DECOMPOSIÇÃO EM PROGRAMAÇÃO LINEAR

COM VARIÁVEIS CANALIZADAS:

APLICAÇÃO À OTIMIZAÇÃO GLOBAL DE RAÇÕES

VALÉRIA DE PODESTÁ GOMES

UNICAMP

UNIVERSIDADE ESTADUAL DE CAMPINAS

INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E CIÊNCIA DA COMPUTAÇÃO

CAMPINAS - SÃO PAULO
BRASIL

UNIVERSIDADE ESTADUAL DE CAMPINAS

UNICAMP

COORDENAÇÃO DOS CURSOS DE PÓS-GRADUAÇÃO

AUTORIZAÇÃO PARA QUE A UNICAMP POSSA FORNECER, A PREÇO DE CUSTO, CÓPIAS DA TESE A INTERESSADOS

Nome do Aluno: Valéria de Podestá Gomes

Nº de Identificação: 775469

Endereço para Correspondência: UNICAMP - IMECC - Depto de Matemática Aplicada caixa Postal 1170

Curso: Matemática Aplicada - Otimização e Pesquisa Operacional

Nome do Orientador: Miguel Taube Netto

Título da Dissertação ou Tese:

Decomposição em Programação Linear com variáveis canalizadas: aplicação à otimização global de rações

Data proposta para a Defesa: 25/06/82

(O Aluno deverá assinar um dos 3 ítens abaixo)

1) Autorizo a Universidade Estadual de Campinas a partir desta data, a fornecer, a preço de custo, cópias de minha Dissertação ou Tese a interessados.

20/06/82 — Data

Valéria — assinatura do aluno

2) Autorizo a Universidade Estadual de Campinas, a fornecer, a partir de dois anos após esta data, a preço de custo, cópias de minha Dissertação ou Tese a interessados.

/ / — Data

assinatura do aluno

3) Solicito que a Universidade Estadual de Campinas me consulte, dois anos após esta data, quanto à minha autorização para o fornecimento de cópias de minha Dissertação ou Tese, a preço de custo, a interessados.

/ / — Data

assinatura do aluno

DE ACORDO

Orientador

DECOMPOSIÇÃO EM PROGRAMAÇÃO LINEAR

COM VARIÁVEIS CANALIZADAS:

APLICAÇÃO À OTIMIZAÇÃO GLOBAL DE RAÇÕES

VALÉRIA DE PODESTÁ GOMES

ORIENTADOR

Prof. Dr. MIGUEL TAUBE NETTO

Dissertação apresentada no Instituto de Matemática, Estatística e Ciência da Computação, UNICAMP, como requisito parcial para obtenção do título de Mestre em Matemática Aplicada.

Junho 1982.

AGRADECIMENTOS:

Ao professor Miguel Taube Netto, pela proposta deste trabalho, pela orientação e incentivo constantes.

Aos amigos que me acompanharam durante este longo período, pela compreensão, apoio e companheirismo, em especial aos que habitaram o nosso inesquecível "Pombal" e ao Marcos, pela ajuda constante e paciente, contribuindo de forma decisiva na elaboração deste trabalho.

À Criança, por ser criança.

À D. Lourdes e ao Sr. Macário, por tudo.

Ao Laboratório de Matemática Aplicada, em cujo contexto este trabalho foi realizado.

SUMÁRIO

Neste trabalho, apresentamos os métodos de decomposição de Dantzig-Wolfe e de Rosen, quando aplicados a problemas de Programação Linear com estrutura bloco-angular e que possuem tanto variáveis quanto restrições canalizadas. Como caso particular, mostramos o problema da otimização simultânea de várias rações.

No capítulo 1, apresentamos o problema geral e, como caso particular deste, o problema da otimização global de rações. Estruturamos o problema de modo a transformá-lo numa forma padrão, onde as restrições canalizadas são transformadas em restrições de igualdade e deixando ainda as variáveis canalizadas com limites inferiores iguais a zero.

No capítulo 2, desenvolvemos o Método de Dantzig-Wolfe aplicado ao problema geral, comentando as simplificações que ocorrem na aplicação deste método ao problema da otimização global de rações.

No capítulo 3, desenvolvemos o Método de Rosen aplicado ao mesmo problema, e também comentamos a aplicação deste método ao problema da otimização global de rações.

No capítulo 4, comentamos as experiências computacionais obtidas com os vários programas desenvolvidos para a aplicação dos dois métodos.

Nos apêndices A, B e C apresentamos, respectivamente: um resumo do Método Simplex Revisado com variáveis canalizadas;

algumas etapas do Método de Rosen que não foram mostradas no desenvolvimento do capítulo 3; a documentação e a listagem do programa que resolve o problema da otimização global de rações.

INDICE

Página

Página

CAPÍTULO 1

CARACTERIZAÇÃO DO TRABALHO

1.1 - INTRODUÇÃO:

Dentre as aplicações mais conhecidas da Programação Linear, podemos destacar o problema clássico da produção de rações a custo mínimo. Uma ração se constitui em uma mistura de ingredientes com teores de certos nutrientes entre limites pré-estabelecidos. Procura-se determinar as porcentagens de cada ingrediente na mistura, de maneira a se minimizar o custo de produção da ração, e, ao mesmo tempo, satisfazer os requisitos nutricionais exigidos.

Neste trabalho, considera-se limitações superiores e inferiores nos estoques dos ingredientes e se faz a otimização global de várias rações simultaneamente, as quais têm quantidades prefixadas e disputam a utilização dos mesmos ingredientes.

A estrutura bloco-angular do problema permite a utilização de métodos de decomposição conhecidos: o Método de Dantzig-Wolfe [Lasdon, 1970] e [Taha, 1971] e o Método de Rosen [Lasdon, 1970] e [Rosen, 1964].

Nas implementações dos métodos, utilizou-se um tratamento específico para as variáveis canalizadas - variáveis limitadas inferior e superiormente. Os subproblemas com restrições canalizadas são transformados em uma forma equivalente onde aparecem apenas variáveis canalizadas.

Foram programadas versões dos Métodos de Dantzig-Wolfe e de Rosen para problemas gerais e para o problema específico de otimização global de rações. Utiliza-se, nestas versões, o Método do Simplex Revisado com variáveis canalizadas [Luenberger, 1973], [Murty, 1976].

1.2 - FORMULAÇÃO DO PROBLEMA DE OBTENÇÃO DE UMA RAÇÃO A CUSTO MÍNIMO:

Dados n ingredientes, o problema da obtenção de uma ração de custo mínimo consiste em escolher frações $y_j, j=1,\dots,n$, de cada ingrediente para compor uma unidade desta ração de forma a minimizar seu custo. São dadas as frações a_{ij} de cada nutriente i $(i=1,\dots,m)$ em cada ingrediente j, as frações mínimas e máximas de cada nutriente na ração e ainda os limitantes inferiores e/ou superiores das frações de cada ingrediente j na ração.

Sejam:

n : número de ingredientes disponíveis no instante da produção;

m : número de nutrientes que serão considerados na ração;

a_{ij} : fração do nutriente i no ingrediente j;

α_i : fração mínima do nutriente i na ração;

β_i : fração máxima do nutriente i na ração;

y_j : fração do ingrediente j na ração;

γ_j : fração mínima do ingrediente j na ração, $j \in I2$, onde I2 é o conjunto de ingredientes que possuem limitantes inferiores e superiores;

δ_j : fração máxima do ingrediente j na ração, $j \in I2$;

c_j : preço por unidade do ingrediente j.

Para eliminarmos as inviabilidades triviais do problema, estamos considerando que $\alpha_i \leq \beta_i$, $i = 1,\ldots,m$, $\gamma_j \leq \delta_j$, $j \in I2$ e $y_j \geq 0$, $j \in I1$, onde I1 é o conjunto de ingredientes que não possuem limitantes.

Assim, o problema é:

$$\left\{\begin{array}{l} \min \sum_{j=1}^{n} c_j y_j \\ \text{s.a.} \\ \qquad \alpha_i \leq \sum_{j=1}^{n} a_{ij} y_j \leq \beta_i \;, \; i = 1,\ldots,m \\ \qquad\qquad y_j \geq 0, \; j \in I1 \\ \qquad\qquad \gamma_j \leq y_j \leq \delta_j, \; j \in I2 \end{array}\right\} \quad (1.1)$$

Notemos que a restrição $\sum_{j=1}^{n} y_j \leq 1$ deve ser considerada. Podemos supor que esta restrição corresponde a um nutriente fictício i', com $a_{i'j} = 1$, $j = 1,\ldots,n$ e $\beta_{i'} = 1$. Quando a ração for tal que $\sum_{j=1}^{n} y_j < 1$, temos o caso em que é possível satisfazer as restrições nutricionais sem completar uma unidade da ração. Costuma-se, por isso, colocar na lista de ingredientes um "enchimento" de custo baixo e considerar $\sum_{j=1}^{n} y_j = 1$.

Estamos supondo também, nesta formulação, que os estoques dos ingredientes são infinitos, isto é, o mercado pode sempre suprir quaisquer quantidades de ingredientes.

1.3 - EXTENSÃO DO PROBLEMA PARA K RAÇÕES:

Nesta formulação, consideramos as limitações de estoques dos ingredientes, o que, frequentemente, acontece na prática. Podemos ter limitações superiores no caso da impossibilidade de dispor mais do que uma certa quantidade ℓ_j do ingrediente j durante o período de produção das rações. As limitações inferiores podem ocorrer na necessidade de consumir pelo menos uma quantidade g_j do ingrediente j, o qual pode estar, por exemplo, em processo de deterioração, e então precisa ser consumido rapidamente.

Então, para se produzir várias rações simultaneamente (K rações), com quantidades $q_1, q_2, \ldots, q_K$, respectivamente, e levando-se em conta as limitações de estoques dos ingredientes, definimos:

K : número de rações a serem produzidas;

n : número de ingredientes disponíveis no instante da produção;

m : número de nutrientes que serão considerados nas rações;

q_k : quantidade a ser produzida da ração k, numa unidade pre-fixada;

ℓ_j : quantidade máxima do ingrediente j a ser usada nas K rações;

g_j : quantidade mínima do ingrediente j a ser usada nas K rações;

a_{ij} : fração do nutriente i no ingrediente j;

y_{kj} : fração do ingrediente j na ração k;

α_{ki} : fração mínima do nutriente i na ração k;

β_{ki} : fração máxima do nutriente i na ração k;

γ_{kj} : fração mínima do ingrediente j na ração k, $j \in I2_k$, onde $I2_k$ é o conjunto dos ingredientes que possuem limitantes inferiores e superiores na ração k;

δ_{kj} : fração máxima do ingrediente j na ração k, $j \in I2_k$;

c_j : preço por unidade do ingrediente j.

Como anteriormente, estamos considerando para $k = 1,\dots,K$, $\alpha_{ki} \leq \beta_{ki}$, $i = 1,\dots,m$, $\gamma_{kj} \leq \delta_{kj}$, $j \in I2_k$ e $y_{kj} \geq 0$, $j \in I1_k$, onde $I1_k$ é o conjunto dos ingredientes que não possuem limitantes.

Então, o problema é:

$$\left\{\begin{array}{l}
\min \sum_{k=1}^{K} \sum_{j=1}^{n} q_k c_j y_{kj} \\
\text{s.a.} \\
\quad g_j \leq \sum_{k=1}^{K} q_k \; y_{kj} \leq \ell_j \;, \; j = 1,\dots,n \\
\quad \alpha_{ki} \leq \sum_{j=1}^{n} a_{ij} \; y_{kj} \leq \beta_{ki}, \; i=1,\dots,m, k=1,\dots,K \\
\qquad\qquad y_{kj} \geq 0, \; j \in I1_k \;, \; k=1,\dots,K \\
\qquad \gamma_{kj} \leq y_{kj} \leq \delta_{kj}, \; j \in I2_k, \; k=1,\dots,K
\end{array}\right\} \quad (1.2)$$

A estrutura do problema (1.2),por exemplo, para K = 2, m = 3 e n = 4 é a seguinte:

	y_{11}	y_{12}	y_{13}	y_{14}	y_{21}	y_{22}	y_{23}	y_{24}	
	q_1c_1	q_1c_2	q_1c_3	q_1c_4	q_2c_1	q_2c_2	q_2c_3	q_2c_4	
g_1	q_1				q_2				ℓ_1
g_2		q_1				q_2			ℓ_2
g_3			q_1				q_2		ℓ_3
g_4				q_1				q_2	ℓ_4
α_{11}	a_{11}	a_{12}	a_{13}	a_{14}					β_{11}
α_{12}	a_{21}	a_{22}	a_{23}	a_{24}					β_{12}
α_{13}	a_{31}	a_{32}	a_{33}	a_{34}					β_{13}
α_{21}					a_{11}	a_{12}	a_{13}	a_{14}	β_{21}
α_{22}					a_{21}	a_{22}	a_{23}	a_{24}	β_{22}
α_{23}					a_{31}	a_{32}	a_{33}	a_{34}	β_{23}

Nota-se, assim, que o problema (1.2) possui uma estrutura bloco-angular para a qual podemos aplicar os Métodos de Dantzig-Wolfe e de Rosen, mas introduzindo restrições e variáveis canalizadas. Além disso, as matrizes de tecnologia são idênticas em todos os blocos e os coeficientes da função objetivo também , exceto pela multiplicação pelas quantidades de cada ração, e as matrizes de acoplamento possuem uma forma muito particular - são matrizes escalares. Esta formulação nos permite considerar ainda, que as variáveis não precisam ser necessariamente canalizadas e que, se um ingrediente j não é usado numa ração k, ele pode ser considerado como uma variável fixa e igual a zero. Da mesma maneira, se um nutriente i não for considerado numa ração k, ele terá seu limite inferior igual a zero e o superior igual a infinito (basta ser um, pois $\beta_{ki} \leq 1$).

1.4 - DEFINIÇÃO DO PROBLEMA GERAL:

O problema da otimização global de rações pode ser visto como um caso particular do seguinte problema de Programação Linear com estrutura bloco-angular:

$$\left\{\begin{array}{lll} \min \sum_{k=1}^{K} c'_k \; y_k & & \\ \text{s.a.} & & \\ g \leq \sum_{k=1}^{K} Q'_k \; y_k & \leq \ell & \\ \alpha_k \leq A'_k \; y_k & \leq \beta_k & , \; k=1,\ldots,K \\ y_{kj} & \geq 0 & , \; j \in I1_k, \; k=1,\ldots,K \\ \gamma_{kj} \leq y_{kj} & \leq \delta_{kj} & , \; j \in I2_k, \; k=1,\ldots,K \end{array}\right\} \qquad (1.3)$$

onde:

y_k : vetor de dimensão $n'_k \times 1$, $k = 1,\ldots, K$, formado pelos ele -

mentos y_{kj}, da forma:

$y_k = (y_{k1} \;\; y_{k2} \;\; \cdots \;\; y_{kn'_k})^T$. Cada y_k possui $n1_k$ componentes não canalizadas (não negativas) e $n2_k$ componentes canalizadas, de modo que $n'_k = n1_k + n2_k$. Em particular, $n1_k$ ou $n2_k$ podem ser iguais a zero. Dizemos ainda que, para cada bloco k, os índices das variáveis não canalizadas deste bloco pertencem ao conjunto $I1_k$ e os índices das variáveis canalizadas pertencem ao conjunto $I2_k$.

c'_k : vetor de dimensão $1 \times n'_k$, $k = 1,\ldots,K$, formado pelos elementos c'_{kj} (coeficientes da função objetivo do bloco k), da forma:

$c'_k = (c'_{1k} \;\; c'_{2k} \;\; \cdots \;\; c'_{n'_k k})$.

Q'_k : matriz de dimensão $m_o \times n'_k$, $k = 1,\ldots,K$ (matriz de acoplamento referente ao bloco k), formada pelos elementos q'_{kij} da forma:

$$Q'_k = \begin{pmatrix} q'_{k11} & q'_{k12} & \cdots & q'_{k1n'_k} \\ q'_{k21} & q'_{k22} & \cdots & q'_{k2n'_k} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ q'_{km_o1} & q'_{km_o2} & & q'_{km_on'_k} \end{pmatrix}$$

g : vetor de dimensão $m_o \times 1$, formado pelos elementos g_i (limites inferiores das restrições de acoplamento), da forma :

$g = (g_1 \quad g_2 \ldots g_{m_o})^T$.

ℓ : vetor de dimensão $m_o \times 1$, formado pelos elementos ℓ_i (limites superiores das restrições de acoplamento), da forma :

$\ell = (\ell_1 \; \ell_2 \ldots \ell_{m_o})^T$.

A'_k : matriz de dimensão $m_k \times n'_k$, $k = 1,\ldots,K$ (matriz de tecnologia do bloco k), formada pelos elementos a'_{kij} , da forma:

$$A'_k = \begin{bmatrix} a'_{k11} & a'_{k12} & \cdots & a'_{k1n'_k} \\ a'_{k21} & a'_{k22} & \cdots & a'_{k2n'_k} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ a'_{km_k1} & a'_{km_k2} & \cdots & a'_{km_kn'_k} \end{bmatrix}$$

α_k : vetor de dimensão $m_k \times 1$, $k = 1,\ldots,K$, formado pelos elementos α_{ki} (limites inferiores das restrições dos blocos), da forma:

$\alpha_k = (\alpha_{k1} \quad \alpha_{k2} \ldots \alpha_{km_k})^T$.

β_k : vetor de dimensão $m_k \times 1$, $k = 1,\ldots,K$, formado pelos elementos β_{ki} (limites superiores das restrições dos blocos), da forma:

$\beta_k = (\beta_{k1} \ \beta_{k2} \ \cdots \ \beta_{km_k})^T.$

γ_k : vetor de dimensão $n2_k \times 1$, $k = 1, \ldots, K$, onde $n2_k$ representa o número de variáveis canalizadas do bloco k, formado pelos elementos γ_{kj} (limites inferiores das variáveis canalizadas), da forma:

$\gamma_k = (\gamma_{k1} \ \gamma_{k2} \ \cdots \ \gamma_{kn2_k})^T.$

δ_k : vetor de dimensão $n2_k \times 1$, $k = 1,\ldots,K$, formado pelos elementos δ_{ki} (limites superiores das variáveis canalizadas), da forma:

$\delta_k = (\delta_{k1} \ \delta_{k2} \ \cdots \ \delta_{kn2_k})^T.$

Embora o problema (1.3) considere variáveis canalizadas e não canalizadas, daqui por diante, para simplificarmos a notação, suporemos que todas as variáveis são canalizadas. O problema, então, será do tipo:

$$\left\{\begin{array}{l}
\min \sum_{k=1}^{K} c_k' \ y_k \\
\text{s.a.} \\
\qquad g \leq \sum_{k=1}^{K} Q_k' \ y_k \leq \ell \\
\qquad \alpha_k \leq A_k' \ y_k \leq \beta_k \ , \ k = 1,\ldots,K \\
\qquad \gamma_k \leq y_k \leq \delta_k \ , \ k = 1,\ldots,K
\end{array}\right\} \qquad (1.4)$$

onde γ_k e δ_k, $k = 1,\ldots,K$ terão dimensões $n'_k \times 1$.

Apesar de considerarmos agora o problema (1.4) em vez do problema (1.3), todos os programas que foram desenvolvidos consideram que cada y_k possui $n1_k$ componentes não canalizadas (não negativas) e $n2_k$ componentes canalizadas. Além disso, ou $n1_k$ ou $n2_k$ podem ser zero para um k específico.

1.5 - ESTRUTURAÇÃO DO PROBLEMA:

Veremos agora como transformar o problema (1.4), que possui restrições canalizadas, num problema de Programação Linear que possui apenas variáveis canalizadas. Para isto, usaremos o seguinte teorema:

Teorema 1.1 [Arenales, 1979]:

Sejam os seguintes problemas de Programação Linear:

$$\left\{\begin{array}{l} \min \quad c\,y \\ \text{s.a.} \\ \qquad \alpha \leq Ay \leq \beta \\ \qquad\qquad y \geq 0 \end{array}\right\} \quad \text{(I)}$$

$$\left\{\begin{array}{l} \min \quad cy \\ \text{s.a.} \\ \qquad Ay + u = \alpha + \beta \\ \qquad \alpha \leq u \leq \beta \\ \qquad\qquad y \geq 0 \end{array}\right\} \quad \text{(II)}$$

onde c é $1 \times n$, y é $n \times 1$, A é $m \times n$, α é $m \times 1$, β é $m \times 1$ e u é $m \times 1$. Então (I) e (II) são equivalentes.

<u>prova</u>: Sejam:

$S_1 = \{y \;/\; \alpha \leq Ay \leq \beta,\ y \geq 0\}$ e

$S_2 = \{(y,u) \;/\; Ay + u = \alpha + \beta,\ \alpha \leq u \leq \beta,\ y \geq 0\}$

Mostraremos que, se $y \in S_1$, então existe u tal que $(y,u) \in S_2$, e se $(y,u) \in S_2$, então $y \in S_1$.

(i) $(\rightarrow)$ Seja $y \in S_1$. Defina u tal que $Ay + u = \alpha + \beta$.

Então $Ay + u = \alpha + \beta \geq \alpha + Ay$, porque $y \in S_1$.

Logo, $u \geq \alpha$.

Mas, $Ay + u = \alpha + \beta \leq \beta + Ay$, porque $y \in S_1$.

Logo, $u \leq \beta$.

Então, temos que $\alpha \leq u \leq \beta$ e $Ay + u = \alpha + \beta$; portanto, $(y,u) \in S_2$.

(ii) $(\leftarrow)$ Seja $(y,u) \in S_2$; logo, $Ay + u = \alpha + \beta$ e $\alpha \leq u \leq \beta$.

Então $Ay + u = \alpha + \beta \geq \alpha + u$ e $Ay + u \geq \alpha + u \implies$ $Ay \geq \alpha$.

Mas $Ay + u = \alpha + \beta \leq u + \beta$ e $Ay + u \leq u + \beta \implies$ $Ay \leq \beta$.

Logo, $\alpha \leq Ay \leq \beta$ e, portanto, $y \in S_1$ (c.q.d.).

Aplicando agora o teorema (1.1) ao problema (1.4) temos:

$$
\left\{
\begin{array}{l}
\min \sum_{k=1}^{K} c_k' \, y_k \\
\text{s.a.} \\
\qquad \sum_{k=1}^{K} Q_k' \, y_k + u_{K+1} = g + \ell \\
\qquad A_k' \, y_k + u_k = \alpha_k + \beta_k, \; k=1,\ldots,K \\
\qquad \alpha_k \leq u_k \leq \beta_k, k= 1,\ldots, K \\
\qquad g \leq u_{K+1} \leq \ell \\
\qquad \gamma_k \leq y_k \leq \delta_k, k= 1,\ldots, K
\end{array}
\right\}
\qquad (1.5)
$$

onde:

u_k : vetor de dimensão m_k x 1, k = 1,...,K, formado pelos elementos u_{ki}, da forma:

$$u_k = (u_{k1} \;\; u_{k2} \;\; \cdots \;\; u_{km_k})^T.$$

u_{K+1}: vetor de dimensão m_o x 1, formado pelos elementos $u_{K+1,i}$, da forma:

$$u_{K+1} = (u_{K+1,1} \;\; u_{K+1,2} \;\; \cdots \;\; u_{K+1,m_o})^T$$

Sendo I_{m_o} a matriz identidade de ordem m_o e I_{m_k} a matriz identidade de ordem m_k, o problema (1.5) também pode ser escrito como:

$$\left.\begin{cases} \min \sum\limits_{k=1}^{K} (c_k' \,/\, 0) \begin{pmatrix} y_k \\ \hline u_k \end{pmatrix} \\ \text{s.a.} \\ \quad \sum\limits_{k=1}^{K} (Q_k' \,/\, 0) \begin{pmatrix} y_k \\ \hline u_k \end{pmatrix} + I_{m_o}\, u_{K+1} = g + \ell \\ \quad (A_k' \,/\, I_{m_k}) \begin{pmatrix} y_k \\ \hline u_k \end{pmatrix} = \alpha_k + \beta_k,\ k=1,\ldots,K \\ \quad \alpha_k \leq u_k \leq \beta_k,\ k = 1, \ldots, K \\ \quad g \leq u_{K+1} \leq \ell \\ \quad \gamma_k \leq y_k \leq \delta_k,\ k = 1, \ldots, K \end{cases}\right\} \quad (1.6)$$

onde:

$(c_k' \,/\, 0)$: vetor de dimensão $(n_k' + m_k) \times 1$, $k = 1, \ldots, K$. Cada um desses vetores é formado pelo vetor c_k' e pelo vetor nulo de dimensão $1 \times m_k$.

$(Q_k' \,/\, 0)$: matriz de dimensão $m_o \times (n_k' + m_k)$, $k = 1, \ldots, K$. Cada uma dessas matrizes é formada pela matriz Q_k' e pela matriz nula de dimensão $m_o \times m_k$.

(A_k'/I_{m_k}) : matriz de dimensão $m_k \times (n_k' + m_k)$, $k = 1, \ldots, K$. Cada uma dessas matrizes é formada pela matriz A_k' e pela matriz identidade de ordem m_k.

Transformamos, assim, o problema original que possuía restrições canalizadas, para um problema equivalente que possui -

apenas variáveis canalizadas. Isto nos permitirá trabalhar com subproblemas mais simples e com o Método Simplex para variáveis canalizadas.

Faremos ainda mais uma transformação: com o objetivo de facilitar a programação dos métodos que descreveremos nos próximos capítulos, faremos mudanças nos limites das variáveis canalizadas de modo a obter que todas as variáveis canalizadas tenham limites inferiores iguais a zero.

No problema (1.6), as restrições para as variáveis canalizadas são as seguintes:

$$\alpha_k \leq u_k \leq \beta_k \;,\; \forall k$$

$$g \leq u_{K+1} \leq \ell$$

$$\gamma_k \leq y_k \leq \delta_k \;,\; \forall k$$

Subtraindo α_k de todos os membros da primeira equação, obtemos:

$$0 \leq u_k - \alpha_k \leq \beta_k - \alpha_k \;,\; \forall k$$

e colocando $u'_k = u_k - \alpha_k$ obtemos $u_k = u'_k + \alpha_k$, $\forall k$.

Fazendo a mesma operação para as outras duas equações, temos:

$$u'_{K+1} = u_{K+1} - g \quad \text{e} \quad u_{K+1} = u'_{K+1} + g.$$

$$y'_k = y_k - \gamma_k \quad \text{e} \quad y_k = y'_k + \gamma_k.$$

Substituindo agora os novos valores de u_k, $\forall k$, u_{K+1} e y_k, $\forall k$, no problema (1.6), obtemos:

$$\left\{\begin{array}{l} \min \sum_{k=1}^{K} (c_k' \,/\, 0) \begin{pmatrix} y_k' + \gamma_k \\ u_k' + \alpha_k \end{pmatrix} \\ \text{s.a.} \\ \quad \sum_{k=1}^{K} (Q_k'/0) \begin{pmatrix} y_k' + \gamma_k \\ u_k' + \alpha_k \end{pmatrix} + I_{m_o} (u_{K+1}' + g) = g + \ell \\ \quad (A_k'/I_{m_k}) \begin{pmatrix} y_k' + \gamma_k \\ u_k' + \alpha_k \end{pmatrix} = \alpha_k + \beta_k, k=1,\ldots,K \\ \quad 0 \leq u_k' \leq \beta_k - \alpha_k, \; k = 1,\ldots,K \\ \quad 0 \leq u_{K+1}' \leq \ell - g \\ \quad 0 \leq y_k' \leq \delta_k - \gamma_k, \; k = 1,\ldots,K \end{array}\right.$$

ou seja:

$$\left\{\begin{array}{l} \min \sum_{k=1}^{K} (c_k'/0) \begin{pmatrix} y_k' \\ u_k' \end{pmatrix} + \sum_{k=1}^{K} (c'_k/0) \begin{pmatrix} \gamma_k \\ \alpha_k \end{pmatrix} \\ \text{s.a.} \quad \sum_{k=1}^{K} (Q_k'/0) \begin{pmatrix} y_k' \\ u_k' \end{pmatrix} + I_{m_o} u_{K+1}' = g + \ell - \sum_{k=1}^{K} (Q_k'/0) \begin{pmatrix} \gamma_k \\ \alpha_k \end{pmatrix} - I_{m_o} g \\ \quad (A_k'/I_{m_k}) \begin{pmatrix} y_k' \\ u_k' \end{pmatrix} = \alpha_k + \beta_k - (A_k'/I_{m_k}) \begin{pmatrix} \gamma_k \\ \alpha_k \end{pmatrix}, \; k=1,\ldots,K \\ \quad 0 \leq u_k' \leq \beta_k - \alpha_k, \; k = 1,\ldots,K \\ \quad 0 \leq u_{K+1}' \leq \ell - g \\ \quad 0 \leq y_k' \leq \delta_k - \gamma_k, \; k=1,\ldots,K \end{array}\right.$$

ou ainda:

$$
(1.7)\quad
\begin{cases}
\min \sum_{k=1}^{K} (c_k' / 0) \begin{pmatrix} y_k' \\ \hline u_k' \end{pmatrix} + \sum_{k=1}^{K} (c_k' / 0) \begin{pmatrix} \gamma_k \\ \hline \alpha_k \end{pmatrix} \\
\text{s.a.} \\
\sum_{k=1}^{K} (Q_k'/0) \begin{pmatrix} y_k' \\ \hline u_k' \end{pmatrix} + I_{m_o}\, u_{K+1}' = \ell - \sum_{k=1}^{K} (Q_k'/0) \begin{pmatrix} \gamma_k \\ \hline \alpha_k \end{pmatrix} \\
(A_k'/I_{m_k}) \begin{pmatrix} y_k' \\ \hline u_k' \end{pmatrix} = \beta_k - A_k' \gamma_k, \quad k=1,\dots,K \\
0 \leq u_k' \leq \beta_k - \alpha_k, \quad k = 1,\dots,K \\
0 \leq u_{K+1}' \leq \ell - g \\
0 \leq y_k' \leq \delta_k - \gamma_k, \quad k = 1,\dots,K
\end{cases}
$$

Vemos, desta forma, que o problema (1.4), que possuía restrições canalizadas e variáveis canalizadas entre quaisquer limites, pode ser transformado num problema equivalente (1.7) que possui apenas variáveis canalizadas e com limites inferiores de canalização zero, e que todas as outras restrições são de igualdade. A parcela $\sum_{k=1}^{K} (c_k' / 0) \begin{pmatrix} \gamma_k \\ \hline \alpha_k \end{pmatrix}$ que aparece na função objetivo é uma constante, e por isso pode ser eliminada da minimização. Podemos, então, redenominar as matrizes e os vetores do problema (1.7), de modo a tratá-lo por uma forma mais simplificada.

Sejam:

$$c_k = (c_k' \;/\; 0) \;,\; k = 1,\dots,K \;;$$

$$x_k = (y_k' \;/\; u_k')^T \;,\; k = 1,\dots,K;$$

$$Q_k = (Q_k' \;/\; 0),\; k = 1,\dots,K;$$

$$b_o = \ell - \sum_{k=1}^{K} (Q_k' \;/\; 0) \begin{pmatrix}\gamma_k \\ \alpha_k\end{pmatrix} \;;$$

$$A_k = (A_k' \;/\; I_{m_k}) \;,\; k = 1,\dots,K \;;$$

$$b_k = \beta_k - A_k' \gamma_k,\quad k = 1,\dots,K \;;$$

$$h_k = (\delta_k - \gamma_k \;/\; \beta_k - \alpha_k)^T \;,\; k = 1,\dots,K.$$

Estamos desprezando a parte referente ao bloco $K + 1$ (u_{K+1}'), que é um bloco muito particular e que será tratado de uma maneira específica nos casos em que for considerado. Então, sem perda de generalidade, uma forma simplificada do problema (1.7) é:

$$\left\{\begin{array}{l}
\min\ z = \displaystyle\sum_{k=1}^{K} c_k\, x_k \\
\text{s.a.} \\
\qquad \displaystyle\sum_{k=1}^{K} Q_k x_k = b_o \\
\qquad\qquad A_k\, x_k = b_k,\; k = 1,\dots,K \\
\qquad 0 \leqslant x_k \leqslant h_k,\; k = 1,\dots,K
\end{array}\right\} \qquad (1.8)$$

onde, $k = 1,\ldots,K$, temos:

c_k : $1 \times n_k$ e de componentes c_{kj};

x_k : $n_k \times 1$ e de componentes x_{kj};

Q_k : $m_o \times m_k$ e de componentes q_{kij};

A_k : $m_k \times n_k$ e de componentes a_{kij};

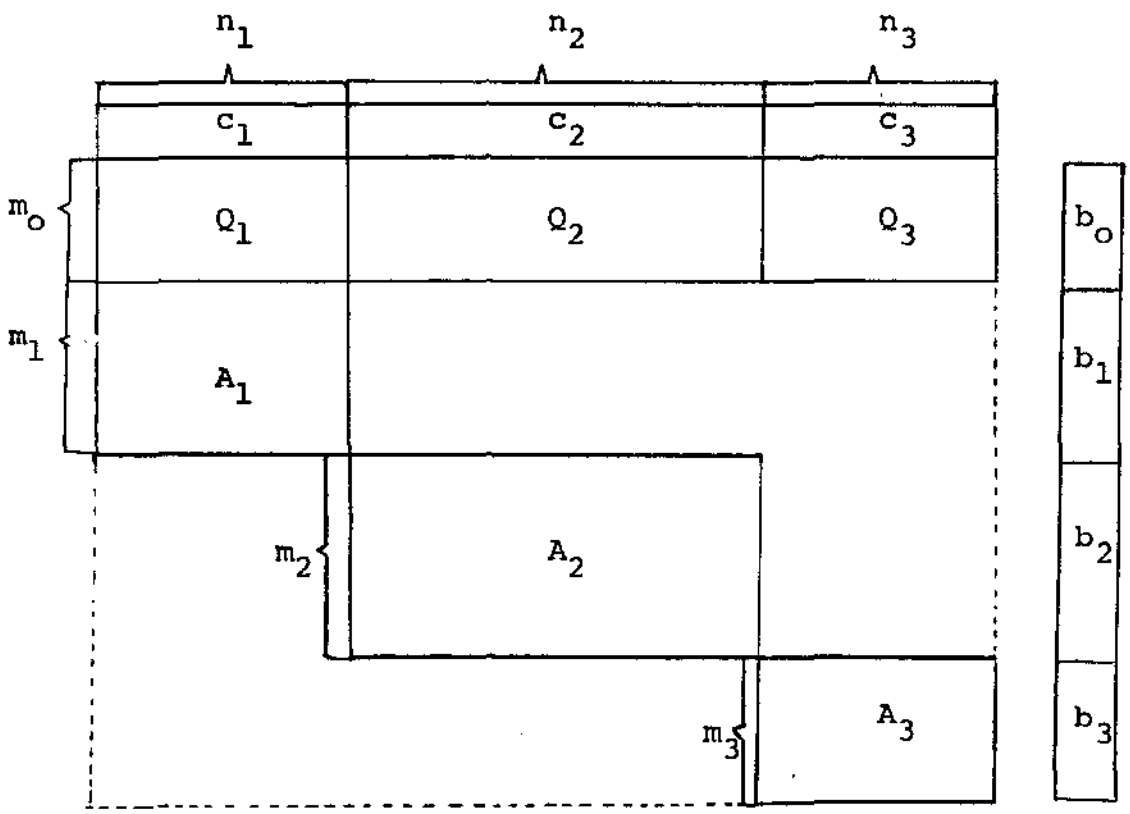
b_k : $m_k \times 1$ e de componentes b_{ki};

h_k : $n_k \times 1$ e de componentes h_{kj};

$m_k < n_k$ e

b_o : $m_o \times 1$ e de componentes b_{o_i} ,

que é um problema de Programação Linear com estrutura bloco - angular e variáveis canalizadas, que, por exemplo, para $K = 3$, tem a seguinte estrutura:



Aos K conjuntos de restrições independentes, dados pelas matrizes A_k, chamaremos de <u>blocos</u>. Chamaremos de restrições de <u>acoplamento</u> às restrições dadas pelas matrizes Q_k, $\forall k$.

