$, $x_N = (x_j,\ j \in J_N)$ ;

$c = \begin{pmatrix} c_B \\ c_N \end{pmatrix}$, $c_B = (c_j,\ j \in J_B)$, $c_N = (c_j,\ j \in J_N)$ ;

$A(I,\ J) = (a_{ij},\ i \in I,\ j \in J) = (a_j,\ j \in J),\ a_j = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{pmatrix}$ est la $j^{ème}$ colonne de la matrice $A$ ;

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

## 3.2 Méthode directe de support pour la résolution d'un problème de programmation linéaire à variables bornées

La méthode directe de support a été développée par R. Gabassov et F. M. kirillova dans les années 70 [53, 56]. Elle a la particularité de tenir compte des spécificités des problèmes tels qu'ils sont formulés lors de leur modélisation première. Les algorithmes qui sont déduits de cette méthode utilisent au maximum la structure particulière des problèmes pour plus d'efficacité.

Soit le problème (3.2) de programmation linéaire à variables bornées.

### 3.2.1 Formule d'accroissement de la fonction objectif

Soit $\{x, J_B\}$ un plan de support du problème (3.2). Considérons un autre plan quelconque $\bar{x} = x + \Delta x$. L'accroissement de la fonction objectif s'écrit alors :

$$\Delta z = z(\bar{x}) - z(x) = c'\bar{x} - c'x = c'\Delta x. \tag{3.3}$$

Par ailleurs on a : $A\bar{x} = Ax = b \Longrightarrow A(\bar{x} - x) = 0 \Longrightarrow A\Delta x = 0$.
L'inégalité $A\Delta x = 0$ peut alors s'écrire : $A\Delta x = A_B\Delta x_B + A_N\Delta x_N = 0$. D'où

$$\Delta x_B = -A_B^{-1}A_N\Delta x_N. \tag{3.4}$$

Ainsi, l'accroissement (3.3) devient :

$$\Delta z = c'\Delta x = -c'_BA_B^{-1}A_N\Delta x_N + c'_N\Delta x_N = -(c'_BA'_BA_N - c'_N)\Delta x_N. \tag{3.5}$$

On définit le vecteur des potentiels $u$ et le vecteur des estimations $E$ comme suit :

$$u = c'_B A_B^{-1},$$

$$E' = u'A - c' \Leftrightarrow E_j = u'a_j - c_j, \quad j \in J,$$

où $E' = (E'_B,\ E'_N)$, avec

$$E'_B = u'A_B - c'_B = c'_B A_B^{-1} A_B - c'_B = 0,$$

$$E'_N = u'A_N - c'_N = c'_B A_B^{-1} A_N - c'_N.$$

Finalement, la formule d'accroissement (3.5) prend la forme finale suivante :

$$\Delta z = -\sum_{j \in J_N} E_j(\bar{x}_j - x_j). \tag{3.6}$$

**Critère d'optimalité**

On a le théorème d'optimalité suivant :

**Théorème 3.2.1.** [54] *(Critère d'optimalité)*
*Soit* $\{x,\ J_B\}$ *un plan de support du problème* (3.2)*. Alors les relations :*

$$\begin{cases} E_j \geq 0, \quad \textit{pour} \quad x_j = d_j^-, \\ E_j \leq 0, \quad \textit{pour} \quad x_j = d_j^+, \\ E_j = 0, \quad \textit{pour} \quad d_j^- < x_j < d_j^+, \qquad j \in J_N, \end{cases} \tag{3.7}$$

*sont suffisantes pour l'optimalité du plan de support* $\{x,\ J_B\}$*. Ces même relations sont aussi nécessaires, si le plan de support est non-dégénéré.*

**Remarque 3.2.1.** Si le plan de support $\{x,\ J_B\}$ est basique, alors les relations d'optimalité sont :

$$\begin{cases} E_j \geq 0, \quad \text{pour} \quad x_j = d_j^-, \\ E_j \leq 0, \quad \text{pour} \quad x_j = d_j^+, \qquad j \in J_N, \end{cases}$$

**Preuve.** *Condition suffisante*
Soit $\{x,\ J_B\}$ un plan de support vérifiant les relations (3.7). Pour tout plan de support $\bar{x}$ du problème (3.2), la formule d'accroissement (3.6) donne :

$$z(\bar{x}) - z(x) = \Delta z = -\sum_{E_j > 0,\ j \in J_N} E_j(\bar{x}_j - d_j^-) - \sum_{E_j < 0,\ j \in J_N} E_j(\bar{x}_j - d_j^+).$$

Comme

$$d_j^- \leq \bar{x}_j \leq d_j^+ \Longrightarrow \bar{x}_j - d_j^- \geq 0 \; et \; \bar{x}_j - d_j^+ \leq 0,$$

on déduit alors :

$$z(\bar{x}) \leq z(x),$$

pour tout $\bar{x}$ solution réalisable.

Par conséquent, le vecteur $x$ est une solution optimale du problème (3.2).

*Condition Nécessaire*

Soit $\{x, \; J_B\}$ un plan de support optimal non dégénéré du problème (3.2), et supposons que les relations (3.7) ne sont pas vérifiées, c'est-à-dire qu'il existe au moins un indice $j_0 \in J_N$ tel que :

$$E_{j_0} > 0 \; et \; x_{j_0} > d_{j_0}^-, \quad ou \; bien \quad E_{j_0} < 0 \; et \; x_{j_0} < d_{j_0}^+.$$

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

$$\begin{cases} l_{j_0} = -signE_{j_0}, \\ \\ l_j = 0 \quad j \neq j_0, \qquad j \in J_N, \\ \\ l_B = l(J_B) = -A_B^{-1} A_N l_N = A_B^{-1} a_{j_0} signE_{j_0}. \end{cases} \tag{3.8}$$

Le vecteur $\bar{x}$ vérifie la contrainte principale $A\bar{x} = b$. Pour que $\bar{x}$ soit un plan du problème (3.2), il doit en plus vérifier l'inégalité :

$$d^- \leq \bar{x} \leq d^+ \Longleftrightarrow d^- \leq x + \theta l \leq d^+ \Longleftrightarrow d^- - x \leq \theta l \leq d^+ - x,$$

c'est-à-dire

$$\begin{cases} d_j^- - x_j \leq \theta l_j \leq d_j^+ - x_j, \quad j \in J_B, \\ \\ d_{j_0}^- - x_{j_0} \leq -\theta signE_{j_0} \leq d_{j_0}^+ - x_{j_0}. \end{cases} \tag{3.9}$$

Puisque le plan de support $\{x, \; J_B\}$ est non dégénéré, on a alors :

$$d_j^- - x_j < 0 < d_j^+ - x_j, \quad j \in J_B,$$

et

$$\begin{cases} d_{j_0}^- - x_{j_0} < 0 \quad si \quad E_{j_0} > 0, \\ \\ d_{j_0}^+ - x_{j_0} > 0 \quad si \quad E_{j_0} < 0. \end{cases}$$

Pour un nombre $\theta > 0$ assez petit, les relations (3.9) seront alors vérifiées et le vecteur $\bar{x}$ sera un plan du problème (3.2).

La formule d'accroissement (3.6) nous donne alors :

$$\Delta z = -\sum_{j \in J_N} E_j(\bar{x}_j - x_j) = -\theta \sum_{j \in J_N} E_j l_j = \theta E_{j_0} sign E_{j_0} = \theta |E_{j_0}| > 0. \tag{3.10}$$

On déduit que $z(\bar{x}) > z(x)$, mais ceci est en contradiction avec le fait que $x$ est un plan optimal du problème (3.2).

