

# Programmation linéaire réelle

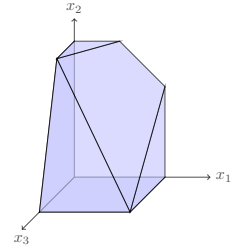
François Schwarzentruher

**Définition 1 (programme linéaire)** Un programme linéaire réel est la donnée d'une fonction linéaire à maximiser ou minimiser, sous des contraintes linéaires (in.égalités), où les variables prennent leurs valeurs dans  $\mathbb{R}$ .

## Exemple 2

Il vend des chocolats simples (1 euro), des pyramides (6 euros) et des pyramides de luxe (13 euros). Au maximum, il peut vendre 200 chocolats simples, 300 pyramides, pas plus de 400 chocolats en tout et le nombre de pyramides plus trois fois le nombre de pyramides de luxe est au plus 600.

$$\begin{cases} \text{maximiser } x_1 + 6x_2 + 13x_3 \\ x_1 \leq 200 \\ x_2 \leq 300 \\ x_1 + x_2 + x_3 \leq 400 \\ x_2 + 3x_3 \leq 600 \\ x_1, x_2, x_3 \geq 0 \\ x_1, x_2, x_3 \in \mathbb{R} \end{cases}$$



## Définition 3 Programmation linéaire réelle

entrée : un programme linéaire réel  $P$

sortie : une solution optimale de  $P$ , non borné s'il n'y a pas d'optimum, ou impossible si les contraintes sont inconsistantes.

## 1 Formes normales pour les programmes linéaires

### 1.1 Programme canonique

**Définition 4 (programme canonique)** Un programme canonique est de la forme :

$$\begin{cases} \text{maximiser } c^t x \\ Ax \leq b \\ x \geq 0 \end{cases} \quad \text{où } c \in \mathbb{R}^d, A \in \mathfrak{M}_{m,d}(\mathbb{R}) \text{ et } b \in \mathbb{R}^m.$$

$d =$ dimension $m =$ nombre d'inégalités linéaires
--

**Proposition 5** Tout programme linéaire peut s'écrire sous la forme d'un programme canonique équivalent.

DÉMONSTRATION.

1. Passer de maximiser à minimiser (ou vice et versa) : multiplier la fonction objectif par -1.
2. Passer d'égalités à des inégalités :  $a^T x = b$  devient  $a^T x \leq b$  et  $a^T x \geq b$
3. N'avoir que des variables positives : Remplacer  $x$  par  $x^+ - x^-$  et ajouter  $x^+, x^- \geq 0$ .

■

### 1.2 Programme équationnel

**Définition 6 (programme équationnel)** Un programme équationnel est de la forme

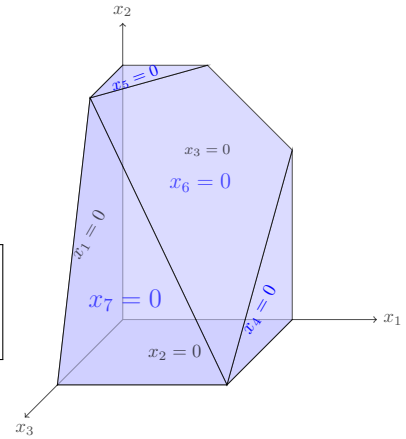
$$\begin{cases} \text{maximiser } c^t x \\ Ax = b \\ x \geq 0 \end{cases} \quad \text{où } c \in \mathbb{R}^n, A \in \mathfrak{M}_{m,n}(\mathbb{R}) \text{ et } b \in \mathbb{R}^m, \text{ où } \underline{A \text{ est de rang } m}.$$

**Proposition 7** Tout programme canonique se réécrit en un programme équationnel équivalent en introduisant des variables d'écart  $x_{d+1}, \dots, x_{d+m}$ .

**Exemple 8** On introduit des variables d'écart  $x_4, x_5, x_6, x_7$  de la façon suivante :

$$\begin{cases} \text{maximiser } x_1 + 6x_2 + 13x_3 \\ x_1 + x_4 = 200 \\ x_2 + x_5 = 300 \\ x_1 + x_2 + x_3 + x_6 = 400 \\ x_2 + 3x_3 + x_7 = 600 \end{cases}$$

$x_1 \dots \dots \dots x_d$   $x_{d+1} \dots \dots \dots x_{d+m}$   
 $d$  variables dans l'espace de départ  $m$  variables d'écart =  $m$  contraintes  
 Face = une équation  $x_i \geq 0$



## 2 Algorithme du simplexe

### 2.1 Principe

**fonction simplexe**  
 | sommet =  $(0, \dots, 0)$   
 | **tant que** sommet non optimal  
 | | trouver un sommet voisin qui améliore l'objectif  
 | **renvoyer** sommet optimal

### 2.2 Représentation d'un sommet du polyèdre

Sommet = intersection de  $d$  faces = la donnée de  $d$  variables nulles appelées variables hors base

**Définition 9 (variables de base)** Une base  $B$  est un sous-ensemble de  $\{1, \dots, n\}$  de cardinal  $m$  telle que  $A_B$  soit de rang  $m$ . Les variables dans  $x_B$  sont les variables de base.

**Définition 10 (variables hors base)** On note  $N = \{1, \dots, n\} \setminus B$ . Les variables dans  $x_N$  sont hors base.

