MACULAN

RBC, Rio de Janeiro, V. 1, nº 1, 1981, pp. 19 a 39

ASPECTOS COMPUTACIONAIS EM PROGRAMAÇÃO LINEAR*

LUIS PAULO VIEIRA BRAGA
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO/COPPE
CAIXA POSTAL 68511 – CEP 21941 – RIO DE JANEIRO – RJ – BRASIL

NELSON MACULAN FILHO
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO/COPPE E INSTITUTO DE MATEMÁTICA
CAIXA POSTAL 68511 – CEP 21941 – RIO DE JANEIRO – RJ – BRASIL

SUMÁRIO

A implantação em computador do algoritmo do simplex, devido a Dantzig, para resolver problemas de médio e grande porte de programação linear, exige o desenvolvimento de técnicas computacionais ligadas à esparsidade de matrizes, o estudo de erros de arredondamento, que podem mascarar sèriamente os resultados finais, e exige finalmente, um bom conhecimento de análise numérica. Apresentamos aqui alguns métodos de inversão de matrizes utilizados na construção de códigos computacionais eficientes de programação linear. Com intuito de expor nossa pequena experiência com o uso e o desenvolvimento de códigos computacionais de programação linear, concebemos esse trabalho.

ABSTRACT

An implementation of Dantzig's Simplex Algorithm oriented to solve medium and large scale linear programming problems must consider sparse matrix techiques, rounding errors and finally numerical analysis. Some methods which are employed in efficient linear programming codes are presented. The objective is to discuss some aspects about the use and development of linear programming software.

*Apresentado na Quinta Escuela Latinoamericana de Matemática, Mar del Plata, Argentina, 28 de julho a 8 de agosto, 1980.

1. INTRODUÇÃO

Problema de Programação Linear. Um problema de programação linear pode ser semprecolocado sob a seguinte forma:

$$\text{maximizar } z = cx \,, \tag{1.1}$$

$$\text{sujeito a } \quad Ax = b \tag{1.2}$$

$$x \geq 0 \,, \tag{1.3}$$

onde são dados $c^T \in \mathbb{R}^n$, A matriz com m linhas e n colunas de posto m, $b \in \mathbb{R}$. Deseja-se encontrar $x = (x_1\ x_2\ \ldots\ x_n)^T \in \mathbb{R}^n$ que maximize a função objetivo (1.1), satisfazendo (1.2) e (1.3).

Particionemos a matriz A em duas matrizes B e N, denominadas respectivamente matrizes básica e não básica, onde B é quadrada de ordem m e inversível, isto é,

$$A = [a_1\ a_2\ \ldots\ a_n] = [B\ N] \tag{1.4}$$

Analogamente particionaremos x em x_B e x_N e c em c_B e c_N. Sejam I_B e I_N os conjuntos dos índices das variáveis básicas e não básicas respectivamente.

Podemos assim, escrever (1.2):

$$Bx_B + Nx_N = b \tag{1.5}$$

Resolvendo (1.5), isto é, tiraremos os valores de x_B em função de x_N:

$$x_B = B^{-1}b - B^{-1}Nx_N \tag{1.6}$$

As componentes de x_B são denominadas variáveis básicas, pois estão associadas às colunas de B que formam uma base do $\mathbb{R}^n$ e as de x_N de não básicas.

Quando fizermos $x_N = 0$, x_B tomará o valor

$$\bar{x}_B = B^{-1}b \tag{1.7}$$

denominaremos essa solução básica de (1.2) ou (1.5).

Se uma das componentes de $\bar{x}_B$ for nula, diremos que a solução básica é degenerada. Da mesma maneira se $\bar{x}_B \geq 0$, denominaremos que $\bar{x}_B$ é uma solução básica viável, para o problema (1.1), (1.2) e (1.3), pois além de satisfazer (1.2) satisfaz também (1.3).

Pela teoria do método do simplex, ver |6|, |12| e |10|, caso (1.1), (1.2) e (1.3) admita uma solução finita, pelo menos uma solução ótima será uma solução básica.

Verificaremos também que o número máximo de matrizes quadradas diferentes entre elas duas a duas de ordem m, inversíveis, formadas de m colunas dentre as n de A tem um limite superior dado por $\binom{n}{m}$. Logo o número de soluções básicas é finito.

O método do simplex parte de uma solução básica viável, testa se essa solução é ótima, caso seja para, caso contrário passa para outra solução básica viável que difere apenas de uma componente em relação à solução anterior, como será visto mais adiante.

Reescrevendo a função objetivo:

$z = c_B x_B + c_N x_N$, substituindo x_B em função de x_N, como indicado em (1.5), teremos:

$$z = c_B B^{-1} b - \sum_{j \in I_N} (z_j - c_j) x_j \qquad (1.8)$$

onde:

$$z_j = c_B B^{-1} a_j \qquad (1.9)$$

O critério de otimalidade é dado por:

$$z_j - c_j \geq 0 \quad \forall\, j \in I_N \qquad (1.10)$$

Caso (1.10) seja verificado, $x_B = B^{-1} b$ e $x_N = 0$ otimizam o problema em questão fornecendo a z o valor $\bar{z} = c_B B^{-1} b$.

Quando (1.10) não for verificado, isto é, existir ao menos um $k \in I_N$, tal que $z_k - c_k < 0$, não poderemos garantir que a solução em questão será ótima, pois se x_k entrar na base, sempre tomará valores não negativos para manter a viabilidade, poderá fazer com que z aumente de valor, isto é, $z = \bar{z} - (z_k - c_k)\theta$ onde θ é o valor não negativo atribuido a x_k, quando esta variável entrar na base.

