OPTIMISATION

LICENCE MATHÉMATIQUE ET GESTION 2013-2014

Table des matières

2 Programmation linéaire

2.1 Optimisation linéaire

Définitions 2.1.

- Une fonction affine (sur $\mathbb{R}^n$) est une fonction de la forme $x \mapsto a \cdot x + b$, avec $a \in \mathbb{R}^n$ et $b \in \mathbb{R}$
- Un problème d'optimisation linéaire est un problème de la forme
  $\min$ ou $\max f(x)$, sous contraintes $g_j(x) \geq 0, h_l(x) \leq 0, k_m(x) = 0, j = 1, \ldots, j_1, l = 1, \ldots, l_1, k = 1, \ldots, k_1$
  Ici, toutes les fonctions qui apparaissent sont affines
- Si $x \in \mathbb{R}^n$, la notation $x \geq 0$ symbolise $x_j \geq 0$, $j = 1, \ldots, n$
  De même $x \geq y$ est une notation pour $x_j \geq y_j$, $j = 1, \ldots, n$
  On utilise aussi $x > 0$ pour $x_j > 0$, $j = 1, \ldots, n$, etc.

Problème sous forme standard

Définition 2.2. Un problème d'optimisation linéaire sous forme standard est un problème de la forme (PLS) $\min\limits_{\substack{Ax=b\\x\geq 0}} c\cdot x$

Proposition 2.3. Tout problème d'optimisation linéaire peut être mis sous la forme standard (PLS), éventuellement au prix d'un ajout de nouvelles variables

Démonstration. Pour transformer un problème en (PLS) :

- $\max \rightsquigarrow \min$: on transforme $\max(c\cdot x + b)$ en $b - \min(-c)\cdot x$
- $c\cdot x + b \rightsquigarrow c\cdot x$: on transforme $\min(c\cdot x+b)$ en $b + \min c\cdot x$
- x_j quelconque $\rightsquigarrow x_j \geq 0$: on peut écrire un scalaire sans contrainte de signe, $y\in\mathbb{R}$, sous la forme $y = z_1 - z_2$, avec $z_1, z_2 \geq 0$
- $v\cdot x \geq \alpha \rightsquigarrow v\cdot x = \alpha$: on a $v\cdot x\geq\alpha \iff v\cdot x - z = \alpha$ pour un scalaire $z\geq 0$
- $v\cdot x \leq \alpha \rightsquigarrow v\cdot x = \alpha$: on a $v\cdot x\leq\alpha \iff v\cdot x + z = \alpha$ pour un scalaire $z\geq 0$
- Dans les deux derniers cas : $z =$ variable d'écart

□

Exemple

Le problème $\max(x_1 - x_2 + 3)$ sous $x_1 + x_2 \geq 1$, $x_1 \leq 0$,
équivaut à $\min(z_1 + z_2 - z_3)$ sous $z_1, z_2, z_3, z_4 \geq 0$, $-z_1 + z_2 - z_3 - z_4 = 1$

Élimination des contraintes redondantes

Définitions 2.4. Soit $A\in M_{m,n}(\mathbb{R})$. La matrice A est pleine si son rang est m
Si le rang de A est $< m$, alors A est déficiente

Proposition 2.5. *Hypothèse :*
le système $Ax = b$ est compatible
Conclusion :
il existe une matrice pleine $\tilde{A}$ et un $\tilde{b}$ tels que

$$\{x\in\mathbb{R}^n\ ;\ Ax = b\} = \{x\in\mathbb{R}^n\ ;\ \tilde{A}x = \tilde{b}\}$$

Démonstration. – Par récurrence sur m, le nombre de lignes de A

- Le cas $m = 1$:
  si le rang de A est 1, on prend $\tilde{A} = A$ et $\tilde{b} = b$
  si le rang de A est nul, on prend $\tilde{A} = 0$ et $\tilde{b} = 0$
- Passage de $m-1$ à m :
  si A est pleine, on prend $\tilde{A} = A$ et $\tilde{b} = b$

si A est déficiente, alors il existe $j \in \{1, \ldots, m\}$ tel que $l^j(A) = \sum_{k \neq j} \lambda_k l^k(A)$, d'où $b_j = \sum_{k \neq j} \lambda_k b_k$ (par compatibilité du système)
La matrice $\tilde{A}$ obtenue en privant A de la j^e ligne et le vecteur $\tilde{b}$ obtenu en privant b de b_j satisfont $\tilde{A}x = \tilde{b}$, alors que $\tilde{A}$ a $(m-1)$ lignes. On peut donc remplacer $\tilde{A}$ (donc A) par une matrice pleine

