UM MÉTODO DUAL-SIMPLEX PARA PROBLEMAS DE PROGRAMAÇÃO LINEAR COM ESTRUTURA BLOCO-ANGULAR E VARIÁVEIS CANALIZADAS

ÁLVARO JOSÉ PERIOTTO

Orientador
Prof. Dr. RAUL VINHAS RIBEIRO

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

À ELIDA e ANA CAROLINA

À minha querida mãe MARIA CREUSA

AGRADECIMENTOS

Durante o desenvolvimento deste trabalho, de uma forma ou de outra, muitas foram as contribuições que recebi. Portanto, se há algum mérito nesta tese, com justiça deve ser dividido entre as pessoas para as quais dirijo a minha admiração e o meu carinho.

Ao Raul, meu orientador, que sempre, pronta e pacientemente, me dispensou seu tempo, sua atenção e suas idéias.

À Elida, minha companheira e maior incentivadora, que com o coração soube me compreender e aliviar o peso nos momentos mais difíceis.

À Ana Carolina, que pequenina, nos trouxe grandes alegrias e esperanças, contribuições que reanimaram as possibilidades de sucesso.

Aos meus familiares e amigos que por seus constantes estímulos e orações nos conduziram às consecutivas vitórias.

Ao Prof. Taube, nosso mestre e aos companheiros da UNICAMP e UEM que sempre demonstraram interesse pelo meu trabalho.

A Deus que semeou em mim as dúvidas e fez germinar a luz do raciocínio.

SUMÁRIO

No primeiro capítulo fazemos a apresentação do problema que deu origem a este trabalho, ou seja, o problema de programação da produção com horizonte de quatro meses.

Este problema trata do planejamento da produção de uma indústria, procurando entre as várias alternativas aquela que minimize os custos de inventário e atenda as vendas previstas, com base nos tempos de produção dos vários itens, respeitando as limitações de horas normais e horas extras de produção.

No segundo capítulo apresentamos os métodos de resolução de problemas de programação linear apropriados à estrutura bloco-angular que servem como base para o método de resolução a ser aplicado ao problema proposto. Deve-se ressaltar que neste capítulo os métodos não tratam com variáveis "canalizadas".

Os dois capítulos subsequentes representam a parte teórica deste trabalho, desenvolvida através de adaptações dos métodos de resolução às necessidades da situação exposta: no terceiro capítulo esquecemos temporariamente a estrutura bloco-angular e mostramos um método dual simplex para problemas com variáveis "canalizadas" e no quarto capítulo mostramos um método para a forma dual do problema bloco-angular com variáveis "canalizadas".

No quinto capítulo apresentamos um programa de computador que baseia-se no método desenvolvido e explora as peculiaridades da programação de produção com horizonte de quatro meses.

Finalmente, no sexto capítulo apresentamos as conclusões e algumas sugestões de continuidade deste trabalho.

APRESENTAÇÃO

Na época de definição de um trabalho de tese, procuramos por um tema que se caracterizasse como "prático", para que pudessemos aplicar, de forma direta, a teoria assimilada durante nosso curso de pós-graduação.

Foi-nos sugerido, então, um problema de programação de produção que apresentava, na estrutura de sua formulação, as peculiaridades da forma bloco-angular e características de esparsidade.

Desta forma, tomamos como objetivo o desenvolvimento de um programa de computador que explorasse tais particularidades.

Porém ao fazermos o levantamento bibliográfico, verificamos que os métodos de resolução considerados mais apropriados ao problema não previam o caso das variáveis serem do tipo "canalizadas", como as que se apresentavam na situação em estudo.

Constatamos, também, que a bibliografia se omitia, ainda mais, quando se tratava da forma dual para problemas que apresentassem variáveis canalizadas.

Assim, este trabalho tomou definitivamente características teóricas quando objetivamos o desenvolvimento de "Um Método Dual Simplex para Problemas de Programação Linear com Estrutura Bloco-Angular e Variáveis Canalizadas".

Contudo, não abdicamos de nosso propósito inicial e apresentamos também um programa de computador para a resolução do problema de programação de produção, baseado no método aqui desenvolvido.

ÍNDICE

CAPÍTULO I

O PROBLEMA DE PROGRAMAÇÃO DA PRODUÇÃO COM HORIZONTE DE QUATRO MESES

1.1 INTRODUÇÃO

O problema que aqui se apresenta, e que deu origem ao nosso estudo, surgiu a partir de um trabalho desenvolvido pelo Prof. Miguel Taube Netto, no qual apresentava um "Modelo Para Programação da Produção com Horizonte de Quatro Meses".

Neste capítulo, além de enunciarmos o referido problema, apresentaremos sua formulação matemática e matricial, objetivando uma melhor visualização da estrutura bloco-angular, que despertou-nos, por tal peculiaridade, para o desenvolvimento deste trabalho.

