OPTIMISATION ET CONTRÔLE

Grégoire Allaire
Département de mathématiques appliquées
Ecole Polytechnique

Programmation linéaire

Programmation linéaire

1. Etude théorique
2. Algorithme du simplexe.
3. Algorithme des points intérieurs.
4. Dualité.
5. Initiation à la programmation linéaire en nombre entiers.

On appelle **programme linéaire** un problème d'optimisation pour lequel la fonction objectif et les contraintes sont affines

$$(LP) \qquad \inf_{\tilde{x}\in\mathbb{R}^{\tilde{n}} \text{ tel que } \tilde{A}\tilde{x}\geq\tilde{b},\ \tilde{A}'\tilde{x}=\tilde{b}'} \tilde{c}\cdot\tilde{x}$$

avec des matrices $\tilde{A}, \tilde{A}'$ et des vecteurs $\tilde{b}, \tilde{b}', \tilde{c}$.

Lemme. Tout programme linéaire (LP) peut se mettre sous la forme standard

$$(SLP) \qquad \inf_{x\in\mathbb{R}^{n} \text{ tel que } Ax=b,\ x\geq 0} c\cdot x,$$

où A est une matrice $m \times n$ (avec $\mathrm{rg}(A) = m$), $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$.
Preuve. On introduit des variables d'écart (slack variables) $y \geq 0$

$$\tilde{A}\tilde{x} \geq \tilde{b} \Leftrightarrow \left(\tilde{A}\tilde{x} = \tilde{b} + y \text{ et } y \geq 0\right)$$

On décompose $\tilde{x} = \tilde{x}^+ - \tilde{x}^-$, $\tilde{x}^+ = \max(\tilde{x}, 0)$, $\tilde{x}^- = \max(-\tilde{x}, 0)$.
On élimine les contraintes redondantes pour avoir $\mathrm{rg}(A) = m$.

Définitions (et vocabulaire)

Désormais on se limite aux programmes linéaires **standards**.

1) L'ensemble X_{ad} des vecteurs de $\mathbb{R}^n$ qui satisfont les contraintes

$$X_{ad} = \{x \in \mathbb{R}^n \text{ tel que } Ax = b, \; x \geq 0\} ,$$

est l'ensemble des **solutions admissibles** (c'est un polyèdre convexe).
(Polyèdre = intersection finie de demi-espaces.)

2) Un point de minimum de

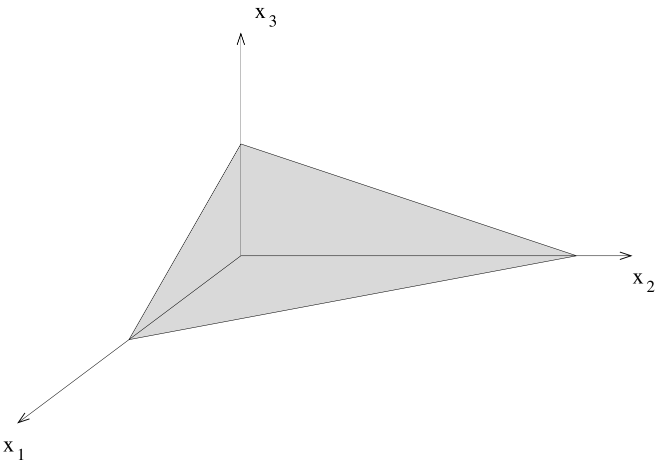
$$\inf_{x \in X_{ad}} \; c \cdot x$$

est appelé **solution optimale**.

3) On appelle **sommet** ou point extrémal de X_{ad} tout point $\overline{x} \in X_{ad}$ qui ne peut pas se décomposer en une combinaison convexe de deux autres points de X_{ad}, c'est-à-dire que, s'il existe $y, z \in X_{ad}$ et $\theta \in]0, 1[$ tels que $\overline{x} = \theta y + (1 - \theta) z$, alors $y = z = \overline{x}$.
Ce sont les sommets (géométriques) du polyèdre.

Exemple simple: polyèdre

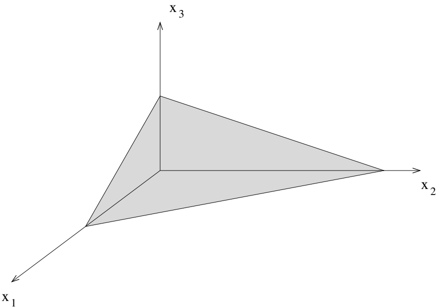
$$\min_{\substack{x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \\ 2x_1 + x_2 + 3x_3 = 6}} x_1 + 4x_2 + 2x_3$$



X_{ad} est un polyèdre, ici un triangle plat (en gris) qui a 3 sommets.

