

Méthodes d'optimisation

BUT Info 2e année

Florent Foucaud
Dipayan Chakraborty, Malika More, Adrien Wohrer



IUT CLERMONT AUVERGNE

Aurillac - Clermont-Ferrand - Le Puy-en-Velay
Montluçon - Moulins - Vichy

2022-2023

Récapitulatif

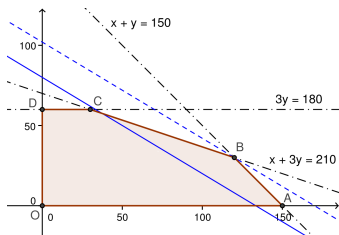
Représentation géométrique d'un programme linéaire

Exemple de programme linéaire

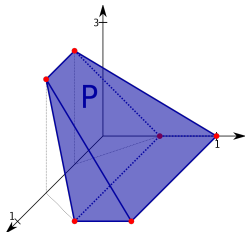
$$\begin{array}{rcllcl} \text{maximiser :} & 10x & + & 5y & & \\ \text{tel que :} & 1.5x & - & 2y & \leq & 1000 \\ & 3x & + & y & \leq & 1500 \\ & x & & & \geq & 0 \\ & & & y & \geq & 0 \end{array}$$

Représentation géométrique d'un programme linéaire

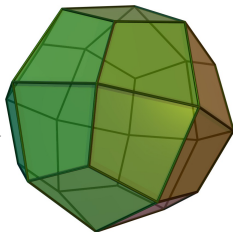
- On se place dans un espace à n dimensions, n : nombre de variables du PL.
- Chaque contrainte est un espace à $n - 1$ dimensions, qui divise notre espace en deux :
→ les points qui satisfont / ne satisfont pas la contrainte.
- Ces contraintes définissent un **polytope** convexe, qui contient tous les points correspondant à une solution du PL.
Chaque contrainte produit une **facette** du polytope.



2 variables : polygone

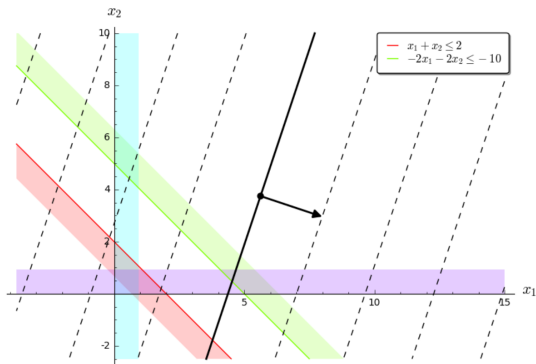


3 variables : polyèdre



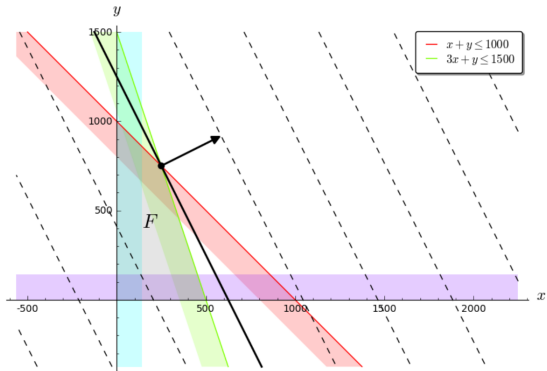
Quatre types de programmes linéaires

1. Pas de solution du tout



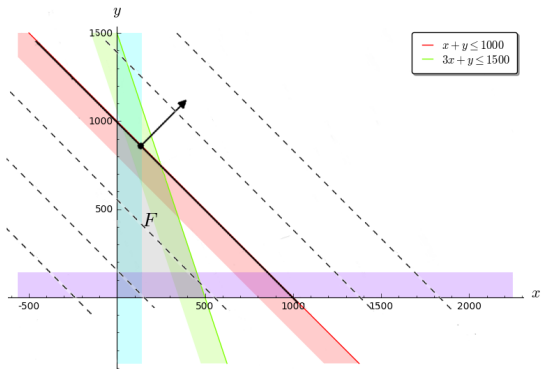
Quatre types de programmes linéaires

1. Pas de solution du tout
2. Une solution optimale unique, sur un sommet du polytope



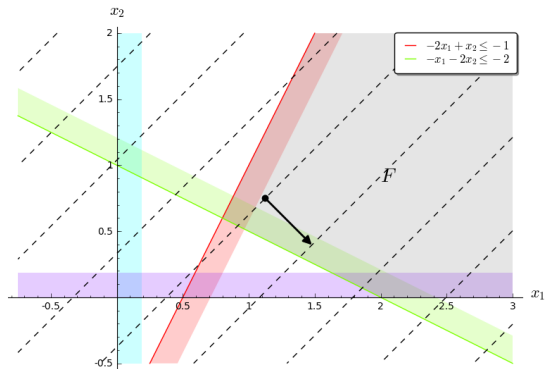
Quatre types de programmes linéaires

1. Pas de solution du tout
2. Une solution optimale unique, sur un sommet du polytope
3. Une infinité de solutions optimales, sur une arête du polytope



Quatre types de programmes linéaires

1. Pas de solution du tout
2. Une solution optimale unique, sur un sommet du polytope
3. Une infinité de solutions optimales, sur une arête du polytope
4. Une infinité de solutions, mais pas de solution optimale : PL non borné



Flots dans les graphes

Flot maximum dans un graphe

On considère un **graphe orienté pondéré** $G = (V, A, c)$ avec une **capacité** $c(\vec{xy}) \in \mathbb{R}^+$ pour chaque arc $\vec{xy} \in A$, une source s et une destination t .

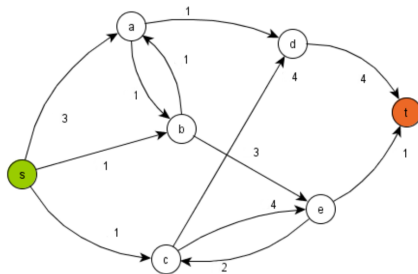
Définition

Un **flot** est une assignation $f : A \rightarrow \mathbb{R}^+$ telle que :

1. pour tout arc $\vec{xy} \in A : f(\vec{xy}) \leq c(\vec{xy})$, *(respect des capacités)*
2. pour tout sommet $v \in V - \{s, t\} : \sum_{x: x \rightarrow v} f(\vec{xv}) = \sum_{y: v \rightarrow y} f(\vec{vy})$.

(conservation locale du flot)

On cherche à **maximiser** la valeur $\sum_{y: s \rightarrow y} f(\vec{sy})$ du flot.



Flot maximum dans un graphe

On considère un **graphe orienté pondéré** $G = (V, A, c)$ avec une **capacité** $c(\overrightarrow{xy}) \in \mathbb{R}^+$ pour chaque arc $\overrightarrow{xy} \in A$, une source s et une destination t .

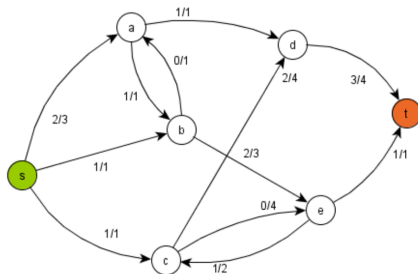
Définition

Un **flot** est une assignation $f : A \rightarrow \mathbb{R}^+$ telle que :

1. pour tout arc $\overrightarrow{xy} \in A : f(\overrightarrow{xy}) \leq c(\overrightarrow{xy})$, *(respect des capacités)*
2. pour tout sommet $v \in V - \{s, t\} : \sum_{x: x \rightarrow v} f(\overrightarrow{xv}) = \sum_{y: v \rightarrow y} f(\overrightarrow{vy})$.

(conservation locale du flot)

On cherche à **maximiser** la valeur $\sum_{y: s \rightarrow y} f(\overrightarrow{sy})$ du flot.



Flot maximum dans un graphe

On considère un **graphe orienté pondéré** $G = (V, A, c)$ avec une **capacité** $c(\overrightarrow{xy}) \in \mathbb{R}^+$ pour chaque arc $\overrightarrow{xy} \in A$, une source s et une destination t .

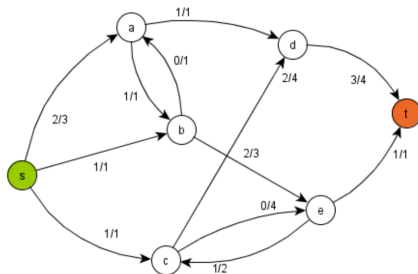
Définition

Un **flot** est une assignation $f : A \rightarrow \mathbb{R}^+$ telle que :

1. pour tout arc $\overrightarrow{xy} \in A : f(\overrightarrow{xy}) \leq c(\overrightarrow{xy})$, *(respect des capacités)*
2. pour tout sommet $v \in V - \{s, t\} : \sum_{x: x \rightarrow v} f(\overrightarrow{xv}) = \sum_{y: v \rightarrow y} f(\overrightarrow{vy})$. *(conservation locale du flot)*

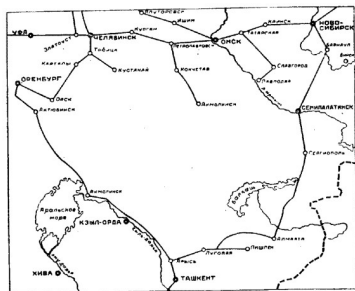
On cherche à **maximiser** la valeur $\sum_{y: s \rightarrow y} f(\overrightarrow{sy})$ du flot.

