

Chapitre 2 : Méthode du Simplexe

J.-F. Scheid

- ① Introduction
- ② Progression de l'algorithme du simplexe (phase 2)
- ③ Méthode des dictionnaires
- ④ Finitude du simplexe
- ⑤ Initialisation du simplexe (phase 1)
- ⑥ Complexité
- ⑦ Quelques solveurs de PL

I. Introduction

On a vu que pour résoudre un PL, il suffit de se restreindre aux solutions de bases réalisables.

Méthode du simplexe due à G. Dantzig (1947).



G. Dantzig, 1914-2005

Les deux phases de la méthode du simplexe :

- ① **Phase 1 – Initialisation** : Trouver une solution de base réalisable (ou bien détecter l'impossibilité).
- ② **Phase 2 – Progression** : On passe d'un sommet à un sommet voisin *pour augmenter la fonction objectif*

Remarque : On appelle n -simplexe ou simplement **simplexe**, l'enveloppe convexe d'un ensemble de $n + 1$ points ($n = 1$: un segment, $n = 2$: un triangle, $n = 3$: un tétraèdre)

II. L'algorithme du simplexe proprement dit : Phase 2

PL sous forme standard

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

On dispose d'une base B et d'une solution de base réalisable $\underline{\mathbf{x}}$ avec (à une permutation près des colonnes de A)

$$A = (A_B \mid A_H) \quad \text{et} \quad \underline{\mathbf{x}} = \begin{pmatrix} \underline{\mathbf{x}}_B \\ \underline{\mathbf{x}}_H \end{pmatrix}$$

où A_B matrice $m \times m$, **inversible** (variables de base)

A_H matrice $m \times (n - m)$ (variables hors-base)

II. L'algorithme du simplexe proprement dit : Phase 2

PL sous forme standard

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

On dispose d'une base B et d'une solution de base réalisable $\underline{\mathbf{x}}$ avec (à une permutation près des colonnes de A)

$$A = (A_B \mid A_H) \quad \text{et} \quad \underline{\mathbf{x}} = \begin{pmatrix} \underline{\mathbf{x}}_B \\ \underline{\mathbf{x}}_H \end{pmatrix}$$

où A_B matrice $m \times m$, **inversible** (variables de base)

A_H matrice $m \times (n - m)$ (variables hors-base)

But : on veut trouver une autre base B^* et une solution de base réalisable $\underline{\mathbf{x}}^*$ telles que $\underline{\mathbf{x}}^*$ est meilleur que $\underline{\mathbf{x}}$ c-à-d

$$F(\underline{\mathbf{x}}^*) > F(\underline{\mathbf{x}})$$

Principe de la méthode du simplexe : faire rentrer une variable hors-base dans la nouvelle base (*variable entrante*) et faire sortir à la place une variable de base (*variable sortante*).

Principe de la méthode du simplexe : faire rentrer une variable hors-base dans la nouvelle base (*variable entrante*) et faire sortir à la place une variable de base (*variable sortante*).

1) Variable entrante - calcul des coûts réduits

Fonction objectif F exprimée en fonction des **variables hors-base**.

Ensemble des solutions réalisables $\mathcal{D}_R = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$.

Proposition (Coûts réduits)

Pour tout $\mathbf{x} \in \mathcal{D}_R$, on a

$$F(\mathbf{x}) = F(\underline{\mathbf{x}}) + \mathbf{L}_H^\top \mathbf{x}_H$$

où

$$\mathbf{L}_H^\top = \mathbf{c}_H^\top - \mathbf{c}_B^\top A_B^{-1} A_H$$

est le vecteur des *coûts réduits*.

Démonstration.

On a $\mathbf{b} = A\mathbf{x} = A_B\mathbf{x}_B + A_H\mathbf{x}_H$ avec A_B inversible donc $\mathbf{x}_B = A_B^{-1}(\mathbf{b} - A_H\mathbf{x}_H)$. On obtient donc

$$\begin{aligned} F(\mathbf{x}) &= \mathbf{c}^\top \mathbf{x} = \mathbf{c}_B^\top \mathbf{x}_B + \mathbf{c}_H^\top \mathbf{x}_H \text{ avec } \mathbf{c} = \begin{pmatrix} \mathbf{c}_B \\ \mathbf{c}_H \end{pmatrix} \\ &= \mathbf{c}_B^\top A_B^{-1}(\mathbf{b} - A_H\mathbf{x}_H) + \mathbf{c}_H^\top \mathbf{x}_H \\ &= \mathbf{c}_B^\top A_B^{-1} \mathbf{b} + (\mathbf{c}_H^\top - \mathbf{c}_B^\top A_B^{-1} A_H) \mathbf{x}_H \end{aligned}$$

Or $\mathbf{x}_B = A_B^{-1} \mathbf{b}$ (car $\mathbf{x}_H = \mathbf{0}$) et $\mathbf{c}_B^\top A_B^{-1} \mathbf{b} = \mathbf{c}^\top \mathbf{x} = F(\mathbf{x})$ donc

$$F(\mathbf{x}) = F(\mathbf{x}) + (\mathbf{c}_H^\top - \mathbf{c}_B^\top A_B^{-1} A_H) \mathbf{x}_H.$$

□

Variable entrante

- Si les coûts réduits sont tous négatifs i.e. $\mathbf{L}_H^T \leq \mathbf{0}$, il n'est alors pas possible d'augmenter la fonction objectif F : l'algorithme se termine *normalement* c'est-à-dire qu'on a trouvé une solution de base réalisable \mathbf{x} optimale.

Variable entrante

- Si les coûts réduits sont tous négatifs i.e. $\mathbf{L}_H^\top \leq \mathbf{0}$, il n'est alors pas possible d'augmenter la fonction objectif F : l'algorithme se termine *normalement* c'est-à-dire qu'on a trouvé une solution de base réalisable \mathbf{x} optimale.
- Dans le cas contraire (i.e. $\exists(\mathbf{L}_H)_i > 0$), on a intérêt à faire entrer dans la base, la variable hors-base qui a le coût réduit positif le plus grand possible.