$\square$

Problème linéaire encore plus standard

- De ce qui précède, on peut toujours supposer A pleine, c'est-à-dire rang $A = m$, ce qui implique $m \leq n$
- Si $m = n$, alors la solution de $Ax = b$ est unique et le problème d'optimisation trivial
- On supposera donc toujours dans la suite :
  - A pleine
  - $m < n$

2.2 Polytopes

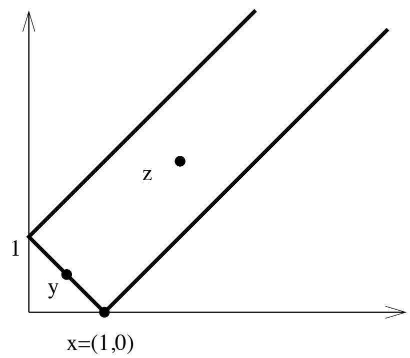
Définitions 2.6.
- Un polytope convexe est un ensemble $\mathcal{P}$ de points de $\mathbb{R}^n$ défini par un nombre fini de contraintes affines satisfaites au sens large
- Un polytope convexe en forme standard est un ensemble de la forme $\mathcal{P} = \{x \in \mathbb{R}^n \; ; \; Ax = b, x \geq 0\}$, avec A pleine de rang $m < n$
- x est un sommet de $\mathcal{P}$ si et seulement si : $x \in \mathcal{P}$ et il n'existe pas $y, z \in \mathcal{P}$ avec $y \neq z$ et $x \in (y, z)$
- Une base de A (avec A pleine de rang $m < n$) est une partie $\mathcal{B} = \{j_1, \ldots, j_m\} \subset \{1, \ldots, n\}$ telle que les colonnes d'indices $j_1, \ldots, j_m$ de A soient libres, ce qui équivaut à : la matrice carrée B formée avec les lignes de A et les colonnes d'indices $j_1, \ldots, j_m$ de A est inversible

Exemple

Le polytope ci-dessus est donné par les contraintes

$$x_1 + x_2 \geq 1, \; x_1 - x_2 \leq 1, \; x_1 - x_2 \geq -1$$

x est un sommet, y et z ne le sont pas



Exemple

Pour la matrice $A = \begin{pmatrix} 1 & 2 & 2 & 3 \\ 1 & 2 & 3 & 4 \end{pmatrix}$, les bases sont $\{1, 3\}$, $\{1, 4\}$, $\{2, 3\}$, $\{2, 4\}$, $\{3, 4\}$

Soient A une matrice pleine, $\mathcal{B}$ une base de A et B la sous-matrice de A associée à $\mathcal{B}$
Après permutation de colonnes, on peut identifier $A \sim (B \mid N)$
Après permutation des indices, un point x de $\mathbb{R}^n$ s'identifie à $x \sim (x_B, x_N)$, avec $x_B \in \mathbb{R}^m$ et $x_N \in \mathbb{R}^{n-m}$, de sorte que $Ax = Bx_B + Nx_N$
Les coordonnées de x_B sont les coordonnées en base, les coordonnées de x_N sont les coordonnées hors base
Si $j \in \mathcal{B}$, alors j est une variable en base ; si $j \notin \mathcal{B}$, alors j est une variable hors base

Exemple

Soient $A = \begin{pmatrix} 1 & 2 & 2 & 3 & 3 \\ 1 & 2 & 3 & 4 & 4 \end{pmatrix}$ et $\mathcal{B} = \{1, 3\}$
Alors $B = \begin{pmatrix} 1 & 2 \\ 1 & 3 \end{pmatrix}$, $N = \begin{pmatrix} 2 & 3 & 3 \\ 2 & 4 & 4 \end{pmatrix}$, $x_B = (x_1, x_3)$, $x_N = (x_2, x_4, x_5)$
Les coordonnées x_1, x_3 sont en base, les coordonnées x_2, x_4, x_5 sont hors base
Les variables 1 et 3 sont en base, les variables $2, 4$ et 5 sont hors base

Définitions 2.7. Soient A une matrice pleine, $\mathcal{B}$ une base de A et $A \sim (B \mid N)$ la décomposition de A associée à $\mathcal{B}$

- La solution de base associée à $\mathcal{B}$ et au système $Ax = b$ est $x \sim (x_B, 0)$, avec x_B la seule solution de $Bx_B = b$. On désigne aussi cet x par $x^{\mathcal{B}}$
  Ou encore : c'est l'unique solution de $Ax = b$ telle que $x_N = 0$
- La solution de base x est admissible si et seulement si $x_B \geq 0$, ce qui équivaut à $x \in \mathcal{P}$
- La base $\mathcal{B}$ est admissible si et seulement si $x^{\mathcal{B}}$ est admissible