Par conséquent, si $\{x,\ J_B\}$ est un plan optimal du problème (3.2) non dégénéré, alors les relations (3.7) sont forcèment vérifiées. □

**Estimation de suboptimalité**

Pour estimer l'écart qui existe entre la valeur optimale $z(x^0)$ et une autre valeur $z(x)$ d'un plan de supprt quelconque $\{x,\ J_B\}$, on remplace dans la formule d'accroissement (3.6) le vecteur $\bar{x}$ par $x^0$ et on aura :

$$\begin{aligned} z(x^0) - z(x) &= -\sum_{j \in J_N} E_j(x_j^0 - x_j) \\ &= \sum_{E_j > 0,\ j \in J_N} E_j(x_j - x_j^0) + \sum_{E_j < 0,\ j \in J_N} E_j(x_j - x_j^0). \end{aligned} \tag{3.11}$$

Puisque le plan optimal $x^0$ vérifie $d_j^- \leq x_j^0 \leq d_j^+,\ \ j \in J$, alors il en résulte que :

$$x_j - d_j^+ \leq x_j - x_j^0 \leq x_j - d_j^-,\ \ j \in J.$$

Donc

$$\begin{cases} E_j(x_j - x_j^0) \leq E_j(x_j - d_j^-), & \text{si } E_j > 0, \\ E_j(x_j - x_j^0) \leq E_j(x_j - d_j^+), & \text{si } E_j < 0. \end{cases}$$

Ainsi, on obtient une majoration du nombre inconnu de gauche de l'égalité (3.11) :

$$z(x^0) - z(x) \leq \sum_{E_j > 0,\ j \in J_N} E_j(x_j - d_j^-) + \sum_{E_j < 0,\ j \in J_N} E_j(x_j - d_j^+). \tag{3.12}$$

Le nombre

$$\beta(x,\ J_B) = \sum_{E_j > 0,\ j \in J_N} E_j(x_j - d_j^-) + \sum_{E_j < 0,\ j \in J_N} E_j(x_j - d_j^+), \tag{3.13}$$

est ainsi appelé estimation de suboptimalité du plan de support $\{x, J_B\}$.

**Théorème 3.2.2.** *Soit* $\{x^\epsilon,\ J_B\}$ *un plan de support du problème* (3.2) *et* $\epsilon$ *un nombre positif ou nul arbitraire* $\epsilon \geq 0$. *Alors*

$$x^\epsilon \ \ est\ \epsilon - optimal \Longleftrightarrow \beta(x^\epsilon,\ J_B) \leq \epsilon.$$

**Preuve.** voir [54].

### 3.2.2 Algorithme de résolution

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

Une itération de l'algorithme consiste à faire le passage de $\{x,\ J_B\}$ vers $\{\bar{x},\ \bar{J}_B\}$, tel que $z(\bar{x}) \geq z(x)$.

Si le critère d'optimalité ou de suboptimalité n'est pas vérifié, c'est-à-dire $\beta(x,\ J_B) > \epsilon$, il faut alors améliorer le plan $x$.

#### Changement de plan

On construit alors un nouveau plan $\bar{x}$ de la manière suivante :

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

où $l$ est un $n-$vecteur appelé direction d'amélioration ; $\theta$ est le pas le long de cette direction.

Dans cet algorithme, on choisira la métrique du simplexe et on ne fera varier qu'une seule composante $x_j$ parmi toutes celles qui ne vérifient pas le critère d'optimalité (3.7).

Pour que l'accroissement (3.10) soit maximal, il faut alors prendre $\theta$ aussi grand que possible et choisir l'indice $j_0$ tel que :

$$|E_{j_0}| = \max_{j \in J_{NNO}} |E_j|,$$

où $J_{NNO}$ représente le sous-ensemble de $J_N$ des indices non optimaux.

D'autre part, la direction $l$ est calculée suivant les relations (3.8) et le pas $\theta$ doit vérifier les relations (3.9). Calculons les différentes valeurs maximales que peut prendre le pas $\theta$ dans (3.9) :

Pour les indices $j \in J_B$, on a la formule suivante :

$$\theta_j = \begin{cases} \frac{d_j^+ - x_j}{l_j}, & l_j > 0, \\ \frac{d_j^- - x_j}{l_j}, & l_j < 0, \\ \infty, & l_j = 0. \end{cases}$$

Le pas maximal $\theta$ pour que les relations (3.9) soient vérifiées, pour $j \in J_B$, se calcule comme suit :

$$\theta = \theta_{j_1} = \min_{j \in J_B} \theta_j.$$

Pour l'indice $j_0$, le pas maximal $\theta$ doit être égal à :

$$\theta_{j_0} = \begin{cases} d_{j_0}^+ - x_{j_0}, & l_{j_0} = 1, \\ x_{j_0} - d_{j_0}^-, & l_{j_0} = -1. \end{cases}$$

Le pas maximal le long de la direction $l$ pour que $x$ soit un plan est égal à :

$$\theta^0 = \min\{\theta_{j_1},\ \theta_{j_0}\}, \quad j_1 \in J_B,\ j_0 \in J_N. \tag{3.14}$$

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

Calculons l'estimation de suboptimalité pour le plan $\{\bar{x},\ J_B\}$ :

$$\beta(\bar{x},\ J_B) = \sum_{E_j > 0,\ j \in J_N} E_j(\bar{x}_j - d_j^-) + \sum_{E_j < 0,\ j \in J_N} E_j(\bar{x}_j - d_j^+).$$

On a pour $j \in J_N$ :

$$\bar{x}_j = x_j, \quad si\ j \neq j_0, \qquad \bar{x}_{j_0} = \begin{cases} x_{j_0} - \theta^0, & \text{si} \quad E_{j_0} > 0, \\ x_{j_0} + \theta^0, & \text{si} \quad E_{j_0} < 0. \end{cases}$$

En remplaçant, on obtient :

$$\beta(\bar{x},\ J_B) = \beta(x,\ J_B) - \theta^0 |E_{j_0}|. \tag{3.15}$$

**Changement de support**

Si $\beta(\bar{x},\ J_B) \leq \epsilon$, alors le plan $\bar{x}$ est $\epsilon$-optimal et on peut arrêter l'algorithme, sinon on remplace $J_B$ par un nouveau support $\bar{J}_B$ de la manière suivante :

1. Si $\theta^0 = \theta_{j_0}$, alors il est inutile de changer de support, on écrira donc : $\bar{x} = x + \theta^0 l, \quad \bar{J}_B = J_B$.
2. Si $\theta^0 = \theta_{j_1}$, le nouveau plan de support $(\bar{x},\ \bar{J}_B)$ s'écrit :

   $$\bar{x} = x + \theta^0 l, \quad \bar{J}_B = (J_B \backslash j_1) \cup j_0.$$

   Si $\beta(\bar{x},\ \bar{J}_B)$ est supérieur à $\epsilon$, on recommence alors une nouvelle itération avec $\{\bar{x},\ \bar{J}_B\}$.

## 3.3 Algorithme de résolution d'un (PLE) par la méthode de support

Le but de cet algorithme est de construire un plan optimal pour le problème (3.1). Le principe est le même que celui des algorithmes de plans sécants. A savoir, on commence par résoudre le problème relaxé (3.2) à l'aide de l'algorithme direct de support. Si la solution optimale du $(PL)$ contient uniquement des composantes entières, elle est également une solution optimale du $(PLE)$ ; dans ce cas, la résolution est terminée. Sinon, si une ou plusieurs variables ne sont pas entières, on doit générer alors une coupe valide. Cette contrainte supplémentaire est ajoutée au programme $(PL)$ et on résout de nouveau le problème résultant. Si la solution obtenue est à valeurs entières, la résolution est terminée. Sinon, on génère une nouvelle coupe et on répète la procédure jusqu'à l'obtention d'une solution optimale à valeurs entières.

Soit $\{x, J_B\}$ un plan de support du problème (3.2).
Suivant les valeurs des composantes du vecteur des estimations $E_N$, on définit les ensembles d'indices suivants :

$$\begin{aligned}
J_N^+ &= \{j \in J_N : \ E_j > 0\}, \quad & J_N^- &= \{j \in J_N : \ E_j < 0\}, \\
J_{N0} &= \{j \in J_N : \ E_j = 0\}, \quad & J_{N0} &= J^+ \cup J^-, \\
\textit{où } J^+ &= \{j \in J_{N0} : \ x_j - d_j^- \leq d_j^+ - x_j\}, \ \ \textit{et} \ \ & J^- &= \{j \in J_{N0} : \ x_j - d_j^- > d_j^+ - x_j\}.
\end{aligned}$$

En fonction de $J_B$, on construit les fonctions pour tout $i \in J_B$ :

$$\begin{aligned}
Z_i(x) = -\left\{ \tilde{b}_i - \sum_{j \in J_N^- \cup J^-} x_{ij} d_j^+ - \sum_{j \in J_N^+ \cup J^+} x_{ij} d_j^- \right\} + \sum_{j \in J_N^+ \cup J^+} \{x_{ij}\}(x_j - d_j^-) \\
+ \sum_{j \in J_N^- \cup J^-} \{-x_{ij}\}(-x_j + d_j^+), \qquad (3.16)
\end{aligned}$$

où $\tilde{b} = (A(I, J_B))^{-1} b = A_B^{-1} b$ *et* $X = (x_{ij}, \ i \in J_B, \ j \in J_N) = (A(I, J_B))^{-1} A_N = A_B^{-1} A_N$.