**Définition 11 (tableau)** Étant donné une base  $B$ , un tableau est un programme linéaire sous la forme

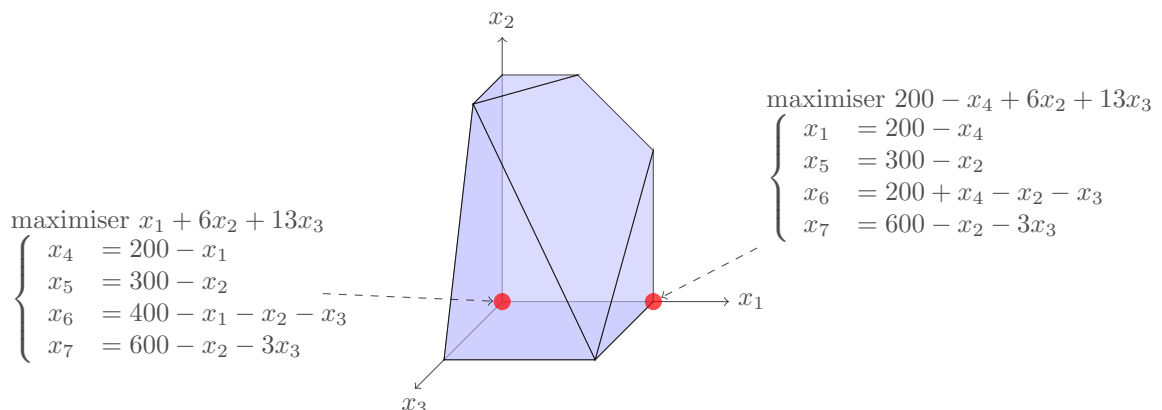
$$\begin{cases} \text{maximiser } v + c^t x_N \\ x_B = p - A' x_N \\ x \geq 0 \end{cases}$$

où  $v \in \mathbb{R}$ ,  $c' \in \mathbb{R}^{n-m}$ ,  $p \in \mathbb{R}^m$ ,  $A' \in \mathfrak{M}_{m, n-m}(\mathbb{R})$  et le vecteur  $x \in \mathbb{R}^n$  est décomposé en  $x_B \in \mathbb{R}^m$  et  $x_N \in \mathbb{R}^{n-m}$ .

tableau = sommet à l'intersection des  $x_i = 0$  où  $i \in N$   
 $v$  = valeur de la fonction objectif en ce sommet  
 $p$  = valeurs des variables de base en ce sommet

**Définition 12 (solution basique)** Un tableau admet une solution basique si  $p \geq 0$ . La solution basique est le vecteur  $x \in \mathbb{R}^n$  défini par  $x_B = p$  et  $x_N = 0$ .

#### Exemple 13



### 2.3 Pseudo-code

entrée : un tableau  $\tau$  admettant une solution basique  
 sortie : le maximum ou alors lève une exception **non borné**  
**fonction** simplexe( $\tau$ )  
 | **tant que** il y a un coefficient strictement positif dans la fonction objectif de  $\tau$   
 | |  $\tau = \text{pivot}(\tau)$   
 | **renvoyer** la solution basique de  $\tau$

entrée : un tableau  $\tau$  admettant une solution basique  
 sortie : un tableau équivalent, admettant une solution basique, améliorant non strictement l'objectif, de même espace de solutions, et ou alors lève une exception **non borné**  
**fonction** pivot( $\tau$ )  
 | **renvoyer** le tableau obtenu à partir de  $\tau$  comme suit.  
 | 1. repérer une variable  $x_e$  avec coefficient strictement positif dans la fonction objectif  
 | 2. repérer la contrainte  $x_s = \dots$  qui limite le plus la croissance de  $x_e$   
 | | s'il n'y a pas de telle contrainte **lever une exception non borné**  
 | 3. augmenter  $x_e$  jusqu'à ce que  $x_s := 0$   $x_e := 0$   $x_s \nearrow$

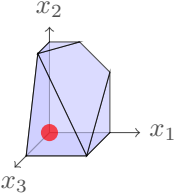
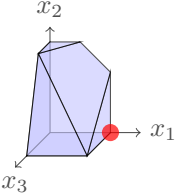
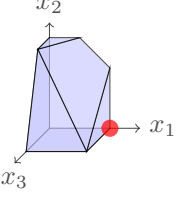
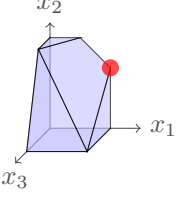
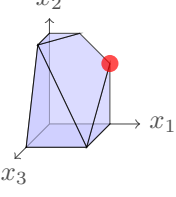
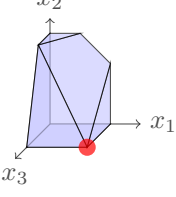
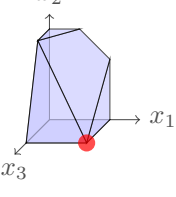
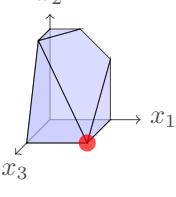
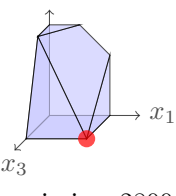
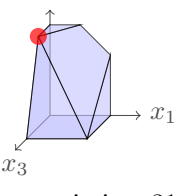
**fonction** pivot  $\left( \begin{array}{l} \text{maximiser } v + c^t x_N \\ \begin{cases} x_B = b - Ax_N \\ x \geq 0 \end{cases} \end{array} \right)$

choisir  $j_0 \in N$  tel que  $c_{j_0} > 0$   
 choisir  $i_0 \in B$  tel que  $-a_{i_0, j_0} < 0$  et  $\frac{b_{i_0}}{a_{i_0, j_0}}$  minimal, s'il n'y a pas, **lever une exception** "non borné"