Intuition géométrique: importance des sommets

$$\min_{\substack{x_1\geq 0, x_2\geq 0, x_3\geq 0\\ 2x_1+x_2+3x_3=6}} \quad x_1 + 4x_2 + 2x_3$$



Pour minimiser on balaie l'espace $\mathbb{R}^3$ avec des plans parallèles $x_1 + 4x_2 + 2x_3 = v$ en augmentant v à partir de $-\infty$: on finit par toucher X_{ad} et atteindre le minimum. Autrement dit, un des sommets du triangle T est un point de minimum.

Existence d'une solution et optimalité des sommets

Lemme. Il existe au moins une solution optimale du programme linéaire standard (SLP) si et seulement si la valeur du minimum est finie

$$-\infty < \inf_{x\in\mathbb{R}^n \text{ tel que } Ax=b,\ x\geq 0} c\cdot x < +\infty.$$

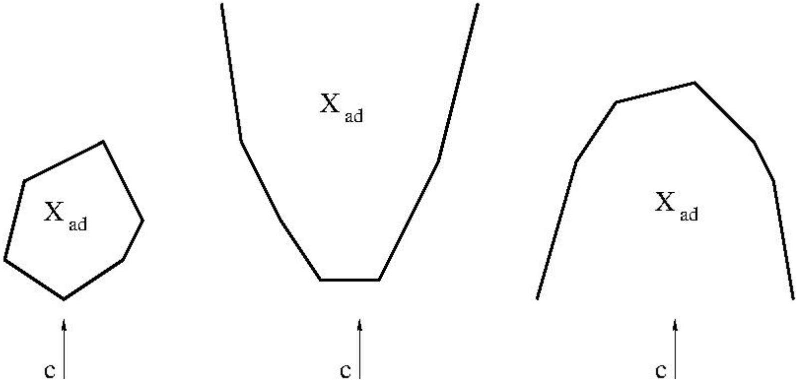
Lemme. S'il existe une solution optimale du programme linéaire, alors il existe une solution optimale qui est un sommet du polyèdre.

Preuve (géométrique). Faire un dessin: X_{ad} est l'intersection du quadrant $\mathbb{R}^n_+$ par un sous-espace affine (voir transparent suivant).

Preuve (analytique). Voir un peu plus loin.

Preuve géométrique

Dans le sous-espace affine $Ax = b$, X_{ad} ressemble à un des 3 cas:

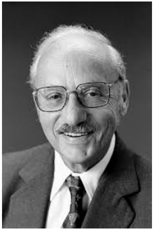


L'ensemble X_{ad} n'est pas forcément borné. Il existe une solution optimale (pas toujours unique mais l'une d'entre elles est un sommet) dans les deux cas à gauche mais à droite inf $= -\infty$.

Principe de résolution

Si un programme linéaire a une valeur minimum finie, alors il existe une solution optimale qui est un sommet de X_{ad}.

- X_{ad} a un nombre **fini** de sommets.
- Il suffit de tester la valeur de $c \cdot x$ sur ce nombre fini de sommets !
- Malheureusement le nombre de sommets est **exponentiel** en n, m. (Cf. le cas du cube en dimension d qui a 2^d sommets.)
- L'algorithme du simplexe de Dantzig énumère intelligemment l'ensemble des sommets en faisant décroitre la fonction $c \cdot x$.

George Dantzig  (1914-2005)

Existence d'une solution optimale

Lemme. Il existe au moins une solution optimale du programme linéaire standard si et seulement si la valeur du minimum est finie

$$-\infty < \inf_{x \in \mathbb{R}^n \text{ tel que } Ax=b,\ x \geq 0} c \cdot x < +\infty.$$