Applications : réseaux de transport



Flots : un peu d'histoire

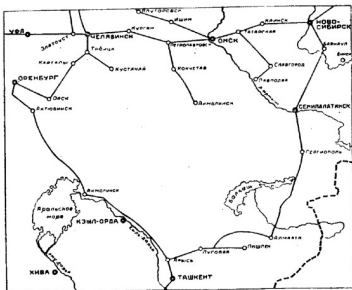
- 1930 : Problème formulé par A. N. Tolstoï, appliqué au réseau de trains en URSS



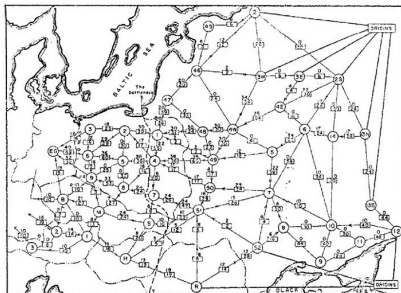
10 sources et 68 destinations soviétiques en 1930

Flots : un peu d'histoire

- 1930 : Problème formulé par A. N. Tolstoï, appliqué au réseau de trains en URSS
- 1954 : T. E. Harris et Gen. F. S. Ross étudient à leur tour le problème !
(rapport secret de l'US Air Force déclassifié en 1999)



10 sources et 68 destinations soviétiques en 1930



Le réseau ferré d'Europe de l'Est en 1954

Flots : un peu d'histoire

- 1930 : Problème formulé par A. N. Tolstoï, appliqué au réseau de trains en URSS
- 1954 : T. E. Harris et Gen. F. S. Ross étudient à leur tour le problème ! (rapport secret de l'US Air Force déclassifié en 1999)
- 1955 : Algorithme optimal de Ford-Fulkerson pour trouver le flot max



Lester R. Ford
(1886-1967)



Delbert R. Fulkerson
(1924-1976)

Flots : un peu d'histoire

- 1930 : Problème formulé par A. N. Tolstoï, appliqué au réseau de trains en URSS
- 1954 : T. E. Harris et Gen. F. S. Ross étudient à leur tour le problème ! (rapport secret de l'US Air Force déclassifié en 1999)
- 1955 : Algorithme optimal de Ford-Fulkerson pour trouver le flot max
- 1970-72 : Algorithme de Dinitz-Edmonds-Karp, qui termine à tous les coups, en temps $O(|V| \cdot |A|^2)$ ou $O(|V|^2 \cdot |A|)$



Lester R. Ford
(1886-1967)



Delbert R. Fulkerson
(1924-1976)



Yefim Dinitz
(1950-)



Jack Edmonds
(1934-)



Richard M. Karp
(1935-)

Flots : un peu d'histoire

- 1930 : Problème formulé par A. N. Tolstoï, appliqué au réseau de trains en URSS
- 1954 : T. E. Harris et Gen. F. S. Ross étudient à leur tour le problème ! (rapport secret de l'US Air Force déclassifié en 1999)
- 1955 : Algorithme optimal de Ford-Fulkerson pour trouver le flot max
- 1970-72 : Algorithme de Dinitz-Edmonds-Karp, qui termine à tous les coups, en temps $O(|V| \cdot |A|^2)$ ou $O(|V|^2 \cdot |A|)$
- ...
- 2013 : Meilleur algorithme à ce jour, en temps $O(|V| \cdot |A|)$ (J. B. Orlin)



Lester R. Ford
(1886-1967)



Delbert R. Fulkerson
(1924-1976)



Yefim Dinitz
(1950-)



Jack Edmonds
(1934-)



Richard M. Karp
(1935-)



James B. Orlin
(1953-)

Flots sous forme de programmes linéaires

Définition

Un **flot** est une assignation $f : A \rightarrow \mathbb{R}^+$ telle que :

1. pour tout arc $\vec{xy} \in A : f(\vec{xy}) \leq c(\vec{xy})$, *(respect des capacités)*
2. pour tout sommet $v \in V - \{s, t\} : \sum_{x: x \rightarrow v} f(\vec{xv}) = \sum_{y: v \rightarrow y} f(\vec{vy})$.

(conservation locale du flot)

On cherche à **maximiser** la valeur $\sum_{y: s \rightarrow y} f(\vec{sy})$ du flot.

Flots sous forme de programmes linéaires

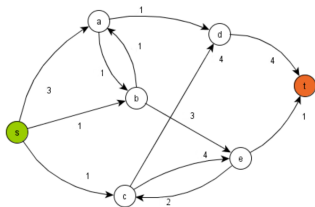
Définition

Un **flot** est une assignation $f : A \rightarrow \mathbb{R}^+$ telle que :

1. pour tout arc $\vec{xy} \in A : f(\vec{xy}) \leq c(\vec{xy})$, *(respect des capacités)*
2. pour tout sommet $v \in V - \{s, t\} : \sum_{x: x \rightarrow v} f(\vec{xv}) = \sum_{y: v \rightarrow y} f(\vec{vy})$.

(conservation locale du flot)

On cherche à **maximiser** la valeur $\sum_{y: s \rightarrow y} f(\vec{sy})$ du flot.



Flots sous forme de programmes linéaires

Définition

Un **flot** est une assignation $f : A \rightarrow \mathbb{R}^+$ telle que :

1. pour tout arc $\vec{xy} \in A : f(\vec{xy}) \leq c(\vec{xy})$,

(respect des capacités)

2. pour tout sommet $v \in V - \{s, t\} : \sum_{x:x \rightarrow v} f(\vec{xv}) = \sum_{y:v \rightarrow y} f(\vec{vy})$.

(conservation locale du flot)

On cherche à **maximiser** la valeur $\sum_{y:s \rightarrow y} f(\vec{sy})$ du flot.

Programme linéaire correspondant :

On prend une variable f_a pour chaque arc a .

maximiser : $f_{\vec{sa}} + f_{\vec{sb}} + f_{\vec{sc}}$

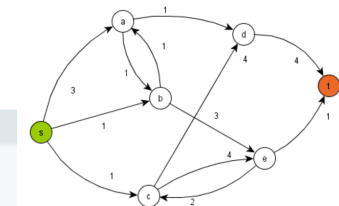
tel que : $f_{\vec{sa}} \leq 3$

$f_{\vec{sa}} \geq 0$

(idem pour tous les arcs)

$f_{\vec{sa}} + f_{\vec{ba}} = f_{\vec{ab}} + f_{\vec{ad}}$

(idem pour tous les sommets)



capacité arc \vec{sa}

flot non-négatif pour arc \vec{sa}

contrainte flot sommet a

Flots sous forme de programmes linéaires

Définition

Un **flot** est une assignation $f : A \rightarrow \mathbb{R}^+$ telle que :

1. pour tout arc $\vec{xy} \in A : f(\vec{xy}) \leq c(\vec{xy})$, *(respect des capacités)*
2. pour tout sommet $v \in V - \{s, t\} : \sum_{x: x \rightarrow v} f(\vec{xv}) = \sum_{y: v \rightarrow y} f(\vec{vy})$.

(conservation locale du flot)

On cherche à **maximiser** la valeur $\sum_{y: s \rightarrow y} f(\vec{sy})$ du flot.

Programme linéaire en général :

On prend une variable f_a pour chaque arc a .

maximiser : $\sum_{y: s \rightarrow y} f_{\vec{sy}}$

tel que :

$$\begin{array}{rcll} f_a & \leq & c(a) & \text{pour tout arc } a \\ f_a & \geq & 0 & \text{pour tout arc } a \\ \sum_{x: x \rightarrow v} f_{\vec{xv}} & = & \sum_{y: v \rightarrow y} f_{\vec{vy}} & \text{pour tout sommet } v \end{array}$$

L'algorithme du simplexe

Reformulation d'un PL

Souvent, on standardise les PL avec les conditions suivantes :

- La fonction objectif est à maximiser
- Les contraintes avec au moins deux variables sont des \leq
- Les variables sont toutes ≥ 0

$$\text{minimiser :} \quad 10x + 5y$$

$$\begin{aligned} \text{tel que :} \quad & 1.5x - 2y \geq 1000 \\ & 3x + y \leq 1500 \\ & y \geq -2 \end{aligned}$$

Reformulation d'un PL

Souvent, on standardise les PL avec les conditions suivantes :

- La fonction objectif est à maximiser
- Les contraintes avec au moins deux variables sont des \leq
- Les variables sont toutes ≥ 0

$$\text{minimiser : } 10x + 5y$$

$$\text{tel que : } \begin{array}{rcll} 1.5x - 2y & \geq & 1000 \\ 3x + y & \leq & 1500 \\ y & \geq & -2 \end{array}$$

$$\text{max. : } -10x - 5y$$

$$\text{t.q. : } \begin{array}{rcll} 1.5x - 2y & \geq & 1000 \\ 3x + y & \leq & 1500 \\ y & \geq & -2 \end{array}$$

Reformulation d'un PL

Souvent, on standardise les PL avec les conditions suivantes :

- La fonction objectif est à maximiser
- Les contraintes avec au moins deux variables sont des \leq
- Les variables sont toutes ≥ 0

$$\text{minimiser : } 10x + 5y$$

$$\text{tel que : } \begin{array}{rcll} 1.5x - 2y & \geq & 1000 & \\ 3x + y & \leq & 1500 & \\ y & \geq & -2 & \end{array}$$

$$\text{max. : } -10x - 5y$$

$$\text{t.q. : } \begin{array}{rcll} -1.5x + 2y & \leq & -1000 & \\ 3x + y & \leq & 1500 & \\ y & \geq & -2 & \end{array}$$

Reformulation d'un PL

Souvent, on standardise les PL avec les conditions suivantes :

- La fonction objectif est à maximiser
- Les contraintes avec au moins deux variables sont des \leq
- Les variables sont toutes ≥ 0

$$\begin{array}{l} \text{minimiser :} \\ \text{tel que :} \end{array} \quad \begin{array}{r} 10x + 5y \\ 1.5x - 2y \geq 1000 \\ 3x + y \leq 1500 \\ y \geq -2 \end{array}$$

$$\begin{array}{l} \text{max. :} \\ \text{t.q. :} \end{array} \quad \begin{array}{r} -10x - 5(y' - 2) \\ -1.5x + 2(y' - 2) \leq -1000 \\ 3x + y' - 2 \leq 1500 \\ y' \geq 0 \end{array}$$

1. On pose $y = y' - 2$ avec $y' \geq 0$

Reformulation d'un PL

Souvent, on standardise les PL avec les conditions suivantes :

- La fonction objectif est à maximiser
- Les contraintes avec au moins deux variables sont des \leq
- Les variables sont toutes ≥ 0

$$\begin{array}{l} \text{minimiser :} \\ \text{tel que :} \end{array} \quad \begin{array}{r} 10x + 5y \\ 1.5x - 2y \geq 1000 \\ 3x + y \leq 1500 \\ y \geq -2 \end{array}$$

$$\begin{array}{l} \text{max. :} \\ \text{t.q. :} \end{array} \quad \begin{array}{r} -10x - 5y' \\ -1.5x + 2y' \leq -996 \\ 3x + y' \leq 1502 \\ y' \geq 0 \end{array}$$

1. On pose $y = y' - 2$ avec $y' \geq 0$

Reformulation d'un PL

Souvent, on standardise les PL avec les conditions suivantes :

- La fonction objectif est à maximiser
- Les contraintes avec au moins deux variables sont des \leq
- Les variables sont toutes ≥ 0

$$\begin{array}{l} \text{minimiser :} \\ \text{tel que :} \end{array} \quad \begin{array}{r} 10x + 5y \\ 1.5x - 2y \geq 1000 \\ 3x + y \leq 1500 \\ y \geq -2 \end{array}$$

$$\begin{array}{l} \text{max. :} \\ \text{t.q. :} \end{array} \quad \begin{array}{r} -10(x^+ - x^-) - 5y' \\ -1.5(x^+ - x^-) + 2y' \leq -996 \\ 3(x^+ - x^-) + y' \leq 1502 \\ y' \geq 0 \\ x^+ \geq 0 \\ x^- \geq 0 \end{array}$$