A própria estrutura do problema (1.8) nos levou a estudar os métodos de decomposição de Dantzig-Wolfe e de Rosen, com a introdução de variáveis canalizadas. Estes métodos serão mostrados nos capítulos subsequentes, e veremos que a inclusão de variáveis canalizadas trará algumas novidades no Método de Rosen.

O problema geral que iremos tratar é o problema (1.8):

$$\begin{cases} \min z = \sum_{k=1}^{K} c_k x_k \\ \text{s.a.} \\ \qquad \sum_{k=1}^{K} Q_k x_k = b_o \\ \qquad A_k x_k = b_k, \; k = 1,\ldots,K \\ \qquad 0 \leq x_k \leq h_k \;, \; k = 1,\ldots,K \end{cases}$$

Durante as explicações dos métodos, faremos, algumas vezes, menção ao problema da otimização global de rações. Neste caso, estaremos nos referindo a um problema do tipo:

$$
\left\{
\begin{array}{l}
\min \sum_{k=1}^{K} (q_k c/0) \begin{pmatrix} y'_k \\ u'_k \end{pmatrix} + \sum_{k=1}^{K} (q_k \; c/0) \begin{pmatrix} \gamma_k \\ \alpha_k \end{pmatrix} \\
\text{s.a.} \\
\quad \sum_{k=1}^{K} (Q_k \,/\, 0) \begin{pmatrix} y'_k \\ u'_k \end{pmatrix} + I \; u'_{K+1} = \ell - \sum_{k=1}^{K} (Q_k/0) \begin{pmatrix} \gamma_k \\ \alpha_k \end{pmatrix} = b'_o \\
\qquad (A/I) \begin{pmatrix} y'_k \\ u'_k \end{pmatrix} \qquad\qquad = \beta_k - A\, \gamma_k = b'_k \; , \; k=1,\dots,K \\
\qquad 0 \leq u'_k \leq \beta_k - \alpha_k , \; k = 1,\dots,K \\
\qquad 0 \leq u'_{K+1} \leq \ell - g \\
\qquad 0 \leq y'_k \leq \delta_k - \gamma_k \; , \; k = 1,\dots,K
\end{array}
\right\} \qquad (1.9)
$$

que é o problema (1.2) já com as transformações de variáveis e com a eliminação de restrições canalizadas, mas onde estamos considerando, para simplificar esta apresentação, que todas as variá veis são canalizadas.

A justificativa para tratarmos separadamente o problema acima, é que ele é um caso especial onde $c_1 = c_2 = \dots = c_K = c$, $A_1 = A_2 = \dots = A_K = A$ e as matrizes de acoplamento são do tipo:

$$
Q_k = \begin{bmatrix} q_k & & & & \\ & q_k & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \cdot \\ & & & & & q_k \end{bmatrix} = q_k \cdot I_n ,
$$

onde I_n é a matriz identidade de ordem n. Estas particularidades facilitam as operações dos métodos que serão apresentados e foram também exploradas na confecção dos programas. Além disso, neste problema aparece o bloco (K+1) que será sempre tratado à parte , por possuir uma estrutura bem mais simples que os demais.

CAPÍTULO 2

APLICAÇÃO DO MÉTODO DA DECOMPOSIÇÃO DE DANTZIG - WOLFE

2.1 - INTRODUÇÃO:

O princípio da decomposição de Dantzig - Wolfe se baseia na formação de um problema mestre com poucas linhas a mais que o número de restrições de acoplamento, mas com um número muito grande de colunas. Como o número de colunas é muito elevado, este princípio se vale da técnica da geração de colunas, isto é, as colunas vão sendo construídas apenas quando necessário.

O Método envolve interações entre um conjunto de sub - problemas independentes e o problema mestre. Os subproblemas recebem um conjunto de parâmetros (multiplicadores Simplex) do problema mestre, buscam seus ótimos individualmente num nível inferior e, em função de suas soluções, o problema mestre fixa novos parâmetros para os subproblemas, até que o ótimo global seja encontrado.

O desenvolvimento do princípio da decomposição de Dantzig - Wolfe depende, basicamente, de dois resultados: o primeiro é um teorema que garante que, se um ponto está num poliedro convexo fechado e limitado, ele pode ser escrito como uma combinação convexa dos pontos extremos deste poliedro. O segundo, é a geração de colunas.

Observação: usaremos, neste capítulo e nos subsequentes, os conceitos usualmente empregados em Programação Linear tais como conjuntos convexos, base, solução básica factível, matriz associada à solução básica corrente, multiplicadores Simplex, bem como as propriedades do Método Simplex. Além disso, estamos abusando um pouco da linguagem, usando, por exemplo, "B" para denotar tanto um conjunto de índices quanto uma matriz, cujas colunas correspondem a estes índices.

2.2 - O TEOREMA DAS COMBINAÇÕES CONVEXAS:

Definição 2.1: envólucro convexo: o envólucro convexo de um conjunto Y é a intersecção de todos os conjuntos convexos que contêm Y, isto é, é o menor conjunto convexo que contém Y.

Teorema 2.1: o envólucro convexo de um conjunto finito de pontos $X = \{x^1, x^2, \ldots, x^p\}$ é o conjunto de todas as combinações convexas de $\{x^1, x^2, \ldots, x^p\}$, isto é,

$$C(X) = \{x / x = \sum_{j=1}^{p} \lambda^j x^j \ , \ \sum_{j=1}^{p} \lambda^j = 1 \ \text{ e } \ \lambda^j \geq 0, \ j = 1, \ldots, p\} \ .$$

A prova deste teorema pode ser encontrada em Hadley [Hadley 1, 1969, pp. 208].

Definição 2.2: Hiperplano: seja c um vetor linha n-dimensional não nulo e seja μ um número real. O conjunto

$$H = \{x \in \mathbb{R}^n \ / \ cx = \mu\}$$

é um hiperplano em $\mathbb{R}^n$.

Definição 2.3: Hiperplano suporte: um hiperplano que contém um conjunto convexo X em um dos seus semi-espaços fechados, e que contém um ponto da fronteira de X é chamado de hiper -

plano suporte de X.

Definição 2.4: ponto extremo: um ponto x de um conjunto convexo X é um ponto extremo de X se não existem dois pontos distintos x_1 e x_2 em X tais que

$$x = \lambda x_1 + (1 - \lambda)\, x_2 \;, \quad 0 < \lambda < 1.$$

Teorema 2.2: se x é um ponto fronteira de um conjunto fechado e convexo, então existe pelo menos um hiperplano suporte em x.

A prova deste teorema pode ser encontrada em Hadley [Hadley 1, 1969, pp. 212 - 214].

Teorema 2.3: teorema das combinações convexas:

Seja X um conjunto compacto e convexo, com p pontos extremos; sejam $\mathcal{E}(X)$ o conjunto de seus pontos extremos e $C(X)$ o envólucro convexo de $\mathcal{E}(X)$ (simplificando a notação $C(\mathcal{E}(X))$) . Então, qualquer ponto de X pode ser escrito como uma combinação convexa de seus pontos extremos, isto é, $C(X) = X$.

prova: queremos mostrar que $C(X) = X$, isto é, $X \supseteq C(X)$ e $C(X) \supseteq X$.

(i) Vamos mostrar que $X \supseteq C(X)$. Seja $y \in C(X)$; então podemos escrever (pelo teorema 2.1):

$$y = \sum_{j=1}^{p} \lambda^j \hat{x}^j, \quad \sum_{j=1}^{p} \lambda^j = 1, \quad \lambda^j \geq 0,\; j = 1,\ldots,p,$$

com $\hat{x}^j \in \mathcal{E}(X)$.

Então, com $\hat{x}^j$ também $\in X$ e X é convexo, temos que $y \in X$, isto é, $X \supseteq C(X)$.

(ii) Vamos mostrar que $C(X) \supseteq X$. Seja $x^* \in X$, mas vamos supor que $x^* \notin C(X)$. Então x^* pode ser separado de $C(X)$ por um hiperplano, isto é, existe (c, μ) tal que:

$$cx^* = \mu$$

$$cy \leq \mu, \ \forall y \in C(X) \tag{2.1}$$

Seja $\mu^\cdot = \max \{cx \ / \ x \in X\}$. O número $\mu^\cdot$ existe porque X é compacto. Então $(c, \mu^\cdot)$ define um hiperplano suporte de X, desde que

$$cx \leq \mu^\cdot \ , \ \forall x \in X.$$

Este hiperplano deve conter um ponto extremo de X, $\hat{x}^j$; então, $c\hat{x}^j = \mu^\cdot$.

Mas isto contradiz o fato de que todos os pontos de $C(X)$ (e logo, todos os pontos extremos de X) satisfaçam (2.1), desde que $\mu^\cdot > \mu$. Então, $\forall \ x^* \in X \Rightarrow x^* \in C(X)$.

<u>Nota</u>: como estaremos tratando de conjuntos da forma $X = \{x \ / \ Ax = b \ \text{ e } \ 0 \leq x \leq h\}$, suporemos sempre conjuntos limitados. Por esta razão, o teorema (2.3) foi apresentado considerando apenas conjuntos limitados.

2.3 - <u>A FORMAÇÃO DO PROBLEMA MESTRE</u>:

O problema (1.8) possui K blocos independentes que são:

$$\left\{\begin{bmatrix} A_k \ x_k = b_k \\ 0 \leq x_k \leq h_k \end{bmatrix}\right\} \quad k = 1,\ldots,K \qquad (2.2)$$

Da Programação Linear, sabemos que cada conjunto (2.2) possui um número finito de pontos extremos ([Luenberger, 1973] , [Bazaraa , 1977]). Sejam $\hat{x}_k^j$, $j = 1,\ldots,p_k$, os pontos extremos do k- ésimo conjunto (2.2). Supondo que cada um destes conjuntos seja limitado, $\forall k$, então pelo teorema (2.3), temos que qualquer ponto x_k de (2.2) pode ser escrito como uma combinação convexa de seus pontos extremos, isto é, para $k = 1,\ldots,K$, temos:

$$x_k = \sum_{j=1}^{p_k} \lambda_k^j \ \hat{x}_k^j \ , \ \sum_{j=1}^{p_k} \lambda_k^j = 1 \quad e \ \lambda_k^j \geq 0, \ j=1,\ldots p_k \qquad (2.3)$$

Substituindo a equação (2.3) na função objetivo e nas restrições de acoplamento do problema (1.8), obtemos um problema equivalente que é chamado de <u>problema mestre</u>:

$$\left\{\begin{bmatrix}
\min z = \sum_{j=1}^{p_1} (c_1\hat{x}_1^j)\ \lambda_1^j + \sum_{j=1}^{p_2} (c_2\hat{x}_2^j)\ \lambda_2^j + \ldots + \sum_{j=1}^{p_K} (c_K\hat{x}_K^j)\lambda_K^j \\
\text{s.a.} \\
\sum_{j=1}^{p_1} (Q_1\hat{x}_1^j)\lambda_1^j + \sum_{j=1}^{p_2} (Q_2\hat{x}_2^j)\ \lambda_2^j + \ldots + \sum_{j=1}^{p_K} (Q_K\hat{x}_K^j)\lambda_K^j = b_o \\
\sum_{j=1}^{p_1} \lambda_1^j \hspace{6cm} = 1 \\
\hspace{2cm} \sum_{j=1}^{p_2} \lambda_2^j \hspace{4cm} = 1 \\
\hspace{3cm} \ddots \hspace{3cm} \vdots \\
\hspace{5cm} \sum_{j=1}^{p_K} \lambda_K^j \hspace{1cm} = 1 \\
\lambda_k^j \geq 0, \ \forall k \ , \ \forall j
\end{bmatrix}\right\} \qquad (2.4)$$

Neste problema, as novas variáveis são λ_k^j que, para $\forall k$, $\forall j$, não são canalizadas. Como o problema (2.4) é equivalente ao problema (1.8), uma solução factível para o problema (2.4) nas variáveis λ_k^j corresponde a uma solução factível para o problema (1.8) nas variáveis x_k. Em particular, a solução ótima do problema (2.4) corresponderá à solução ótima do problema (1.8), isto é, se encontrarmos os valores ótimos de λ_k^j, $\forall k$, $\forall j$, em (2.4), a solução ótima do problema (1.8) será:

$$x_k = \sum_{j=1}^{p_k} \lambda_k^j \, \hat{x}_k^j \ , \ k = 1, \ldots, K \quad .$$

Além disso, enquanto o problema (1.8) possui $m_o + \sum_{k=1}^{K} m_k$ restrições, o problema (2.4) tem seu número de restrições diminuído para $m_o + K$. Entretanto, o número de variáveis aumenta bastante, dependendo do número total de pontos extremos dos blocos. Contudo, como o Método Simplex é sensível ao número de restrições, esperamos que o problema (2.4) seja mais vantajoso que o problema (1.8).

Veremos ainda que os pontos extremos $\hat{x}_k^j$ não precisam ser conhecidos "a priori". Na verdade, dada uma solução básica inicial factível para o problema mestre, cada iteração irá determinar os pontos extremos.

2.4 - COMO PREPARAR UMA ITERAÇÃO:

Sejam:

w : vetor linha que contém os coeficientes da função objetivo do problema mestre, cujas componentes são dadas por $w_k^j = c_k . \hat{x}_k^j$.

B : matriz básica associada à solução básica factível corrente do problema mestre, de dimensão $(m_o+K) \times (m_o+K)$, cuja inversa é B^{-1}.

w^B : vetor linha que contém os coeficientes da função objetivo do problema mestre associados à matriz B, com $(m_o + K)$ componentes.

R_k^j : j-ésima coluna da matriz tecnológica do problema mestre, correspondente ao bloco k, com $(m_o + K)$ componentes, da forma:

$$R_k^j = \left(\begin{array}{c} Q_k \; \hat{x}_k^j \\ \hline 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{array} \right) \begin{array}{l} \} \; m_o \text{ linhas} \\ \\ \\ \\ \left.\begin{array}{l} \\ \\ \\ \\ \\ \\ \end{array}\right\} \text{ K linhas} \\ \longrightarrow \text{posição } (m_o + k) \end{array}$$

π : vetor multiplicador relativo à base corrente do problema mestre, de dimensão $1 \times (m_o + K)$, onde $\pi = w^B . B^{-1}$. Vamos particionar π da seguinte maneira:

$\pi = (\pi_o / \pi_1 \; \pi_2 \; \ldots \; \pi_K)$, onde:

π_o: vetor de dimensão $1 \times m_o$, que são as primeiras m_o posições de π, referentes às primeiras m_o restrições do problema mestre;

π_k, $k = 1,\ldots,K$: são as componentes restantes do vetor π (são escalares), correspondentes às K restrições restantes do problema mestre.

$\hat{w}$: vetor linha que contém os custos relativos do problema mestre, com componentes $\hat{w}_k^j$.

Vamos calcular, agora, os coeficientes $\hat{w}_k^j$ de custos relativos para o problema mestre. Sabemos que:

$$\hat{w}_k^j = w_k^j - \pi\, R_k^j \;.$$

Substituindo os valores w_k^j e R_k^j dados por:

$$w_k^j = c_k\, \hat{x}_k^j \quad \text{e} \quad R_k^j = \begin{bmatrix} Q_k\, \hat{x}_k^j \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}$$

e colocando a partição feita no vetor π, temos:

$$\hat{w}_k^j = c_k\, \hat{x}_k^j - (\pi_o/\pi_1 \ldots \pi_k \ldots \pi_K) \begin{bmatrix} Q_k\, \hat{x}_k^j \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ \leftarrow \text{posição } (m_o+k) \\ \\ \\ \\ \\ \end{matrix}$$

$$= c_k \hat{x}_k^j - \pi_o Q_k\, \hat{x}_k^j - \pi_k \;.$$

Colocando $\hat{x}_k^j$ em evidência, temos:

$$\hat{w}_k^j = (c_k - \pi_o Q_k)\ \hat{x}_k^j - \pi_k \quad .$$

Queremos encontrar uma variável para entrar na base do problema mestre. Sabemos, pelo Método Simplex, que o coeficiente que nos indica a coluna s que entrará na base é dado, usualmente, por:

$$\min_{k,j} \{\hat{w}_k^j\}$$

Mas, neste caso, como possuímos um número muito grande de colunas, não vamos comparar todos os coeficientes de custos relativos. O que vamos fazer é gerar as colunas apenas quando necessário - geração de colunas.

2.4.1 - Geração de colunas:

Veremos, agora, como escolher a coluna s para entrar na base do problema mestre sem compararmos todos os coeficientes de custos relativos. Ao invés de calcularmos todos os $\hat{w}_k^j$ para depois escolhermos o menor deles, vamos procurar um ponto extremo $\hat{x}_k^j$ que minimize $\hat{w}_k^j$. Este ponto extremo $\hat{x}_k^j$ é chamado de gerador de coluna. Então, em vez de procurarmos todos os pontos extremos para os K blocos, procuramos apenas um ponto extremo $\hat{x}_k^j$ em cada bloco, o qual produz o menor coeficiente $\hat{w}_k^j$, isto é, resolvemos os K subproblemas:

$$\left\{\begin{array}{l} \min \hat{w}_k^j = \min\ (c_k - \pi_o\ Q_k)\ x_k - \pi_k \\ \text{s.a.} \\ \qquad A_k\ x_k = b_k \\ \qquad 0 \leq x_k \leq h_k \end{array}\right\} \qquad (2.5)$$

e como π_k é constante para cada subproblema k, k = 1,...,K, (2.5) se reduz a:

$$\left\{\begin{array}{l} \min f_k = (c_k - \pi_o Q_k) x_k \\ \text{s.a.} \\ \quad A_k x_k = b_k \\ \quad 0 \leq x_k \leq h_k \end{array}\right\} \qquad (2.6)$$

Resolvemos os subproblemas (2.6), para cada bloco k, pelo Método Simplex Revisado com variáveis canalizadas, obtendo então uma solução ótima $\hat{x}_k^j$ que é um ponto extremo do bloco k na iteração j, ao qual corresponde um valor ótimo para a função objetivo, $\hat{f}_k^j$.

Calculamos, então, para cada k:

$$\hat{w}_k^j = \hat{f}_k^j - \pi_k \ , \ k = 1,\ldots,K.$$

Se $\hat{w}_k^j \geq 0$, $\forall k$, temos a solução ótima do problema mestre e, consequentemente, a solução ótima do problema original. Senão, tomamos o menor coeficiente $\hat{w}_k^j < 0$, isto é, fazemos

$$\hat{w}_s = \min_{k,j} \{\hat{w}_k^j\} \quad .$$

Sendo s o número do subproblema escolhido, a variável λ_s^j correspondente ao gerador $\hat{x}_s^j$ (solução do subproblema s na iteraçao j) entrará na base do problema mestre.

A variável λ_s^j que entra na base do problema mestre, leva consigo a coluna R_s^j , que é dada por:

$$R_s^j = \left[\begin{array}{c} Q_s \, \hat{x}_s^j \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{array}\right] \begin{array}{l} \\ \\ \\ \longrightarrow \text{posição } (m_o + s) \\ \\ \\ \end{array}$$

A escolha da variável que sai da base do problema mestre é feita pelo critério da razão usual, sobre a coluna R_s^j. A nova base é encontrada usando o Método Simplex Revisado.

<u>Observações</u>:

1) A única alteração que ocorre no Método de Dantzig - Wolfe, pelo fato de estarmos considerando o problema original com variáveis canalizadas (x_k), aparece durante a resolução dos subproblemas, onde utilizamos o Método Simplex com variáveis canalizadas. O problema mestre não se altera, já que suas variáveis (λ_k^j) não são canalizadas.

2) Em geral, aplicamos a fase I do Método Simplex ao problema - mestre para a obtenção de uma base inicial factível. Se chamarmos de w' ao vetor linha que contém os coeficientes da função objetivo artificial, temos que $w' = (00...01...1)$, onde os $(m_o + K)$ coeficientes de w' iguais a 1 correspondem às $m_o + K$ variáveis artificiais. Notemos que, durante a fase I do problema mestre, teremos como objetivo dos subproblemas minimizar

$$f_k = (w_k' - \pi_o Q_k) x_k \quad , \quad k = 1,\ldots,K,$$

onde w'_k, para cada $k = 1,\ldots, K$, é um vetor de dimensão $1 \times n_k$ formado por zeros, e cada w'_k é uma partição do vetor w' correspondente a cada um dos subproblemas. Assim, os objetivos dos subproblemas (2.6), durante a fase I do problema mestre serão minimizar

$$f_k = (-\pi_o Q_k) x_k \ , \ k = 1,\ldots,K.$$

3) O critério de entrada na base do problema mestre é feito pela escolha de um coeficiente $\hat{w}_k^j$ que seja < 0. Por isso não precisamos, em princípio, resolver todos os K subproblemas (2.6) para então encontrarmos o menor $\hat{w}_k^j$ - bastaria encontrarmos um ponto extremo $\hat{x}_k^j$ (solução de um subproblema k na iteração j) tal que o coeficiente $\hat{w}_k^j$ correspondente fosse < 0.

4) Na resolução dos subproblemas (2.6), pelo Método Simplex Revisado com variáveis canalizadas, aplicamos a fase I do Método Simplex apenas na primeira iteração pois, em geral, não possuímos uma base inicial factível. Nas iterações subsequentes, podemos sempre resolver cada subproblema (2.6) a partir da solução ótima obtida na iteração anterior já que, de uma iteração para a seguinte, apenas são mudados os coeficientes da função objetivo de cada subproblema.

5) Se algum dos subproblemas (2.6) for infactível, o problema original não terá solução.

6) Durante as fases I e II do problema mestre, em cada iteração, ao encontrarmos uma variável λ_s^j para entrar na base, devemos calcular o seu coeficiente na função objetivo, que é dado por

$c_s\hat{x}_s^j$, onde s é o número do subproblema escolhido nesta iteração e $\hat{x}_s^j$ é a solução ótima encontrada na resolução do s-ésimo subproblema na iteração j.

2.5 - ALGORÍTMO:

(0) Transformar o problema original no problema mestre;

Fase I:

(I.1) montar a base inicial do problema mestre com variáveis artificiais; fazer $B^{-1} = I$; montar o vetor w' com os coeficientes da função objetivo artificial;

(I.2) calcular $\pi = w'^B B^{-1}$, onde w'^B é o vetor que contém os coeficientes da função objetivo artificial associados à matriz B; particionar π em $(\pi_o/\pi_1 \ldots \pi_K)$;

Critério de entrada - resolução dos subproblemas:

(I.3) calcular $-\pi_o Q_k$, $k = 1,\ldots,K$;

(I.4) resolver os K subproblemas (2.6):

$$\left\{\begin{array}{l} \min \quad f_k = (-\pi_o Q_k)\, x_k \\ \text{s.a.} \\ \qquad A_k x_k = b_k \\ \qquad 0 \leq x_k \leq h_k \end{array}\right\}$$

pelo Método Simplex Revisado com variáveis canalizadas, obtendo como solução ótima um ponto extremo $\hat{x}_k^j$, ao qual corresponde o valor $\hat{f}_k^j$ para a função objetivo;

(I.5) calcular $\hat{w}_k^j = \hat{f}_k^j - \pi_k$, $k = 1,\ldots,K$ e $\hat{w}_s = \min\limits_{k,j} \{\hat{w}_k^j\}$.

Se $\hat{w}_s \geq 0$, chegamos ao fim da fase I. Seja Ψ o valor da função objetivo artificial. Três casos podem ocorrer:

1º) se $|\Psi| \leq \epsilon$, onde ϵ é uma tolerância dada, e a base final obtida na fase I não contiver nenhuma variável artificial, passamos à fase II.

2º) se $|\Psi| \leq \epsilon$ e a base final obtida na fase I contiver alguma variável artificial, passamos à fase II, mas tentaremos escolher, em primeiro lugar, como variáveis de saída nas próximas iterações, as variáveis artificiais que ainda estão na base.

3º) se $|\Psi| > \epsilon$, o problema mestre não tem solução factível; logo, o problema original também não terá solução. Fim.

Se $\hat{w}_s < 0$, a variável λ_s^j correspondente a $\hat{w}_s$ entrará na base, onde s é o número do subproblema escolhido;

<u>Critério de saída:</u>

(I.6) montar a coluna $R_s^j = \begin{pmatrix} Q_s \hat{x}_s^j \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix}$

onde $\hat{x}_s^j$ é a solução ótima encontrada pelo subproblema s na iteração j;

(I.7) atualizar R_s^j , isto é, obter $\bar{R}_s^j = B^{-1} R_s^j$; atualizar b, lado direito do problema mestre, $b = (b_o\ 1\ 1 \ldots 1)^T$, obtendo $\hat{b} = B^{-1} b$;

(I.8) encontrar $\min_i \{\hat{b}_i / \hat{R}_{s_i}^j , \forall i / \hat{R}_{s_i}^j > 0\}$, onde $\hat{R}_{s_i}^j$ é o elemento i da coluna R_s^j. Seja r o índice correspondente a este mínimo;

(I.9) pivotear sobre o elemento $\hat{R}_{s_r}^j$ atualizando B^{-1} pelo Simplex Revisado; atualizar os índices das variáveis básicas;

(I.10) guardar o coeficiente da variável λ_s^j (que entrou na base do problema mestre) na função objetivo, isto é, $w_s^j = c_s\ \hat{x}_s^j$ e o ponto extremo $\hat{x}_s^j$ correspondente.

Voltar ao passo (I.2).

<u>Fase II</u> :

(II.1) Calcular $\pi = w^B B^{-1}$, onde w^B é o vetor que contém os coeficientes da função objetivo associados à matriz B. Note que, cada componente i de w^B, ou contém o valor $c_k \cdot \hat{x}_k^j$ (já calculado durante a fase I), se a i-ésima variável básica é λ_k^j , ou infinito, se a i-ésima variável básica é artificial. Particionar π em $(\pi_o/\pi_1 \ldots \pi_K)$;

Critério de entrada - resolução dos subproblemas:

(II.2) calcular $(c_k - \pi_o Q_k)$, $k = 1,\ldots,K$;

(II.3) resolver os K subproblemas (2.6):

$$\begin{cases} \min f_k = (c_k - \pi_o Q_k)\, x_k \\ \text{s.a.} \\ \quad A_k x_k = b_k \\ \quad 0 \leq x_k \leq h_k \end{cases}$$

obtendo $\hat{x}_k^j$ e $\hat{f}_k^j$;

(II.4) calcular $\hat{w}_k^j = \hat{f}_k^j - \pi_k$, $k = 1,\ldots,K$ e $\hat{w}_s = \min\limits_{k,j} \{\hat{w}_k^j\}$.

Se $\hat{w}_s \geq 0$, estamos na solução ótima do problema mestre e, logo, a solução ótima do problema original será:

$$x_k = \sum_{j\text{básico}} \lambda_k^j \hat{x}_k^j \ , \ k = 1,\ldots,K$$

onde, para cada k, as variáveis λ_k^j consideradas são aquelas que pertencem à última base do problema mestre (que é a base ótima) e $\hat{x}_k^j$ são os pontos extremos correspondentes. Fim.

Se $\hat{w}_s < 0$, a variável λ_s^j correspondente a $\hat{w}_s$ entrará na base, onde s é o número do subproblema escolhido;

<u>Critério de saída</u>:

(II.5) montar a coluna $R_s^j = \begin{bmatrix} Q_s \hat{x}_s^j \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$

onde $\hat{x}_s^j$ é a solução ótima encontrada pelo subproblema s;

(II.6) atualizar R_s^j e b, isto é, obter $\hat{R}_s^j = B^{-1} R_s^j$ e $\hat{b} = B^{-1}b$;

(II.7) se existir alguma variável artificial r na base com $\hat{R}_{s_r}^j > 0$, esta é a variável escolhida para sair da base $(\hat{b}_r / \hat{R}_{s_r}^j = 0)$. Ir para o passo (II.9). Senão ,

(II.8) encontrar $\min\limits_i \{\hat{b}_i / \hat{R}_{s_i}^j \;, \forall i / \hat{R}_{s_i}^j > 0 \}$. Se este mínimo não existir, o problema mestre (e logo, o problema original) não tem solução limitada. Fim.

Senão, seja r o índice correspondente a este mínimo ;

(II.9) pivotear sobre o elemento $\hat{R}_{s_r}^j$ atualizando B^{-1} pelo Simplex Revisado; atualizar os índices das variáveis básicas;

(II.10) guardar o coeficiente da variável λ_s^j (que entrou na base do problema mestre) na função objetivo, isto é ,

$w_s^j = c_s \hat{x}_s^j$ e o ponto extremo $\hat{x}_s^j$ correspondente .

Voltar ao passo (II.1).

2.6 - A FORMAÇÃO DO PROBLEMA MESTRE RESTRITO:

Da maneira como foi aqui descrito, o Método da decomposição de Dantzig - Wolfe executa otimização nos subproblemas e apenas um pivotamento no problema mestre. Uma idéia para se chegar mais rapidamente ao ótimo, é a de construir um problema mestre que não só leva em conta a coluna gerada por um subproblema s, mas também as outras colunas geradas por cada subproblema. A este problema mestre, formado pelas colunas que estão presentemente na base e mais K colunas geradas pelos K subproblemas, damos o nome de <u>problema mestre restrito</u>. Convém notar que, como o problema mestre restrito é formado por várias colunas, podemos efetuar mais que um pivotamento por iteração neste problema. Assim, em cada iteração, resolvemos o problema mestre restrito, isto é, resolvemos um problema de Programação Linear.

Sejam:

R_k^B : colunas que estão presentemente na base do problema mestre restrito (são m_o + K colunas);

w_k^B : coeficientes da função objetivo do problema mestre restrito associados às colunas R_k^B ;

R_k^{*j} : colunas geradas pela última solução de cada subproblema (são K colunas);

w_k^{*j} : coeficientes da função objetivo do problema mestre restrito associados às colunas R_k^{*j} ;

b : lado direito do problema mestre restrito, $b=(b_o \; 1 \ldots 1)^T$.

O problema mestre restrito é então:

$$
\left\{
\begin{array}{l}
\min \displaystyle\sum_{k=1}^{K} \sum_{i \text{ básico}} w_k^i \lambda_k^i + \sum_{k=1}^{K} w_k^{*j} \lambda_k^{*j} \\
\text{s.a.} \\
\underbrace{\displaystyle\sum_{k=1}^{K} \sum_{i \text{ básico}} R_k^i \lambda_k^i}_{m_o+K \text{ componentes}} + \underbrace{\displaystyle\sum_{k=1}^{K} R_k^{*j} \lambda_k^{*j}}_{K \text{ componentes}} = b \\
\lambda_k^i \; , \; \lambda_k^{*j} \geq 0 \; , \; \forall i \; , j, \; k
\end{array}
\right\} \qquad (2.7)
$$

Observações:

1) Em vez de formarmos o problema mestre restrito com K colunas novas (cada uma gerada por um subproblema), poderíamos também escolher apenas algumas destas colunas, isto é, o problema mestre restrito poderia ser formado com as colunas que já estavam na base e mais K^* colunas, onde $1 \leq K^* \leq K$.

