Master M2 IMSD

Recherche Opérationnelle

Chapitre 2 : Programmation linéaire en nombres entiers

J.-F. Scheid

2025-26 (version du 28/12/2025)

Table des matières

I) Introduction

Deux types de problèmes en programmation linéaire en nombres entiers (PLNE) :

- Pas besoin d'imposer le caractère entier de la solution : il résulte directement de la structure du problème (propriétés algébriques de la matrice A des contraintes).
- On doit imposer l'intégrité de la solution, faute de quoi la solution optimale est non-entière (réelle).

Exemple. Problème de sac-à-dos (*knapsack* en anglais).

Un randonneur emporte dans son sac n objets dont le poids total ne doit pas excédé P. Chaque objet i a un poids p_i et possède une utilité c_i. Quels objets le randonneur doit-il prendre pour maximiser l'utilité totale sans dépasser le poids total P ?

Modélisation.

$$\text{variables binaires } x_i = \begin{cases} 1 & \text{si l'objet } i \text{ est emporté} \\ 0 & \text{sinon} \end{cases}$$

$$\max_{\mathbf{x}\in\mathbb{R}^n} F(\mathbf{x}) = \sum_{i=1}^{n} c_i x_i$$
$$\begin{cases} \displaystyle\sum_{i=1}^{n} p_i x_i \leq P \\ x_i \in \{0,1\} \end{cases}$$

Si on remplace la condition $x_i \in \{0,1\}$ par $x_i \in [0,1]$ (les x_i réelles) alors la solution optimale du problème de sac-à-dos n'est pas entière en général (et donc non binaire).

II) Solutions optimales à valeurs entières

Soit le PL sous forme standard

$$\max_{\mathbf{x}\in\mathbb{R}^n} \left[F(\mathbf{x}) = \mathbf{c}^\top \mathbf{x}\right]$$
$$\begin{cases} A\mathbf{x} = \mathbf{b} \\ \mathbf{x} \geq \mathbf{0} \end{cases}$$

où A est une matrice de taille $m \times n$ à coefficients **entiers**. Le vecteur $\mathbf{b}$ est aussi **entier**. En général la solution optimale de PL (quand elle existe) n'est pas entière.

On cherche des conditions sur la matrice A pour que la solution optimale soit entière.

Définition 1. *Matrice totalement unimodulaire*

Une matrice A de taille $m \times n$ est dite totalement unimodulaire (TUM) si toute sous-matrice[a] carrée de A a un déterminant qui vaut 0, $+1$ ou -1.

[a] matrice obtenue en sélectionnant certaines lignes et colonnes de A.

Remarque. Toute matrice TUM est nécessairement composée de 0, $+1$

Théorème 1.

Soit A une matrice TUM et $\mathbf{b}$ un vecteur **entier**. Si un PL sous forme standard admet une solution optimale, il admet nécessairement une solution optimale entière.

Remarque. Le résultat est encore vrai si le PL est sous forme canonique pure (contrainte $A\mathbf{x} \leq \mathbf{b}$).

Démonstration du Théorème 1. On suppose que le PL admet une solution optimale. On sait alors qu'il existe une *solution de base* optimale $\mathbf{x}^* = \begin{pmatrix} \mathbf{x}_B^* \\ \mathbf{x}_H^* \end{pmatrix}$ avec $\mathbf{x}_B^* = A_B^{-1}\mathbf{b} \in \mathbb{R}^m$, $\mathbf{x}_H^* = \mathbf{0} \in \mathbb{R}^{n-m}$ et avec la décomposition (à une permutation près des colonnes de A)

$$A = (A_B \,|\, A_H).$$

On utilise alors le Lemme suivant.

Lemme 1.

Les 3 propositions suivantes sont équivalentes.

1. $\det(A_B) = \pm 1$
2. Pour tout vecteur $\mathbf{b}$ entier, toute solution de base associée à A_B est entière.
3. Pour toute matrice de base A_B, la matrice inverse A_B^{-1} est entière (i.e. à coefficients entiers).

Pour la démonstration du Lemme 1 ($1 \Rightarrow 2$), utiliser la règle de Cramer[1].

Si A est TUM alors on a $\det(A_B) = \pm 1$ car on ne peut avoir $\det(A_B) = 0$ (A_B est inversible). D'après le Lemme 1, toute solution de base réalisable est entière donc la solution optimale $\mathbf{x}^*$ est entière, d'où le Théorème 1. $\square$

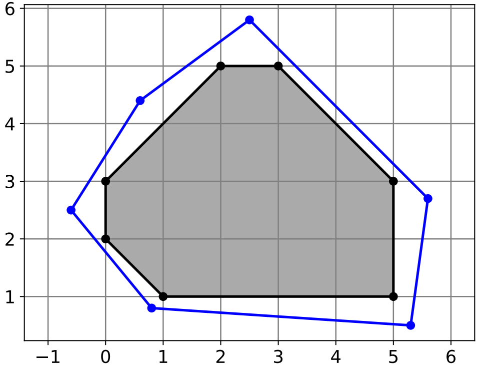
[1] Soit $\mathbf{x}_B$ la solution du système linéaire $A_B\mathbf{x}_B = \mathbf{b}$. On note $x_{B,i}$ la i-ième composante de $\mathbf{x}_B$. Si $A_B^{(i)}$ désigne la matrice obtenue en remplaçant la i-ième colonne de A_B par le vecteur $\mathbf{b}$, alors on a $x_{B,i} = \frac{det(A_B^{(i)})}{det(A_B)}$

Le Lemme 1 permet aussi d'obtenir le résultat suivant.

Théorème 2. *Hoffman-Kruskal, 1956*

Soit A une matrice à coefficients entiers.