**renvoyer**  $\begin{array}{l} \text{maximiser } v' + c'^t x_{N'} \\ \begin{cases} x_{B'} = b' - A'x_{N'} \\ x \geq 0 \end{cases} \end{array}$  où pour tout  $i \in B, j \in N$  :

<ul style="list-style-type: none"> <li>— <math>N' = \{i_0\} \cup N \setminus \{j\}</math>;</li> <li>— <math>B' = \{j_0\} \cup B \setminus \{i\}</math>;</li> <li>— <math>v' = v + \frac{b_{i_0} c_{j_0}}{a_{i_0, j_0}}</math></li> <li>— <math>c'_j = c_j - c_{j_0} \frac{a_{i_0, j}}{a_{i_0, j_0}}</math></li> </ul>	<ul style="list-style-type: none"> <li>— <math>c'_{i_0} = \frac{-c_{j_0}}{a_{i_0, j_0}}</math></li> <li>— <math>b'_i = b_i - a_{i, j_0} \frac{b_{i_0}}{a_{i_0, j_0}}</math></li> <li>— <math>b'_{j_0} = \frac{b_{i_0}}{a_{i_0, j_0}}</math></li> </ul>	<ul style="list-style-type: none"> <li>— <math>a'_{i, j} = a_{i, j} - \frac{a_{i, j_0} a_{i_0, j}}{a_{i_0, j_0}}</math></li> <li>— <math>a'_{i, i_0} = \frac{a_{i, j_0}}{a_{i_0, j_0}}</math></li> <li>— <math>a'_{j_0, i_0} = \frac{1}{a_{i_0, j_0}}</math></li> <li>— <math>a'_{j_0, j} = \frac{-a_{i_0, j}}{a_{i_0, j_0}}</math></li> </ul>
---	--	--

## 2.4 Exemple d'exécution

 <p>maximiser <math>x_1 + 6x_2 + 13x_3</math></p> $\begin{cases} x_4 = 200 - x_1 \\ x_5 = 300 - x_2 \\ x_6 = 400 - x_1 - x_2 - x_3 \\ x_7 = 600 - x_2 - 3x_3 \end{cases}$	$x_4 := 0$	 <p>maximiser <math>200 - x_4 + 6x_2 + 13x_3</math></p> $\begin{cases} x_1 = 200 - x_4 \\ x_5 = 300 - x_2 \\ x_6 = 200 + x_4 - x_2 - x_3 \\ x_7 = 600 - x_2 - 3x_3 \end{cases}$
 <p>maximiser <math>200 - x_4 + 6x_2 + 13x_3</math></p> $\begin{cases} x_1 = 200 - x_4 \\ x_5 = 300 - x_2 \\ x_6 = 200 + x_4 - x_2 - x_3 \\ x_7 = 600 - x_2 - 3x_3 \end{cases}$	$x_6 := 0$	 <p>maximiser <math>1400 + 5x_4 - 6x_6 + 7x_3</math></p> $\begin{cases} x_1 = 200 - x_4 \\ x_5 = 100 - x_4 + x_6 + x_3 \\ x_2 = 200 + x_4 - x_6 - x_3 \\ x_7 = 400 - x_4 + x_6 - 2x_3 \end{cases}$
 <p>maximiser <math>1400 + 5x_4 - 6x_6 + 7x_3</math></p> $\begin{cases} x_1 = 200 - x_4 \\ x_5 = 100 - x_4 + x_6 + x_3 \\ x_2 = 200 + x_4 - x_6 - x_3 \\ x_7 = 400 - x_4 + x_6 - 2x_3 \end{cases}$	$x_2 := 0$	 <p>maximiser <math>2800 + 12x_4 - 13x_6 - 7x_2</math></p> $\begin{cases} x_1 = 200 - x_4 \\ x_5 = 300 - x_2 \\ x_3 = 200 + x_4 - x_6 - x_2 \\ x_7 = -3x_4 + 3x_6 + 2x_2 \end{cases}$
 <p>maximiser <math>2800 + 12x_4 - 13x_6 - 7x_2</math></p> $\begin{cases} x_1 = 200 - x_4 \\ x_5 = 300 - x_2 \\ x_3 = 200 + x_4 - x_6 - x_2 \\ x_7 = -3x_4 + 3x_6 + 2x_2 \end{cases}$	$x_7 := 0$	 <p>maximiser <math>2800 - x_6 - 4x_7 + x_2</math></p> $\begin{cases} x_1 = 200 - x_6 + \frac{x_7}{3} - \frac{2x_2}{3} \\ x_5 = 300 - x_2 \\ x_3 = 200 - \frac{x_7}{3} - \frac{x_2}{3} \\ x_4 = x_6 - \frac{x_7}{3} + \frac{2x_2}{3} \end{cases}$
 <p>maximiser <math>2800 - x_6 - 4x_7 + x_2</math></p> $\begin{cases} x_1 = 200 - x_6 + \frac{x_7}{3} - \frac{2x_2}{3} \\ x_5 = 300 - x_2 \\ x_3 = 200 - \frac{x_7}{3} - \frac{x_2}{3} \\ x_4 = x_6 - \frac{x_7}{3} + \frac{2x_2}{3} \end{cases}$	$x_5 := 0$	 <p>maximiser <math>3100 - 4x_7 - x_6 - x_5</math></p> $\begin{cases} x_1 = 0 + \dots \\ x_2 = 300 - \epsilon_2 \\ x_3 = 100 + \dots \\ \epsilon_1 = \dots \end{cases}$

## 2.5 Propriétés de l'algorithme

**Proposition 14 (admis car lourd)** La fonction pivot est correct.

**Théorème 15** L'algorithme du simplexe, s'il termine, retourne bien la valeur maximale de l'objectif.

DÉMONSTRATION.