2) Usando o problema mestre restrito (2.7), alguns passos do algorítmo descrito anteriormente ficam alterados:

1º) No passo (I.5) (passo (II.4)) não precisamos calcular $\hat{w}_s$. Calculamos os coeficientes $\hat{w}_k^j$ para todo k e, se $\hat{w}_k^j \geq 0$ para $\forall k$, estamos no fim da fase I (na solução ótima).

2º) No passo (I.6) (passo (II.5)) montamos K colunas ao invés de uma:

$$R_k = \begin{bmatrix} Q_k \hat{x}_k^j \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} , \quad k = 1,\ldots,K,$$

onde $\hat{x}_k^j$ é a última solução ótima obtida por cada subproblema k, $k = 1,\ldots,K$.

3º) Os passos (I.7), (I.8) e (I.9) (passos (II.6), (II.7), (II.8) e (II.9)) são agora substituídos pela resolução do problema mestre restrito, pelo Método Simplex Revisado (o passo (II.7) deve estar incluído no algorítmo Simplex). Se o problema mestre restrito for ilimitado, o problema original também o será.

4º) No passo (I.10) (passo (II.10)) guardamos os coeficientes das K variáveis λ_k^j na função objetivo, ao invés de guardarmos apenas um.

2.7 - SOBRE A CONVERGÊNCIA DO MÉTODO DA DECOMPOSIÇÃO DE DANTZIG - WOLFE:

Vimos que o Método da decomposição de Dantzig- Wolfe se baseia na formação de um problema mestre, sobre o qual se aplica o Método Simplex com uma alteração no critério de entrada na base, que é feito, agora, gerando colunas. Sabemos que o número de colunas geradas é finito, já que cada coluna é gerada por um ponto extremo de um bloco e o número de pontos extremos de cada bloco é finito. Conclui-se daí, pela teoria geral da Programação Linear, que o algorítmo converge num número finito de iterações.

2.8 - O CASO PARTICULAR DO PROBLEMA DA OTIMIZAÇÃO GLOBAL DE RAÇÕES:

No capítulo 1, vimos que o problema da otimização global de rações pode ser transformado num problema de Programação Linear que possui apenas variáveis canalizadas. Tal problema foi definido como sendo o problema (1.9) (não estamos escrevendo aqui a parte constante da função objetivo):

$$\begin{cases} \min \sum_{k=1}^{K} (q_k \; c \;/\; 0) \begin{pmatrix} y'_k \\ \hline u'_k \end{pmatrix} \\ \text{s.a.} \\ \quad \sum_{k=1}^{K} (Q_k \;/\; 0) \begin{pmatrix} y'_k \\ \hline u'_k \end{pmatrix} + I \; u'_{K+1} = b'_o \\ \quad (A \;/\; I) \begin{pmatrix} y'_k \\ \hline u'_k \end{pmatrix} = b'_k \;, \; k = 1,\ldots,K \\ \quad 0 \leq u'_k \leq \beta_k - \alpha_k \;, \quad k = 1,\ldots,K \\ \quad 0 \leq u'_{K+1} \leq \ell - g \\ \quad 0 \leq y'_k \leq \delta_k - \gamma_k, \quad k = 1,\ldots,K \end{cases}$$

O Método de Dantzig - Wolfe, quando aplicado ao problema acima, apresenta algumas particularidades:

1ª) Como as matrizes dos blocos são idênticas, armazenamos somente uma matriz (A) e, durante a resolução dos subproblemas independentes, montamos a matriz (A / I) para cada subproblema k, k = 1,...,K.

2ª) A menos da multiplicação pelas quantidades q_k, os coeficientes da função objetivo do problema (1.9) são idênticos para os K blocos. Além disso, as matrizes Q_k, k = 1,...,K, possuem a forma:

$$Q_k = \begin{bmatrix} q_k & & & \\ & q_k & & \\ & & \ddots & \\ & & & q_k \end{bmatrix} = q_k \cdot I \quad ,$$

onde I é a matriz identidade de ordem n. Isto nos permite armazenar apenas um vetor c e, em vez de matrizes Q_k, armazenamos apenas os escalares q_k. Assim, na montagem dos coeficientes da função objetivo para a resolução de cada subproblema k, teremos:

$$\min f_k = \left[(q_k . c \;/\; 0) - \pi_o \; (q_k . I \;/\; 0) \right] \begin{pmatrix} y_k' \\ \hline u_k' \end{pmatrix} =$$

$$= (q_k . c - \pi_o \; q_k) \; y_k' = q_k \; (c - \pi_o) \; y_k' \; .$$

3ª) Uma coluna R_k^j, gerada pelo bloco k, k = 1,...,K, na iteração j, para entrar na base do problema mestre, terá a forma:

$$R_k^j = \begin{bmatrix} (Q_k/0)\left(\dfrac{\hat{y}_k'^{\,j}}{\hat{u}_k'^{\,j}}\right) \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} = \begin{bmatrix} q_k \; I \; \hat{y}_k'^{\,j} \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} = \begin{bmatrix} q_k \; \hat{y}'^{\,j}_k \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}$$

onde $\hat{y}_k'^{\,j}$ é a solução ótima encontrada pelo subproblema k na iteração j.

4ª) Além dos K blocos, constituídos pelas K rações, temos agora um bloco a mais, ao qual chamaremos de bloco K+1. A matriz deste bloco, nas restrições de acoplamento, é a matriz identidade de ordem n, e suas restrições independentes são apenas as restrições de canalização sobre as variáveis u'_{K+1} - por esta simplicidade, este bloco será sempre tratado à parte.

Uma coluna R_{K+1}^j gerada pelo bloco K+1 na iteração j, para entrar na base do problema mestre, terá a forma:

$$R_{K+1}^j = \begin{bmatrix} I \; \hat{u}'^{\,j}_{K+1} \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \hat{u}'^{\,j}_{K+1} \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 1 \end{bmatrix}$$

onde I é a matriz identidade de ordem n e $\hat{u}'^{\,j}_{K+1}$ é a solução ótima encontrada pelo subproblema K+1 na iteração j.

Como os coeficientes da função objetivo no problema (1.9), referentes ao bloco K+1, são zero, teremos como coeficientes da função objetivo para o subproblema K+1:

$$\min f_{K+1} = (0 - \pi_o I)\ u'_{K+1} = - \pi_o\ u'_{K+1}$$

Assim, em cada iteração, ao resolvermos o subproblema K+1, este subproblema será:

$$\begin{cases} \min f_{K+1} = -\pi_o\ u'_{K+1} \\ \text{s.a.} \\ \qquad 0 \leqslant u'_{K+1} \leqslant \ell - g \end{cases}$$

Este problema tem uma solução analítica simples: chamando de π_{o_i} as componentes de π_o, de u'_{K+1_i} as componentes de u'_{K+1} e de $\ell_i - g_i$ as componentes de $\ell - g$, $i = 1,\ldots,n$, temos como solução ótima do subproblema K+1:

$$\text{se } \pi_{0_i} \leqslant 0 \Longrightarrow u'_{K+1_i} = 0$$

$$\text{se } \pi_{0_i} > 0 \Longrightarrow u'_{K+1_i} = \ell_i - g_i$$

5ª.) Podemos checar, facilmente, um tipo de infactibilidade nas restrições de acoplamento: no problema original (1.2), as restrições de acoplamento em forma matricial são:

$$g \leqslant \sum_{k=1}^{K} Q_k\ y_k \leqslant \ell \qquad (2.8)$$

As restrições de canalização sobre y_k são:

$$\gamma_k \leqslant y_k \leqslant \delta_k \quad , \gamma_k \ , \delta_k \geqslant 0, \ \forall k.$$

Então, $\forall y_k$ factível $\Longrightarrow y_k \geq \gamma_k$. Mas $Q_k \; \gamma_k = q_k I \gamma \leq q_k \; y_k$,

porque $q_k > 0$, $\forall k$, e logo,

$$\sum_{k=1}^{K} q_k \gamma_k \leq \sum_{k=1}^{K} q_k \; y_k \; . \tag{2.9}$$

Ou seja: se y_k é factível, $\forall k$ $\Longrightarrow$

$$\Longrightarrow \sum_{k=1}^{K} Q_k y_k = \sum_{k=1}^{K} q_k y_k \leq \ell \text{ (por (2.8))} \Longrightarrow \sum_{k=1}^{K} q_k \gamma_k \leq \ell \text{ (por(2.9))}$$

$$\Longrightarrow \ell - \sum_{k=1}^{K} q_k \; \gamma_k \geq 0 \tag{2.10}$$

Quando fazemos as transformações sobre as variáveis do problema (1.2), chegamos ao problema (1.9) que possui o seguinte lado direito:

$$b_o' = \ell - \sum_{k=1}^{K} q_k \; \gamma_k \quad .$$

Como b_o' é igual ao lado esquerdo de (2.10), verificamos se esta inequação é satisfeita. Se, para alguma componente i, ela não for satisfeita, concluímos então que o problema original é infactível.

Infactibilidades análogas podem ser checadas nos blocos: as restrições relativas aos blocos (rações) no problema (1.2) são:

$$\left\{ \begin{bmatrix} \alpha_k \leq & A \; y_k \leq \beta_k \\ \gamma_k \leq & y_k \leq \delta_k \end{bmatrix} \right\} , \; k = 1,\ldots,K$$

ou, com a introdução das variáveis u_k:

$$\left\{\begin{array}{l} (A/I)\begin{pmatrix} y_k \\ u_k \end{pmatrix} = \alpha_k + \beta_k \\ \gamma_k \leq y_k \leq \delta_k \\ \alpha_k \leq u_k \leq \beta_k \end{array}\right\} , \forall k .$$

Para qualquer par $\begin{pmatrix} y_k \\ u_k \end{pmatrix}$ factível $\Longrightarrow y_k \geq \gamma_k$ e $u_k \geq \alpha_k$, e como $A \geq 0$ (A é formada por elementos positivos ou nulos) , temos que:

$$A\ \gamma_k \leq A\ y_k \quad e \quad I\alpha_k \leq I\ u_k \quad , \quad ou$$

$$A\ \gamma_k + I\ \alpha_k \leq Ay_k + I\ u_k \qquad (2.11)$$

Como $A\ y_k + Iu_k = \alpha_k + \beta_k$, $\forall k$, temos que

$$A\ \gamma_k + I\alpha_k \leq \alpha_k + \beta_k \quad \text{(por (2.11)), ou seja:}$$

$$\alpha_k + \beta_k - A\ \gamma_k - \alpha_k \geq 0 \ , \quad ou$$

$$\beta_k - A\ \gamma_k \geq 0 \ , \quad k= 1,\ldots,K \qquad (2.12)$$

Da mesma maneira, o lado esquerdo de (2.12) é o novo lado direito b_k' de cada bloco k, $\forall k$, do problema modificado (1.9) e logo, a mesma verificação quanto à não negatividade das componentes pode ser feita.

Se alguma componente de b_k' for negativa, o subproblema k será infactível, e logo, o problema original também o será.

Devido a todas estas particularidades, incluindo a do bloco K+1, foram feitos programas especiais do Método de Dantzig - Wolfe para o caso específico da otimização global de rações.

CAPÍTULO 3

APLICAÇÃO DO MÉTODO DE ROSEN

3.1 - INTRODUÇÃO:

Ao contrário do Método de Dantzig - Wolfe, o Método de Rosen não cria novas variáveis na formação do problema mestre : ele particiona as variáveis dos blocos em dependentes e independentes e, posteriormente, estende esta partição até às restrições de acoplamento - é um método de partição. Isto assegura um problema mestre pequeno, que terá sempre m_o linhas (o número de restrições de acoplamento), e que será chamado de problema reduzido. Este problema será sempre formado pelas variáveis independentes, sendo então esquecidas as canalizações sobre as variáveis dependentes - por isso, este também é um método de relaxação. Como as variáveis independentes são canalizadas, o problema reduzido será um problema de Programação Linear com variáveis canalizadas.

O método gera uma sequência de soluções básicas infactíveis, correspondendo a uma sequência de soluções factíveis para o problema dual. Mostraremos que, quando as soluções de todos os blocos forem factíveis para o problema primal (1.8), estaremos na solução ótima do problema, e enquanto esta condição não for satisfeita, é feita uma mudança de base em cada bloco considerado "não ótimo".

Assumiremos, durante o desenvolvimento deste capítulo, que as matrizes de restrições de cada bloco k, k = 1, ..., K, do problema (1.8), têm posto m_k, respectivamente, isto é, cada matriz A_k contém uma submatriz não singular A_k^B de dimensão $m_k \times m_k$. Além disso, assumiremos que, para cada bloco k, $k = 1,\dots, K$, $n_k > m_k$.

3.2 - O PROBLEMA DUAL:

Chamaremos de problema primal ao problema (1.8):

$$\begin{cases} \min z = \sum_{k=1}^{K} c_k x_k \\ \text{s.a.} \\ \sum_{k=1}^{K} Q_k x_k = b_o \\ A_k x_k = b_k, \; k = 1,\dots,K \\ 0 \leq x_k \leq h_k, \; k = 1,\dots,K \end{cases}$$

Vamos dualizar o problema (1.8) em relação às restrições de acoplamento. Seja Γ o vetor das variáveis duais associadas às restrições de acoplamento $\sum_{k=1}^{K} Q_k x_k = b_o$; Γ é um vetor com dimensão $1 \times m_o$. O Lagrangeano $L(x_k,\Gamma)$ é então dado por:

$$L(x_k,\Gamma) = \sum_{k=1}^{K} c_k x_k + \Gamma\,(b_o - \sum_{k=1}^{K} Q_k x_k) \text{ , ou,}$$

$$L(x_k,\Gamma) = \Gamma b_o + \sum_{k=1}^{K} (c_k - \Gamma Q_k)\, x_k \tag{3.1}$$

Definimos a função dual $\phi(\Gamma)$ como:

$$\phi(\Gamma) = \min_{x_k \in \Omega_k} L(x_k, \Gamma),$$

onde $\Omega_k = \{x_k / A_k x_k = b_k,\ 0 \leqslant x_k \leqslant h_k\}$, $k=1,\ldots,K$.

Substituindo a equação (3.1) em $\emptyset\ (\Gamma)$, temos:

$$\emptyset\ (\Gamma) = \min_{x_k \in \Omega_k} \{\Gamma b_o + \sum_{k=1}^{K} (c_k - \Gamma Q_k)\ x_k\} =$$

$$= \Gamma b_o + \min_{x_k \in \Omega_k} \{\sum_{k=1}^{K} (c_k - \Gamma Q_k)\ x_k\}\ ,\ \text{ou}\ ,$$

$$\emptyset\ (\Gamma) = \Gamma b_o + \sum_{k=1}^{K} \left[\min_{x_k \in \Omega_k} (c_k - \Gamma Q_k)\ x_k\right]$$

ou seja, dado Γ, para obtermos $\emptyset\ (\Gamma)$ devemos resolver K subproblemas da forma:

$$\left\{\begin{array}{l} \min\ (c_k - \Gamma Q_k)\ x_k \\ \text{s.a.} \\ \qquad A_k x_k = b_k \\ \qquad 0 \leqslant x_k \leqslant h_k \end{array}\right\} \quad k = 1,\ldots,K \qquad (3.2)$$

Para encontrarmos o dual de cada subproblema (3.2) escreveremos, primeiramente, as canalizações destes subproblemas em forma de restrições, obtendo então:

$$\left\{\begin{array}{l} \min\ (c_k - \Gamma Q_k)\ x_k \\ \text{s.a.} \\ \qquad A_k x_k = b_k \\ \qquad x_k \leqslant h_k \\ \qquad x_k \geqslant 0 \end{array}\right\} \quad k = 1,\ldots,K \qquad (3.3)$$

Sejam:

π_k : vetor das variáveis duais associadas às restrições $A_k x_k = b_k$;

π_k é um vetor linha de dimensão $1 \times m_k$, $k = 1,\dots,K$.

μ_k : vetor das variáveis duais associadas às restrições $x_k \leqslant h_k$;

μ_k é um vetor linha de dimensão $1 \times n_k$, $k = 1,\dots,K$.

Então, o dual de cada subproblema (3.3) será:

$$\left\{\begin{array}{l} \max\ (\pi_k b_k - \mu_k h_k) \\ \text{s.a.} \\ \qquad \pi_k A_k - \mu_k \leqslant c_k - \Gamma Q_k \\ \qquad\qquad \mu_k \geqslant 0 \end{array}\right\} \quad k = 1,\dots,K \qquad (3.4)$$

Assim, o dual completo do problema (1.8) pode ser visto como:

$$\max \emptyset\ (\Gamma) = \max_{\Gamma} \left\{ \Gamma b_o + \sum_{k+1}^{K} [\max(\pi_k b_k - \mu_k h_k) / \pi_k A_k - \mu_k \leqslant c_k - \Gamma Q_k, \mu_k \geqslant 0\ ] \right\}$$

ou seja:

$$\left\{\begin{array}{ll} \max \emptyset = \sum_{k=1}^{K} (\pi_k b_k - \pi_k h_k) + \Gamma b_o & \\ \text{s.a.} \qquad \pi_k A_k - \mu_k + \Gamma Q_k \leqslant c_k \ , \ k = 1,\dots,K & \\ \qquad\qquad \mu_k \geqslant 0,\ k = 1,\dots,K & \end{array}\right\} \qquad (3.5)$$

3.3 - <u>PARTIÇÃO DAS VARIÁVEIS</u> :

Inicialmente, arbitramos um vetor $\overset{o}{\Gamma}$ e resolvemos os K subproblemas (3.2) com $\Gamma = \overset{o}{\Gamma}$. Este vetor $\overset{o}{\Gamma}$ deve ser tal que todo subproblema (3.4) tenha solução factível, isto é, existem $\overset{o}{\pi}_k$ e $\overset{o}{\mu}_k$ tais que $\overset{o}{\pi}_k A_k - \overset{o}{\mu}_k \leqslant c_k - \overset{o}{\Gamma} Q_k$ e $\overset{o}{\mu}_k \geqslant 0$, $k = 1,\ldots,K$. Note que, se o problema dual completo (3.5) tem solução factível, este vetor $\overset{o}{\Gamma}$ existe. Isto fica garantido se o problema (1.8) não for ilimitado.

Na resolução dos K subproblemas (3.2) pelo Método Simplex Revisado com variáveis canalizadas, se necessário, podemos empregar a fase I. Se algum destes subproblemas for infactível, o problema original (1.8) não terá solução.

Obtidas as soluções ótimas de cada subproblema k, faremos então as partições em cada bloco k , $k = 1,\ldots,K$: cada bloco será particionado em uma parte básica e uma parte não básica e, por conveniência, suporemos que as colunas básicas de cada bloco k são as primeiras m_k colunas e as não básicas são as últimas $n_k - m_k$ colunas. Sejam, para $k = 1,\ldots,K$:

$$x_k = (x_k^B \,/\, x_k^N)^T \text{ , onde:}$$

x_k^B : vetor $m_k \times 1$, formado pelas componentes básicas de x_k ;

x_k^N : vetor $(n_k - m_k) \times 1$, formado pelas componentes não básicas de x_k.

$$A_k = (A_k^B \,/\, A_k^N)\text{, onde:}$$

A_k^B : matriz $m_k \times m_k$, formada pelas colunas básicas A_k e cuja inversa é B_k^{-1} ;

A_k^N : matriz m_k x $(n_k - m_k)$, formada pelas colunas não básicas de A_k .

$$h_k = (h_k^B \,/\, h_k^N)^T \text{ , onde:}$$

h_k^B : vetor m_k x 1, formado pelas componentes de h_k que correspondem às variáveis básicas do bloco k;

h_k^N : vetor $(n_k - m_k)$ x 1, formado pelas componentes de h_k que correspondem às variáveis não básicas do bloco k.

Estenderemos esta partição, feita nos blocos, à função objetivo e às restrições de acoplamento do problema (1.8):

$$c_k = (c_k^B \,/\, c_k^N) \text{ , onde:}$$

c_k^B : vetor 1 x m_k , formado pelas componentes de c_k que correspondem às variáveis básicas na solução ótima do bloco k;

c_k^N : vetor 1 x $(n_k - m_k)$, formado pelas componentes de c_k que correspondem às variáveis não básicas na solução ótima do bloco k.

$$Q_k = (Q_k^B \,/\, Q_k^N) \text{, onde :}$$

Q_k^B : matriz m_o x m_k, formada pelas colunas de Q_k que correspondem às variáveis básicas na solução ótima do bloco k;

Q_k^N : matriz m_o x $(n_k - m_k)$, formada pelas colunas de Q_k que correspondem às variáveis não básicas na solução ótima do blo co k.

Podemos ver todas estas partições, por exemplo para $K = 2$, na seguinte figura:

$c_1^B\ x_1^B$	$+\ c_1^N\ x_1^N$	$+c_2^B x_2^B$	$+\ c_2^N\ x_2^N$	$=\ \min\ z$
$Q_1^B\ x_1^B$	$+\ Q_1^N\ x_1^N$	$+Q_2^B x_2^B$	$+\ Q_2^N\ x_2^N$	$=\ b_o$
$A_1^B\ x_1^B$	$+\ A_1^N\ x_1^N$			$=\ b_1$
		$A_2^B x_2^B$	$+\ A_2^N\ x_2^N$	$=\ b_2$

Observação: como as variáveis x_k são canalizadas, uma componente de x_k^N pode ser igual ao seu limite inferior (zero) ou igual ao seu limite superior. Luenberger [1973, pp. 48-53], propõe uma maneira de colocar todas as variáveis não básicas com o valor zero. Esta mudança de variáveis foi implementada na subrotina que resolve os subproblemas pelo Método Simplex Revisado com variáveis canalizadas e foi estendida ao Método de Rosen. Assim, as partes não básicas x_k^N , A_k^N, h_k^N , c_k^N e Q_k^N , $k = 1,\ldots,K$, foram ainda particionadas em uma parte que corresponde às variáveis não básicas no limite inferior e em outra parte que corresponde às variáveis não básicas no limite superior. Embora estas novas partições tenham sido consideradas nos programas, não trataremos delas neste capítulo para que a notação não fique muito complicada. A mudança de variáveis sugerida por Luenberger [1973] e que foi utilizada na subrotina que implementa o Método Simplex Revisado com variáveis canalizadas, está explicada no apêndice A e o tratamento explícito do Método de Rosen com as novas partições nas variáveis não básicas se encontra no apêndice B. Apenas lembraremos aqui, que as variáveis não básicas de cada bloco k podem estar em um dos dois limites de canalização, e faremos referências aos apêndices A e/ou B quando julgarmos necessário.

Usaremos agora as matrizes básicas não singulares A_k^B, $k = 1,\ldots,K$, para eliminarmos os vetores x_k^B de (1.8), expressando as variáveis básicas x_k^B de cada bloco k em função das variáveis não básicas x_k^N. Sabemos que a solução ótima de cada subproblema (3.2) deve satisfazer:

$$x_k^B = B_k^{-1} b_k - B_k^{-1} A_k^N x_k^N \quad , \; k = 1,\ldots,K \, , \qquad (3.6)$$

onde cada x_k^N possui componentes no limite inferior e/ou no limite superior (a expressão (3.6) está melhor explicitada no apêndice B). Com a expressão (3.6), formaremos o <u>problema reduzido</u>.

3.4 - <u>A FORMAÇÃO DO PROBLEMA REDUZIDO</u>:

Tendo expressado as variáveis básicas em função das variáveis não básicas de cada bloco, vamos agora formar o problema reduzido, que é obtido quando substituímos a equação (3.6) nas restrições de acoplamento e na função objetivo do problema (1.8).

3.4.1 - Formação das restrições:

As restrições de acoplamento do problema (1.8) são:

$$\sum_{k=1}^{K} Q_k x_k = b_o \ .$$

Incluindo as partições feitas em Q_k e em x_k, temos:

$$\sum_{k=1}^{K} (Q_k^B \ / \ Q_k^N) \begin{pmatrix} x_k^B \\ \hline x_k^N \end{pmatrix} = b_o \ , \text{ ou } ,$$

$$\sum_{k=1}^{K} [Q_k^B \ x_k^B + Q_k^N \ x_k^N \ ] = b_o \ .$$

Substituindo agora a equação (3.6) na equação acima e desenvolvendo a expressão, temos:

$$\sum_{k=1}^{K} [Q_k^B \ (B_k^{-1} \ b_k - B_k^{-1} \ A_k^N \ x_k^N) + Q_k^N \ x_k^N \ ] = b_o$$

$$\sum_{k=1}^{K} [Q_k^B \ B_k^{-1} \ b_k - Q_k^B \ B_k^{-1} \ A_k^N \ x_k^N + Q_k^N \ x_k^N \ ] = b_o$$

$$\sum_{k=1}^{K} [Q_k^N - Q_k^B \ B_k^{-1} \ A_k^N \ ] \ x_k^N = b_o - \sum_{k=1}^{K} Q_k^B \ B_k^{-1} \ b_k \ .$$

Sendo $\hat{A}_k = B_k^{-1} \ A_k^N$ e substituindo $\hat{A}_k$ na equação acima, temos:

$$\sum_{k=1}^{K} (Q_k^N - Q_k^B \ \hat{A}_k) \ x_k^N = b_o - \sum_{k=1}^{K} Q_k^B \ B_k^{-1} \ b_k \ , \tag{3.7}$$

que são as restrições do problema reduzido.

3.4.2 - Formação da função objetivo:

A função objetivo do problema (1.8) é:

$$\min z = \sum_{k=1}^{K} c_k \, x_k \quad .$$

Incluindo as partições feitas em c_k e em x_k, temos:

$$\min z = \sum_{k=1}^{K} (c_k^B / c_k^N) \begin{pmatrix} x_k^B \\ \hline x_k^N \end{pmatrix} = \sum_{k=1}^{K} \left[c_k^B \, x_k^B + c_k^N \, x_k^N \right] .$$

Substituindo a equação (3.6) na equação acima e desenvolvendo a expressão, obtemos:

$$\min z = \sum_{k=1}^{K} \left[c_k^B \, (B_k^{-1} \, b_k - B_k^{-1} \, A_k^N \, x_k^N) + c_k^N \, x_k^N \right] =$$

$$= \sum_{k=1}^{K} \left[c_k^B \, B_k^{-1} \, b_k - c_k^B \, B_k^{-1} \, A_k^N \, x_k^N + c_k^N \, x_k^N \right] =$$

$$= \sum_{k=1}^{K} (c_k^N - c_k^B \, B_k^{-1} \, A_k^N) \, x_k^N + \sum_{k=1}^{K} c_k^B \, B_k^{-1} \, b_k \quad , \text{ou,}$$

$$\min z - \sum_{k=1}^{K} c_k^B \, B_k^{-1} \, b_k = \sum_{k=1}^{K} (c_k^N - c_k^B \, B_k^{-1} \, A_k^N) \, x_k^N \, .$$

Como anteriormente, vamos substituir $B_k^{-1} \, A_k^N$ por $\hat{A}_k$ para obtermos:

$$\min z - \sum_{k=1}^{K} c_k^B \, B_k^{-1} \, b_k = \sum_{k=1}^{K} (c_k^N - c_k^B \, \hat{A}_k) \, x_k^N \quad .$$

Como o termo $\sum_{k=1}^{K} c_k^B \, B_k^{-1} \, b_k$ é constante e, logo, pode ser dispensado da minimização, a função objetivo do problema reduzido pode ser vista como:

$$\min \sum_{k=1}^{K} (c_k^N - c_k^B \, \hat{A}_k) \, x_k^N \qquad (3.8)$$

O problema reduzido é então:

$$\left\{\begin{array}{l} \min \sum_{k=1}^{K} (c_k^N - c_k^B \hat{A}_k)\, x_k^N \\ \text{s.a.} \\ \qquad \sum_{k=1}^{K} (Q_k^N - Q_k^B \hat{A}_k)\, x_k^N = b_o - \sum_{k=1}^{K} Q_k^B B_k^{-1} b_k \\ \qquad 0 \leq x_k^N \leq h_k^N \quad , \quad k = 1,\dots,K \end{array}\right. \qquad (3.9)$$

<u>Observação</u>: a formação do problema reduzido está melhor explicitada no apêndice B.

A construção das restrições e da função objetivo do problema reduzido pode ser vista, por pivotamento, num esquema prático, no qual estamos esquecendo que as variáveis são canalizadas e estamos supondo K=2.

A partição original seria:

c_1^B	c_1^N	c_2^B	c_2^N	$= z$ (min)
Q_1^B	Q_1^N	Q_2^B	Q_2^N	$= b_o$
A_1^B	A_1^N			$= b_1$
		A_2^B	A_2^N	$= b_2$

Multiplicando o bloco 1 por B_1^{-1} e o bloco 2 por B_2^{-1}, obtemos:

$$
\begin{array}{|c|c|c|c|l}
\hline
c_1^B & c_1^N & c_2^B & c_2^N & = z \text{ (min)} \\
\hline
Q_1^B & Q_1^N & Q_2^B & Q_2^N & = b_o \\
\hline
I & \hat{A}_1 & & & = B_1^{-1} b_1 \\
\hline
 & & I & \hat{A}_2 & = B_2^{-1} b_2 \\
\hline
\end{array}
$$

onde $\hat{A}_1 = B_1^{-1} A_1^N$ e $\hat{A}_2 = B_2^{-1} A_2^N$.

Multiplicamos agora o bloco 1 por Q_1^B e subtraímos o resultado da parte do acoplamento referente ao bloco 1. Multiplicamos depois o bloco 1 por c_1^B e subtraímos o resultado da parte da função objetivo referente ao bloco 1. Fazemos o mesmo para o bloco 2 (multiplicando-o por Q_2^B e c_2^B), obtendo então:

$$
\begin{array}{r|c|c|c|c|l}
\cline{2-5}
 & 0 & c_1^N - c_1^B \hat{A}_1 & 0 & c_2^N - c_2^B \hat{A}_2 & = z\,(\min) - c_1^B B_1^{-1} b_1 - c_2^B B_2^{-1} b_2 \\
\cline{2-5}
m_o \{ & 0 & \underbrace{Q_1^N - Q_1^B \hat{A}_1}_{(*)} & 0 & \underbrace{Q_2^N - Q_2^B \hat{A}_2}_{(**)} & = b_o - Q_1^B B_1^{-1} b_1 - Q_2^B B_2^{-1} b_2 \\
\cline{2-5}
m_1 \{ & I & \hat{A}_1 & & & = B_1^{-1} b_1 \\
\cline{2-5}
m_2 \{ & & & I & \hat{A}_2 & = B_2^{-1} b_2 \\
\cline{2-5}
\end{array}
$$

Procurando uma base para o problema global, vemos que já possuímos $m_1 + m_2$ variáveis básicas, que são dadas pelos blo - cos (subproblemas). Precisamos ainda encontrar as m_o variáveis básicas restantes, que serão tiradas de (*) e de (**), resolvendo o problema reduzido.

O problema reduzido (3.9) é então formado pelas variáveis não básicas de cada bloco e, por isso, é um problema de Programação Linear com variáveis canalizadas, que terá sempre m_o restrições (isto é, o número de restrições de acoplamento) e seu número de variáveis será sempre a soma do número de variáveis não básicas dos blocos, ou seja, $\sum_{k=1}^{K} (n_k - m_k)$.

3.5 - VERIFICAÇÃO DA OTIMALIDADE DOS BLOCOS:

Tendo montado o problema reduzido (3.9), resolvemos este problema pelo Método Simplex Revisado com variáveis canalizadas. Como o problema reduzido não contém explicitamente as variáveis x_k^B , $k = 1,\ldots,K$, ele não força que estas variáveis estejam dentro de seus limites de canalização - assim, estas restrições são relaxadas. Então, o conjunto de soluções de (3.9) e de (3.6) contém o conjunto de soluções do problema primal (1.8) - assim , se o problema reduzido for infactível, o problema original (1.8) também o será. Contudo, o problema reduzido pode apresentar so - lução ilimitada em casos onde o problema original tem uma solução ótima finita. Convém lembrar aqui, que embora estejamos considerando, nesta apresentação, que todas as variáveis são canalizadas (entre limites finitos), nos programas que foram feitos considera mos que as variáveis podem ser canalizadas ou não (neste caso, não negativas). Para eliminarmos a possibilidade de ocorrer uma solução ilimitada, podemos acrescentar ao problema reduzido, como sugerido em Lasdon [1970, pp. 279 e 286], uma restrição do tipo:

$$\sum_{k=1}^{K} e_k' \, x_k^N \leqslant M \tag{3.10}$$

onde e_k' são vetores com dimensão $1 \times (n_k - m_k)$, $k = 1,\ldots,K$, com todas as componentes iguais a 1 e M é um número positivo e suficientemente grande (o acréscimo desta restrição não foi implementado nos programas que efetuam o Método de Rosen).

Resolvido o problema reduzido, teremos então como solução ótima para as variáveis x_k^N, os valores $\bar{x}_k^N$, $k = 1,\ldots, K$. Como as equações dos blocos ($A_k \, x_k = b_k$) são satisfeitas por causa de (3.6), uma solução factível do problema reduzido satisfaz todas as equações do problema original (1.8), mas pode não satisfazer $0 \leqslant x_k^B \leqslant h_k^B$, $k = 1,\ldots,K$, quando substituímos os valores de x_k^N, vindos do problema reduzido ($\bar{x}_k^N$), em (3.6), ou seja,

$$\bar{x}_k^B = B_k^{-1} \, b_k - B_k^{-1} \, A_k^N \, \bar{x}_k^N \quad , \quad \forall k \text{ , ou,}$$

$$\bar{x}_k^B = B_k^{-1} \, b_k - \hat{A}_k \, \bar{x}_k^N \, , \quad k = 1,\ldots,K \, , \tag{3.11}$$

que é a equação (3.6), mas agora com os novos valores $\bar{x}_k^N$, $\forall k$.

