

# **CAHIER DU LAMSADE**

Laboratoire d'Analyse et Modélisation de Systèmes pour l'Aide à la Décision  
(Université de Paris-Dauphine)  
Unité Associée au CNRS n° 825

## **AMELIORATION DES PERFORMANCES DE L'ALGORITHME DE KARMARKAR DANS LE CAS DE PROGRAMMES LINEAIRES A VARIABLES BORNEES SUPERIEUREMENT**

CAHIER N° 82  
novembre 1987

P. TOLLA

# TABLE DES MATIERES

	<u>Pages</u>
ABSTRACT	I
RESUME	I
1. INTRODUCTION	1
2. RESULTAT FONDAMENTAL	2
2.1 Programmes linéaires à variables bornées supérieurement	2
2.2 Théorème fondamental	2
3. APPLICATION A L'ALGORITHME DE KARMARKAR UTILISANT LA FORME PRIMALE DU PROGRAMME LINEAIRE	5
3.1 Généralités	5
3.2 Modification de l'algorithme de Karmarkar	7
4. APPLICATION A L'ALGORITHME DE KARMARKAR UTILISANT LA FORME PRIMALE-DUALE DU PROGRAMME LINEAIRE	9
4.1 Généralités	9
4.2 La matrice $B.B^t$	11
4.3 Modification de l'algorithme de Karmarkar	12
5. EXPERIMENTATION NUMERIQUE	14
BIBLIOGRAPHIE	14

# IMPROVEMENT OF THE KARMARKAR'S ALGORITHM FOR BOUNDED VARIABLES LINEAR PROGRAMS

## ABSTRACT

The solution of linear programming problems by means of the Karmarkar's algorithm needs a very few number of iterations. On the other hand, the cost of each iteration is very high because it must perform the orthogonal projection of the objective function on the polar cone defined by the constraints matrix. When the variables have upper bounds, this matrix has many rows and the orthogonal projection computation is more expensive. We suggest, in this paper, an approximate method which handles implicitly the bounds constraints.

## AMELIORATION DES PERFORMANCES DE L'ALGORITHME DE KARMARKAR DANS LE CAS DE PROGRAMMES LINEAIRES A VARIABLES BORNEES SUPERIEUREMENT

### RESUME

La résolution des programmes linéaires à l'aide de l'algorithme de Karmarkar demande un nombre d'itérations très réduit ; par contre, le coût de chaque itération est très important car il est nécessaire de projeter orthogonalement le vecteur gradient,  $c$ , de la fonction économique sur le cône polaire défini par la matrice des contraintes. Si les variables sont bornées supérieurement, cette matrice comporte un grand nombre de lignes et le calcul de la projection de  $c$  est d'autant plus coûteux. Nous proposons, dans ce cahier, une méthode qui permet de ne pas intégrer les contraintes de borne à la matrice de projection.

## 1. INTRODUCTION

La prise en compte explicite des contraintes de bornes sur les variables d'un programme linéaire présente l'inconvénient d'augmenter de façon très importante la taille de la matrice des contraintes, ce qui peut poser de gros problèmes de stockage en mémoire et de réduction des performances des algorithmes de résolution. C'est pourquoi les logiciels simpliciaux traitent implicitement de ce type de contraintes grâce à une variante très simple du deuxième critère de Dantzig appelée encore règle de pivotage. La partie de très loin la plus coûteuse en temps de calcul de chaque itération de l'algorithme de Karmarkar réside dans le calcul de la matrice  $A$  de projection orthogonale du vecteur gradient de la fonction économique sur le cône polaire défini par la matrice des coefficients des contraintes augmentée d'une ligne de 1 que nous noterons  $B$  :

$$A = I - B^t \cdot [B \cdot B^t]^{-1} \cdot B.$$

En effet, si  $m$  est le nombre de lignes de  $B^t$ , l'inversion de  $B \cdot B^t$  demande de l'ordre de  $O(m^3)$  opérations élémentaires, sa triangularisation de l'ordre de  $O(m^3)$  opérations élémentaires. Si l'on maîtrise bien la convergence de la méthode de Karmarkar qui est à caractère polynomial, l'inversion de  $B \cdot B^t$  est la cause de la supériorité actuelle de l'algorithme du Simplexe sur celle-ci.