Como podemos observar x_k deverá entrar no lugar de uma variável básica, isso será feito pela condição de viabilidade

$$\frac{x_{B_s}}{\bar{a}_{sk}} = \underset{\bar{a}_{ik} > 0}{\text{mínimo}} \quad \frac{x_{B_i}}{\bar{a}_{ik}} \qquad (1.11)$$

$$\forall\, i \in I_B$$

onde $\bar{x}_{B_i}$ é a i-ésima componente de $\bar{x}_B$ e $\bar{a}_{ik}$ é a i-ésima componente de $\bar{a}_k^i = B^{-1} a_k$. A expressão (1.11) indica que x_s sairá da base, devemos lembrar que $\bar{x}_{B_s}$ é o valor que x_s tomou. Quando não houver nenhum $\bar{a}_{ik} > 0$, o problema (1.1), (1.2) e (1.3) tem solução ilimitada. Ver |6|, |12| e |10|.

Lembremos também que quando (1.2) e (1.3) formarem um conjunto vazio então não teremos nenhuma solução viável para o problema, logo não teremos também solução básica viável. O próprio método do simplex, denominado das duas fases, tratará esse problema. Ver |6|, |12| e |10|.

2. MÉTODOS DE INVERSÃO DE MATRIZES BÁSICAS

Considerações Gerais. Como foi visto no parágrafo anterior x_k entrará no lugar de x_s, isso quer dizer que a coluna a_k substitui a_s na base.

Suponhamos que

$B = [a_1\ a_2 \cdots a_{s-1}\ a_s\ a_{s+1} \cdots a_m]$, a nova base será

$\bar{B} = [a_1\ a_2 \cdots a_{s-1}\ a_k\ a_{s+1} \cdots a_m]$. Sabemos que B^{-1} é conhecida, queremos conhecer $(\bar{B})^{-1}$. Poderemos ou inverter diretamente $\bar{B}$, às vezes, necessário quando há uma acumulação de erros de arredondamento maior que um limite tolerado, ou calcular $(\bar{B})^{-1}$ a partir de B^{-1}, por uma operação matricial bem simples.

A seguir apresentaremos alguns tipos de matrizes, seus usos e sua propriedades.

Todas as matrizes aqui tratadas serão quadradas de ordem m, quando isso não ocorrer assinalaremos.

Seja $e_j = (0\ 0 \ldots 0 1 0 \ldots 0)^T$, cuja única componente diferente de zero e igual a um é a j-ésima, o vetor unitário na direção j de $\mathbb{R}^m$.

(i) Matriz com apenas uma coluna diferente de zero.

$$\begin{bmatrix} 0 & 0 & \cdots & 0 & t_{1k} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & t_{2k} & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & & \\ 0 & 0 & & 0 & t_{mk} & 0 & \cdots & 0 \end{bmatrix} = t_k e_k^T \quad , \text{ onde}$$

$$t_k = (t_{1k}\, t_{2k} \cdots t_{mk})^T$$

k-ésima coluna

(ii) Matriz com apenas uma linha diferente de zero:

$$\begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \\ t_{k1} & t_{k2} & \cdots & t_{km} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} = e_k t_k, \text{ onde } t_k = (t_{k1} t_{k2} \cdots t_{km}).$$

k-ésima linha

Utilizaremos a notação t_k para indicar um vetor coluna ou vetor linha.

(iii) Matriz quadrada elementar coluna:

$$\begin{bmatrix} 1 & 0 & \dots & 0 & t_{1k} & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & t_{2k} & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ 0 & 0 & \dots & 1 & t_{k-1k} & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & t_{kk} & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & t_{k+1k} & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 & t_{mk} & 0 & & 1 \end{bmatrix} = E_k = I + (t_k - e_k)e_k^T .$$

onde $t_k = (t_{1k}\ t_{2k} \dots t_{mk})^T$. Verificaremos facilmente que $\det(E_k) = t_{kk}$. Se $t_{kk} \neq 0$ então podemos ter E_k^{-1} :

$$\begin{bmatrix} 1 & 0 & \dots & -\frac{t_{1k}}{t_{kk}} & \dots & 0 \\ 0 & 1 & \dots & -\frac{t_{2k}}{t_{kk}} & \dots & 0 \\ \vdots & \vdots & & \vdots & & \vdots \\ 0 & 0 & \dots & \frac{1}{t_{kk}} & & 0 \\ \vdots & \vdots & & \vdots & & \\ 0 & 0 & & -\frac{t_{mk}}{t_{kk}} & & 1 \end{bmatrix} = I - \frac{1}{t_{kk}} (t_k - e_k)e_k^T$$

É interessante notar que a inversa de uma matriz quadrada elementar coluna, também é matriz quadrada elementar coluna e o grau de esparsidade de ambas é idêntico. Lembremos que:

$$E_k = I + (t_k - e_k)e_k^T \qquad (2.1)$$

$$E_k^{-1} = I - \frac{1}{t_{kk}} (t_k - e_k)e_k^T \qquad (2.2)$$

Para que possamos recompor E_k, basta-nos o conhecimento de t_k

e sua posição (k-ésima coluna). O mesmo deve ser estendido a E_k^{-1}.

Quando $t_k = (0\ 0\ \ldots\ 1\ t_{k+1\,k}\ t_{k+2\,k}\ \ldots\ t_{mk})^T$, isto é as k-1 primeiras componentes são nulas e $t_{kk}=1$. Nesse caso a matriz quadrada elementar coluna será triangular inferior. Temos que:

$$E_k = I + (t_k - e_k)e_k^T = I + t_k' e_k^T \text{ , onde } t_k' = t_k - e_k =$$

$(0\ 0\ \ldots\ 0\ t_{k+1\,k}\ t_{k+2\,k}\ \ldots\ t_{mk})^T$. E_k sempre terá inversa, pois $t_{kk} = 1$. $E_k^{-1} = I - t_k' e_k^T$.