Preuve (analytique). Si $X_{ad} = \{x \in \mathbb{R}^n \text{ t.q. } Ax = b,\ x \geq 0\}$ est borné, alors le minimum est atteint puisque X_{ad} est fermé. Si X_{ad} n'est pas borné, on utilise le Lemme de Farkas pour montrer que le cône suivant est fermé

$$C = \left\{ \sum_{i=1}^{n} x_i \mathcal{A}_i \text{ avec } x_i \geq 0 \right\} \quad \text{avec} \quad \mathcal{A} = \begin{pmatrix} c^* \\ A \end{pmatrix}$$

et $\mathcal{A}_i$ les colonnes de la matrice $\mathcal{A}$. Soit $(x^k)_{k \geq 1} \in X_{ad}$ une suite minimisante. Si le minimum est fini, la suite $\mathcal{A}x^k = (c \cdot x^k, b)^*$ est bornée dans C, qui est fermé, et converge donc vers une limite $\mathcal{A}x^* \in C$, ce qui prouve que $x^* \in X_{ad}$ est un point de minimum. (On n'a pas prouvé que x^k converge vers x^*...)

Rappel sur le Lemme de Farkas

Dans la démonstration du Lemme 5.2.19 de Farkas, il est démontré par récurrence sur le nombre n de vecteurs $\mathcal{A}_i$ que le cône

$$C = \left\{ \sum_{i=1}^{n} x_i \mathcal{A}_i \text{ avec } x_i \geq 0 \right\}$$

est fermé (sans aucune hypothèse sur les vecteurs $\mathcal{A}_i$). On applique ce résultat aux colonnes $\mathcal{A}_i$ de la matrice $\mathcal{A} = \begin{pmatrix} c^* \\ A \end{pmatrix}$ de taille $(m+1) \times n$.

Remarque. On vient de démontrer qu'il existe une solution optimale $x^* \in X_{ad}$. Il reste à donner une démonstration analytique de l'existence d'une solution optimale qui soit un sommet du polyèdre X_{ad}. Pour cela on introduit quelques notations.

Solution basique

Définition.

• On appelle base de X_{ad} une base B de $\mathbb{R}^m$ formée de m colonnes de A. Après permutation de ses colonnes on écrit

$$A = (B, N), \quad x = (x_B, x_N), \quad Ax = Bx_B + Nx_N.$$

avec $x_B \in \mathbb{R}^m$ variables de base, $x_N \in \mathbb{R}^{n-m}$ variables hors base.

• Dimensions: A est une matrice $m \times n$, B une matrice $m \times m$ et N une matrice $m \times (n - m)$.

• Une solution basique (ou de base) est $x \in X_{ad}$ tel que $x_N = 0$. Autrement dit, $(x_B, 0)$ est une solution basique si $Bx_B = b$ et $x_B \geq 0$.

Lemme. Les sommets du polyèdre X_{ad} sont exactement les solutions basiques.

Solutions basiques = sommets

Preuve. Si $x \in X_{ad}$ est une solution basique, alors $x = (x_B, x_N)$ avec $Bx_B = b$ et $x_N = 0$. Supposons qu'il existe $0 < \theta < 1$ et $y, z \in X_{ad}$ tels que $x = \theta y + (1-\theta)z$. Alors

$$0 = \theta y_N + (1-\theta)z_N \text{ et } y \geq 0, z \geq 0 \ \Rightarrow y_N = z_N = 0.$$

On en déduit $By_B = Bz_B = b$, ce qui implique $x_B = y_B = z_B$. Donc x est un sommet de X_{ad}.
Réciproquement, si x est un sommet de X_{ad}, comme $x \geq 0$, après réarrangement on a $x = (x_1, ..., x_k, 0, ..., 0)$ avec $x_1, ..., x_k > 0$ et $b = \sum_{i=1}^k x_i a_i$ où les (a_i) sont les colonnes de A. Alors x est une solution basique si $(a_1, ..., a_k)$ est libre dans $\mathbb{R}^m$ (on complète pour avoir une base B). Si cette famille est liée, il existe $y \neq 0$ tel que $\sum_{i=1}^k y_i a_i = 0$ et on fixe $y_{k+1} = ... = y_n = 0$. Comme $x_1, ..., x_k > 0$, il existe $\epsilon > 0$ tel que $(x + \epsilon y) \in X_{ad}$ et $(x - \epsilon y) \in X_{ad}$. Le fait que $x = (x + \epsilon y)/2 + (x - \epsilon y)/2$ contredit le caractère extrémal de x, donc x est une solution basique.

Existence d'un sommet optimal

Proposition. S'il existe une solution optimale, alors au moins une solution optimale est un sommet du polyèdre.