Então, para cada bloco k, substituímos os valores $\bar{x}_k^N$ na equação (3.11). Se $0 \leqslant \bar{x}_k^B \leqslant h_k^B$, o bloco k é factível, e diremos que o bloco k é ótimo. Senão, devemos fazer uma correção no bloco k para que ele volte a ser factível. Quando todos os blocos forem ótimos, já temos a solução ótima do problema original, como mostramos no teorema a seguir.

<u>Teorema 3.1 - teste de otimalidade</u>:

Os vetores $\bar{x}_k = (\bar{x}_k^B / \bar{x}_k^N)^T$, $k = 1,\ldots,K$, formam a solução ótima do problema original (1.8) se e somente se $0 \leq \bar{x}_k^B \leq h_k^B$, $k = 1,\ldots,K$, onde $\bar{x}_k^B$, para $\forall k$, é dado por (3.11) e $\bar{x}_k^N$, $k = 1,\ldots,K$, formam a solução ótima do último problema reduzido (3.9).

<u>Prova</u>:

Inicialmente, vamos escrever o dual do problema reduzido (3.9):

$$\left\{\begin{array}{l} \max \quad \Gamma \, (b_o - \sum_{k=1}^{K} Q_k^B B_k^{-1} b_k) - \sum_{k=1}^{K} \varphi_k h_k^N \\ \text{s.a.} \\ \qquad \Gamma \, (Q_k^N - Q_k^B \hat{A}_k) - \varphi_k \leq c_k^N - c_k^B \hat{A}_k \ , \ k = 1,\ldots,K \\ \qquad \varphi_k \geq 0 \ , \ k = 1,\ldots,K \end{array}\right\} \quad (3.12)$$

onde Γ $(1 \times m_o)$ e φ_k $(1 \times (n_k - m_k))$, $k = 1,\ldots,K$, são as variáveis duais associadas às restrições

$$\sum_{k=1}^{K} (Q_k^N - Q_k^B \hat{A}_k) \, x_k^N = b_o - \sum_{k=1}^{K} Q_k^B B_k^{-1} b_k \quad \text{e} \quad x_k^N \leq h_k^N \ , \ \forall k \quad ,$$

respectivamente, do problema (3.9).

Sejam:

$\overline{\Gamma}$: parte da solução dual, referente às restrições

$\sum_{k=1}^{K} (Q_k^N - Q_k^B \hat{A}_k)\, x_k^N = b_o - \sum_{k=1}^{K} Q_k^B B_k^{-1} b_k$, obtida do último problema reduzido (3.9).

$\overline{\varphi}_k$, k = 1,...,K : parte da solução dual, referente às restrições de limitações superiores, $x_k^N \leq h_k^N$, obtida do último problema reduzido (3.9). Da Programação Linear (folgas complementares) sabemos que, para cada bloco k, as componentes de φ_k correspondentes às variáveis básicas e às variáveis não básicas no limite inferior, na solução ótima do problema reduzido, são nulas, enquanto que as componentes correspondentes às variáveis não básicas no limite superior, são não negativas. Chamando de JB, JI e JS aos conjuntos dos índices das variáveis básicas, não básicas no limite inferior e não básicas no limite superior, respectivamente, na solução ótima do problema reduzido, correspondentes a cada bloco k, temos que:

$$\varphi_k^{JB} = 0$$

$$\varphi_k^{JI} = 0$$

$$\varphi_k^{JS} = \overline{\Gamma}\,(Q_k^{JS} - Q_k^B \hat{A}_k^{JS}) - (c_k^{JS} - c_k^B \hat{A}_k^{JS}) \geq 0 ,$$

onde $\hat{A}_k^{JS} = B_k^{-1} A_k^{JS}$.

$\overset{o}{\pi}_k$, k = 1, ..., K: parte da solução dual, referente à restrição $A_k x_k = b_k$, obtida do último subproblema k (3.2). Da

Programação Linear sabemos que $\overset{o}{\pi}_k = (c_k^B - \overset{o}{\Gamma} Q_k^B)\, B_k^{-1}$, onde $\overset{o}{\Gamma}$ é a parte da solução dual obtida do problema reduzido anterior.

$\overset{o}{\mu}_k$, $k = 1,\ldots,K$: parte da solução dual, referente à restrição $x_k \leq h_k$, obtida do último subproblema k (3.2). Da Programação Linear (folgas complementares) sabemos que, para cada subproblema k, as componentes de $\overset{o}{\mu}_k$ correspondentes às variáveis básicas e às variáveis não básicas no limite inferior, na solução ótima do subproblema k, são nulas, enquanto que as componentes correspondentes às variáveis não básicas no limite superior, são não negativas. Chamando de B, I e S aos conjuntos de índices das variáveis básicas, não básicas no limite inferior e não básicas no limite superior, respectivamente, na solução ótima do subproblema k, temos que:

$$\overset{o}{\mu}{}_k^B = \overset{o}{\pi}_k A_k^B - (c_k^B - \overset{o}{\Gamma} Q_k^B) = 0$$

$$\overset{o}{\mu}{}_k^I = 0$$

$$\overset{o}{\mu}{}_k^S = \overset{o}{\pi}_k A_k^S - (c_k^S - \overset{o}{\Gamma} Q_k^S) \geq 0$$

e $\overset{o}{\mu}{}_k^N = (\overset{o}{\mu}{}_k^I \;/\; \overset{o}{\mu}{}_k^S)$.

$\bar{\pi}_k$, $k = 1,\ldots,K$: é obtido quando substituímos $\overset{o}{\Gamma}$ por $\bar{\Gamma}$, isto é,

$$\bar{\pi}_k = (c_k^B - \bar{\Gamma} Q_k^B)\, B_k^{-1} \quad , \; k = 1,\ldots,K \qquad (3.13)$$

$\bar{\mu}_k$, $k = 1,\ldots,K$: na parte básica $(\bar{\mu}_k^B)$, é obtido quando substituímos $\overset{o}{\Gamma}$ por $\bar{\Gamma}$ e $\overset{o}{\pi}_k$ por $\bar{\pi}_k$ e na parte não básica $(\bar{\mu}_k^N)$ é obtido substituindo $\overset{o}{\mu}{}_k^I$ e $\overset{o}{\mu}{}_k^S$ por $\bar{\varphi}_k$, ou seja:

$$\bar{\mu}_k^B = \bar{\pi}_k\, A_k^B - (c_k^B - \bar{\Gamma} Q_k^B) = (c_k^B - \bar{\Gamma} Q_k^B)\, B_k^{-1}\, A_k^B - c_k^B + \bar{\Gamma} Q_k^B =$$

$$= c_k^B - \bar{\Gamma} Q_k^B - c_k^B + \bar{\Gamma} Q_k^B = 0.$$

$$\bar{\mu}_k^N = (\bar{\varphi}_k^{JB} \,/\, \bar{\varphi}_k^{JI} \,/\, \bar{\varphi}_k^{JS}) \quad = (0 \,/\, 0 \,/\, \bar{\varphi}_k^{JS}).$$

Então, $\bar{\mu}_k = (\underbrace{0}_{B} \,/\, \underbrace{0 \,/\, 0 \,/\, \bar{\varphi}_k^{JB}}_{N}) = (\underbrace{0}_{B} \,/\, \underbrace{\bar{\varphi}_k}_{N})$ (3.14)

i) Vamos supor que $0 \leqslant \bar{x}_k^B \leqslant h_k^B$, $k = 1,\ldots,K$.

Os vetores $\bar{x}_k = (\bar{x}_k^B \,/\, \bar{x}_k^N)^T$, $\forall k$, formam uma solução factível para o problema primal (1.8), pois, uma vez que x_k^B são dados por (3.6),

$$x_k^B = B_k^{-1}\, b_k - B_k^{-1}\, A_k^N\, x_k^N \quad , \quad k = 1,\ldots,K \; ,$$

segue - se,de (3.11),

$$\bar{x}_k^B = B_k^{-1}\, b_k - \hat{A}_k\, \bar{x}_k^N \quad , \; k = 1,\ldots,K,$$

que os vetores $(\bar{x}_k^B \,/\, \bar{x}_k^N)^T$, $\forall k$, satisfazem as equações $A_k\, x_k = b_k$, $k = 1,\ldots,K$, do problema (1.8). Além disso, como $\bar{x}_k^N$, $\forall k$, formam uma solução factível para o problema reduzido (3.9), segue-se que $(\bar{x}_k^B \,/\, \bar{x}_k^N)^T$, $\forall k$, também satisfazem as equações de acoplamento $\sum_{k=1}^{K} Q_k\, x_k = b_o$ do problema (1.8), já que as restrições do problema reduzido são obtidas da subs - tituição dos vetores x_k^B nas equações de acoplamento do pro - blema (1.8). Como os vetores $\bar{x}_k^B$, $k = 1,\ldots,K$, estão dentro de seus limites de canalização por hipótese, e $\bar{x}_k^N, k=1,\ldots,K$, também estão (são a solução ótima do problema reduzido), te- mos que $(\bar{x}_k^B \ / \ \bar{x}_k^N)^T$ formam uma solução factível para o proble ma (1.8).

Além disso, como $\bar{\pi}_k$ e $\bar{\mu}_k$, $\forall k$, são dados por (3.13) e (3.14), respectivamente, temos que $\bar{\pi}_k$ e $\bar{\mu}_k$, $\forall k$ e $\bar{\Gamma}$ satisfazem as equações $\pi_k A_k - \mu_k + \Gamma Q_k \leq c_k$, $\forall k$, do problema dual com - pleto (3.5), ou seja,

$$\bar{\pi}_k A_k - \bar{\mu}_k + \bar{\Gamma} Q_k - c_k \leq 0 \text{ , pois,}$$

$$\bar{\pi}_k A_k - \bar{\mu}_k + \bar{\Gamma} Q_k - c_k = \bar{\pi}_k (A_k^B / A_k^N) - (\bar{\mu}_k^B / \bar{\mu}_k^N) + \bar{\Gamma}(Q_k^B / Q_k^N) - (c_k^B / c_k^N).$$

Substituindo os valores de $\bar{\pi}_k$ e $\bar{\mu}_k$, dados por (3.13) e (3.14), respectivamente, temos:

$$\bar{\pi}_k A_k - \bar{\mu}_k + \bar{\Gamma} Q_k - c_k =$$

$$= (c_k^B - \bar{\Gamma} Q_k^B) B_k^{-1} (A_k^B / A_k^N) - (0 / \bar{\varphi}_k) + \bar{\Gamma}(Q_k^B / Q_k^N) - (c_k^B / c_k^N) =$$

$$= [\,(c_k^B - \bar{\Gamma} Q_k^B) B_k^{-1} A_k^B - 0 + \bar{\Gamma} Q_k^B - c_k^B \Big/ (c_k^B - \bar{\Gamma} Q_k^B) B_k^{-1} \ A_k^N - \bar{\varphi}_k + \bar{\Gamma} Q_k^N - c_k^N]$$

e como $B_k^{-1} \ A_k^B = I$ e $B_k^{-1} \ A_k^N = \hat{A}_k$, temos que

$$\bar{\pi}_k A_k - \bar{\mu}_k + \bar{\Gamma} Q_k - c_k =$$

$$=[c_k^B-\overline{\Gamma}Q_k^B+\overline{\Gamma}Q_k^B-c_k^B \;/\; \overline{\Gamma}(Q_k^N-Q_k^B\,\hat{A}_k) - \overline{\varphi}_k - c_k^N+c_k^B\,\hat{A}_k] =$$

$$=[0 \;/\; \overline{\Gamma}(Q_k^N-Q_k^B\,\hat{A}_k) - \overline{\varphi}_k - (c_k^N - c_k^B\,\hat{A}_k)] \leq [0 \;/\; 0]$$

onde a parte básica é satisfeita com igualdade e a parte não básica fica satisfeita porque $\overline{\Gamma}$ e φ_k, $\forall k$, são a solução ótima de (3.12) e, portanto, satisfazem as restrições

$$\Gamma(Q_k^N - Q_k^B\,\hat{A}_k) - \varphi_k \leq c_k^N - c_k^B\,\hat{A}_k \quad , \; k = 1,\ldots,K.$$

A restrição $\mu_k \geq 0$, $k = 1,\ldots,K$, do problema dual completo (3.5), também fica satisfeita para $\overline{\mu}_k = (\overline{\mu}_k^B \;/\; \overline{\mu}_k^N)$, uma vez que $\overline{\mu}_k^B = 0$, $\forall k$ e $\overline{\mu}_k^N = \overline{\varphi}_k$ (temos que $\overline{\varphi}_k \geq 0$, $\forall k$, pois φ_k, $k = 1,\ldots,K$ formam parte da solução dual do problema reduzido).

Vamos agora mostrar que, para a solução $(\overline{x}_k^B \;/\; \overline{x}_k^N)^T$, o valor da função objetivo do problema primal (1.8) é igual ao valor da função objetivo do problema dual (3.5) com a solução $(\overline{\pi}_k \;/\; \overline{\mu}_k \;/\; \overline{\Gamma})$.

A função objetivo do problema dual completo (3.5) é

$$\sum_{k=1}^{K} (\pi_k b_k - \mu_k h_k) + \Gamma b_o \;.$$

Para os valores $\overline{\pi}_k$, $\overline{\mu}_k$ e $\overline{\Gamma}$, temos:

$$\sum_{k=1}^{K} (\overline{\pi}_k b_k - \overline{\mu}_k h_k) + \overline{\Gamma} b_o \;, \text{ ou,}$$

$$\sum_{k=1}^{K} [\; \overline{\pi}_k b_k - (\overline{\mu}_k^B \;/\; \overline{\mu}_k^N) \begin{pmatrix} h_k^B \\ \hline h_k^N \end{pmatrix}] + \overline{\Gamma}\, b_o \;.$$

Substituindo os valores de $\bar{\pi}_k$ e $\bar{\mu}_k$, $\forall k$, dados por (3.13) e (3.14), respectivamente, temos:

$$\sum_{k=1}^{K} \left[(c_k^B - \bar{\Gamma} Q_k^B)\, B_k^{-1}\, b_k - (0/\bar{\varphi}_k) \begin{pmatrix} h_k^B \\ \hline h_k^N \end{pmatrix} \right] + \bar{\Gamma}\, b_o =$$

$$= \sum_{k=1}^{K} \left[c_k^B\, B_k^{-1}\, b_k - \bar{\Gamma}\, Q_k^B\, B_k^{-1}\, b_k - \bar{\varphi}_k\, h_k^N \right] + \bar{\Gamma}\, b_o =$$

$$= \bar{\Gamma}\, (b_o - \sum_{k=1}^{K} Q_k^B\, B_k^{-1}\, b_k) + \sum_{k=1}^{K} (c_k^B\, B_k^{-1}\, b_k - \bar{\varphi}_k\, h_k^N) \quad (3.15)$$

Como (3.9) e (3.12) são problemas duais, na solução ótima do problema reduzido temos que

$$\bar{\Gamma}\, (b_o - \sum_{k=1}^{K} Q_k^B\, B_k^{-1}\, b_k) - \sum_{k=1}^{K} \bar{\varphi}_k h_k^N = \sum_{k=1}^{K} (c^N - c_k^B\, \hat{A}_k)\, \bar{x}_k^N \text{ ,ou seja,}$$

$$\bar{\Gamma}\, (b_o - \sum_{k=1}^{K} Q_k^B\, B_k^{-1}\, b_k) = \sum_{k=1}^{K} \bar{\varphi}_k\, h_k^N + \sum_{k=1}^{K} (c_k^N - c_k^B\, \hat{A}_k)\, \bar{x}_k^N \; .$$

Substituindo este resultado em (3.15), obtemos:

$$\sum_{k=1}^{K} (\bar{\pi}_k\, b_k - \bar{\mu}_k\, h_k) + \bar{\Gamma}\, b_o =$$

$$= \sum_{k=1}^{K} \bar{\varphi}_k\, h_k^N + \sum_{k=1}^{K} (c_k^N - c_k^B\, \hat{A}_k)\, \bar{x}_k^N + \sum_{k=1}^{K} c_k^B\, B_k^{-1}\, b_k - \sum_{k=1}^{K} \bar{\varphi}_k\, h_k^N =$$

$$= \sum_{k=1}^{K} c_k^N\, \bar{x}_k^N + \sum_{k=1}^{K} c_k^B\, (B_k^{-1}\, b_k - \hat{A}_k\, \bar{x}_k^N)$$

e como $B_k^{-1}\, b_k - \hat{A}_k\, \bar{x}_k^N = \bar{x}_k^B$, por (3.11), temos que

$$\sum_{k=1}^{K} (\bar{\pi}_k b_k - \bar{\mu}_k h_k) + \bar{\Gamma}\, b_o = \sum_{k=1}^{K} c_k^N \, \bar{x}_k^N + \sum_{k=1}^{K} c_k^B \, \bar{x}_k^B = \sum_{k=1}^{K} c_k \bar{x}_k \quad .$$

ii) A volta é imediata, pois se alguma componente de $\bar{x}_k^B$, $\forall k$, estiver fora de seus limites de canalização, não teremos uma solução factível para o problema (1.8).

3.6 - <u>CORREÇÃO DOS BLOCOS NÃO ÓTIMOS</u>:

Quando o teste de otimalidade não é satisfeito, devemos fazer uma correção em cada bloco k no qual as restrições $0 \leqslant \bar{x}_k^B \leqslant h_k^B$ foram violadas. A correção a ser feita é uma mudança de base neste bloco k, de modo que uma variável básica que foi violada saia da base, e uma variável não básica deste bloco entre na base. Esta mudança de base é mostrada através do seguinte teorema:

<u>Teorema 3.2</u> - se, para algum bloco k, $\bar{x}_k^B$ possui uma ou mais componentes violadas (ou abaixo do limite inferior ou acima do limite superior), uma sequência de operações de pivotamento pode ser feita no bloco k de maneira que pelo menos uma componente de x_k^N que é diferente de $\hat{x}_k^N$ em $\bar{x}_k^N$, entre na base do bloco k e pelo menos uma componente de x_k^B que foi violada em $\bar{x}_k^B$ saia da base.

<u>Prova</u>:

Para simplificarmos a notação, não escreveremos aqui o subscrito k, já que estamos tratando de um bloco k em particular. Usaremos subscritos para indicar as componentes do bloco k.

Sejam:

$\overset{o}{x} = (\overset{o}{x}^B \;/\; \overset{o}{x}^N)^T$: solução ótima encontrada pelo subproblema k;

$\bar{x} = (\bar{x}^B \;/\; \bar{x}^N)^T$: como definido anteriormente.

Temos disponíveis, duas soluções para o sistema

$$\left\{\begin{array}{c} A^B\, x^B + A^N\, x^N = b \\ 0 \leqslant x^B \leqslant h^B \\ 0 \leqslant x^N \leqslant h^N \end{array}\right\} \qquad (3.16)$$

que são:

$(\overset{o}{x}^B \;/\; \overset{o}{x}^N)^T$: solução factível; todas as componentes estão dentro de seus limites de canalização;

$(\bar{x}^B \;/\; \bar{x}^N)^T$: solução infactível, pois algumas componentes de $\bar{x}^B$ estão fora de seus limites de canalização.

Então, todos os vetores da forma:

$$(1-\theta)\begin{pmatrix} \overset{o}{x}^B \\ \hline \overset{o}{x}^N \end{pmatrix} + \theta\begin{pmatrix} \bar{x}^B \\ \hline \bar{x}^N \end{pmatrix} = \begin{pmatrix} \overset{o}{x}^B + \theta\,(\bar{x}^B - \overset{o}{x}^B) \\ \hline \overset{o}{x}^N + \theta\,(\bar{x}^N - \overset{o}{x}^B) \end{pmatrix} \qquad (3.17)$$

também satisfazem $A^B x^B + A^N x^N = b$, com $0 \leqslant x^N \leqslant h^N$. Quando $\theta=0$, esta solução é factível para o sistema (3.16) enquanto que, para $\theta=1$, ela é infactível. Vamos estudar para que valores de θ esta solução é factível. Analisaremos, em primeiro lugar, a parte básica de (3.17), isto é,

$$\overset{o}{x}^B + \theta(\bar{x}^B - \overset{o}{x}^B). \qquad (3.18)$$

Definimos:

$II = \{i \ / \ \bar{x}_i^B < 0\}$

$IS = \{i \ / \ \bar{x}_i^B > h_i^B\}$.

Sabemos que pelo menos um destes dois conjuntos não é vazio, senão o bloco já seria ótimo. Vamos analisar os dois tipos de infactibilidade que podem ocorrer.

1º) Suponhamos que II não é vazio, isto é, $\exists\, i/\bar{x}_i^B < 0$. O maior θ que mantém (3.18) não negativo é limitado pelas componentes de (3.18) com índices $i \in II$. Colocando estas componentes iguais a zero, temos:

$$\overset{o}{x}{}_i^B + \theta\,(\bar{x}_i^B - \overset{o}{x}{}_i^B) = 0 \quad , \ i \in II \ , \text{ ou,}$$

$$\theta_i = \frac{\overset{o}{x}{}_i^B}{\overset{o}{x}{}_i^B - \bar{x}_i^B} \quad , \ i \in II \tag{3.19}$$

Como $\overset{o}{x}{}^B$ é factível, $0 \leq \overset{o}{x}{}_i^B \leq h_i^B$, $\forall i$, e como $\bar{x}_i^B < 0$, para $i \in II$, temos que $0 \leq \theta_i < 1$. Queremos que $\overset{o}{x}{}^B + \theta(\bar{x}^B - \overset{o}{x}{}^B) \geq 0$; então o maior θ que mantém esta condição é

$$\theta_r = \min_{i \in II} \ \{\theta_i\} \tag{3.20}$$

2º) Suponhamos que IS não é vazio, isto é, $\exists i/\ \bar{x}_i^B > h_i^B$. Queremos encontrar o maior θ que mantém $\overset{o}{x}{}^B + \theta\,(\bar{x}^B - \overset{o}{x}{}^B) \leq h^B$. Neste caso, θ será limitado pelas componentes de (3.18) com índices $i \in IS$. Colocando estas componentes iguais a

h_i^B , $i \in IS$, temos:

$$\overset{o}{x}{}_i^B + \theta\,(\bar{x}_i^B - \overset{o}{x}{}_i^B) = h_i^B \ , \ i \in IS \ , \text{ ou,}$$

$$\theta_i = \frac{\overset{o}{x}{}_i^B - h_i^B}{\overset{o}{x}{}_i^B - \bar{x}_i^B} \ , \ i \in IS \tag{3.21}$$

Como $0 \leq \overset{o}{x}{}_i^B \leq h_i^B$, $\forall i$, então o numerador de (3.21) é não positivo, e como $\bar{x}_i^B > h_i^B$, para $i \in IS$, o denominador de (3.21) é negativo, temos também, neste caso, $0 \leq \theta_i < 1$. Então, o maior θ que mantém $\overset{o}{x}{}^B + \theta\,(\bar{x}^B - \overset{o}{x}{}^B) \leq h^B$ é dado por

$$\theta_s = \min_{i \in IS} \{\theta_i\} \tag{3.22}$$

Para $0 \leq \theta < 1$, podemos verificar que a parte não básica de (3.17) será factível, isto é, estará entre zero e h^N. Lembrando que $\overset{o}{x}{}^N$ e $\bar{x}^N$ são factíveis e que $\overset{o}{x}{}_i^N$, por ser uma componente não básica da solução do subproblema k, ou vale zero ou h_i^N , e escrevendo por componentes, temos que:

i) $\underbrace{\underbrace{(1-\theta)}_{>0}\ \overset{o}{x}{}_i^N}_{\geq 0} + \underbrace{\theta\ \underbrace{\bar{x}_i^N}_{\geq 0}}_{\geq 0} \geq 0 \ , \ \forall i$.

ii) $(1-\theta)\ \overset{o}{x}{}_i^N + \theta\ \bar{x}_i^N \leq h_i^N$, $\forall i$, pois:

se $\overset{o}{x}{}_i^N = 0 \Rightarrow \theta\ \bar{x}_i^N \leq h_i^N$ porque $\bar{x}_i^N \leq h_i^N$;

$$\text{se } \overset{o}{x}{}_i^N = h_i^N \Longrightarrow (1-\theta)\, h_i^N + \theta\, \bar{x}_i^N = h_i^N + \theta\, (\bar{x}_i^N - h_i^N) \quad ,$$

$$\text{e como } \bar{x}_i^N - h_i^N \leq 0 \text{ , então } h_i^N + \theta\, (\bar{x}_i^N - h_i^N) \leq h_i^N \quad .$$

Então, como queremos θ tal que a solução (3.17) seja factível, escolhemos

$$\theta = \min \{\theta_r, \theta_s\} \tag{3.23}$$

Tendo determinado θ, sabemos então que, ou a variável x_r^B ou a variável x_s^B sairá da base do bloco k, dependendo se $\theta = \theta_r$ ou $\theta = \theta_s$, respectivamente.

Seja J o conjunto dos índices das variáveis do problema reduzido, relativas ao bloco k, que voltaram com valores diferentes dos anteriores ($\overset{o}{x}{}^N$) na solução $\bar{x}^N$ (básicas no problema reduzido ou não básicas em limites opostos aos que possuíam nos blocos) e $\bar{J}$ o conjunto dos índices das variáveis que permaneceram inalteradas (nos mesmos limites) depois da resolução do problema reduzido, isto é,

$$J = \{j \;/\; \bar{x}_j^N \neq \overset{o}{x}{}_j^N\} \quad \text{e} \quad \bar{J} = \{j \;/\; \bar{x}_j^N = \overset{o}{x}{}_j^N\} \;.$$

Com os conjuntos J e $\bar{J}$, podemos particionar x^N em $(x^J / x^{\bar{J}})^T$ e A^N em $(A^J / A^{\bar{J}})$ e reescrever $\bar{x}^B$ da seguinte maneira:

$$\bar{x}^B = B^{-1}b - \hat{A}\, \bar{x}^N = \overset{o}{x}{}^B + \hat{A}\, \overset{o}{x}{}^N - \hat{A}\, \bar{x}^N =$$

$$= \overset{o}{x}{}^B + (\hat{A}^J / \hat{A}^{\bar{J}}) \begin{pmatrix} \overset{o}{x}{}^J \\ \hline \overset{o}{x}{}^{\bar{J}} \end{pmatrix} - (\hat{A}^J / \hat{A}^{\bar{J}}) \begin{pmatrix} \bar{x}^J \\ \hline \bar{x}^{\bar{J}} \end{pmatrix} =$$

$$= \overset{o}{x}^B + \hat{A}^J \overset{o}{x}^J - \hat{A}^J \bar{x}^J + \hat{A}^{\bar{J}} \overset{o}{x}^{\bar{J}} - \hat{A}^{\bar{J}} \bar{x}^{\bar{J}} \quad , \text{ ou ,}$$

$$\bar{x}^B = \overset{o}{x}^B + \hat{A}^J \; (\overset{o}{x}^J - \bar{x}^J) \quad .$$

O conjunto $\bar{J}$ pode ser vazio, mas J não o será, senão teríamos $\bar{x}^B = \overset{o}{x}^B$ e o bloco já seria ótimo.

Quando $\theta = \theta_r$, (3.17) satisfaz:

$$A^B[\overset{o}{x}^B+\theta_r(\bar{x}^B-\overset{o}{x}^B)]+\sum_{j\in J} A^J[\overset{o}{x}_j^N +\theta_r(\bar{x}_j^N-\overset{o}{x}_j^N)]+\sum_{j\in\bar{J}} A^J[\overset{o}{x}_j^N+\theta_r(\bar{x}_j^N-\overset{o}{x}_j^N)] = b$$

onde A^j é a j-ésima coluna de $A^N = (A^J \;/\; A^{\bar{J}})$. Como $\bar{x}_j^N = \overset{o}{x}_j^N$ para $\forall j \in \bar{J}$, o sistema anterior pode ser escrito como:

$$A^B[\; \overset{o}{x}^B+\theta_r(\bar{x}^B-\overset{o}{x}^B) \;]+\sum_{j\in J} A^J \;[\; \overset{o}{x}_j^N+\theta_r(\bar{x}_j^N-\overset{o}{x}_j^N) \;]+\sum_{j\in\bar{J}} A^j \; \overset{o}{x}_j^N = b. \quad (3.24)$$

Pela definição de θ_r , o coeficiente da r-ésima coluna de A^B , digamos A^{Br} , em (3.24) é zero. Como A^B é m x m e não singular , então a matriz $(A^B \;/\; A^N)$ tem posto m. Vamos eliminar a coluna A^{Br} de (3.24) e examinar o posto do sistema resultante. Multiplicando o sistema, sem a coluna A^{Br} , por B^{-1} , temos:

$$\sum_{\substack{j=1\\ j\neq r}}^{m} [\overset{o}{x}_j^B+\theta_r(\bar{x}_j^B-\overset{o}{x}_j^B)]e_j+\sum_{j\in J}\hat{A}^j[\overset{o}{x}_j^N+\theta_r(\bar{x}_j^N-\overset{o}{x}_j^N)] = B^{-1}b - \sum_{j\in\bar{J}} \hat{A}^j\overset{o}{x}_j^N$$

onde: $e_j = (0\ldots0\underset{\text{posição } j}{1}0\ldots0)^T$

$\hat{A}^J$: j - ésima coluna de $\hat{A} = B^{-1}A^N$.

O sistema anterior tem o mesmo posto que o sistema (3.24) sem a coluna A^{B_r}. Além disso, pelo menos uma coluna $\hat{A}^j$, com $j \in J$, digamos $\hat{A}^*$, tem sua r-ésima componente diferente de zero. Isto ocorre porque, para $i \in II$, o vetor

$$\bar{x}^B = \overset{o}{x}{}^B + \hat{A}^J (\overset{o}{x}{}^J - \bar{x}^J)$$

possui componentes negativas. Como $r \in II$, se todas as componentes de $\hat{A}$ na linha r e colunas $j \in J$ fossem zero, então

$$\bar{x}_r^B = \overset{o}{x}{}_r^B \geq 0$$

o que é uma contradição. Assim, os vetores e_j, $j = 1,\ldots,m$, $j \neq r$ e $\hat{A}^*$ são linearmente independentes; logo o sistema (3.24) sem a coluna A^{B_r} tem posto m. Como este sistema possui uma solução factível, então ele possui uma solução <u>básica</u> factível [Luenberger, 1973, pp. 18]. Esta solução não contém a coluna A^{B_r} e deve conter colunas A^j tais que $j \in J$.

O mesmo desenvolvimento pode ser feito quando $\theta = \theta_s$. Também neste caso, existe uma coluna $\hat{A}^*$ que possui a s-ésima componente diferente de zero, pois, para $i \in IS$, o vetor

$$\bar{x}^B = \overset{o}{x}{}^B + \hat{A}^J (\overset{o}{x}{}^J - \bar{x}^J)$$

possui componentes que excedem os limites superiores h_i para $i \in IS$, e como $s \in IS$, se todas as componentes de $\hat{A}$ na linha s e colunas $j \in J$ fossem zero, então

$$\bar{x}_s^B = \overset{o}{x}{}_s^B \leq h_s^B$$

o que é uma contradição. Então, também neste caso, podemos obter uma solução básica que não contém a coluna A^{B_s} e que deve conter as colunas A^j tais que $j \in J$, ficando assim provado o teorema.

Convém notar que o teorema anterior não afirma que com uma única operação de pivotamento, podemos obter a nova base do bloco k. Contudo, esta nova base pode ser encontrada facilmente, resolvendo-se um problema de Programação Linear com variáveis canalizadas para cada bloco não ótimo. Dois casos podem ocorrer (continuamos omitindo aqui o subscrito k referente ao bloco; os subscritos que aparecem se referem a componentes de um bloco):

1º) $\theta = \theta_r$, então x_r^B foi escolhida para sair da base do bloco k (na nova solução $\bar{x}^B$, a variável x_r^B ficou abaixo de seu limite inferior). Pela definição de θ_r, sabemos que a r-ésima componente de $\overset{o}{x}{}^B + \theta_r (\bar{x}^B - \overset{o}{x}{}^B)$ é zero. Então, como x_r^B é uma variável básica, vamos minimizar x_r^B, isto é x_r^B passará a ser uma variável não básica no limite inferior (zero). Para isso, resolvemos o seguinte subproblema:

$$\left\{\begin{array}{l} \min\ x_r^B \\ \text{s.a.} \\ \qquad x^B + \sum_{j \in J} \hat{A}^j x_j^N = \overset{o}{x}{}^B \\ \qquad 0 \leq x^B \leq h^B \\ \qquad 0 \leq x_j^N \leq h_j^N \quad, j \in J \end{array}\right\} \qquad (3.25)$$

onde $\hat{A} = B^{-1} A^N$, mas só estamos considerando as colunas j de $\hat{A}$ tais que $j \in J$.

Pelo teorema anterior, sabemos que no final da resolução do subproblema (3.25), x_r^B será uma variável não básica no limite inferior e,pelo menos um x_j^N , $j \in J$, estará na base deste subproblema.

2º) $\theta = \theta_s$, então x_s^B foi escolhida para sair da base do bloco k (na nova solução $\bar{x}^B$, a variável x_s^B ficou acima de seu limite superior). Pela definição de θ_s , sabemos que a s-ésima componente de $\overset{o}{x}^B + \theta_s (\bar{x}^B - \overset{o}{x}^B)$ vale h_s^B . Como x_s^B é uma variável básica, vamos elevar x_s^B de seu valor até seu limite superior h_s^B , isto é, x_s^B passará a ser uma variável não básica no limite superior (h_s^B). Resolvemos então o seguinte subproblema:

$$\left\{\begin{array}{l} \max\ x_s^B \\ \text{s.a.} \\ \quad x^B + \sum_{j \in J} \hat{A}^j x_j^N = \overset{o}{x}^B \\ \quad 0 \leq x^B \leq h^B \\ \quad 0 \leq x_j^N \leq h_j^N \quad , \quad j \in J \end{array}\right\} \qquad (3.26)$$

Ainda pelo teorema anterior, sabemos que ao final da resolução do subproblema (3.26), x_s^B será uma variável não básica no limite superior e, pelo menos um x_j^N , $j \in J$, estará na base deste subproblema.