Théorème 2.8. Soit $\mathcal{P} = \{x \in \mathbb{R}^n \ ; \ Ax = b, x \geq 0\}$ un polytope convexe en forme standard (donc A est pleine)
Alors x sommet de $\mathcal{P} \iff x$ solution de base admissible pour une certaine base $\mathcal{B}$ de A

Solution de base implique sommet. Si $y, z \in \mathcal{P}$ et $t \in]0,1[$ sont tels que $x = (1-t)y + tz$, alors (1) $x_B = (1-t)y_B + tz_B$ et (2) $x_N = (1-t)y_N + tz_N$
De (2) et $y_N, z_N \geq 0$, on a $y_N = z_N = 0$
De $Ax = Ay = Az = b$, on a $Bx_B = By_B = Bz_B = b$, d'où $x_B = y_B = z_B$, ce qui implique $x = y = z$ □

Sommet implique solution de base. Après permutation, $A \sim (C \mid M)$, avec $x_C > 0$ et $x_M = 0$
Soit w solution de $Cw = 0$. Pour $\varepsilon > 0$ petit, on a $x_C \pm \varepsilon w > 0$ et $x = 1/2y + 1/2z$, avec $y \sim (x_C - \varepsilon w, 0)$, $z \sim (x_C + \varepsilon w, 0)$. Comme $y, z \in \mathcal{P}$, on doit avoir $y = z$, d'où $w = 0$
Il s'ensuit que C est injective, c'est-à-dire les colonnes de C sont libres. On peut compléter C à une matrice carrée B, extraite de A et de même rang que celui de A. B correspond alors à une base $\mathcal{B}$ de A
Avec $A \sim (B \mid N)$, on a $x_N = 0$, d'où x est la solution de base associée à $\mathcal{B}$
□

Au passage, nous avons établi le résultat suivant

Corollaire 2.9. Soit $x \in \mathcal{P}$. Soit C la sous matrice de A correspondant aux coordonnée strictement positives de A, c'est-à-dire : $A \sim (C \mid M)$ et $x \sim (x_C, x_M)$, avec $x_C > 0$ et $x_M = 0$
Hypothèse :
C est injective, ou encore l'équation $Cw = 0$ n'a que la solution $w = 0$, ou encore les colonnes de C sont libres
Conclusion :
x est un sommet
Par ailleurs, la réciproque est vraie

2.3 Recherche d'un optimum

Théorème 2.10. On considère le (PLS) $\min\limits_{\substack{Ax=b \\ x\geq 0}} c \cdot x$
Hypothèse :
il existe une solution x du (PLS)
Conclusion :
il existe une solution y du (PLS) qui soit, de plus, un sommet de $\mathcal{P}$

Démonstration. On considère, parmi toutes les solutions $z \in \mathcal{P}$ du (PLS), une, y, qui a le moins de coordonnées strictement positives. Soit $A \sim (C \mid M)$, $y \sim (y_C, y_M)$, avec $y_C > 0$ et $y_M = 0$
Soit w solution de $Cw = 0$. D'abord, on a $c_C \cdot w = 0$ (1). En effet, si $z \sim (y_C \pm \varepsilon w, 0)$, alors z est admissible pour ε petit. Comme y est optimal, on trouve $\pm \varepsilon c_C \cdot w \geq 0$, d'où $c_C \cdot w = 0$
On montre ensuite que $w = 0$ (d'où y sommet). Preuve par l'absurde : si $w \neq 0$, on peut supposer, par exemple, $w_1 > 0$. Soit $\varepsilon := \min\limits_{w_j > 0} x_j/w_j$>0
Soit $z \sim \varepsilon(w, 0)$ et soit $x := y - z$. Alors : $x \geq 0$, $x_M = 0$ et $x_C \not> 0$. Par ailleurs, $Ax = Cx_C + Mx_M = Ay - Cz = Ay = b$. Donc x est une solution du (PLS) ayant moins de

coordonnées strictement positives que y. Par ailleurs, on a aussi $c \cdot x = c \cdot y$ (grâce à (1)), d'où x optimal. Contradiction ! $\square$

Recherche naïve d'un optimum

Algorithme

Hypothèse :

il existe une solution du (PLS)

Pour en trouver une :

