UM ALGORITMO ESTÁVEL PARA RESOLUÇÃO DO PROBLEMA DE OTIMIZAÇÃO DE RAÇÕES

LUIZ ROBERTO DE SOUZA AMARAL

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E CIÊNCIA DA COMPUTAÇÃO

CAMPINAS - SÃO PAULO
BRASIL

UM ALGORITMO ESTÁVEL PARA RESOLUÇÃO DO PROBLEMA DE OTIMIZAÇÃO DE RAÇÕES

LUIZ ROBERTO DE SOUZA AMARAL

ESTE EXEMPLAR CORRESPONDE A REDAÇÃO FINAL DA TESE DEVIDAMENTE CORRIGIDA E DEFENDIDA PELO SR. LUIZ ROBERTO DE SOUZA AMARAL E APROVADA PELA COMISSÃO JULGADORA.

CAMPINAS, 06 DE OUTUBRO DE 1987.

ORIENTADOR: PROF. JOSÉ MÁRIO MARTÍNEZ PÉREZ

DISSERTAÇÃO APRESENTADA AO INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E CIÊNCIA DA COMPUTAÇÃO, UNICAMP, COMO REQUISITO PARCIAL PARA OBTENÇÃO DO TÍTULO DE MESTRE EM MATEMÁTICA APLICADA.

AGRADECIMENTOS

AOS MEUS PAIS, POR TODO ESFORÇO
E DEDICAÇÃO EM MINHA VIDA.

A TODOS QUE COMPARTILHARAM COM
FORÇA, INSPIRAÇÃO, ORIENTAÇÃO,
AMIZADE, COLABORAÇÃO, DEDICAÇÃO
E INCENTIVO EM CADA TRECHO DESTE
TRABALHO.

COM CARINHO

INTRODUÇÃO

Uma das primeiras aplicações de programação linear, através do método **SIMPLEX**, foi no problema de formulação de rações. Nos anos 60 as indústrias de rações já operavam programas para minimizar o custo de produção de rações. Ajustes manuais eram necessários para alocar matérias primas restritas.

Posteriormente, muitas indústrias, utilizando-se de computadores de grande porte, desenvolveram algoritmos para resolução do problema de otimização global de rações **("multiblending")**.

O surgimento dos micro-computadores de 16 bits, com grande capacidade de memória, possibilitou o desenvolvimento de softwares matemáticos para resolução de sistemas de grande porte, em especial, para o problema de otimização global de rações.

O presente trabalho tem por objetivo a descrição de um software matemático, em microcomputadores, aplicado à área de formulação de rações.

Especial atenção foi dada ao problema de instabilidade **numérica**, comum em sistemas deste gênero. Fatorizações ortogonais da base foram utilizadas para manipular este tipo de problema.

Devido à estrutura bloco angular da matriz, a fatorização **L-Q** foi utilizada visando economia de memória.

ÍNDICE

1. O PROBLEMA DO BALANCEAMENTO DE RAÇÕES

1.1 - DEFINIÇÃO

Uma ração animal balanceada consiste na mistura de determinadas matérias primas* para satisfazer um conjunto de requisitos nutricionais** especificados.

O objetivo do formulador é o de encontrar não apenas a mistura que atenda todas as especificações técnicas, mas também aquela que apresente **menor custo de produção.**

Os primeiros sistemas para formulação de rações foram desenvolvidos de forma a considerar apenas as restrições de ordem nutricional. Os problemas com limitações de estoque de matérias primas não eram considerados, visto que os computadores existentes na época não possuíam recursos suficientes para tal. O formulador, portanto, deveria aliar à sua experiência técnica, um sentimento adicional: alocar as matérias primas, com problemas de limitação em seus estoques, nos diversos produtos produzidos, de forma a minimizar o custo total de produção.

Com a chegada dos computadores de 3ª geração e com o desenvolvimento de softwares na área de programação linear*** as grandes fábricas de ração passaram a administrar os níveis de estoque de maneira mais conveniente, obtendo diferenças da ordem de **US$ 2,00** a **US$ 3,00** por tonelada de ração produzida.

---

* Milho, soja, trigo, carne, etc.

** Proteína, cálcio, fósforo, energia, etc.

*** Como exemplo citaríamos o "MPSX"-IBM; o "MINOS"-UNIVERSIDADE DE STANFORD e o "TEMPO"-BURROUGHS.

Devido ao alto número de restrições geradas pela formulação com limitações de estoques de matérias primas* e pela complexidade dos programas que compunham o software matemático, o sistema de computação utilizado necessitava de uma grande capacidade de armazenamento em memória principal e auxiliar, o que implicava um alto custo para aquisição do equipamento.

As pequenas indústrias, ficavam então, restritas ao uso dos micro-computadores de 8 bits, utilizando-se, portanto, de softwares pouco complexos** na área de nutrição.

Com o aparecimento dos micro-computadores de 16 bits e dos discos rígidos*** para armazenamento, a custos competitivos, projetos mais ambiciosos puderam tomar forma.

Este trabalho tem por objetivo descrever um destes projetos: a resolução do problema do balanceamento de rações com limitações de estoque de matérias primas.

Três objetivos principais nortearam o desenvolvimento deste trabalho:

1 - O método deve se resguardar contra problemas de "**instabilidade numérica**", tão frequentes na resolução de um sistema de rações.

2 - A **quantidade de memória** utilizada deve ser "a menor possível", possibilitando com que um grande número de produtos possam ser otimizados conjuntamente.

---

* Estatísticas nos fornecem um número em torno de 2100 restrições para uma formulação de 70 produtos, com 30 requisitos nutricionais e 60 matérias primas por produto, e 10 restrições de estoque de matérias primas.

** Não consideravam as restrições de estoque.

*** Winchester.

3 - O tempo de resolução deve ser aceitável permitindo com que o formulador possa realizar vários testes visando uma melhor qualidade e competitividade de seus produtos.

1.2 - O Modelo de Rações Simples

O objetivo principal do formulador é o de obter uma ração balanceada a mínimo custo possível.

Chamemos de "C" ao custo (Cz$/Kg) de um conjunto de matérias primas que podem ser combinadas em quantidades "X" (%), para obtenção de determinado tipo de ração. Desta forma o objetivo de nosso formulador passa a ser representado pela função:

$$\text{"MIN } \sum_{J=1}^{N} C_j X_j \text{"}$$

onde N representa o total de matérias primas alocadas na ração.

A ração a ser produzida deve atender também a determinadas exigências (requisitos) nutricionais que serão satisfeitas através da combinação das diversas matérias primas.

Indiquemos por a_{ij} a quantidade do nutriente i na composição química da matéria prima j. O conjunto de exigências nutricionais passa a ser representado por:

$$\text{" } b_i \leq \sum_{j=1}^{N} a_{ij} x_j \leq B_i \text{ "}$$

onde i representa o conjunto de exigências nutricionais e (b_i, B_i) seus limites de variação.