1.2 APRESENTAÇÃO DO PROBLEMA

Uma indústria é responsável pela produção de n tipos de peças. No setor de produção tem-se m máquinas consideradas "potencialmente gargalo", ou seja, são as máquinas que "seguram" o sistema produtivo, pois a maioria das peças tem que, necessariamente, passar por elas.

O objetivo é obter a programação da produção para quatro meses, de forma tal que as vendas previstas para estes meses sejam atendidas e os custos de inventário sejam minimizados.

Deve ser levada em conta a possibilidade de produzir num dado mês não somente as vendas previstas para este mês, como também parte das vendas previstas para os meses subsequentes.

Trata-se, portanto, de um problema onde as interações entre os quatro meses do horizonte de planejamento devem ser representadas.

Entre as várias alternativas de produção, busca-se aquela que minimize os custos de inventário, respeitando uma exigência de, preferencialmente, utilizar as horas extras dos sábados, das noites dos dias normais e finalmente as horas extras dos domingos, nesta ordem de prioridade.

As restrições que devem ser respeitadas referem-se às quantidades a serem produzidas e às limitações de utilização de horas extras.

1.3 FORMULAÇÃO MATEMÁTICA DO PROBLEMA

1.3.1 DEFINIÇÃO DAS VARIÁVEIS

X_{jk} = quantidade de peças do tipo j a serem produzidas durante o mes k.

T_j = quantidade total requerida de peças do tipo j durante os quatro meses.

α_{jk} = quantidade mínima de peças do tipo j que devem ser produzidas durante o mês k.

E_{jk} = inventário das peças do tipo j, transferido do mês k-1 para o mês k.

C_j = custo unitário da peça do tipo j.

θ = fator de desconto: o valor presente de uma quantida de de Q realizada no mês k, é dado pelo produto $\theta^{k-1}Q$.

A_{ij} = tempo (em horas) necessário para uma peça j ser processada pela máquina i.

H_{ik} = quantidade de horas normais disponíveis na máquina i durante o mês k, já considerando que a utilização prevista é de 75%, isto é, $H_{ik} = 0,75 \times d_k \times h_k$, onde d_k é o número de dias úteis do mês k e h_k o número médio de horas úteis por dia.

U_{ik}, V_{ik}, W_{ik} = quantidade de horas extras nos sábados, nas noites dos dias normais e nos domingos, respectivamente, necessárias durante o mês k para produção na máquina i.

LU_{ik}, LV_{ik}, LW_{ik} = limites de horas extras disponíveis nos sábados, nas noites dos dias normais e nos domingos, respectivamente, no mês k para produção na máquina i.

1.3.2 EQUACIONAMENTO DO PROBLEMA

A "função objetivo" a ser minimizada é:

$$F = \sum_{j=1}^{n} \sum_{k=2}^{4} \theta^{k-1} C_j E_{jk} + \sum_{i=1}^{m} \sum_{k=1}^{4} (1{,}25U_{ik} + 2{,}0V_{ik} + 2{,}5W_{ik}) \qquad (1)$$

Na primeira dupla-somatória, o termo $\theta^{k-1}C_jE_{jk}$, representa o custo de transferir o inventário da peça j do mês k-1 para o mês k.

A segunda dupla-somatória representa os "custos" totais da utilização de horas extras nas máquinas consideradas "gargalo do sistema" durante o horizonte de quatro meses.

Na realidade, no segundo duplo somatório os "custos" não têm o dimensionamento monetário, e com maior rigor não poderiam ser somados com os custos de inventário. Entretanto, como os valores de C_j são superiores a 1000, e as variáveis U_{ik}, V_{ik} e W_{ik} assumirão valores numericamente inferiores a E_{jk}, a primeira dupla-somatória será preponderante.

Com isto, espera-se que a otimização apresente juntamente com a solução de menor custo de inventário, a alternativa que atenda também as exigências de utilização de horas extras.

A função objetivo apresentada deve sujeitar-se às seguintes restrições:

$$\sum_{k=1}^{4} X_{jk} = T_j \quad ; \quad j=1,2,\ldots,n \qquad (2)$$

$$X_{j1} \geq \alpha_{j1} \quad ; \quad j=1,2,\ldots,n \tag{3}$$

$$E_{j2} + X_{j2} \geq \alpha_{j2} \quad ; \quad j=1,2,\ldots,n \tag{4}$$

$$E_{j3} + X_{j3} \geq \alpha_{j3} \quad ; \quad j=1,2,\ldots,n \tag{5}$$

$$\sum_{j=1}^{N} A_{ij} X_{jk} \leq H_{ik} + U_{ik} + V_{ik} + W_{ik} \quad ; \quad \begin{array}{l} i=1,2,\ldots,m \\ k=1,2,3,4 \end{array} \tag{6}$$