$$\begin{cases} \text{maximiser } v + c^t x_N \\ x_B = p - A' x_N \\ x \geq 0 \end{cases}$$

L'algorithme termine donc les coefficients de  $c'$  sont tous négatifs. Sur l'espace des solutions, la valeur  $v$  majore l'objectif. La valeur  $v$  est atteinte pour la solution basique, qui est donc solution optimale. ■

## 2.6 A toute base, son unique tableau

**Proposition 16** Étant donné une base  $B$ , il y a un unique tableau équivalent au programme équationnel. C'est :

$$\begin{cases} \text{maximiser } v + c^t x_N \\ x_B = p - A' x_N \\ x \geq 0 \end{cases} \quad \text{avec } v = c_B^t A_B^{-1} b; c' = c_N - (c_B^t A_B^{-1} A_N)^t; p = A_B^{-1} b; \text{ et } A' = A_B^{-1} A_N.$$

DÉMONSTRATION. Le système de contraintes  $Ax = b$  s'écrit  $A_N x_N + A_B x_B = b$ . D'où  $x_B = A_B^{-1} b - A_B^{-1} A_N x_N$ . L'objectif  $c^t x$  s'écrit  $c_B^t x_B + c_N^t x_N = c_B^t (A_B^{-1} b - A_B^{-1} A_N x_N) + c_N^t x_N = c_B^t (A_B^{-1} b + (c_N - (c_B^t A_B^{-1} A_N)^t) x_N)$ . ■

**Exemple 1** Considérons le programme équationnel

$$\begin{cases} \text{maximiser } x_1 + 6x_2 + 13x_3 \\ x_1 + x_4 = 200 \\ x_2 + x_5 = 300 \\ x_1 + x_2 + x_3 + x_6 = 400 \\ x_2 + 3x_3 + x_7 = 600 \end{cases}$$

Pour la base  $B = \{4, 5, 6, 7\}$ , on a  $A_B = Id_4$ ,  $c_B^t = (0, 0, 0, 0)$  et  $c_N^t = (1, 6, 13)$ .

Pour la base  $B = \{1, 5, 6, 7\}$ , on a  $A_B = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ ,  $c_B = (1, 0, 0, 0)$  et  $c_N = (6, 13, 0)$ .

## 3 Terminaison de l'algorithme du simplexe

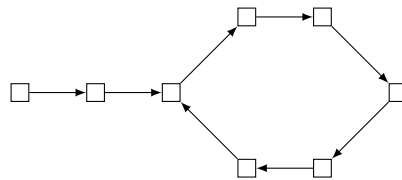
**Définition 17 (règle de Bland)** La règle de Bland est la stratégie consistant à choisir la variable entrante candidate d'indice minimal, et la variable sortante candidate d'indice minimal.

**Proposition 18 (terminaison)** En appliquant la règle de Bland, l'algorithme du simplexe termine.

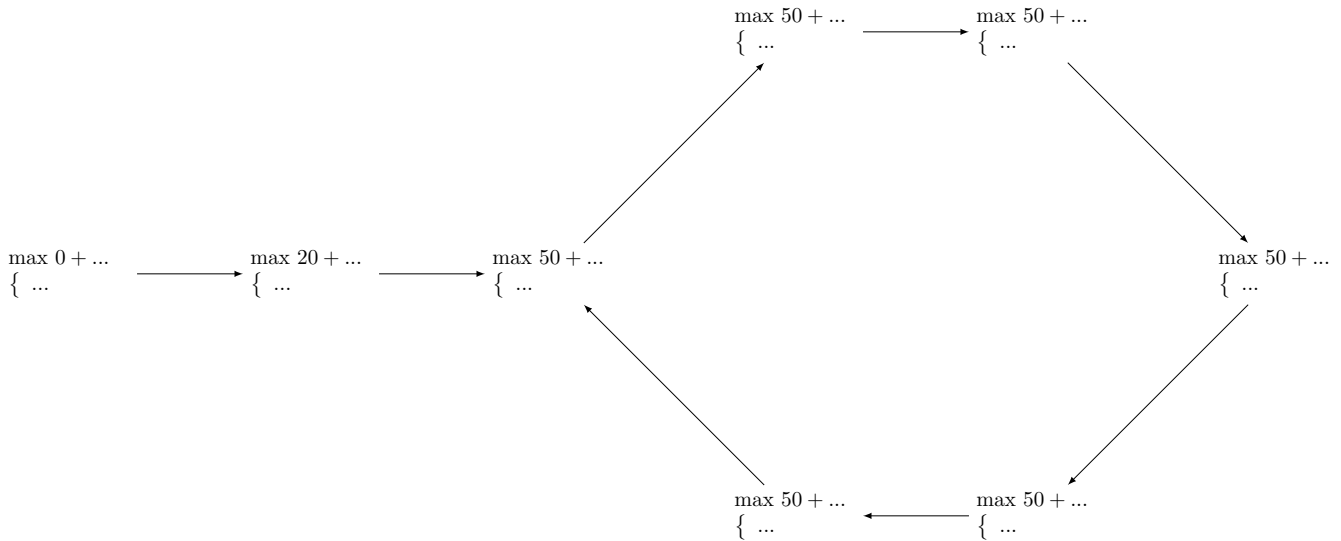
DÉMONSTRATION. Par l'absurde. Supposons que l'algorithme ne termine pas sur le programme équationnel suivant :

$$\begin{cases} \text{maximiser } c^t x \\ Ax = b \\ x \geq 0 \end{cases}$$

Nous avons donc un cycle dans l'exécution. Dans le dessin suivant, chaque point représente une base (et le tableau associé) et une flèche représente un pivotage.