Les meilleurs résultats obtenus à l'aide de la méthode de Karmarkar sont dus à l'utilisation de méthodes numériques très sophistiquées et donc à une prise en compte efficace du caractère peu dense de  $B$ . Il semble, de plus, que ce nouvel algorithme soit beaucoup plus rapide que l'algorithme du Simplexe pour des programmes linéaires de faible densité et comportant plus de 2 000 variables.

Nous proposons dans ce cahier une procédure simple et très peu coûteuse qui permet de prendre en compte implicitement les contraintes de bornes sur les variables dans la méthode de Karmarkar.

## 2. RESULTAT FONDAMENTAL

### 2.1 Programmes linéaires à variables bornées supérieurement

Ils se présentent sous la forme :

$$\left\{ \begin{array}{l} \text{Maximiser } c^t \cdot x \\ \text{avec } A \cdot x \leq b \\ 0 \leq x \leq \beta \end{array} \right.$$

où  $c$ ,  $x$  et  $\beta$  sont des vecteurs à  $n$  composantes,  $b$  un vecteur à  $m$  composantes et  $A$  une matrice à  $m$  lignes et  $n$  colonnes.

Soit  $\mathcal{P}$  le polyèdre convexe défini par :

$$\mathcal{P} = \{x \text{ de } \mathbb{R}^n / A \cdot x \leq b, x \geq 0\}$$

et  $P$  le polytope défini par les contraintes de bornes :

$$P = \{x \text{ de } \mathbb{R}^n / x \leq \beta\}.$$

Toute solution réalisable du programme linéaire appartiendra à  $\mathcal{P} \cap P$  qui est évidemment convexe ou vide.

### 2.2 théorème fondamental

Soit  $x^1$  appartenant à l'intérieur strict de  $\mathcal{P} \cap P$  et  $x^2$  à l'intérieur strict de  $P$  et tels que :

$$c^t \cdot x^2 > c^t \cdot x^1.$$

Soit  $I_{cv}(x^2)$  l'ensemble des indices des composantes de  $x^2$  qui ne vérifient pas les contraintes de bornes. Soit  $x^\mu = (1 - \mu)x^1 + \mu x^2$  avec  $0 \leq \mu \leq 1$  une combinaison linéaire convexe de  $x^1$  et  $x^2$ .

Alors, si au moins une contrainte de borne est violée par l'une des composantes de  $x^2$ , le meilleur point au sens de la fonction économique appartenant au segment  $[x^1, x^2]$  et au domaine  $\mathcal{P} \cap P$  est  $x^*$  défini par :

$$x^* = (1 - \mu^*)x^1 + \mu^* x^2$$

avec

$$\mu^* = \min_{i \in I_{cv}(x^2)} \begin{cases} \frac{\beta_i - x_i^1}{x_i^2 - x_i^1} & \text{si } x_i^2 - x_i^1 > 0 \\ \frac{x_i^1}{x_i^1 - x_i^2} & \text{si } x_i^2 - x_i^1 < 0. \end{cases}$$

### Preuve

Tout point  $x^1$  du segment  $[x^1, x^2]$  s'écrit sous forme d'une combinaison linéaire convexe de  $x^1 + x^2$  :

$$x^\mu = (1 - \mu)x^1 + \mu x^2, \quad 0 \leq \mu \leq 1.$$

$x^2$  est meilleur que  $x^1$  au sens de la fonction économique. Le vecteur  $x^2 - x^1$  est une direction d'amélioration de celle-ci ; en effet :

$$f(x^\mu) = c^t \cdot x^\mu = c^t \cdot x^1 + \mu(c^t \cdot x^2 - c^t \cdot x^1).$$

Comme  $\mu$  est positif et  $c^t \cdot x^2 > c^t \cdot x^1$ ,  $f(x^\mu) > f(x^1)$ .

Trouvons le plus grand scalaire strictement positif  $\mu$  tel que  $x^\mu$  appartienne à  $\mathcal{P} \cap P$ .  $x^1$  appartient à l'intérieur strict de  $\mathcal{P} \cap P$ , donc à  $P$  et vérifie strictement les contraintes de bornes :

$$x^1 \in P \quad \text{et} \quad 0 > x_i^1 > \beta_i \quad \text{pour tout } i = 1, \dots, n.$$

Pour que  $x^\mu$  appartienne à  $\mathcal{P} \cap P$ , ses composantes doivent vérifier :

$$x_i^\mu = x_i^1 + \mu(x_i^2 - x_i^1) > \beta_i.$$

Soit  $J_n = \{1, \dots, n\}$ .