**Proposition 3.3.1.** [54] *$Z_i(x^e)$ est un nombre entier, quelque soit $\{x^e, \ J_B\}$ plan de support du problème (3.1) et de plus, on a : $Z_i(x^e) \geq 0, \quad \forall i \in J_B$.*

**Preuve.** Soit $x^e$ un plan du problème (3.1), c'est à dire :

$$Ax^e = b,\ d^- \leq x^e \leq d^+ \ \ et\ x_j^e\ entier\ \forall j \in J.$$

On a :
$$Ax^e = b \Rightarrow A_B x_B^e + A_N x_N^e = b,$$

$$\begin{aligned}
\Rightarrow x_B^e &= A_B^{-1}(b - A_N x_N^e),\\
\Rightarrow x_B^e &= A_B^{-1}b - A_B^{-1}A_N x_N^e \ = \ \tilde{b} - X x_N^e,\\
\Rightarrow x_i^e &= \tilde{b}_i - \sum_{j\in J_N} x_{ij}x_j^e, \quad \forall i \in J_B,\\
\Rightarrow \tilde{b}_i &= x_i^e + \sum_{j\in J_N} x_{ij}x_j^e, \quad \forall i \in J_B. \qquad (3.17)
\end{aligned}$$

En remplaçant $\tilde{b}$ dans la formule (3.16) et en utilisant les propriétes 1.8.4 du chapitre $I$, on aura pour $i \in J_B$ :

$$\begin{aligned}
Z_i(x^e) &= -\left\{\tilde{b}_i - \sum_{j\in J_N^-\cup J^-} x_{ij}d_j^+ - \sum_{j\in J_N^+\cup J^+} x_{ij}d_j^-\right\} + \sum_{j\in J_N^+\cup J^+}\{x_{ij}\}(x_j^e - d_j^-) +\\
&\quad \sum_{j\in J_N^-\cup J^-}\{-x_{ij}\}(-x_j^e + d_j^+)\\
&= -\left\{x_i^e + \sum_{j\in J_N} x_{ij}x_j^e - \sum_{j\in J_N^-\cup J^-} x_{ij}d_j^+ - \sum_{j\in J_N^+\cup J^+} x_{ij}d_j^-\right\} + \sum_{j\in J_N^+\cup J^+}\{x_{ij}\}(x_j^e -\\
&\quad d_j^-) + \sum_{\substack{j\,\in\, J_N^-\,\cup\, J^-\\ \{x_{ij}\}\neq 0}} (1 - \{x_{ij}\})(-x_j^e + d_j^+),
\end{aligned}$$

*car* $\{-a\} = 1 - \{a\},\ \forall a \in \mathbb{R}\backslash\mathbb{Z}.$ *Comme* $\{a+k\} = \{a\},\ \forall a \in \mathbb{R}$ *et* $\forall k \in \mathbb{Z},$ *on aura*

$$\begin{aligned}
Z_i(x^e) &= -\left\{\sum_{j\in J_N^+\cup J^+} x_{ij}(x_j^e - d_j^-) + \sum_{j\in J_N^-\cup J^-} x_{ij}(x_j^e - d_j^+)\right\} + \sum_{j\in J_N^+\cup J^+}\{x_{ij}\}(x_j^e\\
&\quad -d_j^-) + \sum_{\substack{j\,\in\, J_N^-\,\cup\, J^-\\ \{x_{ij}\}\neq 0}} (d_j^+ - x_j^e) + \sum_{j\in J_N^-\cup J^-}\{x_{ij}\}(x_j^e - d_j^+). \qquad (3.18)
\end{aligned}$$

On sait que :

$$\begin{aligned}
\{\sum_{j\in J_N^+\cup J^+} x_{ij}(x_j^e - d_j^-) + \sum_{j\in J_N^-\cup J^-} x_{ij}(x_j^e - d_j^+)\} &\leq \{\sum_{j\in J_N^+\cup J^+} x_{ij}(x_j^e - d_j^-)\} + \{\sum_{j\in J_N^-\cup J^-} x_{ij}(x_j^e - d_j^+)\},\\
&\leq \sum_{j\in J_N^+\cup J^+}\{x_{ij}(x_j^e - d_j^-)\} + \sum_{j\in J_N^-\cup J^-}\{x_{ij}(x_j^e - d_j^+)\},\\
&\leq \sum_{j\in J_N^+\cup J^+}\{x_{ij}\}(x_j^e - d_j^-) + \sum_{j\in J_N^-\cup J^-}\{x_{ij}\}(x_j^e - d_j^+).
\end{aligned}$$

Comme $\sum\limits_{\substack{j\in J_N^-\cup J^-\\ \{x_{ij}\}\neq 0}}(d_j^+-x_j^e)\geq 0$, alors on a bien : $Z_i(x^e)\geq 0$, $\forall i\in J_B$ et $x^e$ solution réalisable entière de (3.1).

D'autre part, d'après les propriétés 1.8.4 du chapitre $I$, on peut écrire :

$$
\begin{aligned}
&\{\sum_{j\in J_N^+\cup J^+}x_{ij}(x_j^e-d_j^-)+\sum_{j\in J_N^-\cup J^-}x_{ij}(x_j^e-d_j^+)\}\\
&= \{\sum_{j\in J_N^+\cup J^+}x_{ij}(x_j^e-d_j^-)\}+\{\sum_{j\in J_N^-\cup J^-}x_{ij}(x_j^e-d_j^+)\}-[\{\sum_{j\in J_N^+\cup J^+}x_{ij}(x_j^e-d_j^-)\}+\{\sum_{j\in J_N^-\cup J^-}x_{ij}(x_j^e-d_j^+)\}],\\
&= \sum_{j\in J_N^+\cup J^+}\{x_{ij}(x_j^e-d_j^-)\}+\sum_{j\in J_N^-\cup J^-}\{x_{ij}(x_j^e-d_j^+)\}-[\sum_{j\in J_N^+\cup J^+}\{x_{ij}(x_j^e-d_j^-)\}]-[\sum_{j\in J_N^-\cup J^-}\{x_{ij}(x_j^e-d_j^+)\}]\\
&\quad -[\{\sum_{j\in J_N^+\cup J^+}x_{ij}(x_j^e-d_j^-\}+\{\sum_{j\in J_N^-\cup J^-}x_{ij}(x_j^e-d_j^+)\}],\\
&= \sum_{j\in J_N^+\cup J^+}\{x_{ij}\}(x_j^e-d_j^-)+\sum_{j\in J_N^-\cup J^-}\{x_{ij}\}(x_j^e-d_j^+)-\sum_{J_N^+\cup J^+}[\{x_{ij}\}(x_j^e-d_j^-)]-\sum_{J_N^-\cup J^-}[\{x_{ij}\}(x_j^e-d_j^+)]\\
&\quad -[\sum_{j\in J_N^+\cup J^+}\{x_{ij}(x_j^e-d_j^-)\}]-[\sum_{j\in J_N^-\cup J^-}\{x_{ij}(x_j^e-d_j^+)\}]-[\{\sum_{j\in J_N^+\cup J^+}x_{ij}(x_j^e-d_j^-\}+\{\sum_{j\in J_N^-\cup J^-}x_{ij}(x_j^e-d_j^+)\}].
\end{aligned}
$$

En remplaçant dans la formule (3.18), on obtient :

$$
\begin{aligned}
Z_i(x^e) \;=\; &\sum_{\substack{j\in J_N^-\cup J^-\\ \{x_{ij}\}\neq 0}}(d_j^+-x_j^e)+\sum_{J_N^+\cup J^+}[\{x_{ij}\}(x_j^e-d_j^-)]+\sum_{J_N^-\cup J^-}[\{x_{ij}\}(x_j^e-d_j^+)]-[\sum_{j\in J_N^+\cup J^+}\{x_{ij}(x_j^e-d_j^-)\}]\\
&-[\sum_{j\in J_N^-\cup J^-}\{x_{ij}(x_j^e-d_j^+)\}]-[\{\sum_{j\in J_N^+\cup J^+}x_{ij}(x_j^e-d_j^-\}+\{\sum_{j\in J_N^-\cup J^-}x_{ij}(x_j^e-d_j^+)\}]\;\in\mathbb{Z}_+.\quad\square
\end{aligned}
$$

**Proposition 3.3.2.** [54] *Soit* $\{x^0,\ J_B^0\}$ *un plan de support optimal de* (3.2). *S'il existe un indice* $i_1\in J_B^0$ *tel que* $Z_{i_1}(x^0)<0$, *alors l'inégalité*

$$Z_{i_1}(x)\geq 0,$$

*donne une coupe valide.*

**Preuve.** Evident, car on a démontré que pour une solution réalisable $x^e$ du problème (3.1), on a $Z_i(x^e)\geq 0,\ \forall i\in J_B.$ $\square$