A est TUM $\Leftrightarrow$ Pour tout vecteur $\mathbf{b}$ entier, le polyèdre $\mathcal{D}_R = \{\mathbf{x} \mid A\mathbf{x} \leq \mathbf{b},\ \mathbf{x} \geq 0\}$ est *entier*, c'est-à-dire que $\mathcal{D}_R$ est l'enveloppe convexe des points *entiers* contenus dans $\mathcal{D}_R$.



La propriété TUM est en fait une CNS pour avoir une solution entière.

Théorème 3. *(réciproque du Théorème 1)*

Soit A une matrice à coefficients entiers.

A est TUM $\Leftrightarrow$ le PL $\max_{\mathbf{x}\in\mathbb{R}^n}\left(\mathbf{c}^\top\mathbf{x};\ A\mathbf{x}\leq\mathbf{b};\ \mathbf{x}\geq\mathbf{0}\right)$ admet une solution optimale entière pour tout vecteur $\mathbf{b}$ entier pour lequel la valeur optimale du PL est finie.

Il faut comprendre, "un PL admettant une solution optimale finie, possède une solution optimale entière ssi A est TUM."

Comment reconnaitre qu'une matrice est TUM ?

Théorème 4. *Hoffman-Gale*

Soit A une matrice contenant seulement les éléments 0, $+1$ ou -1 et qui satisfait les 2 conditions suivantes :

1. Chaque colonne de A contient **au plus 2** éléments non-nuls.
2. Les lignes de A peuvent être partitionnées en 2 sous-ensembles I_1 et I_2 tels que pour chaque colonne contenant 2 éléments non-nuls :
   - si les 2 éléments non-nuls ont *le même signe* alors l'un est dans I_1 et l'autre dans I_2.
   - si les 2 éléments non-nuls ont des *signes contraires* alors ils sont tous les 2 dans I_1 ou tous les 2 dans I_2.

Alors A est TUM.

Exemple 1. Soient les matrices

$$A = \left(\begin{array}{cccc} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ \hline 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{array}\right) \begin{array}{l} \left.\vphantom{\begin{array}{c}1\\0\end{array}}\right\} I_1 = \{1, 2\} \\ \left.\vphantom{\begin{array}{c}1\\0\end{array}}\right\} I_2 = \{3, 4\} \end{array}$$

$$B = \left(\begin{array}{rrrr} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ -1 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \begin{array}{l} \left.\vphantom{\begin{array}{c}1\\0\\1\end{array}}\right\} I_1 = \{1, 2, 3\} \\ \left.\vphantom{0}\right\} I_2 = \{4\} \end{array}$$

D'après le Théorème 4, les matrices A et B sont TUM.

Exemple 2. Dans les problèmes d'affectation avec variables binaires on rencontre souvent des contraintes de la forme

$$\sum_j x_{ij} = 1, \forall i \quad \text{et} \quad \sum_i x_{ij} = 1, \forall j$$

que l'on peut écrire matriciellement par $A\mathbf{x} = \mathbf{1}$ avec

$$A = \begin{array}{c} \left(\begin{array}{ccc|ccc|c|ccc} 1 & & & 1 & & & & 1 & & \\ & \ddots & & & \ddots & & \dots & & \ddots & \\ & & 1 & & & 1 & & & & 1 \\ \hline 1 & \dots & 1 & 0 & \cdots & 0 & & & & \\ & & & 1 & \cdots & 1 & & & 0 & \\ & 0 & & & 0 & & \ddots & & & \\ & & & & & & & 1 & \cdots & 1 \end{array}\right) \end{array} \begin{array}{l} \left.\vphantom{\begin{array}{c}1\\ \ddots\\ 1\end{array}}\right\} I_1 \\ \left.\vphantom{\begin{array}{c}1\\ 1\\ 0\\ 1\end{array}}\right\} I_2 \end{array}$$

D'après le Théorème 4, la matrice A est TUM.

Conditions nécessaires et suffisantes supplémentaires.

Théorème 5.

Soit A une matrice de taille $m \times n$ contenant seulement les éléments 0, $+1$ ou -1. Les propositions suivantes sont équivalentes.

1. A est TUM.
2. Pour chaque vecteur $\mathbf{b}$ entier, le polyèdre $\mathcal{D}_R = \{\mathbf{x} \mid A\mathbf{x} \leq \mathbf{b},\ \mathbf{x} \geq 0\}$ ne possède que des sommets entiers.
3. Pour tout ensemble $J \in \{1, \cdots, n\}$ d'indices de colonnes de A, il existe une partition (J_1, J_2) de J telle que

$$\sum_{j \in J_1} a_{ij} - \sum_{j \in J_2} a_{ij} = 0, +1 \text{ ou } -1, \quad \forall i \in \{1, \cdots, n\}$$

(1) $\Leftrightarrow$ (2) : *Hoffman-Hruskal [1956]*
(1) $\Leftrightarrow$ (3) : *Ghouila-Houri [1962]*

Exemple. D'après le Théorème 5$-(\mathit{3})$, la matrice

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

est TUM alors que le Théorème 4 ne s'applique pas.

III) Méthode de Branch-and-Bound

PL où l'intégrité de la solution est une contrainte supplémentaire nécessaire (la matrice des contraintes n'est pas TUM).

1. Programmation linéaire en variables binaires

(a) Cas des coefficients positifs.

Problème de *sac-à-dos*.

$$\max_{\mathbf{x}\in\mathbb{R}^n}\left[F(\mathbf{x})=\sum_{i=1}^{n} c_i x_i\right]$$

$$\begin{cases} \displaystyle\sum_{i=1}^{n} a_i x_i \leq d \\ x_i \in \{0,1\} \end{cases}$$

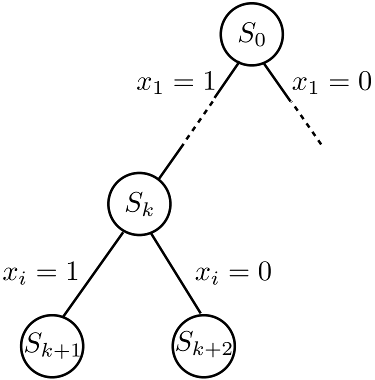
avec les coefficients $a_i \geq 0$, $c_i \geq 0$.