- Si  $i$  appartient à  $J_n - I_{CV}(x^2)$ , alors

$$x_i^1 \leq \beta_i \quad \text{et} \quad x_i^2 < \beta_i,$$

donc

$$x_i^\mu = (1 - \mu)x_i^1 + \mu x_i^2 < \beta_i$$

et  $x_i^\mu$  vérifie sa contrainte de borne.

- Si  $i$  appartient à  $I_{CV}(x^2)$ , alors

$$x_i^1 \leq \beta_i \quad \text{et} \quad x_i^2 > \beta_i.$$

On désire que  $x_i^\mu$  vérifie sa contrainte de borne :

$$(1 - \mu)x_i^1 + \mu x_i^2 \leq \beta_i$$

ou encore

$$\mu(x_i^2 - x_i^1) \leq \beta_i - x_i^1. \quad (1)$$

Trois cas peuvent se présenter :

(i) Si  $x_i^2 - x_i^1 < 0$ , alors l'inégalité (1) est vérifiée pour tout  $\mu$  ; si l'on veut que  $x_i^\mu$  reste positif, il suffit que :

$$x_i^2 - \mu(x_i^2 - x_i^1) \geq 0 \quad \text{ou}$$

$$\mu(x_i^1 - x_i^2) \leq x_i^1 \quad \text{ou}$$

$$\mu \leq \frac{x_i^1}{x_i^1 - x_i^2}.$$

(ii) Si  $x_i^2 - x_i^1 > 0$ ,  $x_i^\mu$  respecte sa contrainte de borne supérieure  
 si  $\mu \leq \frac{\beta_i - x_i^1}{x_i^2 - x_i^1}$ .

(iii) Si  $x_i^2 - x_i^1 = 0$ , alors  $x_i^\mu = x_i^1$  et  $x_i^\mu$  vérifie sa contrainte de borne quel que soit  $\mu$ .

### Remarque

Au moins une des composantes de  $x^{\mu^*}$  atteint une de ses bornes ; en effet, il existe  $i'$  appartenant à  $I_{cv}(x^2)$  tel que :

$$\mu^* = \frac{x_{i'}^1}{x_{i'}^1 - x_{i'}^2} \quad \text{si } x_{i'}^2 - x_{i'}^1 < 0$$

$$\mu^* = \frac{\beta_{i'} - x_{i'}^1}{x_{i'}^2 - x_{i'}^1} \quad \text{si } x_{i'}^2 - x_{i'}^1 > 0.$$

Si  $x_{i'}^2 - x_{i'}^1 > 0$ , alors  $x^{\mu^*} = 0$ . Si  $x_{i'}^2 - x_{i'}^1 < 0$ , alors  $x^{\mu^*} = \beta_{i'}$ .

## 3. APPLICATION A L'ALGORITHME DE KARMARKAR UTILISANT LA FORME PRIMALE DU PROGRAMME LINEAIRE

### 3.1 Généralités

Pour pouvoir être traité par l'algorithme de Karmarkar, un programme linéaire doit être mis sous la forme :

$$\left\{ \begin{array}{l} \text{Minimiser } z = c^t \cdot x \\ \text{avec } A \cdot x = 0 \\ e^t \cdot x = 1 \\ x \geq 0 \end{array} \right.$$

où  $e$  est le vecteur dont les  $n$  composantes sont égales à 1.



On peut remarquer que plusieurs auteurs tels que Meggido, Vial et Gay se sont attachés à s'affranchir de la contrainte  $c^t \cdot x = 1$ .

Soit  $B$  la matrice définie par :

$$B = \begin{bmatrix} A \cdot D \\ e^t \end{bmatrix}$$

où  $D = \text{Diag}(d_1, d_2, \dots, d_n)$  est la matrice diagonale de coefficients diagonaux  $d_i$  ; les  $d_i$  sont les composantes de la solution réalisable fournie par l'itération précédente.