**Proposition 3.3.3.** *Soit* $\{x^0,\ J_B^0\}$ *un plan de support optimal de* (3.2). *Si* $x^0$ *est un plan non entier tel que* $\forall i\in J_B^0$, *on a* $Z_i(x^0)\geq 0$, *alors,* $x^0$ *est forcément un plan non basique.*

**Preuve.** Si $\{x^0,\ J_B^0\}$ est un plan de support optimal basique de (3.2), alors dans ce cas on aura :

$$x_j^0=\begin{cases} d_j^+, & j\in J_N^-\cup J^-,\\ d_j^-, & j\in J_N^+\cup J^+.\end{cases}$$

Soit un indice $i_1$ tel que la composante $x^0_{i_1}$ est non entière, donc, $0 < \{x^0_{i_1}\} < 1$. L'expression de $Z_{i_1}(x^0)$, prend alors la forme suivante :

$$\begin{aligned} Z_{i_1}(x^0) &= -\left\{\tilde{b}_{i_1} - \sum_{j\in J_N^-\cup J^-} x_{i_1j}d_j^+ - \sum_{j\in J_N^+\cup J^+} x_{i_1j}d_j^-\right\} + \sum_{j\in J_N^+\cup J^+}\{x_{i_1j}\}(d_j^- - d_j^-) \\ &\quad + \sum_{j\in J_N^-\cup J^-}\{-x_{i_1j}\}(-d_j^+ + d_j^+), \\ &= -\left\{\tilde{b}_{i_1} - \sum_{j\in J_N^-\cup J^-} x_{i_1j}d_j^+ - \sum_{j\in J_N^+\cup J^+} x_{i_1j}d_j^-\right\}. \end{aligned}$$

D'autre part, on a :

$$\begin{aligned} Ax^0 &= b, \\ \Rightarrow A_Bx_B^0 + A_Nx_N^0 &= b, \\ \Rightarrow x_B^0 &= A_B^{-1}b - A_B^{-1}A_Nx_N^0, \\ &= \tilde{b} - Xx_N^0, \\ \Rightarrow \tilde{b} &= x_B^0 + Xx_N^0, \\ \Rightarrow \tilde{b}_i &= x_i^0 + \sum_{j\in J_N^-\cup J^-} x_{ij}d_j^+ + \sum_{j\in J_N^+\cup J^+} x_{ij}d_j^-, \quad i\in J_B^0. \end{aligned}$$

D'où

$$Z_{i_1}(x^0) = -\{x^0_{i_1}\} < 0, \ \textit{d'où la contradiction.} \quad \square$$

**Proposition 3.3.4.** *Quand le plan $x^0$ est basique, alors on a : $Z_i(x^0) \leq 0, \ \forall\, i \in J_B$.*

### Schéma de l'algorithme

Le schéma général de cet algorithme est décrit comme suit :

**Etape 1** Poser $t = 0$, $I_0 = I = \{1,\ 2,\ ...,\ m\}$ et $J^0 = J = \{1,\ 2,\ ...,\ n\}$.

Soit $\{x^*, J_B^*\}$ un plan de support initial du problème (3.2). On construit alors un plan de support optimal $\{x^0,\ J_B^0\}$ par la méthode directe de support.

Si $x^0 \in \mathbb{Z}^n$, arrêter l'algorithme : $x^0$ est un plan optimal de (3.1). Sinon, aller à l'étape 2.

**Etape 2** On construit les fonctions $Z_i(x^0)$, pour tout $i \in J_B^0$.

**Etape 3** Deux cas peuvent se présenter :

**(i)** $\exists i_1 \in J_B^0 : \quad Z_{i_1}(x^0) < 0,$

Dans ce cas $(i)$ l'inégalité

$$Z_{i_1}(x) \geq 0, \tag{3.19}$$

donne une coupe valide.

On incrémente $t$, $t = t + 1$.

Aux contraintes de (3.2), on ajoute la contrainte (3.19) et on résoud le problème :

$$\begin{cases} z = c'x \longrightarrow \max, \\ Ax \ = \ b, \\ Z_{i_1}(x) \geq 0, \\ d^- \leq x \leq d^+. \end{cases} \tag{3.20}$$

On commence par le plan de support $\{\bar{x}, \bar{J}_B\}$, avec

$$\bar{x} = x^* + \lambda(x^0 - x^*), \quad 0 < \lambda < 1,$$

où $\lambda$ sera trouvé à partir de l'équation $Z_{i_1}(\bar{x}) = 0$.

Comme on a ajouté une inégalité au système (3.20), on rajoute une variable d'écart $x$ d'indice $n + t$ $(x_{n+t})$. On prend alors le support $\bar{J}_B = J_B^0 \cup \{n + t\}$.

On pose $I_t = I_{t-1} \cup \{m + t\}$ et $J^t = J^{t-1} \cup \{n + t\}$.

Soit $\{x^1,\ J_B^1\}$ le plan de support optimal de (3.20). On refait alors le processus comme avec $\{x^0,\ J_B^0\}$.

**(ii)** $Z_i(x^0) \geq 0,\ \forall i \in J_B^0$.

Dans ce cas, on distingue deux situations possibles :

– $x^0$ est non entier et basique, cette situation ne peut avoir lieu d'après la proposition 3.3.3. C'est-à-dire, dans le cas de solutions basiques, le deuxième cas ne se présente pas.

– $x^0$ est non entier et non basique (la solution optimale $x^0$ n'est pas unique). Dans ce cas, il faut construire une solution $x^{0B}$ basique, et construire par cette dernière une coupe régulière comme dans le cas $(i)$.

Montrons que $\{\bar{x},\ \bar{J}_B\}$ est bien un plan de support pour le problème (3.20).

**Preuve.** On a :

$$\begin{aligned} A\bar{x} &= A(x^* + \lambda(x^0 - x^*)), \\ &= Ax^* + \lambda A(x^0 - x^*), \\ &= b + \lambda A\Delta x, \\ &= b. \end{aligned}$$

$A\Delta x = 0$, comme $x^0$ et $x^*$ sont deux plans de (3.2).
Le vecteur $\bar{x}$ vérifie la contrainte principale $A\bar{x} = b$. Il vérifie aussi la contrainte $Z_{i_1}(\bar{x}) \geq 0$, puisque par construction, le pas $\lambda$ est tel que $Z_{i_1}(\bar{x}) = 0$.
Pour que $\bar{x}$ soit un plan du problème (3.20), il doit en plus appartenir à $K = \{x \in \mathbb{R}^n :\ d^- \leq x \leq d^+\}$. En effet, comme $K$ est convexe et $x^*$ et $x^0 \in K$, alors il s'ensuit que

$$\bar{x} = \lambda x^0 + (1-\lambda)x^* \in K.$$

D'où $\bar{x}$ est un plan du problème (3.20).
$\bar{J}_B$ est bien un support, puisque :

$$detA(I_t, \bar{J}_B) = detA(I_{t-1} \cup \{m+t\}, J_B^0 \cup \{n+t\}) = detA(I_{t-1}, J_B^0) \neq 0. \quad \square$$

**Calcul de $\lambda$ à partir de l'équation $Z_{i_1}(\bar{x}) = 0$ :**

$$\begin{aligned} -\{\tilde{b}_{i_1} - \sum_{j\in J_N^-\cup J^-} x_{i_1j}d_j^+ - \sum_{j\in J_N^+\cup J^+} x_{i_1j}d_j^-\} + \sum_{j\in J_N^+\cup J^+} \{x_{i_1j}\}(x_j^* + \lambda(x_j^0 - x_j^*) - d_j^-) & \\ + \sum_{j\in J_N^-\cup J^-} \{-x_{i_1j}\}(-x_j^* - \lambda(x_j^0 - x_j^*) + d_j^+) &= 0, \\ Z_{i_1}(x^*) + \lambda \sum_{j\in J_N^+\cup J^+} \{x_{i_1j}\}(x_j^0 - x_j^*) - \lambda \sum_{j\in J_N^-\cup J^-} \{-x_{i_1j}\}(x_j^0 - x_j^*) &= 0, \\ Z_{i_1}(x^*) + \lambda( \sum_{j\in J_N^+\cup J^+} \{x_{i_1j}\}(x_j^0 - x_j^*) + \sum_{j\in J_N^-\cup J^-} \{x_{i_1j}\}(x_j^0 - x_j^*) - \sum_{\substack{j\,\in\, J_N^-\cup J^- \\ \{x_{i_1j}\}\neq 0}} (x_j^0 - x_j^*)) &= 0, \\ Z_{i_1}(x^*) + \lambda( \sum_{j\in J_N} \{x_{i_1j}\}(x_j^0 - x_j^*) - \sum_{\substack{j\,\in\, J_N^-\cup J^- \\ \{x_{i_1j}\}\neq 0}} (x_j^0 - x_j^*)) &= 0, \end{aligned}$$

d'où

$$\lambda = \frac{Z_{i_1}(x^*)}{\sum\limits_{\substack{j\,\in\, J_N^-\cup J^- \\ \{x_{i_1j}\}\neq 0}} (x_j^0 - x_j^*) - \sum\limits_{j\in J_N} \{x_{i_1j}\}(x_j^0 - x_j^*)}.$$