1. On pose $y = y' - 2$ avec $y' \geq 0$

2. On pose $x = x^+ - x^-$ avec $x^+, x^- \geq 0$

Reformulation d'un PL

Souvent, on standardise les PL avec les conditions suivantes :

- La fonction objectif est à maximiser
- Les contraintes avec au moins deux variables sont des \leq
- Les variables sont toutes ≥ 0

$$\begin{array}{l} \text{minimiser :} \\ \text{tel que :} \end{array} \quad \begin{array}{r} 10x + 5y \\ 1.5x - 2y \geq 1000 \\ 3x + y \leq 1500 \\ y \geq -2 \end{array}$$

$$\begin{array}{l} \text{max. :} \\ \text{t.q. :} \end{array} \quad \begin{array}{r} -10x^+ + 10x^- - 5y' \\ -1.5x^+ + 1.5x^- + 2y' \leq -996 \\ 3x^+ - 3x^- + y' \leq 1502 \\ y' \geq 0 \\ x^+ \geq 0 \\ x^- \geq 0 \end{array}$$

1. On pose $y = y' - 2$ avec $y' \geq 0$

2. On pose $x = x^+ - x^-$ avec $x^+, x^- \geq 0$

Forme standard d'un PL

Un programme linéaire est **sous forme standard** si :

- La fonction objectif est à maximiser
- Les contraintes avec ≥ 2 variables sont des \leq
- Les variables sont toutes ≥ 0
- Les contraintes (autres que "variable ≥ 0 ") sont des égalités

$$\begin{array}{rcll} \text{max. :} & 10x & + & 5y \\ \text{t.q. :} & 1.5x & - & 2y \leq 1000 \\ & 3x & + & y \leq 1500 \\ & x & & \geq 0 \\ & & & y \geq 0 \end{array}$$

Forme standard d'un PL

Un programme linéaire est **sous forme standard** si :

- La fonction objectif est à maximiser
- Les contraintes avec ≥ 2 variables sont des \leq
- Les variables sont toutes ≥ 0
- Les contraintes (autres que "variable ≥ 0 ") sont des égalités

$$\begin{array}{rcll} \text{max. :} & 10x & + & 5y \\ \text{t.q. :} & 1.5x & - & 2y \leq 1000 \\ & 3x & + & y \leq 1500 \\ & x & & \geq 0 \\ & & & y \geq 0 \end{array}$$

$$\begin{array}{rcll} \text{max. :} & 10x & + & 5y \\ \text{t.q. :} & 1.5x & - & 2y + e_1 = 1000 \\ & 3x & + & y + e_2 = 1500 \\ & x & & \geq 0 \\ & & & y \geq 0 \\ & & & e_1 \geq 0 \\ & & & e_2 \geq 0 \end{array}$$

On introduit une **variable d'écart** pour chaque contrainte

Forme standard d'un PL

Un programme linéaire est **sous forme standard** si :

- La fonction objectif est à maximiser
- Les contraintes avec ≥ 2 variables sont des \leq
- Les variables sont toutes ≥ 0
- Les contraintes (autres que "variable ≥ 0 ") sont des égalités

$$\text{max. : } 10x + 5y$$

$$\begin{aligned} \text{t.q. : } 1.5x - 2y &\leq 1000 \\ 3x + y &\leq 1500 \\ x &\geq 0 \\ y &\geq 0 \end{aligned}$$

$$\text{max. : } 10x + 5y$$

$$\begin{aligned} \text{t.q. : } 1.5x - 2y + e_1 &= 1000 \\ 3x + y + e_2 &= 1500 \\ x &\geq 0 \\ y &\geq 0 \\ e_1 &\geq 0 \\ e_2 &\geq 0 \end{aligned}$$

On introduit une **variable d'écart** pour chaque contrainte

Remarque : $2x - 3y \geq 1000$ est équivalent à $2x - 3y - e = 1000$ et $e \geq 0$

Forme standard d'un PL

Un programme linéaire est **sous forme standard** si :

- La fonction objectif est à maximiser
- Les contraintes avec \geq 2 variables sont des \leq
- Les variables sont toutes ≥ 0
- Les contraintes (autres que "variable ≥ 0 ") sont des égalités

$$\text{max. : } 10x + 5y$$

$$\begin{aligned} \text{t.q. : } & 1.5x - 2y \leq 1000 \\ & 3x + y \leq 1500 \\ & x \geq 0 \\ & y \geq 0 \end{aligned}$$

$$\text{max. : } 10x + 5y$$

$$\begin{aligned} \text{t.q. : } & 1.5x - 2y + e_1 = 1000 \\ & 3x + y + e_2 = 1500 \\ & x \geq 0 \\ & y \geq 0 \\ & e_1 \geq 0 \\ & e_2 \geq 0 \end{aligned}$$

On introduit une **variable d'écart** pour chaque contrainte

Remarque : $2x - 3y \geq 1000$ est équivalent à $2x - 3y - e = 1000$ et $e \geq 0$

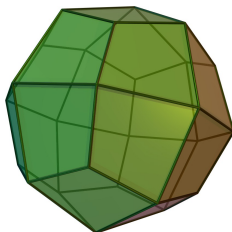
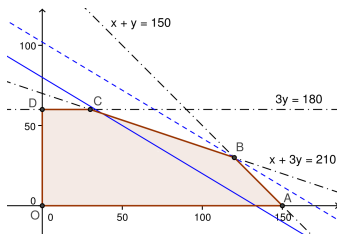
Proposition

Tout PL est équivalent à sa forme standard.

L'algo du simplexe : principe

Principe général :

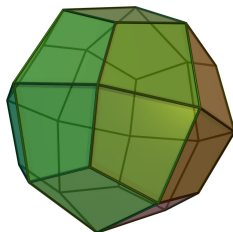
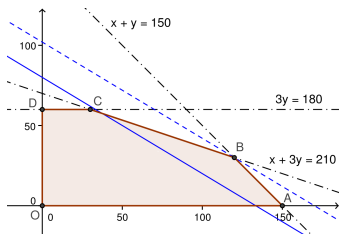
0. On part d'un PL en **forme standard**
1. On trouve une solution non-optimale en un point du polytope associé à notre PL
2. Tant qu'on peut, on évolue vers une **solution proche** qui **améliore** la fonction objectif
(si on ne peut plus améliorer la solution courante, on s'arrête : on a trouvé une solution optimale !)



L'algo du simplexe : principe

Principe général :

0. On part d'un PL en **forme standard**
1. On trouve une solution non-optimale en un point du polytope associé à notre PL
2. Tant qu'on peut, on évolue vers une **solution proche** qui **améliore** la fonction objectif
(si on ne peut plus améliorer la solution courante, on s'arrête : on a trouvé une solution optimale !)



Quelques détails techniques délicats :

- Trouver la **solution initiale** n'est pas forcément si facile !
- Il faut éviter de **boucler** en cours de route
- Pour aller de solution en solution, on utilise la notion de **pivot**

L'algo du simplexe : historique

- Interactions entre Dantzig et von Neumann
- Proposé en 1947 par George B. Dantzig
- Initialement jugé (par erreur) mauvais en performances
- En réalité, il est très performant en pratique
- Implémenté pour la première fois en 1952



George B. Dantzig (1914-2005)

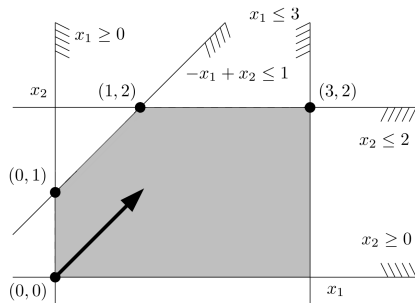


John von Neumann (1903-1957)

L'algo sur un exemple

maximiser : $x_1 + x_2$
tel que :

$$\begin{array}{rcll} -x_1 + x_2 & \leq & 1 \\ x_1 & \leq & 3 \\ x_2 & \leq & 2 \\ x_1 & \geq & 0 \\ x_2 & \geq & 0 \end{array}$$

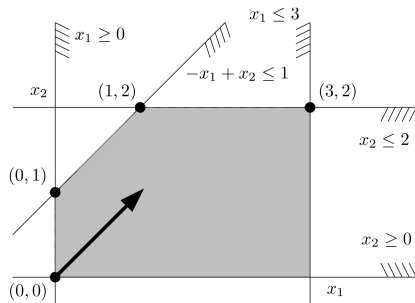


L'algo sur un exemple

maximiser : $x_1 + x_2$

tel que :

$$\begin{array}{rcl} -x_1 + x_2 & \leq & 1 \\ x_1 & \leq & 3 \\ x_2 & \leq & 2 \\ x_1 & \geq & 0 \\ x_2 & \geq & 0 \end{array}$$



On passe le PL en **forme standard** via des variables d'écart x_3, x_4, x_5 :

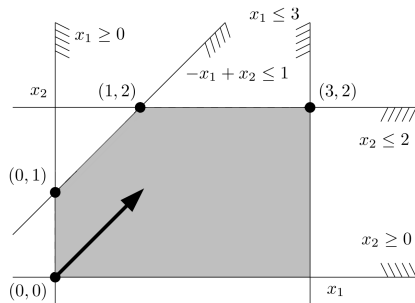
maximiser : $x_1 + x_2$

tel que :

$$\begin{array}{rcll} -x_1 + x_2 + x_3 & = & 1 \\ x_1 + x_4 & = & 3 \\ x_2 + x_5 & = & 2 \\ x_1 & \geq & 0 \\ x_2 & \geq & 0 \\ x_3 & \geq & 0 \\ x_4 & \geq & 0 \\ x_5 & \geq & 0 \end{array}$$

L'algo sur un exemple

$$\begin{array}{l}
 \text{maximiser :} \\
 \text{tel que :}
 \end{array}
 \begin{array}{rcl}
 x_1 & + & x_2 \\
 -x_1 & + & x_2 \leq 1 \\
 x_1 & & \leq 3 \\
 & & x_2 \leq 2 \\
 x_1 & & \geq 0 \\
 & & x_2 \geq 0
 \end{array}$$



On passe le PL en **forme standard** via des variables d'écart x_3, x_4, x_5 :

$$\begin{array}{l}
 \text{maximiser :} \\
 \text{tel que :}
 \end{array}
 \begin{array}{rcl}
 x_1 & + & x_2 \\
 -x_1 & + & x_2 + x_3 = 1 \\
 x_1 & & + x_4 = 3 \\
 & & x_2 + x_5 = 2 \\
 x_1 & & \geq 0 \\
 & & x_2 \geq 0 \\
 & & x_3 \geq 0 \\
 & & x_4 \geq 0 \\
 & & x_5 \geq 0
 \end{array}$$

On voit que $x_1 = x_2 = 0$ est une solution (non-optimale) du PL originel.

→ Cela implique $x_3 = 1, x_4 = 3, x_5 = 2$ dans le nouveau PL : **solution $(0, 0, 1, 3, 2)$** .