Nos dois casos, já temos uma base inicial para o subproblema k, que é a base ótima da iteração anterior (ver apêndice B) e, geralmente, com apenas um pequeno número de pivotamentos chegamos ao ótimo destes subproblemas, que são resolvidos pelo Mé todo Simplex Revisado com variáveis canalizadas.

Os procedimentos de encontrar θ (determinação da variável que sai da base do bloco em questão) e resolver ou o subproblema (3.25) ($\theta = \theta_r$) ou o subproblema (3.26) ($\theta = \theta_s$) , são feitos para todo bloco k não ótimo.

Obtidas as novas bases de cada bloco k, k = 1,..., K (se um determinado bloco k já era ótimo, sua base permanece inalterada para a próxima iteração), fazemos novamente as partições em parte básica e parte não básica e formamos um novo problema reduzido, começando então uma nova iteração.

Este processo é repetido até que o teste de otimalidade seja satisfeito (teorema 3.1).

3.7 - <u>ALGORÍTMO</u>:

(0) Arbitrar $\Gamma = \overset{o}{\Gamma}$ e resolver os K subproblemas (3.2),

$$\left\{\begin{array}{l} \min\ (c_k - \Gamma Q_k)\ x_k \\ \text{s.a.} \\ \qquad A_k\ x_k = b_k \\ \qquad 0 \leq x_k \leq h_k \end{array}\right\} \quad k = 1,\ldots,K$$

pelo Método Simplex Revisado com variáveis canalizadas, obtendo suas soluções ótimas.

(1) Para cada bloco k, k = 1,...,K , obter as seguintes partições:

$$x_k = (x_k^B \ /\ x_k^N)^T$$

$$A_k = (A_k^B \,/\, A_k^N) \quad e \quad B_k^{-1}$$

$$h_k = (h_k^B \,/\, h_k^N)^T$$

$$c_k = (c_k^B \,/\, c_k^N)$$

$$Q_k = (Q_k^B \,/\, Q_k^N)$$

(2) Substituindo a equação (3.6) nas restrições de acoplamento e na função objetivo do problema (1.8), montar o problema reduzido (3.9):

$$\begin{cases} \min \sum_{k=1}^{K} (c_k^N - c_k^B \hat{A}_k)\, x_k^N \\ \text{s.a.} \\ \quad \sum_{k=1}^{K} (Q_k^N - Q_k^B \hat{A}_k)\, x_k^N = b_o - \sum_{k=1}^{K} Q_k^B B_k^{-1} b_k \\ \quad 0 \leq x_k^N \leq h_k^N \quad , \quad k = 1,\ldots,K \end{cases}$$

onde $\hat{A}_k = B_k^{-1} A_k^N, k = 1,\ldots,K$.

(3) Resolver o problema reduzido pelo Método Simplex Revisado com variáveis canalizadas, obtendo como solução ótima para as variáveis x_k^N , os valores $\bar{x}_k^N$, $k = 1,\ldots,K$.

(4) <u>Teste de otimalidade</u>:

Para $k = 1,\ldots,K$, obter:

$$\bar{x}_k^B = B_k^{-1} b_k - \hat{A}_k \bar{x}_k^N \quad .$$

Se $0 \leq \bar{x}_k^B \leq h_k^B$, $\forall k$, então a solução ótima do problema ori -

ginal (1.8) é dada por $\bar{x}_k = (\bar{x}_k^B \ / \ \bar{x}_k^N)^T$, $k = 1,\ldots, K$ e a solução ótima do problema dual (3.5) é dada por $(\bar{\pi}_k, \bar{\mu}_k, \bar{\Gamma})$, $k = 1,\ldots,K$, onde $\bar{\pi}_k$ e $\bar{\mu}_k$ são dados por (3.13) e (3.14), respectivamente, e $\bar{\Gamma}$ é a parte da solução dual, referente às restrições $\sum_{k=1}^{K} (Q_k^N - Q_k^B \hat{A}_k)\ x_k^N = b_o - \sum_{k=1}^{K} Q_k^B\ B_k^{-1}\ b_k$, obtida do último problema reduzido (3.9). Fim.

Correção dos blocos não ótimos:

(5) Determinação da variável de saída do bloco k:

Para todo bloco k onde $0 \leq \bar{x}_k^B \leq h_k^B$ não foi satisfeita, obter (não vamos escrever agora o subscrito k do bloco; o subscrito j se refere às componentes do bloco em questão):

(5.1) $II = \{i \ / \ \bar{x}_i^B < 0\}$;

$$\theta_i = \frac{\overset{o}{x}{}_i^B}{\overset{o}{x}{}_i^B - \bar{x}_i^B} \ , \quad i \in II \ ; \tag{3.19}$$

$$\theta_r = \min_{i \in II} \ \{\theta_i\} \ ; \tag{3.20}$$

(5.2) $IS = \{i \ / \ \bar{x}_i^B > h_i^B\}$;

$$\theta_i = \frac{\overset{o}{x}{}_i^B - h_i^B}{\overset{o}{x}{}_i^B - \bar{x}_i^B} \ , \quad i \in IS \ ; \tag{3.21}$$

$$\theta_s = \min_{i \in IS} \ \{\theta_i\} \ ; \tag{3.22}$$

(5.3) $$\theta = \min \ \{\theta_r, \theta_s\} \ . \tag{3.23}$$

(6) <u>Obtenção das novas bases dos blocos</u>:

Para cada bloco k não ótimo (novamente, o subscrito j se refere às componentes do bloco em questão), obter:

$$J = \{j \; / \; \bar{x}_j^N \neq \overset{o}{x}{}_j^N\}$$

(6.1) Se $\theta = \theta_r$, resolver o subproblema (3.25):

$$\begin{cases} \min x_r^B \\ \text{s.a.} \\ \quad x^B + \sum_{j \in J} \hat{A}^j x_j^N = \overset{o}{x}{}^B \\ \quad 0 \leq x^B \leq h^B \\ \quad 0 \leq x_j^N \leq h_j^N \; , \; j \in J \end{cases}$$

onde $\hat{A}^J$ é a j-ésima coluna de $\hat{A}$;

(6.2) Se $\theta = \theta_s$, resolver o subproblema (3.26):

$$\begin{cases} \max x_s^B \\ \text{s.a.} \\ \quad x^B + \sum_{j \in J} \hat{A}^j x_j^N = \overset{o}{x}{}^B \\ \quad 0 \leq x^B \leq h^B \\ \quad 0 \leq x_j^N \leq h_j^N \; , \; j \in J \end{cases}$$

Obtidas as novas bases dos blocos, voltar ao passo(1).

Observações:

1) No passo (0) do algoritmo, em geral, aplicamos a fase I do Método Simplex para a obtenção de uma base inicial factível para os K subproblemas (se algum subproblema for infactível, o problema original (1.8) não terá solução). Nas iterações subsequentes (passo 6), podemos sempre resolver cada subproblema a partir da solução ótima obtida na iteração anterior.

2) Quando um bloco k é ótimo de uma iteração para a seguinte, não precisamos recalcular, com relação a este bloco, no problema reduzido, os coeficientes da função objetivo, $c_k^N - c_k^B \hat{A}_k$, os coeficientes das restrições, $Q_k^N - Q_k^B \hat{A}_k$ e a parte relativa a este bloco k no lado direito, $Q_k^B B_k^{-1} b_k$, já que a base deste bloco permaneceu inalterada.

3) Na resolução do problema reduzido, usamos o Método Simplex Revisado com variáveis canalizadas - aproveitamos a mesma subrotina para resolver este problema e os subproblemas, já que todos possuem variáveis canalizadas. No entanto, para resolvermos o problema reduzido, poderíamos ter utilizado o Método Dual Simplex com variáveis canalizadas, uma vez que algumas componentes do lado direito podem ser negativas, mas os coeficientes da função objetivo, $c_k^N - c_k^B \hat{A}_k$, são os próprios coeficientes de custos relativos de cada bloco k em sua solução ótima, estando, portanto, em suas respectivas condições de otimalidade.

4) Da maneira como foi feito, em toda iteração devemos aplicar

a fase I do Método Simplex para obtermos uma base inicial factível para o problema reduzido. Isto aumenta o trabalho realizado pelo método, diminuindo sua eficiência. Lasdon [1970, pp. 291 - 296], propõe uma maneira de utilizarmos a matriz básica do problema reduzido de uma iteração para a seguinte, isto é, com algumas operações de pivotamento, podemos obter a inversa da matriz básica inicial de um problema reduzido (B_*^{-1}) a partir da inversa obtida na solução ótima do problema reduzido anterior (B_o^{-1}), quando o problema reduzido é resolvido pelo Método Dual Simplex (B_o e B_* possuem os mesmos multiplicadores Simplex e B_* é uma base ótima, mas não factível, para o novo problema reduzido). Este procedimento não foi implementado nos programas que executam o Método de Rosen pelas dificuldades que encontramos em transpô-lo para o caso de problemas com variáveis canalizadas.

5) O seguinte teorema se encontra provado em Lasdon [1970, pp. 290 - 291] e em Rosen [1964, pp. 258 - 259] (não incluindo o caso explícito de variáveis canalizadas):

"O algorítmo descrito anteriormente fornece uma sequência de soluções factíveis para o problema dual completo (3.5) com um valor não decrescente para a função objetivo. Se o problema (3.5) tem uma solução ótima, ela é obtida em um número finito de iterações."

3.8 - O CASO PARTICULAR DO MÉTODO DE ROSEN APLICADO AO PROBLEMA DA OTIMIZAÇÃO GLOBAL DE RAÇÕES:

O problema da otimização global de rações, descrito no capítulo 1, foi transformado no problema (1.9) que é:

$$\begin{cases} \min \sum_{k=1}^{K} (q_k\ c/0) \begin{pmatrix} y'_k \\ \hline u'_k \end{pmatrix} \\ \text{s.a.} \\ \sum_{k=1}^{K} (Q_k \ / \ 0) \begin{pmatrix} y'_k \\ \hline u'_k \end{pmatrix} + \ I\ u'_{K+1} = b_o \\ (A \ / \ I) \begin{pmatrix} y'_k \\ \hline u'_k \end{pmatrix} = b'_k \ , \ k = 1,\dots,K \\ 0 \leq u'_k \leq \beta_k - \alpha_k \ , \ k = 1,\dots,K \\ 0 \leq u'_{K+1} \leq \ell - g \\ 0 \leq y'_k \leq \delta_k - \gamma_k \ , \ k = 1,\dots,K \end{cases}$$

Da mesma maneira que ocorreram simplificações quando aplicamos o Método de Dantzig-Wolfe ao problema acima, veremos que a aplicação do Método de Rosen a este problema também apresentará particularidades:

1ª) Como no Método de Dantzig-Wolfe, armazenamos apenas uma matriz A e, na resolução dos subproblemas, montamos a matriz (A/I).

2ª) O mesmo ocorre quanto aos coeficientes da função objetivo e às matrizes Q_k - armazenamos apenas um vetor c e K escalares q_k , $k = 1,\dots,K$. Assim, na montagem dos coeficientes da função objetivo para a resolução de cada subproblema (3.2), teremos:

$$\min q_k\ (c - \Gamma)\ y'_k \quad , \forall k.$$

3ª) Na montagem das restrições e da função objetivo do problema reduzido ocorrem simplificações, pois não armazenamos os zeros.

A parte referente a cada bloco k nos coeficientes das restrições do problema reduzido é

$$Q_k^N - Q_k^B \hat{A}_k \quad .$$

Como a matriz de acoplamento correspondente ao bloco k, agora é

$$(Q_k/0),$$

onde

$$Q_k = \begin{bmatrix} q_k & & & \\ & q_k & & \\ & & \ddots & \\ & & & q_k \end{bmatrix} = q_k I$$

e I é a matriz identidade de ordem n, ao fazermos a multiplicação $Q_k^B \hat{A}_k$ (note que, agora, $\hat{A}_k$ é a parte não básica atualizada de (A/I), sabemos que uma coluna de Q_k^B ou é uma coluna da matriz identidade multiplicada pelo escalar q_k , ou é uma coluna composta por zeros. O mesmo ocorre quando fazemos a subtração $Q_k^N - Q_k^B \hat{A}_k$— uma coluna de Q_k^N ou é toda nula ou é uma coluna da matriz identidade multiplicada por q_k. Assim, são efetuadas poucas operações na montagem das restrições do problema reduzido.

Da mesma maneira, a parte referente a cada bloco k nos coeficientes da função objetivo do problema reduzido, em vez de

$(c_k^N - c_k^B \hat{A}_k)$ será

$q_k \ (c_k^N - c_k^B \hat{A}_k)$

onde c_k^B é a parte básica e c_k^N é a parte não básica de $(c/0)$ correspondentes ao bloco k. Também neste caso, não efetuamos as multiplicações quando os coeficientes da função objetivo do bloco k são nulos.

Simplificações análogas ocorrem na construção do lado direito do problema reduzido — como são envolvidas as matrizes de acoplamento, as operações com elementos nulos também não são efetuadas.

4ª) O tratamento do chamado bloco K+1 também é feito de maneira independente, pela simplicidade deste bloco. Ele não possui coeficientes na função objetivo do problema (1.9), e suas restrições são apenas

$$0 \leq u'_{K+1} \leq \ell - g \text{ , ou,}$$

$$\begin{cases} u'_{K+1} \leq \ell - g \\ u'_{K+1} \geq 0 \end{cases}$$

Colocando as folgas u''_{K+1} , as restrições do bloco K+1 podem ser vistas como:

$$\begin{cases} u'_{K+1} + u''_{K+1} = \ell - g \\ u'_{K+1} \geq 0, \ u''_{K+1} \geq 0 \end{cases}$$

Quando resolvemos os K subproblemas (3.2), devemos resolver também o subproblema K+1, que terá como coeficientes em sua função objetivo:

$$[c_{K+1} - \Gamma\, Q_{K+1}]\, u'_{K+1} = [(0/0) - \Gamma(I/0)] \begin{pmatrix} u'_{K+1} \\ u''_{K+1} \end{pmatrix} = -\Gamma u'_{K+1} \quad .$$

O subproblema K+1 será então:

$$\begin{cases} \min & -\Gamma u'_{K+1} \\ \text{s.a.} & \\ & u'_{K+1} + u''_{K+1} = \ell - g \\ & u'_{K+1}\ ,\ u''_{K+1} \geq 0 \end{cases}$$

e, como no Método de Dantzig-Wolfe, terá uma solução analítica bastante simples: sendo Γ_i as componentes de Γ, u'_{K+1_i} as componentes de u'_{K+1} , u''_{K+1_i} as componentes de u''_{K+1} e $\ell_i - g_i$ as componentes de $\ell - g$, para $i = 1,\dots,N$, temos como solução ótima para o subproblema K+1:

$$\text{se } \Gamma_i > 0 \Longrightarrow \begin{cases} u'_{K+1_i} = \ell_i - g_i \\ u''_{K+1_i} = 0 \end{cases}$$

$$\text{se } \Gamma_i \leq 0 \longrightarrow \begin{cases} u'_{K+1_i} = 0 \\ u''_{K+1_i} = \ell_i - g_i \end{cases}$$

e as componentes básicas são aquelas diferentes de zero.

A parte referente ao bloco K+1 nos coeficientes das restri -

ções do problema reduzido será (a parte referente aos coeficientes da função objetivo será nula):

$$Q^N_{K+1} - Q^B_{K+1} \hat{A}_{K+1}$$

onde Q^N_{K+1} é a parte não básica e Q^B_{K+1} é a parte básica de (I/0). Além disso, $\hat{A}_{K+1}$ será a matriz identidade de ordem n. Isto ocorre porque $\hat{A}_{K+1}$ é a parte não básica atualizada de (I/I) e sabemos que, se uma componente i de u'_{K+1} é básica, a componente i de u''_{K+1} será não básica. Assim, a parte básica do bloco K+1, A^B_{K+1} , e a parte não básica, A^N_{K+1}, serão idênticas e serão compostas pelas colunas da matriz identidade (a ordem destas colunas irá depender dos índices das variáveis básicas). Temos então,

$$A^B_{K+1} = A^N_{K+1} \Longrightarrow \hat{A}_{K+1} = B^{-1}_{K+1} A^N_{K+1} = B^{-1}_{K+1} A^B_{K+1} = I ,$$

onde B^{-1}_{K+1} é a matriz inversa da base do bloco K+1.

Por este motivo, a correção do bloco K+1 também será simples. Este bloco será corrigido quando pelo menos uma componente de

$$\bar{\nu}^B_{K+1} = B^{-1}_{K+1} (\ell - g) - \hat{A}_{K+1} \bar{\nu}^N_{K+1}$$

for negativa (u'_{K+1} e u''_{K+1} não são canalizadas), onde:

$$\nu_{K+1} = (u'_{K+1} / u''_{K+1})^T \quad ;$$

$\bar{\nu}^B_{K+1}$: parte básica do bloco K+1 corrigida;

$\bar{\nu}^N_{K+1}$: parte da solução do problema reduzido referente ao bloco K+1;

$\hat{A}_{K+1} = I$.

Neste caso, determinamos somente o valor de θ_r (conforme 3.19) e a componente r deste bloco sairá da base. Se a r-ésima variável básica for uma componente de u'_{K+1} , a componente correspondente em u''_{K+1} entrará na base deste bloco tomando o lugar da r-ésima variável básica, e vice-versa, já que não podemos ter, simultaneamente, uma componente i de u'_{K+1} e de u''_{K+1} na base. Em qualquer dos casos, fazemos apenas a mudança das componentes da base — o valor da solução não irá se alterar.

5ª) O mesmo teste de infactibilidade que foi descrito na aplicação do Método de Dantzig-Wolfe ao problema da otimização global de rações pode ser feito quando aplicamos o Método de Rosen ao problema.

Por estes motivos, fizemos também um programa especial que resolve o problema da otimização global de rações pelo Método de Rosen.

CAPÍTULO 4

EXPERIÊNCIAS COMPUTACIONAIS E CONCLUSÕES

Foram programados, usando o computador PDP-10 da DIGITAL e o compilador FORTRAN IV, o Método de Dantzig - Wolfe , descrito no capítulo 2 e o Método de Rosen, descrito no capítulo 3 e no apêndice B, com versões para problemas gerais e para o caso específico da otimização global de rações. Estes programas são:

RROSEN. F4 - Método de Rosen para problemas gerais.

DANWO. F4 - Método de Dantzig-Wolfe para problemas gerais, efetuando apenas um pivotamento no problema mestre por iteração.

RAROSN. F4 - Método de Rosen para o problema da otimização global de rações.

RADW1. F4 - Método de Dantzig-Wolfe para o problema da otimização global de rações, efetuando apenas um pivotamento no problema mestre por iteração.

RADW2. F4 - Método de Dantzig-Wolfe para o problema da otimização global de rações, resolvendo o problema mestre restrito em cada iteração.

Todos os programas consideram que cada bloco pode possuir ou não variáveis canalizadas e, se uma variável é canalizada, ela está necessariamente entre dois limites. Além disso, os próprios programas fazem a transformação nas variáveis canalizadas descrita no capítulo 1, colocando o limite inferior de uma variável canalizada sempre igual a zero.

Estes programas utilizam a subrotina SUBPR.F4, que resolve um problema de Programação Linear pelo Método Simplex Revisado com variáveis canalizadas (é chamada para resolver ou um subproblema ou o problema reduzido do Método de Rosen). O algorítmo utilizado por esta subrotina é, basicamente, o descrito no apêndice A, levando-se em conta que as variáveis podem ser canalizadas ou não (neste caso, não negativas), e incluindo a fase I do Método Simplex.

Os programas RROSEN.F4 e RAROSN.F4 utilizam as partições e a mudança de variáveis descritas no apêndice B. Os programas DANWO.F4 e RADW1.F4 fazem apenas um pivotamento no problema mestre em cada iteração, ao passo que RADW2.F4 monta e resolve o problema mestre restrito em cada iteração. Neste caso, estamos considerando que o problema mestre restrito possui as colunas básicas e mais K+1 colunas, cada uma gerada por um subproblema.

Os programas para problemas gerais (RROSEN.F4 e DANWO.F4) supõem que os problemas já se encontram no seguinte formato:

$$
\left\{
\begin{array}{l}
\min \sum_{k=1}^{K} c_k \, x_k \\
\text{s.a.} \\
\qquad \sum_{k=1}^{K} Q_k \, x_k = b_o \\
\qquad\qquad A_k \, x_k = b_k \ , \ k = 1,\dots,K \\
\qquad\qquad\quad x_{kj} \geq 0, \quad j \in I1_k \ , \ k = 1,\dots,K \\
\qquad \gamma_{kj} \leq x_{kj} \leq \delta_{kj}, \ j \in I2_k \ , \ k = 1,\dots,K \\
\qquad b_o \geq 0, \ b_k \geq 0, \ \forall k, \gamma_{kj}, \ \delta_{kj} \geq 0, \ \forall k, \ \forall j
\end{array}
\right.
$$

Os programas específicos para as rações (RAROSN.F4 , RADW1.F4 e RADW2.F4) supõem que os problemas estão no seguinte formato:

$$\left[\begin{array}{l} \min \sum_{k=1}^{K} q_k\, c\, x_k \\ \text{s.a.} \\ \quad g \leq \sum_{k=1}^{K} q_k\, x_k \leq \ell \\ \quad \alpha_k \leq A\, x_k \leq \beta_k \;,\; k = 1,\ldots,K \\ \qquad x_{kj} \geq 0 \;,\; j \in I1_k \;,\; k = 1,\ldots,K \\ \quad \gamma_{kj} \leq x_{kj} \leq \delta_{kj} \;,\; j \in I2_k \;,\; k = 1,\ldots,K \\ \quad g,\ \ell \geq 0 \\ \quad \alpha_k,\ \beta_k,\ q_k \geq 0,\ k = 1,\ldots,K \\ \quad \gamma_{kj},\ \delta_{kj} \geq 0,\ \forall k,\ \forall j \end{array}\right.$$

4.1 - <u>TESTES REALIZADOS</u>:

4.1.1 - <u>Alguns testes para os programas RROSEN.F4 e DANWO.F4</u>:

Os programas RROSEN.F4 e DANWO.F4 foram testados com problemas pequenos, dos quais citaremos alguns exemplos:

1) $$\begin{cases}
\min z = -x_{11} - 2x_{12} - x_{21} - 3x_{22} \\
\text{s.a.} \\
x_{11} + x_{12} + x_{21} + x_{22} + x_{31} = 200 \\
x_{11} + 2x_{12} + x_{21} + 3x_{22} + x_{32} = 1400 \\
x_{11} + 2x_{12} + x_{13} = 100 \\
4x_{11} + 6x_{12} + x_{14} = 300 \\
3x_{21} + 6x_{22} + x_{23} = 600 \\
x_{31} + x_{33} = 200 \\
x_{32} + x_{34} = 1400 \\
x_{kj} \geq 0, \ \forall k, \forall j \ \text{e} \\
20 \leq x_{22} \leq 200 \\
0 \leq x_{33} \leq 200 \\
400 \leq x_{34} \leq 1000
\end{cases}$$

<u>Solução</u>: $z = -400$

$x_{11} = 0$; $x_{12} = 50$; $x_{13} = 0$; $x_{14} = 0$

$x_{21} = 0$; $x_{22} = 100$; $x_{23} = 0$

$x_{31} = 50$; $x_{32} = 1000$; $x_{33} = 150$; $x_{34} = 400$

2)
$$\begin{cases} \min z = -x_{11} - x_{12} - 2x_{21} - x_{22} \\ \text{s.a.} \\ x_{11} + 2x_{12} + 2x_{21} + x_{22} + x_{24} = 40 \\ x_{11} + 3x_{12} + x_{13} = 30 \\ 2x_{11} + x_{12} + x_{14} = 20 \\ x_{21} + x_{22} + x_{23} = 15 \\ x_{kj} \geq 0 \quad , \forall k, \forall j \quad e \\ 0 \leq x_{21} \leq 10 \\ 0 \leq x_{22} \leq 10 \\ 0 \leq x_{24} \leq 40 \end{cases}$$

<u>Solução:</u> $z = -36{,}6666$

$x_{11} = 8{,}3333$; $x_{12} = 3{,}3333$; $x_{13} = 11{,}6666$; $x_{14} = 0$

$x_{21} = 10$; $x_{22} = 5$; $x_{23} = 0$; $x_{24} = 0$

3)
$$
\left\{
\begin{array}{llll}
\min z = -2x_{11} - x_{12} - 3x_{13} & -2x_{21} - x_{22} & & \\
\text{s.a.} & & & \\
x_{11} + 2x_{12} + 4x_{13} & + 2x_{21} + 2x_{22} & + x_{31} & = 100 \\
x_{11} + x_{12} \qquad + x_{14} & & & = 40 \\
\qquad 2x_{12} + x_{13} \qquad + x_{15} & & & = 50 \\
 & 2x_{21} + x_{22} + x_{23} & & = 65 \\
 & x_{21} + x_{22} \qquad + x_{24} & & = 60 \\
 & & x_{31} + x_{32} & = 100 \\
 & & & \\
x_{kj} \geq 0,\ \forall k,\ \forall j \quad \text{e} & & & \\
0 \leq x_{32} \leq 100 & & &
\end{array}
\right.
$$

<u>Solução</u>: $z = -140$

$x_{11} = 40$; $x_{12} = 0$; $x_{13} = 0$; $x_{14} = 0$; $x_{15} = 50$

$x_{21} = 30$; $x_{22} = 0$; $x_{23} = 5$; $x_{24} = 30$

$x_{31} = 0$; $x_{32} = 100$

4)
$$
\left\{
\begin{array}{lllllll}
\min z = & -3x_{11} - 4x_{12} - x_{13} & -2x_{21} - 2x_{22} - x_{23} & & & & \\
\text{s.a.} & & & & & & \\
& x_{11} \quad -2x_{13} & + x_{21} \quad - x_{23} & + x_{31} & & & = 2{,}5 \\
& x_{11} - 2x_{12} & + 2x_{21} - x_{22} + x_{23} & \quad + x_{32} & & & = 100{,}5 \\
& x_{11} + x_{12} \quad + x_{14} & & & & & = 20 \\
& \quad x_{12} + x_{13} \quad + x_{15} & & & & & = 1 \\
& & x_{21} + x_{22} + x_{23} & & & & = 1 \\
& & & x_{31} \quad + x_{33} & & & = 2{,}5 \\
& & & \quad x_{32} \quad + x_{34} & & & = 100{,}5 \\
& & & & & & \\
& x_{kj} \geq 0, \ \forall k, \forall j \ \text{e} & & & & & \\
& 0 \leq x_{33} \leq 2{,}5 & & & & & \\
& 0{,}5 \leq x_{34} \leq 100 & & & & &
\end{array}
\right.
$$

<u>Solução</u>: $z = -18{,}5$

$x_{11} = 5{,}5$; $x_{12} = 0$; $x_{13} = 1$; $x_{14} = 14{,}5$; $x_{15} = 0$

$x_{21} = 0$; $x_{22} = 0$; $x_{23} = 1$

$x_{31} = 0$; $x_{32} = 94$; $x_{33} = 2{,}5$; $x_{34} = 6{,}5$

5)
$$
\begin{cases}
\min z = -x_{11} - 8x_{12} - 5x_{21} - 6x_{22} \\
\text{s.a.} \\
x_{11} + 4x_{12} + 5x_{21} + 2x_{22} + x_{31} = 7 \\
2x_{11} + 3x_{12} + x_{13} = 6 \\
5x_{11} + x_{12} + x_{14} = 5 \\
3x_{21} + 4x_{22} - x_{23} = 12 \\
x_{31} + x_{32} = 7 \\
x_{kj} \geqslant 0, \ \forall k, \ \forall j \ \text{ e} \\
0 \leqslant x_{21} \leqslant 4 \\
0 \leqslant x_{22} \leqslant 3 \\
0 \leqslant x_{32} \leqslant 7
\end{cases}
$$

<u>Solução:</u> $z = -20$

$x_{11} = 0$; $x_{12} = 0{,}25$; $x_{13} = 5{,}25$; $x_{14} = 4{,}75$

$x_{21} = 0$; $x_{22} = 3$; $x_{23} = 0$

$x_{31} = 0$; $x_{32} = 7$

6) $\min z = 7x_{11} + 8x_{12} + 7x_{21} + 8x_{22}$

s.a.

$$\left\{\begin{array}{lcl}
10x_{11} + 20x_{21} + x_{31} & = & 50 \\
10x_{12} + 20x_{22} + x_{32} & = & 60 \\
x_{11} + 2x_{12} + x_{13} & = & 8 \\
x_{12} + x_{14} & = & 5 \\
x_{21} + 2x_{22} + x_{23} & = & 11 \\
x_{22} + x_{24} & = & 7 \\
x_{31} + x_{33} & = & 50 \\
x_{32} + x_{34} & = & 60
\end{array}\right.$$

$x_{kj} \geqslant 0,\ \forall k,\ \forall j$ e

$2 \leqslant x_{13} \leqslant 6$

$1 \leqslant x_{14} \leqslant 4$

$3 \leqslant x_{23} \leqslant 8$

$1 \leqslant x_{24} \leqslant 6$

$0 \leqslant x_{33} \leqslant 50$

$0 \leqslant x_{34} \leqslant 60$

Solução: $z = 20$

$x_{11} = 0$; $x_{12} = 1$; $x_{13} = 6$; $x_{14} = 4$

$x_{21} = 0$; $x_{22} = 1{,}5$; $x_{23} = 8$; $x_{24} = 5{,}5$

$x_{31} = 50$; $x_{32} = 20$; $x_{33} = 0$; $x_{34} = 40$

7) $\min z = 200x_{11} + 300x_{12} + 200x_{21} + 300x_{22}$

s.a.

$$
\left\{
\begin{array}{l}
2x_{11} + 3x_{21} + x_{31} = 3 \\
2x_{12} + 3x_{22} + x_{32} = 6 \\
300x_{11} + 600x_{12} + x_{13} = 580 \\
15x_{11} + 10x_{12} + x_{14} = 30 \\
x_{11} + x_{12} + x_{15} = 2 \\
300x_{21} + 600x_{22} + x_{23} = 530 \\
15x_{21} + 10x_{22} + x_{24} = 80 \\
x_{21} + x_{22} + x_{25} = 2 \\
x_{31} + x_{33} = 3 \\
x_{32} + x_{34} = 6 \\
0,1 \leq x_{11} \leq 0,6 \\
x_{12} \geq 0 \\
100 \leq x_{13} \leq 480 \\
5 \leq x_{14} \leq 25 \\
1 \leq x_{15} \leq 1 \\
0,1 \leq x_{21} \leq 1 \\
0,1 \leq x_{22} \leq 0,8 \\
50 \leq x_{23} \leq 480 \\
10 \leq x_{24} \leq 70 \\
1 \leq x_{25} \leq 1 \\
x_{31} , x_{32} \geq 0 \\
0 \leq x_{33} \leq 3 \\
1 \leq x_{34} \leq 5
\end{array}
\right.
$$

__Solução:__ $z = 480$

$x_{11} = 0,6$; $x_{12} = 0,4$; $x_{13} = 160$; $x_{14} = 17$; $x_{15} = 1$
$x_{21} = 0,6$; $x_{22} = 0,4$; $x_{23} = 110$; $x_{24} = 67$; $x_{25} = 1$
$x_{31} = 0$; $x_{32} = 4$; $x_{33} = 3$; $x_{34} = 2$

Os programas RAROSN.F4, RADW1.F4 e RADW2.F4 também foram testados com problemas pequenos, isto é, com rações fictícias que criamos para testes preliminares. Posteriormente, conseguimos um exemplo real com duas rações, o qual descreveremos a seguir.

4.1.2 - <u>Um exemplo real com 2 rações</u>:

São dados 21 ingredientes (n=21) e 20 nutrientes (m=20) com os quais desejamos produzir 2 rações (K=2) com custo mínimo, sendo que a quantidade a ser produzida da ração 1 é de 10 toneladas (q_1 = 10 Ton) e da ração 2 é de 20 toneladas (q_2 = 20 Ton) . Na tabela (4.1) listamos, para cada ingrediente, seu código, nome, preço por Kg (c_j) e as quantidades mínima e máxima (γ_{kj} e δ_{kj}) de alguns ingredientes nas rações 1 e 2. Note que alguns ingredientes não possuem limitantes inferiores nem superiores; outros possuem apenas os limitantes inferiores (neste caso, a variável x_{kj} correspondente será canalizada com limite superior igual a 1) ; outros, ainda, possuem apenas os limitantes superiores (neste caso, a variável x_{kj} correspondente será canalizada com limite inferior igual a zero). Além disso, quando um ingrediente não está disponível no mercado, ele terá o preço de CR$ 9999,00.