(iv) Matriz quadrada elementar linha:

$$\begin{bmatrix} 1 & 0 & \ldots & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 & \ldots & 0 \\ \vdots & \vdots & & \vdots & & \vdots \\ t_{k1} & t_{k2} & & t_{kk} & & t_{km} \\ \vdots & \vdots & & \vdots & & \vdots \\ 0 & 0 & & 0 & & 1 \end{bmatrix} = E_k = I + e_k(t_k - e_k^T) \text{ ,}$$

onde $t_k = (t_{k1}\ t_{k2}\ \ldots\ t_{km})$. Da mesma maneira de $t(E_k) = t_{kk}$. Se $t_{kk} \neq 0$ então E_k^{-1} será:

$$\begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ -\frac{t_{k1}}{t_{kk}} & -\frac{t_{k2}}{t_{kk}} & \ldots\ \frac{1}{t_{kk}}\ \ldots & & -\frac{t_{km}}{t_{kk}} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \ldots\ 0 & & 1 \end{bmatrix} = E_k^{-1} = I - \frac{1}{t_{kk}} e_k(t_k - e_k^T)$$

$$E_k = I + e_k(t_k - e_k^T) \qquad (2.3)$$

$$E_k^{-1} = I - \frac{1}{t_{kk}} e_k(t_k - e_k^T) \qquad (2.4)$$

Quando $t_k = (0\ 0\ \ldots\ 1\ t_{k\,k+1}\ \ldots\ t_{km})$ então teremos de (2.3) $E_k = I + e_k(t_k - e_k^T)$, se $t_k' = (t_k - e_k^T) = (0\ 0\ \ldots\ 0\ t_{k\,k+1}\ \ldots\ t_{km})$ então $E_k = I + e_k t_k'$ e $E_k^{-1} = I - e_k t_k'$.

As matrizes descritas em (iii) e (iv) são responsáveis pelos

métodos de inversão que apresentaremos brevemente.

(v) Matriz triangular inferior:

$$\begin{bmatrix} \ell_{11} & 0 & \dots & 0 \\ \ell_{21} & \ell_{22} & \dots & 0 \\ \vdots & \vdots & & \vdots \\ \ell_{m1} & \ell_{m2} & & \ell_{mm} \end{bmatrix} = \begin{bmatrix} \ell_{11} & 0 & \dots & 0 \\ \ell_{21} & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ \ell_{m1} & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & \ell_{22} & & 0 \\ \vdots & \vdots & & \vdots \\ 0 & \ell_{m2} & \dots & 1 \end{bmatrix} \dots \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & & \ell_{mm} \end{bmatrix}$$

$$L = L_1 L_2 \dots L_m$$

Ou ainda $L_k = I + (\ell_k - e_k)e_k^T \quad k = 1,2,\dots,m$ e onde

$\ell_k = (0 \dots 0 \ \ell_{kk} \dots \ell_{mk})^T$. $L = L_1 L_2 \dots L_{m-1} L_m$

Exemplo:

$$L = \begin{bmatrix} 3 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 2 & 4 \end{bmatrix} = \begin{bmatrix} 3 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

Caso $\ell_{jj} \neq 0 \quad j = 1,2,\dots,m$ então $\det(L) \neq 0$ e

$$L^{-1} = L_m^{-1} L_{m-1}^{-1} \dots L_1^{-1}$$

Assim sendo no exemplo acima:

$$L^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 4 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 3 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}^{-1} =$$

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{4} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{3} & 0 & 0 \\ -\frac{2}{3} & 1 & 0 \\ -\frac{1}{3} & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \frac{1}{3} & 0 & 0 \\ -\frac{2}{3} & 1 & 0 \\ \frac{1}{4} & -\frac{1}{2} & \frac{1}{4} \end{bmatrix}$$

A título de verificação:

$$\begin{bmatrix} 3 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 2 & 4 \end{bmatrix} \begin{bmatrix} \frac{1}{3} & 0 & 0 \\ -\frac{2}{3} & 1 & 0 \\ \frac{1}{4} & -\frac{1}{2} & \frac{1}{4} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(vi) Matriz triangular superior:

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1m} \\ 0 & u_{22} & & u_{2m} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & & u_{mm} \end{bmatrix} = \begin{bmatrix} 1 & 0 & & u_{1m} \\ 0 & 1 & & u_{2m} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & & u_{mm} \end{bmatrix} \cdots \begin{bmatrix} 1 & u_{12} & \cdots & u \\ 0 & u_{22} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

$$U = U_m \cdots U_2 U_1$$

Ou ainda $U_k = I + (u_k - e_k)\, e_k^T$, onde

$$u_k = (u_{1k}\ u_{2k}\ \cdots\ u_{kk}\ 0\ \cdots\ 0)^T,\quad k = 1,2,\ldots,m.$$

$$U = U_m U_{m-1} \cdots U_2 U_1$$

Exemplo:

$$U = \begin{bmatrix} 2 & 1 & 2 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Quando $u_{jj} \neq 0$ $j = 1,2,\ldots,m$ então $\det(U) \neq 0$ e

$$U^{-1} = U_1^{-1} U_2^{-1} \cdots U_{m-1}^{-1} U_m^{-1}$$

U^{-1} no exemplo acima será:

$$U^{-1} = \begin{bmatrix} 2 & 1 & 2 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{bmatrix}^{-1} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{bmatrix}^{-1} =$$

$$\begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} & -\frac{1}{4} \\ 0 & 1 & -\frac{1}{2} \\ 0 & 0 & \frac{1}{2} \end{bmatrix}$$

Como verificação

$$\begin{bmatrix} 2 & 1 & 2 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} & -\frac{1}{4} \\ 0 & 1 & -\frac{1}{2} \\ 0 & 0 & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(vii) Matrizes de permutação:

Seja a matriz A, queremos encontrar uma matriz A' quando permutamos as colunas a_k e a_s.