Exemple d'exécution de l'algorithme du simplexe

$$\text{maximiser } z = x_1 + x_2$$

tel que :

$$-x_1 + x_2 + x_3 = 1$$

$$x_1 + x_4 = 3$$

$$x_2 + x_5 = 2$$

$$x_1, x_2, x_3, x_4, x_5 \geq 0$$

Exemple d'exécution de l'algorithme du simplexe

$$\text{maximiser } z = x_1 + x_2$$

$$\begin{aligned} \text{tel que :} \quad & -x_1 + x_2 + x_3 = 1 \\ & x_1 + x_4 = 3 \\ & x_2 + x_5 = 2 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0 \end{aligned}$$

x_3	$=$	1	$+$	x_1	$-$	x_2
x_4	$=$	3	$-$	x_1		
x_5	$=$	2			$-$	x_2
z	$=$			x_1	$+$	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

On démarre avec notre **solution basique** $x_1 = x_2 = 0$, $x_3 = 1$, $x_4 = 3$, $x_5 = 2$.
Les variables non-nulles x_3 , x_4 , x_5 sont appelées **basiques**.

On réécrit le PL sous forme d'un **tableau de simplexe**.

Exemple d'exécution de l'algorithme du simplexe

x_3	=	1	+	x_1	-	x_2
x_4	=	3	-	x_1		
x_5	=	2			-	x_2
z	=			x_1	+	x_2

Solution : (0, 0, 1, 3, 2), $z = 0$

On veut améliorer la solution en augmentant **une seule variable** qui était à 0.

Exemple d'exécution de l'algorithme du simplexe

x_3	=	1	+	x_1	-	x_2
x_4	=	3	-	x_1		
x_5	=	2			-	x_2
z	=			x_1	+	x_2

Solution : (0, 0, 1, 3, 2), $z = 0$

On veut améliorer la solution en augmentant **une seule variable** qui était à 0.

On peut choisir x_1 ou x_2 : prenons x_2 . C'est le **pivot**.

Exemple d'exécution de l'algorithme du simplexe

x_3	=	1	+	x_1	-	x_2
x_4	=	3	-	x_1		
x_5	=	2			-	x_2
z	=			x_1	+	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

On veut améliorer la solution en augmentant **une seule variable** qui était à 0.

On peut choisir x_1 ou x_2 : prenons x_2 . C'est le **pivot**.

Par exemple, si on prend $x_2 = 1$, on obtient $z = 1$.

Si on prend $x_2 = 2$, on obtient $z = 2$, c'est encore mieux.

Exemple d'exécution de l'algorithme du simplexe

x_3	=	1	+	x_1	-	x_2
x_4	=	3	-	x_1		
x_5	=	2			-	x_2
z	=			x_1	+	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

On veut améliorer la solution en augmentant **une seule variable** qui était à 0.

On peut choisir x_1 ou x_2 : prenons x_2 . C'est le **pivot**.

Par exemple, si on prend $x_2 = 1$, on obtient $z = 1$.

Si on prend $x_2 = 2$, on obtient $z = 2$, c'est encore mieux.

Par contre, on aurait alors $x_3 = 1 + 0 - 2 < 0$ ce qui n'est pas autorisé....

Exemple d'exécution de l'algorithme du simplexe

x_3	=	1	+	x_1	-	x_2
x_4	=	3	-	x_1		
x_5	=	2			-	x_2
z	=			x_1	+	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

On veut améliorer la solution en augmentant **une seule variable** qui était à 0.

On peut choisir x_1 ou x_2 : prenons x_2 . C'est le **pivot**.

De combien peut-on augmenter x_2 ?

Regardons nos contraintes (avec $x_1 = 0$) :

Exemple d'exécution de l'algorithme du simplexe

x_3	$=$	1	$+$	x_1	$-$	x_2
x_4	$=$	3	$-$	x_1		
x_5	$=$	2			$-$	x_2
z	$=$			x_1	$+$	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

On veut améliorer la solution en augmentant **une seule variable** qui était à 0.

On peut choisir x_1 ou x_2 : prenons x_2 . C'est le **pivot**.

De combien peut-on augmenter x_2 ?

Regardons nos contraintes (avec $x_1 = 0$) :

$$x_3 = 1 + x_1 - x_2 \geq 0 \text{ donc } 1 - x_2 \geq 0 \text{ et } 1 \geq x_2$$

Exemple d'exécution de l'algorithme du simplexe

x_3	$=$	1	$+$	x_1	$-$	x_2
x_4	$=$	3	$-$	x_1		
x_5	$=$	2			$-$	x_2
z	$=$			x_1	$+$	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

On veut améliorer la solution en augmentant **une seule variable** qui était à 0.

On peut choisir x_1 ou x_2 : prenons x_2 . C'est le **pivot**.

De combien peut-on augmenter x_2 ?

Regardons nos contraintes (avec $x_1 = 0$) :

$$x_3 = 1 + x_1 - x_2 \geq 0 \text{ donc } 1 - x_2 \geq 0 \text{ et } 1 \geq x_2$$

$$x_4 = 3 - x_1 \geq 0 \rightarrow \text{aucune influence sur } x_2$$

Exemple d'exécution de l'algorithme du simplexe

x_3	=	1	+	x_1	-	x_2
x_4	=	3	-	x_1		
x_5	=	2			-	x_2
z	=			x_1	+	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

On veut améliorer la solution en augmentant **une seule variable** qui était à 0.

On peut choisir x_1 ou x_2 : prenons x_2 . C'est le **pivot**.

De combien peut-on augmenter x_2 ?

Regardons nos contraintes (avec $x_1 = 0$) :

$$x_3 = 1 + x_1 - x_2 \geq 0 \text{ donc } 1 - x_2 \geq 0 \text{ et } 1 \geq x_2$$

$$x_4 = 3 - x_1 \geq 0 \rightarrow \text{aucune influence sur } x_2$$

$$x_5 = 2 - x_2 \geq 0 \text{ donc } 2 - x_2 \geq 0 \text{ et } 2 \geq x_2$$

Exemple d'exécution de l'algorithme du simplexe

x_3	$=$	1	$+$	x_1	$-$	x_2
x_4	$=$	3	$-$	x_1		
x_5	$=$	2			$-$	x_2
z	$=$			x_1	$+$	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

On veut améliorer la solution en augmentant **une seule variable** qui était à 0.

On peut choisir x_1 ou x_2 : prenons x_2 . C'est le **pivot**.

De combien peut-on augmenter x_2 ?

Regardons nos contraintes (avec $x_1 = 0$) :

$$x_3 = 1 + x_1 - x_2 \geq 0 \text{ donc } 1 - x_2 \geq 0 \text{ et } 1 \geq x_2$$

$$x_4 = 3 - x_1 \geq 0 \rightarrow \text{aucune influence sur } x_2$$

$$x_5 = 2 - x_2 \geq 0 \text{ donc } 2 - x_2 \geq 0 \text{ et } 2 \geq x_2$$

On **augmente** x_2 **au maximum autorisé** : $x_2 = 1$, et on garde $x_1 = 0$.

Exemple d'exécution de l'algorithme du simplexe

x_3	$=$	1	$+$	x_1	$-$	x_2
x_4	$=$	3	$-$	x_1		
x_5	$=$	2			$-$	x_2
z	$=$			x_1	$+$	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

On veut améliorer la solution en augmentant **une seule variable** qui était à 0.

On peut choisir x_1 ou x_2 : prenons x_2 . C'est le **pivot**.

De combien peut-on augmenter x_2 ?

Regardons nos contraintes (avec $x_1 = 0$) :

$$x_3 = 1 + x_1 - x_2 \geq 0 \text{ donc } 1 - x_2 \geq 0 \text{ et } 1 \geq x_2$$

$$x_4 = 3 - x_1 \geq 0 \rightarrow \text{aucune influence sur } x_2$$

$$x_5 = 2 - x_2 \geq 0 \text{ donc } 2 - x_2 \geq 0 \text{ et } 2 \geq x_2$$

On **augmente** x_2 **au maximum autorisé** : $x_2 = 1$, et on garde $x_1 = 0$.

On calcule x_3 , x_4 et x_5 grâce au tableau : $x_3 = 0$, $x_4 = 3$, $x_5 = 1$

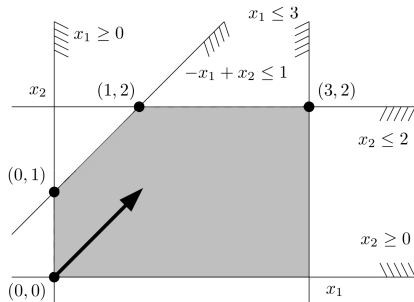
Nouvelle solution : $(0, 1, 0, 3, 1)$ qui donne $z = 1$.

Exemple d'exécution de l'algorithme du simplexe

x_3	$=$	1	$+$	x_1	$-$	x_2
x_4	$=$	3	$-$	x_1		
x_5	$=$	2			$-$	x_2
z	$=$			x_1	$+$	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

Nouvelle solution : $(0, 1, 0, 3, 1)$ qui donne $z = 1$.



Exemple d'exécution de l'algorithme du simplexe

x_3	=	1	+	x_1	-	x_2	
x_4	=	3	-	x_1			
x_5	=	2			-	x_2	
z	=			x_1	+	x_2	

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

Nouvelle solution : $(0, 1, 0, 3, 1)$ qui donne $z = 1$.

x_2 est maintenant une **variable basique**, mais plus x_3 ! On réécrit le tableau :

Exemple d'exécution de l'algorithme du simplexe

x_3	=	1	+	x_1	-	x_2
x_4	=	3	-	x_1		
x_5	=	2			-	x_2
z	=			x_1	+	x_2

Solution : (0, 0, 1, 3, 2), $z = 0$

Nouvelle solution : (0, 1, 0, 3, 1) qui donne $z = 1$.

x_2 est maintenant une **variable basique**, mais plus x_3 ! On réécrit le tableau :

$$x_2 = 1 + x_1 - x_3$$

Exemple d'exécution de l'algorithme du simplexe

x_3	=	1	+	x_1	-	x_2
x_4	=	3	-	x_1		
x_5	=	2			-	x_2
z	=			x_1	+	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

Nouvelle solution : $(0, 1, 0, 3, 1)$ qui donne $z = 1$.

x_2 est maintenant une **variable basique**, mais plus x_3 ! On réécrit le tableau :

$$x_2 = 1 + x_1 - x_3$$

$$x_4 = 3 - x_1$$

Exemple d'exécution de l'algorithme du simplexe

x_3	$=$	1	$+$	x_1	$-$	x_2
x_4	$=$	3	$-$	x_1		
x_5	$=$	2			$-$	x_2
z	$=$			x_1	$+$	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

Nouvelle solution : $(0, 1, 0, 3, 1)$ qui donne $z = 1$.