- On initialise $z = \infty$
- On considère toutes les parties de m éléments $\mathcal{B} \subset \{1, \ldots, n\}$
- On vérifie si la matrice B associée à $\mathcal{B}$ est inversible
- Sinon, on passe au $\mathcal{B}$ suivant. Si oui, on calcule $x_B := B^{-1}b$
- Si $x_B \ngeq 0$, on passe au $\mathcal{B}$ suivant. Sinon, on remplace z par $\min\{z, c_B \cdot x_B\}$, où $c \sim (c_B, c_N)$
- À la fin, z est le minimum du (PLS)

Inconvénients

- Nombre prohibitif de calculs : C_n^m parties $\mathcal{B}$ à regarder, résolution de $Bx_B = b$ pour chaque base
- On peut obtenir un z fini même si l'infimum du (PLS) vaut $-\infty$

Exemples

- Pour le problème (PLS) $\min\limits_{\substack{x_1=1 \\ x_1,x_2 \geq 0}} (x_1 - x_2)$, on a $z = 1$ alors que l'infimum est $-\infty$
- Pour un problème avec 100 inconnues et 10 contraintes, il faut regarder C_{100}^{10} familles de vecteurs de $\mathbb{R}^{10}$. Le nombre de familles est à douze chiffres

2.4 Algorithme du simplexe à une phase

Calcul de base

- Soient $\mathcal{B}$ une base admissible, B la matrice associée, $A \sim (B \mid N)$, et $x \sim (x_B, x_N)$ un point de $\mathcal{P}$
- Alors $x_B = B^{-1}(b - Nx_N)$ (car $Ax = b$)
- On a donc

$$c \cdot x = c^T x = c_B^T B^{-1} b + (c_N^T - c_B^T B^{-1} N) x_N$$

Définition 2.11. Le vecteur $\overline{c}_N := (c_N^T - c_B^T B^{-1} N)^T = c_N - N^T (B^{-1})^T c_B$ est le vecteur des coûts réduits (associé à la base $\mathcal{B}$)

Étape $\#0$

On part d'une base admissible $\mathcal{B}$

Étape $\#1$
Si $\overline{c}_N \geq 0$, STOP : $x \sim (B^{-1}b, 0)$ est solution optimale et $z = c_B^T B^{-1} b$ est le minimum du (PLS)

Démonstration. Si $x \in \mathcal{P}$, alors $c \cdot x = c^T x = c_B \cdot B^{-1}b + \overline{c}_N \cdot x_N \geq c_B \cdot B^{-1}b$, car $x_N \geq 0$ et $\overline{c}_N \geq 0$
Le minimum est atteint quand $x_N = 0$, ce qui donne $x \sim (B^{-1}b, 0)$ □

Numérotation des coordonnées : un exemple
Si $n = 5$, $m = 2$ et $\mathcal{B} = \{1, 3\}$, alors les coordonnées de $\overline{c}_N$ ont comme indices $2, 4$ et 5

Par abus de langage, la première, deuxième, troisième coordonnée de $\overline{c}_N$ sont appelées deuxième, quatrième et cinquième coordonnée

Étape $\#2$
Si $\overline{c}_N \ngeq 0$, soit c^i la première coordonnée < 0 de $\overline{c}_N$ et soit i son rang
Soit aussi v^i la i^{e} colonne de $B^{-1}A$ (par abus de langage, v^i est appelée la i^{e} colonne de $B^{-1}N$)

Étape $\#3$
Si $v^i \leq 0$, STOP : l'inf du (PLS) est $-\infty$ et n'est pas atteint

Démonstration. Pour simplifier les notations, on suppose $\mathcal{B} = \{1, \ldots, m\}$, $i = m+1$
Soit $f = (1, 0, \ldots, 0) \in \mathbb{R}^{n-m}$. Soit $t \geq 0$. On pose $x_N = t\, f \geq 0$ et $x_B = B^{-1}b - B^{-1}N\, t\, f = B^{-1}b - B^{-1}N\, t\, f = B^{-1}b - tv^i \geq 0$. Soit $x \sim (x_B, x_N) \in \mathcal{P}$

On a $c \cdot x = c_B^T B^{-1}b + t(c_N^T - B^{-1}N)f = c_B^T B^{-1}b + t\, c^i \to -\infty$ quand $t \to \infty$ □

Numérotation des coordonnées : un exemple
Le colonnes de $B^{-1}A$ ou $B^{-1}N$ ont m coordonnées. On numérote ces coordonnées selon les indices qui donnent $\mathcal{B}$
Exemple : si $\mathcal{B} = \{1, 3, 4\}$ et une colonne de $B^{-1}N$ est $d = (2, 4, 3)^T$, alors 2 est la coordonnée de rang 1, 4 de rang 3 et 3 de rang 4. Donc $d = (d_1, d_3, d_4)^T$
Même numérotation pour les coordonnées de $B^{-1}b$