Variable entrante

- Si les coûts réduits sont tous négatifs i.e. $\mathbf{L}_H^T \leq \mathbf{0}$, il n'est alors pas possible d'augmenter la fonction objectif F : l'algorithme se termine *normalement* c'est-à-dire qu'on a trouvé une solution de base réalisable \mathbf{x} optimale.
- Dans le cas contraire (i.e. $\exists (\mathbf{L}_H)_i > 0$), on a intérêt à faire entrer dans la base, la variable hors-base qui a le coût réduit positif le plus grand possible.

On note $e \notin B$ l'indice de la *variable entrante*. On choisit e tel que

$$(\mathbf{L}_H)_e = \max_j \{ (\mathbf{L}_H)_j, (\mathbf{L}_H)_j > 0 \}$$

ce qu'on note par

$$e = \operatorname{argmax}_j \{ (\mathbf{L}_H)_j, (\mathbf{L}_H)_j > 0 \}$$

Remarque. Si on traite d'un problème de minimisation c'est-à-dire avec

$$\min F(\mathbf{x}),$$

alors la variable entrante x_e est déterminée par l'indice

$$e = \operatorname{argmin}_j \left\{ (\mathbf{L}_H)_j, (\mathbf{L}_H)_j < 0 \right\}$$

2) Variable sortante

Une fois l'indice e choisi, il faut déterminer la variable qui doit quitter la base. En maintenant la relation $A\mathbf{x} = \mathbf{b}$ avec $\mathbf{x} \geq \mathbf{0}$, on augmente la variable entrante x_e jusqu'à annuler une des variables de base. Cette variable sera alors la *variable sortante*.

2) Variable sortante

Une fois l'indice e choisi, il faut déterminer la variable qui doit quitter la base. En maintenant la relation $A\mathbf{x} = \mathbf{b}$ avec $\mathbf{x} \geq \mathbf{0}$, on augmente la variable entrante x_e jusqu'à annuler une des variables de base. Cette variable sera alors la *variable sortante*.

$$A\mathbf{x} = \mathbf{b} \Leftrightarrow A_B\mathbf{x}_B + A^e x_e = \mathbf{b} \quad \text{où } A^e \text{ désigne la } e\text{-ième colonne de } A$$

2) Variable sortante

Une fois l'indice e choisi, il faut déterminer la variable qui doit quitter la base. En maintenant la relation $A\mathbf{x} = \mathbf{b}$ avec $\mathbf{x} \geq \mathbf{0}$, on augmente la variable entrante x_e jusqu'à annuler une des variables de base. Cette variable sera alors la *variable sortante*.

$$\begin{aligned} A\mathbf{x} = \mathbf{b} &\Leftrightarrow A_B\mathbf{x}_B + A^e x_e = \mathbf{b} \quad \text{où } A^e \text{ désigne la } e\text{-ième colonne de } A \\ &\Leftrightarrow \mathbf{x}_B = A_B^{-1}(\mathbf{b} - A^e x_e) \end{aligned}$$

2) Variable sortante

Une fois l'indice e choisi, il faut déterminer la variable qui doit quitter la base. En maintenant la relation $A\mathbf{x} = \mathbf{b}$ avec $\mathbf{x} \geq \mathbf{0}$, on augmente la variable entrante x_e jusqu'à annuler une des variables de base. Cette variable sera alors la *variable sortante*.

$$A\mathbf{x} = \mathbf{b} \Leftrightarrow A_B\mathbf{x}_B + A^e x_e = \mathbf{b} \quad \text{où } A^e \text{ désigne la } e\text{-ième colonne de } A$$

$$\Leftrightarrow \mathbf{x}_B = A_B^{-1}(\mathbf{b} - A^e x_e)$$

$$\Leftrightarrow \mathbf{x}_B = \underline{\mathbf{x}}_B - A_B^{-1}A^e x_e$$

2) Variable sortante

Une fois l'indice e choisi, il faut déterminer la variable qui doit quitter la base. En maintenant la relation $A\mathbf{x} = \mathbf{b}$ avec $\mathbf{x} \geq \mathbf{0}$, on augmente la variable entrante x_e jusqu'à annuler une des variables de base. Cette variable sera alors la *variable sortante*.

$$A\mathbf{x} = \mathbf{b} \Leftrightarrow A_B\mathbf{x}_B + A^e x_e = \mathbf{b} \quad \text{où } A^e \text{ désigne la } e\text{-ième colonne de } A$$

$$\Leftrightarrow \mathbf{x}_B = A_B^{-1}(\mathbf{b} - A^e x_e)$$

$$\Leftrightarrow \mathbf{x}_B = \underline{\mathbf{x}}_B - A_B^{-1}A^e x_e$$

$$\Leftrightarrow \mathbf{x}_B = \underline{\mathbf{x}}_B - \mathbf{z}x_e$$

avec

$$\mathbf{z} = A_B^{-1}A^e \in \mathbb{R}^m.$$

On doit avoir : $\mathbf{x}_B = \underline{\mathbf{x}}_B - \mathbf{z}x_e \geq \mathbf{0}$

On doit avoir : $\mathbf{x}_B = \underline{\mathbf{x}}_B - \mathbf{z}x_e \geq \mathbf{0}$

- Si $\mathbf{z} \leq \mathbf{0}$, on peut augmenter x_e autant qu'on veut, on aura toujours la positivité de la variable de base \mathbf{x}_B . La fonction objectif n'est pas majorée sur \mathcal{D}_R ($\max F = +\infty$) \Rightarrow arrêt de l'algorithme.