TABELA 4.1:

INGREDIENTES			RAÇÃO 1		RAÇÃO 2	
CÓDIGO	NOME	PREÇO/ KG.	MIN.	MÁX.	MIN.	MÁX.
021	P - BOVINO	538,40				
100	ARAXA	9999,00		0,01		
107	MILHO	11,70	0,15		0,20	
121	SORGO	9999,00				0,15
274	VITACENO	9999,00		0,10		0,10
348	F. ARROZ GORDO	9999,00				
350	CASCA ARROZ	3,40		0,10		0,10
353	F. TRIGO	10,00		0,50		0,25
409	MAMONA DETOX	1000,00		0,10		0,10
440	F. ALGODÃO 39	15,20	0,05	0,20	0,075	0,30
451	F. AMENDOIM	16,00		0,25		0,25
556	SOJA 46	18,50				0,90
577	OSSO CALC.	9999,00				
689	FOSFATO 18%	9999,00				
703	CALCÁREO	2,20				
707	FOSFATO BI QUINOP	33,00				
713	SAL	7,00				
907	LECITINA	1000,00		0,03		0,02
913	AC. GRAXO	26,50		0,03		0,03
939	MELAÇO	7,00	0,05	0,05	0,05	0,05
952	URÉIA 45%	34,50		0,003		0,01

Na tabela (4.2) estão os ingredientes que possuem quantidades limitadas inferior e/ou superiormente, quando repartidos entre as 2 rações (componentes dos vetores g e ℓ nas restrições de acoplamento).

TABELA 4.2:

INGREDIENTES COMPARTILHADOS			
CÓDIGO	NOME	QUANTIDADE MÍNIMA A SER USADA NAS 2 RAÇÕES	QUANTIDADE MÁXIMA A SER USADA NAS 2 RAÇÕES
107	MILHO	0	15
353	F. TRIGO	8	21,5
440	F. ALGODÃO 39	6	8
703	CALCÁREO	0	60

Na tabela (4.3) listamos, para cada nutriente considerado, seu código, nome, a unidade em que aparece e as quantidades mínima e máxima (α_{ki} e β_{ki}) destes nutrientes nas rações 1 e 2. Também neste caso, notamos que nem todos os nutrientes possuem limitações superiores, isto é, nem todas as restrições dos blocos (rações) são canalizadas, embora os programas considerem canalizações em todas as restrições (quando estas não existem explicitamente, consideramos o limite inferior zero e o limite superior infinito). Podemos ver também que, embora tivéssemos definido os nutrientes em frações, eles podem aparecer em outras unidades.

Na tabela (4.4), apresentamos a matriz nutrientes x ingredientes (matriz A). Podemos notar, tanto na tabela (4.3) quanto na tabela (4.4), que o nutriente de código 01, chamado de "peso", é o nutriente i' que está representando a restrição adicional $\sum_{j=1}^{21} x_{kj} = 1$, para k = 1 e k = 2, descrita no capítulo 1.

TABELA 4.3:

NUTRIENTES			RAÇÃO 1		RAÇÃO 2	
CÓDIGO	NOME	UNIDADE	MIN.	MÁX.	MIN.	MÁX.
01	PESO	PCT	1,00	1,00	1,00	1,00
03	GORDURA	PCT	2,50		3,50	
04	FIBRA	PCT		12,00		10,00
05	UMIDADE	PCT				
07	CINZAS	PCT				
10	PROTEÍNA	PCT	12,00		14,00	
16	P - TOTAL	PCT	0,75		0,60	
17	CÁLCIO	PCT	1,00	1,25	1,50	1,75
20	POTÁSSIO	PCT				
21	SAL	PCT	1,00	1,05	2,00	2,00
32	GLICINA	PCT				
34	LEUCINA	PCT				
36	METIONINA	PCT				
45	LEUCINA DISP.	PCT				
47	METIONINA DISP.	PCT				
50	TRIPTOFANO DISP.	PCT				
56	AC. PANTOTÊNICO	mg/kg				
57	RIBOFLAVINA	mg/kg				
59	DENSIDADE	g/cc				
60	PROT. SOLÚVEL	PCT				

TABELA 4.4:

ingredientes / nutrientes	021	100	107	121	274	348	350	353	409	440	451	556	577	689	703	707	713	907	913	939	952
01	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
03			3,6	3	1	15	0,8	3,7	1	0,6	1,5	0,9						100	97	0,1	
04			2,7	2,5	42	10	40	10	30	14	9	6	0,4								
05			13	14	7	9,4	8	12		9	8	12,5	4,8				2		2	25	
07		99	1,5	2	7,1	9	14	5	30	6	7	6	98	83,5	57	83,5	100		3	8,1	
10			8,8	9	5	14	3	16	38	39,5	49	46	1							3,2	284
16		15,5	0,2	0,3	0,14	1,5	0,08	1	0,6	1	0,57	0,58	16,5	18,5		18,5		1,4		0,08	
17		22	0,1	0,03	0,4	0,12	0,08	0,25	0,4	0,25	0,3	0,3	35	22	38	21,5			0,2	0,7	
20		0,02	0,28	0,35		1,41	0,08	1,09	0,75	1,5	1,1	2,05	1,66	0,043	0,001	0,05	0,03	0,005	0,05	1,45	0,17
21			0,025	0,025	0,5	0,12		0,1	0,075	0,12	0,2	0,1	0,1	0,4		1,5	97			1,25	
32			0,33	0,35		0,63		0,43	1,8	1,78	2,38	2,369								0,052	
34			1,05	1,4		0,6		1	2,74	2,31	2,51	3,588								0,042	
36			0,2	0,1	0,08	0,2	0,2	0,24	0,57	0,56	0,44	0,661									
46			0,86	1		0,4		0,82		2,1	2,7	3,288								0,037	
47			0,19	0,12		0,12		0,18		0,51	0,45	0,676									
50			0,055	0,06		0,06		0,14		0,45	0,45	0,622									
56			7	9		22	22	20,46		11	53,02	14								39,6	
57			1,1	0,88		2,64	0,044	1,76		5,28	5,28	3,08								2,2	
59	0,81	1,65	0,63	0,68	0,25	0,31	0,35	0,25	0,62	0,58	0,63	0,6	1,1	0,96	1,48	0,96	0,99	0,97	1	1,4	0,75
60			11,9							10	57,4	17									

Observação: os elementos não preenchidos correspondem a elementos nulos.

A solução ótima deste problema é a seguinte:

valor da função objetivo (custo de produção das 2 rações) : 318 208,00

NOME DO INGREDIENTE J, j=1,...,21	FRAÇÃO DO INGREDIENTE j NA RAÇÃO 1 (x_{1j},j=1,...,21)	FRAÇÃO DO INGREDIENTE j NA RAÇÃO 2 (x_{2j},j=1,...,21)
P - BOVINO	0	0
ARAXA	0	0
MILHO	0,15	0,3175452
SORGO	0	0
VITACENO	0	0
F.ARROZ GORDO	0	0
CASCA ARROZ	0,10	0,0895138
F. TRIGO	0,50	0,25
MAMONA DETOX	0	0
F. ALGODÃO 39	0,1625365	0,2187197
F. AMENDOIM	0	0
SOJA 46	0	0
OSSO CALC.	0	0
FOSFATO 18%	0	0
CALCÁREO	0,0256197	0,0392259
FOSFATO BI QUINOP	0,0024565	0,0030602
SAL	0,0093872	0,0193167
LECITINA	0	0
AC. GRAXO	0	0,0126183
MELAÇO	0,05	0,05
URÉIA 45%	0	0

4.2 - RESULTADOS OBTIDOS:

Os programas RROSEN.F4 e DANWO.F4 foram comparados entre si e com o programa SIRECA, que implementa o Método Simplex Revisado com variáveis canalizadas. SIRECA é, basicamente, a subrotina SUBPR.F4, mas inclui a entrada de dados por leitura e a impressão dos resultados. Neste caso, o programa despreza a estrutura bloco-angular dos problemas, considerando todos os elementos que são nulos. Os resultados obtidos através destes programas, para os problemas descritos na seção (4.1.1) e dois outros problemas, estão descritos na tabela (4.5).

TABELA 4.5:

PROBLEMA	DANWO.F4		RROSEN.F4 $\Gamma=(0...0)$		RROSEN.F4 $\Gamma=(1...1)$		SIRECA	
	nº de iterações	tempo de execução	nº de iterações	tempo de execução	nº de iterações	tempo de execução	nº de iterações	tempo de execução
1	7	0,23	3	0,10	3	0,09	14	0,09
2	8	0,15	2	0,03	2	0,03	5	0,02
3	8	0,20	4	0,12	4	0,11	14	0,08
4	8	0,24	4	0,15	5	0,20	16	0,12
5	7	0,16	2	0,06	2	0,06	12	0,05
6	11	0,36	2	0,08	3	0,15	21	0,17
7	8	0,31	2	0,12	2	0,12	24	0,28
8	7	0,15	4	0,09	4	0,09	10	0,04
9	8	0,18	3	0,08	4	0,11	7	0,03

Observações:

1ª) Os problemas 8 e 9 da tabela (4.5) foram obtidos de modificações feitas nos problemas 3 e 4, respectivamente. Por esta razão, eles não foram colocados na seção (4.1.1).

2ª) Nos problemas que implementam o Método de Rosen, precisamos fornecer o vetor das variáveis duais Γ, descrito no capítulo 3. Na tabela (4.5), os testes foram realizados considerando $\Gamma = (0...0)$ e $\Gamma = (1...1)$. Executamos os mesmos problemas testes para outros valores de Γ, mas os resultados obtidos (número de iterações e tempo de execução) não se modificam muito.

3ª) Não colocamos os problemas pequenos, criados para testar os programas RAROSN.F4, RADW1.F4 e RADW2.F4, nem a tabela comparativa de seus resultados, por julgarmos que estas " rações fictícias" serviram apenas como testes preliminares.

4ª) Os programas RROSEN.F4 e DANWO.F4 requerem, na entrada de dados, o valor da variável "TOL", que é usada como uma tolerância no critério de parada de cada subproblema e do problema reduzido (ou do problema mestre). Em todos os problemas da tabela (4.5), o valor da variável "TOL" foi de 10^{-5}. O mesmo valor desta tolerância (10^{-5}) foi usado pelo programa SIRECA.

5ª) O programa RAROSN.F4 requer, na entrada de dados, o valor da variável "TOL" descrita acima. Já os programas RADW1.F4 e RADW2.F4 requerem duas tolerâncias: "TOL", que é a tolerân -

cia usada no critério de parada dos subproblemas e "TOLM", que é a tolerância usada no critério de parada do problema mestre. Quando executamos estes 3 programas com problemas pequenos, variamos estas tolerâncias. O programa RAROSN.F4 sempre se comportou bem com todas as tolerâncias usadas, ao passo que os programas RADW1.F4 e RADW2.F4 conseguiram soluções não ótimas, em alguns problemas, para certos valores de "TOL" e de "TOLM" (nestes casos, o número de iterações também foi alterado).

6ª) O tempo de execução mostrado na tabela anterior é dado em segundos.

Executamos os programas RAROSN.F4, RADW1.F4 e RADW2.F4 com o problema real das 2 rações, descrito na seção (4.1.2), obtendo como resultados:

RAROSN.F4		RADW1.F4	RADW2.F4
nº de iterações	tempo de execução (segundos)	não convergiu	convergiu para soluções não ótimas em 3 casos
5	93.98		

RAROSN.F4 foi executado usando $\Gamma = (0,\ldots 0)$ e TOL=10^{-5}.

RADW1.F4 foi executado com vários valores para as variáveis "TOL" e "TOLM" e, para alguns casos, o tempo limite de CPU foi até 3 horas. Em nenhum dos casos este programa chegou à solução e, em todos os casos, o número de iterações efetuadas até o final da fase I do problema mestre foi bastante alto - entre 344 e 698 iterações.

RADW2.F4 também foi executado com vários valores para as variáveis "TOL" e "TOLM", sendo que, na grande maioria dos casos, o programa não chegou à solução (a execução era interrompida quando o tempo limite de CPU era atingido, estando este tempo entre 2 e 3 horas). Para 3 valores de "TOL" e de "TOLM", este programa conseguiu chegar à uma solução, embora em nenhum dos casos a solução encontrada fosse a ótima - além do valor da função objetivo ser maior, algumas variáveis u_{kj}, que são introduzidas para se eliminar as restrições canalizadas, como mostrado no capítulo 1, excedem seus limites. Estes resultados estão descritos abaixo:

caso	TOL	TOLM	número de iterações	tempo de execução (segundos)	valor da função objetivo
1	10^{-2}	10^{-2}	220	902,57	335070,1
2	10^{-2}	10^{-4}	195	863,39	361920,4
3	10^{-2}	10^{-5}	195	847,46	361920,4

Além disso, os casos 2 e 3 apresentaram a mesma solução.

4.3 - CONCLUSÕES:

Na tabela (4.5) estão os resultados dos problemas descritos na seção (4.1.1), obtidos pelos programas RROSEN.F4,DANWO.F4 e SIRECA. Comparando estes resultados, vemos que RROSEN.F4 se mostrou sempre mais eficiente que DANWO.F4 - o número de iterações e o tempo de execução foram menores em todos os problemas. Quando comparamos os resultados entre os programas RROSEN.F4 e SIRECA, vemos que o número de iterações apresentado por SIRECA, em todos os problemas, foi maior, porém o mesmo não ocorreu quanto ao tempo de execução. Isto se deve ao fato de que uma iteração executada por SIRECA é bem menos trabalhosa do que uma iteração executada por RROSEN.F4 ou pelos outros programas, pois, enquanto SIRECA faz apenas um pivotamento em cada iteração, os outros programas precisam resolver K subproblemas e ainda o problema reduzido, no caso dos programas que implementam o Método de Rosen, ou o problema mestre (ou um pivotamento no problema mestre), no caso dos programas que implementam o Método de Dantzig-Wolfe.

Embora não tenhamos colocado aqui a tabela referente aos problemas pequenos quando executamos os programas RAROSN.F4, RADW1.F4 e RADW2.F4, fizemos também uma comparação entre eles. Como no caso anterior, o número de iterações e o tempo de execução obtidos por RAROSN.F4 foram menores que nos outros dois programas (não comparamos os resultados destes programas com o programa SIRECA). Fizemos,também, uma comparação entre os programas RADW1.F4 e RADW2.F4 separadamente: em todos os problemas pequenos, o número de iterações obtido por RADW1.F4 foi maior que em

RADW2.F4, ao passo que o tempo de execução foi bem semelhante . Este fato é bem razoável, se considerarmos que em cada iteração de RADW2.F4 é feita a montagem e a resolução do problema mestre restrito, ao passo que RADW1.F4 faz apenas um pivotamento no problema mestre, por iteração. Além disso, pudemos ver também que, no problema real das 2 rações, RADW1.F4 não convergiu em nenhuma tentativa, enquanto que RADW2.F4 conseguiu encontrar uma solução (embora não ótima) em três casos, como mostramos na seção anterior. O que podemos concluir é que, em todos os testes realizados, tanto para problemas gerais pequenos, quanto nas " rações fictícias", quanto no problema real das 2 rações, os programas que implementam o Método de Rosen deram melhores resultados.

Encontramos algumas dificuldades em alguns testes nos programas RADW1.F4 e RADW2.F4, talvez devido ao fato de que as matrizes nutrientes x ingredientes fossem mal condicionadas, fato este que, ao que parece, ocorre frequentemente quando temos que resolver um problema real de otimização de rações. Além disso, geralmente, a matriz A possui elementos com ordem de grandeza bastante diferentes entre si. Mesmo quando resolvemos problemas pequenos, obtivemos, em alguns testes com os programas RADW1.F4 e RADW2.F4, soluções não ótimas. Por isso, resolvemos trabalhar com duas tolerâncias distintas nestes programas (TOL e TOLM). Pudemos observar que, para alguns valores de "TOL" e "TOLM", os programas RADW1.F4 e RADW2.F4 encontravam a solução ótima dos problemas, e para outros valores, ou a solução encontrada não era ótima, ou estes programas não conseguiam encontrar nenhuma solução (neste último caso, a execução era interrompida quando o tempo limite de CPU, já estabelecido, era alcançado). Lembramos aqui que os programas que implementam o Método de Rosen sempre chegaram às soluções ótimas dos problemas, e com um pequeno número de iterações.

Neste sentido, poderiam ser feitas algumas melhorias nos programas, objetivando a diminuição do acúmulo de erros de arredondamento (que crescem à medida que o número de iterações aumenta):

- uso da precisão dupla nos programas em vez da precisão simples, como fizemos;
- escalonamento das matrizes de tecnologia antes da aplicação de qualquer método;
- reinversão das matrizes básicas de cada bloco a cada "NR" iterações, onde "NR" é um número estabelecido "a priori".

Pudemos também comparar os resultados obtidos com RAROSN.F4, no problema real das 2 rações, com os resultados obtidos pelo programa MINOS (Modular Incore Nonlinear Optimization System). A solução encontrada pelos dois programas foi idêntica; o número de iterações obtido através do MINOS foi de 56 e o tempo de execução foi bastante semelhante (infelizmente, não foi anotado o tempo de execução obtido através do MINOS).

Analisando todos estes resultados, acreditamos que o Método de Rosen é um bom método para se resolver problemas de Programação Linear com estrutura bloco-angular. Todas as dificuldades que constatamos na obtenção dos resultados de problemas, foram encontradas no Método de Dantzig-Wolfe. Convém lembrarmos que, embora o acréscimo da restrição (3.10), descrita no capítulo 3 , não tenha sido implementada nos programas que efetuam o Método de Rosen, em nenhum dos vários testes realizados ocorreu o caso do problema reduzido ser ilimitado quando o problema original possuía uma solução ótima finita. Também, como já dissemos no capítulo 3, o problema reduzido não foi resolvido pelo Método Dual Simplex e, em toda iteração, foi feita a fase I do Método Simplex no problema reduzido.

No apêndice C, apresentamos a listagem e uma rápida documentação do programa RAROSN.F4.

APÊNDICE A

O MÉTODO SIMPLEX REVISADO COM VARIÁVEIS CANALIZADAS

Seja o problema:

$$\left\{\begin{array}{l} \min\ c\,x \\ \text{s.a.} \\ \quad Ax = b \\ \quad 0 \leqslant x \leqslant h \end{array}\right\} \tag{A.1}$$

onde:

c : vetor linha de dimensão 1 x n com componentes c_j ;

x : vetor de dimensão n x 1 com componentes x_j ;

A : matriz de dimensão m x n com componentes a_{ij}. Chamaremos de A^j à j - ésima coluna de A;

b : vetor de dimensão m x 1 com componentes b_i ;

h : vetor de dimensão n x 1 com componentes h_j .

<u>Definição</u>: <u>solução básica estendida factível</u> correspondente ao problema (A.1): é uma solução factível tal que:

i) m variáveis correspondem a m colunas linearmente independentes de A — variáveis básicas.

ii) n-m variáveis estão ou no limite inferior (zero) ou no limite superior (h_j) — variáveis não básicas.

Definimos ainda:

B : conjunto de índices que indicam as variáveis básicas;

N : conjunto de índices que indicam as variáveis não básicas.

Com estes dois conjuntos, particionamos x, c e A da seguinte maneira:

$$x = (x^B \ / \ x^N)^T$$

$$c = (c^B \ / \ c^N)$$

$$A = (A^B \ / \ A^N)$$

e chamaremos de B^{-1} à inversa de A^B.

Os coeficientes de custos relativos são dados por:

$$\hat{c}_j = c_j - (c^B \ B^{-1}) \ A^j \ , \ j = 1,\ldots,n.$$

Inicialmente, supomos conhecida uma solução básica estendida factível para o problema (A.1). Examinamos então as variáveis não básicas como possíveis candidatas a melhorar a solução. Uma variável x_j em seu limite <u>inferior</u> somente poderá crescer, e este acréscimo irá melhorar o valor da função objetivo se o seu custo relativo $\hat{c}_j$ for negativo. Da mesma maneira, uma variável x_j em seu limite <u>superior</u> somente poderá decrescer, e este decréscimo irá melhorar o valor da função objetivo se o seu custo relativo $\hat{c}_j$ for positivo. Suponhamos que a variável x_j foi escolhida para ser alterada, de modo que a solução continue sendo factível. O valor de x_j poderá ser alterado até que:

i) uma variável básica atinja um de seus limites,

ii) x_j atinja seu limite oposto.

Se a condição (i) ocorrer primeiro, então aquela variá vel básica passará a ser não básica e a variável não básica x_j passará a ser básica.

Se a condição (ii) ocorrer primeiro, então x_j ficará no seu limite oposto, não havendo alteração na base.

Este processo de alteração de variáveis é repetido até que a condição de otimalidade, expressa no teorema abaixo, seja satisfeita.

Teorema da otimalidade:

Uma solução básica estendida factível é ótima para o problema (A.1) se, para as variáveis não básicas x_j, temos:

$$\hat{c}_j \geqslant 0, \text{ se } x_j = 0$$

$$\bar{c}_j \leqslant 0, \text{ se } x_j = h_j.$$

Para escrevermos o algorítmo, faremos a seguinte mudança de variáveis:

$$\left[\begin{array}{l} x_j^+ = x_j \text{ , se a variável } x_j \text{ está no limite inferior;} \\ x_j^- = h_j - x_j\text{, se a variável } x_j \text{ está no limite superior.} \end{array}\right] \quad (A.2)$$

Para indicarmos em que limite está uma variável, usaremos o vetor de sinal σ [Luenberger, 1973, pp. 48-53] com componentes σ_j, da seguinte maneira:

se $\sigma_j = +$, a variável x_j está no limite inferior (x_j^+) ;

se $\sigma_j = -$, a variável x_j está no limite superior (x_j^-) .

Desta maneira, qualquer variável não básica terá sempre o valor zero, pois, se x_j atingir seu limite superior, isto é,

$x_j = h_j$, então $x_j^- = h_j - x_j = 0$. Com esta mudança de variáveis, procuraremos sempre crescer o valor de uma variável não básica , isto é, procuraremos coeficientes de custos relativos $\hat{c}_j$ negativos, ficando assim o teste de otimalidade idêntico ao do Método Simplex usual, ou seja, quando todos os $\hat{c}_j$ forem não negativos , estaremos na solução ótima do problema.

A.1 - <u>ALGORÍTMO</u>:

(0) Encontrar uma solução básica estendida factível inicial para o problema (se necessário, usar a fase I do Método Simplex) , obtendo os índices das variáveis básicas (B), os índices das variáveis não básicas (N), B^{-1} e $\hat{b} = B^{-1}$ b.

(1) Calcular $\pi = c^B B^{-1}$.

(2) Calcular $\hat{c}_j = c_j - \pi A^j$, para $j \in N$.

(3) Se $\hat{c}_j \geq 0$, $\forall j \in N$, então a solução ótima é a solução básica estendida factível associada à B. Retornar as variáveis x_j^+ e x_j^- para x_j, conforme (A.2) e parar.

Senão, escolher uma variável x_j tal que $\hat{c}_j < 0$.

(4) Obter $\hat{A}^j = B^{-1} A^j$ e determinar:

(4.1) h_j ;

(4.2) $\min\limits_i \left\{ \dfrac{\hat{b}_i}{\hat{a}_{ij}} \;/\; \hat{a}_{ij} > 0 \right\}$, onde $\hat{a}_{ij}$ é a i - ésima componente da coluna $\hat{A}^j$ e $\hat{b}_i$ é a i-ésima componente de $\hat{b}$;

$$(4.3) \quad \min_i \left\{ \frac{\hat{b}_i - h_{j_i}}{\hat{a}_{ij}} \;/\; \hat{a}_{ij} < 0 \right\}$$, onde j_i é a i-ésima variável básica.

Encontrar o índice r que dá o mínimo entre (4.1),(4.2) e (4.3).

(5) Dependendo da origem do índice r encontrado no passo (4), temos um procedimento:

(5.1) se o índice r foi dado por (4.1), então a variável x_j atingiu seu limite oposto (mudança de variável). Não há mudança de base, isto é, não há pivotamento.

- atualizar $\hat{b}$: $\hat{b} \Longleftarrow \hat{b} - h_j \hat{A}^j$;
- trocar o sinal de A^j , c_j e σ_j ;
- voltar ao passo (2);

(5.2) se o índice r foi dado por (4.2), então:

- pivotear em $\hat{a}_{rj}$, atualizando B^{-1} e $\hat{b}$;
- atualizar os índices de B e de N ;
- voltar ao passo (1);

(5.3) se o índice r foi dado por (4.3), então a r - ésima variável básica x_{j_r} atingiu seu limite oposto (mudança de variável).

- atualizar a r - ésima componente de $\hat{b}$: $\hat{b}_r \Longleftarrow \hat{b}_r - h_{j_r}$;
- trocar o sinal de A^{j_r}, c_{j_r} e σ_{j_r} ;

- pivotear em $\hat{a}_{rj}$, atualizando B^{-1} e $\hat{b}$;
- atualizar os índices de B e de N;
- voltar ao passo (1).

APÊNDICE B

ALGUNS DETALHES DO MÉTODO DE ROSEN

Descreveremos aqui, com mais detalhes, algumas etapas do Método de Rosen quando consideramos a partição feita nas variáveis não básicas. Como a subrotina que resolve os subproblemas usa, basicamente, o algorítmo descrito no apêndice A, a mesma transformação feita nas variáveis foi estendida aos programas que implementam o Método de Rosen.

B.1 - A NOVA PARTIÇÃO:

Uma vez que os subproblemas possuem variáveis canalizadas, num bloco k qualquer, uma componente da solução x_k que seja não básica pode ser igual ao seu limite inferior (zero) ou igual ao seu limite superior. Desta maneira, cada bloco k será composto de três partes:

i) uma parte básica;

ii) uma parte não básica correspondente às variáveis que estão no limite inferior;

iii) uma parte não básica correspondente às variáveis que estão no limite superior.

Suporemos que as colunas básicas de cada bloco k são as primeiras m_k colunas e, dentre as $n_k - m_k$ colunas não básicas, as que correspondem às variáveis que estão no limite inferior estão colocadas em primeiro lugar. Teremos então, na solução ótima de cada bloco k, para $k = 1,\dots,K$:

$x_k = (x_k^B \,/\, x_k^N)^T$ e $x_k^N = (x_k^I \,/\, x_k^S)^T$, onde:

x_k^I : contém as componentes não básicas de x_k que estão no limite inferior;

x_k^S : contém as componentes não básicas de x_k que estão no limite superior.

$A_k = (A_k^B \,/\, A_k^N)$ e $A_k^N = (A_k^I \,/\, A_k^S)$, onde:

A_k^I : contém as colunas não básicas de A_k correspondentes às variáveis que estão no limite inferior;

A_k^S : contém as colunas não básicas de A_k correspondentes às variáveis que estão no limite superior.

$h_k = (h_k^B \,/\, h_k^N)^T$ e $h_k^N = (h_k^I \,/\, h_k^S)^T$, onde:

h_k^I : parte de h_k^N correspondente às variáveis não básicas que estão no limite inferior;

h_k^S : parte de h_k^N correspondente às variáveis não básicas que estão no limite superior.

Estendendo esta partição à função objetivo e às restrições de acoplamento do problema (1.8), temos:

$c_k = (c_k^B \,/\, c_k^N)$ e $c_k^N = (c_k^I \,/\, c_k^S)$, onde:

c_k^I : contém as componentes não básicas de c_k correspondentes às variáveis que estão no limite inferior;

c_k^S : contêm as componentes não básicas de c_k correspondentes às variáveis que estão no limite superior.

$Q_k = (Q_k^B \;/\; Q_k^N)$ e $Q_k^N = (Q_k^I \;/\; Q_k^S)$, onde:

Q_k^I : contêm as colunas não básicas de Q_k correspondentes às variáveis que estão no limite inferior;

Q_k^S : contêm as colunas não básicas de Q_k correspondentes às variáveis que estão no limite superior.

B.2 - A MUDANÇA DE VARIÁVEIS:

Com o objetivo de deixar todas as variáveis não básicas no limite inferior (zero), faremos a mudança de variáveis descrita no apêndice A. Assim, dado um subproblema k fixo e, esquecendo o subscrito k que se refere ao subproblema, fazemos:

$$\left\{\begin{array}{l} x_j^+ = x_j \text{ , se a variável } x_j \text{ está no limite inferior;} \\ x_j^- = h_j - x_j \text{ , se a variável } x_j \text{ está no limite superior;} \end{array}\right\} \quad (B.1)$$

onde, para um bloco k fixo, temos:

x_j : componente j do vetor x_k ;

h_j : componente j do vetor h_k .

Expressando agora as variáveis básicas de cada bloco k, para k = 1,...,K, em função das variáveis não básicas, temos:

$$x_k^B = B_k^{-1} b_k - B_k^{-1} A_k^N x_k^N \quad , \quad k = 1,\ldots,K.$$

Incluindo as partições feitas em A_k^N e em x_k^N , temos:

$$x_k^B = B_k^{-1} b_k - B_k^{-1} (A_k^I / A_k^S) \left(\frac{x_k^I}{x_k^S} \right) , \; k = 1,\ldots,K. \tag{B.2}$$

De (B.1) temos que:

$$\begin{cases} x_k^I = x_k^{I+} \\ x_k^S = h_k^S - x_k^{S-} \end{cases}$$

Substituindo esta última expressão em (B.2), obtemos :

$$x_k^B = B_k^{-1} b_k - B_k^{-1} A_k^I x_k^{I+} + B_k^{-1} A_k^S x_k^{S-} - B_k^{-1} A_k^S h_k^S \; ,k=1,\ldots,K \tag{B.3}$$

onde as componentes de x_k^{I+} e de x_k^{S-} são nulas, para $\forall k$.

B.3 - O NOVO PROBLEMA REDUZIDO:

Sabemos que o problema reduzido será agora formado quando substituímos a equação (B.3) na função objetivo e nas restrições de acoplamento do problema (1.8).

B.3.1 - Formação das restrições:

As restrições de acoplamento do problema (1.8) são:

$$\sum_{k=1}^{K} Q_k x_k = b_o \; .$$

Incluindo as partições feitas em Q_k e em x_k, temos:

$$\sum_{k=1}^{K} \left(Q_k^B \; / \; Q_k^I \; / \; Q_k^S \right) \begin{pmatrix} x_k^B \\ \hline x_k^I \\ \hline x_k^S \end{pmatrix} = b_o \; .$$

Substituindo agora a equação (B.3) e a mudança de variáveis descrita em (B.1) na equação acima, e desenvolvendo a expressão, obtemos:

$$\sum_{k=1}^{K} \Big[Q_k^B \left(B_k^{-1} b_k - B_k^{-1} A_k^I x_k^{I+} + B_k^{-1} A_k^S x_k^{S-} - B_k^{-1} A_k^S h_k^S\right) + Q_k^I x_k^{I+} +$$

$$+ Q_k^S \left(h_k^S - x_k^{S-}\right) \Big] = b_o$$

$$\sum_{k=1}^{K} \Big[Q_k^B B_k^{-1} b_k - Q_k^B B_k^{-1} A_k^I x_k^{I+} + Q_k^B B_k^{-1} A_k^S x_k^{S-} - Q_k^B B_k^{-1} A_k^S h_k^S +$$

$$+ Q_k^I x_k^{I+} + Q_k^S h_k^S - Q_k^S x_k^{S-} \Big] = b_o$$

$$\sum_{k=1}^{K} \left[Q_k^I - Q_k^B B_k^{-1} A_k^I \right] x_k^{I+} - \sum_{k=1}^{K} \left[Q_k^S - Q_k^B B_k^{-1} A_k^S \right] x_k^{S-} =$$

$$= b_o - \sum_{k=1}^{K} Q_k^S h_k^S - \sum_{k=1}^{K} \left[Q_k^B \left(B_k^{-1} b_k - B_k^{-1} A_k^S h_k^S\right)\right] \quad (B.4)$$

Notemos que o termo $(B_k^{-1} b_k - B_k^{-1} A_k^S h_k^S)$ da igualdade acima é a própria parte básica da solução ótima obtida pelo subproblema k — basta compararmos este termo com a equação (B.3) e lembrarmos que nela, $x_k^{I+} = x_k^{S-} = 0$.

Sendo $\overset{o}{x}{}_k^B = B_k^{-1} b_k - B_k^{-1} A_k^S h_k^S$, $k = 1,\dots,K$, e substituindo $\overset{o}{x}{}_k^B$ em (B.4), temos:

$$\sum_{k=1}^{K} \left[(Q_k^I / -Q_k^S) - Q_k^B\, B_k^{-1}\, (A_k^I / A_k^S)\right] \left(\frac{x_k^{I+}}{x_k^{S-}}\right) = b_o - \sum_{k=1}^{K} Q_k^S\, h_k^S - \sum_{k=1}^{K} Q_k^B\, \overset{o}{x}{}_k^B \,,$$

que são as restrições do problema reduzido.

Convém notar que, para cada bloco k, as colunas relativas às variáveis não básicas no limite superior, aparecem multiplicadas por -1, tanto nas matrizes Q_k quanto nas matrizes A_k, devido à transformação de variáveis definidas em (B.1).