Étape $\#4$
Si $v^i \nleq 0$, soit $\mathcal{J} := \{k \in \mathcal{B}\,;\ v_k^i > 0\} \neq \emptyset$
On pose $t_k = \dfrac{(B^{-1}b)_k}{v_k^i}$, $\forall\, k \in \mathcal{J}$
On choisit le plus petit j tel que $t_j = \min\limits_{k \in \mathcal{J}} t_k$
On remplace $\mathcal{B}$ par $\tilde{\mathcal{B}} := \mathcal{B} \setminus \{j\} \cup \{i\}$

Rappel

Lemme 2.12 (de substitution)**.** Soit $\mathcal{F} = (e_k)_{k \in I}$ une famille libre dans un espace vectoriel V. Soit $f \in V$, $f = \sum\limits_k \lambda_k f_k$. Si $\lambda_i \neq 0$, alors $\mathcal{F} \setminus \{f_i\} \cup \{f\}$ est encore une famille libre

Proposition 2.13 (L'étape $\#4$ diminue le coût)**.** $\tilde{\mathcal{B}}$ est une base admissible et $c \cdot x^{\tilde{\mathcal{B}}} \leq c \cdot x^{\mathcal{B}}$

$\tilde{\mathcal{B}}$ est une base. On désigne par A^k les colonnes de A
Il faut montrer que la famille $\{A^k\ ;\ k \in \mathcal{B}\} \setminus \{A^j\} \cup \{A^i\}$ est libre
Avec $\lambda^T := (\lambda_k)_{k\in\mathcal{B}}$, on a

$$A^i = \sum_{k\in\mathcal{B}} \lambda_k A^k \Longleftrightarrow A^i = B\lambda \Longleftrightarrow \lambda = B^{-1}A^i = v^i$$

Comme $v^i_j > 0$, la conclusion suit du lemme de substitution □

$\tilde{\mathcal{B}}$ est une base admissible. On peut supposer $\mathcal{B} = \{1, \ldots, m\}$ et $i = m+1$
Avec des notations déjà utilisées, soient $x_N = t_j f$, $x_B = B^{-1}b - B^{-1}Nx_N$, de sorte que $x \sim (x_B, x_N)$ vérifie $Ax = b$, $x_{m+1} \geq 0$ et $x_k = 0$ si $k \geq m+2$
Pour conclure, il suffit de montrer que $x_k \geq 0$, $k = 1, \ldots, m$ et $x_j = 0$
Pour $k = 1, \ldots, m$, on a $x_k = (B^{-1}b)_k - t_j v^i_k \geq 0$. Par choix de j, on a $x_j = 0$ □

Corollaire 2.14. Dans la base $\mathcal{B}$, on a $x^{\tilde{\mathcal{B}}} \sim (B^{-1}b - t_j\, v^i, t_j\, f)$

Le coût diminue en passant de $\mathcal{B}$ à $\tilde{\mathcal{B}}$. Dans la base $\mathcal{B}$ on a $x^{\tilde{\mathcal{B}}} \sim (B^{-1}b - t_j\, v^i, t_j\, f)$, de sorte que

$$\begin{aligned} c \cdot x^{\tilde{\mathcal{B}}} &= c_B^T B^{-1}b + t_j(c_N^T\, f - c_B^T(B^{-1}N)^i) \\ &= c_B^T B^{-1}b + t_j(c_N^T - c_B^T B^{-1}N)f = c_B^T B^{-1}b + t_j\, c^i \leq c_B^T B^{-1}b = c \cdot x^{\mathcal{B}} \quad \square \end{aligned}$$

Remarque
Si $B^{-1}b > 0$, alors $t_j > 0$ pour tout $j \in \mathcal{J}$

Définition 2.15. Une base $\mathcal{B}$ est non dégénérée si $B^{-1}b > 0$

Théorème 2.16 (non dégénérescence $\Longrightarrow$ convergence)**.** *Hypothèse :*
Les bases admissibles sont non dégénérées
Conclusion :
L'algorithme du simplexe donne en un nombre fini d'itérations la solution du (PLS)

Démonstration. Si on passe par l'étape $\#4$, on trouve $c \cdot x^{\tilde{\mathcal{B}}} < c \cdot x^{\mathcal{B}} \Longrightarrow$ le coût décroît strictement $\Longrightarrow$ on ne passe pas deux fois par la même base $\Longrightarrow$ le nombre d'étapes est fini $\Longrightarrow$ on s'arrête à une étape type $\#1$ ou $\#2 \Longrightarrow$ on trouve la solution du (PLS) □