On doit avoir : $\mathbf{x}_B = \underline{\mathbf{x}}_B - \mathbf{z}x_e \geq \mathbf{0}$

- Si $\mathbf{z} \leq \mathbf{0}$, on peut augmenter x_e autant qu'on veut, on aura toujours la positivité de la variable de base \mathbf{x}_B . La fonction objectif n'est pas majorée sur \mathcal{D}_R ($\max F = +\infty$) \Rightarrow arrêt de l'algorithme.
- Sinon (i.e. il existe $z_i > 0$), pour avoir la positivité $(\underline{\mathbf{x}}_B)_i - z_i x_e \geq 0$ pour tout i , on choisit la **variable sortante** x_s pour laquelle le rapport $(\underline{\mathbf{x}}_B)_i / z_i$ pour $i = 1, \dots, m$ avec $z_i > 0$, est **le plus petit possible** :

On doit avoir : $\mathbf{x}_B = \underline{\mathbf{x}}_B - \mathbf{z}x_e \geq \mathbf{0}$

- Si $\mathbf{z} \leq \mathbf{0}$, on peut augmenter x_e autant qu'on veut, on aura toujours la positivité de la variable de base \mathbf{x}_B . La fonction objectif n'est pas majorée sur \mathcal{D}_R ($\max F = +\infty$) \Rightarrow arrêt de l'algorithme.
- Sinon (i.e. il existe $z_i > 0$), pour avoir la positivité $(\underline{\mathbf{x}}_B)_i - z_i x_e \geq 0$ pour tout i , on choisit la **variable sortante** x_s pour laquelle le rapport $(\underline{\mathbf{x}}_B)_i / z_i$ pour $i = 1, \dots, m$ avec $z_i > 0$, est **le plus petit possible** :

Variable sortante (indice) :

$$s = \operatorname{argmin}_i \left\{ \frac{(\underline{\mathbf{x}}_B)_i}{z_i}, z_i > 0 \right\}$$

On a, dans ce cas, $x_s = 0$ et $\mathbf{x}_B \geq \mathbf{0}$.

Remarque. La valeur de la variable entrante est donnée par

$$x_e = \min_i \left\{ \frac{(\underline{x}_B)_i}{z_i}, z_i > 0 \right\}$$

Méthode du simplexe en phase 2 (progression)

- 1
 - *Calcul des variables de base réalisables :*
Etant donné $A = (A_B \mid A_H)$, on calcule $\underline{\mathbf{x}}_B = A_B^{-1} \mathbf{b} \geq \mathbf{0}$.
 - *Calcul des coûts réduits :*
$$\mathbf{L}_H^\top = \mathbf{c}_H^\top - \mathbf{c}_B^\top A_B^{-1} A_H \quad (F(\mathbf{x}) = F(\underline{\mathbf{x}}) + \mathbf{L}_H^\top \mathbf{x}_H)$$
 - Si $\mathbf{L}_H \leq \mathbf{0}$ alors $\underline{\mathbf{x}}_B$ est une solution optimale (\rightarrow arrêt de l'algo.).

Méthode du simplexe en phase 2 (progression)

- 1
 - *Calcul des variables de base réalisables :*
Etant donné $A = (A_B \mid A_H)$, on calcule $\underline{\mathbf{x}}_B = A_B^{-1} \mathbf{b} \geq \mathbf{0}$.
 - *Calcul des coûts réduits :*
$$\mathbf{L}_H^\top = \mathbf{c}_H^\top - \mathbf{c}_B^\top A_B^{-1} A_H \quad (F(\mathbf{x}) = F(\underline{\mathbf{x}}) + \mathbf{L}_H^\top \mathbf{x}_H)$$
 - Si $\mathbf{L}_H \leq \mathbf{0}$ alors $\underline{\mathbf{x}}_B$ est une solution optimale (\rightarrow arrêt de l'algo.).
- 2 *variable entrante* : $e = \operatorname{argmax}_j \{ (\mathbf{L}_H)_j, (\mathbf{L}_H)_j > 0 \}$

Méthode du simplexe en phase 2 (progression)

- 1
 - *Calcul des variables de base réalisables* :
Etant donné $A = (A_B \mid A_H)$, on calcule $\underline{\mathbf{x}}_B = A_B^{-1} \mathbf{b} \geq \mathbf{0}$.
 - *Calcul des coûts réduits* :
$$\mathbf{L}_H^\top = \mathbf{c}_H^\top - \mathbf{c}_B^\top A_B^{-1} A_H \quad (F(\mathbf{x}) = F(\underline{\mathbf{x}}) + \mathbf{L}_H^\top \mathbf{x}_H)$$
 - Si $\mathbf{L}_H \leq \mathbf{0}$ alors $\underline{\mathbf{x}}_B$ est une solution optimale (\rightarrow arrêt de l'algo.).
- 2 *variable entrante* : $e = \operatorname{argmax}_j \{ (\mathbf{L}_H)_j, (\mathbf{L}_H)_j > 0 \}$
- 3 *variable sortante* :
 - Calcul de $\mathbf{z} = A_B^{-1} A^e$ puis
 - $s = \operatorname{argmin}_i \left\{ \frac{(\underline{\mathbf{x}}_B)_i}{z_i}, z_i > 0 \right\}$.