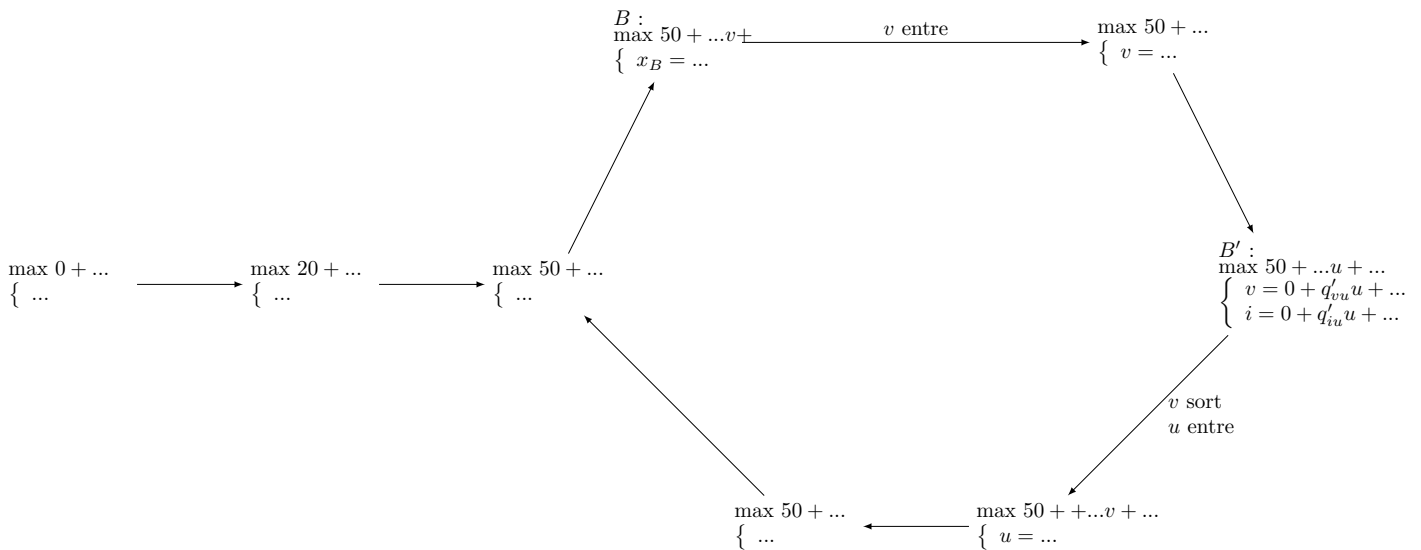
L'exécution commence du tableau initial puis on atteint un tableau sur lequel on boucle. Comme on boucle la valeur de l'objectif stagne le long du cycle.



Soit  $F$  les variables qui entrent (et aussi sortent, puisqu'on retombe sur la même base !) durant le cycle. Les variables correspondantes sont dites "folles" (mais appelées aussi "capricieuses" dans la littérature).

**Fait 19** Les variables "folles" sont nulles dans les solutions basiques du cycle (sinon l'objectif augmenterait strictement !).

Soit  $v$  la variable dans  $F$  d'indice le plus grand. On considère maintenant une base  $B$  du cycle où le pivotage qui suit fait entrer  $v$  dans la base. On considère aussi une base  $B'$  du cycle où le pivotage qui suit fait sortir  $v$ , et qui fait entrer une certaine variable d'indice que l'on note  $u$ .



Afin de ne pas alourdir la démonstration, nous ferons des abus de langage. On dira : "en  $B$ " pour dire "dans le tableau correspondant à la base  $B$ ", "dans l'objectif de  $B$ " pour dire "dans la fonction objectif écrite comme  $z_0 + cx_N$  dans le tableau de la base  $B$ , où  $N = \{1, \dots, n\} \setminus B$ ".

Commençons par deux faits qui découlent de l'utilisation de la règle de Bland.

**Fait 20** Dans l'objectif de  $B$ , les coefficients devant les variables de  $F \cap N \setminus \{v\}$  sont  $\leq 0$ , le coefficient devant  $v$  est strictement positif.

**DÉMONSTRATION.** La variable  $v$  entre dans la base, donc son coefficient dans l'objectif est strictement positif. La variable  $v$  a été sélectionnée en  $B$  pour entrer dans la base par la règle de Bland. C'est donc la variable d'indice le plus petit. Mais comme parmi  $F$ ,  $v$  est la variable d'indice le plus grand, cela signifie que les autres variables de  $F$  ne sont pas candidates à entrer dans la base. Autrement dit, les coefficients devant ces dernières sont négatifs ou nuls. ■

**Fait 21** En  $B'$ ,  $q'_{vu} < 0$  et  $q'_{iu} \geq 0$  pour tout  $i \in F \cup B' \setminus \{v\}$ .

**DÉMONSTRATION.** Le raisonnement est similaire mais concerne le choix de  $v$  à faire sortir de la base  $B'$ . Primo,  $v$  est candidate pour sortir et donc  $q'_{vu} < 0$  (on simule une "croissance" de 0 de  $u$  qui entre dans la base). Deuxio, la variable  $v$  est le seul candidat sortant dans  $F$  (s'il y en avait eu une autre, il aurait été d'indice plus petit et aurait donc été sélectionné, blablabla...). Ainsi, pour  $i \in F \cup B'$ ,  $i$  n'était pas candidat, on a  $q'_{iu} \geq 0$ . ■

Considérons maintenant le programme linéaire auxiliaire suivant :

$$\begin{cases} \text{maximiser } c^t x \\ Ax = b \\ x_{F \setminus \{v\}} \geq 0 \\ x_v \leq 0 \\ x_{N \setminus F} = 0 \end{cases}$$

où  $N = \{1, \dots, n\} \setminus B$ .

