La programmation linéaire

1. Exemple de programme linéaire
2. Forme générale d'un PL, propriétés
3. Algorithme du simplexe : méthode algébrique, méthode des tableaux
4. Dégénérescences
5. Démarrage de l'algorithme du simplexe : problème de la base initiale
6. Dualité

La programmation linéaire : exemple à 2 variables

Soit une entreprise qui utilise 3 composants :
- A : 180 unités en stock,
- B : 120 unités en stock,
- C : 150 unités en stock.

pour fabriquer 2 produits :
- P_1 : constitué de 2 A, 2 B et 1 C ; gain = 3.
- P_2 : constitué de 3 A, 1 B et 3 C ; gain = 4.

Question : combien faut-il fabriquer de P_1 et de P_2 pour avoir le revenu maximal ?

Réponse intuitive :
- maximiser $P_2 = min(180/3, 120/1, 150/3) = 50 => $ gain = 200,
- stock restant : 30 A, 70 B, 0 C,

Exemple à 2 variables : formulation algébrique

Appelons x_1 et x_2 les quantités respectives des produits P_1, P_2.

Les contraintes sur les composants donnent :

$$2x_1 + 3x_2 \leq 180$$

$$2x_1 + x_2 \leq 120$$
$$x_1 + 3x_2 \leq 150$$

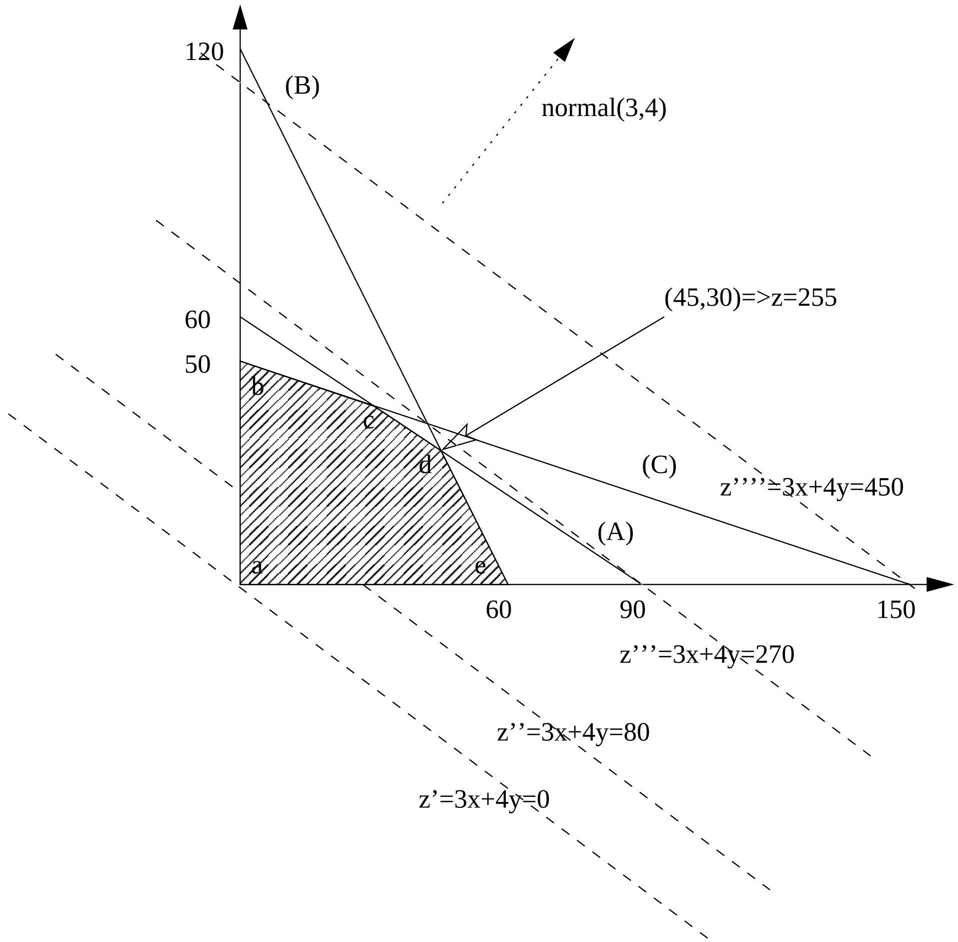
Le critère à maximiser est le revenu :