$$A = \left[\begin{array}{c|c|c|c|c} \cdots & a_k & \cdots & a_s & \cdots \end{array}\right] \qquad A' = \left[\begin{array}{c|c|c|c|c} \cdots & a_s & \cdots & a_k & \cdots \end{array}\right]$$

$A' = AP$ onde $P = |e_1 e_2 \cdots e_{k-1} e_s e_{k+1} \cdots e_{s-1} e_k e_{s+1} \cdots e_m|$

Desejamos permutar linhas:

$$C = \left[\begin{array}{c} \vdots \\ \hline a_k \\ \hline \vdots \\ \hline a_s \\ \hline \vdots \end{array}\right] \qquad C' = \left[\begin{array}{c} \vdots \\ \hline a_s \\ \hline \vdots \\ \hline a_k \\ \hline \vdots \end{array}\right]$$

$C' = PC$ onde $P = |e_1 e_2 \cdots e_{k-1} e_s e_{k+1} \cdots e_{s-1} e_k e_{s+1} \cdots e_m|$

Inversão de Matrizes pela Forma do Produto da Inversa (PFI).

Desenvolveremos aqui um método para inverter matrizes denominado de forma do produto da inversa, ver também |8| e |13|.

Seja $B_q = (a_1\ a_2\ \cdots\ a_{k-1}\ q\ a_{k+1}\ \cdots\ a_m)$ matriz quadrada mxm, inversível cuja inversa B_q^{-1} é conhecida. Desejamos inverter $B_r = (a_1 a_2 \cdots a_{k-1}\ r\ a_{k+1}\ \cdots\ a_m)$, B_r possue as mesmas colunas de B_q só que no lugar da coluna q está r. Tomemos $B_q^{-1} r = y = (y_1 y_2 \cdots y_k \cdots y_m)^T$.

Seja a matriz elementar $E_k = I + (y-e_k)e_k^T$, se $y_k \neq 0$ então E_k é inversível, isto é, $E_k^{-1} = I - \frac{1}{y_k}(y-e_k)e_k^T$.

Facilmente podemos verificar que $B_r^{-1} = E_k^{-1} B_q^{-1}$, quando E_k for inversível, isto é, quando $y_k \neq 0$.

Essas considerações são importantes pois quando utilizamos o método do simplex temos a cada iteração inverter uma matriz que difere de uma matriz, cuja inversa é conhecida, apenas por uma coluna.

Podemos todavia utilizar essa técnica para inverter uma matriz. Para explicar o método da forma do produto da inversa faremos esquematicamente sob forma de passos de um algorítmo.

Seja $B = (a_1\ a_2\ a_3\ a_4)$ matriz 4x4 que desejamos inverter.

Tomemos a matriz unitária, de ordem 4, I que sabemos inverter: $I = I^{-1}$.

i) $E_1 = I + (a_1 - e_1)e_1^T$, supondo que E_1 seja inversível, sabemos calcular E_1^{-1}, pois E_1 é uma matriz elementar.

ii) $a_2^1 = E_1^{-1} a_2$, $E_2 = I + (a_2^1 - e_2)e_2^T$, sendo E_2 inversível calculemos E_2^{-1}.

iii) $a_3^1 = E_2^{-1} E_1^{-1} a_3$, $E_3 = I + (a_3^1 - e_3)e_3^T$, sendo E_3 inversível calculemos E_3^{-1}.

iv) $a_4^1 = E_3^{-1} E_2^{-1} E_1^{-1} a_4$, $E_4 = I + (a_4^1 - e_4)e_4^T$, sendo E_4 inversível, calculemos E_4^{-1}.

Finalmente $B^{-1} = E_4^{-1} E_3^{-1} E_2^{-1} E_1^{-1}$.

Isso será sempre possível quando B for inversível. Às vezes E_1, tal como foi apresentado acima não é inversível, pois a primeira componente a_1 pode ser nula, quando isso acontecer basta permutarmos as colunas de B.

Queremos encontrar B^{-1}, no entanto às vezes precisamos realizar antes do algorítmo de inversão operações de permutação entre as colunas de B. Seja então: $C = BP$, $C^{-1} = P^{-1}B^{-1}$, $B^{-1} = PC^{-1}$.

A título de esclarecimento seja:

$B = (a_1\ a_2\ a_3\ a_4)$, tomemos $C = (a_2\ a_3\ a_1\ a_4)$, cuja inversa

$$C^{-1} = \begin{bmatrix} b_2 \\ b_3 \\ b_1 \\ b_4 \end{bmatrix}, \text{ logo } B^{-1} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}, \text{ onde } b_i^T \in \mathbb{R}^4 .$$

A idéia de deixarmos a inversa sob a forma do produto de inversa está ligada ao problema de esparsidade da matriz considerada, isto é, desejamos manter a esparsidade da matriz inversa, pelo menos de maneira implícita.

Quando implementamos um código de programação linear em computador, estocamos de maneira sequencial em memória auxiliar (disco ou fita magnéticos) as matrizes E_k^{-1}, $k=1,2,\ldots,m$ sem explicitar seu produto.

Ao resolvermos um problema de programação linear em computador pelo método do simplex sob forma revisada, temos que calcular vetores coluna do tipo $v = s\ B^{-1}$ ou seja:

$$v = (\ldots((s\ E_m^{-1})E_{m-1}^{-1})\ldots\ E_2^{-1})E_1^{-1}$$

Procederemos da seguinte maneira: leremos E_m^{-1} em memória auxiliar trazendo-a para uma área de trabalho em memória central e executaremos o produto $s\ E_m^{-1} = v_1$, depois leremos E_{m-1}^{-1} e faremos $v_1 E_{m-1}^{-1} = v_2$, etc, até termos lido E_1^{-1} e executado a operação $v_m E_1^{-1} = v$.

