RAIRO. RECHERCHE OPÉRATIONNELLE

J. EDMONDS
J.-F. MAURRAS

Note sur les Q-matrices d'Edmonds

RAIRO. Recherche opérationnelle, tome 31, nº 2 (1997), p. 203-209

<http://www.numdam.org/item?id=RO_1997__31_2_203_0>

Recherche opérationnelle/Operations Research
(vol. 31, n° 2, 1997, p. 203-209)

NOTE SUR LES Q-MATRICES D'EDMONDS (*)

par J. EDMONDS (1) et J.-F. MAURRAS (2)

Communiqué par Catherine ROUCAIROL

Résumé. – *Lorsque l'on veut résoudre un Programme Linéaire dont les données sont des entiers ou des rationnels, et certifier dans tous les cas le résultat, on ne peut pas se contenter de représentation des nombres par les* flottants. *On travaille en entiers* longs. *Récemment, Jack Edmonds [3], a introduit un nouveau tableau, appelé Q-matrice, dont la mise à jour, en entiers longs, se fait simplement. Nous intèrprétons ici les calculs matriciels, en terme d'inverse de base. Ces calculs peuvent être aussi utilisés pour donner une représentation (rationnelle), en entiers longs, de l'inverse d'une matrice d'entiers donnée.*

Mots clés : Programmation linéaire, méthode simplex, calculs matriciels, pivotage, déterminant, matrice inverse.

Abstract. – *When we want to solve Linear Programming Problems with integer or rationnal entries, and, in each case, to certify the result, classical floating point calculations are not allowed. We should work with Long Integers. Recently, Jack Edmonds [3] introduced a new tableau called Q-matrix which update is easy. We revisit these in term of the inverse of the basis. These calculations can be also used to give a rational representation of the inverse of a given integer matrix.*

Keywords: Linear programming, simplex method, matrix calculus, pivoting, determinant, inverse matrix.

1. INTRODUCTION

Des langages de programmation, en particulier les langages de programmation par contraintes, et des démonstrateurs, utilisent la méthode Simplex comme *solveur.* Ces utilisateurs ont besoin de *procédures de décision.* Les calculs flottants auxquels sont habitués les praticiens de la Programmation Linéaire ne peuvent plus les satisfaire. À titre d'exemple Prolog 3 [1], qui travaille au moyen de nombres rationnels représentés par des couples d'entiers, utilise intensivement l'algorithme d'Euclide pour réduire les fractions. Des calculs matriciels exacts sont aussi nécessaires dans d'autres

(*) Recu en juin 1994.
(1) 265 Glasgow Street, Futchener, Ontario, Canada.
(2) Laboratoire d'Informatique de Marseille, pôle sud, Faculté des sciences de Luminy, 163 avenue de Luminy, 13288 Marseille, France.

domaines dans les cas où les matrices sont (très) mal conditionnées. En 1967 déjà, Jack Edmonds [2], a proposé une variante de l'algorithme de Gauss qui permet de s'affranchir des calculs de PGCD en proposant, à chacune des itérations, un diviseur de chacun des nombres modifiés. Cette modification assure que la longueur des nombres produits croit polynomialement. Sauf *hasard* numérique, cette croissance est même la meilleure possible.

Les calculs de la méthode Simplex peuvent se mener au moyen de la simple connaissance de l'inverse de base, c'est ce qui se fait dans la plupart des codes industriels. Une itération de cette méthode peut donc s'effectuer polynomialement au moyen de la modification précédente de l'algorithme de Gauss. Le *changement de base*, à chaque itération, fait passer d'une base à une base voisine dans laquelle une seule colonne a changé. La complexité des calculs traditionnels d'une itération (pivotage), est celle de la modification de chacun des termes de la matrice, et donc en n^2, si n est le nombre de colonnes d'une base, et non en n^3, ce qui est le cas si on réinverse la matrice de base à chaque itération. Il est donc utile de pouvoir faire ce gain de complexité lorsque l'on travaille sur des *entiers longs.* Dans un article récent [3], Jack Edmonds a introduit les Q-Matrices. Son but était d'améliorer la vitesse des calculs lors des pivotages de la méthode Simplex quand on travaille sur ces entiers (longs). Ces Q-Matrices correspondent au *tableau simplicial.*

Pour les utilisateurs de la forme dite *révisée* ou de l'*inverse explicite,* ainsi que pour ceux qui veulent simplement inverser une matrice ou modifier faiblement une matrice inverse, nous les présentons ici au moyen d'une interprétation, en terme de déterminant, des calculs d'un pivotage effectués sur la matrice inverse.