Il s'agit du programme équationnel initial, mais avec les trois contraintes supplémentaires

$$\begin{aligned} x_{F \setminus \{v\}} &\geq 0 \\ x_v &\leq 0 \\ x_{N \setminus F} &= 0 \end{aligned}$$

On montre deux faits contradictoires : que ce programme linéaire admet un optimum, mais qu'il est aussi non borné (on est capable de rendre l'objectif aussi grand que l'on souhaite). Ainsi, on aboutit à une contradiction. Ce qui montre que l'algorithme du simplexe termine. Terminons sur ces deux faits contradictoires.

**Fait 22** Le programme linéaire auxiliaire admet un optimum.

DÉMONSTRATION. Soit  $\tilde{x}$  la solution basique en  $B$ . Montrons que  $\tilde{x}$  est un optimum. Montrons d'abord que  $c$  est une solution du programme auxiliaire. Tout d'abord,  $c$  est une solution du programmation équationnel donc on a  $Ax = b$ . Maintenant, il suffit de vérifier que les trois contraintes supplémentaires sont vraies :

$x_{F \setminus \{v\}} \geq 0$  : ok car ces variables sont nulles (cf. Fait 19)

$x_v \leq 0$  : Comme  $v \in F$ , pareil par le fait 19,  $x_v = 0$

$x_{N \setminus F} = 0$  : car  $\tilde{x}_N = 0$  car  $\tilde{x}$  est basique en  $B$ .

Il reste à montrer que  $\tilde{x}$  est une solution optimale. Regardons l'objectif comme écrit dans le tableau de base  $B$  :  $z_0 + r^t x_N$ . Comme il est contraint que  $x_{N \setminus F} = 0$ , on ne peut pas modifier ces variables : elles sont nulles à tout jamais. Donc les coefficients dans  $r$  correspondant à  $N \setminus F$  ne sont pas importants. Regardons maintenant les coefficients correspondants à  $N \cap F$ . Comme les variables dans  $F$  sont capricieuses, on a  $\tilde{x}_F = 0$ .

D'après le fait 20, les coefficients devant  $F \setminus \{v\}$  sont négatifs et  $x_{F \setminus \{v\}} \geq 0$ , et le coefficient devant  $v$  est positif mais  $x_v \leq 0$ . Donc quelque soit la façon de changer les valeurs de  $\tilde{x}$ , la valeur objectif diminue. Le tableau suivant résume la situation.

Variables	Contraintes	Coefficients	Possibilité d'augmenter l'objectif
$x_{F \setminus \{v\}}$	$x_{F \setminus \{v\}} \geq 0$	négatif	non
$x_v$	$x_v \leq 0$	positif	non
$x_{N \setminus F}$	$x_{N \setminus F} = 0$	?	non

Ainsi,  $\tilde{x}$  est une solution optimale du programme linéaire auxiliaire.

■

**Fait 23** L'objectif du programme linéaire auxiliaire est non borné.

DÉMONSTRATION. Considérons  $\tilde{x}$  solution basique en  $B$ . Comme  $B$  et  $B'$  sont dans le cycle, ils ne diffèrent que par des variables de  $F$  qui sont rentrées ou sorties par définition de  $F$ . Pareil,  $N$  et  $N'$  ne diffèrent que par des variables de  $F$ . Par le fait 19, on a  $\tilde{x}_{N'} = 0$  et  $\tilde{x}$  est en fait aussi solution basique en  $B'$ .

Définissons la solution potentielle suivante, paramétrée par  $t \geq 0$  :

$$\begin{cases} \tilde{x}(t)_u = t \\ \tilde{x}(t)_i = 0 \text{ pour tout } i \in N' \setminus \{u\} \\ \tilde{x}(t)_{B'} = p' + Q' \tilde{x}(t)_{N'} \end{cases}$$

Autrement dit, toutes les coordonnées de  $\tilde{x}(t)_{N'}$  sont nulles, sauf celles de  $u$  qui vaut  $t$ . Puis,  $\tilde{x}(t)_{B'}$  est définie par le système d'équation en  $B'$  :  $x_{B'} = p' + Q' x_{N'}$ .

Montrons que  $\tilde{x}(t)$  est dans l'espace de solutions du programme auxiliaire.

$Ax = b$  Tout d'abord,  $A\tilde{x}(t) = b$  comme conséquence de  $x_{B'} = p' + Q' x_{N'}$ .

$x_{N \setminus F} = 0$  On a  $N' \setminus F = N \setminus F$ , par définition  $F$ , car seulement les variables de  $F$  se promènent. Donc  $\tilde{x}(t)_{N \setminus F} = \tilde{x}(t)_{N' \setminus F} = 0$ .

$x_v \leq 0$ . Comme  $v$  est une variable capricieuse,  $\tilde{x}_v = 0$ . Comme  $v$  sort, d'après le fait 21,  $q'_{vu} < 0$ .  $\tilde{x}(t)_v = \tilde{x}_v + tq'_{vu} < 0$ .