Consideremos agora o seguinte produto:

$$(x_1\ x_2\ldots x_k\ldots x_m)\underbrace{\begin{bmatrix} 1 & 0 & \ldots & g_1 & \ldots & 0 \\ 0 & 1 & \ldots & g_2 & \ldots & 0 \\ \vdots & \vdots & & \vdots & & \vdots \\ 0 & 0 & & g_k & & 0 \\ \vdots & \vdots & & \vdots & & \vdots \\ 0 & 0 & & g_m & & 1 \end{bmatrix}}_{E_k} = (x_1\ x_2\ldots x_{k-1}\ \sum_{i=1}^{m} x_i g_i\ x_{k+1}\ldots x_m)$$

somente a k-ésima componente de x foi modificada, tomando o valor $x^T g = \sum_{i=1}^{m} x_i g_i$.

Para calcularmos $\bar{b} = B^{-1}b$ procederemos da seguinte maneira: $\bar{b} = E_m^{-1}(\ldots(E_2^{-1}(E_1^{-1} b))\ldots)$. Como já foi descrito logo acima traremos para a área de trabalho em memória central E_1^{-1}, e calcularemos $b_1 = E_1^{-1}b$, depois E_2^{-1} e calcularemos $b_2 = E_2^{-1}b_1$ e assim por diante.

Chamaremos atenção para a seguinte operação matricial:

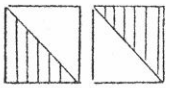
$$\begin{bmatrix} 1 & 0 & \ldots & g_1 & \ldots & 0 \\ 0 & 1 & \ldots & g_2 & \ldots & 0 \\ \vdots & \vdots & & \vdots & & \vdots \\ 0 & 0 & \ldots & g_k & \ldots & 0 \\ \vdots & \vdots & & \vdots & & \vdots \\ 0 & 0 & \ldots & g_m & \ldots & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \\ \vdots \\ x_m \end{bmatrix} \quad \begin{bmatrix} x_1 + g_1 x_k \\ x_2 + g_2 x_k \\ \vdots \\ x_{k-1} + g_{k-1} x_k \\ g_k x_k \\ x_{k+1} + g_{k+1} x_k \\ \vdots \\ x_m + g_m x_k \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{k-1} \\ 0 \\ x_{k+1} \\ \vdots \\ x_m \end{bmatrix} + x_k \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_{k-1} \\ g_k \\ g_{k+1} \\ \vdots \\ g_m \end{bmatrix}$$

A pré-multiplicação de B^{-1} por um vetor linha ou sua pós-multiplicação por um vetor coluna são as operações mais realizadas em um código de programação linear, além dos cálculos de inversas.

Para realizarmos todas essas operações aqui descritas basta armazenarmos apenas a coluna diferente das do tipo e_j das matrizes elementares E_k e dizer sua posição. Na realidade apenas os elementos diferentes de zero serão armazenados.

Inversão de matrizes sob Forma LU. Outra forma interessante de invertermos uma matriz, quando tratamos de inversas de matrizes básicas do algorítmo do simplex é quando essas matrizes se apresentam sob forma LU, ver |1| , |2| , |5| e |6|.

Então $B =$  e $B^{-1} = U^{-1} L^{-1}$

L U

Exemplo de decomposição LU, utilizando operação elementar.

Seja

$$B = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 2 & 0 \\ 1 & 4 & 2 \end{bmatrix}$$, queremos calcular L e U tais que B=LU ou

ainda no nosso caso supondo B inversível. Logo L também o será: $L^{-1}B = U$. Sabemos que L^{-1} também será triangular inferior.

1º Passo

$$\underset{L_1^{-1}}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}} \quad \underset{B}{\begin{bmatrix} 1 & 2 & 1 \\ 0 & 2 & 0 \\ 1 & 4 & 2 \end{bmatrix}} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 2 & 0 \\ 0 & 2 & 1 \end{bmatrix}$$

2º Passo

$$\underset{L_2^{-1}}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & -1 & 1 \end{bmatrix}} \quad \underset{L_1^{-1} B}{\begin{bmatrix} 1 & 2 & 1 \\ 0 & 2 & 0 \\ 0 & 2 & 1 \end{bmatrix}} = \underset{U}{\begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}$$

3º Passo

$$L^{-1} = L_2^{-1} \; L_1^{-1} \qquad L = L_1 \, L_2$$

$$L_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \qquad L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 2 & 1 \end{bmatrix}$$ diretamente

de L_1^{-1} e L_2^{-1} , pois são matrizes elementares. Então:

$$L = L_1 \, L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

4º Passo

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 2 & 0 \\ 1 & 4 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 1 & 2 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$B = L \qquad U$$

Obs.: Temos também

$$L^{-1} = L_2^{-1} L_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ -1 & -1 & 1 \end{bmatrix}$$

O exemplo aqui realizado utiliza a forma de eliminação, utilizando operações com matrizes elementares do tipo E.F.I. |11|. Observamos também que a menos de permutações toda matriz tem uma decomposição LU.

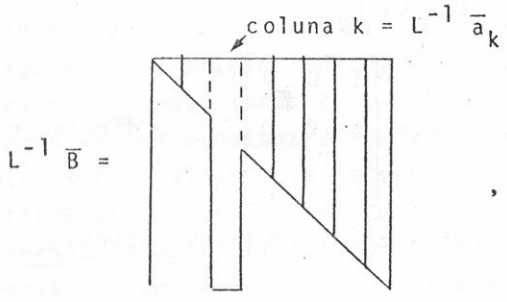
Método de Bartels e Golub |1|. Suponhamos agora que já tenhamos $B = LU$ e sua inversa, isto é $B^{-1} = U^{-1} L^{-1}$, ou ainda escrita sob forma de matrizes elementares:

$B^{-1} = U_1^{-1} U_2^{-1} \ldots U_m^{-1} L_m^{-1} L_{m-1}^{-1} \ldots L_1^{-1}$ como já foi visto anteriormente.