Chaque itération exige le calcul du vecteur  $c_p$  projection orthogonale du vecteur  $D \cdot c$  sur l'espace défini par :

$$B \cdot x = 0.$$

Pour cela, Karmarkar propose d'utiliser la matrice de projection  $I - B^t \cdot (B \cdot B^t)^{-1} \cdot B$  :

$$c_p = [I - B^t \cdot (B \cdot B^t)^{-1} \cdot B] \cdot D \cdot c.$$

Le calcul de  $[B \cdot B^t]^{-1}$  ou la triangularisation de  $B \cdot B^t$  demande respectivement un nombre d'opérations élémentaires de l'ordre de  $p^3/2$  ou  $p^3/3$  si  $p$  est la dimension de la matrice  $B \cdot B^t$ .

Cette opération est évidemment de très loin la plus coûteuse de la matrice de Karmarkar ; si l'on désire que celle-ci soit plus rapide que l'algorithme du Simplexe, il faut :

- Avoir un nombre d'itérations beaucoup plus réduit, ce qui est généralement le cas. Il faut remarquer que plusieurs chercheurs tels que Minoux, Lisser et Maculan, puis Zimmermann, ont obtenu un nombre d'itérations plus réduit en autorisant de grands pas de déplacement.

- Disposer d'une procédure de calcul de l'inverse ou de la décomposition triangulaire de  $B \cdot B^t$  extrêmement peu coûteuse car une itération de la méthode du simplexe demande de l'ordre de  $km^2$  opérations élémentaires.

Une première amélioration notable de l'algorithme de Karmarkar consiste à ne pas incorporer dans la matrice  $B$  les contraintes de bornes, ce qui réduit de  $n$  le nombre de lignes de  $B$  et donc de  $B.B^t$ . Avec les contraintes de borne et en rajoutant des variables d'écart pour obtenir des contraintes d'égalité,  $B$  devient, sachant que  $I_m$  et  $I_n$  sont les matrices identités de dimensions respectives  $m$  et  $n$  :

$$B = \begin{array}{|c|c|c|} \hline & A & I_m \\ \hline I_n & & I_n \\ \hline \end{array}$$

$B.B^t$  a donc  $m + n$  lignes et colonnes.

Sans les contraintes de bornes,  $B$  s'écrit :

$$B = \begin{array}{|c|c|} \hline & A \\ \hline I_m & \\ \hline \end{array}$$

$B.B^t$  a donc  $m$  lignes et colonnes.

Nous avons intégré dans ces deux cas la contrainte  $e^t \cdot x = 1$  à la matrice  $A$ .

Si par exemple  $n = 5m$ , le nombre d'opérations élémentaires nécessaires à la projection orthogonale de  $D.c$  sera de l'ordre de  $(6m)^3/3$  lorsque l'on conserve les contraintes de borne et  $m^3/3$  dans le cas contraire. A nombre d'itérations égal, l'algorithme de Karmarkar sera à peu près 200 fois plus rapide dans le second cas que dans le premier !

### 3.2 modification de l'algorithme de Karmarkar

L'algorithme modifié se présente de la façon suivante :

Pas 1 : Soit  $x^{(k)} = (x_1^{(k)}, \dots, x_n^{(k)})^t$  la solution fournie à l'itération précédente et  $D_k = \text{diag}(x_1^{(k)}, \dots, x_n^{(k)})$ . La matrice  $B^{(k)}$  est définie par :

$$B^{(k)} = \left| \frac{A \cdot D_k}{e^t} \right|$$

et n'inclut pas les contraintes de bornes.

Pas 2 : Calculer  $c_p$  projection du vecteur  $D_k \cdot c$  sur le noyau de  $B^{(k)}$  en utilisant la matrice de projection définie précédemment ou une méthode de moindres carrés par exemple. Soit  $u = c_p / \|c_p\|$  le vecteur unité de même direction que .

Pas 3 : Soit  $b^k = e/n - \eta u/n(n-1)$  le point obtenu en faisant un pas de longueur  $\eta$  dans la direction  $-u$  à partir du centre du simplexe  $S$ . Le pas  $\eta$  doit être tel que les composantes de  $b^k$  soient toutes positives.