$$maxz = 3x_1 + 4x_2$$

Il faut rajouter les contraintes de positivité des variables

$$x_1 \geq 0; x_2 \geq 0$$

Exemple à 2 variables : représentation graphique



Forme générale d'un PL

Définition 1. *Un programme linéaire consiste à maximiser (ou minimiser) une fonction objectif linéaire sous contraintes linéaires, il s'agit donc d'un programme mathématique de la forme :*

$$\begin{array}{lrcrl}
\textit{Max} & c_1 \times x_1 & + \ldots + & c_n \times x_n & \\
 & a_{11} \times x_1 & + \ldots + & a_{1n} \times x_n & \leq b_1 \\
 & \ldots & & & \\
 & a_{m1} \times x_1 & + \ldots + & a_{mn} \times x_n & \leq b_m \\
 & x_1 \geq 0 & ; \ldots ; & x_n \geq 0 &
\end{array}$$

Définition 2. *Un programme linéaire est dit sous* **forme standard** *si toutes les contraintes sont des égalités.*

Forme matricielle d'un PL : max $c.x$ avec $A.x \leq b$ et $x \geq 0$.

PL : Définitions

Définition 3. *Un sous-ensemble* $S \subset IR^n$ *est* **convexe** *sissi* $\forall x, y \in S, \forall \lambda \in [0,1], \lambda.x + (1-\lambda).y \in S$

Définition 4. *Un ensemble convexe de la forme* $S = \{x/A.x = b, x \geq 0\}$ *est appelé un* **polytope convexe**. *Un polytope borné sera appelé un* **polyèdre convexe**.

Conséquence : l'ensemble des solutions d'un PL est un polytope ou polyèdre convexe.

Définition 5. *On appelle* **point extrême** *d'un polytope convexe* S*, tout point* x *de* S *qui ne peut être exprimé comme combinaison convexe d'autres points* y *de* S *(*$y \neq x$*).*

Conséquence : tout point d'un polyèdre convexe S est combinaison convexe des points extrêmes de S.

PL : Théorème de l'optimalité

Théorème 1. *Si l'ensemble des contraintes d'un PL forme un polyèdre non vide, il existe une solution optimale qui est un sommet de ce polyèdre.*

Preuve. polyèdre ens. des solutions non vide $=>$ au moins une solution qu'on suppose unique. Récurrence sur la dimension d du polyèdre. Si $d = 0$, le polyèdre réduit à un point, qui est un sommet. Si $d > 0$, soit z la solution optimale située à l'intérieur du polyèdre. Soit P_z, l'hyperplan passant par z dont la direction est fixée par les coefficients de la fonction économique. P_z partage le polyèdre en 2 parties non vides. Il existe alors une solution réalisable dans l'une de ces deux parties de valeur supérieure strictement à la valeur de z. Solution optimale située sur une face du polyèdre. La face d'un polyèdre de dimension $d =$ polyèdre de dimension $d' < d$, même raisonnement, cqfd.

Enumération des sommets d'un polyèdre : n variables et m contraintes $=> C_{n+m}^{n} = (m+n)!/(m!n!)$ intersections possibles. (3'268'760 pour $n = 15$ et $m = 10$).

Algorithme du simplexe : Fondements

L'algorithme le plus connu pour la résolution des programmes linéaires : **algorithme du simplexe**. Calcule la valeur de z pour une suite de sommets telle que : $z_n \geq z_{n-1}$. On est sûr d'aboutir à l'optimum puisque le nombre de sommets est fini.

Les sommets qui constituent la suite sont adjacents 2 à 2 (le k^{ieme} et le $(k+1)^{ieme}$ sont les extrémités d'une même arête. Pour chaque sommet, il existe un sommet adjacent de valeur supérieure ou égale (polyèdre).

L'algorithme du simplexe a un fondement purement géométrique. A partir d'un point de départ (sommet du polyèdre), on passe d'un sommet M à un sommet voisin M' qui a une valeur meilleure qu'en M. Quand on atteint le sommet O dont aucun voisin est meilleur, on a atteint l'optimum.

