Programmation linéaire en nombres entiers

Fabian Bastin
DIRO
Université de Montréal

Présentation générale

Certaines quantités ne peuvent s'écrire sous forme de nombres réels, issus d'un domaine continu. Au contraire, certaines décisions sont par nature discrètes, et doivent se représenter à l'aide de nombres entiers. Considérons par exemple une entreprise de transport, qui décide de renouveller sa flotte de camions. Le nombre de camions à acheter est un nombre naturel.

Dans le cas linéaire, nous parlerons de programmation linéaire en nombres entiers (PLNE)

Classification

La présence de telles variables entières modifie profondément la nature des programmes sous-jacents.

Lorsqu'un problème est linéaire avec des variables entières, nous parlerons de *programmation entière mixte*.

Si toutes les variables sont entières, nous utiliserons la terminologie de de *programmation (pure) en nombres entiers*.

Si les variables entières sont à valeurs 0 ou 1 (binaires), nous parlerons de programmation 0–1 (binaire).

Programme linéaire en nombres entiers

Considérons tout d'abord le cas où toutes les variables sont entières.

$$\begin{aligned} \min\ & c^T x \\ \text{s.à}\ & Ax = b \\ & x_i \in \mathbb{N},\ i = 1, \ldots, n, \end{aligned} \tag{P}$$

où $\mathbb{N}$ est l'ensemble des entiers naturels.

Notons $\mathcal{F} = \{x \mid Ax = b, x \in \mathbb{N}^n\}$, l'ensemble réalisable de (P).

Relaxation linéaire

Il est tentant “d’oublier” les contraintes d’intégralité, et de résoudre le problème en nombres continus ainsi produit :

$$\begin{aligned} &\min\ c^T x \\ &\text{s.à } Ax = b \\ &\qquad x_i \geq 0,\ i = 1, \ldots, n, \end{aligned} \tag{PL}$$

Nous parlerons alors de *relaxation* linéaire.

Une fois le programme relaxé résolu, nous pourrions arrondir aux valeurs entières les plus proches.

Dans certains cas, cela peut fonctionner, mais cette méthode par arrondissement est parfois désastreuse.

Exemple

Considérons le programme

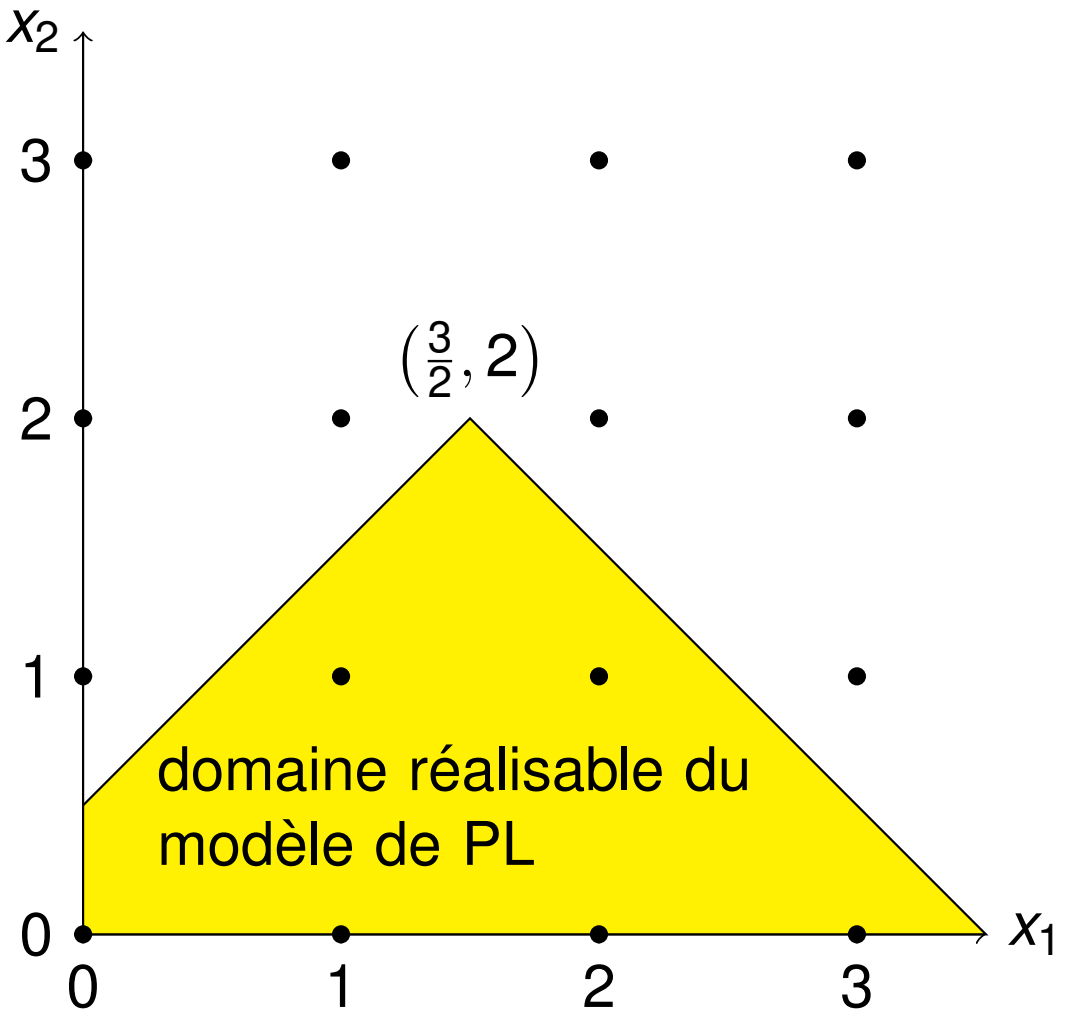
$$\begin{aligned} \max\ & z = x_2 \\ \text{t.q. } & -x_1 + x_2 \leq \frac{1}{2}, \\ & x_1 + x_2 \leq \frac{7}{2}, \\ & x_1, x_2 \geq 0 \text{ et entiers.} \end{aligned}$$

La relaxation en programme linéaire donne

$$\begin{aligned} \max\ & z = x_2 \\ \text{t.q. } & -x_1 + x_2 \leq \frac{1}{2}, \\ & x_1 + x_2 \leq \frac{7}{2}, \\ & x_1, x_2 \geq 0. \end{aligned}$$

Exemple

Solution : $\left(\frac{3}{2}, 2\right)$. Que nous arrondissions cette solution à $(1, 2)$ ou $(2, 2)$, nous n'obtenons pas de solution réalisable.