Il y a 2^n valeurs possibles de $\mathbf{x}$ (toutes ne sont pas réalisables). L'idée de la méthode de Branch-and-Bound est de ne pas construire l'arbre binaire en entier et de réduire l'exploration des variables par estimations.

Principe du Branch-and-Bound.

On examine successivement les variables (valeur 0 ou 1) en construisant un arbre binaire dont chaque sommet correspond à un sous-ensemble de solutions réalisables. En un sommet de cet arbre, on a déjà examiné k variables sans connaitre les $n - k$ autres et en ayant écarté :

- les sous-ensembles impossibles (solution non-réalisable)
- les sous-ensembles dont on sait que l'optimum ne s'y trouve pas.



En chaque sommet, on va évaluer 2 estimations.

1. **Estimation principale.** Elle permet de savoir si un sous-ensemble ne contient pas l'optimum. Il s'agit d'une *majoration* de la fonction coût F. Pour chaque sommet S_k, on évalue la plus petite valeur b_k telle que

$$F(\mathbf{x}) \leq b_k, \quad \forall \mathbf{x} \in \{0,1\}^n \tag{1}$$

   - **Initialement**, on choisit

$$b_0 = c_1 + c_2 + \cdots + c_n \tag{2}$$

   - On suppose connue une solution réalisable particulière ayant le coût $\overline{F}$.

     Condition d'arrêt : si $b_k < \overline{F}$, alors on arrête de construire l'arbre à partir de S_k car on sait que l'optimum ne se trouve dans les sous-ensembles issus de S_k.

2. **Estimation secondaire.** Elle permet de savoir si un sous-ensemble n'est pas réalisable. Il s'agit de déterminer une majoration e_k de la variable d'écart e de la contrainte :

$$0 \leq e = d - \sum_{i=1}^{n} a_i x_i \leq e_k \tag{3}$$

au niveau du sous-ensemble S_k.

- **Initialement**, on choisit

$$e_0 = d. \tag{4}$$

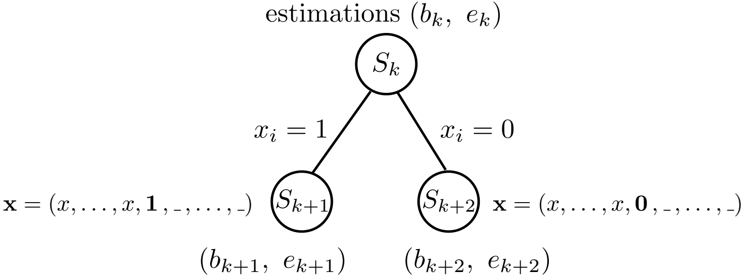
- **Condition d'arrêt :** si $e_k < 0$, on arrête d'explorer S_k.

Calcul des estimations b_k et e_k.

On se situe au sommet S_k de l'arbre et on examine la variable x_i :

$$\mathbf{x} = (x, \ldots, x, \_, \ldots, \_)$$

$$\begin{cases} \text{variables déjà examinées } x = 0 \text{ ou } 1 \\ \text{variables non encore examinées : } \_ \end{cases}$$



En tenant compte de la *positivité des coefficients* $a_i \geq 0$ et $c_i \geq 0$, les estimations (b_{k+1}, e_{k+1}) et (e_{k+2}, e_{k+2}) sont calculées à partir de (b_k, e_k) selon les formules de mise-à-jour suivantes:

$$\begin{array}{cc} (x_i = 1) & (x_i = 0) \\ \boxed{\begin{array}{l} b_{k+1} = b_k \\ e_{k+1} = e_k - a_i \end{array}} & \boxed{\begin{array}{l} b_{k+2} = b_k - c_i \\ e_{k+2} = e_k \end{array}} \end{array} \tag{5}$$

Stratégie de parcours. On explore en priorité le sommet S_k qui a la plus grande estimation b_k.

Exemple. On considère le PL en variables binaires

$$\max_{\mathbf{x}} \left[F(\mathbf{x}) = 16x_1 + 18x_2 + 15x_3\right]$$
$$\begin{cases} x_1 + 4x_2 + 3x_3 \leq 7 \\ x_1, x_2, x_3 \in \{0, 1\} \end{cases}$$

• *Détermination d'une solution réalisable particulière.*

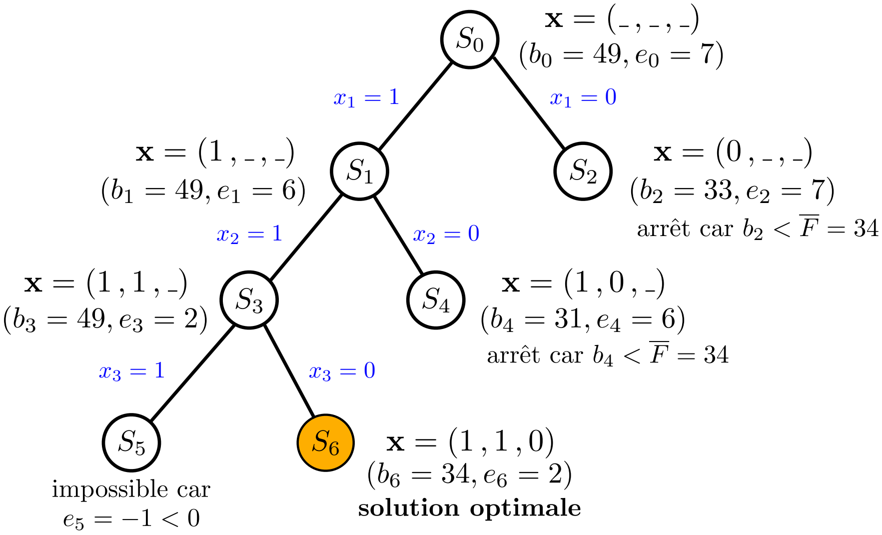
On a intérêt à trouver un $\overline{F}$ le plus grand possible. On examine les variables par coefficients décroissants dans F : d'abord x_2 ($c_2 = 18$) puis x_1 ($c_1 = 16$), enfin x_3 ($c_3 = 15$). On note $e = 7 - (x_1 + 4x_2 + 3x_3)$ la variable d'écart.