$$U_{ik} \leq LU_{ik} \quad ; \quad \begin{array}{l} i=1,2,\ldots,m \\ k=1,2,3,4 \end{array} \tag{7}$$

$$V_{ik} \leq LV_{ik} \quad ; \quad \begin{array}{l} i=1,2,\ldots,m \\ k=1,2,3,4 \end{array} \tag{8}$$

$$W_{ik} \leq LW_{ik} \quad ; \quad \begin{array}{l} i=1,2,\ldots,m \\ k=1,2,3,4 \end{array} \tag{9}$$

$$X_{jk},\ E_{jk},\ U_{ik},\ V_{ik},\ W_{ik} \geq 0 \quad ; \quad \begin{array}{l} i=1,2,\ldots,m \\ j=1,2,\ldots,n \\ k=1,2,3,4 \end{array} \tag{10}$$

A restrição (2) refere-se à exigência da quantidade total a ser produzida durante o horizonte de planejamento,ser igual a T_j para cada peça do tipo j.

As restrições (3), (4) e (5) referem-se às necessidades mínimas de produção em cada mês para cada peça do tipo j.

A equação (6) representa a restrição de que, matematicamente, as quantidades de horas a serem utilizadas em uma máquina i, devam ser no máximo iguais às horas disponíveis nesta máquina (horas normais + horas extras).

As restrições (7), (8) e (9) limitam a utilização das horas extras aos sábados, noites e domingos respectivamente.

Trataremos como restrições propriamente ditas, apenas as restrições de números (2), (4), (5) e (6), pois as demais são restrições de canalização.

Levando-se em conta que:

$$E_{j1} = 0 \tag{11}$$

$$E_{j,k+1} = E_{jk} + X_{jk} - \alpha_{jk} \quad ; \quad k = 1,\ 2,\ 3 \tag{12}$$

Para maior simplicidade, podemos colocar as variáveis X_{jk} em termos das variáveis E_{jk}. Nestas condições, a função objetivo assume a seguinte forma:

$$F = \sum_{j=1}^{n} \sum_{k=1}^{3} \zeta_k C_j X_{jk} + \sum_{i=1}^{m} \sum_{k=1}^{4} (1{,}25U_{ik} + 2{,}0V_{ik} + 2{,}5W_{ik}) - \sum_{j=1}^{n} \sum_{k=1}^{4} \zeta_k C_j \alpha_{jk} \tag{13}$$

onde $$\zeta_1 = \theta^1 + \theta^2 + \theta^3 \tag{14}$$

$$\zeta_2 = \theta^2 + \theta^3 \tag{15}$$

$$\zeta_3 = \theta^3 \tag{16}$$

Como a terceira dupla somatória da nova função objetivo é um termo constante, não precisamos considerá-la.

Executando a mesma operação para as restrições (4) e (5), ficamos com:

$$X_{j1} + X_{j2} + X_{j3} \geqslant \alpha'_j \quad ; \quad j = 1,2,\ldots,m \tag{17}$$

$$X_{j1} + X_{j2} \geqslant \alpha_j'' \quad ; \quad j = 1,2,\ldots,m \tag{18}$$

onde $\alpha_j' = \alpha_{j1} + \alpha_{j2} + \alpha_{j3}; \quad j = 1,2,\ldots,m$ (19)

$$\alpha_j'' = \alpha_{j1} + \alpha_{j2} \quad ; \quad j = 1,2,\ldots,m \tag{20}$$

Com estas simplificações, o modelo matemático fica reduzido à função objetivo (13) sujeita às restrições (2), (6), (17) e (18).

1.3.3 A FORMA MATRICIAL DO MODELO MATEMÁTICO

Consideremos os seguintes símbolos:

X^k = vetor-coluna de n elementos X_{jk} ; $k = 1,2,3,4$.

T = vetor-coluna de n elementos T_j.

α' = vetor-coluna de n elementos α_j'.

α'' = vetor-coluna de n elementos α_j''.

C = vetor-linha de n elementos C_j.

A = matriz de m x n elementos A_{ij}.

H^k = vetor-coluna de m elementos H_{ik} ; $k = 1,2,3,4$.

U^k = vetor-coluna de m elementos U_{ik} ; $k = 1,2,3,4$.

V^k = vetor-coluna de m elementos V_{ik} ; $k = 1,2,3,4$.

W^k = vetor-coluna de m elementos W_{ik} ; $k = 1,2,3,4$.

Consideremos ainda:

Y^k = vetor-coluna de m elementos Y_{ik}, que representam as horas ociosas para cada máquina i em cada período k; k = 1,2,3,4 (variáveis de folga para a restrição (6)).

Z' = vetor-coluna de n elementos Z'_j, que representam as quantidades produzidas, da peça j, além do mínimo pedido α'_j (variáveis de falta para a restrição (17)).