2. CALCULS MATRICIELS

Soient L et C deux ensembles finis et R un anneau. On appelle matrice $A(L, C)$ une application de $L \times C \to R$. Cette définition d'une matrice, apparamment très générale, est tout à fait adaptée aux types de calculs de la méthode Simplex. Les ensembles d'index L et C représentent, dans la pratique, des objets bien concrets. À titre d'exemple, le coefficient A (*Calories, Carottes*) se comprend assez bien comme le nombre de calories apportées par un Kilogramme de carottes et, est plus *parlant*, que le simple A_{ij}. De plus on voit bien ici que son inverse *1/A* (*Calories, Carottes*) est un certain A' (*Carottes, Calories*) qui est la masse de carottes contenant une calorie, et que par inversion *les lignes deviennent des colones et inversement.* D'autres raisons, théoriques, disparition de la notion de sous-matrice entre autre, et suppression de l'inévitable *à une permutation des lignes et des colonnes près...*, plaident en faveur de cette définition.

Lorsque $R = \mathbb{Z}$, A sera une matrice entière. Dans ce paragraphe, les matrices que nous allons considérer seront entières. On sait que la simple suppression des divisions dans l'algorithme de Gauss ne conduit pas à un algorithme polynomial [4]. Cependant le théorème d'Hadamard permet de borner polynomialement la longueur d'écriture du déterminant d'une matrice. Lorsque l'on remarque que toutes les valeurs produites lors de l'application de l'algorithme de Gauss sont des sous-déterminants de la matrice de départ, il vient que la simple réduction par l'algorithme d'Euclide des termes produits par l'algorithme de Gauss conduit à un algorithme polynomial. Pour l'algorithme de Gauss on peut faire beaucoup mieux, c'est ce que fait Jack Edmonds [2] en exploitant une remarque de Vandermonde. Pour cette modification de l'algorithme de Gauss, il n'est pas nécessaire d'interpréter les valeurs produites lors de l'application de cet algorithme comme des sous-déterminants, la polynomialité de la longueur des nombres produits vient de la résolution d'une récurrence. Cette résolution donne aussi la démonstration d'un affaiblissement du théorème d'Hadamard [4].

Comme il est d'usage pour l'initialisation de la méthode Simplex, on *rajoute* à la matrice $A(L, C)$ une matrice identité indicée par les lignes L, soit $A(L, L)$. On remplace alors C par $C \cup L$. Nous supposons donc que $L \subset C$, et que $A(L, L)$ est la matrice identité.

Soit $I \subset C$, supposons que $A(L, I)$ soit une matrice carrée et régulière. L'ensemble I est appelé *base*, la matrice $A(L, I)$ est donc appelée *matrice de base.* Appelons $B(I, L)$ son inverse. Appelons $\overline{A}(I, C)$ le produit $B(I, L) \times A(L, C)$ (et donc $B(I, L) = \overline{A}(I, L)$). Soit q le déterminant de $A(L, I)$. Remarquons au passage que cette matrice $\overline{A}(I, L)$, produit de $B(I, L)$ par $A(L, C)$ est indicé en lignes par I. Remarquons aussi que par construction $\overline{A}(I, I)$ est une matrice *identité* indicée par I.

Remarque 1 : $q \times B(I, L)$ *est une matrice entière.*

Preuve: En effet par définition d'une matrice inverse, $B(I, L)$ est la transposée de la matrice des cofacteurs de $A(L, I)$ divisée par le déterminant de $A(L, I)$. Ce déterminant est justement q. Bien entendu, la matrice des cofacteurs, matrice de sous-déterminants d'une matrice entière, est entière. $\square$

Soit $I' = (I \setminus \{r\}) \cup \{s\}$ une nouvelle base. Nous allons étudier le passage de $B(I, L)$ à $B(I', L)$, et en déduire celui de $q \times B(I, L)$ à $q' \times B(I', L)$, en appelant q' le déterminant de $A(L, I')$.