x_2 est maintenant une **variable basique**, mais plus x_3 ! On réécrit le tableau :

$$x_2 = 1 + x_1 - x_3$$

$$x_4 = 3 - x_1$$

$$x_5 = 2 - x_2 = 2 - (1 + x_1 - x_3) = 1 - x_1 + x_3$$

Exemple d'exécution de l'algorithme du simplexe

x_3	$=$	1	$+$	x_1	$-$	x_2
x_4	$=$	3	$-$	x_1		
x_5	$=$	2			$-$	x_2
z	$=$			x_1	$+$	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

Nouvelle solution : $(0, 1, 0, 3, 1)$ qui donne $z = 1$.

x_2 est maintenant une **variable basique**, mais plus x_3 ! On réécrit le tableau :

$$x_2 = 1 + x_1 - x_3$$

$$x_4 = 3 - x_1$$

$$x_5 = 2 - x_2 = 2 - (1 + x_1 - x_3) = 1 - x_1 + x_3$$

$$z = x_1 + x_2 = x_1 + (1 + x_1 - x_3) = 1 + 2x_1 - x_3$$

Exemple d'exécution de l'algorithme du simplexe

x_3	=	1	+	x_1	-	x_2
x_4	=	3	-	x_1		
x_5	=	2			-	x_2
z	=			x_1	+	x_2

Solution : (0, 0, 1, 3, 2), $z = 0$

x_2	=	1	+	x_1	-	x_3
x_4	=	3	-	x_1		
x_5	=	1	-	x_1	+	x_3
z	=	1	+	$2x_1$	-	x_3

Solution : (0, 1, 0, 3, 1), $z = 1$

Nouvelle solution : (0, 1, 0, 3, 1) qui donne $z = 1$.

x_2 est maintenant une **variable basique**, mais plus x_3 ! On réécrit le tableau :

$$x_2 = 1 + x_1 - x_3$$

$$x_4 = 3 - x_1$$

$$x_5 = 2 - x_2 = 2 - (1 + x_1 - x_3) = 1 - x_1 + x_3$$

$$z = x_1 + x_2 = x_1 + (1 + x_1 - x_3) = 1 + 2x_1 - x_3$$

Exemple d'exécution de l'algorithme du simplexe

x_3	=	1	+	x_1	-	x_2
x_4	=	3	-	x_1		
x_5	=	2			-	x_2
z	=			x_1	+	x_2

Solution : (0, 0, 1, 3, 2), $z = 0$

x_2	=	1	+	x_1	-	x_3
x_4	=	3	-	x_1		
x_5	=	1	-	x_1	+	x_3
z	=	1	+	$2x_1$	-	x_3

Solution : (0, 1, 0, 3, 1), $z = 1$

Exemple d'exécution de l'algorithme du simplexe

x_3	=	1	+	x_1	-	x_2
x_4	=	3	-	x_1		
x_5	=	2			-	x_2
z	=			x_1	+	x_2

Solution : (0, 0, 1, 3, 2), $z = 0$

x_2	=	1	+	x_1	-	x_3
x_4	=	3	-	x_1		
x_5	=	1	-	x_1	+	x_3
z	=	1	+	$2x_1$	-	x_3

Solution : (0, 1, 0, 3, 1), $z = 1$

Quel **pivot** choisir? → x_1 , car x_3 ferait baisser la fonction objectif.

Exemple d'exécution de l'algorithme du simplexe

x_3	=	1	+	x_1	-	x_2	
x_4	=	3	-	x_1			
x_5	=	2			-	x_2	
z	=			x_1	+	x_2	

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

x_2	=	1	+	x_1	-	x_3	
x_4	=	3	-	x_1			
x_5	=	1	-	x_1	+	x_3	
z	=	1	+	$2x_1$	-	x_3	

Solution : $(0, 1, 0, 3, 1)$, $z = 1$

Quel **pivot** choisir? → x_1 , car x_3 ferait baisser la fonction objectif.

De combien peut-on augmenter x_1 ?

Regardons nos contraintes (avec $x_3 = 0$) :

Exemple d'exécution de l'algorithme du simplexe

x_3	$=$	1	$+$	x_1	$-$	x_2
x_4	$=$	3	$-$	x_1		
x_5	$=$	2			$-$	x_2
z	$=$			x_1	$+$	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

x_2	$=$	1	$+$	x_1	$-$	x_3
x_4	$=$	3	$-$	x_1		
x_5	$=$	1	$-$	x_1	$+$	x_3
z	$=$	1	$+$	$2x_1$	$-$	x_3

Solution : $(0, 1, 0, 3, 1)$, $z = 1$

Quel **pivot** choisir? $\rightarrow x_1$, car x_3 ferait baisser la fonction objectif.

De combien peut-on augmenter x_1 ?

Regardons nos contraintes (avec $x_3 = 0$) :

$$x_2 = 1 + x_1 - x_3 \geq 0 \text{ donc } 1 + x_1 \geq 0 \text{ et } x_1 \geq -1$$

Exemple d'exécution de l'algorithme du simplexe

x_3	$=$	1	$+$	x_1	$-$	x_2
x_4	$=$	3	$-$	x_1		
x_5	$=$	2			$-$	x_2
z	$=$			x_1	$+$	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

x_2	$=$	1	$+$	x_1	$-$	x_3
x_4	$=$	3	$-$	x_1		
x_5	$=$	1	$-$	x_1	$+$	x_3
z	$=$	1	$+$	$2x_1$	$-$	x_3

Solution : $(0, 1, 0, 3, 1)$, $z = 1$

Quel **pivot** choisir? $\rightarrow x_1$, car x_3 ferait baisser la fonction objectif.

De combien peut-on augmenter x_1 ?

Regardons nos contraintes (avec $x_3 = 0$) :

$$x_2 = 1 + x_1 - x_3 \geq 0 \text{ donc } 1 + x_1 \geq 0 \text{ et } x_1 \geq -1$$

$$x_4 = 3 - x_1 \geq 0 \text{ donc } 3 \geq x_1$$

Exemple d'exécution de l'algorithme du simplexe

x_3	$=$	1	$+$	x_1	$-$	x_2
x_4	$=$	3	$-$	x_1		
x_5	$=$	2			$-$	x_2
z	$=$			x_1	$+$	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

x_2	$=$	1	$+$	x_1	$-$	x_3
x_4	$=$	3	$-$	x_1		
x_5	$=$	1	$-$	x_1	$+$	x_3
z	$=$	1	$+$	$2x_1$	$-$	x_3

Solution : $(0, 1, 0, 3, 1)$, $z = 1$

Quel **pivot** choisir? $\rightarrow x_1$, car x_3 ferait baisser la fonction objectif.

De combien peut-on augmenter x_1 ?

Regardons nos contraintes (avec $x_3 = 0$) :

$$x_2 = 1 + x_1 - x_3 \geq 0 \text{ donc } 1 + x_1 \geq 0 \text{ et } x_1 \geq -1$$

$$x_4 = 3 - x_1 \geq 0 \text{ donc } 3 \geq x_1$$

$$x_5 = 1 - x_1 + x_3 \geq 0 \text{ donc } 1 - x_1 \geq 0 \text{ et } 1 \geq x_1$$

Exemple d'exécution de l'algorithme du simplexe

x_3	$=$	1	$+$	x_1	$-$	x_2
x_4	$=$	3	$-$	x_1		
x_5	$=$	2			$-$	x_2
z	$=$			x_1	$+$	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

x_2	$=$	1	$+$	x_1	$-$	x_3
x_4	$=$	3	$-$	x_1		
x_5	$=$	1	$-$	x_1	$+$	x_3
z	$=$	1	$+$	$2x_1$	$-$	x_3

Solution : $(0, 1, 0, 3, 1)$, $z = 1$

Quel **pivot** choisir? $\rightarrow x_1$, car x_3 ferait baisser la fonction objectif.

De combien peut-on augmenter x_1 ?

Regardons nos contraintes (avec $x_3 = 0$) :

$$x_2 = 1 + x_1 - x_3 \geq 0 \text{ donc } 1 + x_1 \geq 0 \text{ et } x_1 \geq -1$$

$$x_4 = 3 - x_1 \geq 0 \text{ donc } 3 \geq x_1$$

$$x_5 = 1 - x_1 + x_3 \geq 0 \text{ donc } 1 - x_1 \geq 0 \text{ et } 1 \geq x_1$$

On **augmente** x_1 **au maximum autorisé** : $x_1 = 1$, et on garde $x_3 = 0$.

Exemple d'exécution de l'algorithme du simplexe

x_3	=	1	+	x_1	-	x_2	
x_4	=	3	-	x_1			
x_5	=	2			-	x_2	
z	=			x_1	+	x_2	

Solution : (0, 0, 1, 3, 2), $z = 0$

x_2	=	1	+	x_1	-	x_3	
x_4	=	3	-	x_1			
x_5	=	1	-	x_1	+	x_3	
z	=	1	+	$2x_1$	-	x_3	

Solution : (0, 1, 0, 3, 1), $z = 1$

Quel **pivot** choisir? → x_1 , car x_3 ferait baisser la fonction objectif.

De combien peut-on augmenter x_1 ?

Regardons nos contraintes (avec $x_3 = 0$) :

$$x_2 = 1 + x_1 - x_3 \geq 0 \text{ donc } 1 + x_1 \geq 0 \text{ et } x_1 \geq -1$$

$$x_4 = 3 - x_1 \geq 0 \text{ donc } 3 \geq x_1$$

$$x_5 = 1 - x_1 + x_3 \geq 0 \text{ donc } 1 - x_1 \geq 0 \text{ et } 1 \geq x_1$$

On **augmente** x_1 **au maximum autorisé** : $x_1 = 1$, et on garde $x_3 = 0$.

On calcule x_2 , x_4 et x_5 grâce au tableau : $x_2 = 2$, $x_4 = 2$, $x_5 = 0$

Nouvelle solution : (1, 2, 0, 2, 0) qui donne $z = 3$.

Exemple d'exécution de l'algorithme du simplexe

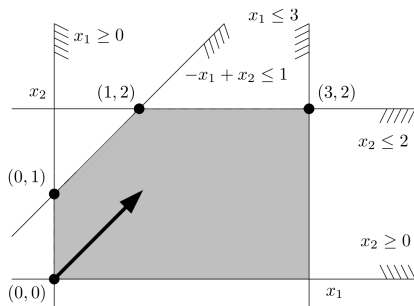
x_3	$=$	1	$+$	x_1	$-$	x_2
x_4	$=$	3	$-$	x_1		
x_5	$=$	2			$-$	x_2
z	$=$			x_1	$+$	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

x_2	$=$	1	$+$	x_1	$-$	x_3
x_4	$=$	3	$-$	x_1		
x_5	$=$	1	$-$	x_1	$+$	x_3
z	$=$	1	$+$	$2x_1$	$-$	x_3

Solution : $(0, 1, 0, 3, 1)$, $z = 1$

Nouvelle solution : $(1, 2, 0, 2, 0)$ qui donne $z = 3$.