Exemple 2

Considérons le programme

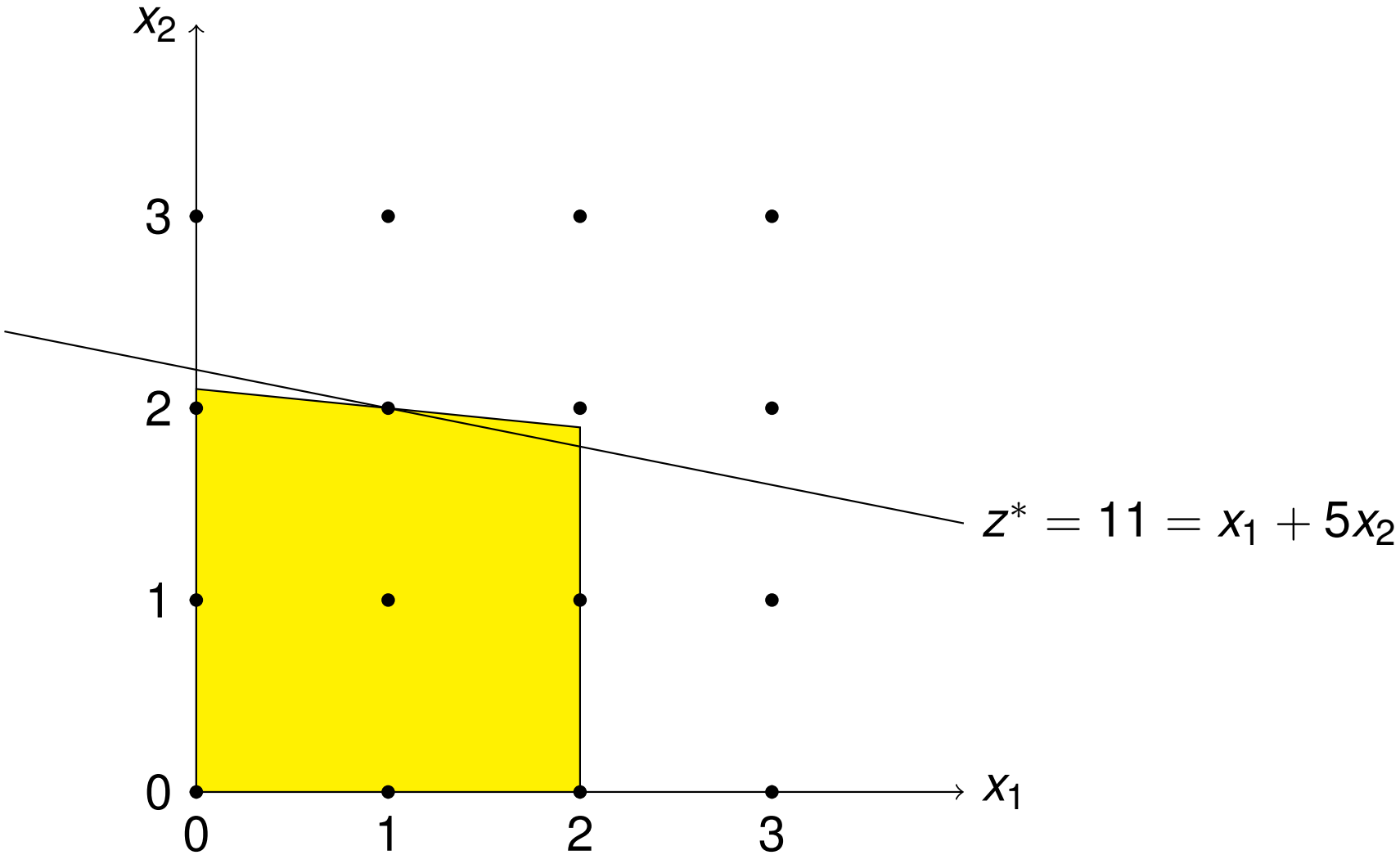
$$\begin{aligned} \max\ & z = x_1 + 5x_2 \\ \text{t.q. } & x_1 + 10x_2 \leq 21, \\ & x_2 \leq 2, \\ & x_1, x_2 \geq 0 \text{ et entiers.} \end{aligned}$$

La version relâchée de programme est

$$\begin{aligned} \max\ & z = x_1 + 5x_2 \\ \text{t.q. } & x_1 + 10x_2 \leq 21, \\ & x_2 \leq 2, \\ & x_1, x_2 \geq 0, \end{aligned}$$

qui a pour solution optimale $(2, 1.9)$. En arrondissant à $(2, 1)$ afin de garantir l'admissibilité, nous obtenons la valeur 7 pour la fonction objectif, loin de la valeur optimale du PLNE, avec pour valeur optimale de 11, en $(1, 2)$.

Exemple 2



Approche par énumération

Un modèle en nombres entiers borné (par exemple, un modèle avec uniquement des variables 0–1) possède un nombre fini de solutions. Nous pourrions envisager de toutes les énumérer, mais le nombre de solutions explose rapidement. Pour $n = 20$ variables 0–1, il y a plus d'un million de solutions possibles. Pour $n = 30$, c'est plus d'un milliard. Vite déraisonnable de procéder à une énumération complète des solutions.

