Compléments de cours sur l'algorithme du simplexe

III) Difficultés de l'algorithme du simplexe

III.1. Choix de la base initiale (ou méthode du big M)

La première étape de l'algorithme du simplexe consiste à trouver une base admissible initiale, ce qui n'est pas toujours évident (par exemple, pour un problème de minimisation de coût où les contraintes d'objectif ne permettent pas le choix des variables d'écart comme variables de base). On peut faire appel entre autre à la méthode suivante.

Méthode du big M

On considère la programme sous forme standard

$$\max_{\substack{BX=b\\ X\geq 0}} z = q\cdot X, \tag{1}$$

avec $b \geq 0$ (sinon on s'y ramène en multipliant les contraintes d'égalité qui le nécessitent par -1 quitte à changer de signe certaines lignes de B).
On retranche à la fonction objectif la quantité $\widetilde{q}\cdot Y = M\sum_{i=1}^m y_i$ où $\widetilde{q} = M(1,\ldots,1) \in M_{1,m}(\mathbb{R})$, $Y = {}^t(y_1,\ldots,y_m) \geq 0$ et les y_i, $i = 1,\ldots,m$ sont m variables artificielles ≥ 0 et M est un nombre très grand.
On utilise les variables y_i pour introduire artificiellement une matrice identité en considérant le nouveau programme

$$\max_{\substack{BX+Y=b\\ X,Y\geq 0}} \widetilde{z} = q\cdot X - M\sum_{i=1}^m y_i, \tag{2}$$

La matrice standard $\widetilde{B}$ associée à ce programme est $\widetilde{B} = \left(B\,\middle|\,Id_m\right)$. La nouvelle fonction économique $\widetilde{z} = q\cdot X - M\sum_{i=1}^m y_i$ prend des valeurs très petites (en comparaison car M est très grand) lorsque $Y \neq 0$. A l'optimum du programme modifié (2), on aura donc $Y = 0$ et $X = \hat{X}$, où $\hat{X}$ est l'optimum du programme initial (1).
Pour trouver une base initiale pour l'algorithme du simplexe pour le programme (1), on fait tourner l'algorithme du simplexe pour le programme (2) en prenant initialement les variables de Y comme variables de base (et donc X comme variables hors base). Il s'agit bien d'une base admissible car la solution réalisable de base correspondante est donnée par $X = 0$ et donc $Y = b \geq 0$. A la première itération où toutes les variables de Y sont sorties de la base (et donc $Y = 0$), on obtient une base admissible pour le programme (1).

Exemple 1 (Tiré du livre de Y. Dodge)**.** Appliquer la méthode du big M au programme

$$\text{Minimiser} \quad z = -2x_1 + 5x_2 + 3x_3 - 4x_4$$

$$s.l.c.\begin{cases} 4x_1 + 2x_2 - x_3 \leq -5 \\ x_1 + 4x_2 + x_4 = 7 \\ 8x_1 - 7x_2 \geq 10 \\ x_1\ ,\ x_2\ ,\ x_3\ ,\ x_4 \geq 0 \end{cases}$$

en introduisant une seule variable artificielle y.

III.2. Cyclage

C'est la difficulté principale de l'algorithme du simplexe. Si l'algorithme passe par un sommet dégénéré $X(b, I_k)$, il peut se produire que $\theta_{\max} = 0$ et, après avoir interverti variable entrante et sortante, le sommet correspondant à la nouvelle base n'ait pas changé i.e. $X(b, I_{k+1}) = X(b, I_k)$. De sorte que la fonction économique n'est pas améliorée. L'algorithme peut même cycler i.e. revenir à une base à laquelle il est déjà passé : $I_{k+i} = I_k$ pour certains k et i. Il existe des procédures anti-cyclages, mais la rareté des cas de cyclage dans la pratique fait que certains logiciels commerciaux ne les utilisent pas.