Preuve (analytique). Soit $x \in X_{ad}$ une solution optimale. Comme $x \geq 0$, après réarrangement on a $x = (x_1, ..., x_k, 0, ..., 0)$ avec $x_1, ..., x_k > 0$ et

$$b = \sum_{i=1}^{k} x_i a_i \quad \text{avec } (a_i) \text{ les colonnes de } A.$$

Si la famille $(a_1, ..., a_k)$ est libre dans $\mathbb{R}^m$, alors x est une solution optimale basique. Si $(a_1, ..., a_k)$ est lié, alors il existe $y \neq 0$ tel que

$$\sum_{i=1}^{k} y_i a_i = 0 \quad \text{et} \quad (y_{k+1}, ..., y_n) = 0.$$

Comme $x_1, ..., x_k > 0$, il existe $\epsilon > 0$ tel que $(x \pm \epsilon y) \in X_{ad}$.
Comme x est un point de minimum, on a nécessairement

$$c \cdot x \leq c \cdot (x \pm \epsilon y),$$

c'est-à-dire $c \cdot y = 0$.

Fin de la preuve de la proposition

On définit une famille $z_\epsilon = x + \epsilon y$, paramétrée par $\epsilon \in \mathbb{R}$, qui vérifie

$$Az_\epsilon = b \quad \text{et} \quad c \cdot z_\epsilon = c \cdot x.$$

En partant de $\epsilon = 0$, si on augmente ou on diminue ϵ, z_ϵ reste dans l'ensemble X_{ad} jusqu'à une valeur ϵ_0 au delà de laquelle la contrainte $z_\epsilon \geq 0$ est violée.

Autrement dit, $z_{\epsilon_0} \in X_{ad}$ possède au plus $(k-1)$ composantes non nulles et est solution optimale.

On répète l'argument précédent avec $x = z_{\epsilon_0}$ et une famille de $(k-1)$ colonnes (a_i). A force de diminuer la taille de cette famille, on obtient finalement une famille libre et une solution optimale basique.

Condition d'optimalité d'une solution basique

Proposition. Soit une solution basique $\overline{x}$ pour une base B, supposée non dégénérée, c.-à-d. $\overline{x}_B = B^{-1}b > 0$. Alors $\overline{x}$ est optimal **si et seulement si** le vecteur des coûts réduits est positif

$$\tilde{c}_N = c_N - N^*(B^{-1})^* c_B \geq 0.$$

Remarque. Si la solution basique $\overline{x}$ est dégénérée, $\overline{x}_B = B^{-1}b \geq 0$, alors la condition $\tilde{c}_N \geq 0$ est seulement suffisante pour que $\overline{x}$ soit optimal.

Preuve. On décompose $x = (x_B, x_N)$ et $c = (c_B, c_N)$. Pour tout $x \in X_{ad}$, $x_B = B^{-1}(b - Nx_N)$ et, comme $\overline{x} = (B^{-1}b, 0)$, on a

$$c \cdot x - c \cdot \overline{x} = c_B \cdot B^{-1}(b - Nx_N) + c_N \cdot x_N - c_B \cdot B^{-1}b = \tilde{c}_N \cdot x_N.$$

Si $\tilde{c}_N \geq 0$, alors $\overline{x}$ est optimal car

$$c \cdot x - c \cdot \overline{x} = \tilde{c}_N \cdot x_N \geq 0 \quad \text{puisque } x_N \geq 0.$$

La condition est donc toujours suffisante.

Condition d'optimalité d'une solution basique (fin)

Réciproquement, on suppose que $\exists i$ tel que $(\tilde{c}_N \cdot e_i) < 0$.
Pour $\epsilon > 0$ on définit $x(\epsilon) = (x_B(\epsilon), x_N(\epsilon))$ par

$$x_N(\epsilon) = \epsilon e_i \quad \text{et} \quad x_B(\epsilon) = B^{-1}(b - N x_N(\epsilon)).$$

A cause de l'hypothèse de non-dégénérescence, $B^{-1}b > 0$, on a bien $x(\epsilon) \geq 0$ et $x(\epsilon) \in X_{ad}$ pour ϵ petit.
Comme $x(0) = \overline{x}$, on en déduit

$$c \cdot x(\epsilon) = c \cdot x(0) + \epsilon(\tilde{c}_N \cdot e_i) < c \cdot \overline{x},$$

ce qui montre que $\overline{x}$ n'est pas optimal. La condition est donc nécessaire.

On utilise la condition d'optimalité et le coût réduit.