Alternative : énumération partielle. Cette technique est connue sous le vocable de *branch-and-bound* (B&B). Il s'agit d'une approche *diviser-pour-régner* :

- décomposition du problème en sous-problèmes plus simples ;
- combinaison de la résolution de ces sous-problèmes pour obtenir la solution du problème original.

Algorithme de branch-and-bound

Construction d'un arbre de solutions, où chaque sous-problème correspond à un sommet.

Nous résolvons la relaxation linéaire de chaque sous-problème. L'information tirée de la relaxation linéaire nous permettra (peut-être) d'éliminer toutes les solutions pouvant être obtenues à partir de ce sommet.

Algorithme de branch & bound : cas 0–1

Un algorithme simple pour énumérer toutes les solutions d'un modèle 0–1 consiste à :

- choisir un sommet dans l'arbre des solutions ;
- choisir une variable x non encore fixée relativement à ce sommet ;
- générer les deux variables $x = 0$ et $x = 1$ (la variable x est dite *fixée*) : chaque alternative correspond à un sommet de l'arbre des solutions ;
- recommencer à partir d'un sommet pour lequel certaines variables ne sont pas encore fixées.

Algorithme de branch & bound : cas 0–1

- Racine de l'arbre : aucune variable fixée ;
- Feuilles de l'arbre : toutes les variables sont fixées. Le nombre maximal de feuilles est 2^n (pour n variables 0–1).

Le calcul de borne (ou évaluation) consiste à résoudre la relaxation linéaire en chaque sommet. L'élagage (ou élimination) consiste à utiliser l'information tirée de la résolution de la relaxation linéaire pour éliminer toutes les solutions émanant du sommet courant. Dès lors, le branch-and-bound est un algorithme de séparation et d'évaluation successives.

Exemple : California Mfg

Rappelons le problème

$$\begin{aligned}
\max\ & z = 9x_1 + 5x_2 + 6x_3 + 4x_4 \\
\text{t.q. } & 6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 10; \\
& x_3 + x_4 \leq 1; \\
& -x_1 + x_3 \leq 0; \\
& -x_2 + x_4 \leq 0; \\
& x_1, x_2, x_3, x_4 \leq 1; \\
& x_1, x_2, x_3, x_4 \geq 0; \\
& x_1, x_2, x_3, x_4 \text{ entiers.}
\end{aligned}$$

La relaxation linéaire permet aux variables de prendre les valeurs fractionnaires entre 0 et 1, ce qui conduit à la solution

$$\left(\frac{5}{6}, 1, 0, 1\right),$$

avec comme valeur $z = 16,5$. Branchons sur la variable x_1.

Exemple : California Mfg

Dénotons sous-problème 1 celui obtenu avec $x_1 = 0$:

$$\begin{aligned} \max\ & z = 5x_2 + 6x_3 + 4x_4 \\ \text{t.q. } & 3x_2 + 5x_3 + 2x_4 \leq 10; \\ & x_3 + x_4 \leq 1; \\ & x_3 \leq 0; \\ & -x_2 + x_4 \leq 0; \\ & x_2, x_3, x_4 \text{ binaires.} \end{aligned}$$

La solution de la relaxation linéaire est $(x_1, x_2, x_3, x_4) = (0, 1, 0, 1)$, avec $z = 9$.

Exemple : California Mfg

Le sous-problème 2 est celui obtenu avec $x_1 = 1$:

$$\begin{aligned} \max\ & z = 5x_2 + 6x_3 + 4x_4 + 9 \\ \text{t.q. } & 3x_2 + 5x_3 + 2x_4 \leq 4; \\ & x_3 + x_4 \leq 1; \\ & x_3 \leq 1; \\ & -x_2 + x_4 \leq 0; \\ & x_2, x_3, x_4 \text{ binaires.} \end{aligned}$$

La solution de la relaxation linéaire est alors

$$(x_1, x_2, x_3, x_4) = \left(1, \frac{4}{5}, 0, \frac{4}{5}\right),$$

avec $z = 16 + \frac{1}{5}$.

Exemple : California Mfg

Nous obtenons dès lors les bornes suivantes :

- sous-problème 1 : $Z_1 \leq 9$;
- sous-problème 2 : $Z_2 \leq 16 + \frac{1}{5}$.

Toutes les variables sont binaires et tous les paramètres dans l'objectif sont des valeurs entières. Dès lors, la borne supérieure pour le sous-problème 2 est 16.

Pour le sous-problème 1, la solution obtenue est entière : c'est la meilleure solution courante, notée Z^*, et la valeur optimale cherchée, Z, satisfait $Z \geq Z^* = 9$.

Quels sous-problèmes pouvons-nous à présent considérer afin de nous approcher de la solution optimale ? Tous les sous-problèmes actuellement traités doivent-ils donner naissance à d'autres problèmes ?

Exemple : California Mfg

Si un sous-problème ne donne lieu à aucun autre problème, nous parlerons d'élagage, en référence avec l'idée de couper la branche correspondante dans l'arbre d'exploration.

Considérons le sous-problème 1 : la solution optimale de la relaxation PL est entière. Il ne sert à rien de brancher sur les autres variables, puisque toutes les autres solutions entières (avec $x_1 = 0$) sont nécessairement de valeur inférieures ou égales à 9. Nous pouvons donc élaguer ce sommet.

Pour le sous-problème 2, la solution optimale de la relaxation PL n'est pas entière :

$$Z^* = 9 \leq Z \leq 16.$$

La branche ($x_1 = 1$) peut encore contenir une solution optimale. Mais si nous avions eu $Z_2 \leq Z^*$, nous aurions pu conclure que la branche ne pouvait améliorer la meilleure solution courante.