Théorème 2.17 (Bland 1977 ; admis)**.** L'algorithme du simplexe donne, en un nombre fini d'étapes, la solution optimale du (PLS)

Voir une preuve à l'adresse
http ://www.cmi.univ-mrs.fr/∼lugiez/Enseignement/Master1/RO/Cours/cours3.pdf
pages 6 et 7
L'argument essentiel est : si on passe plusieurs fois par une étape du type $\#4$, alors les bases $\mathcal{B}$ correspondantes sont différentes (l'algorithme ne cycle pas)

Algorithme du simplexe à une phase
Entrées :

- un (PLS)
- une base admissible

Sorties :
- le coût optimal z du (PLS)
- si $z > -\infty$, une solution optimale $x^{\mathcal{B}}$

Corollaire 2.18. Si le coût optimal d'un problème de programmation linéaire n'est pas $-\infty$, alors il existe une solution optimale

Inconvénient du simplexe à une phase : suppose connue une base admissible, qui peut être difficile à trouver dans la pratique
Un cas particulier où il y a une base admissible "toute faite" est le $(PL\Sigma)$ $\min\limits_{\substack{Ax\leq b\\ x\geq 0}} c \cdot x$, avec
- $A \in M_{m,n}(\mathbb{R})$
- $b \in \mathbb{R}^m$, $b \geq 0$

Comment obtenir une base admissible pour $(PL\Sigma)$:
- on introduit les variables d'écart x_j, $j = n+1, \ldots, n+m$, de sorte que $Ax+(x_{n+1}, \ldots, x_{n+m})^T = b$
- alors $\{n+1, \ldots, n+m\}$ est une base admissible

2.5 Algorithme du simplexe à deux phases

Phase $\#0$
Entrée :
Un (PLS)
Sortie :
- si $\mathcal{P} = \emptyset$, l'information $\mathcal{P} = \emptyset$
- si $\mathcal{P} \neq \emptyset$, une base admissible $\mathcal{B}$

Phase $\#1$ **: celle du simplexe à une phase**
Entrées :
- un (PLS)
- une base admissible

Sorties :
- le coût optimal z du (PLS)
- si $z > -\infty$, une solution optimale $x^{\mathcal{B}}$

Simplexe à deux phases : phase $\#0$

Étape $\#1$
On transforme $A \rightsquigarrow \tilde{A}$, $b \rightsquigarrow \tilde{b}$ de sorte que $\tilde{b} \geq 0$

Pour ce faire, si $b_j \geq 0$, alors on ne modifie ni b_j, ni $l^j(A)$. Si $b_j < 0$, alors on remplace b_j par $-b_j$ et $l^j(A)$ par $-l^j(A)$
Cette étape ne change pas le rang de A

Ainsi, il suffit de traiter la phase $\#0$ sous les hypothèses :
- A pleine, rang $A < n$
- $b \geq 0$

Étape $\#2$
On résout, avec l'algorithme du simplexe à une phase, le problème auxiliaire (en forme standard)

$$(PA) \min_{\substack{Ax+y=b \\ x\geq 0, y\geq 0}} (y_1 + \ldots + y_m),$$

problème :
- de variable $X = (x_1, \ldots, x_n, y_1, \ldots, y_m) \equiv (X_1, \ldots, X_{m+n})$
- de matrice $(A \mid I_m)$, qui est pleine
- de base admissible $\mathcal{B} = \{n+1, \ldots, n+m\}$, qui donne la solution de base admissible $X \sim (0, b)$ (c'est ici que la première étape sert)

Pour le problème (PA), le coût est toujours ≥ 0. Il s'ensuit qu'il existe toujours une solution optimale de base, associée à une base $\mathcal{B}$, et donnant le coût optimal z_0 de (PA)

Étape $\#3$
Si $z_0 > 0$, STOP : on a $\mathcal{P} = \emptyset$. Les contraintes du (PLS) sont incompatibles

Démonstration. Par l'absurde. Si $x \in \mathcal{P}$, alors $X \sim (x, 0)$ est admissible pour (PA) et le coût associé à $X \sim (x, 0)$ est nul. Contradiction avec $z_0 > 0$ □

Étape $\#4$
Si $z_0 = 0$, alors $\mathcal{P} \neq \emptyset$

Démonstration. Soit $X \sim (x, y)$ une solution optimale. On a alors :
- $y = 0$
- $Ax = b$
- $x \geq 0$