On prend

$x_2 = 1$, ce qui donne $e \leq 7 - 4 = 3$
$x_1 = 1$, ce qui donne alors $e \leq 3 - 1 = 2$
$x_3 = 1$, ce qui donne $e \leq 2 - 3 = -1$, impossible donc on prend $x_3 = 0$.

On a trouvé ainsi la solution réalisable : $x_1 = 1$, $x_2 = 1$, $x_3 = 0$ avec un coût correspond $\overline{F} = 34$.

- *Estimations initiales.* On choisit $b_0 = 16 + 18 + 15 = 49$, $e_0 = 7$.



On trouve la solution optimale $\mathbf{x}^* = (1, 1, 0)^\top$ et $\max_{\mathbf{x}} F(\mathbf{x}) = F(\mathbf{x}^*) = 34$.

(b) Cas général.

On ne suppose plus que $a_i \geq 0$ ni $c_i \geq 0$ pour tout i. S'il existe un coefficient $c_k < 0$, on fait le changement de variable

$$x'_k = 1 - x_k \in \{0, 1\}. \tag{6}$$

On obtient alors le problème

$$\max_{\mathbf{x}' \in \mathbb{R}^n} \left[F(\mathbf{x}') = \sum_{i=1}^{n} c'_i x'_i + K \right]$$
$$\begin{cases} \displaystyle\sum_{i=1}^{n} a'_i x'_i \leq d' = d - \sum_{\substack{k \text{ tq} \\ c_k < 0}} a_k \\ x'_i \in \{0, 1\} \end{cases}$$

où K est une constante (on peut la supprimer puisqu'elle ne joue aucun rôle dans la procédure d'optimisation) et $c'_i > 0$ **mais** a'_i **de signe quelconque**, pour tout i.

On procède de la même façon que précédemment pour l'estimation de F sauf pour l'estimation de la variable d'écart e où **il faut tenir compte des signes des** a'_i. On a les formules de mise-à-jour suivantes pour les estimations (b'_k, e'_k) :

$$(x'_i = 1)$$

$$\boxed{\begin{array}{l} b'_{k+1} = b'_k \\ e'_{k+1} = \begin{cases} e'_k - a'_i & \text{si } a'_i > 0 \\ e'_k & \text{si } a'_i \leq 0 \end{cases} \end{array}}$$

$$(x'_i = 0)$$

$$\boxed{\begin{array}{l} b'_{k+2} = b'_k - c'_i \\ e'_{k+2} = \begin{cases} e'_k & \text{si } a'_i > 0 \\ e'_k + a'_i & \text{si } a'_i \leq 0 \end{cases} \end{array}}$$

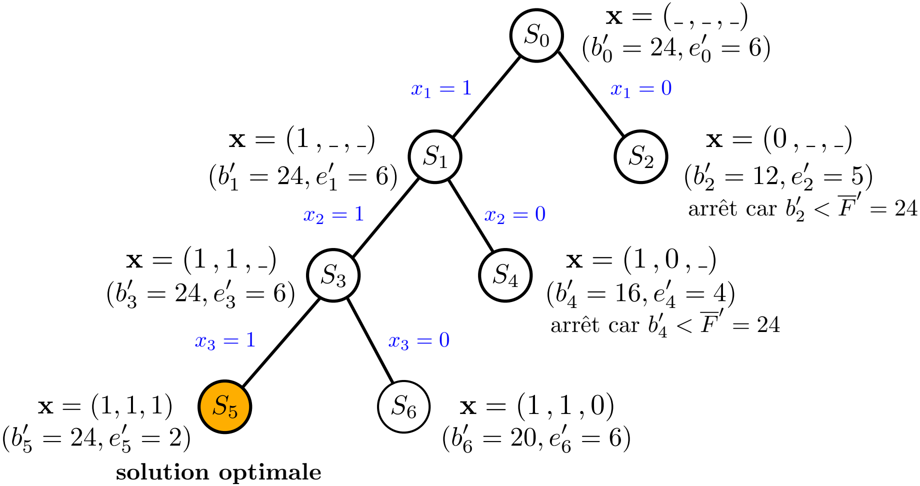
Exemple. On considère le problème suivant

$$\max_{\mathbf{x}} \left[F(\mathbf{x}) = 12x_1 - 8x_2 + 4x_3 \right]$$
$$\begin{cases} -x_1 + 2x_2 + 4x_3 \leq 5 \\ x_1, x_2, x_3 \in \{0, 1\} \end{cases}$$

On pose $x_2' = 1 - x_2$, $x_1' = x_1$, $x_3' = x_3$. Le problème devient (on supprime la constante $K = -8$)

$$\max_{\mathbf{x}'} \left[F'(\mathbf{x}') = 12x_1' + 8x_2' + 4x_3' \right]$$
$$\begin{cases} -x_1' - 2x_2' + 4x_3' \leq 3 \\ x_1', x_2', x_3' \in \{0, 1\} \end{cases}$$

- *Solution réalisable particulière :* $\mathbf{x}' = (1, 1, 1)^\top$ avec $\overline{F}' = 24$.
- *Estimations initiales.* On a $b_0' = 12 + 8 + 4 = 24$ et
$e' = 3 + x_1' + 2x_2' - 4x_3' \leq e_0' = 3 + 1 + 2 = 6$ pour tout $x_1', x_2', x_3' \in \{0, 1\}$.