Élagage

Un sous-problème est elagué si une des trois conditions suivantes est satisfaite :

- test 1 : sa borne supérieure (valeur optimale de la relaxation PL) est inférieure ou égale à Z^* (valeur de la meilleure solution courante) ;
- test 2 : sa relaxation PL n'a pas de solution réalisable ;
- test 3 : la solution optimale de sa relaxation PL est entière.

Lorsque le test 3 est verifié, nous testons si la valeur optimale de la relaxation PL du sous-problème, Z_i, est supérieure à Z^*. Si $Z_i > Z^*$, alors $Z^* := Z_i$, et nous conservons la solution, qui devient la meilleure solution courante.

Algorithme branch-and-bound : cas binaire

1. Initialisation :
   (a) $Z^* := -\infty$;
   (b) appliquer le calcul de borne et les critères d'élagage à la racine (aucune variable fixée).
2. Arrêt si aucun sous-probleme non élagué.
3. Branchement :
   (a) parmi les sous-problèmes non encore élagués, choisir celui qui a été crée le plus récemment (s'il y a égalité, choisir celui de plus grande borne supérieure) ;
   (b) appliquer le Test 1 : si le sous-problème est élagué, retourner en 2.
   (c) brancher sur la prochaine variable non fixée.

Algorithme branch-and-bound : cas binaire

4. Calcul de borne :
   (a) résoudre la relaxation PL de chaque sous-problème ;
   (b) arrondir la valeur optimale si tous les paramètres de l'objectif sont entiers.
5. Élaguer un sous-problème si le test 1, 2 ou 3 est satisfait. Si le test 3 est vérifié (solution optimale de la relaxation PL est entière) ET la borne supérieure est strictement supérieure à Z^*, Z^* est mise à jour et la solution de la relaxation PL devient la meilleure solution courante. Retour en 2.

Branchement

Plusieurs choix possibles du noeud duquel brancher.

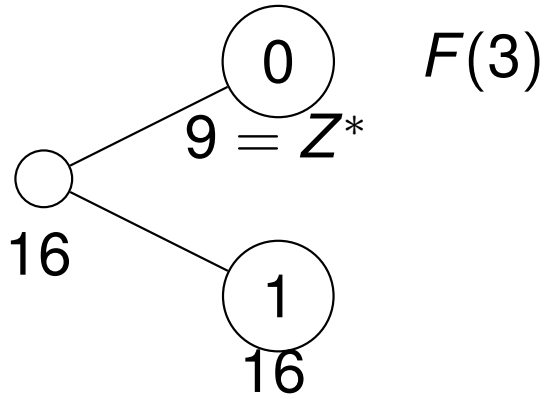
1. Choix du sous-problème le plus récemment créé. L'avantage est que cette approche facilite la réoptimisation lors du calcul de borne, car il n'y a que peu de changements apportés par rapport au dernier sous-problème traité. Le désavantage est que cela peut créer un grand nombre de sous-problèmes.
2. Règle de la meilleure borne : choisir le sous-problème ayant la plus grande borne supérieure.

Branchement

Dans cette version, la règle de branchement consiste à choisir la prochaine variable non fixée. Il est souvent plus intéressant de choisir une variable à valeur fractionnaire. En branchant sur une telle variable, il est certain que les deux sous-problèmes crées mènent à des solutions différentes de la solution courante. De nombreux autres critères existent de facon à orienter la recherche vers un élagage rapide.

Exemple : suite

Jusqu'à maintenant, voici l'arbre obtenu, en branchant sur la variable x_1.



$F(3)$ indique que le sous-probleme a été élagué (*fathomed*) en raison du Test 3.

Sélection : nous choisissons le sous-problème 2, le seul qui n'a pas encore été élagué. Nous branchons sur la prochaine variable, soit x_2. Deux nouveaux sous-problèmes sont crées :

- sous-problème 3 : $x_1 = 1$, $x_2 = 0$;
- sous-problème 4 : $x_1 = 1$, $x_2 = 1$.

Exemple : sous-problème 3

$x_1 = 1$, $x_2 = 0$. Sous-problème :

$$\begin{aligned} \max z_3 = {} & 6x_3 + 4x_4 + 9 \\ \text{t.q. } & 5x_3 + 2x_4 \leq 4 \\ & x_3 + x_4 \leq 1 \\ & x_3 \leq 1 \\ & x_4 \leq 0 \\ & x_3,\ x_4 \text{ binaires.} \end{aligned}$$

La solution de la relaxation PL est

$$(x_1, x_2, x_3, x_4) = \left(1, 0, \frac{4}{5}, 0\right),$$

et

$$Z = 13 + \frac{4}{5};\ Z_3 \leq 13.$$

Exemple : sous-problème 4

$x_1 = 1$, $x_2 = 1$. Sous-problème :

$$\begin{aligned} \max z_4 &= 6x_3 + 4x_4 + 14 \\ \text{t.q. } & 5x_3 + 2x_4 \leq 1 \\ & x_3 + x_4 \leq 1 \\ & x_3 \leq 1 \\ & x_4 \leq 1 \\ & x_3,\ x_4 \text{ binaires.} \end{aligned}$$

La solution de la relaxation PL est

$$(x_1, x_2, x_3, x_4) = \left(1, 1, 0, \frac{1}{2}\right).$$

et

$$Z = 16;\ Z_4 \leq 16.$$

Exemple : suite

Aucun des tests d'élagage ne s'applique sur ces deux sous-problèmes. Nous en choisissons un pour effectuer un branchement, puisque ce sont ceux créés le plus récemment. Nous prenojs celui de plus grande borne supérieure, soit le sous-problème 4.

Deux possibilités existent : brancher sur la variable fractionnaire, x_4, ou choisir la prochaine variable non fixée, x_3. Nous explorons la deuxième approche : nous branchons sur x_3.