Finalmente, por questões técnicas, algumas matérias pri

mas devem ser misturadas em quantidades limitadas.

O conjunto de limitações técnicas das matérias primas passa a ser representado por:

$$\ell_j \leq x_j \leq u_j$$

onde j representa o conjunto de matérias primas e (ℓ_j, u_j) seus limites técnicos.

Do ponto de vista matemático a formulação do problema da ração simples passa a ser indicado por:

$$\text{Minimizar } f(x) = C^T X \quad , C \in R^N \quad , \quad C = (c_1, \ldots, c_N)$$

$$, X \in R^N \quad , \quad X = (x_1, \ldots, x_N)$$

$$\text{S.A} \quad b \leq AX \leq B \quad , A \in R^{MxN} \quad , \quad A = (a_1, \ldots, a_M)^T$$

$$L \leq X \leq U \quad , L \in R^N \quad , \quad L = (\ell_1, \ldots, \ell_N)$$

$$U \in R^N \quad , \quad U = (u_1, \ldots, u_N)$$

1.3 - O Modelo de Rações Múltiplas

O objetivo principal do formulador é o de obter o mínimo custo total de produção de um lote de rações.

Chamemos de "C" ao custo (Cz$/Kg) de um conjunto de matérias primas que podem ser combinadas em quantidades "X" (%) em cada ração do lote e "Q" a quantidade a ser produzida de cada ração do lote. O objetivo do formulador passa a ser representado pela função:

$$\text{"MIN} \sum_{i=1}^{R} \sum_{j=1}^{N} Q_i \, (C_j X_{ij})\text{"}$$

onde R indica o número de rações;

As exigências (requisitos) nutricionais devem ser atendidas pela combinação das diversas matérias primas, em cada ração do lote.

Indiquemos por a_{ij} a quantidade do nutriente i na composição química da matéria prima j. O conjunto de exigências nutricionais, para cada ração, passa a ~~ser~~ representado por:

$$b_{1i} \leq \sum_{j=1}^{N} a_{ij} x_{1j} \leq B_{1i} \quad \rightarrow \text{Ração 1}$$

$$b_{2i} \leq \sum_{j=1}^{N} a_{ij} x_{2j} \leq B_{2i} \quad \rightarrow \text{Ração 2}$$

$$\vdots \qquad\qquad\qquad \vdots$$

$$b_{Ri} \leq \sum_{j=1}^{N} a_{ij} x_{Rj} \leq B_{Ri} \quad \rightarrow \text{Ração R}$$

onde i representa o conjunto de exigências nutricionais de cada produto, e (b_{Ri}, B_{Ri}) os limites de variação.

Por questões técnicas, algumas matérias primas devem ser misturadas em quantidades limitadas.

O conjunto de limitações técnicas das matérias primas, em cada ração, passa a ser representada por:

$$L_1 \leq X_1 \leq U_1 \quad \rightarrow \quad \text{Ração 1}$$

$$L_2 \leq X_2 \leq U_2 \quad \rightarrow \quad \text{Ração 2}$$

$$\vdots \quad\quad \vdots \quad\quad \vdots$$

$$L_R \leq X_R \leq U_R \quad \rightarrow \quad \text{Ração R}$$

onde X representa o conjunto de matérias primas de cada produto, e (L, U) seus limites técnicos.

Finalmente, algumas matérias primas, ou não podem ser consumidas livremente, ou, por questões de armazenagem, devem obrigatóriamente ser consumidas.

O conjunto de restrições de estoque passa a ser representado por:

$$e_1 \leq \sum_{i=1}^{R} Q_i x_{i1} \leq E_1 \rightarrow \text{Matéria prima 1}$$

$$e_2 \leq \sum_{i=1}^{R} Q_i x_{i2} \leq E_2 \rightarrow \text{Matéria prima 2}$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$e_t \leq \sum_{i=1}^{R} Q_i x_{it} \leq E_t \rightarrow \text{Matéria prima t}$$

onde t representa o total de matérias primas com limitações de estoque alocadas no lote de rações, e (e_t, E_t) seus limites de variação.

Do ponto de vista matemático a formulação do problema da ração múltipla passa a ser indicado por:

Minimizar

$$f(x) = Q_1 \overset{T}{C_1} X_1 + Q_2 \overset{T}{C_2} X_2 \ldots\ldots + Q_R \overset{T}{C_R} X_R$$

S.A. $\quad b_1 \leq A_1 X_1 \leq B_1$

$$L_1 \leq X_1 \leq U_1$$

$$b_2 \leq A_2 X_2 \leq B_2$$

$$L_2 \leq X_2 \leq U_2$$

$$b_R \leq A_R X_R \leq B_R$$

$$L_R \leq X_R \leq U_R$$

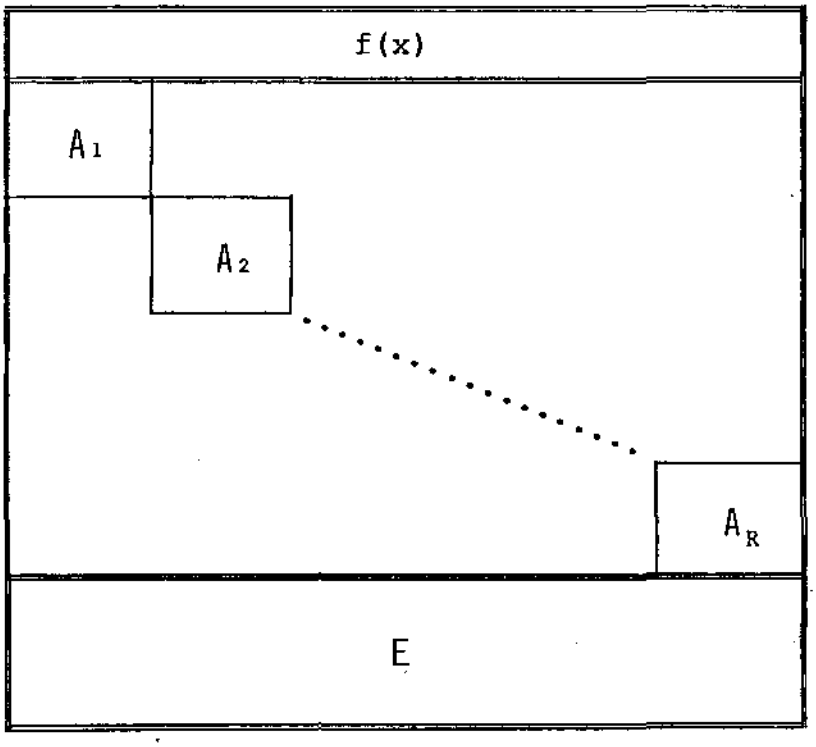
$$e_1 \leq Q_1 x_{11} + Q_2 x_{21} + \ldots\ldots Q_R x_{R1} \leq E_1$$
$$\vdots \quad \vdots \quad\quad \vdots$$
$$e_t \leq Q_1 x_{1t} + Q_2 x_{2t} + \ldots\ldots Q_R x_{Rt} \leq E_t$$