Solution optimale : $\mathbf{x}'^* = (1, 1, 1)^\top$, $\max_{\mathbf{x}'} F'(\mathbf{x}') = F'(\mathbf{x}'^*) = 24$.

Retour aux variables d'origine, solution optimale du problème initial : $\mathbf{x}^* = (1, 0, 1)^\top$ et $\max_{\mathbf{x}} F(\mathbf{x}) = F(\mathbf{x}^*) = 16$.

2. PL en nombres entiers à valeurs bornées

Variables entières $x_j \in \{1, \ldots, m_j\}$ pour $i = 1, \ldots, n$

$\Rightarrow$ arbre avec $(m_j + 1)$ branches (sous-ensembles) au sommet où on examine la variable x_j.

Pour déterminer une solution réalisable particulière, on examine les variables par ordre décroissant des coefficients dans F multipliés par les valeurs maximales permises de chaque variable.

IV) Méthodes des coupes.

1. Principe général.

Pour résoudre un PL en nombres entiers, on pourrait penser résoudre le problème en relachant la contrainte d'intégrité, obtenir une solution optimale réelle et en prendre un arrondi entier. Cependant, on n'obtient pas en général la solution optimale entière de cette façon.

Exemple. Soit le PL

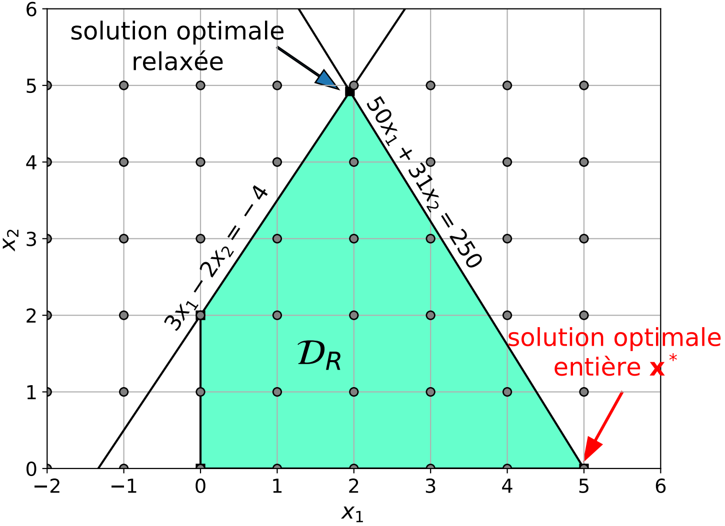
$$\max_{\mathbf{x}} \left[F(\mathbf{x}) = x_1 + 0.64 x_2\right]$$
$$\begin{cases} 50x_1 + 31x_2 \leq 250 \\ 3x_1 - 2x_2 \geq -4 \\ x_1, x_2 \in \mathbb{N} \end{cases}$$

Si on résout ce problème en relachant la contrainte d'intégrité (pb relaxé), on trouve la solution optimale *relaxée* (solution réelle) :

$\mathbf{x} = (376/193,\ 950/193) \simeq (1.948,\ 4.92)$

alors que la solution optimale *entière* est donnée par

$\mathbf{x} = (0,\ 5)$.


solution optimale
relaxée
$50x_1 + 31x_2 = 250$
$3x_1 - 2x_2 = -4$
$\mathcal{D}_R$
solution optimale
entière $\mathbf{x}^*$
x_1
x_2
-2
-1
0
1
2
3
4
5
6

Principe général des méthodes de coupes.

On considère le PL en nombres entiers

$$
(P)\quad
\begin{aligned}
&\max_{\mathbf{x}} \left[F(\mathbf{x}) = \mathbf{c}^\top \mathbf{x}\right]\\
&\begin{cases}
A\mathbf{x} \leq \mathbf{b}\\
\mathbf{x} \geq \mathbf{0}\\
\mathbf{x} \in \mathbb{N}^n
\end{cases}
\end{aligned}
$$

Il s'agit de méthodes itératives pour résoudre (P) où à chaque étape :

1. On résout un PL sans contrainte d'intégrité. On obtient une solution optimale qui est généralement non entière (sinon c'est gagné)
2. On rajoute une contrainte une supplémentaire (une coupe) à partir de la solution optimale précédente pour forcer la solution à devenir entière.
3. On recommence en (1) jusqu'à obtenir une solution entière.

Il y a différentes coupes possibles.

2. Coupes entières.

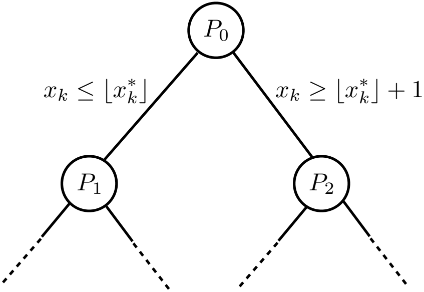
1. On résout (P_0), le problème (P) *relaxé* i.e. sans contrainte d'intégrité sur les variables; on obtient une solution optimale $\mathbf{x}^*$ (si elle existe) non nécessairement entière.
2. Si la solution optimale $\mathbf{x}^*$ de (P_0) est entière alors on arrête.
3. Sinon, il existe une composante x_k^* *non-entière*. On construit alors deux problèmes *auxiliaires* (P_1) et (P_2) en ajoutant à (P_0) des contraintes supplémentaires :