Exemple : sous-problème 5

$x_1 = 1$, $x_2 = 1$, $x_3 = 0$. Sous-problème :

$$\begin{aligned} \max z_5 &= 4x_4 + 14 \\ \text{t.q. } 2x_4 &\leq 1 \\ x_4 &\leq 1 \\ x_4 &\leq 1 \\ x_4 &\text{ binaire.} \end{aligned}$$

La solution de la relaxation PL est

$$(x_1, x_2, x_3, x_4) = \left(1, 1, 0, \frac{1}{2}\right)$$

et

$$Z = 16 : \ Z_5 \leq 16.$$

Exemple : sous-problème 6

$x_1 = 1$, $x_2 = 1$, $x_3 = 1$. Sous-problème :

$$\begin{aligned} \max z_5 &= 4x_4 + 20 \\ \text{t.q. } & 2x_4 \leq -1 \\ & x_4 \leq 0 \\ & x_4 \leq 1 \\ & x_4 \text{ binaire.} \end{aligned}$$

La relaxation PL n'a pas de solution réalisable : ce sous-problème est élagué.

Exemple : suite

Le sous-problème 5 ne peut pas être élagué. Il est créé le plus récemment parmi les sous-problèmes non élagués (3 et 5) ; nous le choisissons pour effectuer un branchement. En branchant sur x_4, nous générons

- sous-problème 7 : $x_1 = 1$, $x_2 = 1$, $x_3 = 0$, $x_4 = 0$;
- sous-problème 8 : $x_1 = 1$, $x_2 = 1$, $x_3 = 0$, $x_4 = 1$.

Toutes les variables sont fixées.

- Le sous-problème 7 a pour solution $(x_1, x_2, x_3, x_4) = (1, 1, 0, 0)$, pour $Z_7 = 14$. Élagage en vertu du Test 3 (solution entière). Puisque $Z_7 > Z^*$, $Z^* = 14$ et $(1, 1, 0, 0)$ devient la meilleure solution courante.
- La solution du sous-problème 8 est $(1, 1, 0, 1)$, non-réalisable réalisable, car la première contrainte $(2x_4 \leq 1)$ est violée. Élagage en vertu du Test 2.

Exemple : sous-problème 3 et fin

Le sous-problème 3 est le seul non encore élagué.

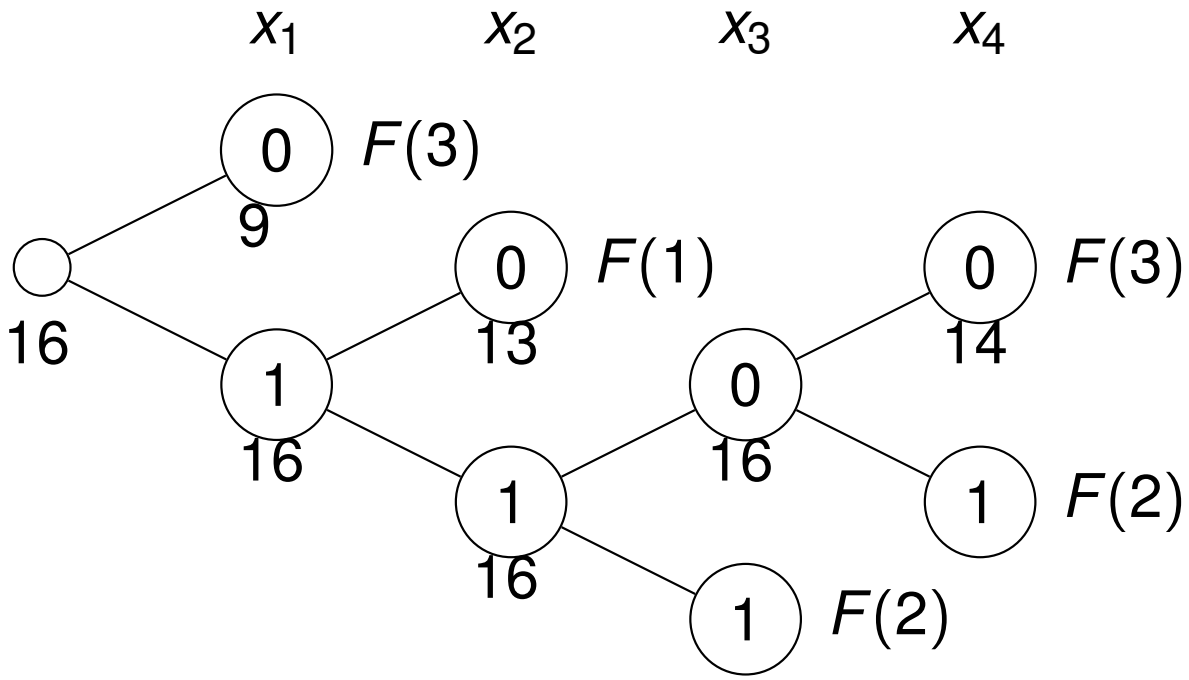
Nous l'élaguons suite à l'application du Test 1 :
$Z_3 = 13 \leq 14 = Z*$.

Arrêt : il n'y a plus de sous-problèmes non élagués.

$$x^* = (1, 1, 0, 0), \qquad Z^* = 14.$$

Exemple : arbre

L'arbre obtenu suite à l'exécution de l'algorithme se présente comme suit :



$F(j)$: le sous-problème est élagué par le Test j

Algorithme de branch & bound : cas général

- Programmation (mixte) en nombres entiers : variables entières générales et variables continues.
- Nous ignorons dans un premier temps les contraintes d'intégralité (les valeurs des variables entières sont traitées comme variables continues), et résolvons le programme linéaire résultant.
- Si la solution optimale de ce programme satisfait aux contraintes d'intégralité, alors cette solution est aussi solution optimale du programme avec variables entières. Sinon, il doit exister au moins une variable x_j dont la valeur α est fractionnaire.

Algorithme de branch & bound : cas général

Soit $d \in \mathbb{R}$. Dénotons $\lfloor d \rfloor$ le plus grand entier tel que $\lfloor d \rfloor \leq d$.