Méthode du simplexe en phase 2 (progression)

- *Calcul des variables de base réalisables* :
Etant donné $A = (A_B \mid A_H)$, on calcule $\underline{\mathbf{x}}_B = A_B^{-1} \mathbf{b} \geq \mathbf{0}$.
 - *Calcul des coûts réduits* :
$$\mathbf{L}_H^\top = \mathbf{c}_H^\top - \mathbf{c}_B^\top A_B^{-1} A_H \quad (F(\mathbf{x}) = F(\underline{\mathbf{x}}) + \mathbf{L}_H^\top \mathbf{x}_H)$$
 - Si $\mathbf{L}_H \leq \mathbf{0}$ alors $\underline{\mathbf{x}}_B$ est une solution optimale (\rightarrow arrêt de l'algo.).
- *variable entrante* : $e = \operatorname{argmax}_j \{ (\mathbf{L}_H)_j, (\mathbf{L}_H)_j > 0 \}$
- *variable sortante* :
 - Calcul de $\mathbf{z} = A_B^{-1} A^e$ puis
 - $s = \operatorname{argmin}_i \left\{ \frac{(\underline{\mathbf{x}}_B)_i}{z_i}, z_i > 0 \right\}$.
- On obtient une nouvelle base \tilde{B} et une nouvelle matrice $A_{\tilde{B}}$ dans laquelle la colonne A^e remplace la colonne A^s . Calcul de $A_{\tilde{B}}^{-1}$ et retour en 1.

III. Méthode des dictionnaires

PL sous forme standard $\max_{\mathbf{x}} F(\mathbf{x}) = \mathbf{c}^\top \mathbf{x}$

$$\begin{cases} A\mathbf{x} = \mathbf{b} \\ \mathbf{x} \geq \mathbf{0} \end{cases}$$

Principe: on exprime les variables de base \mathbf{x}_B ainsi que F en fonction des variables hors-base \mathbf{x}_H . On obtient un système linéaire qu'on appelle **dictionnaire**.

III. Méthode des dictionnaires

PL sous forme standard $\max_{\mathbf{x}} F(\mathbf{x}) = \mathbf{c}^\top \mathbf{x}$

$$\begin{cases} A\mathbf{x} = \mathbf{b} \\ \mathbf{x} \geq \mathbf{0} \end{cases}$$

Principe: on exprime les variables de base \mathbf{x}_B ainsi que F en fonction des variables hors-base \mathbf{x}_H . On obtient un système linéaire qu'on appelle **dictionnaire**.

Exemple du problème de production.

Forme standard (variables d'écart e_1, e_2, e_3)

$$\begin{aligned} \max F(x_1, x_2) &= 6x_1 + 4x_2 \\ \begin{cases} 3x_1 + 9x_2 + e_1 = 81 \\ 4x_1 + 5x_2 + e_2 = 55 \\ 2x_1 + x_2 + e_3 = 20 \\ x_1, x_2 \geq 0, e_1, e_2, e_3 \geq 0 \end{cases} \end{aligned}$$

★ Etape 1.

Solution de base réalisable initiale :

$$x_1 = 0, x_2 = 0, e_1 = 81, e_2 = 55, e_3 = 20 \text{ avec } F = 0.$$

Dictionnaire: On exprime les variables de base e_1, e_2, e_3 en fonction des variables hors-base x_1, x_2 .

$e_1 = 81 - 3x_1 - 9x_2$
$e_2 = 55 - 4x_1 - 5x_2$
$e_3 = 20 - 2x_1 - x_2$
$F = 6x_1 + 4x_2$

★ Etape 1.

Solution de base réalisable initiale :

$$x_1 = 0, x_2 = 0, e_1 = 81, e_2 = 55, e_3 = 20 \text{ avec } F = 0.$$

Dictionnaire: On exprime les variables de base e_1, e_2, e_3 en fonction des variables hors-base x_1, x_2 .

$e_1 = 81 - 3x_1 - 9x_2$
$e_2 = 55 - 4x_1 - 5x_2$
$e_3 = 20 - 2x_1 - x_2$
$F = 6x_1 + 4x_2$

Variable entrante x_e : $\max_{>0}\{6, 4\} = 6 \Rightarrow x_e = x_1$.

★ Etape 1.

Solution de base réalisable initiale :

$$x_1 = 0, x_2 = 0, e_1 = 81, e_2 = 55, e_3 = 20 \text{ avec } F = 0.$$

Dictionnaire: On exprime les variables de base e_1, e_2, e_3 en fonction des variables hors-base x_1, x_2 .

$e_1 = 81 - 3x_1 - 9x_2$
$e_2 = 55 - 4x_1 - 5x_2$
$e_3 = 20 - 2x_1 - x_2$
$F = 6x_1 + 4x_2$

Variable entrante x_e : $\max_{>0}\{6, 4\} = 6 \Rightarrow x_e = x_1$.

Variable sortante x_s : on maintient $e_1 \geq 0, e_2 \geq 0, e_3 \geq 0$

$$\Rightarrow x_1 = \min_{>0}\left\{\frac{81}{3}, \frac{55}{4}, \frac{20}{2}\right\} = 10 \Rightarrow x_s = e_3.$$

★ Etape 1.

Solution de base réalisable initiale :

$$x_1 = 0, x_2 = 0, e_1 = 81, e_2 = 55, e_3 = 20 \text{ avec } F = 0.$$

Dictionnaire: On exprime les variables de base e_1, e_2, e_3 en fonction des variables hors-base x_1, x_2 .

$e_1 = 81 - 3x_1 - 9x_2$
$e_2 = 55 - 4x_1 - 5x_2$
$e_3 = 20 - 2x_1 - x_2$
$F = 6x_1 + 4x_2$

Variable entrante x_e : $\max_{>0}\{6, 4\} = 6 \Rightarrow x_e = x_1$.

Variable sortante x_s : on maintient $e_1 \geq 0, e_2 \geq 0, e_3 \geq 0$

$$\Rightarrow x_1 = \min_{>0}\left\{\frac{81}{3}, \frac{55}{4}, \frac{20}{2}\right\} = 10 \Rightarrow x_s = e_3.$$

Nouvelle Solution de base réalisable :

$$x_1 = 10, x_2 = 0, e_1 = 51, e_2 = 15, e_3 = 0 \text{ avec } F = 60.$$

★ Etape 2.