**Exemple 3.3.1.** Soit à résoudre le problème de programmation linéaire en nombres entiers suivant par la méthode de support :

$$\begin{cases} z = & 3x_1 + 5x_2 + 4x_3 + 2x_4 \longrightarrow \max, \\ & 2x_1 + 7x_2 + 3x_3 + \quad +x_5 = 18, \\ & x_1 + 3x_2 + 2x_3 + x_4 + x_6 = 14, \\ & 3x_1 + 2x_2 + \quad +2x_4 + x_7 = 11, \\ & 0 \leq x_i \leq 10, \; x_i \; entier, \quad i = \overline{1,7}. \end{cases} \tag{3.21}$$

Le programme relaxé associé est

$$\begin{cases} z = & 3x_1 + 5x_2 + 4x_3 + 2x_4 \longrightarrow \max, \\ & 2x_1 + 7x_2 + 3x_3 + \quad +x_5 = 18, \\ & x_1 + 3x_2 + 2x_3 + x_4 + x_6 = 14, \\ & 3x_1 + 2x_2 + \quad +2x_4 + x_7 = 11, \\ & 0 \leq x_i \leq 10, \quad i = \overline{1,7}. \end{cases} \tag{3.22}$$

**Etape 1 :** Résolution par la méthode directe de support du problème relaxé (3.22).
Soit $x^* = (2,1,2,0,1,5,3)'$ un plan initial de (3.22), $z(x^*) = 20$.
On prend le support $J_B^* = \{1,2,3\}$.
On construit alors le plan de support optimal :

$$x^0 = \left(\frac{21}{11}, 0, \frac{52}{11}, \frac{29}{11}, 0, 0, 0\right)', \quad J_B^0 = \{1,4,3\}, \quad z(x^0) = \frac{329}{11}.$$

**Etape 2 :** Le plan $x^0$ est non entier, on rajoute alors au programme (3.22) une coupe.

**Itération 1** Le vecteur des estimations vaut

$$E_N^0 = (E_6^0, E_5^0, E_2^0, E_7^0)' = \left(\frac{16}{11}, \frac{4}{11}, \frac{27}{11}, \frac{3}{11}\right)', \quad J_N^0 = \{6,5,2,7\}.$$

Selon les valeurs des $E_j^0, \; j \in J_N^0$, on partitionne l'ensemble $J_N^0$ comme suit :
$J_N^+ = \{j \in J_N^0 : \; E_j^0 > 0\} = \{6,5,2,7\}$,
$J_N^- = \; J^+ = \; J^- = \emptyset$.
On calcule alors la valeur des fonctions $Z_i(x^0), \quad i \in J_B^0$ :

On trouve

$$Z_B(x^0) = \left(-\frac{10}{11}, -\frac{7}{11}, -\frac{8}{11}\right)'.$$

On se trouve donc dans le cas 1. On rajoute alors aux contraintes de (3.22) la contrainte $Z_{i_1}(x) \geq 0$.

L'indice $i_1$ est choisi comme suit :

$$Z_{i_1}(x^0) = \min_{i \in J_B^0}(Z_i(x^0)).$$

On prend donc : $i_1 = 1$. La contrainte $Z_1(x) \geq 0$ s'écrit ainsi :

$$Z_1(x) = -\frac{10}{11} + \frac{5}{11}x_6 + \frac{4}{11}x_5 + \frac{5}{11}x_2 + \frac{3}{11}x_7 \geq 0.$$

Le programme résultant est :

$$\begin{cases} z = 3x_1 + 5x_2 + 4x_3 + 2x_4 \longrightarrow \max, \\ 2x_1 + 7x_2 + 3x_3 + \quad + x_5 = 18, \\ x_1 + 3x_2 + 2x_3 + x_4 + x_6 = 14, \\ 3x_1 + 2x_2 + \quad + 2x_4 + x_7 = 11, \\ \frac{5}{11}x_2 + \frac{4}{11}x_5 + \frac{5}{11}x_6 + \frac{3}{11}x_7 - x_8 = \frac{10}{11}, \\ 0 \leq x_i \leq 10, \quad i = \overline{1,7}, \;\; x_8 \geq 0. \end{cases} \tag{3.23}$$

On résout le programme (3.23) en commençant par le plan initial suivant :

$$\bar{x} = x^* + \lambda(x^0 - x^*),$$

$\lambda$ est calculé à partir de l'équation : $Z_1(\bar{x}) = 0$.

On trouve $\lambda = \frac{33}{43}$, et $\bar{x}$ aura la valeur :

$$\bar{x} = \left(\frac{83}{43}, \frac{10}{43}, \frac{176}{43}, \frac{87}{43}, \frac{10}{43}, \frac{50}{43}, \frac{30}{43}\right)'.$$

Après l'ajout de la variable d'écart $x_8$, le plan $\bar{x}$ sera égal :

$$\bar{x} = \left(\frac{83}{43}, \frac{10}{43}, \frac{176}{43}, \frac{87}{43}, \frac{10}{43}, \frac{50}{43}, \frac{30}{43}, 0\right)',$$

et $\bar{J}_B = \{1, 4, 3\} \cup \{8\}$.

Le plan de support optimal du programme (3.23) est :

$$x^1 = \left(1, 0, \frac{16}{3}, \frac{7}{3}, 0, 0, \frac{10}{3}, 0\right)', \quad J_B^1 = \{1, 4, 3, 7\}.$$

**Itération 2** $x^1$ n'étant pas entier, on recommence une nouvelle itération. Le vecteur des estimations vaut

$$E_N^1 = (E_6^1, E_2^1, E_8^1, E_5^1)' = \{1, 2, 1, 0\}.$$

L'ensemble $J_N^1$ aura la partition suivante :

$J_N^+ = \{j \in J_N^1 : \ E_j^1 > 0\} = \{6, 2, 8\}$,

$J^+ = \{j \in J_N^1 : \ E_j^1 = 0, \ \ x_j^1 - d_j^- \leq d_j^+ - x_j^1\} = \{5\}$,

$J_N^- = J^- = \emptyset$.

On calcule alors la valeur des fonctions $Z_i(x^1), \ \ i \in J_B^1$ :

$$Z_B(x^1) = \left(0, -\frac{1}{3}, -\frac{1}{3}, -\frac{1}{3}\right)'.$$

On est toujours dans le cas 1. La nouvelle coupe est : $Z_2(x) \geq 0$.

On rajoute alors au programme (3.23) la contrainte suivante :

$$\frac{2}{3}x_2 + \frac{1}{3}x_5 + \frac{2}{3}x_6 + \frac{1}{8}x_8 - x_9 = \frac{1}{3}.$$

On résout le problème résultant, en commençant par le plan de support $\{\bar{x}^1, \bar{J}_B^1\}$, où $\bar{J}_B^1 = J_B^1 \cup \{9\} = \{1, 4, 3, 7, 9\}$.

Et $\bar{x}^1 = \bar{x} + \lambda(x^1 - \bar{x})$, $\lambda$ est tel que $Z_2(\bar{x}^1) = 0$.

On trouve $\lambda = \frac{87}{130}$, et

$$\bar{x}^1 = \left(\frac{17}{13}, \frac{1}{13}, \frac{64}{13}, \frac{29}{13}, \frac{1}{13}, \frac{5}{13}, \frac{32}{13}, 0, 0\right)'.$$

Le plan optimal obtenu, après l'ajout de la coupe $Z_2(x) \geq 0$ au programme (3.23), par la méthode de support est :

$$x^2 = (1, 0, 5, 3, 1, 0, 2, 0, 0)', \quad z(x^2) = 29.$$

Ce plan étant entier, alors le processus s'arrête.