Procédure de branchement nous séparons alors le problème relaxé en deux sous-problèmes :

- un sous-problème contiendra la contrainte $x_j \leq \lfloor \alpha \rfloor$;
- ajouter au second la contrainte $x_j \geq \lceil \alpha \rceil = \lfloor \alpha \rfloor + 1$.

Nous répétons le processus pour chacun des sous-problèmes.

Cette procédure est habituellement représentée sous forme d'un arbre binaire où, à chaque niveau, une partition du sommet père s'effectue suivant la règle décrite précédemment.

Il s'agit alors de parcourir cet arbre d'énumération afin d'y trouver la solution optimale.

Algorithme de branch & bound : cas général

L'exploration d'un chemin de l'arbre peut prendre fin pour trois raisons :

1. la valeur de l'objectif correspondant à la solution optimale du problème relaxé est inférieure (moins bonne) à celle d'une solution admissible connue, possiblement obtenue à un autre sommet de l'arbre.
2. le domaine admissible d'un sous-problème devient vide ;
3. la solution devient entière ;

Dans chacun de ces trois cas on dit que le sommet est sondé, et il est inutile de pousser plus loin dans cette direction.

L'algorithme s'arrête lorsque tous les sommets sont sondés. La meilleure solution obtenue au cours du déroulement de l'algorithme est alors l'optimum global de notre problème.

Algorithme de B&B : cas général

1. Initialisation :
   (a) Poser $Z^* = -\infty$.
   (b) Appliquer le calcul de borne et les critères d'élagage à la racine (aucune variable fixée).
   (c) Si tous les sous-problèmes ont été élagués, arrêt.
2. Branchement :
   (a) Choisir le sous-problème non encore élagué créé le plus récemment. Si égalité, choisir celui de plus grande borne supérieure.
   (b) Appliquer le Test 1 : si le sous-problème est élagué, retourner en 2.
   (c) Brancher sur la prochaine variable entière à valeur non entière dans la relaxation PL.
3. Calcul de borne : résoudre la relaxation PL de chaque sous-problème.

Algorithme de B&B : cas général

4. Élagage : élaguer un sous-problème si
   (a) La borne superieure est inférieure ou égale à Z^*.
   (b) La relaxation PL n'a pas de solution réalisable.
   (c) Dans la solution optimale de la relaxation PL, toutes les variables entières sont à valeurs entières : si la borne supérieure est strictement supérieure à Z^*, Z^* est mise à jour et la solution de la relaxation PL devient la meilleure solution courante.
5. Retourner en 2.

Méthodes de coupes

Principe de base

Nous cherchons à générer un ensemble de contraintes linéaires que nous ajoutons à (P) pour engendrer un nouveau problème restreint (PR) tel que

$$\mathcal{F}(PRL) \subseteq \mathcal{F}(\text{PL})$$
$$\mathcal{F}(PR) = \mathcal{F}(\text{P})$$

où (PRL) désigne la relaxation linéaire du problème restreint.

But : si suffisamment de coupes sont ajoutées, en résolvant le problème relaxé, la solution optimale est entière et donc optimale pour (P).