$$A_1, A_2, \ldots, A_R \in \mathbb{R}^{M \times N},$$

$$A_1, A_2, \ldots, A_R = (a_1, \ldots, a_M)^T$$

$$C_i \in \mathbb{R}^N, \quad C_i = (C_1, \ldots, C_R)$$
$$X_i \in \mathbb{R}^N, \quad X_i = (X_1, \ldots, X_R)$$
$$L_i \in \mathbb{R}^N, \quad L_i = (L_1, \ldots, L_R)$$
$$U_i \in \mathbb{R}^N, \quad U_i = (U_1, \ldots, U_R)$$
$$e \in \mathbb{R}^t, \quad e = (e_1, \ldots, e_t)$$
$$E \in \mathbb{R}^t, \quad E = (E_1, \ldots, E_t)$$

Esquematicamente:



2. O ALGORÍTMO MATEMÁTICO

2.1 - AS CONDIÇÕES DE KUHN-TUCKER

A condição necessária (KUHN-TUCKER) [8,9] para que um ponto $X^{\phi} \in \mathbb{R}^N$, seja ótimo para problemas do tipo

$$\begin{aligned} &\text{Min } f(x) = C^T X \quad ; && C \in \mathbb{R}^N;\ C = (c_1, \ldots, c_N)^T \\ &\text{S.A} && X \in \mathbb{R}^N;\ X = (x_1, \ldots, x_N)^T \\ &AX \lesseqgtr b && A \in \mathbb{R}^{M \times N};\ A = (a_1, \ldots, a_M)^T \\ &L \leq X \leq U && b \in \mathbb{R}^M;\ b = (b_1, \ldots, b_M)^T \\ &X \geq \emptyset \end{aligned} \tag{1}$$

é que em X^{ϕ}, o $\nabla f(x)$ esteja contido no cone gerado pelos gradientes das restrições que passam por X^{ϕ}.

Em outras palavras, existem vetores $\pi \in \mathbb{R}^M$ e $\mu \in \mathbb{R}^N$ tais que:

$$C = \sum_{i \in I} \pi_i a_i + \sum_{j \in J} \mu_j k_j$$

$$I = \{i: \sum_{j=1}^{N} a_{ij} x_j^{\phi} = b_i\} \; ; \; J = \{J: x_j^{\phi} = \emptyset\}$$

$$k_j = (\emptyset, \emptyset, \ldots, \underset{\substack{\uparrow \\ j \text{ posição}}}{1}, \emptyset, \ldots, \emptyset)^T$$

ou seja $\nabla f(x)$ é formado por uma combinação linear dos gradientes das Restrições Ativas em X^{ϕ}.

2.2. O MÉTODO MATEMÁTICO

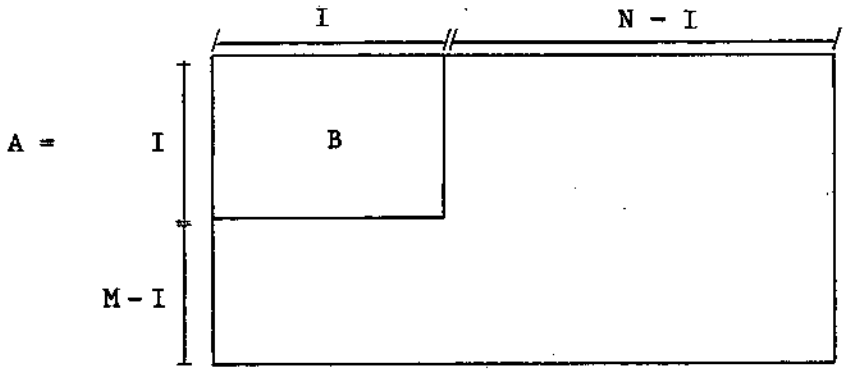
2.2.1 - DETERMINAÇÃO DE SOLUÇÃO ÓTIMA

Seja $A \in \mathbb{R}^{M \times N}$ e $X^{\phi} \in \mathbb{R}^{N}$ um ponto qualquer factível, para o problema representado por (1). Façamos as seguintes operações em "A":

1ª - Rearranjar as linhas de "A" de tal forma que as I primeiras linhas sejam formadas pelas restrições ativas em X^{ϕ}.

2ª - Rearranjar as colunas de "A" de tal forma que as I primeiras colunas correspondam às canalizações inativas.

A matriz "A" passa à forma:



A matriz $B \in \mathbb{R}^{I \times I}$, formada pelas linhas ativas de $A \in \mathbb{R}^{M \times N}$ e pelas colunas correspondentes às canalizações inativas é quadrada, formando base para o problema representado por (1).[5]

Com efeito,

Seja o sistema $AX \leq b$, $A \in \mathbb{R}^{M \times N}$, $b \in \mathbb{R}^{M}$; adicionando-se

as variáveis de folga, temos $AX + X_S = b$, $X \geq 0$ e $X_S \geq 0$. A matriz aumentada $A' \in \mathbb{R}^{Mx(M+N)}$ tem posto M. Uma solução inicial para o novo sistema é obtida, fazendo $X_S = b$ e $X = 0$, portanto, para M restrições ativas, teremos sempre M canalizações inativas.

Aplicando as condições de KUHN-TUCKER para verificação da solução em X^{ϕ} teríamos:

$$\begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_I \\ C_{I+1} \\ C_{I+2} \\ \vdots \\ C_N \end{bmatrix} = \pi_1 \begin{bmatrix} a_{11} \\ a_{12} \\ \vdots \\ a_{1I} \\ a_{1I+1} \\ a_{1I+2} \\ \vdots \\ a_{1N} \end{bmatrix} + \pi_2 \begin{bmatrix} a_{21} \\ a_{22} \\ \vdots \\ a_{2I} \\ a_{2I+1} \\ a_{2I+2} \\ \vdots \\ a_{2N} \end{bmatrix} + \ldots + \pi_I \begin{bmatrix} a_{I1} \\ a_{I2} \\ \vdots \\ a_{II} \\ a_{II+1} \\ a_{II+2} \\ \vdots \\ a_{IN} \end{bmatrix} + \mu_1 \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \mu_2 \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} + \ldots \mu_{N-I} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \tag{3}$$