**Remarque 3.3.1.** On a résolu le problème (3.21) par les deux méthodes classiques :

- La méthode des coupes de Gomory : la solution optimale est obtenue après l'ajout de (02) coupes.
- La méthode de branch and bound : la solution optimale est obtenue après 15 branchements.
- Notre méthode : la solution optimale est obtenue après l'ajout de (02) coupes.

# Conclusion générale

Un grand nombre de résultats dans l'optimisation combinatoire a été établi dans les cinquantes dernières années. Beaucoup de nouveaux développements ont permis de résoudre une large game de problèmes de programmation en nombres entiers, ce qui a ouvert de nouveaux horizons pour obtenir des algorithmes plus efficaces dans ce domaine.

Le renforcement de la relaxation de programmes linéaires en nombres entiers par des approches polyédriques est maintenant une méthode de base pour la résolution pratique et efficace de programmes linéaires en nombres entiers. Au cours des vingt dernières années, de très nombreuses méthodes de génération d'inégalités valides ont été développées.

Après avoir rappelé certains notions d'algèbre linéaire et d'analyse convexe ainsi que les théorèmes de séparation des ensembles convexes dans le premier chapitre, nous avons présenté dans le deuxième chapitre quelques techniques classiques qui permettent de résoudre des problèmes de programmation linéaire où certaines variables sont astreintes à prendre des valeurs entières.

Notre travail a consisté en la résolution des problèmes de programmation linéaire en nombres entiers par la méthode de support, conjuguée à une procédure de coupes. Cette méthode a la particularité de tenir compte des caractéristiques des problèmes tels qu'ils sont formulés lors de leur modélisation première.

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

- Généralisation de cette méthode aux cas des problèmes à variables mixtes.
- Faire des comparaisons numériques avec les méthodes de résolution classiques.
- Application de la méthode en utilisant d'autres variantes de la méthode de support, voir même appliquer la méthode duale de support pour la résolution des programmes obtenus après l'ajout d'une coupe.
- Adaptation de cette méthode aux cas des problèmes à variables bivalentes.
- Généralisation de cette méthode aux cas des problèmes multiobjectifs.

# Bibliographie


[1] K. Aardal and S.V. Hoesel. Polyhedral Techniques in Combinatorial Optimization II. *Statistica Neerlandica*, 53, 129-178, 1999.

[2] K. Aardal, G. L. Nemhauser, and R. Weismantel. D*iscrete* O*ptimization.* Elsevier, Amsterdam, 2005.

[3] T. Achterberg. Constraint Integer Programing. *PhD thesis, Berlin*, 2007.

[4] T. Achterberg. Conflict Analysis in Mixed Integer Programming. *Discrete Optimization*, 4(1), 4-20, 2007.

[5] D. Alveras and M. W. Padberg. L*inear* O*ptimization and* E*xtensions.* Springer, Berlin, 2001.

[6] K. Andersen, G. Cornuéjols, and Y. Li. Reduce-and-Split Cuts : Improving the Performance of Mixed Integer Gomory Cuts. *Management Science*, 51, 1720-1732, 2005.

[7] A. Atamtürk. Integer-Programming Software Systems. *Annals of Operations Research*, 140, 67–124, 2005.

[8] A. Bachem and M. Grötschel. New Aspects of Polyhedral Theory. *Optimization and Operation research*, 51-106, 1982.

[9] A. Bachem, E. L. Johnson, and R. Schader. A Characterization of Minimal Valid-Inequalities for Mixed Integer Programs. *Operation Research Letters*, 1, 63-66, 1982.

[10] G. Baillargeon. P*rogrammation* L*inéaire* A*ppliquée,* O*utils d'*O*ptimisation et d'*A*ide à la* D*écision.* SMG, Québec, 1996.

[11] E. Balas. Intersection Cuts - a New Type of Cutting Planes for Integer Programming. *Journal of Operations Research*, 19, 19-39, 1971.

[12] E. Balas. Facets of the Knapsack Polytope. *Mathematical Programming, 8, 146-164*, 1975.

[13] E. Balas, S. Ceria, G. Cornuéjols, and N. Natraj. Gomory Cuts Revisited. *Operations Research Letters*, 19, 1-9, 1996.

[14] E. Balas and M. Perregaard. A Precise Correspondence Between Lift-and-Project Cuts, Simple Disjunctive Cuts, and Mixed Integer Gomory Cuts for 0-1 Programming. *Journal of Mathematical Programming*, 94, 221-245, 2003.

[15] C. Barnhart, E. Johnson, G. Nemhauser, M. Savelsbergh, and P. Vance. Branch and Price : Column Generation for Solving Huge Integer Programs. *Journal of Operations Research*, 46(3), 316-329, 1998.

[16] A. Barnivok. Polynomial Time Algorithm for Counting Integral Points in Polyhedra when the Dimension is Fixed. *Operation research*, 19, 769-779, 1994.

[17] E. M. L. Beale. Branch and Bound Methods for Mathematical Programming Systems. *Journal of Discrete Optimization*, 2, 201-219, 1979.

[18] J. F. Benders. Partitioning Procedures for Solving Mixed Variables Programming Problems. *Numerische Mathematik*, 4, 238-252, 1962.

[19] M. Bentobache. Nouvelle Méthode pour la Résolution des Problèmes de Programmation Linéaire sous forme Canonique et à Variables Bornées. *Mémoire de Magister, Université de Béjaia*, 2005.

[20] M. O. Bibi. Méthodes Adaptées de la Programmation Linéaire : Cours de Post-graduation en Recherche Opérationnelle. *Université de Béjaia*, 2007.

[21] M. O. Bibi. Support Method for Solving a Linear Quadratic Problem with Polyhedral Constraints on Control. *Optimization*, 37, 139–149, 1996.

[22] M. O. Bibi and N. Ikheneche. Optimisation par la Méthode Adaptée d'un Problème Linéaire-Quadratique Convexe à Variables Bornées. *Actes du Colloque international MSS'4, U.S.T.H.B, Alger*, 1-6, Avril 17-19, 2004.

[23] M. Bierlaire. I*ntroduction à l'*O*ptimisation* D*ifférentiable.* Presses polytechniques et universitaires romandes, Lausanne, 2006.

[24] A. Bockmayr and N. Pisaruk. Detecting Infeasibility and Generating Cuts for Mixed Integer Programming Using Constraint Programming. *Computers and Operations Research*, 33(10), 2777-2786, 2006.

[25] P. Bonami. Etude et Mise en Oeuvre d'Approches Polyédriques pour la Résolution de Programmes en Nombres Entiers ou Mixtes Généraux. *PhD Thesis*, Université de Paris 6, 2003.

[26] P. Bonami and G. Cornuejols. A Note on the MIR Closure. *Operations Research Letters*, 36, 4-6, 2008.

[27] P. Bonami, G. Cornuejols, S. Dash, M. Fischetti, and A. Lodi. Projected Chvátal-Gomory Cuts for Mixed Integer Linear Programs. *Mathematical Programming*, 113, 241-257, 2008.

[28] S. Boyd and L. Vandenberghe. C*onvex* O*ptimization.* Cambridge University Press, New York, 2004.

[29] B. Brahmi. Méthode Primale et Duale pour la Résolution des Problèmes de Programmation Quadratique Convexe. *Mémoire de Magister, Université de Béjaia*, 2007.

[30] F. Cadoux. Génération de Coupes en Optimisation Combinatoire. *Université Josef Fourier, Grenoble*, 2006.

[31] V. Chvátal. Edmonds Polytope and a Hierarchy of Combinatorial Problems. *Discrete Mathematics*, 4, 185-224, 1973.

[32] W. Cook, A. M. H. Gerards, A. Schrijver, and E. Tardos. Sensitivity Theorems in Integer Linear Programming. *Mathematical Programming, 34 : 251-264*, 1985.

[33] W. Cook, R. Kannan, and A. Schrijver. Chvátal Closures for Mixed Integer Programming Problems. *Journal of Mathematical Programming*, 47 , 155-174, 1990.

[34] G. Cornuejols. Valid Inequalities for Mixed Integer Linear Programs. *Mathematical Programming*, 112, 3-44, 2008.

[35] G. Cornuéjols and Y. Li. Elementary Closures for Integer Programs. *Operations Research Letters*, 28, 1–8, 2001.

[36] H. Crowder, E. L. Johnson, and M. W. Padberg. Solving Large–Scale Zero-One Linear Programming Problems. *Operations Research*, 31, 803-834, 1983.

[37] R. J. Dakin. A Tree-Search Algorithm for Mixed Integer Programming Problems. *The Computer Journal*, 8, 250-255, 1965.