Z" = vetor-coluna de n elementos Z''_j, que representam as quantidades produzidas, da peça j, além do mínimo pedido α''_j (variáveis de falta para a restrição (18)).

Desta forma, o modelo simplificado proposto anteriormente, pode ser escrito matricialmente da seguinte forma:

$$\text{Minimizar } F = \sum_{k=1}^{3} \zeta_k C X^k + \sum_{k=1}^{4} 1{,}25U^k + 2{,}0V^k + 2{,}5W^k \qquad (21)$$

Sujeito às restrições:

$$X^1 + X^2 + X^3 + X^4 = T \qquad (22)$$

$$X^1 + X^2 + X^3 - Z' = \alpha' \qquad (23)$$

$$X^1 + X^2 - Z'' = \alpha'' \qquad (24)$$

$$AX^1 - U^1 - V^1 - W^1 + Y^1 = H^1 \qquad (25)$$

$$AX^2 - U^2 - V^2 - W^2 + Y^2 = H^2 \qquad (26)$$

$$AX^3 - U^3 - V^3 - W^3 + Y^3 = H^3 \qquad (27)$$

$$AX^4 - U^4 - V^4 - W^4 + Y^4 = H^4 \qquad (28)$$

1.4 ALGUMAS CONSIDERAÇÕES SOBRE ESTE PROBLEMA

Através da forma em que colocamos o modelo matemático, podemos observar algumas peculiaridades deste problema.

Inicialmente verificamos tratar-se de um problema de porte considerável, já que a quantidade de variáveis e restrições podem crescer muito, conforme os valores de m e n.

Observamos também que apesar do dimensionamento do problema, verifica-se a ocorrência de uma grande quantidade de elementos nulos na matriz de restrições.

Apenas estas observações são suficientes para concluirmos que um "simplex" usual apesar de "resolver" o problema, não o faz de maneira eficiente por não considerar tais particularidades.

Se observarmos mais atentamente a disposição dos elementos não nulos na matriz de restrições, notamos a existência de alguns blocos dispostos estrategicamente.

Quatro blocos independentes entre si formam uma diagonal na estrutura apresentada (restrições de (25) a (28)) e um bloco maior localizado na parte superior da estrutura "amarra" as várias incógnitas que aparecem nos vários blocos (restrições de acoplamento: (22), (23) e (24)).

Esta estrutura é conhecida por "Estrutura Bloco-Angular".

Em vista das observações anteriores e conhecendo-se esta estrutura particular, concluimos que um "simplex" voltado às peculiaridades do problema e desta estrutura apresentará vantagens computacionais em relação a um "simplex" usual em termos de melhor aproveitamento de memória e tempo computacionais.

CAPÍTULO II

O PROBLEMA DE PROGRAMAÇÃO LINEAR COM ESTRUTURA BLOCO-ANGULAR

2.1 INTRODUÇÃO

Entre os vários processos utilizados na solução de sistemas de grande porte, muitos surgiram em vista da existência de estruturas especiais.

Na realidade, para o caso de sistemas lineares, estes métodos são especializações do "Método Simplex Revisado", na busca de redução de tempo e memória do computador, baseados nas propriedades particulares da matriz básica do problema a ser resolvido.

Os chamados métodos de "Upper Bound", "Generalized Upper Bound" e particularmente suas extensões para estruturas angulares também se incluem na classe dos métodos acima descrito.

Os métodos de "Upper Bound", utilizados na resolução de sistemas envolvendo restrições de limite superior das variáveis, foram introduzidos por George B. Dantzig na Rand Corporation em Outubro de 1954. Foram publicados em um memorando de pesquisa sob o título "Notes on Linear Programming: Part VIII - Upper Bounds".

A evolução dos estudos nesta área seguiu-se com a publicação "Generalized Upper Bounded Techniques for Linear Programming I, II" por G.B. Dantzig e R.M. Von Slyke em Agosto de 1964 e Feve reiro de 1965, pela Universidade da California em seu Centro de Pesquisa Operacional.

Trata-se de uma especialização do Método Simplex para resolver problemas com a seguinte estrutura:

$$
\begin{array}{l}
\text{m linhas} \left\{ A^{o}X_{o}+\dots+A^{n_o}X_{n_o}+A^{n_o+1}X_{n_o+1}+\dots+A^{n_1}X_{n_1}+\dots+A^{n_{q-1}+1}X_{n_{q-1}+1}+\dots+A^{n_q}X_{n_q}=b \right. \\
\text{p linhas} \left\{ \begin{array}{lr}
X_{n_o+1}+\dots+X_{n_1} & =1 \\
\quad \ddots & \\
\qquad X_{n_{q-1}+1}+\dots+X_{n_q} & =1
\end{array} \right.
\end{array}
$$

$$X_j \geq 0$$

A grande vantagem deste método, é que mantém uma base de trabalho de dimensão m x m, e isto é substancial quando p é bem maior que m.