Notons $E(I, I')$ la matrice $\overline{A}(I, I')$. Ses colonnes pour $j \neq s$ sont des colonnes de la matrice identité $\overline{A}(I, I)$ et sa colonne s est $E(I, s) = \overline{A}(I, s)$.

Lorsque $E(r, s) \neq 0$, la matrice $E(I, I')$, sous-matrice de $\overline{A}(I, C)$, est une matrice de base pour cette dernière matrice. La nouvelle matrice $\overline{A}(I', C)$ s'obtient donc au moyen du produit :

$$\overline{A}(I', C) = E(I, I')^{-1} \times \overline{A}(I, C).$$

C'est aussi :

$$B(I', L) \times \overline{A}(L, C).$$

La nouvelle inverse de base est donc :

$$B(I', L) = E(I, I')^{-1} \times B(I, L).$$

Nous nous proposons d'interpréter le terme indicé par (r, s) de la matrice $\overline{A}(I, C)$. Par définition de cette matrice on a :

$$\overline{A}(I, s) = B(I, L) \times A(L, s).$$

On a donc : $\overline{A}(r, s) = B(r, L) \times A(L, s)$; si l'on multiplie $B(I, L)$ par q, tous les calculs portent sur des entiers (remarque 1).

On vient de démontrer que $q \times A(r, s)$ est un entier.

Plus précisément :

$$q \times A(r, s) = \sum_{l \in L} (q \times B(r, l)) \times A(l, s).$$

Souvenons nous de l'interprétation que nous avons donnée de la matrice $q \times B(I, L)$. C'est la transposée de la matrice des cofacteurs de $A(L, I)$.

La ligne $q \times B(r, L)$ est donc la ligne des cofacteurs convenablement signés de la colonne r de la matrice $A(L, I)$, $\sum_{l \in L} (q \times B(r, l)) \times A(l, s)$ n'est donc rien d'autre que le déterminant de la matrice $A(L, I')$, déterminant développé suivant la colonne r. On vient de démontrer que :

Remarque 2 : $q \times \overline{A}(r, s) = \det(A(L, I')) = q'$.

Ce qui est vrai de la ligne r de la colonne $\overline{A}(I, s)$ est vrai de ses autres lignes, $q \times \overline{A}(I, s)$ est donc une colonne d'entiers.

La matrice $E(I, I')^{-1}$ est une matrice indicée par (I', I). Notons-la $\overline{E}'(I', I)$. Ses colonnes indicées par $I\backslash\{r\}$ sont celles de la matrice identité $\overline{A}(I', I')$.

Les éléments de cette inverse se déduisent, de façon classique de ceux de $E(I, I')$. La colonne r est la seule colonne qui ne soit pas une colonne de matrice identité.

Pour $i = s$, on a $\overline{E}'(i, r) = 1/E(r, s)$, et pour $i \neq s$ on a $\overline{E}'(i, r) = -E(i, s)/E(r, s)$.

Bien entendu (tout se passe ici comme pour $B\ (I, L)$), $q' \times \overline{E}(I', I)$ est une matrice entière. Le seul terme en dénominateur dans l'expression de $\overline{E}'(I', I)$ est $E(r, s) = q'$. Ce terme q' est le déterminant de $E(I, I')$, c'est le seul terme différent de 1 de l'expression de celui-ci.

On a donc :

$$q' \times \overline{E}(I', I) \times q \times B\ (I, L) = q' \times q \times B(I', L).$$

C'est une matrice entière, mais $q' \times B(I', L)$ est, déjà, une matrice entière, et donc $q' \times \overline{E}(I', I) \times q \times B\ (I, L)$ a tous ses éléments divisibles par q.