Dictionnaire: On exprime la nouvelle variable de base x_1 en fonction de x_2 et e_3 (nouvelle variable hors-base). On utilise la 3ème équation du dictionnaire de l'étape 1 et on substitue x_1 dans les autres relations.

$$x_1 = 10 - \frac{1}{2}x_2 - \frac{1}{2}e_3$$

$$e_1 = 81 - 3\left(10 - \frac{1}{2}x_2 - \frac{1}{2}e_3\right) - 9x_2$$

$$e_2 = 55 - 4\left(10 - \frac{1}{2}x_2 - \frac{1}{2}e_3\right) - 5x_2$$

$$F = 6\left(10 - \frac{1}{2}x_2 - \frac{1}{2}e_3\right) + 4x_2$$

On obtient ainsi le dictionnaire (étape 2)

$$x_1 = 10 - \frac{1}{2}x_2 - \frac{1}{2}e_3$$

$$e_1 = 51 - \frac{15}{2}x_2 + \frac{3}{2}e_3$$

$$e_2 = 15 - 3x_2 + 2e_3$$

$$F = 60 + x_2 - 3e_3$$

On obtient ainsi le dictionnaire (étape 2)

$$x_1 = 10 - \frac{1}{2}x_2 - \frac{1}{2}e_3$$

$$e_1 = 51 - \frac{15}{2}x_2 + \frac{3}{2}e_3$$

$$e_2 = 15 - 3x_2 + 2e_3$$

$$F = 60 + x_2 - 3e_3$$

Variable entrante x_e : $\max_{>0} \{1, -3\} = 1 \Rightarrow x_e = x_2$.

On obtient ainsi le dictionnaire (étape 2)

$$\begin{array}{l} x_1 = 10 - \frac{1}{2}x_2 - \frac{1}{2}e_3 \\ e_1 = 51 - \frac{15}{2}x_2 + \frac{3}{2}e_3 \\ e_2 = 15 - 3x_2 + 2e_3 \\ \hline F = 60 + x_2 - 3e_3 \end{array}$$

Variable entrante x_e : $\max_{>0} \{ \mathbf{1}, -3 \} = 1 \Rightarrow x_e = x_2$.

Variable sortante x_s : on maintient $x_1 \geq 0$, $e_1 \geq 0$, $e_2 \geq 0$

$$\Rightarrow x_2 = \min_{>0} \left\{ \frac{10}{1/2}, \frac{51}{15/2}, \frac{\mathbf{15}}{\mathbf{3}} \right\} = 5 \Rightarrow x_s = e_2.$$

On obtient ainsi le dictionnaire (étape 2)

$x_1 = 10 - \frac{1}{2}x_2 - \frac{1}{2}e_3$
$e_1 = 51 - \frac{15}{2}x_2 + \frac{3}{2}e_3$
$e_2 = 15 - 3x_2 + 2e_3$
$F = 60 + x_2 - 3e_3$

Variable entrante x_e : $\max_{>0} \{ \mathbf{1}, -3 \} = 1 \Rightarrow x_e = x_2$.

Variable sortante x_s : on maintient $x_1 \geq 0$, $e_1 \geq 0$, $e_2 \geq 0$

$$\Rightarrow x_2 = \min_{>0} \left\{ \frac{10}{1/2}, \frac{51}{15/2}, \frac{\mathbf{15}}{3} \right\} = 5 \Rightarrow x_s = e_2.$$

Nouvelle Solution de base réalisable (étape 2) :

$$x_1 = \frac{15}{2}, x_2 = 5, e_1 = \frac{27}{2}, e_2 = 0, e_3 = 0 \text{ avec } F = 65.$$

★ Etape 3.

Dictionnaire: On exprime la nouvelle variable de base x_2 en fonction des variables hors-base e_2 et e_3 . On utilise la 3ème équation du dictionnaire de l'étape 2 et on substitue x_2 dans les autres relations.

$$x_2 = 5 - \frac{1}{3}e_2 + \frac{2}{3}e_3$$

$$x_1 = \frac{15}{2} + \frac{1}{6}e_2 - \frac{5}{6}e_3$$

$$e_1 = \frac{27}{2} + \frac{5}{2}e_2 - \frac{7}{2}e_3$$

$$F = 65 - \frac{1}{3}e_2 - \frac{7}{3}e_3$$

★ Etape 3.

Dictionnaire: On exprime la nouvelle variable de base x_2 en fonction des variables hors-base e_2 et e_3 . On utilise la 3ème équation du dictionnaire de l'étape 2 et on substitue x_2 dans les autres relations.

$$\begin{array}{l} x_2 = 5 - \frac{1}{3}e_2 + \frac{2}{3}e_3 \\ x_1 = \frac{15}{2} + \frac{1}{6}e_2 - \frac{5}{6}e_3 \\ e_1 = \frac{27}{2} + \frac{5}{2}e_2 - \frac{7}{2}e_3 \\ F = 65 - \frac{1}{3}e_2 - \frac{7}{3}e_3 \end{array}$$

Tous les coûts réduits sont ≤ 0 donc on ne peut plus augmenter F : l'optimum est atteint et la solution optimale est

$$x_1^* = \frac{15}{2}, x_2^* = 5, e_1^* = \frac{27}{2}, e_2^* = 0, e_3^* = 0 \text{ avec } \max F = 65.$$

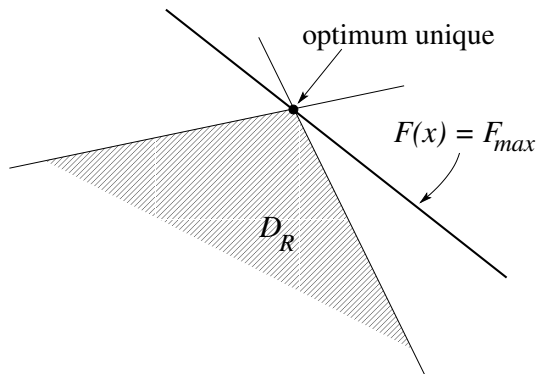
IV. Finitude du simplexe

A chaque étape de l'algorithme du simplexe (en phase 2), il y a des cas remarquables qui conduisent tous à l'arrêt de l'algorithme.