$$(P_1) \quad \begin{array}{l} \max_{\mathbf{x}} \left[F(\mathbf{x}) = \mathbf{c}^\top \mathbf{x}\right] \\ \left\{ \begin{array}{l} A\mathbf{x} \leq \mathbf{b} \\ x_k \leq \lfloor x_k^* \rfloor \\ \mathbf{x} \geq \mathbf{0} \end{array} \right. \end{array} \qquad (P_2) \quad \begin{array}{l} \max_{\mathbf{x}} \left[F(\mathbf{x}) = \mathbf{c}^\top \mathbf{x}\right] \\ \left\{ \begin{array}{l} A\mathbf{x} \leq \mathbf{b} \\ x_k \geq \lfloor x_k^* \rfloor + 1 \\ \mathbf{x} \geq \mathbf{0} \end{array} \right. \end{array}$$

   où $\lfloor \cdot \rfloor$ désigne la partie entière inférieure.
4. On résout (P_1) et (P_2). Parmi tous les problèmes auxiliaires, on sélectionne celui dont la solution est réalisable et qui possède le coût F le plus élevé. On retourne en (2) avec ce problème à la place de (P_0).

Procédure de Branch-and-Bound (coupes entières).

- La phase de séparation (branch) correspond à la construction des deux problèmes (P_1) et (P_2).
- La phase d'évaluation (bound) correspond à la détermination d'une valeur $\overline{F}$ du coût pour une solution réalisable *entière* particulière.



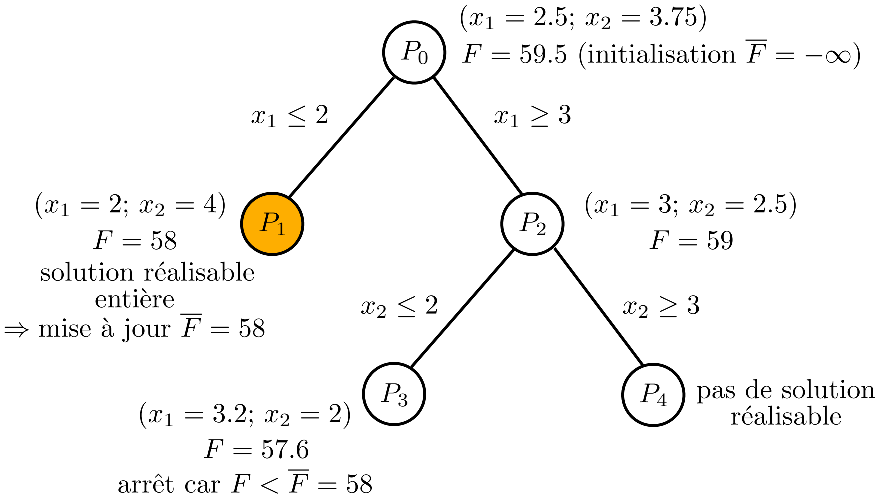
Exemple. On considère le problème suivant

$$\max_{\mathbf{x}} \left[F(\mathbf{x}) = 13x_1 + 8x_2\right]$$
$$\begin{cases} x_1 + 2x_2 \leq 10 \\ 5x_1 + 2x_2 \leq 20 \\ x_1, x_2 \in \mathbb{N} \end{cases}$$

qu'on veut résoudre par *coupes entières*.

Dans la procédure Branch-and-Bound, on utilise également le coût $\overline{F}$ d'une solution réalisable **entière** particulière.

- Si on rencontre un coût $F < \overline{F}$ à un sommet, on arrête l'exploration de ce sommet.
- Initialement, on choisit $\overline{F} = -\infty$ et on actualise $\overline{F}$ quand on rencontre pour la première fois une solution réalisable *entière*.
- On actualise la valeur $\overline{F}$ à chaque fois qu'on obtient une solution réalisable entière avec un coût $\widetilde{F}$ plus grand que $\overline{F}$. On prend alors $\overline{F} = \widetilde{F}$.



La solution optimale obtenue est $\mathbf{x}^* = (2, 4)$ avec $\max F(\mathbf{x}) = F(\mathbf{x}^*) = 58$.

3. Coupes de Gomory.

A nouveau, on veut résoudre le problème PL

$$(P) \quad \begin{array}{l} \max_{\mathbf{x}} \left[F(\mathbf{x}) = \mathbf{c}^\top \mathbf{x}\right] \\ \begin{cases} A\mathbf{x} = \mathbf{b} \\ \mathbf{x} \geq \mathbf{0} \\ \mathbf{x} \in \mathbb{N}^n \end{cases} \end{array}$$

On commence par résoudre le problème (P) *relaxé* i.e. sans les contraintes d'intégrité sur les variables. On obtient une solution de base optimale $\mathbf{x}^*$ (par simplexe) non nécessairement entière. On a

$$\mathbf{x}^* = \begin{pmatrix} \mathbf{x}_B^* \\ \mathbf{x}_H^* \end{pmatrix} \text{ avec } \mathbf{x}_H^* = \mathbf{0}.$$

On note $\lfloor x \rfloor$ la partie entière inférieure et $\{x\}$ la partie fractionnaire de x :

$$\lfloor x \rfloor \leq x < \lfloor x \rfloor + 1,$$
$$0 \leq \{x\} = x - \lfloor x \rfloor < 1.$$

Proposition 1. *Coupes de Gomory entières*

Soit $\mathbf{x} = \begin{pmatrix} \mathbf{x}_B \\ \mathbf{x}_H \end{pmatrix}$ une solution réalisable du problème (P) relaxé (pas forcément entière donc), décomposée selon la base optimale B. On a

$$\mathbf{x}_B = \mathbf{x}_B^* - M\mathbf{x}_H \qquad (\text{avec } M = A_B^{-1}A_H)$$

On suppose que la composante $(\mathbf{x}_B^*)_i$ n'est pas entière. On a

$$\boxed{e := -\{(\mathbf{x}_B^*)_i\} + \textstyle\sum_j \{M_{ij}\}(\mathbf{x}_H)_j \geq 0} \tag{7}$$

Cette nouvelle contrainte sur les variables hors-base $\mathbf{x}_H$ (avec une nouvelle variable d'écart e) est appelée **coupe de Gomory** entière.