On a donc un algorithme polynomial pour calculer $q' \times B(I', L)$ à partir de $q \times B\ (I, L)$:

1. Multiplier $q \times B\ (I, L)$ à gauche par $q' \times \overline{E}(I', I)$,
2. Diviser chacun des éléments par q.

On effectue alors $2 \times |L|^2$ multiplications et $|L|^2$ divisions par q.

3. CALCULS LORSQUE LES ÉLÉMENTS DE $A(L, C)$ SONT DES FRACTIONS

Il n'est pas nécessaire ici de multiplier les lignes par le produit des dénominateurs de leurs éléments. Il suffit de les multiplier par celui des dénominateurs des éléments de $I \cup I'$. Cela revient à mettre à jour un dénominateur par ligne. Avant chaque pivotage il suffira de multiplier $B(I, L)$ par le produit des dénominateurs de la colonne $A(L, s)$. Après chaque pivotage on divisera par le produit des dénominateurs de la colonne $A(L, r)$.

4. Q-MATRICES ET FORME PRODUIT DE L'INVERSE

Au lieu de stocker l'inverse de base $B\ (I, L)$ on peut stocker les différentes $\overline{E}(I', I)$. On appelle cette façon de préparer les calculs à droite et à gauche avec $A(L, I)$ la *forme produit de l'inverse* (PFI pour Product Form of the

Inverse). Appelons I_j le j-ième ensemble de colonnes rencontré. On a : $I_j = (I_{j-1} \backslash \{r_{j-1}\}) \cup \{s_{j-1}\}$.

Lorsque l'on initialise le procédé avec une matrice identité $A(L, L)$ (*voir* section 5), les produits de $A(L, L)$ et des k premières $\overline{E}(I_j, I_{j-1})$, n'est rien d'autre que l'inverse $B(I_j, L)$ de $A(L, I_j)$.

Appelons q_j le déterminant de $A(L, I_j)$. D'après la remarque 2, $q_j \times B(I_j, L)$ est une matrice à coefficients entiers, et le produit par la colonne $A(L, s)$ est une colonne de déterminants. On vient de montrer que l'on peut calculer de proche en proche la nouvelle colonne $\overline{A}(I, s)$. Il suffit de stocker les différentes matrices $\overline{E}(I_j, I_{j-1})$ et les déterminants q_j.

5. INITIALISATION

On a convenu d'adjoindre une matrice Identité $U(L, L)$ à la matrice $A(L, C)$, ses colonnes deviennent alors $C \cup L$. La matrice $q \times B$ initiale est alors $A(L, L)$ et la valeur initiale de q est 1.

6. CONCLUSION

Lorsque l'on programme ces opérations, on remarque que les expressions habituellement employées sont très peu modifiées. L'exemple qui suit concerne la mise à jour de l'inverse $B(I, L)$, *abars* étant, dans le premier texte, la colonne $\overline{A}(I, s)$, et la colonne $q' \times \overline{A}(I, s)$ dans le second.

Un *pivotage* s'écrit, en Pascal, et en flottants, de la façon suivante :

```
aa :=1 / abars[r];
for i := 1 to m do
  B[r,i] := B[r,i] * aa;
for i := 1 to m do
  if i <> r then
    for j :=1 to m do
      B[i,j] := B[i,j] - B[r,j] * abars[i];
```

En entiers longs, ce texte est remplacé par :

```
qq := abars[r];
for i := 1 to m do
  if i <> r then
    for j :=1 to m do
      B[i,j] := (qq * B[i,j] - B[r,j] * abars[i]) / q;
```

Dans ce second texte de programme, qq est le nouveau déterminant q'. Les calculs sont même légèrement simplifiés : la première boucle a été supprimée.

BIBLIOGRAPHIE

1. A. Colmerauer, *An introduction to Prolog III,* Communications of the ACM, July 1990, *33*, n° 7.
2. J. R. Edmonds, *System of distinct representatives and linear algebra,* Journal of the National Bureau of Standards B 71, 1967.
3. J. R. Edmonds, *Exact Pivoting,* For ECCO VII, February 21, 1994.
4. J. F. Maurras, *Cours de Complexité des algorithmes,* Luminy, July 1990, *33*, n° 7.