IV. Finitude du simplexe

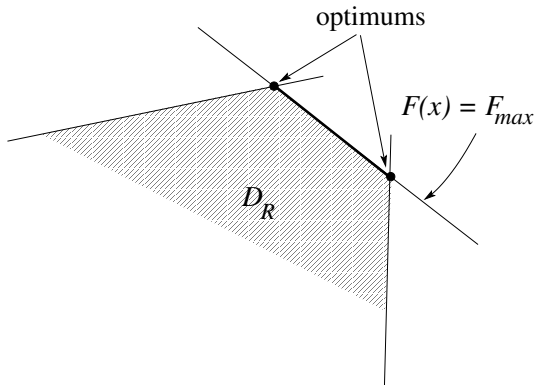
A chaque étape de l'algorithme du simplexe (en phase 2), il y a des cas remarquables qui conduisent tous à l'arrêt de l'algorithme.

- 1 Si les coûts réduits $L_H < \mathbf{0}$, alors la solution de base réalisable courante est l'unique optimum.

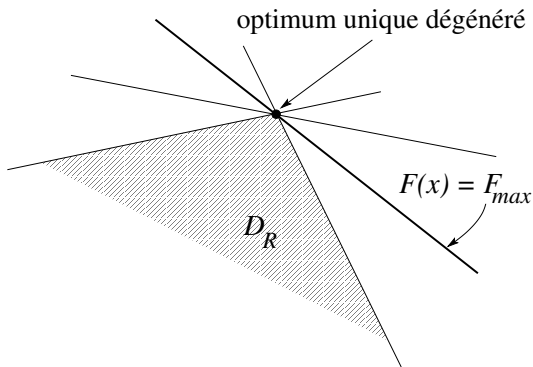


2 Si les coûts réduits $\mathbf{L}_H \leq \mathbf{0}$, alors il y a deux cas remarquables :

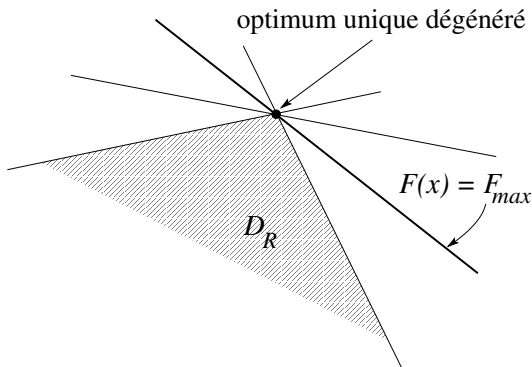
i) si $(\mathbf{L}_H)_e = 0$ et $x_e > 0$, alors l'optimum n'est pas unique.



- ii) si $(\mathbf{L}_H)_e = 0$ et $x_e = 0$, alors l'optimum est unique (a priori). Dans ce cas, la base est dite **dégénérée** c'est-à-dire qu'il existe une variable de base *nulle*.



- ii) si $(\mathbf{L}_H)_e = 0$ et $x_e = 0$, alors l'optimum est unique (a priori). Dans ce cas, la base est dite **dégénérée** c'est-à-dire qu'il existe une variable de base *nulle*.



- ③ Si $(\mathbf{L}_H)_e > 0$ et x_e est non borné alors la fonction objectif F n'est pas majorée.

Finitude du simplexe

Une solution de base réalisable est dite **dégénérée** si au moins une des variables de base est *nulle*.

Théorème

Si au cours de l'algorithme du simplexe, aucune base rencontrée n'est dégénérée, alors l'algorithme se termine en un nombre fini d'itérations.

Finitude du simplexe

Une solution de base réalisable est dite **dégénérée** si au moins une des variables de base est *nulle*.

Théorème

Si au cours de l'algorithme du simplexe, aucune base rencontrée n'est dégénérée, alors l'algorithme se termine en un nombre fini d'itérations.

Démonstration. A une itération donnée de l'algorithme :

- soit on détecte une fonction objectif non majorée (\rightarrow arrêt de l'algo.),
- soit elle est strictement croissante car $\tilde{F}_{opt} - F_{opt} = (\mathbf{L}_H)_e x_e > 0$ puisque $(L_H)_e > 0$ et $x_e > 0$ (par hypothèse, aucune base rencontrée n'est dégénérée).

Par conséquent, on ne rencontre jamais une base déjà rencontrée à une itération précédente. Le nombre de solution de base réalisable étant fini ($\leq C_n^m$), l'algorithme s'arrête nécessairement en un nombre fini d'itérations. □

Remarque: S'il existe une base dégénérée, alors on peut rencontrer un éventuel *cyclage* de l'algorithme : on retrouve une base déjà rencontrée et on boucle indéfiniment. Pour traiter les cas de dégénérescence, on peut appliquer la règle de Bland (1977) qui assure l'arrêt de l'algorithme en un nombre fini d'itérations.

Règle de Bland

Lorsque plusieurs variables sont susceptibles d'entrer ou de sortir de la base, on choisit toujours celle qui a l'indice le plus petit.

V. Initialisation du simplexe (phase 1)

1) Introduction

Pour un PL sous forme *canonique pure* avec les contraintes

$$A\mathbf{x} \leq \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0},$$

on peut déterminer facilement une solution de base réalisable dans le cas où $\mathbf{b} \geq \mathbf{0}$. En effet, sous forme standard les contraintes deviennent $A\mathbf{x} + \mathbf{e} = \mathbf{b}$, avec $\mathbf{x}, \mathbf{e} \geq \mathbf{0}$ où \mathbf{e} sont les *variables d'écart*.