[38] E. Danna, M. Fenelon, Z. Gu, and R. Wunderling. Generating Multiple Solutions for Mixed Integer Programming Problems, in Integer Programming and Combinatorial Optimization. *LNCS*, 4513, 280–294, 2007.

[39] G. B. Dantzig. Discrete Variable Extremum Problems. *Journal of Operations Research*, 5(2), 266-277, 1957.

[40] G. B. Dantzig. Note on Solving Linear Programs in Integers. *Naval Research Logistic Quarterly*, 6, 75-76, 1959.

[41] G. B. Dantzig and M. N. Thapa. *Linear Programming, volume II : Theory and Extensions.* Springer-Verlag, New York, 2003.

[42] S. Dash and O. Günlük. On the Strength of Gomory Mixed-Integer Cuts as Group Cuts. *IBM research report*, RC23967, 2006.

[43] S. Dey and J. P. Richard. Linear Programming-Based Lifting and its Application to Primal Cutting Plane Algorithms. *INFORMS Journal on Computing*, 21(1), 137-150, 2009.

[44] J. Edmonds and E. L. Johnson. Matching : A Well-Solved Class of Integer Linear Programs. *Journal of the Association for Computing Machinery*, 31(1), 99–113, 1984.

[45] F. Eisenbrand. On the Membership Problem for the Elementary Closure of a Polyhedron. *Combinatorica*, 19(2), 1999.

[46] S. Elhedhli. L'intégration de Méthodes de Points Intérieurs, de Concepts de Décomposition et d'Approches par Séparation et Evalution Progressive pour Résoudre des Programmes Mixtes en Nombres Entiers. *Université McGill, Québec*, 2001.

[47] R. Faure. *Précis de Recherche Opérationnelle.* Dunod, Paris, 2000.

[48] S. D. Feit. A Fast Algorithm for the two-variable Integer Programming Problem. *Journal of the Association for Computing Machinery*, 31(1), 99–113, 1984.

[49] M. Fischetti and A. Lodi. Local Branching. *Mathematical Programming*, 98, 23–47, 2003.

[50] M. Fischetti and D. P. Williamson. *Integer Programming and Combinatorial Optimization.* Springer, 12th International IPCO Conference Ithaca, New York, 2007.

[51] C. A. Floudas and P. M. Pardalos. *Encyclopedia of Optimization.* Kluwer Academic Publishers, Netherlands, 2001.

[52] A. Frangioni. About Lagrangian Methods in Integer Optimization. *Annals of Operations Research*, 139, 163-193, 2005.

[53] R. Gabassov. *Adaptive Method of Linear Programming.* Reprints of the University of Karlsruhe, Institute of Statisics and Matematics, Karlsruhe, Germany, 1993.

[54] R. Gabassov and F. M. Kirillova. Méthodes de Programmation Linéaire, volumes 1, 2 et 3. *Edition de l'Université de Minsk*, 1977, 1978 et 1980.

[55] R. Gabassov and F. M. Kirillova. Méthodes d'Optimisation. *Edition de l'Université de Minsk*, 1981.

[56] R. Gabassov, F. M. Kirillova, and O. I. Kostyukova. A Method of Solving General Linear Programming Problems. *Doklady AN BSSR*, 23(3), 197–200, 1979.

[57] R. Gabassov, F. M. Kirillova, V. M. Raketsky, and O. I. Kostyukova. Constructive Methods of Optimization, volume 4 : Convex Problems. *University Press, Minsk*, 1987.

[58] J. Gauvin. *Leçons de Programmation Mathématique.* Ecole polythechnique de Montréal, Montréal, 1995.

[59] A. M. Geoffrion. Lagrangean Relaxation for Integer Programming. *Journal of Mathematical programming*, 2, 82-114, 1974.

[60] A. Ghouila-Houri. Caractérisation des Matrices Totalement Unimodulaires. *Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences*, 254, 1192-1194, 1962.

[61] O. Günlük and Y. Pochet. Mixing Mixed Integer Inequalities. *Journal of Mathematical Programming*, 90, 429-457, 2001.

[62] R. E. Gomory. Solving Linear Programming Problems in Integers. *American Mathematical Society, 211-216*, 1958.

[63] R. E. Gomory. An Algorithm for Mixed Integer Problem. *Technical Report RM-2597*, 1960.

[64] R. E. Gomory. On the Relation Between Integer and Noninteger Solutions to Linear Programs. *Proceedings of the National Academy of Science*, 53, 260-265, 1965.

[65] R. E. Gomory. Outline of an Algorithm for Integer Solutions to Linear Programs. *Bulletin of the American Mathematical Society*, 64, 275-278, 1958.

[66] M. Gondran and M. Minoux. G*raphe et* A*lgorithmes.* Eyrolles, Paris, 1995.

[67] J. E. Graver. On the Foundations of Linear and Integer Linear Programming. *Journal of Mathematical Programming*, 9, 207-226, 1975.

[68] M. Grötschel, L. Lavász, and A. Schrijver. The Ellipsoid Method and its Consequences in Combinatorial Optimization. *Combinatorica*, 1, 169-197, 1981.

[69] Z. Gu, G. L. Nemhauser, and M. W. P. Savelsbergh. Lifted Cover Inequalities for 0-1 Integer Programs. *INFORMS Journal on Computing*, 10, 427-437, 1998.

[70] C. Guéret, C. Prins, and M. Seraux. P*rogrammation* L*inéaire.* Eyrolles, Paris, 2003.

[71] W. Harvey. Computing two-dimensional Integer Hulls. *SIAM Journal on Computing*, 28, 2285-2299, 1999.

[72] M. Henk, M. Kappe, and R. Weismantel. Integral Decomposition of Polyhedra and some Applications in Mixed Integer Programming. *Journal of Mathematical Programming*, 94(2), 2003.

[73] A. J. Hoffman and J. B. Kruskal. Integral Boundary Points of Convex Polyhedra. *Annals of Mathematics study*, 38, 223-247, 1956.

[74] K. L. Hoffman. Combinatorial Optimization : Current Successes and Directions for the Future. *George Mason University, Virginia*, 2000.

[75] S. Hosten and B. Sturmfels. Computing the Integer Programming Gap. *Combinatorica*, 27(3), 367-382, 2007.

[76] N. Ikheneche. Méthode de Support pour la Minimisation d'une Fonctionnelle Quadratique Convexe. *Mémoire de magister, Université de Bejaia*, 2004.

[77] M. Jünger, G. Reinelt, and S. Thienel. Practical Problem Solving with Cutting Plane Algorithms in Combinatorial Optimization. *DIMACS*, 20, 1995.

[78] E. L. Johnson, G. L. Nemhauser, and M. W. P. Savelsbergh. Progress in Linear Programming-Based Algorithms for Integer Programming. *INFORMS Journal on Computing*, 12(1), 2-23, 2000.

[79] R. Kannan. A Polynomial Algorithm for the two-variable Integer Programming Problem. *Journal of the Association for Computing Machinery*, 27(1), 118–122, 1980.

[80] R. M. Karp and C. H. Papadimitriou. On Linear Characterization of Combinatorial Optimization. *SIAM Journal on Computing*, 11, 620-632, 1982.

[81] A. Kaufman and A. H. Labordère. *Méthodes et Modèles de la Recherche Opérationnelle.* Dunod, Paris, 1974.

[82] H. B Korte and J. Vygen. *Combinatorial Optimization : Theory and Algorithms.* Birkhäuser, Berlin, 2006.

[83] M. Köppe, Q. Louveaux, R. Weismantel, and L. A. Wolsey. Extended Formulations for Gomory Corner Polyhedra. *Journal of Discrete Optimization*, 1(2) , 141-165, 2004.

[84] M. Köppe and R. Weismantel. Cutting Planes From a Mixed Integer Farkas Lemma. *Journal of Operations Research*, 32 , 207-211, 2004.

[85] A. H. Land and A. G. Doig. An Automatic Method of Solving Discrete Programming Problems. *Econometrica*, 28, 497-520, 1960.

[86] J. B. Lasserre. The Integer Hull of a Convex Rational Polytope. *Discrete and Computational Geometry*, 32, 129-139, 2004.

[87] A. Laugier. *Cônes de Matrices et Programmation Mathématique : Quelques Applications.* Thèse de doctorat, Université de Nice-Sophia Antipolis, 2002.

[88] J. Lee, G. Editors, and A. N. Letchford. Mixed Integer Programming. *Journal of Discrete Optimization*, 4(1), 1-2, 2007.