B.3.2 - Formação da função objetivo:

A função objetivo do problema (1.8) é:

$$\min z = \sum_{k=1}^{K} c_k\, x_k \,.$$

Incluindo as partições feitas em c_k e em x_k, temos :

$$\min z = \sum_{k=1}^{K} (c_k^B \,/\, c_k^I \,/\, c_k^S) \begin{pmatrix} x_k^B \\ \hline x_k^I \\ \hline x_k^S \end{pmatrix} .$$

Substituindo agora a equação (B.3) e a mudança de variáveis descrita em (B.1) na equação acima e desenvolvendo a expressão, obtemos:

$$\min z = \sum_{k=1}^{K} \left[c_k^B\, (B_k^{-1}\, b_k - B_k^{-1}\, A_k^I\, x_k^{I+} + B_k^{-1}\, A_k^S\, x_k^{S-} - B_k^{-1}\, A_k^S\, h_k^S) \right. +$$

$$\left. + c_k^I\, x_k^{I+} + c_k^S\, (h_k^S - x_k^{S-}) \right] =$$

$$= \sum_{k=1}^{K} \left[c_k^B\, B_k^{-1}\, b_k - c_k^B\, B_k^{-1}\, A_k^I\, x_k^{I+} + c_k^B\, B_k^{-1}\, A_k^S\, x_k^{S-} - c_k^B B_k^{-1} A_k^S h_k^S + \right.$$

$$+ c_k^I x_k^{I+} + c_k^S h_k^S - c_k^S x_k^{S-}] =$$

$$= \sum_{k=1}^{K} [c_k^I - c_k^B B_k^{-1} A_k^I] x_k^{I+} - \sum_{k=1}^{K} [c_k^S - c_k^B B_k^{-1} A_k^S] x_k^{S-} +$$

$$+ \sum_{k=1}^{K} c_k^S h_k^S + \sum_{k=1}^{K} c_k^B (B_k^{-1} b_k - B_k^{-1} A_k^S h_k^S)$$

e como já foi visto que $B_k^{-1} b_k - B_k^{-1} A_k^S h_k^S = \overset{o}{x}{}_k^B$, temos que :

$$\min z - \sum_{k=1}^{K} c_k^B \overset{o}{x}{}_k^B - \sum_{k=1}^{K} c_k^S h_k^S = \sum_{k=1}^{K} (c_k^I - c_k^B B_k^{-1} A_k^I) x_k^{I+}$$

$$- \sum_{k=1}^{K} (c_k^S - c_k^B B_k^{-1} A_k^S) x_k^{S-}$$

Os termos $\sum_{k=1}^{K} c_k^B \overset{o}{x}{}_k^B$ e $\sum_{k=1}^{K} c_k^S h_k^S$ são constantes e, logo, podem ser dispensados da minimização. Assim, a função objetivo do problema reduzido pode ser vista como:

$$\min \sum_{k=1}^{K} \left[(c_k^I \,/\, -c_k^S) - c_k^B B_k^{-1} (A_k^I \,/\, -A_k^S)\right] \begin{pmatrix} x_k^{I+} \\ \hline x_k^{S-} \end{pmatrix}$$

Como nas restrições do problema reduzido, as componentes de c_k relativas às variáveis não básicas no limite superior, também aparecem multiplicadas por -1.

O problema reduzido é então:

$$\left\{\begin{array}{l} \min \sum_{k=1}^{K} \left[(c_k^I/\!-c_k^S) - c_k^B\, B_k^{-1}\, (A_k^I/\!-A_k^S)\right] \left(\dfrac{x_k^{I+}}{x_k^{S-}}\right) \\ \text{s.a.} \\ \quad \sum_{k=1}^{K} \left[(Q_k^I/\!-Q_k^S) - Q_k^B\, B_k^{-1}\, (A_k^I/\!-A_k^S)\right] \left(\dfrac{x_k^{I+}}{x_k^{S-}}\right) = \\ \qquad = b_o - \sum_{k=1}^{K} Q_k^S\, h_k^S - \sum_{k=1}^{K} Q_k^B\, \overset{o}{x}{}_k^B \\ \quad 0 \leqslant x_k^{I+} \leqslant h_k^I \quad , \quad k = 1,\ldots,K \\ \quad 0 \leqslant x_k^{S-} \leqslant h_k^S \quad , \quad k = 1,\ldots,K \end{array}\right.$$

Sendo:

$$d_k = (c_k^I/\!-c_k^S) \quad , \quad \forall k \quad ;$$

$$E_k = (Q_k^I/\!-Q_k^S) \quad , \quad \forall k \quad ;$$

$$P_k = B_k^{-1}\, (A_k^I\,/\!-A_k^S) \quad , \quad \forall k \quad ;$$

$$v_k = \left(\dfrac{x_k^{I+}}{x_k^{S-}}\right) \quad , \quad \forall k \quad ;$$

$$b' = b_o - \sum_{k=1}^{K} Q_k^S\, h_k^S - \sum_{k=1}^{K} Q_k^B\, \overset{o}{x}{}_k^B \quad ,$$

o problema reduzido se transforma em:

$$\left\{\begin{array}{l} \min \sum_{k=1}^{K} (d_k - c_k^B P_k)\, v_k \\ \text{s.a.} \\ \qquad \sum_{k=1}^{K} (E_k - Q_k^B P_k)\, v_k = b' \\ \qquad 0 \leq v_k \leq h_k^N \quad , \ k = 1,\ldots,K. \end{array}\right. \tag{B.5}$$

B.4 - VERIFICAÇÃO DA OTIMALIDADE E CORREÇÃO DOS BLOCOS NÃO ÓTIMOS:

Na verificação da otimalidade dos blocos, substituímos a solução do problema reduzido, digamos $\bar{v}_k$, $k = 1,\ldots,K$, na equação (B.3), que também pode ser escrita como:

$$x_k^B = B_k^{-1} b_k - B_k^{-1} (A_k^I / -A_k^S) \left(\frac{x_k^{I+}}{x_k^{S-}} \right) - B_k^{-1} A_k^S h_k^S =$$

$$= B_k^{-1} b_k - B_k^{-1} A_k^S h_k^S - P_k v_k =$$

$$= \overset{o}{x}{}_k^B - P_k v_k \quad , \quad k = 1,\ldots,K \, ,$$

ou seja, verificamos se

$$\bar{x}_k^B = \overset{o}{x}{}_k^B - P_k \bar{v}_k \quad , \quad \forall k \quad , \tag{B.6}$$

satisfaz $\quad 0 \leq \bar{x}_k^B \leq h_k^B \quad , \quad \forall k$.

Se $0 \leq \bar{x}_k^B \leq h_k^B$ for satisfeita para $k = 1,\ldots,K$, então o teste de otimalidade está verificado e a solução ótima do problema (1.8) será:

$$\bar{x}_k = [\underbrace{\bar{x}_k^B}_{B} / \underbrace{\bar{v}_k^I}_{I} / \underbrace{h_k^S - \bar{v}_k^S}_{S}]^T \quad , \ k = 1,\ldots,K \ ,$$

onde $\bar{v}_k^I$ é a parte da solução ótima do problema reduzido, referente ao bloco k, correspondente às variáveis que estavam no limite inferior deste bloco, e $\bar{v}_k^S$ é a parte correspondente às variáveis que estavam no limite superior.

Senão, encontramos θ_r e θ_s como mostrado no teorema (3.2). Convém notar que, agora, o conjunto J será definido como

$$J = \{j \ / \ \bar{v}_j > 0\} \tag{B.7}$$

(estamos omitindo o subscrito k relativo ao bloco; o subscrito j se refere às componentes de um bloco k específico). A definição do conjunto J é agora mais simples, pois, uma vez que todas as variáveis não básicas valiam zero, elas retornam com valores distintos dos que possuíam nos blocos quando elas são básicas no problema reduzido ou não básicas no limite superior.

Para cada bloco k não ótimo, resolvemos um dos seguintes subproblemas (equivalentes aos subproblemas (3.25) e (3.26)do capítulo 3):

$$\left\{\begin{array}{l} \min \ x_r^B \\ \text{s.a.} \\ \qquad x^B + \sum\limits_{j \in J} P^j \, x_j^N = \overset{o}{x}{}^B \\ \qquad 0 \leq x^B \leq h^B \\ \qquad 0 \leq x_j^N \leq h_j^N \quad , \ j \in J \end{array}\right\} \tag{B.8}$$

ou:

$$\left\{\begin{array}{l} \max\ x_s^B \\ \text{s.a.} \\ \qquad x^B + \sum\limits_{j \in J} P^j\, x_j^N = \overset{o}{x}{}^B \\ \qquad 0 \leqslant x^B \leqslant h^B \\ \qquad 0 \leqslant x_j^N \leqslant h_j^N \quad , \quad j \in J \end{array}\right\} \tag{B.9}$$

onde:

P^j : j-ésima coluna da matriz $P = B^{-1}\ (A^I/-A^S)$;

$\overset{o}{x}{}^B$: parte básica da solução ótima encontrada pelo subproblema k na iteração anterior.

Os programas que implementam o Método de Rosen utilizam o algorítmo descrito no capítulo 3, mas incluindo as novas partições nas variáveis não básicas descritas neste apêndice, bem como a mudança de variáveis definida por (B.1). Por este motivo, o problema reduzido utilizado é o (B.5) ao invés do (3.9); a verificação da otimalidade dos blocos é feita com base na equação (B.6); o conjunto J considerado é o (B.7) e os novos subproblemas são os (B.8) e (B.9), ao invés dos subproblemas (3.25) e (3.26).

APÊNDICE C

PROGRAMA RAROSN.F4

C.1 - DESCRIÇÃO SUMÁRIA:

C.1.1 - Propósito:

Resolver o problema da otimização global de rações pelo Método de Rosen.

C.1.2 - Entrada de dados:

Para a entrada de dados, deve ser criado um arquivo com um nome qualquer dado pelo usuário e com a extensão "DAT". Se o nome dado for "FOR20.DAT", o programa será executado sem que nada seja dito ao usuário. Se o nome do arquivo for outro qualquer, este nome será pedido ao usuário. De qualquer maneira, este arquivo deve conter as seguintes informações (em formato livre, inteiro ou real), nesta ordem:

M, N, K : variáveis inteiras, contidas na mesma linha do arquivo de dados e separadas por, pelo menos, um espaço em branco, onde:

M: número total de nutrientes;

N: número total de ingredientes;

K: número de rações a serem produzidas.

L : vetor real que contém as quantidades mínimas a serem utilizadas de cada um dos N ingredientes, distribuídos entre as K rações (limites inferiores das restrições de acoplamento) - formato: 50F.

LL : vetor real que contém as quantidades máximas a serem utilizadas de cada um dos N ingredientes, distribuídos entre as K rações (limites superiores das restrições de acoplamento) - formato: 50F.

C : vetor real que contém os custos dos N ingredientes - formato: 50F.

AM : matriz real que contém as porcentagens dos nutrientes nos ingredientes (matriz de tecnologia, nutrientes x ingredientes), dada por linhas (M linhas e N colunas) formato: 50F.

Para cada ração k, k = 1,...,K, devem ser dados:

Q(k) : variável real que contém a quantidade a ser produzida da ração k.

N2 (k) : variável inteira que contém o número de ingredientes que possuem quantidades limitadas inferior e/ou superiormente na ração k (número de variáveis canalizadas da ração k). Um ingrediente não será considerado como uma variável canalizada apenas quando ele possuir somente o limite inferior igual a zero (isto será assumido pelo programa). Em qualquer outro caso (o ingrediente possui apenas o limite inferior diferente de zero, ou apenas o limite superior, ou os dois limites), este ingrediente será considerado canalizado, e os dois limites deverão ser fornecidos.

I2(k) : vetor inteiro que contém os índices dos N2 ingredientes que possuem quantidades limitadas inferior e/ou superiormente na ração k (índices das variáveis canalizadas da ração k) - formato: 50I.

ALFA(k) : vetor real que contém os limites inferiores dos N2 ingredientes limitados da ração k - formato: 50F. Se um ingrediente possuir apenas o limite superior, deve ser fornecido o valor zero como limite inferior deste ingrediente.

BETA(k) : vetor real que contém os limites superiores dos N2 ingredientes limitados da ração k - formato: 50F. Se um ingrediente possuir apenas o limite inferior (diferente de zero), deverá ser fornecido o valor 1, como limite superior deste ingrediente.

Se N2(k) = 0, isto é, se a ração k não possui ingredientes com quantidades limitadas, não serão fornecidos os valores de I2, ALFA e BETA anteriores para esta ração.

ALFA(k) : vetor real que contém as quantidades mínimas dos M nutrientes usados na ração k (limites inferiores das restrições da ração k) - formato: 30F.

BETA(k) : vetor real que contém as quantidades máximas dos M nutrientes usados na ração k (limites superiores das restrições da ração k) - formato: 30F.

Observação: Q(k), N2(k), I2(k), ALFA(k) e BETA(k) devem ser fornecidos para toda ração k, k=1,...,K.

MMAX, MII, ND1, IMPR, TOL: variáveis que devem estar na mesma linha do arquivo de dados, separadas por, pelo menos, um espaço em branco, onde:

MMAX: variável inteira, usada no dimensionamento de matrizes e que contém uma superestimativa para o valor de M.

MII : variável inteira, usada no dimensionamento de matrizes e que contém o valor max {MMAX, NMAX} , onde NMAX é uma superestimativa para o valor de N.

ND1 : variável inteira, usada no dimensionamento de matrizes e vetores e que contém o valor max {MMAX, NMAX, KMAX}, onde KMAX é uma superestimativa para o valor de K.

IMPR: variável inteira que controla as impressões: se IMPR = 0, são impressos apenas os resultados finais (FI, X, XK1, IT e TEMP);

se IMPR ≠ 0, são impressos os resultados intermediários a cada iteração (BC, CI, F, X, XK1, V , BASE, IB, R1, Q1, BASEM) e também os resultados finais.

TOL : variável real que contém a tolerância usada no critério de parada de cada um dos subproblemas e também do problema reduzido.

SIG : vetor real de N componentes que contém uma aproximação inicial para as variáveis duais Γ .

C.1.3 - Variáveis de saída:

Para qualquer valor de "IMPR", são impressas as seguintes variáveis (se o problema tem solução ótima e limitada):

X : matriz real que contém, em cada coluna k, a solução ótima da ração k, k = 1,...,K, na iteração corrente . Para cada ração k, uma coluna de X é da forma $(x_k/u_k)^T$, onde u_k são as variáveis introduzidas para se eliminar as canalizações das restrições. No final, X conterá a solução ótima do problema, ração por ração.

XK1 : vetor real que contém, em cada iteração, a solução ótima da "ração K+1". XK1 é da forma $(u'_{K+1}/u''_{K+1})^T$, como foi mostrado na seção (3.8) do capítulo 3. No final , XK1 conterá a solução ótima do subproblema K+1.

FI : variável real que contém o custo total ótimo das K rações.

IT : variável inteira que contém o número total de iterações.

TEMP : variável real que mede o tempo de execução do programa.

Se IMPR≠ 0, são impressas, a cada iteração, as seguintes variáveis:

Para os subproblemas (rações k, k = 1,...,K):

BC : coeficientes do lado direito do subproblema k.

CI : coeficientes da função objetivo do subproblema k.

F : valor ótimo da função objetivo do subproblema k.

X : solução ótima do subproblema k, na forma $(x_k/u_k)^T$, onde as primeiras N componentes se referem às componentes de x_k e as M últimas se referem às componentes de u_k. Além disso, x_k e u_k estão sob a transformação que coloca seus limites inferiores iguais a zero.

BASE : índices das variáveis básicas na solução ótima do subproblema k.

Para o subproblema K+1:

CI : coeficientes da função objetivo do subproblema K+1 , com os sinais trocados.

F : valor ótimo da função objetivo do subproblema K+1.

XK1 : solução ótima do subproblema K+1, na forma $(u'_{K+1}/u''_{K+1})^T$ onde as N primeiras componentes são as componentes de u'_{K+1} e as N últimas são as componentes de u''_{K+1} .

IB : índices das variáveis básicas na solução ótima do subproblema K+1.

Para o problema reduzido:

R1 : coeficientes das restrições do problema reduzido.

BC : coeficientes do lado direito do problema reduzido.

Q1 : coeficientes da função objetivo do problema reduzido.

F : valor ótimo da função objetivo do problema reduzido.

V : solução ótima do problema reduzido. As componentes de V estão sob a transformação que coloca seus limites inferiores iguais a zero.

BASEM : índices das variáveis básicas na solução ótima do problema reduzido.

São também impressos a nova solução de um subproblema quando se substitui os valores das variáveis V, vindas do problema reduzido, neste subproblema; o número da iteração corrente e se um bloco é ótimo ou não.

O arquivo de saída criado pelo programa tem o nome de "FOR03.DAT".

C.1.4 - Dimensionamento de matrizes e vetores usados pelo programa:

As variáveis abaixo são usadas no dimensionamento de matrizes e vetores de RAROSN.F4:

KMAX : superestimativa para o valor de K.

MMAX : superestimativa para o valor de M.

NMAX : superestimativa para o valor de N.

ND1 : max {MMAX, NMAX, KMAX} .

ND2 : N2MAX + MMAX, onde $N2MAX = \max_{1 \leq k \leq K} \{N2(k)\}$ e pode ser superestimada.

NDV : NMAX . (KMAX + 1) .

MII : max {MMAX, NMAX} .

NII : NMAX + MMAX .

O programa está dimensionado, atualmente, com os seguintes valores: KMAX = 5, MMAX = 30, NMAX = 40, ND1 = 40, ND2 = 70, NDV = 240, MII = 40 e NII = 70. Qualquer mudança a ser efetuada nas dimensões deve ser feita apenas no programa principal. Para isso, citamos a seguir, as dimensões das matrizes e dos vetores utilizados.

matrizes e vetores inteiros	dimensões
INDP	NMAX
N2	KMAX
I2	ND2 x KMAX
E	ND2 x KMAX
BASE	2 x ND1 x KMAX
IA	ND1
JOTA	MII
JLS	MMAX
NB	NMAX x KMAX
I2M	NDV
EM	NDV
BASEM	2 x ND1
IB	NMAX

matrizes e vetores reais	dimensões
CI	NII
H	ND2 x KMAX
X	NII x KMAX
XK1	2. NMAX
V	NDV
B1	ND1 x ND1 x KMAX
BC	ND1
AC	ND1
PI	ND1
ALFA	ND2 x KMAX
BETA	ND2 x KMAX
AM	MMAX x NMAX
C	NII
Q	KMAX
L	NMAX
LL	NMAX
P	MMAX x NMAX
Q1	NDV
R1	MII x NDV
A	MII x NII
HM	NDV
B1M	ND1 x ND1
SIG	NMAX
B	NMAX

C.1.5 - <u>Subrotinas usadas pelo programa</u>:

O programa RAROSN.F4 utiliza 4 subrotinas: KALKP, SUBPR, SAIDA e PIVOTA. A subrotina KALKP, incluída no arquivo RAROSN.F4, atualiza a parte não básica da matriz de tecnologia (A/I) de cada ração. As outras 3 subrotinas estão no arquivo SUBPR.F4, que resolve os subproblemas e o problema reduzido pelo Método Simplex Revisado com variáveis canalizadas.

SUBPR é a subrotina principal do Método Simplex canalizado: controla as fases I e II, escolhe a variável de entrada na base, monta a solução final e faz todas as impressões.

A subrotina SAIDA, chamada por SUBPR, faz o critério de saída da base do Método Simplex canalizado.

PIVOTA faz o pivotamento sobre o elemento pivô, atualizando a matriz inversa da base e o lado direito do problema. É chamada por SAIDA e por SUBPR.

C.2 - LISTAGEM DO PROGRAMA:

Programa RAROSN.F4 e subrotinas KALKP, SUBPR, SAIDA e PIVOTA :

```
C
C*****************************************************************
C  PROGRAMA PRINCIPAL
C*****************************************************************
C
C  PROPOSITO : RESOLVER O PROBLEMA DA RACAO PELO METODO DE ROSEN.
C  ---------
C
       COMMON /ENTR/ TOL,IT,IMPR,P2
       INTEGER P2,R,S,SP,SAI,OTIM,INDP(40),N2(5),I2(70,5),
      1E(70,5),BASE(2,40,5),IA(40),JOTA(40),JLS(30),NB(40,5),
      1I2M(240),EM(240),BASEM(2,40),IB(40)
       REAL CI(70),H(70,5),X(70,5),XK1(80),V(240),B1(40,40,5),
      1BC(40),AC(40),PI(40),ALFA(70,5),BETA(70,5),AM(30,40),
      1C(70),Q(5),L(40),LL(40),P(30,40),Q1(240),R1(40,240),
      1B(40,70),HM(240),B1M(40,40),SIG(40),B(40)
C
C  ENTRADA DE DADOS
C
       READ(20,10010) M,N,K
       READ(20,10020)(L(I),I=1,N)
       READ(20,10020) (LL(I),I=1,N)
       READ(20,10020) (C(I),I=1,N)
       DO 10 I=1,M
10     READ(20,10020) (AM(I,J),J=1,N)
       DO 30 KK=1,K
       READ(20,10030) Q(KK)
       READ(20,10005) N2(KK)
       IF(N2(KK).GE.0.AND.N2(KK).LE.N) GO TO 15
       WRITE(3,10060) KK,N2(KK)
       CALL EXIT
15     IF(N2(KK).EQ.0) GO TO 25
       READ(20,10050) (I2(I,KK),I=1,N2(KK))
       READ(20,10020) (ALFA(I,KK),I=1,N2(KK))
       READ(20,10020) (BETA(I,KK),I=1,N2(KK))
25     READ(20,10040) (ALFA(N2(KK)+I,KK),I=1,M)
       READ(20,10040) (BETA(N2(KK)+I,KK),I=1,M)
       DO 27 I=1,M
27     I2(N2(KK)+I,KK)=N+I
       N2(KK)=N2(KK)+M
30     CONTINUE
       READ(20,10055) MMAX,MII,ND1,IMPR,TOL
       READ(20,10020) (SIG(I),I=1,N)
C
C  FIN DA ENTRADA DE DADOS
C
```

```
      CALL SETRUN(0)
      IERR=0
C
C  TRANSFORMA AS CANALIZACOES PARA ZERO E H, ONDE H=BETA-ALFA
C
      DO 40 KK=1,K
      DO 35 I=1,N2(KK)
35    H(I,KK)=BETA(I,KK)-ALFA(I,KK)
40    CONTINUE
C
C  SUBTRAI DE LL A PARTE CORRESPONDENTE AOS LIMITES INFERIORES
C  DAS VARIAVEIS CANALIZADAS ( Q(K) * ALFA(K) )
C
      DO 42 I=1,N
42    PI(I)=0.
      DO 50 KK=1,K
      IF(N2(KK).EQ.M) GO TO 50
      DO 45 J=1,N2(KK) - M
45    PI(I2(J,KK))=PI(I2(J,KK))+Q(KK) * ALFA(J,KK)
50    CONTINUE
      DO 52 I=1,N
52    B(I)=LL(I)-PI(I)
      DO 53 I=N+1,N+M
53    C(I)=0.
      DO 54  I=1,N
      IF(B(I).GE.0.) GO TO 54
      WRITE(3,10190)
      CALL  EXIT
54    CONTINUE
C
      PZ=0
      NB=0
      IT=1
      IF(IMPR.NE.0) WRITE(3,10120) IT
C
C  RESOLUCAO DOS K SUBPROBLEMAS
C
      NB=M
      NO=N+M
      NOL=N
      DO 100 KK=1,K
      N2KK=N2(KK)
      F=0.
C
C  MONTA A MATRIZ DE TRABALHO A = (AM/I)
C
      DO 55 I=1,M
      DO 55 J=1,N
55    A(I,J)=AM(I,J)
      DO 58 I=1,M
```

```
      DO 58 J=N+1,NC
      IF(J.EQ.I+N) GO TO 56
      A(I,J)=0.
      GO TO 58
56    A(I,J)=1.
58    CONTINUE
C
C  MONTA O LADO DIREITO DO SUBPROBLEMA KK:
C  BC = BETA - A*ALFA
C
      K1=N2(KK)-M
      DO 60 I=1,M
60    BC(I)=BETA(K1+I,KK)
      IF(N2(KK).EQ.M) GO TO 75
      DO 70 I=1,M
      PI(I)=0.
      DO 65 J=1,K1
65    PI(I)=PI(I)+A(I,I2(J,KK))*ALFA(J,KK)
70    BC(I)=BC(I)-PI(I)
C
C  MONTA OS COEFICIENTES DA FUNCAO OBJETIVO DO SUBPROBLEMA KK
C
75    DO 80 I=1,NC
80    CI(I)=0.

      DO 85 I=1,N
85    CI(I)=(C(I)-SIG(I)) * Q(KK)
      CALL SUBPR(KK,NL,NC,CI,A,NP,INDP,N2(KK),I2(1,KK),
     1W(1,KK),NC,MII,NCL,NL,ND1,N2KK,F,X(1,KK),V,IERR,
     1B1(1,1,KK),E(1,KK),BASE(1,1,KK),BC,AC,PI,IA)
      IF(IERR.NE.0) CALL EXIT
C
C  ORDENA O VETOR X (PARTE BASICA E NAO BASICA)
C  MONTA O VETOR NB DE INDICES DAS VARIAVEIS NAO BASICAS
C
      DO 90 I=1,NC
90    CI(I)=X(I,KK)
      DO 92 I=1,M
92    X(I,KK)=CI(BASE(1,I,KK))
      K1=0
      DO 96 J=1,NC
      DO 94 I=1,M
      IF(J.EQ.BASE(1,I,KK)) GO TO 96
94    CONTINUE
      K1=K1+1
      NB(K1,KK)=J
      X(M+K1,KK)=CI(J)
96    CONTINUE
100   CONTINUE
```

```
C
C  RESOLVE O SUBPROBLEMA (K+1)
C
       KK=K+1
       F=0.
       K2=2*N
       DO 102 I=1,K2
102    XK1(I)=0.
       DO 106 I=1,N
       IF(SIG(I).GT.0.) GO TO 104
       XK1(I+N)=LL(I)-L(I)
       IB(I)=N+I
       GO TO 106
104    XK1(I)=LL(I)-L(I)
       IB(I)=I
       F=F-SIG(I)*XK1(I)
106    CONTINUE
       IF(IMPR.EQ.0) GO TO 110
       WRITE(3,10070) KK
       WRITE(3,10080) (SIG(I),I=1,N)
       WRITE(3,10090) F
       WRITE(3,10100)
       WRITE(3,10110) KK,(XK1(I),I=1,K2)
       WRITE(3,10115) (IB(I),I=1,N)
C
C  LOOP PARA MONTAGEM DO PROBLEMA MESTRE E CORRECAO DOS SUBPROBLEMAS
C
110    IT=IT+1
       IF(IMPR.NE.0) WRITE(3,10120) IT
       DO 111 I=1,N
       BC(I)=0.
111    PI(I)=0.
       K2=0
       NB=M
       NC=N+M
       NCL=N
```

```
       DO 170 KK=1,K
       N2KK=N2(KK)
C
C  TESTA SE O SUBPROBLEMA KK ERA OTIMO NA ITERACAO ANTERIOR
C
       IF(IA(KK).EQ.KK) GO TO 162
C
C  SUBPROBLEMA KK NAO ERA OTIMO
C  CALCULA A MATRIZ P = B1*A(NB)
C
```

```
      CALL KALKP(M,N,B1(1,1,KK),AM,NB(1,KK),N2(KK),I2(1,KK),E(1,KK),
     1ND1,N2KK,MMAX,P)
C
C  MONTA A FUNCAO OBJETIVO DO PROBLEMA MESTRE (Q1)
C  REFERENTE AO SUBPROBLEMA KK, Q1 = CNB - CB*P
C
      DO 120 I=1,N
      Q1(I+K2)=0.
      DO 115 J=1,M
115   Q1(I+K2)=Q1(I+K2)+P(J,I)*C(BASE(1,J,KK))
      DO 116 J=1,N2(KK)
      IF(NB(I,KK).EQ.I2(J,KK)) GO TO 117
116   CONTINUE
      GO TO 119
117   IF(E(J,KK).EQ.1) GO TO 119
      Q1(I+K2)=(-C(NB(I,KK))-Q1(I+K2)) * Q(KK)
      GO TO 120
119   Q1(I+K2)=(C(NB(I,KK))-Q1(I+K2)) * Q(KK)
120   CONTINUE
C
C  MONTA A MATRIZ DE RESTRICOES DO PROBLEMA MESTRE (R1)
C  REFERENTE AO SUBPROBLEMA KK, R1 = QB * P
C
      DO 125 I=1,N
      DO 125 J=1,N
125   R1(I,J+K2)=0.
      DO 140 I=1,N
      DO 135 J=1,M
      IF(BASE(1,J,KK).NE.I) GO TO 135
      DO 130 K1=1,N
130   R1(I,K1+K2)=Q(KK)*P(J,K1)
      GO TO 140
135   CONTINUE
140   CONTINUE
C
C  R1 = QNB - R1
C
      DO 161 I=1,N
      IF(NB(I,KK).GT.N) GO TO 155
      DO 142 J=1,N2(KK)
      IF(NB(I,KK).EQ.I2(J,KK)) GO TO 143
142   CONTINUE
      GO TO 146
143   IF(E(J,KK).EQ.1) GO TO 146
      DO 145 K1=1,N
      IF(K1.EQ.NB(I,KK)) GO TO 144
      R1(K1,I+K2) = -R1(K1,I+K2)
      GO TO 145
```

```
144     R1(K1,I+K2) = -Q(KK)-R1(K1,I+K2)
145     CONTINUE
        GO TO 161
146     DO 148 K1=1,N
        IF(K1.EQ.NB(I,KK)) GO TO 147
        R1(K1,I+K2) = -R1(K1,I+K2)
        GO TO 148
147     R1(K1,I+K2) = Q(KK)-R1(K1,I+K2)
148     CONTINUE
        GO TO 161
155     DO 160 J=1,N
160     R1(J,I+K2) = -R1(J,I+K2)
161     CONTINUE
C
C  MONTA O LADO DIREITO DO PROBLEMA MESTRE:
C
C  BC = SOMAT(QB *XB) PARA OS K SUBPROBLEMAS
C
162     DO 164 I=1,N
        DO 163 J=1,M
        IF(BASE(1,J,KK).NE.I) GO TO 163
        BC(I)=BC(I)+Q(KK)*X(J,KK)
        GO TO 164
163     CONTINUE
164     CONTINUE
C
C  PI = SOMAT(QK(NB) * HK), PARA AS VARIAVEIS CANALIZADAS
C  NAO BASICAS NO LIMITE SUPERIOR
C
        DO 168 I=1,N
        IF(NB(I,KK).GT.N) GO TO 168
        DO 165 K1=1,N2(KK)
        IF(NB(I,KK).EQ.I2(K1,KK)) GO TO 167
165     CONTINUE
        GO TO 168
167     IF(E(K1,KK).EQ.1) GO TO 168
        PI(NB(I,KK))=PI(NB(I,KK))+Q(KK)*H(K1,KK)
168     CONTINUE
        K2=K2+N
170     CONTINUE
C
C  MONTA A MATRIZ DE RESTRICOES DO PROBLEMA MESTRE (R1)
C  REFERENTE AO SUBPROBLEMA (K+1)
C  R1 = QNB - QB
C
        K2=N*K
        DO 175 I=1,N
175     Q1(I+K2)=0.
        DO 180 I=1,N
        DO 180 J=1,N
180     R1(I,J+K2)=0.
```

```
      DO 190 I=1,N
      IF(IB(I).GT.N) GO TO 185
      R1(IB(I),I+K2)=-1.
      GO TO 190
185   R1(IB(I)-N,I+K2)=1.
190   CONTINUE
C
C  AUMENTA BC NA PARTE REFERENTE AO SUBPROBLEMA (K+1)
C  BC = BC + QB(K+1) * XB(K+1)




C
      DO 200 I=1,N
      DO 195 J=1,N
      IF(IB(J).NE.I) GO TO 195
      BC(I)=BC(I)+XK1(I)
      GO TO 200
195   CONTINUE
200   CONTINUE
C
C  LADO DIREITO COMPLETO DO PROBLEMA MESTRE:
C  BC  =  B - BC - PI
C
      DO 210 I=1,N
      BC(I)=B(I)-BC(I)-PI(I)
210   P2(I)=0.
C
C  MONTA HM,I2M,N2M PARA O PROBLEMA MESTRE
C
      N2M=0
      K1=0
      DO 250 KK=1,K
      DO 250 I=1,N
      K1=K1+1
      DO 225 J=1,N2(KK)
      IF(NB(I,KK).EQ.I2(J,KK)) GO TO 230
225   CONTINUE
      GO TO 250
C
C  VARIAVEL NAO BASICA DO SUBPROBLEMA KK E' CANALIZADA
C
230   N2M=N2M+1
      I2M(N2M)=K1
      HM(N2M)=BETA(J,KK)-ALFA(J,KK)
250   CONTINUE
C
C  RESOLVE O PROBLEMA MESTRE
C
      P2=2
      NB=N
```

```
      NO=N*(K+1)
      NDV=NC
      NOL=N
      F=0.
      KK=0
      CALL SUBPR(KK,NL,NC,Q1,R1,NP,INDP,N2M,I2M,HM,NDV,MII,NCL,
     1NDV,ND1,N2M,F,X,V,IERR,B1M,EM,BASEM,BC,AC,PI,IA)
      IF(IERR.NE.0) CALL EXIT
      IF(IMPR.EQ.0) GO TO 310
      WRITE(3,10125)
C
C  LOOP PARA VERIFICACAD DA OTIMALIDADE NOS BLOCOS
C  E CORRECAO DOS BLOCOS NAO OTIMOS
C
310   K3=0
      OTIM=0
      DO 315 I=1,K
315   IA(I)=0
      NB=M
      NO=N+M
      NOL=N
```

```
      DO 600 KK=1,K
      N2KK=N2(KK)
C
C  CALCULA A MATRIZ P = B1 * A(NB)
C
      CALL KALKP(M,N,B1(1,1,KK),AM,NB(1,KK),N2(KK),I2(1,KK),E(1,KK),
     1ND1,N2KK,MMAX,P)
C
C  PI = P * V
C
      DO 320 I=1,M
      PI(I)=0.
      DO 320 J=1,N
320   PI(I)=PI(I)+P(I,J)*V(J+K3)
C
C  X* = X - P*V
C
      DO 330 I=1,M
      BO(I)=X(I,KK)
      X(I,KK)=X(I,KK)-PI(I)
330   PI(I)=0.
      IF(IMPR.EQ.0) GO TO 335
      WRITE(3,10130) KK
      WRITE(3,10140) (X(I,KK),I=1,M)
C
C  TESTA SE X* ESTA DENTRO DE SEUS LIMITES DE CANALIZACAO
C
```

```
335     K1=0
        K2=0
        DO 360 I=1,M
        IF(BASE(2,I,KK).EQ.1) GO TO 340
C
C  VARIAVEL NAO E' CANALIZADA
C
        IF(X(I,KK).GT.-TOL) GO TO 360
        GO TO 355
C
C  VARIAVEL E' CANALIZADA
C
340     DO 345 J=1,N2(KK)
        IF(BASE(1,I,KK).EQ.I2(J,KK)) GO TO 350
345     CONTINUE
        WRITE(3,10150)
        CALL EXIT
350     IF(X(I,KK).GE.-TOL) GO TO 357
355     K1=K1+1
        JOTA(K1)=I
        GO TO 360
357     IF(X(I,KK).LE.H(J,KK)) GO TO 360
        K2=K2+1
        JBS(K2)=I
360     CONTINUE
        IF(K1+K2.NE.0) GO TO 375
        OTIM=OTIM+1
        IA(KK)=KK
        IF(IMPR.NE.0) WRITE(3,10160) KK
        DO 370 I=1,M
        A(KK,I)=X(I,KK)
370     X(I,KK)=BC(I)
        GO TO 597


C
C  CALCULO DE TETA (DETERMINACAO DA VARIAVEL DE SAIDA DO BLOCO KK)
C
C  TETA(R) = MIN ( TETA(J) = X(J) / (X(J)-X*(J) )
C
375     IF(IMPR.NE.0) WRITE(3,10180) KK
        IA(KK)=0
        TETAR=1.E30
        R=0
        IF(K1.EQ.0) GO TO 395
        DO 390 I=1,K1
        AUX=BC(JOTA(I))-X(JOTA(I),KK)
```

```
      IF(ABS(AUX).LT.TOL) GO TO 390
      TETA=BC(JOTA(I))/AUX
      IF(TETAR.LE.TETA) GO TO 390
      TETAR=TETA
      R=JOTA(I)
390   CONTINUE
      IF(R.NE.0) GO TO 395
      WRITE(3,10170) KK
      CALL EXIT
C
C  TETA(S) = MIN ( TETA(J) = (X(J)-H(J)) / (X(J)-X*(J)) )
C
395   TETAS=1.E30
      SP=0
      IF(K2.EQ.0)GO TO 430
      DO 420 I=1,K2
      AUX=BC(JLS(I))-X(JLS(I),KK)
      IF(ABS(AUX).LT.TOL) GO TO 420
      DO 400 J=1,N2(KK)
      IF(BASE(1,JLS(I),KK).EQ.T2(J,KK)) GO TO 410
400   CONTINUE
      WRITE(3,10150)
      CALL EXIT
410   TETA=(BC(JLS(I))-H(J,KK))/AUX
      IF(TETAS.LE.TETA) GO TO 420
      TETAS=TETA
      SP=JLS(I)
420   CONTINUE
      IF(SP.NE.0) GO TO 430
      WRITE(3,10170) KK
      CALL EXIT
C
C  RESOLVE O SUBPROBLEMA KK A PARTIR DA SOLUCAO ANTERIOR
C
430   P2=3
      F=0.
C
C  TETA = MIN ( TETA(R),TETA(S) )
C  MONTA A NOVA FUNCAO OBJETIVO DO BLOCO KK
C
      DO 435 I=1,NC
435   C2(I)=0.
      IF(TETAR.LE.TETAS) GO TO 440
      SAI=BASE(1,SP,KK)
      C2(SAI)=-1.
      TETA=TETAS
      GO TO 550
440   SAI=BASE(1,R,KK)
```

```
      CI(SAI)=1.
      TETA=TETAR
C
C  MONTA O VETOR 'INDP' QUE CONTEM OS INDICES DAS VARIAVEIS
C  NAO BASICAS CANDIDATAS A ENTRAR NA BASE DO SUBPROBLEMA KK
C
550   NP=0
      DO 555 J=1,N
      IF(V(J+K3).EQ.0.) GO TO 555
      NP=NP+1
      INDP(NP)=NB(J,KK)
555   CONTINUE
C
C  MONTA A MATRIZ DE TRABALHO A = (AM/I)
C
      DO 560 I=1,M
      DO 560 J=1,N
560   A(I,J)=AM(I,J)
      DO 570 I=1,M
      DO 570 J=N+1,NC
      IF(J.EQ.I+N) GO TO 565
      A(I,J)=0.
      GO TO 570
565   A(I,J)=1.
570   CONTINUE
C
C  TROCA O SINAL DAS COLUNAS DA MATRIZ A QUE POSSUEM VARIAVEIS
C  CANALIZADAS NO LIMITE SUPERIOR
C
      DO 580 J=1,NC
      DO 572 I=1,N2(KK)
      IF(J.EQ.I2(I,KK)) GO TO 574
572   CONTINUE
      GO TO 580
574   IF(E(I,KK).EQ.1) GO TO 580
      DO 576 K1=1,M
576   A(K1,J)=-A(K1,J)
580   CONTINUE
C
      CALL SUBPR(KK,NL,NC,CI,A,NP,INDP,N2(KK),I2(1,KK),
      1M(1,KK),NC,MII,NCL,NL,ND1,N2KK,F,X(1,KK),V,IERR,
      1B1(1,1,KK),E(1,KK),BASE(1,1,KK),BC,AC,PI,IA)
      IF(IERR.NE.0) CALL EXIT
C
C  ORDENA O VETOR X (PARTE BASICA E NAO BASICA)
C  MONTA O VETOR NB DE INDICES DE VARIAVEIS NAO BASICAS
C
      DO 586 I=1,NC
586   CI(I)=X(I,KK)
      DO 588 I=1,M
588   X(I,KK)=CI(BASE(1,I,KK))
```

```
      K1=0
      DO 595 J=1,NC
      DO 590 I=1,M
      IF(J.EQ.BASE(1,I,KK)) GO TO 595
590   CONTINUE
      K1=K1+1
      NB(K1,KK)=J
      X(M+K1,KK)=CI(J)
595   CONTINUE




597   K3=K3+N
600   CONTINUE
C
C  VERIFICACAO DO BLOCO (K+1)
C
      KK=K+1
      K3=N*K
C
C  U(K+1) = U(K+1) - I*V
C
      DO 605 I=1,N
      BC(I)=XK1(IB(I))
605   XK1(IB(I))=XK1(IB(I))-V(I+K3)
      IF(IMPR.EQ.0) GO TO 610
      WRITE(3,10130) KK
      WRITE(3,10140) (XK1(IB(I)),I=1,N)
610   K1=0
      K3=0
      DO 615 I=1,N
      IF(XK1(IB(I)).GE.-TOL) GO TO 615
      K1=K1+1
      JOTA(K1)=I
615   CONTINUE
      IF(K1.EQ.0) GO TO 640
C
C  CORRECAO DO BLOCO (K+1)
C
      IF(IMPR.NE.0) WRITE(3,10180) KK
      TETAR=1.E30
      R=0
      DO 620 I=1,K1
      AUX=BC(JOTA(I))-XK1(IB(JOTA(I)))
      IF(ABS(AUX).LT.TOL) GO TO 620
      TETA=BC(JOTA(I))/AUX
      IF(TETAR.LE.TETA) GO TO 620
      TETAR=TETA
      R=JOTA(I)
620   CONTINUE
      IF(R.NE.0) GO TO 625
```

```
      WRITE(3,10170) KK
      CALL EXIT
625   SAI=IB(R)
      IF(SAI.GT.N) GO TO 630
      IB(R)=N+SAI
      GO TO 635
630   IB(R)=SAI-N
635   DO 637 I=1,N
637   XK1(IB(I))=BC(I)
      XK1(SAI)=0.
      K1=2*N
      F=0.
      CI(SAI)=1.
      F=F+CI(SAI)*XK1(SAI)
      IF(IMPR.EQ.0) GO TO 650
      WRITE(3,10070) KK
      WRITE(3,10090) F
      WRITE(3,10100)
      WRITE(3,10110) KK,(XK1(I),I=1,K1)
      WRITE(3,10115) (IB(I),I=1,N)
      GO TO 650


640   OTIM=OTIM+1
      IF(IMPR.NE.0) WRITE(3,10160) KK
      DO 645 J=1,N
      A(KK,J)=XK1(IB(J))
645   XK1(IB(J))=BC(J)
650   IF(OTIM.NE.K+1) GO TO 110
C
C  FIM
C  MONTA A SOLUCAO FINAL (X/U) PARA OS K SUBPROBLEMAS
C
      K2=0
      DO 690 KK=1,K
      DO 655 J=1,N
655   CI(J)=X(M+J,KK)
      DO 660 I=1,M
660   X(BASE(1,I,KK),KK)=A(KK,I)
      DO 680 I=1,N
      K1=NB(I,KK)
      DO 665 J=1,N2(KK)
      IF(K1.EQ.I2(J,KK)) GO TO 670
665   CONTINUE
      GO TO 675
670   IF(E(J,KK).EQ.1) GO TO 675
      X(K1,KK)=CI(I)-V(I+K2)
      GO TO 680
675   X(K1,KK)=V(I+K2)
680   CONTINUE
```

```
690     K2=K2+N
C
C  RETORNA AS VARIAVEIS CANALIZADAS PARA ALFA E BETA
C
        DO 700 KK=1,K
        DO 700 I=1,N2(KK)
700     X(I2(I,KK),KK)=X(I2(I,KK),KK)+ALFA(I,KK)
C
C  SUBPROBLEMA (K+1) : U = L+U'
C
        KK=K+1
        K2=K*N
        DO 702 I=1,N
702     XK1(IB(I))=A(KK,I)
        DO 715 I=1,N
        IF(IB(I).LE.N) GO TO 710
        J=IB(I)-N
        XK1(J)=V(I+K2)
        GO TO 715
710     XK1(N+I)=V(I+K2)
715     CONTINUE
        DO 720 I=1,N
        XK1(I)=XK1(I)+L(I)
720     XK1(N+I)=XK1(N+I)+L(I)
C
C  CALCULA O VALOR DA FUNCAO OBJETIVO : FI = SOMAT (QK * C * XK)
C
        FI=0.
        DO 731 KK=1,K
        W=0.
        DO 730 I=1,N
730     W=W+X(I,KK)*C(I)
731     FI=FI+W*Q(KK)




        CALL RUNTIME(ITEMP)
        TEMP=FLOAT(ITEMP)*1.E-3
C
C  ESCREVE A SOLUCAO FINAL
C
        IF(K.EQ.1) GO TO 740
        WRITE(3,10220) K,FI
        GO TO 750
740     WRITE(3,10225) FI
750     WRITE(3,10200)
        DO 760 J=1,K
```

```
      WRITE(3,10210) J,(X(I,J),I=1,N)
760   WRITE(3,10215) J,(X(I,J),I=N+1,N+M)
      J=K+1
      WRITE(3,10215) J,(XK1(I),I=1,2*N)
      WRITE(3,10230) IT
      WRITE(3,10240) TEMP
      CALL EXIT

10005 FORMAT(I)
10010 FORMAT(3I)
10020 FORMAT(50F)
10030 FORMAT(F)
10040 FORMAT(30F)
10050 FORMAT(50I)
10055 FORMAT(4I,F)
10060 FORMAT(/,1X,'NUMERO DE VARIAVEIS CANALIZADS NO BLOCO ',
      2I4,' INVALIDO (',I6,' )')
10070 FORMAT(///,1X,'* SUBPROBLEMA NUMERO',I3,/)
10080 FORMAT(3X,'-CI : ',50G)
10090 FORMAT(/,3X,'VALOR DA FUNCAO OBJETIVO: ',G)
10100 FORMAT(/,3X,'SOLUCAO BASICA OTIMA:',/)
10110 FORMAT(3X,'U',I3,' : ',50G)
10115 FORMAT(/,3X,'BASE :',50I4)
10120 FORMAT(////,1X,'*** ITERACAO NUMERO',I4,/)
10125 FORMAT(//)
10130 FORMAT(////,3X,'NOVA SOLUCAO DO SUBPROBLEMA ',I3,' :',/)
10140 FORMAT(3X,'XB :',50G)
10150 FORMAT(/,1X,'ERRO - INDICE DE VARIAVEL CANALIZADA DEVERIA
      1 ESTAR EM I2')
10160 FORMAT(/,3X,'BLOCO ',I3,'  E OTIMO',/)
10170 FORMAT(/,1X,'ERRO - RACAO ',I3,' CONTINUA INFACTIVEL')
10180 FORMAT(/,3X,'BLOCO ',I3,'  NAO E OTIMO',/)
10190 FORMAT(//,1X,'O PROBLEMA NAO TEM SOLUCAO FACTIVEL')
10200 FORMAT(///,1X,'SOLUCAO OTIMA :',//)
10210 FORMAT(//,1X,'X',I3,' = ',50G)
10215 FORMAT(/,1X,'U',I3,' = ',50G)
10220 FORMAT(////,1X,'CUSTO TOTAL DAS ',I2,' RACOES :',G,//)
10225 FORMAT(////,1X,'CUSTO TOTAL DA RACAO : ',G,//)
10230 FORMAT(/////,1X,'NUMERO TOTAL DE ITERACOES : ',I4)
10240 FORMAT(//,1X,'TEMPO DE EXECUCAO (SEGUNDOS) :',F12.2)
      END
C
C**************************************************************
C  CALCULA A MATRIZ P = B1 * A(NB)
C**************************************************************
C
      SUBROUTINE KALKP(M,N,B1,AM,NB,N2,I2,E,ND1,ND2,MMAX,P)
      INTEGER NB(N),I2(ND2),E(ND2)
      REAL B1(ND1,ND1),AM(MMAX,N),P(MMAX,N)
```

```
      DO 10 I=1,M
      DO 10 J=1,N
10    P(I,J)=0.
      DO 40 I=1,M
      DO 40 K1=1,N
      IF(NB(K1).GT.N) GO TO 20
      DO 15 J=1,M
15    P(I,K1)=P(I,K1)+B1(I,J)*AM(J,NB(K1))
      GO TO 25
20    P(I,K1)=B1(I,NB(K1)-N)
25    DO 30 J=1,N2
      IF(NB(K1).EQ.I2(J)) GO TO 35
30    CONTINUE
      GO TO 40
35    IF(E(J).NE.1) P(I,K1)=-P(I,K1)
40    CONTINUE
      RETURN
      END
```

```
C
C***********************************************************************
C  RESOLVE OS K SUBPROBLEMAS E O PROBLEMA MESTRE
C***********************************************************************
C
      SUBROUTINE SUBPR(KK,NL,NC,CI,A,NP,INDP,N2,I2,H,NII,MII,
     1NM,NDV,ND1,ND2,F,X,V,IERR,B1,E,BASE,BC,AC,PI,IA)
      COMMON /ENTR/ TOL,IT,IMPR,P2
      INTEGER R,S,P2,INDP(NM),N2,I2(ND2),E(ND2),BASE(2,ND1),IA(ND1)
      REAL CI(NII),A(MII,NII),H(ND2),X(NII),V(NDV),B1(ND1,ND1),
     1BC(ND1),AC(ND1),PI(ND1)
C
      IF(IMPR.EQ.0) GO TO 30
      IF(P2.EQ.2) GO TO 10
      WRITE(3,10070) KK
      GO TO 25
10    WRITE(3,10075)
      WRITE(3,10100)
      DO 20 I=1,NL
20    WRITE(3,10110) (A(I,J),J=1,NC)
25    WRITE(3,10030) (BC(I),I=1,NL)
30    IF(IT.EQ.1.OR.P2.EQ.2) GO TO 40
      GO TO 300
C
C  TESTA SE BC(I) >= 0
C
```

```
40      DO 45 I=1,NL
45      V(I)=1.
        DO 60 I=1,NL
        IF(BC(I).GE.0.)    GO TO 60
        DO 50 J=1,NC
50      A(I,J)=-A(I,J)
        BC(I)=-BC(I)
        V(I)=-1.
60      CONTINUE
C
C  FASE 1
C  INICIALIZACAO DE B1(COM VARIAVEIS ARTIFICIAIS) E,BASE INICIAL
C
        DO 100 I=1,NL
        DO 100 J=1,NL
        IF(I.EQ.J)     GO TO 90
        B1(I,J)=0.
        GO TO 100
90      B1(I,I)=1.
100     CONTINUE
        DO 130 I=1,NL
        IA(I)=0
        BASE(1,I)=NC+I
130     BASE(2,I)=0
        IF(N2.EQ.0) GO TO 140
        DO 135 I=1,N2
135     E(I)=1
C
C  PI=DB*B1
C
140     DO 150 I=1,NL
        PI(I)=0.
        DO 150 J=1,NL
        IF(BASE(1,J).LE.NC) GO TO 150



        PI(I)=PI(I)+B1(J,I)
150     CONTINUE
C
C  DC(J)=B(J)-PI*AJ, J NAO PERTENCENTE A BASE, DC(J)<0
C  ( D SERIA O VETOR DOS COEFICIENTES DA F.O. ARTIFICIAL)
C  ESCOLHA DA VARIAVEL DE ENTRADA X(S)
C
160     DO 190 J=1,NC+NL
        DO 170 K1=1,NL
        IF(J.EQ.BASE(1,K1))    GO TO 190
170     CONTINUE
```

```
C
C  J NAO PERTENCE A BASE
C
       IF(J.LE.NC) GO TO 175
       AUX=PI(J-NC)
       DC=1.-AUX
       GO TO 185
175    AUX=0.
       DO 180 I=1,NL
180    AUX=AUX+PI(I)*A(I,J)
       DC=-AUX
185    IF(DC.LT.-TOL)    GO TO 210
190    CONTINUE
C
C  FIM DA FASE 1
C
       W=0.
       DO 200 I=1,NL
       IF(BASE(1,I).LE.NC) GO TO 200
       W=W+BC(I)
200    CONTINUE
       IF(ABS(W).LE.TOL)    GO TO 220
       WRITE(3,10060) KK,W,IT
       WRITE(3,10066)
       DO 205 I=1,2
205    WRITE(3,10067) I,(BASE(I,J),J=1,NL)
       WRITE(3,10030) (BC(I),I=1,NL)
       WRITE(3,10068) (H(I),I=1,N2)
       WRITE(3,10069) (E(I),I=1,N2)
       IERR=1
       RETURN
C
C  X(S) ENTRA NA BASE
C
210    S=J
       CALL SAIDA(KK,S,NL,NC,CI,A,N2,I2,H,NII,MII,ND1,
      1ND2,R,IERR,B1,E,BASE,BC,AC)
       IF(R.EQ.0)    GO TO 160
       GO TO 140
C
C  VERIFICA SE EXISTE ALGUMA VARIAVEL ARTIFICIAL NA BASE
C
220    JJ=0
       DO 230 I=1,NL
       IF(BASE(1,I).LE.NC)  GO TO 230
       JJ=JJ+1
       IA(JJ)=I
230    CONTINUE
       IF(JJ.EQ.0)  GO TO 300
```

```
      DO 290 I=1,JJ
      DO 260 S=1,NC
      DO 240 J=1,NL
      IF(S.EQ.BASE(1,J))  GO TO 260
240   CONTINUE
      DO 250 K1=1,NL
      AC(K1)=0.
      DO 250 J=1,NL
250   AC(K1)=AC(K1)+B1(K1,J)*A(J,S)
      R=IA(I)
      IF(AC(R).NE.0.)  GO TO 270
260   CONTINUE
      WRITE(3,10060) KK,W,IT
      WRITE(3,10065)
      WRITE(3,10066)
      DO 265 K1=1,2
265   WRITE(3,10067) K1,(BASE(K1,J),J=1,NL)
      WRITE(3,10030) (BC(K1),K1=1,NL)
      WRITE(3,10068) (H(K1),K1=1,N2)
      WRITE(3,10069) (E(K1),K1=1,N2)
      IERR=1
      RETURN
270   CALL PIVOTA(ND1,NL,R,AC,B1,BC)
      BASE(1,R)=S
      IF(N2.EQ.0) GO TO 277
      DO 275 K1=1,N2
      IF(S.EQ.I2(K1)) GO TO 280
275   CONTINUE
277   BASE(2,R)=0
      GO TO 290
280   BASE(2,R)=1
290   CONTINUE
C
C  FASE 2
C  PI=CB*B1
C
300   DO 310 I=1,NL
      PI(I)=0.
      DO 310 J=1,NL
310   PI(I)=PI(I)+CI(BASE(1,J))*B1(J,I)
C
C  CC(J)=CI(J)-PI*AJ, J NAO PERTENCENTE A BASE,  CC(J)<0
C  ESCOLHA DA VARIAVEL DE ENTRADA X(S)
C
320   DO 350 J=1,NC
      DO 330 K1=1,NL
      IF(J.EQ.BASE(1,K1))    GO TO 350
330   CONTINUE
C
C  J NAO PERTENCE A BASE
C
```

```
      IF(IT.EQ.1.OR.P2.EQ.2) GO TO 335
      IF(P2.EQ.4) GO TO 335
      DO 337 K2=1,NP
      IF(J.EQ.INDP(K2)) GO TO 335
337   CONTINUE
      GO TO 350
335   AUX=0.
      DO 340 I=1,NL
340   AUX=AUX+PI(I)*A(I,J)




      CC=CI(J)-AUX
      IF(CC.LT.-TOL)    GO TO 450
350   CONTINUE
C
C  FIM DA FASE 2
C
      F=0.
      IF(IT.NE.1.AND.P2.NE.2) GO TO 367
C
C  TROCA D SINAL DAS COLUNAS DE B1 E DAS LINHAS DE A QUE FORAM
C  MULTIPLICADAS POR -1
C
      DO 364 J=1,NL
      IF(V(J).EQ.1.) GO TO 364
      DO 360 I=1,NL
360   B1(I,J)=-B1(I,J)
      DO 362 I=1,NC
362   A(J,I)=-A(J,I)
364   CONTINUE
C
367   IF(P2.EQ.2) GO TO 370
      DO 368 I=1,NC
368   X(I)=0.
      GO TO 374
370   DO 372 I=1,NC
372   V(I)=0.
C
C  RECALCULA O VALOR DAS VARIAVEIS CANALIZADAS QUE ESTAO
C  NO LIMITE SUPERIOR
C
374   IF(N2.EQ.0) GO TO 410
      DO 400 J=1,N2
      DO 375 I=1,NL
      IF(I2(J).EQ.BASE(1,I))    GO TO 390
375   CONTINUE
C
C  VARIAVEL CANALIZADA E' NAO BASICA
C
      IF(E(J).NE.-1)    GO TO 400
```

```
C
C  VARIAVEL CANALIZADA NAO BASICA ESTA' NO LIMITE SUPERIOR
C
       JJ=I2(J)
       IF(P2.NE.2) GO TO 380
       V(JJ)=H(J)
       CI(JJ)=-CI(JJ)
       F=F+CI(JJ)*V(JJ)
       GO TO 400
380    X(JJ)=H(J)
       CI(JJ)=-CI(JJ)
       F=F+CI(JJ) * X(JJ)
       GO TO 400
C
C  VARIAVEL CANALIZADA E' BASICA
C
390    IF(E(J).EQ.1)    GO TO 400
       BC(I)=H(J)-BC(I)
       CI(I2(J))=-CI(I2(J))
       E(J)=-E(J)
       DO 395 K1=1,NL
       A(K1,I2(J))=-A(K1,I2(J))
395    B1(I,K1)=-B1(I,K1)
400    CONTINUE
410    IF(P2.EQ.2) GO TO 441
C
C  CALCULA O VALOR DA FUNCAO OBJETIVO (F)
C
       DO 420 I=1,NL
       X(BASE(1,I))=BC(I)
420    F=F+CI(BASE(1,I))*BC(I)
       IF(IMPR.EQ.0) RETURN
       WRITE(3,10010) (CI(I),I=1,NC)
       WRITE(3,10080) F
       WRITE(3,10090)
       WRITE(3,10020) KK,(X(I),I=1,NC)
       WRITE(3,10040) (BASE(1,I),I=1,NL)
       RETURN
C
C  SOLUCAO E FUNCAO OBJETIVO DO PROBLEMA MESTRE
C
441    DO 442 I=1,NL
       V(BASE(1,I))=BC(I)
442    F=F+CI(BASE(1,I))*BC(I)
C
C  TROCA O SINAL DAS COLUNAS DE A PARA A CONTINUACAO DO PROBLEMA MESTRE
C
```

```
      IF(N2.EQ.0) GO TO 447
      DO 445 J=1,N2
      IF(E(J).EQ.1) GO TO 445
      K1=I2(J)
      DO 444 I=1,NL
444   A(I,K1)=-A(I,K1)
445   CONTINUE
447   IF(IMPR.EQ.0) RETURN
      WRITE(3,10010) (CI(I),I=1,NC)
      WRITE(3,10080) F
      WRITE(3,10090)
      WRITE(3,10095) (V(I),I=1,NC)
      WRITE(3,10040) (BASE(1,I),I=1,NL)
      RETURN
C
C  X(S) ENTRA NA BASE
C
450   S=J
      CALL SAIDA(KK,S,NL,NC,CI,A,N2,I2,H,NII,MII,ND1,
     1ND2,R,IERR,B1,E,BASE,BC,AC)
      IF(R.EQ.0)    GO TO 320
      GO TO 300
10010 FORMAT(/,3X,'CI : ',50G)
10020 FORMAT(3X,'X',I3,' : ' ,50G)
10030 FORMAT(/,3X,'BC : ',30G)
10040 FORMAT(/,3X,'BASE :',30I4)
10050 FORMAT(/,3X,'ERRO4 - VARIAVEL CANALIZADA DEVERIA ESTAR NA BASE')
10060 FORMAT(/,3X,'O SUBPROBLEMA ',I3,' NAO TEM SOLUCAO FACTIVEL',
     1/,3X,'W (F. OU A. DO SUBPROBLEMA) =',G,/,3X,'ITERACAO
     2 NUMERO ',I6)
10065 FORMAT(/,3X,'TENTATIVA DE TIRAR VARIAVEL ARTIFICIAL DA BASE')
10066 FORMAT(/,3X,'BASE:')
10067 FORMAT(/,3X,'LINHA ',I3,2X,30I4)
10068 FORMAT(/,3X,'H :',50G)



10069 FORMAT(/,3X,'E :',50I4)
10070 FORMAT(///,1X,'* SUBPROBLEMA NUMERO',I3)
10075 FORMAT(///,1X,'* PROBLEMA MESTRE',/)
10080 FORMAT(/,3X,'VALOR DA FUNCAO OBJETIVO: ',G)
10090 FORMAT(/,3X,'SOLUCAO BASICA OTIMA:',/)
10095 FORMAT(3X,'V :',50G)
10100 FORMAT(/,3X,'MATRIZ DAS RESTRICOES :',/)
10110 FORMAT(50G)
      END
```

```
C
C***********************************************************************
C  ESCOLHE A VARIAVEL DE SAIDA X(R), ATUALIZANDO A BASE
C***********************************************************************
C
      SUBROUTINE SAIDA(KK,S,NL,NC,CI,A,N2,I2,H,NII,MII,ND1,
     1ND2,R,IERR,B1,E,BASE,BC,AC)
      INTEGER S,R,N2,I2(ND2),E(ND2),BASE(2,ND1)
      REAL CI(NII),A(MII,NII),H(ND2),B1(ND1,ND1),BC(ND1),AC(ND1)
C
      IF(S.LE.NC) GO TO 7
      DO 5 I=1,NL
5     AC(I)=B1(I,S-NC)
      GO TO 17
7     DO 10 I=1,NL
      AC(I)=0.
      DO 10 J=1,NL
10    AC(I)=AC(I)+B1(I,J)*A(J,S)
      IF(N2.EQ.0) GO TO 17
      DO 15 K1=1,N2
      IF(S.EQ.I2(K1)) GO TO 20
15    CONTINUE
17    ICANA=0
      AMENO=1.E30
      GO TO 30
C
C  VARIAVEL DE ENTRADA E' CANALIZADA
C
20    ICANA=1
      AMENO=H(K1)
30    R=0
      DO 50 I=1,NL
      IF(BASE(2,I).EQ.0)    GO TO 40
      IF(AC(I).GE.0.)    GO TO 40
C
C  VARIAVEL DE SAIDA E' CANALIZADA
C
      DO 35 J=1,N2
      IF(BASE(1,I).EQ.I2(J)) GO TO 37
35    CONTINUE
      WRITE(3,10040)
      IERR=1
      RETURN
37    RAZAO=(BC(I)-H(J))/AC(I)
      IF(RAZAO.GE.AMENO)    GO TO 50
      AMENO=RAZAO
      R=I
      MARCA=2
      GO TO 50
C
C  VARIAVEL DE SAIDA E' NAO CANALIZADA
```

```
C
40        IF(AC(I).LE.0.)    GO TO 50
          RAZAO=BC(I)/AC(I)
          IF(RAZAO.GE.AMENO)    GO TO 50
          AMENO=RAZAO
          R=I
          MARCA=1
50        CONTINUE
C
          IF(R.NE.0)    GO TO 70
          IF(ICANA.NE.1)    GO TO 100
C
C  NAO HA MUDANCA DE BASE
C
          DO 60 I=1,NL
          BO(I)=BC(I)-H(K1)*AC(I)
60        A(I,S)=-A(I,S)
          E(K1)=-E(K1)
          CI(S)=-CI(S)
          RETURN
C
70        IF(MARCA.EQ.1)    GO TO 90
          DO 75 J=1,N2
          IF(BASE(1,R).EQ.I2(J)) GO TO 77
75        CONTINUE
          WRITE(3,10050)
          IERR=1
          RETURN
77        BC(R)=BC(R)-H(J)
          DO 80 I=1,NL
80        A(I,BASE(1,R))=-A(I,BASE(1,R))
          E(J)=-E(J)
          CI(BASE(1,R))=-CI(BASE(1,R))
90        CALL PIVOTA(ND1,NL,R,AC,B1,BC)
          BASE(1,R)=S
          BASE(2,R)=ICANA
          RETURN
C
100        WRITE(3,10010)
          DO 110 I=1,2
110       WRITE(3,10020) I,(BASE(I,J),J=1,NL)
          WRITE(3,10025) (BC(I),I=1,NL)
          WRITE(3,10030) S,KK
          IERR=1
          RETURN
10010     FORMAT(3X,'ULTIMA BASE :',/)
10020     FORMAT(/,3X,'LINHA ',I3,2X,30I4)
10025     FORMAT(/,3X,'BC :',30G)
10030     FORMAT(3X,'X(',I3,') PODE CRESCER INDEFINIDAMENTE',//,3X,
         1'O SUBPROBLEMA ',I4,' NAO TEM SOLUCAO LIMITADA')
10040     FORMAT(/,3X,'ERRO1 - VARIAVEL CANALIZADA DEVERIA ESTAR NA BASE')
10050     FORMAT(/,3X,'ERRO2 - VARIAVEL CANALIZADA DEVERIA ESTAR NA BASE')
          END
```

```
C
C**********************************************************************
C  FAZ O PIVOTAMENTO EM AC(R), QUE E' O ELEMENTO A(R,S)
C  ATUALIZADO
C**********************************************************************
C
       SUBROUTINE PIVOTA(ND1,M1,R,AC,B1,BC)




       INTEGER R,M1
       REAL AC(ND1),B1(ND1,ND1),BC(ND1)
       PIVO=AC(R)
       IF(PIVO.EQ.1.)    GO TO 20
C
C  DIVIDE A LINHA R PELO PIVO (ERO1)
C
       DO 10 J=1,M1
10     B1(R,J)=B1(R,J)/PIVO
       BC(R)=BC(R)/PIVO
C
C  LINHA I - LINHA R * A(I,S)/PIVO (ERO2)
C
20     DO 40 I=1,M1
       IF(I.EQ.R)    GO TO 40
       DO 30 J=1,M1
30     B1(I,J)=B1(I,J)-B1(R,J)*AC(I)
       BC(I)=BC(I)-BC(R)*AC(I)
40     CONTINUE
       RETURN
       END
```

BIBLIOGRAFIA:

[1] Arenales, M.N., "Programação Linear com Restrições Canalizadas", Tese de Mestrado, Departamento de Matemática Aplicada, UNICAMP, 1979.

[2] Bazaraa, Mokhtar S., "Linear Programming and Network Flows", John Wiley & Sons, 1977.

[3] Dantzig, G.B., "Linear Programming and Extensions", Princeton, New Jersey, 1963.

[4] Hadley, G. (1), "Linear Algebra", Addison-Wesley Publishing Company, Inc., 1969.

[5] Hadley, G. (2), "Linear Programming", Addison - Wesley Publishing Company, Inc., 1969.

[6] Ho, James K. e Loute, E., "An Advanced Implementation of the Dantzig-Wolfe Decomposition Algorithm for Linear Programming", Mathematical Programming, 20, 1981, pp. 303 - 326.

[7] Krekó, Béla, "Linear Programming", American Elsevier Publishing Company, Inc., New York, 1968.

[8] Lasdon, Leon S., "Optmization Theory for Large Systems", The Macmillan Company, London, 1970.

[9] Luenberger, David G., "Introduction to Linear and Nonlinear Programming", Addison - Wesley Publishing Company, 1973.

[10] Murty, Katta G., "Linear and Combinatorial Programming", John Wiley & Sons, 1976.

[11] Rosen, J.B., "Primal Partition Programming for Block Diagonal Matrices", Numerische Mathematik, 6, 1964, pp.250-260.

[12] Sakarovitch, M., "Notes on Linear Programming",Van Nostrand Reinhold Company, 1971.

[13] Simonnard, M., "Programacion Lineal", Paraninfo,Madrid, 1972.

[14] Taha, H.A., "Operations Research: An Introduction", Macmillan, New York, 1971.

[15] Zoutendijk, G., "Mathematical Programming Methods" North-Holland Publishing Company, 1976.