- On part d'un sommet x^0, ou solution basique initiale, associée à une base B^0.
- A l'itération $k \geq 0$, on passe d'un sommet x^k à un nouveau sommet x^{k+1} en étudiant le coût réduit $\tilde{c}_N^k$:
  1. Si $\tilde{c}_N^k \geq 0$, alors x^k est optimal et on a convergé !
  2. S'il existe i tel que $\tilde{c}_N^k \cdot e_i < 0$, alors on bouge et on change une colonne de la base B^k pour obtenir une nouvelle base B^{k+1} et diminuer le coût

$$c \cdot x^{k+1} \leq c \cdot x^k.$$

     La décroissance est stricte si x^k est non dégénérée (cf. Proposition précédente).

Remarque. X_{ad} a un nombre fini (exponentiel) de sommets mais l'algorithme du simplexe n'en visite qu'un petit nombre en diminuant le coût.

Initialisation: x^0 solution basique associée à la base B^0.
Itérations pour $k \geq 0$: pour la base B^k on calcule le coût réduit

$$\tilde{c}_N^k = c_N^k - (N^k)^* (B^{k^{-1}})^* c_B^k.$$

Si $\tilde{c}_N^k \geq 0$, alors x^k est optimal.
Sinon, il existe i tel que $\tilde{c}_N^k \cdot e_i < 0$, et pour $a_i = Ae_i$ on pose

$$x^k(\epsilon) = (x_B^k(\epsilon), x_N^k(\epsilon)) \text{ avec } x_N^k(\epsilon) = \epsilon e_i \ , \ x_B^k(\epsilon) = (B^k)^{-1}(b - \epsilon a_i).$$

Soit $\epsilon^k \geq 0$ tel que $x^k(\epsilon^k) \geq 0$ mais il existe j tel que $x^k(\epsilon^k) \cdot e_j = 0$. On pose

$$x^{k+1} = x^k(\epsilon^k),$$

et la base B^{k+1} est déduite de B^k en remplaçant sa j-ème colonne par la colonne a_i.

Remarques.

- Si toutes les solutions basiques x^k sont non dégénérées, $x_B^k > 0$, alors l'algorithme du simplexe converge en un nombre fini d'itérations.
- En théorie, on peut avoir besoin d'un nombre exponentiel d'itérations pour converger.
- En pratique, l'algorithme du simplexe converge en un nombre polynomial d'itérations.
- Il pourrait arriver (si x^k est dégénérée) que $c \cdot x^{k+1} = c \cdot x^k$ et que l'algorithme ne converge pas (phénomène du cyclage). Mais ça n'arrive jamais en pratique...
- C'est un algorithme de type gradient ! Le vecteur des coûts réduits est un gradient projeté.

Algorithme du simplexe (fin)

Remarques pratiques.

- Le coût de calcul est lié à l'inversion de la base B^k (matrice $m \times m$).
- Il existe d'excellents logiciels utilisant cet algorithme (avec beaucoup de petits "trucs"): ne pas essayer de le reprogrammer...
- Trouver une initialisation basique n'est pas toujours évident. Pour cela, on peut résoudre un autre programme linéaire

$$\min_{\substack{x \geq 0,\ y \geq 0 \\ Ax+y=b}} \sum_{i=1}^{m} y_i \,,$$

où on a préalablement changé le signe des contraintes pour que $b \geq 0$. On démarre de $x^0 = 0$ et $y^0 = b$ et on converge vers $y = 0$ et x basique.

3 - Algorithme des points intérieurs

$$(LP) \qquad \inf_{x\in\mathbb{R}^n \text{ tel que } Ax=b,\ x\geq 0} c\cdot x.$$

On définit une barrière logarithmique pour $x > 0$

$$\pi(x) = -\sum_{i=1}^{n} \log x_i.$$

On remplace (LP) par une version pénalisée avec $\varepsilon > 0$

$$(LP_\varepsilon) \qquad \min_{x\in\mathbb{R}^n \text{ tel que } Ax=b,\ x>0} \varepsilon\pi(x) + c\cdot x.$$

L'algorithme de trajectoire centrale minimise (LP_ε) par une méthode de Newton pour des valeurs $\varepsilon \to 0$.

- En pratique la contrainte $x > 0$ n'est jamais active.
- Motivation théorique: complexité polynomiale (Khachian 1979).
- Algorithme pratique de Karmarkar (1984).