Por inspeção, verificamos que a determinação de $\pi \in \mathbb{R}^I$ é feita por:

$$B^T \pi = \bar{C} \quad , \quad \bar{C} = (C_1, \ldots, C_I)^T$$

Em consequência, $\mu \in \mathbb{R}^{N-I}$ é obtido por substituição em (3):

$$\mu_i = C_{I+i} - \sum_{j=1}^{I} a_{j,I+i} \; \pi_j \; , \; i = 1, \ldots, N-I \tag{4}$$

Para o problema definido em (1), X^{ϕ} é solução do sistema se os sinais dos multiplicadores π e μ estiverem corretos, ou seja, se em X^{ϕ} ocorrer:

$$\sum_{j=1}^{N} a_{ij} x_j^{\phi} = b_i^L \qquad \Rightarrow \pi_i > 0$$

$$\sum_{j=1}^{N} a_{ij} x_j^{\phi} = b_i^u \qquad \Rightarrow \pi_i < 0$$

$$x_j^{\phi} = \ell_j \qquad \Rightarrow \mu_j > 0$$

$$x_j^{\phi} = u_j \qquad \Rightarrow \mu_j < 0$$

Portanto, dado um ponto $X^{\phi} \in \mathbb{R}^N$, o primeiro passo do método é determinar se X^{ϕ} é solução do problema descrito em (1).

2.2.2 - CÁLCULO DE UMA DIREÇÃO FACTÍVEL

Se o ponto $X^{\phi} \in \mathbb{R}^N$ não representa uma solução ótima para o problema descrito em (1), então alguns dos multiplicadores $\pi \in \mathbb{R}^I$ e $\mu \in \mathbb{R}^{(N-I)}$ tem "o sinal errado". Neste caso devemos calcular uma direção factível para determinação de um novo ponto $X \in \mathbb{R}^N$.

Tomemos então o multiplicador que possua o sinal "mais errado", ou seja, o maior em valor absoluto.

Dois casos devem ser considerados:

Caso 1) $\pi_i \in \mathbb{R}^I$ é o multiplicador escolhido

Neste caso a i-ésima restrição deve deixar de ser ativa. A direção $d \in \mathbb{R}^N$ deve manter ativas todas as restrições que eram ativas, exceto em relação a i-ésima restrição. Se a i-ésima restrição for do tipo $\leq$ devemos resolver o seguinte sistema:

$$a_j^T d = 0 \text{ p/ } j \in \{1, 2, \ldots, (i-1), (i+1), \ldots, I\}$$

$$a_i^T d = -1$$

$$d_j = 0 \text{ p/ } j > I$$

(5)

Esquematicamente:

$$\begin{array}{lllllllllllll}
a_{11} & d_1 & + & a_{12} & d_2 & + & a_{13} & d_3 & + \ldots + & a_{1N} & d_N & = & 0 \\
a_{21} & d_1 & + & a_{22} & d_2 & + & a_{23} & d_3 & + \ldots + & a_{2N} & d_N & = & 0 \\
\vdots & & & \vdots & & & \vdots & & & \vdots & & & \vdots \\
a_{i1} & d_1 & + & a_{i2} & d_2 & + & a_{i3} & d_3 & + \ldots + & a_{iI} & d_N & = & -1 \\
a_{(i+1)_1} & d_1 & + & a_{(i+1)_2} & d_2 & + & a_{(i+1)_3} & d_3 & + \ldots + & a_{(i+1)_N} & d_N & = & 0 \\
\vdots & & & \vdots & & & \vdots & & & \vdots & & & \vdots \\
a_{I1} & d_1 & + & a_{I2} & d_2 & + & a_{I3} & d_3 & + \ldots + & a_{IN} & d_N & = & 0
\end{array}$$

Como $d_j = 0$ p/ $j > I$, então o sistema acima passa a ser determinado por:

$$B\bar{d} = (0, 0, \ldots, -1, 0, \ldots, 0)^T$$

$$d = (\bar{d}, 0)$$

Se a i-ésima restrição for do tipo $\geq$ o sistema a ser resolvido é o mesmo, com exceção da i-ésima restrição que passa a:

$$a_i^T d = 1$$

(6)

Caso 2) $\mu_i \in \mathbb{R}^{(N-k)}$ é o multiplicador escolhido

O processo é semelhante ao descrito no caso 1.

Se a i-ésima variável for do tipo $\leq$ devemos resolver o sistema:

$$\begin{aligned} &{}^{T}_{j}\, d_j = 0 \quad p/\ j = 1, \ldots, I \\ &d_j = 0 \quad p/\ j > I \\ &d_i = -1 \end{aligned} \tag{7}$$

Se a i-ésima variável for do tipo $\geq$ o sistema a ser resolvido é o mesmo com exceção da i-ésima variável que passa a:

$$d_i = 1 \tag{8}$$

Em ambos os casos devemos resolver também, sistemas do tipo

$$Bd = Z \qquad Z \in \mathbb{R}^{I}$$

2.2.3 - CÁLCULO DE NOVO PONTO X FACTÍVEL

O novo ponto $X \in \mathbb{R}^{N}$ será calculado por meio da expressão:

$$X = X^{\phi} + \lambda d \ ; \ d \in \mathbb{R}^{N} \ \lambda > 0$$
$$X,\ X^{\phi} \in \mathbb{R}^{N} \tag{9}$$

Consideremos os casos:

Caso 1) π_i é o multiplicador com sinal incorreto

Neste caso uma restrição que era ativa no ponto X^{ϕ}, não o será em X. Em contra partida, uma restrição ou canalização que não era ativa em X^{ϕ}, passa a ser ativa em .

Caso 2) μ_i é o multiplicador com sinal incorreto

Neste caso uma canalização que era ativa em X^{ϕ}, não o será em X. Em contra partida, uma canalização ou restrição que não era ativa em X^{ϕ}, passa a ser ativa em X.

Em ambos os casos basta verificarmos dentre o conjunto de restrições inativas e o conjunto de canalizações inativas, a restrição ou canalização que cumpre o mínimo entre:

1) Para as restrições inativas: $i = \{I / b_i^L < a_{ij} x_j^{\phi} < b_i^u\}$

$$\sum_{j=1}^{N} a_{ij} (x_j^{\phi} + \lambda d_j) = b_i$$

$$\Rightarrow \quad \sum_{j=1}^{N} a_{ij} x_j^{\phi} + \lambda \sum_{j=1}^{N} a_{ij} d_j = b_i$$

$$A = \frac{b_i^u - \sum_{j=1}^{N} a_{ij} x_j^{\phi}}{\sum_{j=1}^{N} a_{ij} d_j} \qquad B = \frac{b_i^L - \sum_{j=1}^{N} a_{ij} x_j^{\phi}}{-\sum_{j=1}^{N} a_{ij} d_j}$$

(A) **(B)**

2) Para as canalizações inativas: $j = \{j / \ell_j < x_j^{\phi} < u_j\}$

$$C = \frac{u_j - x_j^{\phi}}{d_j} \qquad D = \frac{\ell_j - x_j^{\phi}}{-d_j}$$

(C) **(D)**

$$\therefore \lambda = \text{MIN} \ (\text{MAX}(A, 0), \text{MAX}(B, 0), \text{MAX}(C, 0), \text{MAX}(D, 0)) \qquad \textbf{(10)}$$

É interessante notar que a restrição (canalização) ativa que possuía o multiplicador escolhido pode apenas mudar de sinal ("≤" para "≥" ou vice-versa), em consequência, a base não sofreria modificações em sua estrutura.