Exemple

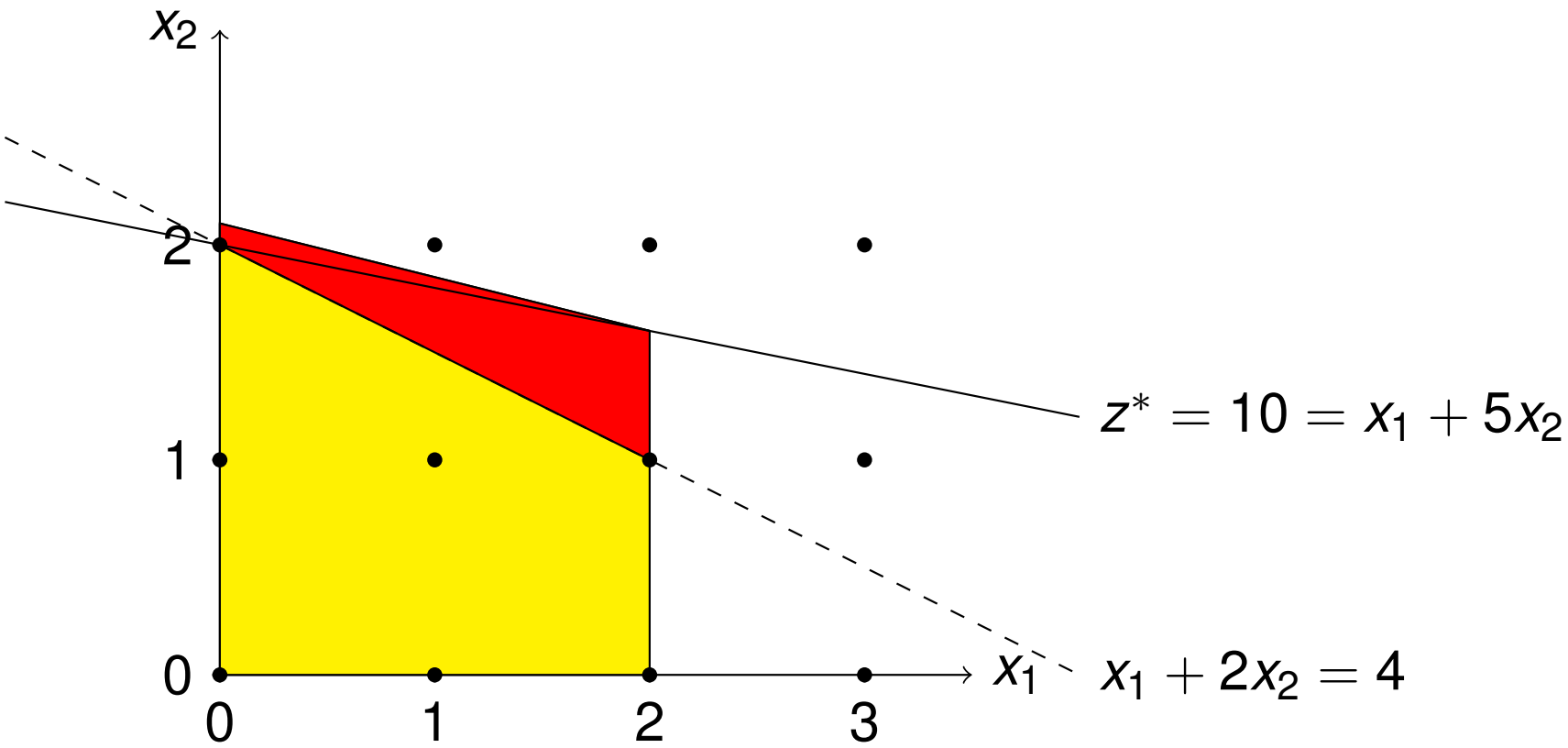
Considérons à nouveau le programme

$$\begin{aligned} \max\ & z = x_1 + 5x_2 \\ \text{t.q. } & x_1 + 4x_2 \leq 8,4, \\ & x_2 \leq 2, \\ & x_1, x_2 \geq 0 \text{ et entiers.} \end{aligned}$$

Ajoutons la contrainte $x_1 + 2x_2 \leq 4$:

$$\begin{aligned} \max\ & z = x_1 + 5x_2 \\ \text{t.q. } & x_1 + 4x_2 \leq 8,4, \\ & x_2 \leq 2, \\ & x_1 + 2x_2 \leq 4, \\ & x_1, x_2 \geq 0 \text{ et entiers.} \end{aligned}$$

Exemple



Méthode des coupes de Gomory

- Principe des méthodes de coupes : ajouter de nouvelles contraintes linéaires d'inégalité au problème pour réduire le domaine réalisable du problème relaxé sans pour autant éliminer de points du domaine réalisable du problème avec les contraintes de nombre entier sur les variables. Une telle contrainte est une coupe valide.
- La procédure consiste à résoudre une suite de problèmes relaxés jusqu'à ce qu'une solution optimale en nombres entiers soit obtenue.
- Un problème de la suite est obtenu du précédent en lui ajoutant une contrainte linéaire (coupe) supplémentaire.

Construction des coupes

Reprenons le problème

$$\begin{aligned} \min\ & c^T x \\ \text{s.à}\ & Ax = b \\ & x_i \in \mathbb{N},\ i = 1, \ldots, n. \end{aligned} \tag{P}$$

Soit B une base optimale de (PL), et x_k la variable de base dans la i^e ligne du tableau optimal prenant une valeur qui n'est pas entière.

Construction des coupes

Le tableau optimal du simplexe du problème relaxé est alors de la forme

	x_1	$\dots$	x_k	$\dots$	x_{j_1}	$\dots$	x_j	$\dots$	x_{j_m}	$\dots$	x_n	
x_{j_1}	$y_{1,1}$	$\dots$	0	$\dots$	1	$\dots$	$y_{1,j}$	$\dots$	0	$\dots$	$y_{1,n}$	$y_{1,0}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
x_k	$y_{i,1}$	$\dots$	1	$\dots$	0	$\dots$	$y_{i,j}$	$\dots$	0	$\dots$	$y_{i,n}$	$y_{i,0}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
x_{j_m}	$y_{m,1}$	$\dots$	0	$\dots$	0	$\dots$	$y_{m,j}$	$\dots$	1	$\dots$	$y_{m,n}$	$y_{m,0}$
	r_1	$\dots$	0	$\dots$	0	$\dots$	r_j	$\dots$	0	$\dots$	r_n	$-z_0$

La ligne correspondante dans le tableau final du simplexe est

$$x_k + \sum_{j \in J} y_{i,j} x_j = y_{i,0},$$

où $y_{i,0} \in \mathbb{R}^+ \setminus \mathbb{N}$ et J est l'ensemble des indices des variables hors base.

Construction des coupes

Du tableau du simplexe,

$$x_k + \sum_{j \in J} y_{i,j} x_j = y_{i,0}. \tag{1}$$

Comme $x \geq 0$, nous avons aussi

$$x_k + \sum_{j \in J} \lfloor y_{i,j} \rfloor x_j \leq y_{i,0}$$

et si x satisfait les contraintes d'intégralité,

$$x_k + \sum_{j \in J} \lfloor y_{i,j} \rfloor x_j \leq \lfloor y_{i,0} \rfloor. \tag{2}$$

Cette dernière inégalité doit être satisfaite $\forall x \in \mathcal{F}(P)$, comme par rapport à la base B, les contraintes linéaires peuvent se réécrire $B^{-1}Ax = B^{-1}b = y$.

Construction des coupes

En soustrayant (1) de (2), nous obtenons que si $x \in \mathcal{F}(P)$,

$$\sum_{j \in J} \left(\lfloor y_{i,j} \rfloor - y_{i,j} \right) x_i \leq \lfloor y_{i,0} \rfloor - y_{i,0}.$$

L'introduction de cette contrainte dans (P) n'élimine donc aucune solution de (P).

Par contre, la solution actuelle du problème relaxé (PL) ne satisfait pas cette inégalité, et est rejetée en l'incluant dans le problème. En effet, la solution actuelle a comme composante $x_k = y_{i,0} > \lfloor y_{i,0} \rfloor$, et $x_j = 0, j \in J$.

Retour au simplexe

Pour poursuivre la résolution, il suffit d'introduire la contrainte

$$\sum_{j \in J} \left(\lfloor y_{i,j} \rfloor - y_{i,j} \right) x_j \leq \lfloor y_{i,0} \rfloor - y_{i,0}.$$

qui revient à

$$\sum_{j \in J} \left(\lfloor y_{i,j} \rfloor - y_{i,j} \right) x_j + s_\tau = \lfloor y_{i,0} \rfloor - y_{i,0}.$$

où s_τ est une variable d'écart avec coût nul.