Pas 4 : Projeter  $b^k$  dans le domaine  $P$  à l'aide de la transformation projective définie par :

$$x'^{(k+1)} = D_k \cdot b^k / [e^t \cdot D_k \cdot b^k].$$

Pas 5 :  $x'^{(k+1)}$  est dans l'intérieur strict de  $\mathcal{P}$ . Si  $x'^{(k+1)}$  vérifie les contraintes de borne, poser  $x^{(k+1)} = x'^{(k+1)}$  et aller au pas 6. Sinon, calculer  $\mu^*$  comme indiqué dans le théorème fondamental et déterminer  $\mu'^{(k+1)}$  par :

$$\mu'^{(k+1)} = \text{Min} \{ \mu^*, 1 \}.$$

Choisir  $\mu^{(k+1)}$  tel que  $0 < \mu^{(k+1)} < \mu'^{(k+1)}$ . Le nouveau point courant est :

$$x^{(k+1)} = (1 - \mu^{(k+1)}) x^{(k)} + \mu^{(k+1)} x'^{(k+1)}$$

et vérifie strictement les contraintes de borne.

Pas 6 : Si  $c \cdot x^{(k+1)} - c \cdot x^{(k)}$  peut être considéré comme un zéro théorique, arrêter l'exécution de l'algorithme ; sinon, revenir au pas 1 en posant  $k := k + 1$ .

Remarques :

- Dans le pas 5, nous avons choisi  $\mu^{(k+1)} < \mu'^{(k+1)}$  ; en effet, l'algorithme de Karmarkar fonctionne dans l'intérieur strict de  $\mathcal{P} \cap P$ . Si nous prenons  $\mu^{(k+1)} = \mu'^{(k+1)}$ , une des variables peut atteindre sa borne supérieure, c'est-à-dire que, pour un indice  $i_0$ ,  $x_{i_0}^{(k+1)} = 1$  ; alors, à l'itération suivante, si  $x_{i_0}^{(k+2)}$  dépasse sa borne et  $\mu^{(k+1)} = 0$ , alors l'algorithme peut cycloter en ce point.

- Dans le même pas, on peut remarquer que, comme  $x_i^{(k)}$  et  $x_i^{(k+1)}$  sont strictement positives, leur quotient est toujours supérieur à l'unité. Or, le pas  $\mu$  est strictement inférieur à 1 ; on peut donc éviter le calcul de cette quantité et le remplacer par 1 dans l'expression de  $\mu^*$ . On peut remarquer dans ce cas que si  $\mu^* = 1$ , le point obtenu à partir de  $\mu^*$  est simplement  $x^{(k+1)}$  qui est dans ce cas intérieur au domaine ; ce cas a déjà été envisagé au début du pas 5.

#### 4. APPLICATION A L'ALGORITHME DE KARMARKAR UTILISANT LA FORME PRIMALE-DUALE DU PROGRAMME LINEAIRE

##### 4.1 Généralités

L'intérêt de la présentation d'un programme linéaire sous forme primale-duale est que l'on dispose d'une solution initiale réalisable évidente, ce qui évite de mettre en oeuvre une première phase de la méthode pour trouver une solution réalisable.

Soit le programme linéaire PL :

$$\left\{ \begin{array}{l} \text{Maximiser } c^t \cdot x \\ \text{avec } A \cdot x \leq b \\ x \geq 0 \end{array} \right.$$

Son dual PL\* s'écrit :

$$\left\{ \begin{array}{l} \text{Minimiser } u^t \cdot b \\ \text{avec } u^t \cdot A \geq c^t \\ u \geq 0 \end{array} \right.$$

où  $u$  est un vecteur à  $m$  composantes.

Résoudre le programme linéaire PL est équivalent à trouver une solution du système d'inéquations linéaires :

$$\left\{ \begin{array}{l} A \cdot x \leq b \\ u^t \cdot A \geq c^t \\ u^t \cdot b - c^t \cdot x = 0 \\ x \geq 0, u \geq 0 \end{array} \right.$$

En utilisant les vecteurs de variables d'écart non négatives  $y$  et  $v$ , on obtient :

$$\left\{ \begin{array}{l} A \cdot x + y = b \\ u^t \cdot A - v^t = c^t \\ u^t \cdot b - c^t \cdot x = 0 \\ x \geq 0, u \geq 0, y \geq 0, v \geq 0 \end{array} \right.$$