2.2.4 - FATORIZAÇÃO DA BASE

Dada uma matriz $B \in \mathbb{R}^{I \times I}$ (descrita em 2.2.1), existe uma matriz $Q \in \mathbb{R}^{I \times I}$, ortogonal tal que:

$$B.Q = L \quad ; \quad L \in \mathbb{R}^{I \times I} \qquad \text{(A)}$$

Pela definição de matrizes ortogonais temos

$$Q\,Q^T = Q^T Q = I$$

pós-multiplicando-se ambos os membros de (A) por Q^T teremos:

$$BQQ^T = LQ^T \Longrightarrow B = LQ^T \qquad \text{(B)}$$

transpondo-se ambos os membros de (B)

$$B^T = (LQ^T)^T \Longrightarrow B^T = QL^T \qquad \text{(C)}$$

multiplicando-se (B) x (C)

$$BB^T = L\underbrace{Q^T Q}_{I}L^T \Longrightarrow BB^T = LL^T$$

Na determinação do algorítmo matemático devemos ser capazes de resolver sistemas do tipo:

$$B^T\pi = C \quad \text{e} \quad Bd = Y \ ;$$

01) Determinar $B^T\pi = C$

multiplicando-se ambos os lados por B teremos:

$BB^T\pi = BC$; façamos $BC = W$

então

$BB^T\pi = W \Longrightarrow LL^T\pi = W$; façamos $L^T\pi = Z$

O sistema $B^T\pi = C$ passa a ser determinado da seguinte maneira:

A. Calcular $BC = W$

multiplicação de matriz por vetor

B. Calcular $LZ = W$

Como L é uma matriz triangular inferior, o sistema acima é resolvido por "Forward substitutions"

C. Calcular $L^T\pi = Z$

como L^T é uma matriz triangular superior o sistema acima é resolvido por "backward substitutions"

02) Determinar $Bd = Y$ (d = variável de decisão)

Façamos $B^TW = d$;

então

$BB^TW = Y \Longrightarrow LL^TW = Y$; façamos $L^TW = Z$

O sistema $Bd = Y$ passa a ser determinado da seguinte maneira:

A. Calcular $LZ = Y$

Como L e uma matriz triangular inferior, o sistema acima é resolvido por "forward substitutions".

B. Calcular $L^T W = Z$

Como L^T é uma matriz triangular superior, o sistema é resolvido por "backward substitutions"

C. Calcular $B^T W = d$

multiplicação de matriz transposta por vetor

Em ambos os casos (determinação de π e d) é necessário armazenar as matrizes B e L, o que seria dispendioso em termos de memória.

Utilizando-se o fato de $QQ^T = Q^T Q = I$ vamos reescrever os casos acima:

01) Determinar $B^T \pi = C$

Multiplicando-se ambos os lados por Q^T teremos:

$Q^T B^T \pi = Q^T C \Longrightarrow (B.Q)^T \pi = Q^T C$, mas $B.Q = L$

$\Longrightarrow L^T \pi = Q^T C$

O sistema $B^T \pi = C$ passa a ser determinado por:

A. Calcular $Q^T C = W$

multiplicação de matriz por vetor

B. Calcular $L^T \pi = W$

Como L^T é uma matriz triangular superior, o sistema acima é resolvido por "backward substitutions"

Verificamos agora, que a determinação dos multiplicadores π_i foi feita sem a utilização explícita da matriz B.

02) Determinar $Bd = Y$

como $QQ^T = Q^T Q = I$

então

$$BQQ^T d = Y \implies LQ^T d = Y \ ; \text{façamos } Q^T d = W$$

O sistema $Bd = Y$ passa a ser determinado por

A. Calcular $LW = Y$

Como L é triangular inferior, o sistema acima é resolvido por "forward substitutions"

B. Calcular $Q^T d = W$ ou $QW = d$

multiplicação de matriz por vetor

Novamente, não é necessário a utilização explícita da matriz B. Em contra partida, em ambos os casos, devemos utilizar a matriz de transformações lineares Q, utilizada na transformação $BQ = L$.

O número de operações aritméticas, envolvidas na resolução dos sistemas lineares anteriormente descritos, é especificado nas tabelas abaixo.

TABELA 01 - Resolução do sistema $B^T\pi = C$ (modo 01)

SISTEMA	DIVISÕES	MULTIPLICAÇÕES	ADIÇÕES	SUBTRAÇÕES
$BC = W$		N^2	$(N^2 - N)$	
$LZ = W$	N	$(N^2 - N)/2$		$(N^2 - N)/2$
$L^T\pi = Z$	N	$(N^2 - N)/2$		$(N^2 - N)/2$
TOTAL	2N	$2N^2 - N$	$N^2 - N$	$N^2 - N$

TABELA 02 - Resolução do sistema $B^T\pi = C$ (modo 02)

SISTEMA	DIVISÕES	MULTIPLICAÇÕES	ADIÇÕES	SUBTRAÇÕES
$Q^TC = W$		$(N^2+N)/2$	$(N^2-N)/2$	
$L^T\pi = W$	N	$(N^2-N)/2$		$(N^2-N)/2$
TOTAL	N	N^2	$(N^2-N)/2$	$(N^2-N)/2$

TABELA 03 - Resolução do sistema Bd = Y (modo 01)

SISTEMA	DIVISÕES	MULTIPLICAÇÕES	ADIÇÕES	SUBTRAÇÕES
$LZ = Y$	N	$(N^2-N)/2$		$(N^2-N)/2$
$L^TW = Z$	N	$(N^2-N)/2$		$(N^2-N)/2$
$B^TW = d$		N^2	(N^2-N)	
TOTAL	N	$2N^2-N$	N^2-N	$(N^2-N)/2$

TABELA 04 - Resolução do sistema Bd = Y (modo 02)

SISTEMA	DIVISÕES	MULTIPLICAÇÕES	ADIÇÕES	SUBTRAÇÕES
$LW = Y$	N	$(N^2-N)/2$		$(N^2-N)/2$
$QW = d$		$(N^2+N)/2$	$(N^2-N)/2$	
TOTAL	N	N^2	$(N^2-N)/2$	$(N^2-N)/2$

De acordo com as tabelas 02 e 04, notamos um decréscimo de aproximadamente 50% no número de operações aritméticas em comparação com as tabelas 01 e 03. O problema agora é de como armazenar de forma eficiente a matriz de transformações Q.