[89] H. W. Lenstra. Integer Programming with a Fixed Number of Variables. *Journal Mathematics of Operations Research*, 8(4), 538 – 548, 1983.

[90] A. N. Letchford and A. Lodi. Strengthening Chvátal-Gomory Cuts and Gomory Fractional Cuts. *Operations Research Letters*, 30(2), 74-82, 2002.

[91] J. T. Linderoth and M. W. P. Savelsbergh. A Computational Study of Search Strategies for Mixed Integer Programming. *INFORMS Journal on Computing*, 11(2), 173-187, 1999.

[92] Q. Louveaux. Exploring Structure and Reformulations in Different Integer Programming Algorithms. *Thèse de doctorat, Université Catholique de Louvain*, 2006.

[93] A. Mahjoub. *Efficacité des Approches Polyèdrales en Optimisation Combinatoire.* Journées Franciliennes de Recherche Opérationnelle, CNAM, Clermont-Ferrand, 2005.

[94] H. Marchand, A. Martin, R. Weismantel, and L. A. Wolsey. Cutting Planes in Integer and Mixed Integer Programming. *Discrete Applied Mathematics*, 123/124, 391-440, 2002.

[95] H. Marchand and L. A. Wolsey. Aggregation and Mixed Integer Rounding to Solve MIPs. *Journal of Operations Research*, 49, 363-371, 2001.

[96] F. Margot. *Composition de Polytopes Combinatoires.* Presses polytechniques et universitaires romandes, Lausanne, 1994.

[97] A. Matel. *Techniques et Applications de la Recherche Opérationnelle.* Gaëten Morin, Québec, 1979.

[98] J. F. Maurras. Sous les Facettes des Polyèdres. *Laboratoire d'Informatique de Marseille*, Université de Marseille, 1999.

[99] T. Mernache. Méthode Adaptée pour la Résolution des Problèmes de Programmation Linéaire Multiobjectifs. *Mémoire de Magister, Université de Béjaia*, 2007.

[100] R. R. Meyer. On the Existence of Optimal Solutions to Integer and Mixed-Integer Programming Problems. *Mathematical Programming*, 7,223-235, 1974.

[101] A. Migdalas, P. M. Pardalos, and P. Varbrand. *From Local to Global Optimization.* Kluwer Academic Publishers, New York, 2001.

[102] M. Minoux. *Programmation Mathématique- Théorie et Algorithmes, Tome1 et Tome2.* Bordas et C.N.E.T- E.N.S.T., Paris, 1983.

[103] J. Monnot, V. Paschos, and S. Toulouse. A*pproximation* P*olynomiale des* P*roblèmes* NP*-difficiles.* Lavoisier, Paris, 2002.

[104] G. L. Nemhauser, M. W. P Savelsbergh, and G. C. Sigismondi. A mixed Integer Optimizer. *Operations Research Letters*, 15, 47-58, 1993.

[105] G. L. Nemhauser and L. A. Wolsey. I*nteger and* C*ombinatorial* O*ptimization.* Wiley-Interscience, New Jersy, 1999.

[106] G. L Nemhauser and L. A. Wolsey. A Recursive Procedure for Generating all Cuts for 0-1 Mixed Integer Programs. *Mathematical Programming*, 46, 379-390, 1990.

[107] M. W. Padberg and G. Rinaldi. A Branch-and-Cut Algorithm for the Resolution of Large-Scale Symmetric Traveling Salesman Problems. *SIAM Review*, 33(1), 60-100, 1991.

[108] C. H. Papadimitriou and K. Steiglitz. C*ombinatorial* O*ptimization,* A*lgorithms and* C*omplexity.* Prentice-Hall, New Jersey, 1982.

[109] V. T. Paschos. C*omplexité et* A*pproximation* P*olynomiale.* Hermès science, Paris, 2004.

[110] Y. Pochet and L. A. Wolsey. P*roduction* P*lanning by* M*ixed* I*nteger* P*rogramming.* Springer, New York, 2006.

[111] C. Prins. A*lgorithmes de* G*raphes.* Eyrolles, Paris, 1997.

[112] T. Ralphs and M. Galati. Decomposition and Dynamic Cut Generation in Integer linear Programming. *Journal of Mathematical Programming*, 106, 261-285, 2006.

[113] T.J. Van Roy and L. A. Wolsey. Solving Mixed Integer Programming Problems Using Automatic Reformulation. *Operations Research Letters*, 35(1), 45-57, 1987.

[114] M. Sakarovitch. O*ptimisation* C*ombinatoire,* P*rogrammation* D*iscrète.* Hermann, Paris, 1984.

[115] A. Schrijver. Polyhedral Proof Methods in Combinatorial Optimisation. *Discrete Applied Mathematics, 14, 111-134*, 1986.

[116] A. Schrijver. T*heory of* L*inear and* I*nteger* P*rogramming.* John Wiley, Chichester, 1986.

[117] A. Schrijver. C*ombinatorial* O*ptimization :* P*olyhedra and* E*fficiency.* Springer, Chichester, 2003.

[118] A. Schrijver. On total dual integrality. *Journal of Linear Algebra and its Applications*, 38, 27-32, 1981.

[119] A. Schrijver. On Cutting Planes. *Annals of Discrete Mathematics*, 9, 291-296, 1980.

[120] H. D. Sherali and W. P. Adams. A Hierarchy of Relaxations and Convex hull Characterizations for Mixed-Integer Zero-One Programming Problems. *Discrete Applied Mathematics*, 52(1), 83-106, 1994.

[121] J. Teghem. *Programmation Linéaire.* Ellipses, Paris, 1996.

[122] M. Truchon. *Théorie de l'Optimisation Statistique et Différentiable.* Gaëtan morin, Canada, 1987.

[123] D. Villeneuve and als. On Compact Formulations for Integer Programs Solved by Column Generation. *GERAD and Ecole Polytechnique de Montréal*, 2003.

[124] J. Vygen. *Combinatorial Optimization (Theory and Application).* Springer, New York, 2000.

[125] D. Werra, T. M. Liebling, and J. F. Hêche. *Recherche Opérationnelle pour l'Ingénieur I.* Presses polytechniques et universitaires romandes, Lausanne, 2003.

[126] C. Wilbaut and S. Hanafi. New Convergent Heuristics for 0-1 Mixed Integer Programming. *European Journal of Operational Research*, 195(1), 62-74, 2009.

[127] D. Wolf. Recherche Opérationnelle. *Université du Littoral Côte d'Opale, Dunkerque*, 2006.

[128] L. A. Wolsey. *Integer Programming.* Wiley-Interscience, New Jersy, 1998.

[129] L. A. Wolsey. Facets and Strong Valid Inequalities for Integer Programs. *Operations Research*, 24, 367-372, 1976.

[130] L. A. Wolsey. Valid Inequalities for Mixed Integer Programs with Generalized and Variable Upper Bound Constraints. *Discrete Applied Mathematics*, 25, 251-261, 1990.

# *Résumé*

Les problèmes d'optimisation combinatoire se rencontrent dans beaucoup de disciplines. Ces problèmes sont naturellement faciles à formuler mathématiquement, mais difficiles à résoudre. Il est connu que la plupart d'entres eux appartient à la classe des problèmes NP-difficiles.
Dans ce mémoire, il a été présenté un état de l'art des méthodes classiques de résolution des problèmes de programmation linéaire en nombres entiers et mixtes. On a d'abord donné un rappel sur l'analyse convexe, les polyèdres ainsi que sur les matrices unimodulaires. Ensuite, on a fait une synthèse sur les méthodes classiques élaborées dans ce domaine, à savoir, la méthode de Branch and Bound et les coupes de Gomory.
Dans ce travail, on a élaboré une méthode de support pour la résolution d'un problème de programmation linéaire en nombres entiers.

**Mots clés :** Convexité, matrices unimodulaires, programmation linéaire en nombres entiers et mixtes, méthode des plans sécants, méthode de Branch and Bound, méthode de support.

# *Abstract*

Combinatorial optimization problems occur in many disciplines. These problems are naturaly easy to formulate mathematically, but more difficult to solve. It is known that more of them belongs to the class of the NP-difficult problems.
In this thesis, it is presented a state of the art on the classical resolution methods of mixed integer programming problems. After recalling some fundamental concepts of linear algebra, convex analysis, polyhedrons and unimodular matrices, we presente the classical methods elaborated in this field, namely, Branch and Bound and cutting plane methods.
In this work, we have elaborated a support method for the resolution of a linear integer problem.

**Keywords :** Convexity, unimodular matrices, mixed integer linear programming, cutting plane method, branch and bound method, support method.