Une solution de base réalisable évidente dans ce cas, est

$$\mathbf{x} = \mathbf{0}, \quad \mathbf{e} = \mathbf{b} \geq \mathbf{0}.$$

V. Initialisation du simplexe (phase 1)

1) Introduction

Pour un PL sous forme *canonique pure* avec les contraintes

$$A\mathbf{x} \leq \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0},$$

on peut déterminer facilement une solution de base réalisable dans le cas où $\mathbf{b} \geq \mathbf{0}$. En effet, sous forme standard les contraintes deviennent $A\mathbf{x} + \mathbf{e} = \mathbf{b}$, avec $\mathbf{x}, \mathbf{e} \geq \mathbf{0}$ où \mathbf{e} sont les *variables d'écart*.

Une solution de base réalisable évidente dans ce cas, est

$$\mathbf{x} = \mathbf{0}, \quad \mathbf{e} = \mathbf{b} \geq \mathbf{0}.$$

Mais pour un PL *sous forme standard*, il n'y a pas toujours de solution de base réalisable évidente.

Construction des solutions de base réalisable = phase d'initialisation du simplexe (phase 1).

2) Variables auxiliaires

PL sous forme standard

$$(PL) \quad \begin{cases} \max_{\mathbf{x}} [F(\mathbf{x}) = \mathbf{c}^T \mathbf{x}] \\ A\mathbf{x} = \mathbf{b} \\ \mathbf{x} \geq \mathbf{0} \end{cases}$$

On ne suppose pas que la matrice $A \in \mathcal{M}_{m \times n}$ est de rang plein, ni qu'il existe bien des solutions réalisables.

2) Variables auxiliaires

PL sous forme standard

$$(PL) \quad \begin{cases} \max_{\mathbf{x}} [F(\mathbf{x}) = \mathbf{c}^T \mathbf{x}] \\ A\mathbf{x} = \mathbf{b} \\ \mathbf{x} \geq \mathbf{0} \end{cases}$$

On ne suppose pas que la matrice $A \in \mathcal{M}_{m \times n}$ est de rang plein, ni qu'il existe bien des solutions réalisables.

Pour obtenir une solution de base réalisable ou bien pour détecter l'impossibilité, on introduit un problème de programmation linéaire **auxiliaire** pour des variables supplémentaires appelées **variables artificielles**.

Programme auxiliaire

Le programme auxiliaire associé à (PL) s'écrit

$$(PLA) \quad \left\{ \begin{array}{l} \min_{(\mathbf{x}, \mathbf{a})} \sum_{i=1}^m a_i \\ \mathbf{Ax} + \mathbf{a} = \mathbf{b} \\ \mathbf{x} \geq \mathbf{0} \\ \mathbf{a} \geq \mathbf{0} \end{array} \right.$$

où $\mathbf{a} = (a_1, \dots, a_m)$ sont appelées **variables artificielles**.

On a la propriété (évidente) suivante.

Proposition

Un (PL) admet une solution réalisable si et seulement si le problème auxiliaire (PLA) admet une solution de base optimale avec $\mathbf{a} = \mathbf{0}$.

On a la propriété (évidente) suivante.

Proposition

Un (PL) admet une solution réalisable si et seulement si le problème auxiliaire (PLA) admet une solution de base optimale avec $\mathbf{a} = \mathbf{0}$.

Détermination d'une solution de base réalisable via le problème

auxiliaire : On applique l'algorithme du simplexe au problème auxiliaire (PLA). A la fin du simplexe, le coût minimal est nul sinon on a détecté l'impossibilité pour (PL) (i.e. $\mathcal{D}_R = \emptyset$). Si tout s'est déroulé normalement (coût nul), on cherche à éliminer de la base toutes les variables artificielles.

Deux cas possibles :

- 1 On a réussi à faire sortir toutes les variables artificielles. On passe à la phase 2 du simplexe.
- 2 S'il reste des variables artificielles dans la base (base *dégénérée*) alors les lignes associées à ces variables sont des contraintes redondantes qu'on élimine.

Résumé de la phase d'initialisation du simplexe (phase 1)

On note F_{aux} la valeur de la fonction objectif du problème auxiliaire (PLA)

à la fin du simplexe, c'est-à-dire $F_{aux} = \min_{(x,a)} \sum_{i=1}^m a_i$.

- 1 Si $F_{aux} = 0$ et $\nexists a_j \in X_B$ où X_B désigne l'ensemble des variables de base pour (PLA), alors fin normale de la phase 1. On passe à la phase 2 du simplexe.
- 2 Si $F_{aux} = 0$ et $\exists a_j \in X_B$ avec $a_j = 0$, alors on supprime les lignes et colonnes associées aux a_j et on passe à la phase 2.
- 3 Si $F_{aux} > 0$ alors pas de solution réalisable ($\mathcal{D}_R = \emptyset$).

VI. Complexité du simplexe

Complexité = nombre d'itération dans le simplexe (phase2).

- On peut construire des exemples avec une complexité **exponentielle** en $\mathcal{O}(2^n)$ itérations (Klee-Minty, 1972).
- Mais dans la pratique la complexité du simplexe croît peu avec le nombre n de variables. En pratique, le nombre d'itérations est proportionnel au nombre m de contraintes (de m à $3m$ itérations).
- Si on tient compte de la résolution des systèmes linéaires avec une formule de mise à jour de l'inverse (Shermann-Morrison), on a $\mathcal{O}(m^2)$ opérations pour l'inverse.

VII. Quelques solveurs de PL

- GLPK (GNU Linear Programming Kit). Données au format MPS/CPLEX (standards de l'industrie).
- COIN-OR (COmputational INfrastructure for Operations Research) Optimization Suite, logiciel OpenSource.
- GUROBI Code commercial, licence éducation gratuite.
- CPLEX (IBM ILOG). Code commercial, licence éducation gratuite.
- FICO Xpress Optimization Suite
- AMPL (Automatic Mathematical Programming Language). Plateforme commerciale, version 'bridée' gratuite (limitation du nb de variables).