Exemple d'exécution de l'algorithme du simplexe

x_3	=	1	+	x_1	-	x_2	
x_4	=	3	-	x_1			
x_5	=	2			-	x_2	
z	=			x_1	+	x_2	

Solution : (0, 0, 1, 3, 2), $z = 0$

x_2	=	1	+	x_1	-	x_3	
x_4	=	3	-	x_1			
x_5	=	1	-	x_1	+	x_3	
z	=	1	+	$2x_1$	-	x_3	

Solution : (0, 1, 0, 3, 1), $z = 1$

Nouvelle solution : (1, 2, 0, 2, 0) qui donne $z = 3$.

x_1 est maintenant une **variable basique**, mais plus x_5 ! On réécrit le tableau :

Exemple d'exécution de l'algorithme du simplexe

x_3	=	1	+	x_1	-	x_2
x_4	=	3	-	x_1		
x_5	=	2			-	x_2
z	=			x_1	+	x_2

Solution : (0, 0, 1, 3, 2), $z = 0$

x_2	=	1	+	x_1	-	x_3
x_4	=	3	-	x_1		
x_5	=	1	-	x_1	+	x_3
z	=	1	+	$2x_1$	-	x_3

Solution : (0, 1, 0, 3, 1), $z = 1$

Nouvelle solution : (1, 2, 0, 2, 0) qui donne $z = 3$.

x_1 est maintenant une **variable basique**, mais plus x_5 ! On réécrit le tableau :

$$x_1 = 1 + x_3 - x_5$$

Exemple d'exécution de l'algorithme du simplexe

x_3	=	1	+	x_1	-	x_2
x_4	=	3	-	x_1		
x_5	=	2			-	x_2
z	=			x_1	+	x_2

Solution : (0, 0, 1, 3, 2), $z = 0$

x_2	=	1	+	x_1	-	x_3
x_4	=	3	-	x_1		
x_5	=	1	-	x_1	+	x_3
z	=	1	+	$2x_1$	-	x_3

Solution : (0, 1, 0, 3, 1), $z = 1$

Nouvelle solution : (1, 2, 0, 2, 0) qui donne $z = 3$.

x_1 est maintenant une **variable basique**, mais plus x_5 ! On réécrit le tableau :

$$x_1 = 1 + x_3 - x_5$$

$$x_2 = 1 + x_1 - x_3 = 1 + (1 + x_3 - x_5) - x_3 = 2 + x_5$$

Exemple d'exécution de l'algorithme du simplexe

x_3	$=$	1	$+$	x_1	$-$	x_2
x_4	$=$	3	$-$	x_1		
x_5	$=$	2			$-$	x_2
z	$=$			x_1	$+$	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

x_2	$=$	1	$+$	x_1	$-$	x_3
x_4	$=$	3	$-$	x_1		
x_5	$=$	1	$-$	x_1	$+$	x_3
z	$=$	1	$+$	$2x_1$	$-$	x_3

Solution : $(0, 1, 0, 3, 1)$, $z = 1$

Nouvelle solution : $(1, 2, 0, 2, 0)$ qui donne $z = 3$.

x_1 est maintenant une **variable basique**, mais plus x_5 ! On réécrit le tableau :

$$x_1 = 1 + x_3 - x_5$$

$$x_2 = 1 + x_1 - x_3 = 1 + (1 + x_3 - x_5) - x_3 = 2 + x_5$$

$$x_4 = 3 - x_1 = 3 - (1 + x_3 - x_5) = 2 - x_3 + x_5$$

Exemple d'exécution de l'algorithme du simplexe

x_3	$=$	1	$+$	x_1	$-$	x_2
x_4	$=$	3	$-$	x_1		
x_5	$=$	2			$-$	x_2
z	$=$			x_1	$+$	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

x_2	$=$	1	$+$	x_1	$-$	x_3
x_4	$=$	3	$-$	x_1		
x_5	$=$	1	$-$	x_1	$+$	x_3
z	$=$	1	$+$	$2x_1$	$-$	x_3

Solution : $(0, 1, 0, 3, 1)$, $z = 1$

Nouvelle solution : $(1, 2, 0, 2, 0)$ qui donne $z = 3$.

x_1 est maintenant une **variable basique**, mais plus x_5 ! On réécrit le tableau :

$$x_1 = 1 + x_3 - x_5$$

$$x_2 = 1 + x_1 - x_3 = 1 + (1 + x_3 - x_5) - x_3 = 2 + x_5$$

$$x_4 = 3 - x_1 = 3 - (1 + x_3 - x_5) = 2 - x_3 + x_5$$

$$z = 1 + 2x_1 - x_3 = 1 + 2(1 + x_3 - x_5) - x_3 = 3 + x_3 - 2x_5$$

Exemple d'exécution de l'algorithme du simplexe

x_3	$=$	1	$+$	x_1	$-$	x_2
x_4	$=$	3	$-$	x_1		
x_5	$=$	2			$-$	x_2
z	$=$			x_1	$+$	x_2

Solution : (0, 0, 1, 3, 2), $z = 0$

x_2	$=$	1	$+$	x_1	$-$	x_3
x_4	$=$	3	$-$	x_1		
x_5	$=$	1	$-$	x_1	$+$	x_3
z	$=$	1	$+$	$2x_1$	$-$	x_3

Solution : (0, 1, 0, 3, 1), $z = 1$

Nouvelle solution : (1, 2, 0, 2, 0) qui donne $z = 3$.

x_1 est maintenant une **variable basique**, mais plus x_5 ! On réécrit le tableau :

$$x_1 = 1 + x_3 - x_5$$

$$x_2 = 1 + x_1 - x_3 = 1 + (1 + x_3 - x_5) - x_3 = 2 + x_5$$

$$x_4 = 3 - x_1 = 3 - (1 + x_3 - x_5) = 2 - x_3 + x_5$$

$$z = 1 + 2x_1 - x_3 = 1 + 2(1 + x_3 - x_5) - x_3 = 3 + x_3 - 2x_5$$

x_1	$=$	1	$+$	x_3	$-$	x_5
x_2	$=$	2			$+$	x_5
x_4	$=$	2	$-$	x_3	$+$	x_5
z	$=$	3	$+$	x_3	$-$	$2x_5$

Solution : (1, 2, 0, 2, 0), $z = 3$

Exemple d'exécution de l'algorithme du simplexe

x_3	$=$	1	$+$	x_1	$-$	x_2
x_4	$=$	3	$-$	x_1		
x_5	$=$	2			$-$	x_2
z	$=$			x_1	$+$	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

x_2	$=$	1	$+$	x_1	$-$	x_3
x_4	$=$	3	$-$	x_1		
x_5	$=$	1	$-$	x_1	$+$	x_3
z	$=$	1	$+$	$2x_1$	$-$	x_3

Solution : $(0, 1, 0, 3, 1)$, $z = 1$

x_1	$=$	1	$+$	x_3	$-$	x_5
x_2	$=$	2			$+$	x_5
x_4	$=$	2	$-$	x_3	$+$	x_5
z	$=$	3	$+$	x_3	$-$	$2x_5$

Solution : $(1, 2, 0, 2, 0)$, $z = 3$

Exemple d'exécution de l'algorithme du simplexe

x_3	$=$	1	$+$	x_1	$-$	x_2
x_4	$=$	3	$-$	x_1		
x_5	$=$	2			$-$	x_2
z	$=$			x_1	$+$	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

x_2	$=$	1	$+$	x_1	$-$	x_3
x_4	$=$	3	$-$	x_1		
x_5	$=$	1	$-$	x_1	$+$	x_3
z	$=$	1	$+$	$2x_1$	$-$	x_3

Solution : $(0, 1, 0, 3, 1)$, $z = 1$

x_1	$=$	1	$+$	x_3	$-$	x_5
x_2	$=$	2			$+$	x_5
x_4	$=$	2	$-$	x_3	$+$	x_5
z	$=$	3	$+$	x_3	$-$	$2x_5$

Solution : $(1, 2, 0, 2, 0)$, $z = 3$

Quel **pivot** choisir? $\rightarrow x_3$, car x_5 ferait baisser la fonction objectif.

Exemple d'exécution de l'algorithme du simplexe

x_3	=	1	+	x_1	-	x_2
x_4	=	3	-	x_1		
x_5	=	2			-	x_2
z	=			x_1	+	x_2

Solution : (0, 0, 1, 3, 2), $z = 0$

x_2	=	1	+	x_1	-	x_3
x_4	=	3	-	x_1		
x_5	=	1	-	x_1	+	x_3
z	=	1	+	$2x_1$	-	x_3

Solution : (0, 1, 0, 3, 1), $z = 1$

x_1	=	1	+	x_3	-	x_5
x_2	=	2			+	x_5
x_4	=	2	-	x_3	+	x_5
z	=	3	+	x_3	-	$2x_5$

Solution : (1, 2, 0, 2, 0), $z = 3$

Quel **pivot** choisir ? $\rightarrow x_3$, car x_5 ferait baisser la fonction objectif.

De combien peut-on augmenter x_3 ?

Regardons nos contraintes (avec $x_5 = 0$) :

Exemple d'exécution de l'algorithme du simplexe

x_3	$=$	1	$+$	x_1	$-$	x_2
x_4	$=$	3	$-$	x_1		
x_5	$=$	2			$-$	x_2
z	$=$			x_1	$+$	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

x_2	$=$	1	$+$	x_1	$-$	x_3
x_4	$=$	3	$-$	x_1		
x_5	$=$	1	$-$	x_1	$+$	x_3
z	$=$	1	$+$	$2x_1$	$-$	x_3

Solution : $(0, 1, 0, 3, 1)$, $z = 1$

x_1	$=$	1	$+$	x_3	$-$	x_5
x_2	$=$	2			$+$	x_5
x_4	$=$	2	$-$	x_3	$+$	x_5
z	$=$	3	$+$	x_3	$-$	$2x_5$

Solution : $(1, 2, 0, 2, 0)$, $z = 3$

Quel **pivot** choisir ? $\rightarrow x_3$, car x_5 ferait baisser la fonction objectif.

De combien peut-on augmenter x_3 ?