Algorithme du simplexe : méthode algébrique

Exemple : soit le PL (de la forme $A.x \leq b$) suivant :

$$\begin{array}{ll} \text{Max } z = & 3x_1 + 4x_2 \\ & 2x_1 + 3x_2 \leq 180 \\ & 2x_1 + x_2 \leq 120 \\ & x_1 + 3x_2 \leq 150 \\ & x_i \geq 0 \end{array}$$

Introduction des **variables d'écart**.

$$\begin{array}{llllll} \text{Max } z = & 3x_1 + 4x_2 & +0x_3 & +0x_4 & +0x_5 & \\ & 2x_1 + 3x_2 & +x_3 & & & = 180 \\ & 2x_1 + x_2 & & +x_4 & & = 120 \\ & x_1 + 3x_2 & & & +x_5 & = 150 \\ & x_i \geq 0 & & & & \end{array}$$

Solution de départ : $x_1 = x_2 = 0$ (**hors base**) et $x_3 = 180$, $x_4 = 120$, $x_5 = 150$ (**base**)

Algorithme du simplexe : méthode algébrique

Etape 1 : choix de x_2 pour augmenter l'objectif.

$$
\begin{array}{llrrr}
\text{Max} & z = 200 & +5/3x_1 & -4/3x_5 \\
 & x_2 = 50 & -1/3x_1 & -1/3x_5 \\
 & x_3 = 30 & -x_1 & +x_5 \\
 & x_4 = 70 & -5/3x_1 & +1/3x_5 \\
 & x_i \geq 0 & &
\end{array}
$$

Etape 2 : choix de x_1 pour augmenter l'objectif.

$$
\begin{array}{llrrr}
\text{Max} & z = 250 & -5/3x_3 & +1/3x_5 \\
 & x_1 = 30 & -x_3 & +x_5 \\
 & x_2 = 40 & +1/3x_3 & -2/3x_5 \\
 & x_4 = 20 & +5/3x_3 & -4/3x_5 \\
 & x_i \geq 0 & &
\end{array}
$$

Algorithme du simplexe : méthode algébrique

Etape 3 : choix de x_5 pour augmenter l'objectif.

$$\begin{array}{lrll}
\text{Max} & z = 255 & -5/4x_3 & -1/4x_4 \\
 & x_1 = 45 & +1/4x_3 & -3/4x_4 \\
 & x_2 = 30 & -1/2x_3 & +1/2x_4 \\
 & x_5 = 15 & +5/4x_3 & -3/4x_4 \\
 & x_i \geq 0 & &
\end{array}$$

Solution finale : $x_1 = 45$, $x_2 = 30$, $z = 255$.

Pour un PL qui contient n variables et m contraintes linéairement indépendantes ($n > m$) :

- la base contient m variables non nulles,
- il reste $n - m$ variables hors base nulles.

Algorithme du simplexe

Il faut à chaque étape exprimer l'objectif et les variables de base en fonction des variables hors base.

1. A partir d'une solution initiale admissible,
2. Variable entrante (premier critère de Dantzig) : x_e telle que $c_e = max_i(c_i)$ pour les $c_i \geq 0$ des variables hors base. Si tous les $c_i < 0$, l'optimum est atteint.
3. Variable sortante (deuxième critère de Dantzig) : x_s telle que $b_s/a_{se} = min_k(b_k/a_{ke})$ pour $b_k/a_{ke} \geq 0$. Si ce n'est pas possible, alors solution optimale infinie.
4. Diviser la contrainte s par a_{se} et éliminer x_e de toutes les autres contraintes et de l'objectif.
5. Retourner en 2.

Algorithme du simplexe : transformations d'un PL

L'algorithme du simplexe s'applique sur un PL sous forme standard ($max\ c.x$ avec $A.x = b$ et $x \geq 0$).

Transformations d'un PL :

- minimisation z : maximisation $z' = -z$,
- contrainte $X \leq b$: variable d'écart $y \geq 0$ avec $X + y = b$,
- contrainte $X \geq b$: variable d'écart $y \geq 0$ avec $X - y = b$,
- contrainte $X = b$: 2 contraintes $X \leq b$ et $X \geq b$,
- variable x négative : on la remplace par la variable $x' = -x$ positive,
- variable x sans contrainte de signe : on la remplace par les variables positives x' et x'' avec $x = x' - x''$.