Utilisation de la coupe : on résout le problème (P) *relaxé* en rajoutant cette contrainte (7) et on itère jusqu'à obtenir une solution optimale entière.

Preuve de la Proposition 1. On a

$$(\mathbf{x}_B)_i = (\mathbf{x}_B^*)_i - M_i \mathbf{x}_H \quad \text{où } M_i \text{ est la } i\text{-ème ligne de } M$$

$$\Leftrightarrow \; (\mathbf{x}_B)_i + \sum_j M_{ij} (\mathbf{x}_H)_j = (\mathbf{x}_B^*)_i$$

$$\Leftrightarrow \; (\mathbf{x}_B)_i + \sum_j \left(\lfloor M_{ij} \rfloor + \underbrace{\{M_{ij}\}}_{\geq 0} \right) \underbrace{(\mathbf{x}_H)_j}_{\geq 0} = \lfloor (\mathbf{x}_B^*)_i \rfloor + \{(\mathbf{x}_B^*)_i\} \tag{8}$$

d'où

$$(\mathbf{x}_B)_i + \sum_j \lfloor M_{ij} \rfloor (\mathbf{x}_H)_j \leq \lfloor (\mathbf{x}_B^*)_i \rfloor + \{(\mathbf{x}_B^*)_i\}$$

On cherche une solution réalisable **entière** donc $(\mathbf{x}_B)_i + \sum_j \lfloor M_{ij} \rfloor (\mathbf{x}_H)_j$ est entier, d'où

$$(\mathbf{x}_B)_i + \sum_j \lfloor M_{ij} \rfloor (\mathbf{x}_H)_j \leq \lfloor (\mathbf{x}_B^*)_i \rfloor \tag{9}$$

En combinant (8) et (9), on obtient $\{(\mathbf{x}_B^*)_i\} \leq \sum_j \{M_{ij}\} (\mathbf{x}_H)_j$ $\square$

Exemple. On veut résoudre le problème suivant avec les coupes de Gomory :

$$\max_{\mathbf{x}} \left[F(\mathbf{x}) = 2x_1 + x_2\right]$$
$$\begin{cases} x_1 + x_2 \leq 4 \\ x_1 - 4x_2 \leq 0 \\ x_1, x_2 \in \mathbb{N} \end{cases}$$

On considère le problème *relaxé* sous forme standard :

$$\max_{\mathbf{x}} \left[F(\mathbf{x}) = 2x_1 + x_2\right]$$
$$(P) \quad \begin{cases} x_1 + x_2 + e_1 = 4 \\ x_1 - 4x_2 + e_2 = 0 \\ x_1, x_2, e_1, e_2 \geq 0 \end{cases}$$

La solution de base optimale de (P) est (par simplexe)

$$\mathbf{x}_B^* = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 16/5 \\ 4/5 \end{pmatrix} = \begin{pmatrix} 3.2 \\ 0.8 \end{pmatrix}; \ \mathbf{x}_H^* = \begin{pmatrix} e_1 \\ e_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

avec

$$A_B = \begin{pmatrix} 1 & 1 \\ 1 & -4 \end{pmatrix}, \ A_B^{-1} = \frac{1}{5}\begin{pmatrix} 4 & 1 \\ 1 & -1 \end{pmatrix}, \ A_H = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

et

$$M = A_B^{-1} A_H = \frac{1}{5}\begin{pmatrix} 4 & 1 \\ 1 & -1 \end{pmatrix}.$$

On a

$$\begin{aligned} M_1 &= (4/5,\ 1/5) &\Rightarrow&\ \{M_1\} = (4/5,\ 1/5) \\ M_2 &= (1/5,\ -1/5) &\Rightarrow&\ \{M_2\} = (1/5,\ 4/5) \text{ car } \lfloor -1/5 \rfloor = -1. \end{aligned}$$

Deux coupes de Gomory possibles :

- *Coupe de Gomory 1.* Elle s'écrit $\{M_1\} \cdot \mathbf{x}_H \geq \{(x_B^*)_1\} = 0.2 = 1/5$, soit

$$\frac{4}{5} e_1 + \frac{1}{5} e_2 \geq \frac{1}{5} \Leftrightarrow 4e_1 + e_2 \geq 1 \tag{10}$$

- *Coupe de Gomory 2.* Elle s'écrit $\{M_2\} \cdot \mathbf{x}_H \geq \{(x_B^*)_2\} = 0.8 = 4/5$, soit

$$\frac{1}{5}e_1 + \frac{4}{5}e_2 \geq \frac{4}{5} \Leftrightarrow e_1 + 4e_2 \geq 4 \tag{11}$$

Pour mieux comprendre et interpréter les coupes de Gomory, on peut exprimer (e_1, e_2) en fonction des variables de base x_1 et x_2. On a