$x_{F \setminus \{v\}} \geq 0$ . Pour tout  $i \in F \setminus \{v\}$ , on a aussi  $\tilde{x}_i = 0$  car  $i$  est capricieuse. Si  $i \in F \cap N'$ , alors  $\tilde{x}(t)_i = 0$ . Sinon, si  $i \in B' \cap F \setminus \{v\}$ , par le fait 21,  $q'_{iu} \geq 0$ . Et donc  $\tilde{x}(t)_i = \tilde{x}_i + tq'_{iu} \geq 0$ .

Montrons maintenant que la fonction objectif n'est pas bornée. Comme  $u$  est candidate pour entrer dans la base  $B'$ , le coefficient devant  $x_u$  dans la fonction objectif du tableau  $B'$  est strictement positif et donc :

$$c^t \tilde{x}(t) = z'_0 + t \times \text{ce coefficient positif} \xrightarrow{t \rightarrow +\infty} +\infty.$$

La fonction objectif n'est pas bornée. ■

■

**Remarque 24** En pratique, la règle de Bland est lente. On préfère perturber un peu  $b$  pour supprimer les dégénérescences. ([DPV08], p. 218)

## 4 Prétraitement : obtenir tableau avec solution basique

entrée : un programme canonique  
 sortie : impossible ou un tableau avec solution basique de même maximum

**fonction** prétraitement  $\left( L : \begin{cases} \text{maximiser } c^t x \\ Ax \leq b \\ x \geq 0 \end{cases} \right)$

**si**  $b \geq 0$  **alors**  
   | **renvoyer**  $L$

  soit le programme  $L_{aux} : \begin{cases} \text{maximiser } -x_0 \\ Ax - x_0 \leq b \\ x \geq 0 \end{cases}$  où  $x_0$  est une variable auxiliaire

    //  $L_{aux}$  admet des solutions  
 // l'espace des solutions de  $L$  est non vide ssi l'optimum de  $L_{aux}$  est 0.

  1. mettre  $L_{aux}$  sous forme équationnelle, puis sous tableau  
   2. faire entrer  $x_0$  dans la base, et sortir la variable  $x_k$  où  $b_k$  minimum

    // le tableau courant admet une solution basique

  3. Lancer l'algorithme du simplexe sur ce tableau

**si** le maximum est 0 **alors**  
   | **renvoyer** le tableau obtenu depuis celui à la fin de 4., en réécrivant la fonction objectif de  $L$   
   | avec les variables hors bases et en enlever  $x_0$

**sinon**  
   | **lever une exception impossible**

**Proposition 25** Le programme  $L_{aux}$  admet une solution.

DÉMONSTRATION. Prendre  $x = 0$  et  $x_0 = -\min_i b_i$ . ■

**Proposition 26** L'espace des solutions de  $L$  est non vide ssi l'optimum de  $L_{aux}$  est 0.

DÉMONSTRATION. Si l'espace est non vide, alors  $Ax \leq b$  admet une solution. Mais alors on l'étend en prenant  $x_0 = 0$ . Et cela montre que l'optimum de  $L_{aux}$  est 0.

Réciproquement, si l'optimum de  $L_{aux}$  est 0, alors  $Ax \leq b$ . ■

**Proposition 27** Après l'étape 2, le tableau admet une solution basique.

DÉMONSTRATION. On note  $x_N = (x_1, \dots, x_d)$  les variables présentes dans  $L_{aux}$  sauf  $x_0$ . On note  $x_{d+1}, \dots, x_{d+m}$  les variables d'écart introduites.

Le tableau à l'étape 2 est :

$$\begin{cases} \text{maximiser } -x_0 \\ x_B = b + x_0 - Ax_N \\ x \geq 0 \end{cases}$$

La ligne du pivot est  $x_{d+k} = b_{d+k} - (Ax_N)_{d+k} + x_0$ . La nouvelle base contient les même variables sauf  $x_{d+k}$ , et avec  $x_0$  en plus :  $B' := B \cup \{0\} \setminus \{d+k\}$ . Les variables hors bases (considérées comme nulles) sont  $x_N$  et  $x_k$ . La ligne du pivot devient :  $x_0 = -b_k + (Ax_N)_k - x_{d+k}$ . La valeur de  $x_0$  est bien positive dans la solution basique. Pour les variable  $x_i$  avec  $i$  dans  $B'$ , la ligne devient

$$x_b = b_b - b_k + (Ax_N)_k - x_{d+k}$$

Et donc la valeur de  $x_b$  dans la solution basique est  $x_b \leq 0$ .

On obtient donc :

$$\begin{cases} \text{maximiser } -x_k + b_k - (Ax_N)_k \\ x_0 = b_k - (Ax_N)_k \\ x_B = b - Ax_N + x_0 \\ x \geq 0 \end{cases}$$



■ **Exemple 28** Le programme canonique suivant n'admet pas  $(0, 0)$  comme solution :

$$\begin{cases} \text{maximiser } 2x_1 - x_2 \\ 2x_1 - x_2 \leq 2 \\ x_1 - 5x_2 \leq -4 \\ x_1, x_2 \geq 0 \end{cases}$$

1. Programme auxiliaire.

$$\begin{cases} \text{maximiser } -x_0 \\ 2x_1 - x_2 - x_0 \leq 2 \\ x_1 - 5x_2 - x_0 \leq -4 \\ x_1, x_2, x_0 \geq 0 \end{cases}$$

2. Tableau du programme auxiliaire

$$\begin{cases} \text{maximiser } -x_0 \\ x_3 = 2 - 2x_1 + x_2 + x_0 \\ x_4 = -4 - x_1 + 5x_2 + x_0 \\ x_1, x_2, x_0, x_3, x_4 \geq 0 \end{cases}$$