Le cas le plus simple étant un PL sous la forme $A.x \leq b$ car il donne avec les variables d'écart x_e : $A.x + I.x_e = b$.

Algorithme du simplexe : méthode des tableaux

Soit le PL :

$$\begin{cases} max\ z = x_1 + 2x_2 & & \\ -3x_1 + 2x_2 & \leq & 2 \\ -x_1 + 2x_2 & \leq & 4 \\ x_1 + x_2 & \leq & 5 \\ x_i \geq 0 & & \end{cases}$$

On regroupe dans un tableau toutes les informations relatives au PL.

x_1	x_2	x_3	x_4	x_5	$-z$
-3	2	1	0	0	2
-1	2	0	1	0	4
1	1	0	0	1	5
1	2	0	0	0	0

x_1	x_2	x_3	x_4	x_5	$-z$
0	1	0	$1/3$	$1/3$	3
1	0	0	$-1/3$	$2/3$	2
0	0	1	$-5/3$	$4/3$	2
0	0	0	$-1/3$	$-4/3$	-8

Variables entrantes dans la base : x_2, x_1, x_3.

Dégénérescence de première espèce

Pour tout PL sous forme standard à n variables et m contraintes, une solution de base doit avoir m composantes non nulles (base) et $n-m$ composantes nulles (variables hors base).

On parle de **dégénérescence de première espèce** quand l'optimum est atteint en plusieurs points.

Exemple :

$$\begin{cases} max\ z = 6x_1 + 4x_2 & & \\ -3x_1 + 2x_2 & \leq & 4 \\ 3x_1 + 2x_2 & \leq & 16 \\ x_1 & \leq & 3 \\ x_i \geq 0 & & \end{cases}$$

En faisant entrer x_1 et x_2 dans la base, on obtient $z = 32 - 2x_4 + \mathbf{0}x_5$. En faisant entrer x_5 dans la base, on change de base sans améliorer l'objectif : $z = 32 + \mathbf{0}x_3 - 2x_4$.

Dégénérescence de deuxième espèce

La **dégénérescence de deuxième espèce** s'obtient lorsqu'une variable de base s'annule.

Si on ajoute la contrainte $x_1 + 4x_2 \leq 22$ au PL précédent et que l'on modifie la fonction économique : $z = x_1 + x_2$, le tableau simplexe après introduction de x_2 donne :

$\mathbf{x_1}$	x_2	x_3	x_4	x_5	x_6	$-z$
-3/2	1	1/2	0	0	0	2
6	0	-1	1	0	0	12
1	0	0	0	1	0	3
7	0	-2	0	0	1	14
5/2	0	-1/2	0	0	0	-2

Si on introduit x_1 à la place de x_6, on obtient $x_4 = 0$. Ce qui correspond géométriquement au fait que 3 droites passent par un même sommet.

Programme linéaire : Aspect matriciel

PL sous forme standard : max $c.x$ avec $A.x = b$ et $x \geq 0$.

On suppose nbr lignes de A (m) $<$ nbr colonnes (n) $=>$ système sous-déterminé admettant en général une infinité de solutions.

On suppose que l'on peut extraire de A, m colonnes qui forment une matrice carrée B inversible (det(B) $\neq$ 0). Par définition, une **base** d'un PL est un ensemble de m vecteurs-colonne indépendants extraits de A.

$A = (A_1, A_2, ..., A_n)$, notons $B = (A_{j_1}, A_{j_2}, ..., A_{j_m})$ la matrice associée à la base B.

Les **variables de base** sont : x_{j_1}, ..., x_{j_m}. On note x_B le vecteur-colonne des m variables de base : $x_B^t = (x_{j_1}, ..., x_{j_m})$. Les $n - m$ autres variables sont les **variables hors base** et on note x_N leur vecteur-colonne.