VII. Quelques solveurs de PL

- GLPK (GNU Linear Programming Kit). Données au format MPS/CPLEX (standards de l'industrie).
- COIN-OR (COmputational INfrastructure for Operations Research) Optimization Suite, logiciel OpenSource.
- GUROBI Code commercial, licence éducation gratuite.
- CPLEX (IBM ILOG). Code commercial, licence éducation gratuite.
- FICO Xpress Optimization Suite
- AMPL (Automatic Mathematical Programming Language). Plateforme commerciale, version 'bridée' gratuite (limitation du nb de variables).

Mais aussi ...

- MATLAB (fonction `linprog`, interface pour GUROBI, CPLEX, GLPK, ...)
- EXCEL
- PyGLPK interface GLPK pour Python.

Exemples d'utilisation de solveurs.

Problème de production (cf. Exercice 1, TD1)

$$\begin{aligned} \max F(\mathbf{x}) &= 1700x_1 + 3200x_2 \\ \left\{ \begin{array}{l} 3x_2 \leq 39 \\ 1.5x_1 + 4x_2 \leq 60 \\ 2x_1 + 3x_2 \leq 57 \\ 3x_1 \leq 57 \\ x_1, x_2 \geq 0 \end{array} \right. \end{aligned}$$

Exemples d'utilisation de solveurs.

Problème de production (cf. Exercice 1, TD1)

$$\begin{aligned} \max F(\mathbf{x}) &= 1700x_1 + 3200x_2 \\ \left\{ \begin{array}{l} 3x_2 \leq 39 \\ 1.5x_1 + 4x_2 \leq 60 \\ 2x_1 + 3x_2 \leq 57 \\ 3x_1 \leq 57 \\ x_1, x_2 \geq 0 \end{array} \right. \end{aligned}$$

- 1 GLPK avec format de données CPLEX

Exemples d'utilisation de solveurs.

Problème de production (cf. Exercice 1, TD1)

$$\begin{aligned} \max F(\mathbf{x}) &= 1700x_1 + 3200x_2 \\ &\left\{ \begin{array}{l} 3x_2 \leq 39 \\ 1.5x_1 + 4x_2 \leq 60 \\ 2x_1 + 3x_2 \leq 57 \\ 3x_1 \leq 57 \\ x_1, x_2 \geq 0 \end{array} \right. \end{aligned}$$

- 1 GLPK avec format de données CPLEX
- 2 MATLAB avec la fonction `linprog`

① GLPK avec format de données CPLEX : le fichier `exo1.lp`

maximize

$$F : 1700 x_1 + 3200 x_2$$

subject to

$$M1 : 3 x_2 \leq 39$$

$$M2 : 1.5 x_1 + 4 x_2 \leq 60$$

$$M3 : 2 x_1 + 3 x_2 \leq 57$$

$$M4 : 3 x_1 \leq 57$$

general

x_1

x_2

end

① GLPK avec format de données CPLEX : le fichier `exo1.lp`

```
maximize
```

```
    F : 1700 x1 + 3200 x2
```

```
subject to
```

```
    M1 : 3 x2 <= 39
```

```
    M2 : 1.5 x1 + 4 x2 <= 60
```

```
    M3 : 2 x1 + 3 x2 <= 57
```

```
    M4 : 3 x1 <= 57
```

```
general
```

```
    x1
```

```
    x2
```

```
end
```

Commande d'exécution (linux) :

```
>>glpsol --lp exo1.lp --nomip -o resultats.txt
```


Fichier des résultats `resultats.txt`

Problem:

Rows: 4

Columns: 2

Non-zeros: 6

Status: OPTIMAL

Objective: F = 54857.14286 (MAXimum)

No.	Row name	St	Activity	Lower bound	Upper bound	Marginal
1	M1	B	29.5714		39	
2	M2	NU	60		60	371.429
3	M3	NU	57		57	571.429
4	M4	B	41.1429		57	

No.	Column name	St	Activity	Lower bound	Upper bound	Marginal
1	x1	B	13.7143	0		
2	x2	B	9.85714	0		

② MATLAB avec la fonction `linprog` :

`[x, fval] = linprog(c,A,b,Aeq,beq,lb,ub,x0)`

résout le PL :

$$\begin{cases} \min_{\mathbf{x}} \mathbf{c}^T \mathbf{x} \\ \mathbf{Ax} \leq \mathbf{b} \\ \mathbf{A}_{\text{eq}} \mathbf{x} = \mathbf{b}_{\text{eq}} \\ \mathbf{lb} \leq \mathbf{x} \leq \mathbf{ub} \end{cases}$$

- `lb` et `ub` sont des *vecteurs*. Mettre `ub=[]` s'il n'y a pas de contrainte de bornes supérieures.
- En option, `x0` est une solution réalisable initiale.
- Par défaut, `linprog` utilise une méthode de *points intérieurs*. Pour le simplexe :

```
options = optimset('Algorithm', 'simplex');  
x = linprog(..., options);
```

Le fichier `exo1.m`

```
A = [0, 3;  
     1.5, 4;  
     2, 3;  
     3, 0];  
  
b = [39; 60; 57; 57];  
  
c = -[1700; 3200]; % Attention, linprog cherche un min !  
  
options = optimset('Algorithm','simplex');  
  
Aeq=[]; beq=[]; % pas de contraintes d'égalités  
lb=zeros(2,1); % bornes inférieures = 0  
ub=[]; % pas de borne supérieure  
x0=[]; % on ne fournit pas de solution réalisable au départ  
  
[X,fval]=linprog(c,A,b,Aeq,beq,lb,ub,x0,options)
```

Dans MATLAB :

```
>> exo1
```

```
Optimization terminated.
```

```
X =
```

```
13.7143
```

```
9.8571
```

```
fval =
```

```
-5.4857e+04
```