Exemple 2 (Tiré du cours de ... à Stanford)**.** Soit le programme linéaire

$$\begin{array}{c} \text{Maximiser} \qquad z = 2x_1 + 3x_2 - x_3 - 12x_4 \\ s.l.c. \left\{ \begin{array}{rcrcrcrcrcrcl} -2x_1 & - & 9x_2 & + & x_3 & + & 9x_4 & + & x_5 & & & = & 0 \\ \frac{1}{3}x_1 & + & x_2 & - & \frac{1}{3}x_3 & - & 2x_4 & & & + & x_6 & = & 0 \\ x_1 & , & x_2 & , & x_3 & , & x_4 & , & x_5 & , & x_6 & \geq & 0 \end{array} \right. \end{array}.$$

Un peu de patience montre que l'algorithme du simplexe peut passer par les bases $I_0 = \{5, 6\}$, $I_1 = \{2, 5\}$, $I_2 = \{1, 2\}$, $I_3 = \{1, 4\}$, $I_4 = \{3, 4\}$, $I_5 = \{3, 6\}$ et finalement $I_6 = \{5, 6\} = I_0$.

III.3. Complexité

L'algorithme du simplexe peut ne pas prendre le plus court chemin pour atteindre l'optimum. Faire rentrer dans la base la variable de plus grande marge réduite n'est pas nécessairement le meilleur choix (même si c'est le plus naturel car c'est la variable qui accroît le plus la fonction économique-mais **localement** seulement). En effet, ce choix ne prend pas en compte la longueur de l'arête correspondante...

Exemple 3 (Un cas particulier du problème de Klee et Minty)**.** Klee et Minty ont montré que, pour une classe de programmes linéaires la complexité de l'algorithme du simplexe est exponentielle (cf. ci-dessous pour plus de détails). Sur le cas particulier suivant, on s'aperçoit bien que le critère de plus grande marge réduite pour l'entrée dans la base n'est pas le meilleur.

Soit le programme linéaire

$$\begin{array}{c} \text{Maximiser} \qquad z = 100x_1 + 10x_2 + x_3 \\ s.l.c. \left\{ \begin{array}{rcrcrcr} x_1 & & & & & \leq & 1 \\ 20x_1 & + & x_2 & & & \leq & 100 \\ 200x_1 & + & 20x_2 & + & x_3 & \leq & 10000 \\ x_1 & , & x_2 & , & x_3 & \geq & 0 \end{array} \right. \end{array}.$$

Partir de la base donnée par les variables d'écart et appliquer l'algorithme du simplexe tel que nous l'avons vu (i.e. avec le critère de plus grande marge réduite pour l'entrée dans la base). Recommencer mais en choisissant maintenant le critère le moins "naturel" i.e. celui de plus faible marge réduite. Conclure.

Dans le pire des cas (réalisé dans les programmes de Klee et Minty), le temps (i.e. le nombre d'itérations) qu'il faut pour que l'algorithme se termine est très grand : de l'ordre de $\exp N$. Ici N est une mesure de la "taille" du problème qui s'explicite à l'aide du nombre n de variables et de la longueur L du codage des données (cf. J.C. Culioli. *Introduction à l'Optimisation* Editions Ellipses). Heureusement, les problèmes pratiques évitent souvent les cas pathologiques. Expérimentalement, en moyenne, le temps nécessaire est de l'ordre de N. On conjecture que N serait la meilleure complexité possible pour un algorithme résolvant tout programme linéaire. Dans les années 80, d'autres algorithmes que l'algorithme du simplexe ont été élaborés. Leur complexité est polynomiale. Ils améliorent donc l'algorithme du simplexe en théorie mais pas forcément dans la pratique. Ils sont plus compliqués car non linéaires.

- 1947 : Algorithme du simplexe par Dantzig (complexité $\exp N$).
- 1979 : Algorithme de Kachian ou Algorithme de l'Ellipsoïde (A.E). De complexité N^6.
- 1984 : Algorithme de Karmarkar. De complexité $N^{3,5}$.