Uma generalização natural deste método refere-se a problemas com Estrutura Bloco Angular. Em 1965 surge então a publicação "An Extension of Generalized Upper Bounding Techniques for Linear Programming", que apresenta um método voltado a este tipo de problema (conhecido por G.G.U.B.).

Este e outros métodos, bem como algumas variações estão reunidos na tese de doutorado "Estudos em Programação Linear", do Prof. Raul Vinhas Ribeiro (FEC-UNICAMP, 1980), na qual se baseia este capítulo.

2.2 O PROBLEMA BLOCO-ANGULAR

Através das discussões anteriores temos uma idéia geral da estrutura de um Problema Bloco-Angular, porém, para o estudo que desenvolveremos, vamos fixar a seguinte notação:

$$\text{Minimizar} \quad Z = C_1X_1 + C_2X_2 + \ldots + C_qX_q \qquad (1)$$

Sujeita às restrições:

$$A_o^1X_1 + A_o^2X_2 + \ldots + A_o^qX_q = b_o \qquad (2)$$

$$\left.\begin{array}{llllr} A_1X_1 & & & & = b_1 \\ & A_2X_2 & & & = b_2 \\ & & \ddots & & \vdots \\ & & & A_qX_q & = b_q \end{array}\right\} \qquad (3)$$

$$X_1, X_2, \ldots, X_q \geq 0$$

A_o^i e A_i são matrizes com dimensões $m_o \times n_i$ e $m_i \times n_i$ respectivamente.

C_i é um vetor-linha com n_i elementos.

X_i, b_o e b_i são vetores-coluna com n_i, m_o e m_i elementos respectivamente.

Assumiremos que não há redundância de restrições, ou seja, o sistema (2) - (3) tem "rank completo".

Segue-se então que cada matriz A_i tem "rank" igual a m_i.

O teorema que em seguida apresentamos, suporte para o método que estudaremos, caracteriza uma base para o sistema (2)-(3).

TEOREMA 1: Uma base para o sistema (2) - (3) pode ser particionada da seguinte forma:

$$A^B = \left[\begin{array}{c} A_o^B \\ \hdashline A^B \end{array}\right] = \left[\begin{array}{c:c} A_o^L & A_o^I \\ \hdashline A^L & A^I \end{array}\right] = \left[\begin{array}{c:cccc} A_o^L & A_o^{I_1} & A_o^{I_2} & \cdots & A_o^{I_q} \\ \hdashline A^L & A^{I_1} & & & \\ & & A^{I_2} & & \\ & & & \ddots & \\ & & & & A^{I_q} \end{array}\right] \tag{4}$$

Com $B = L \oplus I = L \oplus (I_1 \oplus I_2 \oplus \ldots \oplus I_q)$, onde cada A^{I_i} tem dimensão $m_i \times m_i$ e é não singular.

PROVA: B é um conjunto ordenado de $m_o + \sum_{i=1}^{q} m_i$ elementos, que correspondem aos índices das colunas linearmente independentes do sistema (2) - (3). As primeiras m_o componentes pertencem ao conjunto L, as m_1 componentes seguintes pertencem ao conjunto I_1 e assim sucessivamente.

Consideremos a submatriz, retirada do sistema (2) - (3):

$$\left[\begin{array}{c} A_o^i \\ \hdashline o \\ \hdashline A_i \\ \hdashline o \end{array}\right] \begin{array}{l} \} \ m_o \text{ linhas} \\ \\ \} \text{ partição } i \\ \\ \end{array}$$

Devemos observar que qualquer outra submatriz desta forma, apresentará componentes nulas na partição i. A partir desta submatriz podemos obter uma outra, constituída pelas colunas que pertencem à base A^B:

$$\left[\begin{array}{c} A_o^{B_i} \\ \hdashline o \\ \hdashline A^{B_i} \\ \hdashline o \end{array}\right] \begin{array}{l} \left.\right\} m_o \text{ linhas} \\ \\ \left.\right\} \text{partição } i \\ \\ \end{array}$$

Se o rank A^{B_i} fosse menor que m_i, teríamos algumas linhas de A^B linearmente dependentes. Contudo, isto não é possível, já que inicialmente assumimos a não redundância de restrições.

Segue-se então, que A^{B_i} tem rank m_i e constitui uma matriz não singular ∎

Trataremos por "colunas chaves" aquelas cujos índices pertençam ao conjunto I e "colunas não chaves" aquelas cujos índices pertençam ao conjunto L.