3. Pivot. La variable de valeur la plus petite  $(-4)$  est  $x_4$ .

$$\begin{cases} \text{maximiser } -x_0 \\ x_3 = 2 - 2x_1 + x_2 + x_0 \\ x_4 = -4 - x_1 + 5x_2 + x_0 \\ x_1, x_2, x_0, x_3, x_4 \geq 0 \end{cases} \xrightarrow{x_4 := 0 \quad x_0 \nearrow} \begin{cases} \text{maximiser } -4 - x_1 + 5x_2 + x_4 \\ x_3 = 6 - x_1 - 4x_2 + x_4 \\ x_0 = 4 + x_1 - 5x_2 - x_4 \\ x_1, x_2, x_0, x_3, x_4 \geq 0 \end{cases}$$

4. Exécution de l'algorithme du simplexe. Le tableau courant admet une solution basique. On lance l'algo du simplexe pour voir si le  $\max = 0$ . Si ce n'est pas le cas, on répond 'l'espace des solutions de  $L$  est vide'.

$$\begin{cases} \text{maximiser } -4 - x_1 + 5x_2 + x_4 \\ x_3 = 6 - x_1 - 4x_2 + x_4 \\ x_0 = 4 + x_1 - 5x_2 - x_4 \\ x_1, x_2, x_0, x_3, x_4 \geq 0 \end{cases} \xrightarrow{x_0 := 0 \quad x_3 \nearrow} \begin{cases} \text{maximiser } -x_0 \\ x_3 = \frac{14}{5} + \frac{4x_0}{5} - \frac{9x_1}{5} + \frac{x_4}{5} \\ x_2 = \frac{4}{5} - \frac{x_0}{5} + \frac{x_1}{5} + \frac{x_4}{5} \\ x_1, x_2, x_0, x_3, x_4 \geq 0 \end{cases}$$

L'algo du simplexe termine et le  $\max$  vaut bien 0. L'espace des solutions de  $L$  est non vide.

5. Réécriture de la fonction objectif. On remet l'objectif initial, que l'on réécrit avec les variables "nulles" (hors base), puis on enlève  $x_0$  qui est nulle :

$$\begin{cases} \text{maximiser } 2x_1 - x_2 \\ x_3 = \frac{14}{5} + \frac{4x_0}{5} - \frac{9x_1}{5} + \frac{x_4}{5} \\ x_2 = \frac{4}{5} - \frac{x_0}{5} + \frac{x_1}{5} + \frac{x_4}{5} \\ x_1, x_2, x_0, x_3, x_4 \geq 0 \end{cases} \xrightarrow{\text{maximiser } -\frac{4}{5} + \frac{x_0}{5} + \frac{9x_1}{5} - \frac{x_4}{5}} \begin{cases} \text{maximiser } -\frac{4}{5} + \frac{x_0}{5} + \frac{9x_1}{5} - \frac{x_4}{5} \\ x_3 = \frac{14}{5} + \frac{4x_0}{5} - \frac{9x_1}{5} + \frac{x_4}{5} \\ x_2 = \frac{4}{5} - \frac{x_0}{5} + \frac{x_1}{5} + \frac{x_4}{5} \\ x_1, x_2, x_0, x_3, x_4 \geq 0 \end{cases}$$

$$\begin{cases} \text{maximiser } -\frac{4}{5} + \frac{9x_1}{5} - \frac{x_4}{5} \\ x_3 = \frac{14}{5} - \frac{9x_1}{5} + \frac{x_4}{5} \\ x_2 = \frac{4}{5} + \frac{x_1}{5} + \frac{x_4}{5} \\ x_1, x_2, x_3, x_4 \geq 0 \end{cases}$$

## Notes bibliographiques

L'exemple initial provient de [DPV08]. Ce livre vulgarise très bien la programmation linéaire. Malheureusement, il ne donne aucune démonstration.

Dans [CLRS09], un programme linéaire équationnel s'appelle un programme standard (mais l'adjectif standard n'a pas trop de sens). Ici, nous empruntons La terminologie de [GM07] : "programme sous forme équationnelle" que nous abrégons en "programme équationnel". [CLRS09] donne des démonstrations mais parfois les justifications sont lourdes.

Comme mentionné dans [GM07], le plus petit exemple de programme linéaire où le simplexe boucle est donné dans Chvátal's textbook cited in Chapter 9 [CC<sup>+</sup>83].

Le prétraitement pour avoir un tableau avec solution basique vient de (cf. [CLRS09], p. 886, 29.5). J'ai fait un choix d'écrire explicitement les programmes linéaires et d'éviter d'écrire des appels cryptiques comme  $\text{pivot}(N, B, A, b, c, v, \ell, 0)$ , cf. [CLRS09]!

Les démonstrations dans [GM07] sont pas mal, mais parfois un peu lourdes concernant les indices dans les matrices. Ici, nous avons fait le choix d'être flexible sur les indices une matrice : les indices d'une matrice de 4 lignes peuvent être 1, 3, 5, 6, alors que dans [GM07] (p. 68) les indices sont toujours 1, 2, 3, 4 ce qui force d'utiliser des double-indices  $\ell_1, \ell_2, \ell_3, \ell_4$ .

## Références

[CC<sup>+</sup>83] Vasek Chvatal, Vaclav Chvatal, et al. *Linear programming*. Macmillan, 1983.

- [CLRS09] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, 3rd Edition*. MIT Press, 2009.
- [DPV08] Sanjoy Dasgupta, Christos H. Papadimitriou, and Umesh V. Vazirani. *Algorithms*. McGraw-Hill, 2008.
- [GM07] Bernd Gärtner and Jirí Matousek. *Understanding and using linear programming*. Universitext. Springer, 2007.