Commençons par quelques rappels de dualité.

Définition. Soit un Lagrangien $\mathcal{L}(v,q)$ défini sur $V \times P$.

Pour $v \in V$ on définit la fonction primale $\mathcal{J}(v) = \sup_{q \in P} \mathcal{L}(v,q)$.

Pour $q \in P$ on définit la fonction duale $\mathcal{G}(q) = \inf_{v \in V} \mathcal{L}(v,q)$.

On appelle problème primal $\inf_{v \in V} \mathcal{J}(v)$.

On appelle problème dual $\sup_{q \in P} \mathcal{G}(q)$.

Théorème (dualité forte). Le couple (u,p) est un point-selle de $\mathcal{L}$ sur $V \times P$ si et seulement si

$$\mathcal{J}(u) = \min_{v \in V} \mathcal{J}(v) = \max_{q \in P} \mathcal{G}(q) = \mathcal{G}(p) \ .$$

Définition du problème dual en programmation linéaire

Soit le programme linéaire standard ou problème primal

$$(P) \qquad \inf_{x\in\mathbb{R}^n \text{ tel que } Ax=b,\ x\geq 0} c\cdot x$$

avec $x, c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ et A matrice $m\times n$. Pour $p \in \mathbb{R}^m$ (multiplicateur de Lagrange), le Lagrangien est

$$L(x,p) = c\cdot x + p\cdot(b - Ax),$$

où on ne "dualise" que les contraintes d'égalité. On introduit la fonction duale

$$G(p) = \min_{x\geq 0} L(x,p) = \begin{cases} p\cdot b & \text{si } A^*p - c \leq 0 \\ -\infty & \text{sinon.} \end{cases}$$

Le problème dual est donc

$$(D) \qquad \sup_{p\in\mathbb{R}^m \text{ tel que } A^*p-c\leq 0} p\cdot b$$

Théorème de dualité en programmation linéaire

$$(P) \qquad \inf_{x \in X_{ad}} c \cdot x \,, \quad X_{ad} = \{x \in \mathbb{R}^n \text{ tel que } Ax = b,\ x \geq 0\}$$

$$(D) \qquad \sup_{p \in P_{ad}} p \cdot b \,, \quad P_{ad} = \{p \in \mathbb{R}^m \text{ tel que } A^* p - c \leq 0\}$$

Théorème. Si (P) ou (D) a une valeur finie, alors il existe un point selle (x^*, p^*) du Lagrangien et x^* est une solution optimale de (P) tandis que p^* est une solution optimale de (D).

Remarque. Le problème primal (P) a n inconnues et m contraintes, tandis que le problème dual (D) a m inconnues et n contraintes.

Conséquence: (P) ou (D) peut être plus facile. Résoudre les deux permet d'avoir une estimation d'erreur

$$p \cdot b \leq p^* \cdot b = c \cdot x^* \leq c \cdot x \quad \forall p \in P_{ad}, \forall x \in X_{ad}$$

Preuve. On suppose que X_{ad} et P_{ad} sont non vides (si ce n'est pas le cas, voir le polycopié). Soit $x \in X_{ad}$ et $p \in P_{ad}$. Comme $x \geq 0$ et $A^* p \leq c$, on a

$$c \cdot x \geq A^* p \cdot x = p \cdot Ax = p \cdot b,$$

puisque $Ax = b$. En particulier, cette inégalité implique que les valeurs optimales des deux problèmes, primal et dual, sont finies, donc qu'ils admettent des solutions optimales, notées x^* et p^*.

On applique alors le théorème de Kuhn et Tucker (les contraintes d'inégalité affines sont toujours qualifiées) et (x^*, p^*) est un point selle du Lagrangien, défini sur $\mathbb{R}^n_+ \times \mathbb{R}^m$ par

$$L(x, p) = c \cdot x + p \cdot (b - Ax).$$

Une ouverture vers la recherche opérationnelle.

Supposons que les données A, b soient des **entiers** (mais pas c !).

$$(LP) \qquad \inf_{x\in\mathbb{R}^n \text{ tel que } Ax=b,\ x\geq 0} c\cdot x$$

Existe-t-il une solution optimale de (LP) telle que $x \in \mathbb{Z}^n$?

On propose un exemple (parmi d'autres) où la réponse est positive et la programmation linéaire est un outil utile:

problème d'affectation (cf. amphi 1).