Soit les vecteurs  $x^0, y^0, u^0$  et  $v^0$ , respectivement de même dimension que  $x, y, u$  et  $v$  ; le système précédent a une solution si et seulement si le programme linéaire suivant a zéro pour valeur optimale de sa fonction économique :

$$\left\{ \begin{array}{l} \text{Minimiser } X \\ \text{avec } A \cdot x + y + X(b - A \cdot x^0 - y^0) = b \\ u^t \cdot A - v^t + X(c^t - u^{0t} \cdot A + v^{0t}) = c \\ u^t \cdot b = X(u^{0t} \cdot b - c^t \cdot x^0) = 0 \\ x \geq 0, y \geq 0, u \geq 0, v \geq 0, X \geq 0 \end{array} \right.$$

De plus, ce programme linéaire équivalent à PL admet comme solution initiale réalisable :

$$x = x^0, y = y^0, u = u^0, v = v^0, X = 1.$$

#### 4.2 La matrice $B \cdot B^t$

La matrice  $B$  a la forme suivante, sachant que la contrainte  $e^t = 1$  est incluse dans la matrice  $A$  et que l'on y conserve les contraintes de bornes :

$$B = \begin{array}{|c|c|c|c|c|c|c|c|} \hline A & I_m & 0 & 0 & 0 & 0 & b & \\ \hline I_n & 0 & I_n & 0 & 0 & 0 & \beta & \\ \hline 0 & 0 & 0 & A^t & I_n & -I_n & c & \\ \hline c^t & 0 & 0 & b^t & \beta^t & 0 & 0 & \\ \hline \end{array}$$

La dernière colonne contient les coefficients dans les contraintes de la variable  $X$ .

Remarque : Les composantes du vecteur  $c'$ , coefficients de la fonction économique du problème transformé, sont tous nuls en-dehors de la dernière qui vaut 1.

Si l'on exclut les contraintes de bornes de  $B$ , celle-ci devient :

$$B = \begin{array}{|c|c|c|c|c|c|c|} \hline A & I_m & 0 & 0 & 0 & b & \\ \hline 0 & 0 & A^t & I_n & -I_n & & \\ \hline c^t & 0 & b^t & 0 & 0 & 0 & \\ \hline \end{array}$$

Dans ce cas, la matrice  $B.B^t$  est de dimension  $m + n + 1$  alors que, dans le cas précédent, elle était de dimension  $m + 2n + 1$ . Le temps de calcul de  $(B.B^t)^{-1}$  ou de la décomposition triangulaire de  $B.B^t$  est divisé par 8.

Nous remarquons que la méthode que nous proposons est plus intéressante lorsque  $B$  utilise la forme primale que lorsqu'elle est basée sur la forme primale-duale.

#### 4.3 Modification de l'algorithme de Karmarkar

Le programme linéaire avec ses contraintes de bornes s'écrit :

$$PL \left\{ \begin{array}{l} \text{Maximiser } c^t \cdot x \\ \text{avec } A \cdot x \leq b \\ x \leq \beta \\ x \geq 0 \end{array} \right.$$

et son dual :

$$PL^* \left\{ \begin{array}{l} \text{Minimiser } u^t \cdot b + v^t \cdot \beta \\ \text{avec } u^t \cdot A + v^t \geq c^t \\ u \geq 0, v \geq 0 \end{array} \right.$$

Soit  $P^*$  l'ensemble des vecteurs  $u$  et  $v$  vérifiant les contraintes du programme  $PL^*$ .

Soit un triplet  $X' = (x', u', v')$  vérifiant strictement les contraintes de  $PL$  et  $PL^*$  et un triplet  $X'' = (x'', u'', v'')$  vérifiant les contraintes de  $PL$  à l'exception des contraintes de bornes et les contraintes de  $PL^*$ .

Une combinaison convexe de  $X'$  et  $X''$  s'écrira :

$$X^\mu = (1 - \mu)X' + \mu X'' \quad \text{avec } 0 \leq \mu \leq 1.$$

Alors :

$$A \cdot [(1 - \mu)x' + \mu x''] = (1 - \mu)A \cdot x' + \mu A \cdot x'' < b,$$

$$[(1 - \mu)u' + \mu u'']^t \cdot A + (1 - \mu)v' + \mu v'' = (1 - \mu)[u'^t \cdot A + v'] + \mu[\mu'^t \cdot A + v''] \geq c.$$

De même :

$$\begin{aligned} (1 - \mu)x' + \mu x'' &\geq 0, \\ (1 - \mu)u' + \mu u'' &\geq 0, \\ (1 - \mu)v' + \mu v'' &\geq 0. \end{aligned}$$

Donc,  $x^\mu$  vérifiera les contraintes de PL et PL\*, excepté peut-être les contraintes de bornes. Pour déterminer  $\mu$  tel que  $x^\mu$  vérifie les contraintes de bornes, on va utiliser exactement la même procédure que celle qui est décrite dans le paragraphe 2.1.