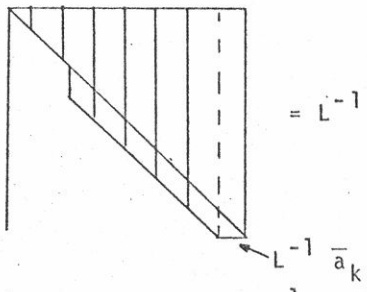
Gostaríamos de calcular a inversa de $\bar{B}$ que difere de B somente na coluna k, isto é, no lugar de a_k, colocaremos $\bar{a}_k$. Operação de pivoteamento no algorítmo do simplex.

Se $B = LU$, $L^{-1} B = U$ então

$L^{-1} \bar{B}$ diferirá de U apenas na coluna k, isto é:



, aplicando uma permutação a $L^{-1} \bar{B}$ de maneira que k ocupe a última coluna, isto é, vá para a coluna m, e k+1, passe para k, assim por diante até que m passe para m-1. Denominemos de P essa matriz de permutação, então:



Por operações elementares, tornaremos $L^{-1} \bar{B} P$ triangular superior, como já foi visto no último exemplo.

$$Q_{m-1}^{-1} \ldots Q_k^{-1} L^{-1} \bar{B} P = \quad = \bar{U}$$

$$(\bar{L})^{-1} = Q_{m-1}^{-1} \ldots Q_k^{-1} L^{-1}$$

$$\bar{B}P = LQ_k \ldots Q_{m-1} \bar{U}$$

logo $P^{-1}(\bar{B})^{-1} = (\bar{U})^{-1} Q_{m-1}^{-1} \ldots Q_k^{-1} L^{-1}$, ou ainda

$(\bar{B})^{-1} = P(\bar{U})^{-1} (\bar{L})^{-1}$, foi mantida a inversa de $\bar{B}$ sob a forma LU, acrescentando-se a matriz de permutação P que pode ser guardada implicitamente.

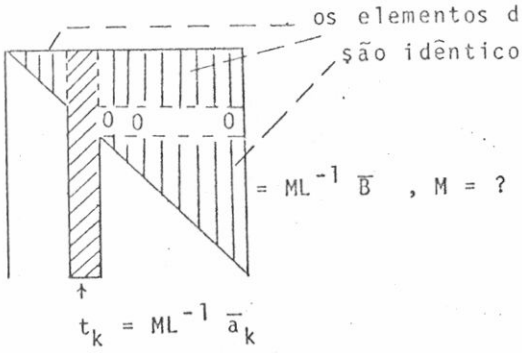
Métodos Pseudo LU. Veremos agora uma outra técnica apresentada por Brayton, Gustavson e Willoughby |5|. Nesse método, que denominaremos de pseudo-LU, a matriz L será formada pelo produto de matrizes elementares, linhas e colunas, não mais apenas de matrizes elementares colunas como foi mostrado no método anterior.

Suponhamos B = LU ou de maneira equivalente
$B^{-1} = U_1^{-1} U_2^{-1} \ldots U_m^{-1} L_m^{-1} L_{m-1}^{-1} \ldots L_1^{-1}$. Gostaríamos agora de calcular a inversa de $\bar{B}$, quando $\bar{B}$ difere de B apenas por uma coluna, no local de a_k em B coloca-se $\bar{a}_k$.

O método faz o seguinte:
seja $v_k = L^{-1} \bar{a}_k$, como no método anterior $L^{-1} \bar{B}$ é idêntico a U exceto no que se refere à coluna k que agora será v_k.

Para reduzir $L^{-1} \bar{B}$ a uma matriz triangular superior operare - mos como ilustramos a seguir:

RBC, Rio de Janeiro, V. 1, nº 1, 1981



, isto é, devemos zerar os elementos da linha k de k+1 a m. Seja o vetor
$w_k = (0 \ldots 0\ w_{k,k+1} \cdots w_{k,m}) = (0 \ldots 0\ u_{k,k+1} \cdots u_{k,m})\ U^{-1}$.
Formemos a matriz elementar $W_k = I + e_k w_k$, cuja inversa $W_k^{-1} = I - e_k w_k$.

Verificamos que $M = W_k^{-1}$.

Então a matriz $L^{-1}\ \bar{B}$ é modificada para $W_k^{-1}\ L^{-1}\ \bar{B}$ ficando na forma desejada. Interessa-nos agora realizar outra transformação nesta matriz para que fique com a seguinte forma:

$$\bar{U} = \begin{bmatrix} 0 & & & \\ 0 & & & \\ \vdots & & & \\ 1 & 0 & \cdots & 0 \\ \vdots & & & \\ 0 & & & \end{bmatrix} = N\ W_k^{-1}\ L^{-1}\ \bar{B}\ ,\ N = ?$$

Seja $t_k = W_k^{-1}\ L^{-1}\ \bar{a}_k$ então seja $T_k = I + (t_k - e_k)\ e_k^T$

$T_k^{-1} = 1 - \frac{1}{t_{kk}}\ (t_k - e_k)\ e_k^T$. $t_{kk} \neq 0$ pois $\bar{B}$ é inversível, caso contrário o método do simplex não deixaria $\bar{a}_k$ entrar no lugar de a_k . $N = T_k^{-1}$.

Finalmente realizamos as seguintes operações:

$$T_k^{-1}\ W_k^{-1}\ L^{-1}\ \bar{B} = \bar{U}$$

$$(\bar{B})^{-1} = (\bar{U})^{-1}\ T_k^{-1}\ W_k^{-1}\ L^{-1}$$

$(\bar{U})^{-1}$ pode ser formada facilmente da decomposição anterior de $U^{-1} = U_1^{-1}\ U_2^{-1} \ldots U_m^{-1}$.