Regardons nos contraintes (avec $x_5 = 0$) :

$$x_1 = 1 + x_3 - x_5 \geq 0 \text{ donc } 1 + x_3 \geq 0 \text{ et } x_3 \geq -1$$

Exemple d'exécution de l'algorithme du simplexe

x_3	$=$	1	$+$	x_1	$-$	x_2
x_4	$=$	3	$-$	x_1		
x_5	$=$	2			$-$	x_2
z	$=$			x_1	$+$	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

x_2	$=$	1	$+$	x_1	$-$	x_3
x_4	$=$	3	$-$	x_1		
x_5	$=$	1	$-$	x_1	$+$	x_3
z	$=$	1	$+$	$2x_1$	$-$	x_3

Solution : $(0, 1, 0, 3, 1)$, $z = 1$

x_1	$=$	1	$+$	x_3	$-$	x_5
x_2	$=$	2			$+$	x_5
x_4	$=$	2	$-$	x_3	$+$	x_5
z	$=$	3	$+$	x_3	$-$	$2x_5$

Solution : $(1, 2, 0, 2, 0)$, $z = 3$

Quel **pivot** choisir? $\rightarrow x_3$, car x_5 ferait baisser la fonction objectif.

De combien peut-on augmenter x_3 ?

Regardons nos contraintes (avec $x_5 = 0$) :

$$x_1 = 1 + x_3 - x_5 \geq 0 \text{ donc } 1 + x_3 \geq 0 \text{ et } x_3 \geq -1$$

$$x_2 = 2 + x_5 \geq 0 \rightarrow \text{aucune influence sur } x_3$$

Exemple d'exécution de l'algorithme du simplexe

x_3	$=$	1	$+$	x_1	$-$	x_2
x_4	$=$	3	$-$	x_1		
x_5	$=$	2			$-$	x_2
z	$=$			x_1	$+$	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

x_2	$=$	1	$+$	x_1	$-$	x_3
x_4	$=$	3	$-$	x_1		
x_5	$=$	1	$-$	x_1	$+$	x_3
z	$=$	1	$+$	$2x_1$	$-$	x_3

Solution : $(0, 1, 0, 3, 1)$, $z = 1$

x_1	$=$	1	$+$	x_3	$-$	x_5
x_2	$=$	2			$+$	x_5
x_4	$=$	2	$-$	x_3	$+$	x_5
z	$=$	3	$+$	x_3	$-$	$2x_5$

Solution : $(1, 2, 0, 2, 0)$, $z = 3$

Quel **pivot** choisir? $\rightarrow x_3$, car x_5 ferait baisser la fonction objectif.

De combien peut-on augmenter x_3 ?

Regardons nos contraintes (avec $x_5 = 0$) :

$$x_1 = 1 + x_3 - x_5 \geq 0 \text{ donc } 1 + x_3 \geq 0 \text{ et } x_3 \geq -1$$

$$x_2 = 2 + x_5 \geq 0 \rightarrow \text{aucune influence sur } x_3$$

$$x_4 = 2 - x_3 + x_5 \geq 0 \text{ donc } 2 - x_3 \geq 0 \text{ et } 2 \geq x_3$$

Exemple d'exécution de l'algorithme du simplexe

x_3	=	1	+	x_1	-	x_2
x_4	=	3	-	x_1		
x_5	=	2			-	x_2
z	=			x_1	+	x_2

Solution : (0, 0, 1, 3, 2), $z = 0$

x_2	=	1	+	x_1	-	x_3
x_4	=	3	-	x_1		
x_5	=	1	-	x_1	+	x_3
z	=	1	+	$2x_1$	-	x_3

Solution : (0, 1, 0, 3, 1), $z = 1$

x_1	=	1	+	x_3	-	x_5
x_2	=	2			+	x_5
x_4	=	2	-	x_3	+	x_5
z	=	3	+	x_3	-	$2x_5$

Solution : (1, 2, 0, 2, 0), $z = 3$

Quel **pivot** choisir? → x_3 , car x_5 ferait baisser la fonction objectif.

De combien peut-on augmenter x_3 ?

Regardons nos contraintes (avec $x_5 = 0$) :

$$x_1 = 1 + x_3 - x_5 \geq 0 \text{ donc } 1 + x_3 \geq 0 \text{ et } x_3 \geq -1$$

$$x_2 = 2 + x_5 \geq 0 \rightarrow \text{aucune influence sur } x_3$$

$$x_4 = 2 - x_3 + x_5 \geq 0 \text{ donc } 2 - x_3 \geq 0 \text{ et } 2 \geq x_3$$

On **augmente** x_3 **au maximum autorisé** : $x_3 = 2$, et on garde $x_5 = 0$.

Exemple d'exécution de l'algorithme du simplexe

x_3	$=$	1	+	x_1	-	x_2
x_4	$=$	3	-	x_1		
x_5	$=$	2			-	x_2
z	$=$			x_1	+	x_2

Solution : (0, 0, 1, 3, 2), $z = 0$

x_2	$=$	1	+	x_1	-	x_3
x_4	$=$	3	-	x_1		
x_5	$=$	1	-	x_1	+	x_3
z	$=$	1	+	$2x_1$	-	x_3

Solution : (0, 1, 0, 3, 1), $z = 1$

x_1	$=$	1	+	x_3	-	x_5
x_2	$=$	2			+	x_5
x_4	$=$	2	-	x_3	+	x_5
z	$=$	3	+	x_3	-	$2x_5$

Solution : (1, 2, 0, 2, 0), $z = 3$

Quel **pivot** choisir? $\rightarrow x_3$, car x_5 ferait baisser la fonction objectif.

De combien peut-on augmenter x_3 ?

Regardons nos contraintes (avec $x_5 = 0$) :

$$x_1 = 1 + x_3 - x_5 \geq 0 \text{ donc } 1 + x_3 \geq 0 \text{ et } x_3 \geq -1$$

$$x_2 = 2 + x_5 \geq 0 \rightarrow \text{aucune influence sur } x_3$$

$$x_4 = 2 - x_3 + x_5 \geq 0 \text{ donc } 2 - x_3 \geq 0 \text{ et } 2 \geq x_3$$

On **augmente** x_3 **au maximum autorisé** : $x_3 = 2$, et on garde $x_5 = 0$.

On calcule x_1 , x_2 et x_4 grâce au tableau : $x_1 = 3$, $x_2 = 2$, $x_4 = 0$

Nouvelle solution : (3, 2, 2, 0, 0) qui donne $z = 5$.

Exemple d'exécution de l'algorithme du simplexe

x_3	=	1	+	x_1	-	x_2	
x_4	=	3	-	x_1			
x_5	=	2			-	x_2	
z	=			x_1	+	x_2	

Solution : (0, 0, 1, 3, 2), $z = 0$

x_2	=	1	+	x_1	-	x_3	
x_4	=	3	-	x_1			
x_5	=	1	-	x_1	+	x_3	
z	=	1	+	$2x_1$	-	x_3	

Solution : (0, 1, 0, 3, 1), $z = 1$

x_1	=	1	+	x_3	-	x_5	
x_2	=	2			+	x_5	
x_4	=	2	-	x_3	+	x_5	
z	=	3	+	x_3	-	$2x_5$	

Solution : (1, 2, 0, 2, 0), $z = 3$

Nouvelle solution : (3, 2, 2, 0, 0) qui donne $z = 5$.

x_3 est maintenant une **variable basique**, mais plus x_4 ! On réécrit le tableau :

Exemple d'exécution de l'algorithme du simplexe

x_3	=	1	+	x_1	-	x_2	
x_4	=	3	-	x_1			
x_5	=	2			-	x_2	
z	=			x_1	+	x_2	

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

x_2	=	1	+	x_1	-	x_3	
x_4	=	3	-	x_1			
x_5	=	1	-	x_1	+	x_3	
z	=	1	+	$2x_1$	-	x_3	

Solution : $(0, 1, 0, 3, 1)$, $z = 1$

x_1	=	1	+	x_3	-	x_5	
x_2	=	2			+	x_5	
x_4	=	2	-	x_3	+	x_5	
z	=	3	+	x_3	-	$2x_5$	

Solution : $(1, 2, 0, 2, 0)$, $z = 3$

Nouvelle solution : $(3, 2, 2, 0, 0)$ qui donne $z = 5$.

x_3 est maintenant une **variable basique**, mais plus x_4 ! On réécrit le tableau :

$$x_3 = 2 - x_4 + x_5$$

Exemple d'exécution de l'algorithme du simplexe

x_3	$=$	1	$+$	x_1	$-$	x_2
x_4	$=$	3	$-$	x_1		
x_5	$=$	2			$-$	x_2
z	$=$			x_1	$+$	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

x_2	$=$	1	$+$	x_1	$-$	x_3
x_4	$=$	3	$-$	x_1		
x_5	$=$	1	$-$	x_1	$+$	x_3
z	$=$	1	$+$	$2x_1$	$-$	x_3

Solution : $(0, 1, 0, 3, 1)$, $z = 1$

x_1	$=$	1	$+$	x_3	$-$	x_5
x_2	$=$	2			$+$	x_5
x_4	$=$	2	$-$	x_3	$+$	x_5
z	$=$	3	$+$	x_3	$-$	$2x_5$

Solution : $(1, 2, 0, 2, 0)$, $z = 3$

Nouvelle solution : $(3, 2, 2, 0, 0)$ qui donne $z = 5$.

x_3 est maintenant une **variable basique**, mais plus x_4 ! On réécrit le tableau :

$$x_3 = 2 - x_4 + x_5$$

$$x_1 = 1 + x_3 - x_5 = 1 + (2 - x_4 + x_5) - x_5 = 3 - x_4$$

Exemple d'exécution de l'algorithme du simplexe

x_3	$=$	1	$+$	x_1	$-$	x_2
x_4	$=$	3	$-$	x_1		
x_5	$=$	2			$-$	x_2
z	$=$			x_1	$+$	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

x_2	$=$	1	$+$	x_1	$-$	x_3
x_4	$=$	3	$-$	x_1		
x_5	$=$	1	$-$	x_1	$+$	x_3
z	$=$	1	$+$	$2x_1$	$-$	x_3

Solution : $(0, 1, 0, 3, 1)$, $z = 1$

x_1	$=$	1	$+$	x_3	$-$	x_5
x_2	$=$	2			$+$	x_5
x_4	$=$	2	$-$	x_3	$+$	x_5
z	$=$	3	$+$	x_3	$-$	$2x_5$

Solution : $(1, 2, 0, 2, 0)$, $z = 3$

Nouvelle solution : $(3, 2, 2, 0, 0)$ qui donne $z = 5$.

x_3 est maintenant une **variable basique**, mais plus x_4 ! On réécrit le tableau :

$$x_3 = 2 - x_4 + x_5$$

$$x_1 = 1 + x_3 - x_5 = 1 + (2 - x_4 + x_5) - x_5 = 3 - x_4$$

$$x_2 = 2 - x_5$$

Exemple d'exécution de l'algorithme du simplexe

x_3	$=$	1	$+$	x_1	$-$	x_2
x_4	$=$	3	$-$	x_1		
x_5	$=$	2			$-$	x_2
z	$=$			x_1	$+$	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

x_2	$=$	1	$+$	x_1	$-$	x_3
x_4	$=$	3	$-$	x_1		
x_5	$=$	1	$-$	x_1	$+$	x_3
z	$=$	1	$+$	$2x_1$	$-$	x_3