□

Étape $\#5$
On écrit $(A|I_m) \sim (C\,D \mid E\,F)$, avec les sous-matrices choisies de sorte que la solution optimale de base de (PA) s'écrive $X \sim (x_C, x_D, y_E, y_F)$, avec $x_C > 0$, et y_E en base, alors que x_D et y_F sont hors base
Alors les colonnes de C sont libres
On complète C à une sous-matrice $B \sim (C \mid G)$ de A, carrée et de rang m. Les indices des colonnes de B forment une base admissible $\mathcal{B}$ de A, car si $A \sim (C \mid G \mid N)$, alors $x^{\mathcal{B}} \sim (x_C, 0, 0)$

Comment compléter C : nous le verrons dans la section "tableau du simplexe"
On est à la fin de la phase $\#0$: on dispose d'une base admissible $\mathcal{B}$ pour le (PLS)

2.6 Tableau du simplexe

Pour le calculer, il faut :
- un (PLS)
- une base admissible $\mathcal{B}$, de matrice associée B

Le tableau du simplexe est

$B^{-1}A$	$B^{-1}b$
$c^T - c_B^T B^{-1}A$	w

avec $w = -c_B^T \cdot x^{\mathcal{B}}$ (donc w est l'opposé du coût estimé en la solution admissible de base correspondant à $\mathcal{B}$)
C'est un tableau à $(m+1)$-lignes et $(n+1)$-colonnes

Après permutation éventuelle des colonnes, on a $A \sim (B \mid N)$, $c \sim (c_B, c_N)$, et le tableau devient

I_m	$B^{-1}N$	$B^{-1}b$
0	$c_N^T - c_B^T B^{-1}N$	w

Proposition 2.19 (Pivotage du tableau)**.** Quand on passe de la base $\mathcal{B}$ à une nouvelle base $\tilde{\mathcal{B}} = \mathcal{B} \setminus \{i_j\} \cup \{i\}$ (avec $i_j \in \mathcal{B}$ et $i \notin \mathcal{B}$), le tableau du simplexe change de la façon suivante : on fait un pivot de Gauss par dans la i$^{\text{e}}$ colonne, en prenant comme pivot $(B^{-1}A)_j^i$
Plus spécifiquement, si $(t_k^l)_{k=1,\ldots,m+1}^{l=1,\ldots,n+1}$ sont les éléments du tableau correspondant à la base $\mathcal{B}$ et si l^k, $k = 1, \ldots, m+1$ sont les lignes de ce même tableau, alors on obtient les nouvelles lignes $\tilde{l}^k$ de la manière suivante :
- $\tilde{l}^j = l^j / t_j^i$
- pour $k \neq j$, $\tilde{l}^k = l^k - \dfrac{t_k^i}{t_j^i} l^j$

Pratique du simplexe à une phase

Étape $\#1$
On regarde le vecteur ligne $c_N^T - c_B^T B^{-1}N$ (c'est la partie hors base de la dernière ligne du tableau)

– Si toutes ses coordonnées sont ≥ 0, STOP : la base $\mathcal{B}$ est optimale, la solution optimale est $x^{\mathcal{B}} \sim (B^{-1}b, 0)$, le coût optimal est $-w$
– Sinon, on passe à l'étape $\#2$

Étape $\#2$
On trouve le premier i tel que la i^{e} coordonnée de $c_N^T - c_B^T B^{-1} N$ soit < 0
On passe à l'étape $\#3$

Étape $\#3$
Soit v^i la i^{e} colonne de $B^{-1}A$
– Si $v^i \leq 0$, STOP : l'inf du (PLS) est $-\infty$ et n'est pas atteint
– Sinon, on passe à l'étape $\#4$

Étape $\#4$
Soit $\mathcal{B} = \{j_1, \ldots, j_m\}$. On considère
– $\mathcal{J} := \{k \ ; \ v_k^i > 0\}$
– $t_k = \dfrac{(B^{-1}b)_k}{v_k^i}$, $k \in \mathcal{J}$

On considère $k \in \mathcal{J}$ tel que t_k soit le plus petit possible. En cas de ballotage, on choisit le k qui donne le plus petit j_k
On change $\mathcal{B}$ en $\{j_1, \ldots, j_{k-1}, i, j_{k+1}, \ldots, j_m\}$
On change le tableau du simplexe en utilisant la technique du pivotage du tableau (i rentre, k sort)

Passage de la phase $\#0$ **à la phase** $\#1$
La phase $\#0$ donne (après résolution du (PA)) une base $\mathcal{B} = \{j_1, \ldots, j_s, p_1, \ldots, p_r\}$, avec les j_l correspondant à des variables x, les p_l à des variables d'écart y
Pour passer à la phase $\#1$, il faut remplacer, dans $\mathcal{B}$, les variables y par des variables x
Pour ce faire :
– Pour chaque $l = 1, \ldots, r$, on considère la ligne L^l du tableau correspondant à la variable p_l
– On prend un $i \in \{1, \ldots, n\}$ tel que $L^i \neq 0$ (on peut montrer qu'un tel i existe toujours)
– On sort de $\mathcal{B}$ la variable p_l et on fait rentrer la variable i (selon l'algorithme de pivotage)