Programme linéaire : Aspect matriciel

On peut écrire (en permutant les colonnes de A et les lignes de x) :

$$A.x = b => (B, N).\begin{pmatrix} x_B \\ x_N \end{pmatrix} = b => B.x_B + N.x_N = b$$

Si on pose $x_N = 0$, il reste $B.x_B = b$ système de m équations à m inconnues de déterminant non nul (système de Cramer) admettant une solution unique : $x_B = B^{-1}.b$. Si $x_B \geq 0$, alors x_B est une **solution de base admissible**.

Matriciellement, la fonction économique peut s'écrire :

$$z = c.x = (c_B, c_N).\begin{pmatrix} x_B \\ x_N \end{pmatrix} = c_B.x_B + c_N.x_N = z_B + \Delta_N.x_N$$

avec : $z_B = c_B.B^{-1}.b$ et $\Delta_N = c_N - c_B.B^{-1}.N$

Démarrage de l'algorithme du simplexe : problème de la base initiale

Cas favorable : système $A.x \leq b$ avec $b \geq 0$ (p inconnues et m contraintes).

L'introduction des variables d'écart donne : $A.x + I.x_e = b$

La base de départ est : $B_0 = I$. Les variables de base s'expriment simplement en fonction des variables hors base : $x_{e_i} = b_i - \Sigma_{j=1}^{p} a_{ij}.x_j$.

La solution de base de départ est admissible : $x_e = b$ et $x = 0$. La fonction de coût est exprimée en fonction des variables hors base.

Solution connue à l'avance : la forme matricielle donne $x_B = B^{-1}.b - B^{-1}.N.x_N$. Si on connait B^{-1}, on peut initialiser le simplexe (la connaissance de $B^{-1}.b$ et $B^{-1}.N$ suffisent).

Initialisation du simplexe : variables artificielles

Un PL peut avoir des contraintes de type (P) ($\leq$), de type (Q) ($\geq$) ou de type (R) ($=$). Il peut être nécessaire d'introduire des variables artificielles pour constituer la matrice initiale I.

Ex : 2 variables d'écart x_4 et x_5 et 2 variables artificielles $x_{a'}$ et $x_{a''}$.

$$\begin{cases} max\ z = 4x_1 + 5x_2 + 3x_3 \\ x_1 + 2x_2 + x_3 \leq 5(P) \\ 2x_1 + x_2 + x_3 \geq 1(Q) \\ x_1 + x_2 = 4(R) \\ x_i \geq 0 \end{cases} \qquad \begin{cases} max\ z = 4x_1 + 5x_2 + 3x_3 \\ x_1 + 2x_2 + x_3 + x_4 = 5(P) \\ 2x_1 + x_2 + x_3 - x_5 + x_{a'} = 1(Q) \\ x_1 + x_2 + x_{a''} = 4(R) \\ x_i \geq 0 \end{cases}$$

La résolution d'un tel système peut se faire suivant 2 méthodes :
- méthode des 2 phases,
- méthode du grand “M”.

Méthode des 2 phases

Phase 1 : On résoud le PL avec variables artificielles en remplaçant z par $z' =$ somme des variables artificielles (minimisation). z' doit être exprimé en ft des variables hors base. Après résolution de ce PL, on obtient :

x_1	x_2	x_3	x_4	x_5	$x_{a'}$	$x_{a''}$	$-z$
0	1	1	1	0	0	-1	1
1	1	0	0	0	0	1	4
0	1	-1	0	1	-1	2	7
0	0	0	0	0	-1	-1	0

Phase 2 : A partir de la base admissible trouvée à la fin de la phase 1, on applique le simplexe en supprimant les variables artificielles, ce qui donne les tableaux simplexes :

x_1	x_2	x_3	x_4	x_5	$-z$
0	1	1	1	0	**1**
1	1	0	0	0	4
0	1	-1	0	1	7
0	1	**3**	0	0	-16

x_1	x_2	x_3	x_4	x_5	$-z$
0	1	1	1	0	1
1	1	0	0	0	4
0	2	0	1	1	8
0	-2	0	-3	0	-19

Méthode du grand “M”

Les 2 phases de la précédente méthode sont combinées en une seule en introduisant dans la ft économique les variables artificielles pondérées d'un coefficient $-M$ très négatif. La nouvelle ft économique doit être exprimée en ft des variables hors base.

x_1	x_2	x_3	x_4	x_5	$x_{a'}$	$x_{a''}$	$-z$
1	2	1	1	0	0	0	5
2	1	1	0	-1	1	0	1
1	1	0	0	0	0	1	4
3M+4	2M+5	M+3	0	-M	0	0	5M

Ce qui donne :

x_1	x_2	x_3	x_4	x_5	$x_{a'}$	$x_{a''}$	$-z$
0	1	1	1	0	0	-1	1
1	1	0	0	0	0	1	4
0	1	-1	0	1	-1	2	7
0	1	3	0	0	-M	-M	-16

Dualité

A chaque PL, on peut associer un autre PL, appelé **programme dual**. Le premier PL est alors appelé **programme primal**.