Cet exemple repose sur la notion de matrices totalement unimodulaires.
Pour en savoir plus, voir le cours MAP557 de S. Gaubert (recherche opérationnelle).

Matrices totalement unimodulaires

Une solution basique pour une base B, avec $A = (B, N)$, est

$$x_B = B^{-1}b, \quad x_N = 0.$$

Pour que x_B soit entier $\forall\, b \in \mathbb{Z}^m$, il faut que B^{-1} le soit aussi.

Définition. On dit qu'une matrice $A \in \mathbb{Z}^{m\times n}$ est totalement unimodulaire si toute sous-matrice carrée extraite de A a un déterminant égal à ± 1 ou 0.

Proposition. Toutes les solutions basiques de (LP) sont entières si A est totalement unimodulaire. En particulier, (LP) admet une solution entière.

Preuve. On utilise les formules de Cramer pour calculer $x_B = B^{-1}b$ et on conclut car $\det B = \pm 1$.

Existe-t-il des matrices totalement unimodulaires ?

Condition de Poincaré pour l'unimodularité

Lemme (Poincaré). Si A est une matrice à coefficients ± 1 ou 0, avec au plus un coefficient 1 par colonne, et au plus un coefficient -1 par colonne, alors A est totalement unimodulaire.

Preuve. L'hypothèse sur A est aussi vraie pour toutes ses sous-matrices. Donc il suffit de considérer le cas où la matrice A est carrée et de vérifier que $\det A \in \{\pm 1, 0\}$.

• S'il existe une colonne nulle de A, alors $\det A = 0$.

• S'il existe une colonne de A avec un seul coefficient non-nul, on développe son déterminant par rapport à cette colonne, et on conclut par récurrence sur la dimension de A.

• Si toutes les colonnes de A ont exactement un coefficient 1 et un coefficient -1 (tous les autres sont 0), alors le vecteur $(1, \cdots, 1)$ appartient au noyau de la transposée de A et donc $\det A = 0$.

Application: problème d'affectation

Soit une matrice $N \times N$ de coefficients $a_{ij} \geq 0$. Si $\mathcal{S}_N$ est l'ensemble des permutations de $\{1, ..., N\}$, on maximise

$$(P) \qquad \max_{\sigma \in \mathcal{S}_N} \sum_{i=1}^{N} a_{i\sigma(i)}.$$

Si on remplace la permutation σ par sa matrice (v_{ij}), (P) se réécrit

$$\max_{(v_{ij}) \in \{0,1\}^{N\times N}} \sum_{i=1}^{N} \sum_{j=1}^{N} a_{ij} v_{ij}$$

sous les contraintes

$$\sum_{j=1}^{N} v_{ij} = 1 \text{ pour } 1 \leq i \leq N, \quad \sum_{i=1}^{M} v_{ij} = 1 \text{ pour } 1 \leq j \leq N.$$

Unimodularité du problème d'affectation

On ré-écrit le problème d'affectation (P) sous la forme

$$\max_{(v_{ij})\in\mathbb{Z}^{N\times N}} \sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij}v_{ij}$$

sous les contraintes $v_{ij}\geq 0$, pour $1\leq i,j\leq N$, et

$$(C)\ \sum_{j=1}^{N} v_{ij} = 1 \text{ pour } 1\leq i\leq N,\ \ -\sum_{i=1}^{M} v_{ij} = -1 \text{ pour } 1\leq j\leq N.$$

On peut écrire (C) comme $Av=b$ avec $v=(v_{ij})$ et A matrice de taille $2N\times N^2$ où chaque colonne a exactement un coefficient $+1$ et un -1, les autres étant 0.

Donc la matrice A est unimodulaire.

Résolution pratique du problème d'affectation

On applique l'algorithme du simplexe au programme linéaire

$$\max_{v=(v_{ij})\geq 0, Av=b} \sum_{i=1}^{N} \sum_{j=1}^{N} a_{ij} v_{ij} .$$

Comme la matrice A est unimodulaire, toutes les solutions basiques sont entières.

1. Une initialisation évidente du simplexe est $v = \mathrm{Id}$.
2. Toutes les solutions successives du simplexe sont entières.
3. Toutes les solutions successives vérifient $v_{ij} \in \{0, 1\}$ à cause des contraintes

$$\sum_{j=1}^{N} v_{ij} = 1, \quad \sum_{i=1}^{M} v_{ij} = 1.$$