2.7 Programmation linéaire avec R

Le logiciel R traite des problèmes d'optimisation linéaire pas forcément en forme standard : la seule obligation est la positivité des variables. Les autres contraintes peuvent apparaître comme des contraintes d'égalité ou d'inégalité au sens large

Exemple
Le problème $\min(x_1 + x_2 + x_3)$ sous les contraintes $x_1, x_2, x_3 \geq 0, x_1 - x_2 \leq 1, x_1 + x_3 \geq 3, x_1 - x_2 - x_3 = 1$ est acceptable pour R

Il faut charger le package lpSolve

- On rentre le vecteur de coût c par la commande
  $> f.obj < -c($là on met les coordonnées de c, séparées par des virgules$)$
- On rentre la matrice A des contraintes par la commande
  $> f.con < -matrix$ $(c($là on met les éléments de A, par ligne et séparés par des virgules, $nrow =$là on met le nombre de lignes de A, $byrow = TRUE)$
- On rentre les signes des contraintes dans $Ax?b$ par la commande
  $> f.dir < -c($là, on met les signes sous la forme $" = "$ ou $" <= "$ ou $" >= "$, séparés par des virgules$)$
- On rentre b par la commande
  $> f.rhs < -c($là, on met les coordonnées de b séparées par des virgules$)$

- La commande
  $> lp("max"$ ou $"min", f.obj, f.con, f.dir, f.rhs)$
  donne le coût optimal
- La commande
  $> lp("max"$ ou $"min", f.obj, f.con, f.dir, f.rhs)\$solution$
  donne une solution optimale de base

Exemple

Pour résoudre le problème $\max(x_1+9x_2+x_3)$ sous les contraintes $x_1, x_2, x_3 \geq 0$, $x_1+2x_2+3x_3 \leq 9$ et $3x_1 + 2x_2 + 2x_3 \leq 15$:
- $> f.obj < -c(1, 9, 2)$
- $> f.con < -matrix(c(1, 2, 3, 3, 2, 2), nrow = 2, byrow = TRUE)$
- $> f.dir < -c(" <= ", " <= ")$
- $> f.rhs < -c(9, 15)$

Les résultats s'obtiennent avec les commandes
- $> lp("max", f.obj, f.con, f.dir, f.rhs)$
  qui donne le coût optimal 40.5
- $> lp("max", f.obj, f.con, f.dir, f.rhs)\$solution$
  qui donne une solution optimale de base $(0, 4.5, 0)^T$

2.8 Exercices

Exercice : Mettre le système $x_1 + x_2 \leq 1$, $x_1 - x_2 \leq 1$, $x_1 \geq 0$, $x_2 \geq 0$ sous forme standard
Pour le système standard, trouver les bases et les solutions de base admissibles

Exercice : Mêmes questions pour $x_1 + x_2 \leq 3$, $x_1 + x_3 \leq 7$, $x_1 \geq 0$, $x_2 \geq 0$, $x_3 \geq 0$

Exercice : Résoudre, par énumération des sommets, le problème $\max(x_1+2x_2)$ sous contraintes $x_1+x_2 \leq 1$, $x_1 - x_2 \leq 1$, $x_1 \geq 0$, $x_2 \geq 0$

Exercice : Résoudre le problème précédent par la méthode du simplexe

Exercice : Même question pour $\max(10x_1 + 12x_2 + 12x_3)$ sous $x_1, x_2, x_3 \geq 0$, $x_1 + 2x_2 + 2x_3 \leq 20$, $2x_1 + 2x_2 + x_3 \leq 20$ et $2x_1 + x_2 + 2x_3 \leq 20$

Exercice : Résoudre $\min(x_1 + x_2 + x_3)$ sous contraintes $x_1, x_2, x_3, x_4 \geq 0$, $x_1 + 2x_2 + 3x_3 = 3$, $-x_1 + 2x_2 + 6x_3 = 5$, $4x_2 + 9x_3 = 5$, $3x_3 + x_4 = 1$

Exercice : Idem pour $\min(2x_1+3x_2+3x_3+x_4-2x_5)$ sous $x_1, x_2, x_3, x_4, x_5 \geq 0$, $x_1+3x_2+4x_4+x_5 = 2$, $x_1 + 2x_2 - 3x_4 + x_5 = 2$, $x_1 + 4x_2 - 3x_3 = -1$