$(\bar{U})^{-1} = U_1^{-1}\ U_2^{-1} \ldots U_{k-1}^{-1} (\bar{U}_{k+1})^{-1}\ (\bar{U}_{k+2})^{-1} \ldots (\bar{U}_m)^{-1}$, onde as ma

trizes $\bar{U}_j$ $j = k+1, \ldots, m$ são obtidas zerando a k-ésima componente de U_j.

Na memória auxiliar do computador, colocaremos no lugar de U_k, a matriz unidade (na realidade essa operação não é necessária, bastando ignorar U_k).

Teremos agora duas listas:

a primeira: $U_1^{-1} U_2^{-1} \ldots U_{k-1}^{-1} (\bar{U}_{k+1})^{-1} (\bar{U}_m)^{-1} T_k^{-1}$

a segunda : $W_k^{-1} L_m^{-1} L_{m-1}^{-1} \ldots L_1^{-1}$.

T_k é estocada na lista dos U_j e W_k na lista dos L_j. Podemos ainda indicar que:

$$T_k^{-1} W_k^{-1} L^{-1} \bar{B} = \bar{U}$$

, denominamos

$\bar{L}^{-1} \rightarrow L^{-1}$ e $\bar{U} \rightarrow U$

Acabamos de descrever nessa seção três métodos de atualização e inversão de uma base quando difere da anterior pela mudança de uma única coluna. No trabalho de Benichou et alii |2| há uma comparação estatística da utilização dos métodos descritos anteriormente. Nessa comparação o método pseudo LU fornece para todos os exemplos aí tratados uma maior esparsidade da matriz sob forma implícita e um menor tempo de computação.

Para a implantação de algorítmos que trabalham com sistemas lineares em computadores com sistemas de memória virtual, ver |7|.

Além dos aspectos computacionais de inversão de matrizes aqui tratados podemos colocar alguns problemas tais como: critério de entrada das variáveis na base, aparecimento de soluções básicas degeneradas e erros de arrendondamento. Esses pontos serão vistos a seguir.

3. CARACTERÍSTICAS ADICIONAIS DO MÉTODO DO SIMPLEX

Critério de Entrada. Como critério de entrada poderemos utilizar para entrar na base e variável não básica x_k associada ao menor $z_k - c_k$, isto é:

$$z_k - c_k = \underset{j \in I_N}{\text{mínimo}} \{z_j - c_j\} \qquad (3.1)$$

quando $z_k - c_k$ for negativo. No entanto, caso esse critério tenha sido utilizado em cada iteração, a operação de cálculo pode ser muito lenta, pois a avaliação de toods os $z_j - c_j$ para $j \in I_N$ dada

por $c_B B^{-1} a_j - c_j$, onde $c_B B^{-1} = u$ é calculado e colocado em memória central, temos ainda que realizar as operações $ua_j - c_j$ lendo todo o arquivo dos dados originais onde se encontram as colunas a_j e os coeficientes c_j. É sugerido que façamos essa leitura de todo arquivo e guardemos apenas as p colunas associadas aos, p menores $z_j - c_j$ (essa quantidade p é denominada, muitas vezes, nos códigos comerciais existentes de *"pricing"*). Resolvemos esse problema relaxado e verificamos se sua otimalidade vale para as demais soluções não básicas, caso essa afirmação seja verdadeira teremos encontrado o ótimo, caso contrário é repetida a operação da seleção dos p menores $z_j - c_j$, que na realidade, é realizada em paralelo com a verificação da otimalidade.

Um outro critério é a escolha de x_k para entrar na base associada a

$$\frac{z_k - c_k}{s_k} = \underset{j \in I_N}{\text{mínimo}} \left\{ \frac{z_j - c_j}{s_j} \right\} \qquad (3.2)$$

quando $z_j - c_j$ for negativo, onde s_j é a norma euclideana do vetor que indica a direção da aresta sobre a qual o método do simplex caminhará na iteração considerada. Na realidade s_j é calculado de maneira aproximada, ver |9|. Esse método é denominado de DEVEX.

As comparações feitas na leitura especializada entre os critérios (3.1) e (3.2) mostra, estatisticamente, que o método DEVEX (3.2) reduz o número de iterações necessárias para encontrar-se a solução ótima, assim como provê uma melhor estabilidade numérica, ver |2| e |9|.

Soluções Básicas Degeneradas. Não devemos esquecer os problemas de degeneração da solução básica, isto é, quando ao menos uma das componentes do vetor $B^{-1}b$ for nula. Nesse caso poderá acontecer o fenômeno de ciclagem, isto é, uma solução básica já obtida em iterações interiores volta a ser básica, fazendo com que o algorítmo cicle.

O método mais atual de evitar ciclagem é a regra de Bland |3|, é bem simples, no entanto exige uma certa ordem de entrada e saída da base, não permitindo, por exemplo, que se use os critérios (3.1) e (3.2).

Computacionalmente os métodos mais frequentemente utilizados são os do tipo permutação, abaixo exporemos um de simples implantação, para maiores detalhes, ver |4|.

Suporemos que em uma certa iteração do algorítmo do simplex a base seja B e que $B^{-1}b$ tenha ao menos uma componente nula. Para evitarmos que a solução básica seja degenerada, daremos um acréscimo de valor $\varepsilon > 0$ a todas suas componentes nulas.