Cette contrainte est ajoutée au dernier tableau du simplexe pour générer une solution de base au nouveau problème en considérant s_τ comme la variable de base dans la nouvelle ligne du tableau. Cette solution de base n'est pas réalisable puisque

$$s_\tau = \lfloor y_{i,0} \rfloor - y_{i,0} < 0.$$

Tableau du simplexe

	x_1	$\dots$	x_k	$\dots$	x_{j_1}	$\dots$	x_j	$\dots$	x_{j_m}	$\dots$	x_n	s_τ	
x_{j_1}	$y_{1,1}$	$\dots$	0	$\dots$	1	$\dots$	$y_{1,j}$	$\dots$	0	$\dots$	$y_{1,n}$	0	$y_{1,0}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
x_k	$y_{i,1}$	$\dots$	1	$\dots$	0	$\dots$	$y_{i,j}$	$\dots$	0	$\dots$	$y_{i,n}$	0	$y_{i,0}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
x_{j_m}	$y_{m,1}$	$\dots$	0	$\dots$	0	$\dots$	$y_{m,j}$	$\dots$	1	$\dots$	$y_{m,n}$	0	$y_{m,0}$
s_τ	$\lfloor y_{i,1} \rfloor - y_{i,1}$	$\dots$	0	$\dots$	0	$\dots$	$\lfloor y_{i,j} \rfloor - y_{i,j}$	$\dots$	1	$\dots$	$\lfloor y_{i,n} \rfloor - y_{i,n}$	1	$\lfloor y_{i,0} \rfloor - y_{i,0}$
	r_1	$\dots$	0	$\dots$	0	$\dots$	r_j	$\dots$	0	$\dots$	r_n	0	$-z_0$

Simplexe dual

Il suffit de poursuivre la résolution avec l'algorithme dual du simplexe.

1. Si $\lfloor y_{ij} \rfloor = y_{ij}$, $\forall j \in J$, et si y_{i0} n'est pas entier, alors

$$x_k + \sum_{j \in J} y_{ij} x_j = y_{i0},$$

   indique que (P) n'est pas réalisable puisque le terme de gauche prend une valeur entière pour toute solution réalisable de (P) alors que le terme de droite n'est pas entier.
2. Une dérivation similaire s'applique à toutes les itérations.

Exemple 2

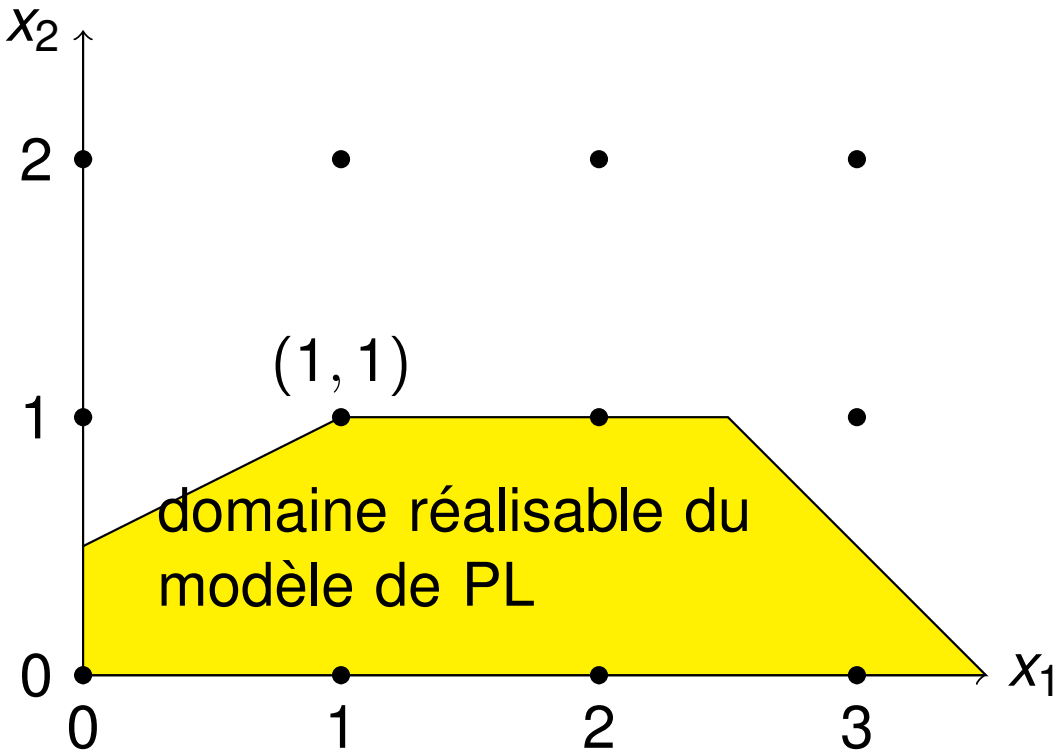
Reprenons à présent l'exemple

$$\begin{aligned} \max\ & z = x_2 \\ \text{t.q. } & -x_1 + x_2 \leq \frac{1}{2}, \\ & x_1 + x_2 \leq \frac{7}{2}, \\ & x_1, x_2 \geq 0 \text{ et entiers.} \end{aligned}$$

Si nous ajoutons l'inégalité $x_2 \leq 1$, nous obtenons le problème

$$\begin{aligned} \max\ & z = x_2 \\ \text{t.q. } & -x_1 + x_2 \leq \frac{1}{2}, \\ & x_1 + x_2 \leq \frac{7}{2}, \\ & x_2 \leq 1, \\ & x_1, x_2 \geq 0 \text{ et entiers.} \end{aligned}$$

Exemple



Dans ce cas, le point extrême $(1, 1)$ est bien solution, mais il existe d'autres solutions pour le problème relaxé qui ne sont pas entières.