2.3 - FATORIZAÇÃO DA BASE POR HOUSEHOLDER

Dada uma matriz $A \in \mathbb{R}^{M \times N}$, de posto M, existe uma matriz Q, ortogonal, $\in \mathbb{R}^{N \times N}$ que cumpre

$$AQ = [L\ 0] \quad ; \ L \in \mathbb{R}^{M \times N}, \text{ triangular inferior} \tag{11}$$

A matriz Q é expressa como um produto de M transformações lineares de Householder ((M - 1) transformações no caso de $A \in \mathbb{R}^{M \times M}$), ou seja,

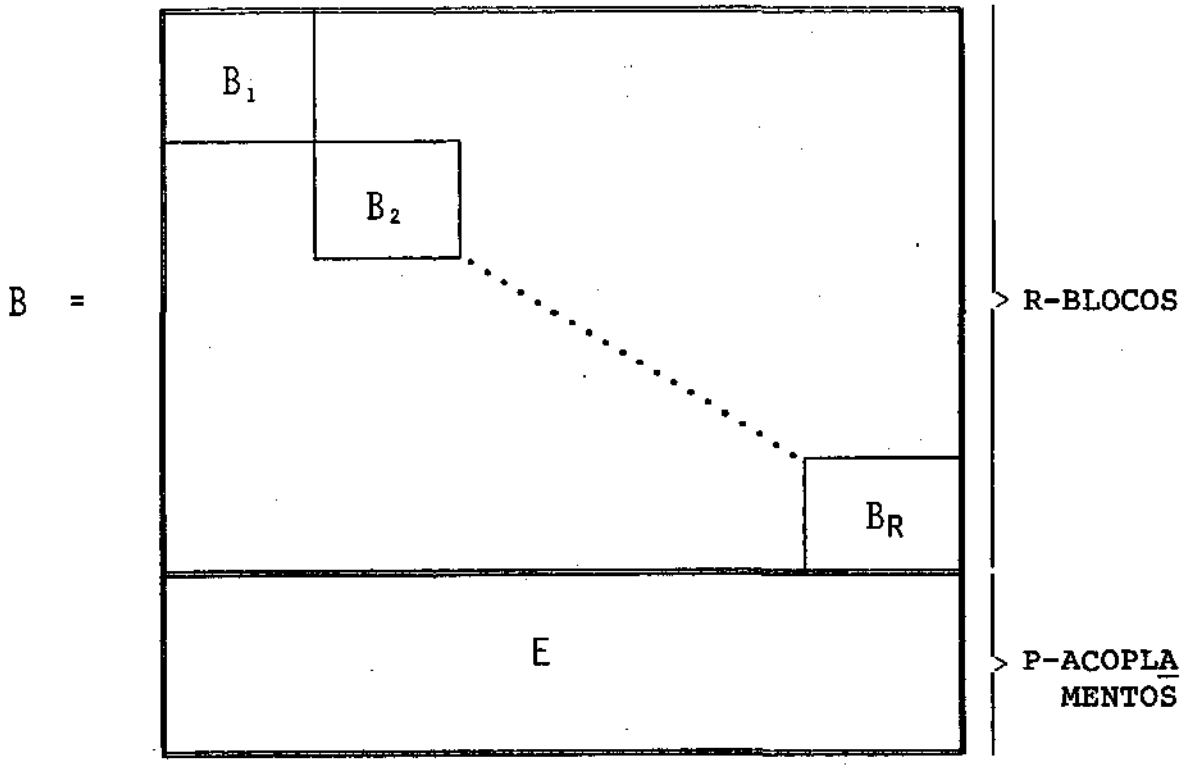
$$Q = \underbrace{(I - 2W_1 W_1^T)}_{Q_1} \underbrace{(I - 2W_2 W_2^T)}_{Q_2} \ldots \underbrace{(I - 2W_M W_M^T)}_{Q_M} \quad [7, 12]$$

Armazenar explicitamente a matriz Q é dispendioso, visto que as dimensões do problema definido em 1.3 são, na maioria dos casos, elevadas. Porém os vetores W são da forma

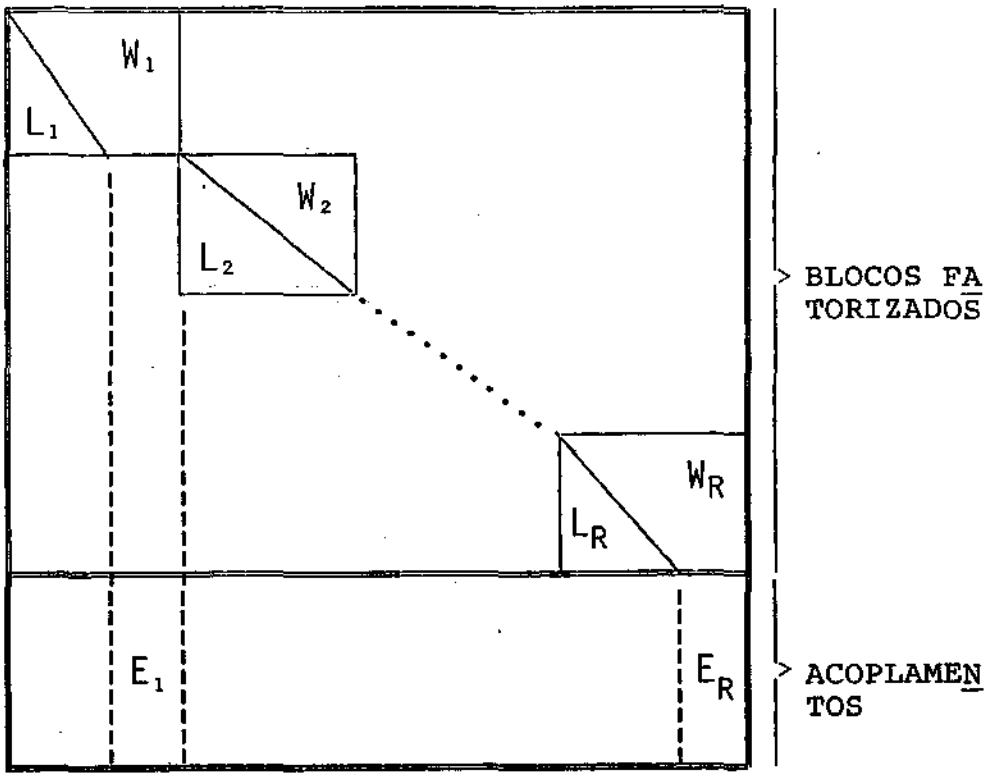
$$W_1 = \begin{bmatrix} W_{11} \\ W_{12} \\ \vdots \\ W_{N-1} \\ W_N \end{bmatrix} \quad W_2 = \begin{bmatrix} 0 \\ W_{22} \\ \vdots \\ W_{N-1} \\ W_N \end{bmatrix} \quad \ldots\ldots \quad W_{M-1} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ W_{N-1} \\ W_N \end{bmatrix} \quad W_M = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ W_N \end{bmatrix}$$

Podendo, portanto, ser armazenado nos espaços de zeros da L. Será necessário apenas a alocação de um vetor auxiliar $\epsilon\ \mathbb{R}^{M}$ para armazenar a diagonal de L.