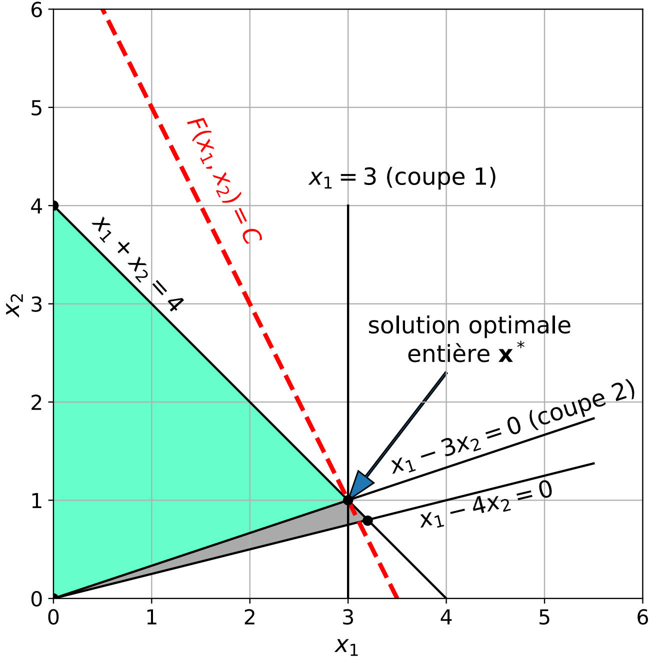
$$\begin{aligned} e_1 &= 4 - x_1 - x_2 \\ e_2 &= -x_1 + 4x_2 \end{aligned}$$

Les contraintes (10), (11) deviennent alors

$$x_1 \leq 3 \tag{12}$$

$$x_1 - 3x_2 \leq 0 \tag{13}$$

On ajoute les contraintes (12), (13) à (P). On obtient alors la solution optimale $\mathbf{x}_B^* = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$ qui est entière, avec $\max F(\mathbf{x}) = 7$.



$x_1 = 3$ (coupe 1)
$F(x_1, x_2) = C$
$x_1 + x_2 = 4$
solution optimale entière $\mathbf{x}^*$
$x_1 - 3x_2 = 0$ (coupe 2)
$x_1 - 4x_2 = 0$
x_1
x_2
0
1
2
3
4
5
6

4. Coupes de Gomory et algorithme du simplexe dual.

Dans la pratique, une fois une coupe de Gomory déterminée, on ajoute la variable d'écart e associée à cette contrainte (cf. (7)), comme une variable de base :

$$e := -\{(\mathbf{x}_B^*)_i\} + \sum_j \{\alpha_{ij}\}(\mathbf{x}_H)_j \tag{14}$$

☛ On obtient ainsi une nouvelle solution de base **qui reste optimale** (les coûts réduits restent négatifs) **mais qui n'est pas réalisable** car sa valeur $-\{(\mathbf{x}_B^*)_i\}$ est < 0.

L'algorithme du simplexe dual est alors bien adapté pour obtenir une solution de base **réalisable** : on cherche un changement de base $B \to B'$ permettant d'annuler la variable e.

- La variable *sortante* est donc e.

- Pour déterminer la variable *entrante*, on maintient les coûts réduits négatifs.

Reprenons l'exemple précédent en ajoutant la coupe de Gomory 1. dans le dictionnaire i.e. $4e_1 + e_2 \geq 1 \Leftrightarrow e_3 = -1 + 4e_1 + e_2 \geq 0$ (e_3 est la nouvelle variable) :

$$\begin{array}{lcl}
x_1 & = & \frac{16}{5} - \frac{4}{5}e_1 - \frac{1}{5}e_2 \\
x_2 & = & \frac{4}{5} - \frac{1}{5}e_1 + \frac{1}{5}e_2 \\
e_3 & = & -1 + 4e_1 + e_2 \\
\hline
F & = & \frac{36}{5} - \frac{9}{5}e_1 - \frac{1}{5}e_2
\end{array}$$

- Variable sortante : c'est e_3 !
- Variable entrante.
  1. On envisage le passage de e_1 en base. En substituant l'expression de e_1 dans F, on obtient

     $$F = \frac{135}{20} - \frac{9}{20}e_3 + \underbrace{(\frac{9}{20} - \frac{1}{5})}_{\geq 0}e_2$$

     Le coût réduit de e_2 devient positif, la solution n'est plus optimale $\Rightarrow$ e_1 ne peut pas rentrer en base.

2. On envisage le passage de e_2 en base. En substituant l'expression de e_2 dans F, on obtient

$$F = 7 - \frac{1}{5}e_3 + (\underbrace{\frac{1}{5} - \frac{9}{20}}_{\leq 0})e_2$$

Les coûts réduits sont tous négatifs, la solution obtenue est réalisable et optimale $\Rightarrow$ e_2 entre en base. La solution optimale est

$$x_1 = 3, \; x_2 = 1, \; e_2 = 1.$$

On a obtenu une solution optimale entière.

Remarques sur les coupes de Gomory.

1. L'utilisation réitérée des coupes de Gomory avec le simplexe dual converge en un nombre fini d'itérations vers une solution optimale *entière* sous les conditions suivantes :
   - *Choix de la coupe* : on choisit la coupe (7) avec la composante i telle que $\{(\mathbf{x}_B^*)_i\} = \max_{k \in B}\{(\mathbf{x}_B^*)_k\}$ i.e. la variable de base non-entière qui a la partie fractionnaire la plus grande.
   - Dans le simplexe dual, si une variable d'écart e d'une coupe rentre en base, supprimer la coupe (supprimer la variable e et la ligne correspondante dans le dictionnaire).

2. Ralph Gomory a inventé les coupes pour les PLNE dans les années 1960. Pour des problèmes mixtes (MILP) où toutes les variables ne sont pas forcément entières, Gomory établit peu de temps après un autre type de coupes, appelées coupes GMI (Gomory mixed integer cuts). Les premiers essais numériques furent assez décevants à l'époque.

Au début des années 90, des chercheurs (G. Cornuéjols, S. Ceria) sont amenés à re-tester les coupes de Gomory (les GMI) pour comparer avec leurs propres travaux et s'aperçoivent qu'elles sont finalement performantes, en particulier pour de grands problèmes[a].

Les coupes GMI sont plus générales et plus performantes que les coupes initiales de Gomory, même pour les PLNE. *Ce sont ces coupes qui sont utilisées désormais dans la plupart des solveur actuels.*

[a] les solveurs simplexes ayant fait aussi beaucoup de progrès depuis 30 ans et continuent d'en faire.