Solution : $(0, 1, 0, 3, 1)$, $z = 1$

x_1	$=$	1	$+$	x_3	$-$	x_5
x_2	$=$	2			$+$	x_5
x_4	$=$	2	$-$	x_3	$+$	x_5
z	$=$	3	$+$	x_3	$-$	$2x_5$

Solution : $(1, 2, 0, 2, 0)$, $z = 3$

Nouvelle solution : $(3, 2, 2, 0, 0)$ qui donne $z = 5$.

x_3 est maintenant une **variable basique**, mais plus x_4 ! On réécrit le tableau :

$$x_3 = 2 - x_4 + x_5$$

$$x_1 = 1 + x_3 - x_5 = 1 + (2 - x_4 + x_5) - x_5 = 3 - x_4$$

$$x_2 = 2 - x_5$$

$$z = 3 + x_3 - 2x_5 = 3 + (2 - x_4 + x_5) - 2x_5 = 5 - x_4 - x_5$$

Exemple d'exécution de l'algorithme du simplexe

x_3	=	1	+	x_1	-	x_2
x_4	=	3	-	x_1		
x_5	=	2			-	x_2
z	=			x_1	+	x_2

Solution : (0, 0, 1, 3, 2), $z = 0$

x_2	=	1	+	x_1	-	x_3
x_4	=	3	-	x_1		
x_5	=	1	-	x_1	+	x_3
z	=	1	+	$2x_1$	-	x_3

Solution : (0, 1, 0, 3, 1), $z = 1$

x_1	=	1	+	x_3	-	x_5
x_2	=	2			+	x_5
x_4	=	2	-	x_3	+	x_5
z	=	3	+	x_3	-	$2x_5$

Solution : (1, 2, 0, 2, 0), $z = 3$

x_3	=	2	-	x_4	+	x_5
x_1	=	3	-	x_4		
x_2	=	2			+	x_5
z	=	5	-	x_4	-	x_5

Solution : (3, 2, 2, 0, 0), $z = 5$

Nouvelle solution : (3, 2, 2, 0, 0) qui donne $z = 5$.

x_3 est maintenant une **variable basique**, mais plus x_4 ! On réécrit le tableau :

$$x_3 = 2 - x_4 + x_5$$

$$x_1 = 1 + x_3 - x_5 = 1 + (2 - x_4 + x_5) - x_5 = 3 - x_4$$

$$x_2 = 2 - x_5$$

$$z = 3 + x_3 - 2x_5 = 3 + (2 - x_4 + x_5) - 2x_5 = 5 - x_4 - x_5$$

Exemple d'exécution de l'algorithme du simplexe

x_3	=	1	+	x_1	-	x_2
x_4	=	3	-	x_1		
x_5	=	2			-	x_2
z	=			x_1	+	x_2

Solution : (0, 0, 1, 3, 2), $z = 0$

x_1	=	1	+	x_3	-	x_5
x_2	=	2			+	x_5
x_4	=	2	-	x_3	+	x_5
z	=	3	+	x_3	-	$2x_5$

Solution : (1, 2, 0, 2, 0), $z = 3$

x_2	=	1	+	x_1	-	x_3
x_4	=	3	-	x_1		
x_5	=	1	-	x_1	+	x_3
z	=	1	+	$2x_1$	-	x_3

Solution : (0, 1, 0, 3, 1), $z = 1$

x_3	=	2	-	x_4	+	x_5
x_1	=	3	-	x_4		
x_2	=	2			+	x_5
z	=	5	-	x_4	-	x_5

Solution : (3, 2, 2, 0, 0), $z = 5$

Exemple d'exécution de l'algorithme du simplexe

x_3	=	1	+	x_1	-	x_2
x_4	=	3	-	x_1		
x_5	=	2			-	x_2
z	=			x_1	+	x_2

Solution : (0, 0, 1, 3, 2), $z = 0$

x_1	=	1	+	x_3	-	x_5
x_2	=	2			+	x_5
x_4	=	2	-	x_3	+	x_5
z	=	3	+	x_3	-	$2x_5$

Solution : (1, 2, 0, 2, 0), $z = 3$

x_2	=	1	+	x_1	-	x_3
x_4	=	3	-	x_1		
x_5	=	1	-	x_1	+	x_3
z	=	1	+	$2x_1$	-	x_3

Solution : (0, 1, 0, 3, 1), $z = 1$

x_3	=	2	-	x_4	+	x_5
x_1	=	3	-	x_4		
x_2	=	2			+	x_5
z	=	5	-	x_4	-	x_5

Solution : (3, 2, 2, 0, 0), $z = 5$

Quel **pivot** choisir?

Exemple d'exécution de l'algorithme du simplexe

x_3	$=$	1	$+$	x_1	$-$	x_2
x_4	$=$	3	$-$	x_1		
x_5	$=$	2			$-$	x_2
z	$=$			x_1	$+$	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

x_2	$=$	1	$+$	x_1	$-$	x_3
x_4	$=$	3	$-$	x_1		
x_5	$=$	1	$-$	x_1	$+$	x_3
z	$=$	1	$+$	$2x_1$	$-$	x_3

Solution : $(0, 1, 0, 3, 1)$, $z = 1$

x_1	$=$	1	$+$	x_3	$-$	x_5
x_2	$=$	2			$+$	x_5
x_4	$=$	2	$-$	x_3	$+$	x_5
z	$=$	3	$+$	x_3	$-$	$2x_5$

Solution : $(1, 2, 0, 2, 0)$, $z = 3$

x_3	$=$	2	$-$	x_4	$+$	x_5
x_1	$=$	3	$-$	x_4		
x_2	$=$	2			$+$	x_5
z	$=$	5	$-$	x_4	$-$	x_5

Solution : $(3, 2, 2, 0, 0)$, $z = 5$

Quel **pivot** choisir ?

Aucun, car on baisserait la valeur de la fonction objectif.

L'algorithme est terminé !

La **solution optimale** est $(x_1, x_2, x_3, x_4, x_5) = (3, 2, 2, 0, 0)$ avec $z = 5$.

Exemple d'exécution de l'algorithme du simplexe

x_3	$=$	1	$+$	x_1	$-$	x_2
x_4	$=$	3	$-$	x_1		
x_5	$=$	2			$-$	x_2
z	$=$			x_1	$+$	x_2

Solution : $(0, 0, 1, 3, 2)$, $z = 0$

x_2	$=$	1	$+$	x_1	$-$	x_3
x_4	$=$	3	$-$	x_1		
x_5	$=$	1	$-$	x_1	$+$	x_3
z	$=$	1	$+$	$2x_1$	$-$	x_3

Solution : $(0, 1, 0, 3, 1)$, $z = 1$

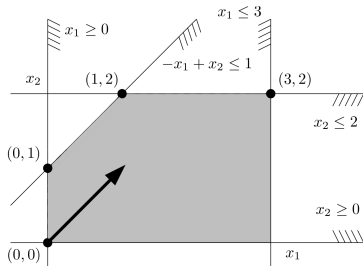
x_1	$=$	1	$+$	x_3	$-$	x_5
x_2	$=$	2			$+$	x_5
x_4	$=$	2	$-$	x_3	$+$	x_5
z	$=$	3	$+$	x_3	$-$	$2x_5$

Solution : $(1, 2, 0, 2, 0)$, $z = 3$

x_3	$=$	2	$-$	x_4	$+$	x_5
x_1	$=$	3	$-$	x_4		
x_2	$=$	2			$+$	x_5
z	$=$	5	$-$	x_4	$-$	x_5

Solution : $(3, 2, 2, 0, 0)$, $z = 5$

La **solution optimale** est $(x_1, x_2, x_3, x_4, x_5) = (3, 2, 2, 0, 0)$ avec $z = 5$.



Résumé de l'algorithme

0. On part d'un PL en **forme standard**
1. On trouve une solution non-optimale en un point du polytope associé à notre PL
2. Tant qu'on peut, on évolue vers une **solution proche** qui **améliore** la fonction objectif. On réitère :
 - a. Déterminer les variables basiques (non-nulles dans la solution courante)
 - b. Écrire le tableau de simplexe qui exprime les variables basiques et la fonction objectif z en fonction des variables non-basiques
 - c. Trouver une variable non-basique à augmenter pour augmenter z : c'est le **pivot**
 - d. Si aucun pivot n'existe (on ne peut plus augmenter z), on a trouvé la solution optimale! STOP
 - e. Sinon, l'augmenter au maximum possible en fonction des contraintes de type $x_i \geq 0$ avec x_i les variables basiques (s'il n'y a pas de restriction sur le pivot, le PL est non borné : STOP)
 - f. Calculer les nouvelles valeurs des variables : on obtient une nouvelle solution.

Pourquoi le nom "simplexe" ?

Un **simplexe** dans un espace à n dimensions, c'est le polytope le plus simple dans cet espace.

→ En 2D : triangle, en 3D : tétraèdre, etc.

Pourquoi le nom "simplexe" ?

Un **simplexe** dans un espace à n dimensions, c'est le polytope le plus simple dans cet espace.

→ En 2D : triangle, en 3D : tétraèdre, etc.

Interprétation géométrique de l'algorithme :

On est dans un espace à n dimensions, avec m variables basiques et $n - m$ variables non-basiques.

Pourquoi le nom "simplexe" ?

Un **simplexe** dans un espace à n dimensions, c'est le polytope le plus simple dans cet espace.

→ En 2D : triangle, en 3D : tétraèdre, etc.

Interprétation géométrique de l'algorithme :

On est dans un espace à n dimensions, avec m variables basiques et $n - m$ variables non-basiques.

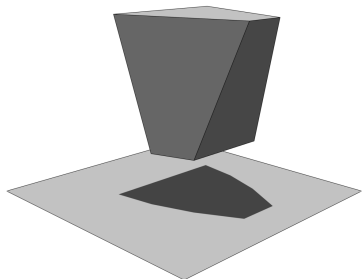
Les m variables basiques forment un **simplexe** en m dimensions.

Quand on **pivote** en changeant les valeurs de certaines variables, on trouve un nouveau simplexe en m dimensions.

Un mot sur la complexité de l'algo

En pratique, l'algo du simplexe est rapide : la plupart du temps, $\approx 3m$ étapes de pivot (pour m contraintes) suffisent.

MAIS il existe des **cas pathologiques** (construits par Klee et Minty en 1973) avec environ 2^m étapes de pivot : on doit visiter **tous les sommets** du polytope...



Le cube de Klee et Minty



Victor L. Klee Jr.
(1925-2007)



George J. Minty Jr.
(1929-1986)

Il existe maintenant des algorithmes **plus rapides**, mais **plus compliqués** :

→ exemple : la méthode des points intérieurs