Esquematicamente

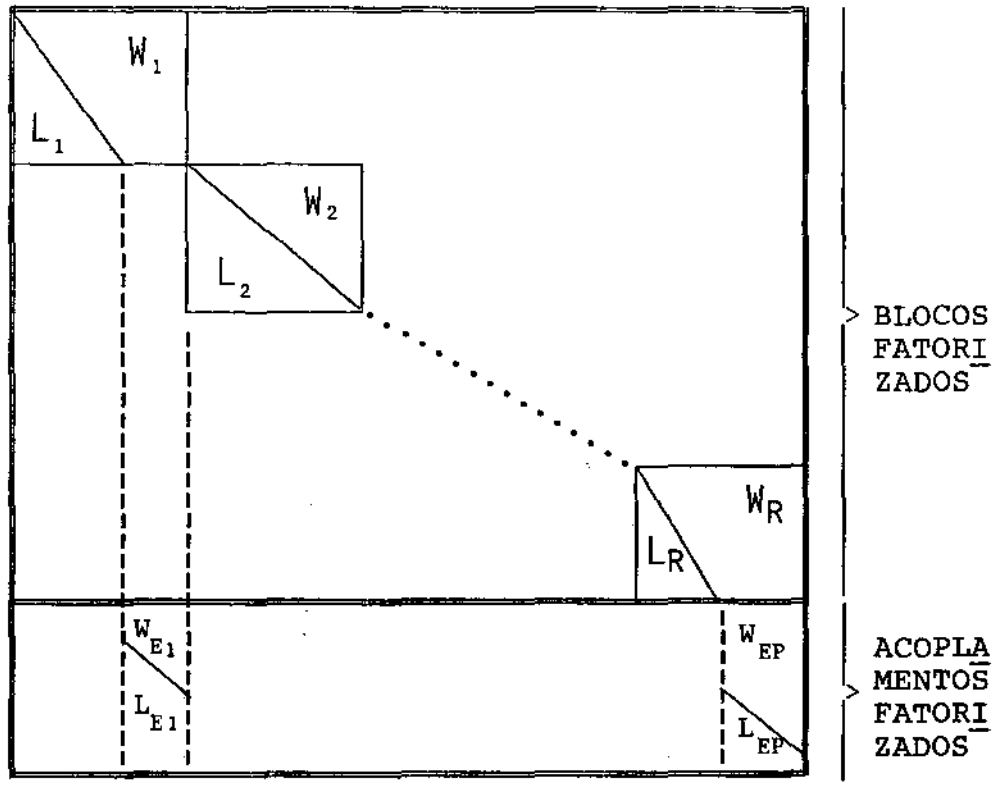


onde $B_i \epsilon\ \mathbb{R}^{I_i \times K_i}$ $(I_i \leq K_i)$; $E \epsilon\ \mathbb{R}^{P \times N}$, após a fatoração ortogonal nos blocos teremos



onde $L_i \in \mathbb{R}^{I_i \times I_i}$; $E_i \in \mathbb{R}^{P \times (K_i - I_i)}$

A mesma fatoração deve ser aplicada aos blocos do acoplamento. Ao final do processo teremos



onde $L_{Ei} \in \mathbb{R}^{P \times (K_i - I_i)}$

Existem dois algorítmos utilizados para calcular "BQ = L". O algorítmo clássico e o algorítmo de Nai-Kuan-Tsao.

Utilizaremos o algorítmo de Nai-Kuan-Tsao que fornece ligeira economia de tempo e de memória durante o processo.

Os passos são os seguintes:

Para cada linha i da matriz B

faça

Calcular $Sxx = \sum_{j=i}^{K} a_{ij}^2$

Se $Sxx \neq 0$

então faça

$S = \sqrt{Sxx}$

Se $a_{ii} \geq 0$

então

$u_1 = a_{ii} + S$

Senão

$u_1 = a_{ii} - S$

Calcular $Sxy = \sum_{j=i}^{K} a_{ij} * a_{(i+1)j}$

$Z_1 = - Sxy/S$

$d = (a_{(i+1)i} - Z_1) / u_1$

$a_{(i+1)i} = Z_1$

Calcular $a_{(i+1)j} = a_{(i+1)j} - a_{ij} * d \quad ; \quad j = (i+1), \ldots, k$

FIM

Ao final do processo a matriz L estará na parte triangular inferior e os vetores W na parte triangular superior do espaço ocupado pela matriz B.

Os motivos principais que nos levaram a optar pelo uso de transformações Householder são:

1. Preservar a estabilidade numérica do método, visto que as transformações ortogonais preservam norma e número de condição quando aplicadas.
2. Devido a estrutura bloco angular, característica do problema descrito em 1.3, a memória uti

lizada pelos blocos não sofreria modificação após o processo de fatorização da base.

2.4 - FLUXOGRAMA DO SISTEMA

Os seguintes passos descrevem o algorítmo proposto.

começo

faça

Determinar fatoração da base * ítem 2.3 -(11) *

Calcular multiplicadores * ítem 2.2.1 -(3,4) *

Se não existe "sinal errado"

então

pare

senão

começo

Calcular direção factível * ítem 2.2.2 -(5,6,7,8) *

Calcular novo ponto * ítem 2.2.3 - (9,10) *

FIM

Até não existir "sinal errado"

FIM

3. RESULTADOS NUMÉRICOS

Definiremos alguns termos que serão utilizados no contexto deste capítulo:

A. NEQUAC → número total de equações*

B. NVAR → número total de variáveis

C. NMAX → tamanho máximo da base** (atingido na i-ésima iteração)

D. NMED → tamanho médio da base

E. THOUSE → tempo total gasto com fatorizações***

F. TTOTAL → tempo total gasto para resolução

G. PORCENTO → porcentagem de thouse em Ttotal

Os testes foram feitos em um micro-computador compatível com o IBM PC/AT com a seguinte configuração:

Sistema Operacional	→ DOS 3.2
Memória	→ 640 Kb
Winchester	→ 20 Mb
Co-processador de ponto flutuante	→ 80287
Clock	→ 8 MHZ

3.1 - APLICAÇÃO À RAÇÕES MÚLTIPLAS

TESTE 01:

---

* Equivale à base do método SIMPLEX

** Ver capítulo 2, ítem 2.2

*** Ver capítulo 2, ítem 2.3

Produtos	....	6
Restrições de Acoplamento	....	3
Nequac	....	123
Nvar	....	180
Nmax	....	52 x 52
Nmed	....	44 x 44
Thouse	....	0'17"
Ttotal	....	2'50"
Porcento	....	10%

TESTE 02:

Produtos	....	18
Restrições de Acoplamento	....	3
Nequac	....	136
Nvar	....	324
Nmax	....	110 x 110
Nmed	....	109 x 109
Thouse	....	0'45"
Ttotal	....	3'12"
Porcento	....	23%

TESTE 03:

Produtos	....	26
Restrições de Acoplamento	....	4
Nequac	....	404
Nvar	....	547

Nmax	....	173 x 173
Nmed	....	164 x 164
Thouse	....	5'5"
Ttotal	....	26'3"
Porcento	....	19.5%

TESTE 04:

Produtos	....	30
Restrições de Acoplamento	....	10
Nequac	....	441
Nvar	....	862
Nmax	....	189 x 189
Nmed	....	171 x 171
Thouse	....	11'3"
Ttotal	....	60'
Porcento	....	18.4%

TESTE 05:

Produtos	....	42
Restrições de Acoplamento	....	10
Nequac	....	631
Nvar	....	1196
Nmax	....	266 x 266
Nmed	....	244 x 244
Thouse	....	25'52"
Ttotal	....	127'6"
Porcento	....	20%

3.2 - COMPARAÇÃO COM ALGORÍTMOS CONVENCIONAIS

Os testes comparativos foram efetuados com o "MINOS", desenvolvido pela Universidade de Stanford, rodando em um micro-computador compatível com IBM PC/AT (configuração descrita no ítem 3.1).

O termo "espaço em disco", a ser utilizado, refere-se ao número total de bytes gastos para geração do arquivo de dados, referente ao modelo, utilizado pelo "MINOS" e por nosso algorítmo. A figura abaixo ilustra este arquivo

Para efeito de simplificação nos referenciaremos à nosso algorítmo como "GRADPRO".

TESTE 01:

Produtos		....	18
Restrições de Acoplamento		....	3
Nequac		....	136
Nvar		....	324
Espaço em Disco	"MINOS"	....	160 Kbytes
	"GRADPRO"	....	14 Kbytes
Ttotal	"GRADPRO"	....	3'12"

TESTE 02:

Produtos		....	26
Restrições de Acoplamento		....	4
Nequac		....	404
Nvar		....	547
Espaço em Disco	"MINOS"	....	291 Kbytes
	"GRADPRO"	....	18 Kbytes
Ttotal	"GRADPRO"	....	26'3"

TESTE 03:

Produtos		....	30
Restrições de Acoplamento		....	10
Nequac		....	441
Nvar		....	862
Espaço em Disco	"MINOS"	....	451 Kbytes
	"GRADPRO"	....	35 Kbytes
Ttotal	"GRADPRO"	....	60'

TESTE 04:

Produtos		....	42
Restrições de Acoplamento		....	10
Nequac		....	631
Nvar		....	1196
Espaço em Disco	"MINOS"	....	641 kbytes
	"GRADPRO"	....	47 Kbytes
Ttotal	"GRADPRO"	....	127'6"

4. CONCLUSÕES

Devido à característica bloco angular do problema de balanceamento de rações, e pelo fato da matriz básica ser, na maioria dos casos, densa, é de grande valia a utilização de fatorizações pelo método de Householder.

A quantidade de memória, principal problema em um micro-computador, permanece praticamente inalterado durante o processo de transformação da base.

Visto que a fatorização representa, em média, 20% do tempo total gasto para resolução do modelo, optamos por mantê-la a cada iteração. Desta forma não haverá um aumento no código fonte ocasionado pelo algorítmo de atualização da fatorização.

Para garantir a estabilidade numérica do algorítmo, outro motivo que nos levou a utilizar fatorizações ortogonais da base, optamos também por escalar a matriz básica de duas maneiras:

A. Por linha

Cada linha é dividida por seu maior elemento (em módulo).

B. Por coluna

O conjunto de matérias primas alocadas nas rações é dividido em duas categorias: "MACRO" e

"MICRO" ingredientes.

As colunas, na matriz básica, correspondentes aos "MICRO" ingredientes são divididas por 100, o que normalmente leva a um equilíbrio de valores das variáveis.

BIBLIOGRAFIA

[1] Bartels, R.H. and Golub, G.H. (1969). The simplex method for linear Programming using the LU decomposition Comm. ACM 12, pp. 266 - 268.

[2] Dantzig, G.B. (1963). Linear Programming and Extensions, Princeton University Press, Princeton, New Jersey.

[3] Dantzig, G.B. and Wolfe, P. (1960). Decomposition principle for linear Programs. Operations Research 8, pp. 101 - 111.

[4] Gill, P.E., Golub, G.H., Murray, W. and Saunders, M. A. (1974). Methods for modifying matrix factorizations. Mathematics of Computation 28, pp. 505 - 535.

[5] Gill, P.E. and Murray, W. (1973). A numerically stable form of the simplex method. Linear Algebra and its Applics 7, pp. 99 - 138.

[6] Gill, P.E., Murray W. and Saunders, M.A. (1975) Methods for Computing and modifying the LDV factors of a matrix. Mathematics of Computation 29, pp. 1051 - 1077.

[7] Gill, P.E., Murray, W. and Wright, M.H. (1981). Practical Optimization, Academic Press, London.

[8] Lasdon, L.S. (1979). Optimization Theory for Large Systems, Mac Millan, New York.

[9] Luenberger, D.G. (1984). Linear and Nonlinear Programming, Addison Wesley, Reading, Massachussets.

[10] Murtagh, B.A. and Saunders, M.A. (1978). Large-scale linearly constrained optimization, Mathematical Programming 14, pp. 41 - 72.

[10] Murtagh, B.A. and Saunders, M.A. (1978). Large-scale Linearly constrained optimization, Mathematical Programming 14, pp. 41 - 72.

[11] Noble, Daniel (1977). Applied Linear Algebra, Prentice-hall, New Jersey.

[12] Rosen, J.B. (1964). Primal partition programming for block diagonal matrices, Numerische Mathematik 6, pp. 250 - 260.