O conjunto de índices relativos às colunas não básicas será denotado por N e a submatriz formada por estas colunas será identificada através da notação:

$$A^N = \left[\begin{array}{c} A_o^N \\ \hdashline A^N \end{array}\right]$$

Com base nos resultados do Teorema 1, o problema bloco-angular pode ser reescrito sob a seguinte forma:

Sistema de Restrições Globais:

$$\left[\begin{array}{c:c:c} C^L & C^I & C^N \\ \hdashline A_o^L & A_o^I & A_o^N \end{array}\right] \left[\begin{array}{c} X^L \\ \hdashline X^I \\ \hdashline X^N \end{array}\right] = \left[\begin{array}{c} Z \\ \hdashline b_o \end{array}\right] \tag{5}$$

Sistema de Restrições Locais:

$$\left[\begin{array}{c:c:c} A^L & A^I & A^N \end{array}\right] \left[\begin{array}{c} X^L \\ \hdashline X^I \\ \hdashline X^N \end{array}\right] = \left[\begin{array}{c} b \end{array}\right] \tag{6}$$

Por conveniência colocamos a função objetivo no primeiro sistema.

Como a matriz A^I é não singular podemos pré-multiplicar (6) pela sua inversa, obtendo:

$$\left[\begin{array}{c:c:c} (A^I)^{-1} A^L & (A^I)^{-1} A^I & (A^I)^{-1} A^N \end{array}\right] \left[\begin{array}{c} X^L \\ \hdashline X^I \\ \hdashline X^N \end{array}\right] = \left[(A^I)^{-1} b\right] \tag{7}$$

Esta operação é equivalente a efetuar um pivoteamento interno no sistema de restrições locais para colocá-lo em sua forma canônica.

A operação de pivoteamento interno será denotada por um circunflexo (chapéu). A equação (7), sob esta notação, será tratada por R.L.C. (Sistema de Restrições Locais sob forma Canônica):

$$\left[\, \hat{A}^L \;\vdots\; E \;\vdots\; \hat{A}^N \,\right] \begin{bmatrix} X^L \\ \hdashline X^I \\ \hdashline X^N \end{bmatrix} = \left[\, \hat{b} \,\right] \qquad \text{(R.L.C.)}$$

A matriz identidade é denotada por E.

A partir de R.L.C. obtemos X^I em função de X^L e X^N. Substituindo este valor em (5), ficamos com:

$$\left[\begin{array}{c|c|c} C^L - C^I \hat{A}^L & 0 & C^N - C^I \hat{A}^N \\ \hdashline A_o^L - A_o^I \hat{A}^L & 0 & A_o^N - A_o^I \hat{A}^N \end{array}\right] \begin{bmatrix} X^L \\ \hdashline X^I \\ \hdashline X^N \end{bmatrix} = \begin{bmatrix} Z - C^I \hat{b} \\ \hdashline b_o - A_o^I \hat{b} \end{bmatrix} \qquad (8)$$

Esta operação, chamada de pivoteamento externo (denotada por um til), é equivalente a estender as operações de pivoteamento, que permitiram a obtenção de R.L.C., ao sistema global.

Trataremos por P.R. (Problema Reduzido) ao sistema apresentado por (8) sob a nova notação:

$$\left[\begin{array}{c|c|c} \tilde{C}^L & o & \tilde{C}^N \\ \hdashline \tilde{A}_o^L & o & \tilde{A}_o^N \end{array}\right] \begin{bmatrix} X^L \\ \hdashline X^I \\ \hdashline X^N \end{bmatrix} = \begin{bmatrix} \tilde{Z} \\ \hdashline \tilde{b}_o \end{bmatrix} \qquad \text{(P.R.)}$$

Após estas transformações, a matriz básica A^B passa a ter a seguinte estrutura:

$$A^B = \left[\begin{array}{c:c} \tilde{A}_o^L & o \\ \hdashline \hat{A}^L & E \end{array}\right] \qquad (9)$$

É claro que as operações de pivoteamento interno e externo não alteraram o rank de $\mathcal{A}^B$, que é igual a $m_o + \sum_{i=1}^{q} m_i$.

Como a identidade E é de ordem $\sum_{i=1}^{q} m_i$, temos que rank $(\tilde{A}_o^L) = m_o$.

Segue-se então, que $\tilde{A}_o^L$ é uma matriz não singular e será referenciada daqui para frente como "Base de Trabalho".

Fazendo $X^N = 0$ em R.L.C. e P.R., teremos caracterizada uma solução básica correspondente a base $\mathcal{A}^B$:

$$X^L = (\tilde{A}_o^L)^{-1} \tilde{b}_o \tag{10}$$

$$X^I = \hat{b} - \hat{A}^L X^L \tag{11}$$

Os custos relativos referentes a esta solução podem ser obtidos com o auxílio de um vetor multiplicador P tal que

$$P \mathcal{A}^B = C^B \tag{12}$$

Se $P = [p_o \,\vdots\, p]$, o sistema definido em (12) pode ser decomposto em:

$$p_o A_o^L + p A^L = C^L \tag{13}$$

$$p_o A_o^I + p A^I = C^I$$

A solução do sistema (13) é dada por:

$$p_o = \tilde{C}^L (\tilde{A}_o^L)^{-1} \tag{14}$$

$$p = (C^I - p_o A_o^I)(A^I)^{-1} \tag{15}$$

O custo relativo para as variáveis não básicas, denotado por $\bar{C}^N$, é definido como sendo

$$\bar{C}^N = C^N - C^B = C^N - pA^B \tag{16}$$

substituindo (14) e (15) em (16), obtemos

$$\bar{C}^N = \tilde{C}^N - p_o \tilde{A}_o^N \tag{17}$$

Com as definições acima, de solução básica e custo relativo para as variáveis não básicas, temos esquematizado um simplex revisado explorando as peculiaridades da estrutura. Resta-nos estudar uma forma eficiente para a atualização da matriz básica, cuja inversa está calculada no apêndice.

2.3 O MÉTODO G.G.U.B. PARA A FORMA PRIMAL DO PROBLEMA BLOCO ANGULAR

O método que aqui apresentamos coloca em uma sequência lógica os cálculos descritos na seção anterior e introduz sua especialização na atualização da base, com o cuidado de manter a estrutura bloco-angular da base a cada iteração.

Para a inicialização do método, devemos ter em mãos uma base factível ordenada de acordo com (4) e as inversas da base de trabalho e das matrizes A^{Ii}; $i = 1,2,\ldots,q$. Estas matrizes poderão ser obtidas através de uma Fase I.

Em seguida, a cada iteração, devemos observar os seguintes passos:

Passo 1: Calcular os custos relativos para as variáveis não básicas de acordo com (17).

Se $\bar{C}^N \geq 0$ então a base corrente é ótima e os valores das variáveis básicas podem ser obtidos através de (10) e (11).

Caso contrário devemos executar o passo seguinte.

Passo 2: Escolher uma coluna de A^N, suponhamos $(A^N)^s$, com $s \in N$, tal que $(\bar{C}^N)^s < 0$. Esta coluna é candidata a entrar na base.

Passo 3: Escolher uma coluna de A^B, suponhamos $(A^B)^\rho$ com ρ ocupando a posição r em B, para deixar a base.

O índice r é determinado pelo coeficiente

$$\theta_r = \frac{(\bar{b})_r}{(\bar{A}^N)^s_r} = \underset{j/(\bar{A}^N)^s_j > 0}{\text{Mínimo}} \left\{ \frac{(\bar{b})_j}{(\bar{A}^N)^s_j} \right\} \qquad (18)$$

onde $\bar{b}$ e $(\bar{A}^N)^s$ representam respectivamente o vetor $\begin{bmatrix} b_o \\ \hdashline b \end{bmatrix}$ e a coluna $(A^N)^s$ escritos em termos da base corrente:

$$(\bar{A}^N)^s = \begin{bmatrix} (\bar{A}^N_o)^s \\ \hdashline (\bar{A}^N)^s \end{bmatrix} = (A^B)^{-1} \begin{bmatrix} (A^N_o)^s \\ \hdashline (A^N)^s \end{bmatrix} \overset{*}{=} \begin{bmatrix} (\tilde{A}^L_o)^{-1} \ (\tilde{A}^N_o)^s \\ \hdashline (\hat{A}^N)^s - \hat{A}^L (\tilde{A}^L_o)^{-1} (\tilde{A}^N_o)^s \end{bmatrix} \qquad (19)$$

* vide apêndice.

$$\bar{\bar{b}} = \begin{bmatrix} \bar{b}_o \\ \hdashline \bar{b} \end{bmatrix} = (A^B)^{-1} \begin{bmatrix} b_o \\ \hdashline b \end{bmatrix} \overset{*}{=} \begin{bmatrix} (\tilde{A}_o^L)^{-1} \tilde{b}_o \\ \hdashline \tilde{b} - \tilde{A}^L(\tilde{A}_o^L)^{-1} \tilde{b}_o \end{bmatrix} \tag{20}$$

Caso $(\bar{\bar{A}}^N)^s \leq 0$ devemos parar com os cálculos, pois o problema não tem solução finita.

Passo 4: Atualização da base.

Os passos descritos até aqui são os passos clássicos do simplex, a não ser o fato de não termos utilizado $(A^B)^{-1}$ na forma explícita, pois tudo foi feito através de $(\tilde{A}_o^L)^{-1}$ e $(A^I)^{-1}$. Neste passo, porém, introduziremos a especia - lização do método.

Observando-se o passo anterior verificamos que $(A^B)^\rho$, com ρ ocupando a posição r em B, pode ser uma coluna cha ve ou não. A partir disto, a atualização da base será feita de conformidade com os seguintes casos:

Caso 1: A coluna a deixar a base é não chave ($r \leq m_o$).