La matrice B est calculée à partir de la matrice de contraintes suivante :

A	0	0
0	$A^t$	$I_n$

Les cinq premiers pas de l'algorithme sont identiques à ceux du paragraphe 3.2 ; par contre, le sixième pas doit tenir compte du fait que les composantes de  $u^{(k)}$  interviennent dans la fonction économique.

Pas 6 : Si  $c^t \cdot x^{(k+1)} - b^t \cdot u^{(k+1)} + \beta^t \cdot v^{(k+1)}$  peut être considéré comme un zéro théorique, arrêter l'exécution de l'algorithme ; sinon, revenir au pas 1.

### Remarques

Nous avons proposé, dans un article cité en référence, une procédure dérivée de la méthode statistique de J. Vignes qui permet de déterminer si la fonction économique calculée au point courant est un zéro théorique, ce qui permet de stopper l'exécution lorsque des itérations supplémentaires n'apporteraient aucune amélioration de la solution.



L'algorithme de Karmarkar relève des méthodes itératives ; dans le cas de variables bornées, lorsque certaines variables sont très proches de leur borne supérieure qu'elles auraient atteinte si l'on avait effectué le pas maximal, il y a de fortes chances pour, qu'à l'optimum, ces variables atteignent leur borne supérieure. On peut encore réduire le temps de calcul en mettant à leur borne supérieure ces variables et en réduisant ainsi le nombre de variables du problème initial. Cette procédure est souvent utilisée dans les codes de résolution des programmes linéaires utilisant la méthode du simplexe. Dans ce cas précis, on évitera une augmentation du nombre d'itérations lorsque des variables sont très proches de leur borne.

## 5. EXPERIMENTATION NUMERIQUE

Lisser, dans le cadre de sa thèse d'Université, a mis en oeuvre cette procédure. La méthode utilisée s'appuyait sur une méthode primale en deux phases avec utilisation d'une fonction glissante. Les exemples traités étaient des problèmes de planning à court et moyen termes à EDF dont les matrices des contraintes, très peu denses, ne comportaient que des coefficients égaux à 0, 1 ou - 1. Ils comptaient jusqu'à 3 000 variables et 2 500 contraintes, sans compter les contraintes de bornes. Les résultats obtenus sont très encourageants ; en effet, si les solutions obtenues se trouvaient seulement en moyenne à 6 % des solutions optimales, le temps de calcul était près de 10 fois inférieur à celui du code MPSX d'IBM qui constitue, en l'occurrence, une véritable référence.

## BIBLIOGRAPHIE

- G.B. DANTZIG, Linear programming and extensions, Princeton University Press, Princeton, N.J., 1963.
- M.D. GAY, A variant of Karmarkar's linear programming algorithm for problems in standard form, AT&T Laboratories, October 1985.
- N. KARMARKAR, A new polynomial-time algorithm for linear-programming, Combinatorica 4 (4), 1984, 373-395.

A. LISSER, N. MACULAN, M. MINOUX, Large steps preserving polynomiality in Karmarkar's algorithm, Université de Paris-Dauphine, Cahier du LAMSADE n° 77, mai 1987.

N. MEGGIDO, A variation on Karmarkar's algorithm, IBM Research Laboratory, San Jose, CA 95193.

W. NICKERS, W. RODDER, L. XU, H.J. ZIMMERMANN, Intelligent gradient search in linear programming, European Journal of Operational Research 22, 1985, 293-303.

P. TOLLA, Optimal termination criterion and accuracy tests for linear programming, Mathematics and Computers in Simulation (forthcoming).

P. TOLLA, Validation de l'algorithme de Karmarkar, Université de Paris-Dauphine, Cahier du LAMSADE n° 76, mai 1987.

J.P. VIAL, A fully polynomial time projective method, CORE, D.P. n° 8707, février 1987.

J. VIGNES, Implémentation des méthodes d'optimisation : test d'arrêt optimal, contrôle et précision de la solution, RAIRO Recherche Opérationnelle, Vol. 18, n° 1? 1984.