Consideremos

$$x_B + B^{-1}Nx_N = B^{-1}b \tag{3.3}$$

adicionaremos ao segundo membro de (3.3) o vetor coluna εv, onde v é um vetor que contém o valor um nas componentes relativas às nulas de $B^{-1}b$ e zero nas demais:

$$x_B + B^{-1}Nx_N = B^{-1}b + \varepsilon v \tag{3.4}$$

Na realidade (3.4) quer dizer que o segundo membro do problema inicial não será mais b e sim $b + \varepsilon Bv$, pois

$$Bx_B + Nx_N = b + \varepsilon Bv$$

ou ainda

$$Ax = b + \varepsilon Bv \tag{3.5}$$

Mostra-se que dessa maneira o fenômeno de ciclagem será evitado, devemos, no entanto, lembrar que o problema originado foi modificado, pois encontraremos a solução do problema.

$$\text{maximizar} \quad z = cx \tag{3.6}$$

$$\text{sujeito a} \quad Ax = b + \varepsilon Bv \tag{3.7}$$

$$x \geq 0 \tag{3.8}$$

e não de 1.1), (1.2) e (1.3). Sabemos que a solução ótima para (3.5), (3.6) e (3.7) será também dual viável para (1.1), (1.2) e (1.3); testamos se será também viável, caso afirmativo a base ótima é a mesma. Caso contrário o algorítmo dual do simplex será ativado.

Foi verificado praticamente em vários exemplos no trabalho |2|, que quando ε estiver entre 10^{-2} e 10^{-3} o algorítmo dual do simplex não precisou ser ativado.

Erros de Arredondamento. Um dos problemas mais delicados na implantação de um algorítmo está ligado à instabilidade numérica provocada pelos erros de arredondamento das máquinas digitais. Como o procedimento principal visto até aqui é a inversão de uma matriz B, temos que ter em mente os erros de arredondamento acumulados em cada iteração para que possamos ter uma boa aproximação

de B^{-1}

Suponhamos que após um certo número de iterações do método do simplex tenhamos obtido uma aproximação da inversa de B, seja $\bar{B}^{-1}$. Faremos, a seguir, o produto $\bar{B}^{-1}B$ e verificamos sua distância em relação à matriz unitária I, caso seja acima de um valor dado, reinverteremos B.

As operações que realizam $\bar{B}^{-1}B$ podem ser muito onerosas em termos computacionais devido ao tamanho de B. Na prática após termos computado $c_B\bar{B}^{-1}$, tomamos a_i, uma coluna básica e executamos $c_B\bar{B}^{-1}a_i$ que fornecerá $\hat{c}_i$. Compararemos $\hat{c}_i$ com o c_i dado original da seguinte maneira:

se $|\hat{c}_i - c_i| \leq \varepsilon$ continuaremos com procedimento,
se $|\hat{c}_i - c_i| > \varepsilon$ reinverteremos B.

O valor de ε depende da precisão do computador, por exemplo, naqueles em que os números em vírgula flutuante são representados por 48 bits temos tomado ε da ordem de 10^{-6}.

4. ASPECTOS PARTICULARES

Às vezes, dado a estrutura da matriz A, é melhor em termos computacionais resolver sistemas de equações lineares simultâneos no lugar de inverter B.

Lembremos que B^{-1} é calculada para podermos especificar vetores:

$$u = c_B B^{-1}, \quad \bar{x}_B = B^{-1}b \quad \text{e} \quad \bar{a}_j = B^{-1}a_j$$

Essas três expressões poderão ser substituídas, respectivamente, por:

$$uB = c_b, \quad B\bar{x}_B = b \quad \text{e} \quad B\bar{a}_j = a_j$$

Resolvendo diretamente esses três sistemas de equações simultâneas calcularemos u, $\bar{x}_B$ e $\bar{a}_j$.

Nos casos particulares em que a solução de sistemas lineares possa ser mais eficiente do que o cálculo de inversas, é aconselhável o desenvolvimento de códigos específicos.

5. REFERÊNCIAS BIBLIOGRAFICAS

|1| BARTELS, R.H., GOLUB, G.H. (1969), "The Simplex Method of Linear Programming Using LU Decomposition", *Comm. ACM, 12.*

|2| BENICHOU, M., GAUTHIER J.M., HENTGES G., RIBIÈRE G., (1976), "The Efficient Solutiob of Large Scale Linear Programming Problems: Some Algorithmic Techiques and Computation Results", *IXth International Symposium on Mathematical Programming, Budapeste.*

|3| BLAND, R.G., (1977), "New Finite Pivoting Rules for the Simplex Method", *Mathematics of Operations Research,* vol. 2

|4| BORNSTEIN, C.T., OLIVEIRA A.A.F., DO CARMO, P.F.B., Introdução à Programação Linear" (1980), *COPPE/UFRJ, Rio de Janeiro, PDD-03/80.*

|5| BRAYTON, J.M., GUSTAVSON, F.G., WILLOUGHBY, R.A. (1969), "Some Results on Sparse Matrices", *RC-2332, IBM Research Center, Yorkton Heights, NY.*

|6| DANTZIG, G.B. (1963), "Linear Programming and Extensions" *Princeton University Press, Princeton, New Jersey.*

|7| DUBRULLE A.A., (1979), "The Design of Matrix Algorithms for Fortran and Virtual Storage", *IBM Palo Alto Scientific Center, G320-3396.*

|8| HADLEY, G.F., (1962), "Linear Programming", *Addison Wesley, Reading, Mass.*

|9| HARRIS, P.M.J. (1975), "Pivot Solutions of the Devex LP Codes", *Mathematical Programming Study 4.*

|10| MACULAN, N.F., PEREIRA, M.V.F., (1980), "Programação Linear", *Editora Atlas S.A., São Paulo (SP).*

|11| MARKOWITZ, H.M., (1957), "The Elimination Form of Inverse and its Application to Linear Programming", *Management Science, 3.*

|12| SIMONNARD, M.A., (1973), "Programming Linéarie:Technique du Calcul Économique", *Dunod, Paris, 1º volume.*

|13| TOMLIN, J.A., (1970), "Maintaining a Sparse Inverse in the Simplex Method", *Technical Report nº 70-16, Deptº of Operations Research, Stanford University, Stanford, Califórnia.*