Forme standard de passage au dual

$$\begin{cases} A.x \leq b \\ x \geq 0 \\ max\ z = c.x \end{cases}$$

Programme dual

$$\begin{cases} y.A \geq c \\ y \geq 0 \\ min\ z' = y.b \end{cases}$$

Exemple :

Soit le PL primal :

$$\begin{cases} Max\ z = 3x_1 + 4x_2 \\ 2x_1 + 3x_2 \leq 180 \\ 2x_1 + x_2 \leq 120 \\ x_1 + 3x_2 \leq 150 \\ x_i \geq 0 \end{cases}$$

Son dual est :

$$\begin{cases} Min\ z' = 180y_1 + 120y_2 + 150y_3 \\ 2y_1 + 2y_2 + y_3 \geq 3 \\ 3y_1 + y_2 + 3y_3 \geq 4 \\ y_i \geq 0 \end{cases}$$

Dualité : propriétés

– Le dual du dual est le primal.
– Si x est une solution du primal et y une solution du dual, on a : $c.x \leq y.b$
– Si $c.x = y.b$, alors x est une solution optimale du primal et y une solution optimale du dual.

Un seul des 3 cas suivants se trouve réalisé :
– Le primal et le dual ont des solutions optimales finies x^* et y^*, alors on a, à l'optimum : $c.x^* = y^*.b$,
– L'un des 2 PLs n'a pas de solution admissible, l'autre a une solution optimale infinie,
– Ni le primal, ni le dual n'ont de solutions admissibles.

Dualité : Correspondance des optimaux

Tableau du Primal à l'optimal

x_1	x_2	$x_{\overline{3}}$	$x_{\overline{4}}$	$x_{\overline{5}}$	$-z$
1	0	-1/4	3/4	0	45
0	1	1/2	-1/2	0	30
0	0	-5/4	3/4	1	15
0	0	-5/4	-1/4	0	-255

Tableau du Dual à l'optimal

y_1	y_2	y_3	$y_{\overline{1}}$	$y_{\overline{2}}$	$-z''$
0	1	-3/4	-3/4	1/2	1/4
1	0	5/4	1/4	-1/2	5/4
0	0	-15	-45	-30	255

On constate que :

- Sachant que $z' = -z''$, la valeur de $z' = 255$ de l'optimum du dual égale celle du primal ;
- les valeurs des variables de base du dual y_1 et y_2 sont, au signe près, les coefficients de $x_{\overline{3}}$ et $x_{\overline{4}}$ dans la ft objectif z au primal ;
- les valeurs des coefficients des variables hors base dans la ft objectif z' du dual sont égales, au signe près, aux valeurs des variables de base du primal.

Dualité : Relations d'exclusion

Soient 2 PLs en dualité :

Le primal (P) : $A.x \leq b$; $x \geq 0$; max $z = c.x$

Le dual (D) : $y.A \geq c$; $y \geq 0$; min $z' = y.b$

Alors x^* et y^* sont optimales sissi :

$$\left\{ \begin{array}{cccl} y_i^* & . & x_{\bar{i}}^* & = 0 \quad (i = 1, 2, ..., m) \\ y_{\bar{j}}^* & . & x_j^* & = 0 \quad (j = 1, 2, ..., n) \end{array} \right.$$

Les relations d'exclusion permettent :

- de trouver l'optimum du dual si l'on connaît l'optimum du primal et réciproquement (si on ne connaît pas le tableau optimal).
- de tester si une solution du primal ou du dual est optimale ou non. La solution du primal doit donnée grâce aux relations d'exclusion une solution admissible du dual.