Neste caso a atualização da inversa da base é feita pela matriz de pivoteamento MP(r,s):

$$(A^{B'})^{-1} = MP(r,s)\ (A^B)^{-1} \tag{21}$$

* vide apêndice

$$\text{onde } MP(r,s) = \left[\begin{array}{c|c} \alpha & 0 \\ \hline \beta & E \end{array}\right] = \left[\begin{array}{ccccc|c} 1 & \cdots & \alpha_1 & & & \\ & \ddots & \vdots & & & \\ & & \alpha_r & & & \\ & & \vdots & \ddots & & 0 \\ & & \alpha_{m_o} & \cdots & 1 & \\ \hline & & \beta_1 & & & \\ & & \vdots & & & E \\ & & \beta_m & & & \end{array}\right] \qquad (22)$$

$$\alpha_j = \begin{cases} -(\bar{A}^N_o)^s_j \;/\; (\bar{A}^N_o)^s_r & ;\; j = 1,\ldots,m_o;\; j \neq r \\ 1/\;(\bar{A}^N_o)^s_r & ;\; j = r \end{cases} \qquad (23)$$

$$\beta_j = -(\bar{A}^N_o)^s_j \;/\; (\bar{A}^N_o)^s_r \quad ;\; j = 1,\ldots,m \;;\quad m = \sum_{i=1}^{q} m_i \qquad (24)$$

Através de (21) e conhecendo-se a forma da inversa da matriz básica (apêndice) deduzimos que:

$$(\tilde{A}^{L'}_o)^{-1} = \alpha\,(\tilde{A}^{L}_o)^{-1} \qquad (25)$$

Assim, a atualização da base de trabalho pode ser feita como uma simples operação de pivoteamento em P.R.
A matriz $(A^I)^{-1}$ não necessita de atualização.

Caso 2: A coluna que deve deixar base é chave ($r = m_o + r'$).
Temos então dois subcasos a estudar:

Subcaso 2.1: A coluna candidata a entrar na base e a coluna que deve deixar a base pertencem a sub-sistemas distintos.

Suponhamos, por exemplo, que a coluna $(A^B)^\rho$,com $\rho \in I_h$ deva deixar a base para que $(A^N)^s$, com $s \notin N_h$, entre na base.

Quando a coluna $(A^B)^\rho$ deixar a base, o "rank" da matriz A^{I_h}, agora sem a coluna $(A^B)^\rho$, será m_h-1.

Assim para que continuemos a ter uma base, deve existir uma coluna $(A^B)^\tau$, com $\tau \in L_h$, linearmente independente das colunas restantes de A^{I_h}, que possa "substituir" a coluna que deixou o conjunto I_h.

Vamos supor que a coluna $(A^B)^\tau$ seja tal que τ ocupe a posição t no conjunto L.

Para que a estrutura bloco-angular não seja destruída, executamos primeiramente a troca entre $(A^B)^\tau$ e $(A^B)^\rho$.

Devemos notar que esta troca altera a ordenação de B porém a estrutura de A^B é mantida.

Após esta troca basta aplicarmos o caso anterior, pois a coluna $(A^B)^\rho$, que deve deixar a base, passou a ser uma coluna não chave.

A permutação das colunas $(A^B)^\tau$ e $(A^B)^\rho$ é equivalente a troca de posições das linhas t $(\leq m_o)$ e r $(=m_o + r')$ em $(A^B)^{-1}$.

Após esta troca em $(A^B)^{-1}$ (vide apêndice), a nova base de trabalho poderá ser obtida através da seguinte operação:

$$(\tilde{A}_o^{L'})^{-1} = \begin{array}{l} \\ \\ \\ \\ \text{linha } t \\ \\ \\ \end{array}\!\!\begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ \hline & & -(\hat{A}^L)_{r'} & & \\ \hline & & & 1 & \\ & & & & \ddots \\ & & & & 1 \end{bmatrix} \cdot (\tilde{A}_o^L)^{-1} \qquad (26)$$

onde $(\hat{A}^L)_{r'}$ é a linha r' de $\hat{A}^L$.

É importante lembrar que a t-ésima componente de $(\hat{A}^L)_{r'}$, ou seja, $(\hat{A}^L)^t_{r'}$, é não nula, pois trata-se da condição essencial para haver a permutação descrita por este subcaso.

A atualização de $(A^I)^{-1}$ é feita a nível de A^{I_h}:

$$(A^{I_h'})^{-1} = MP\,(r',t)\ .\ (A^{I_h})^{-1}$$

onde $MP(r',t) = \begin{bmatrix} 1 & \cdots & \alpha_{j+1} & & \\ & \ddots & \vdots & & \\ & & \alpha_{j+r''} & & \\ & & \vdots & \ddots & \\ & & \alpha_{j+m_h} & \cdots & 1 \end{bmatrix}$

$$\alpha_{j+i} = \begin{cases} -(\hat{A}^L)^t_{j+i} \ /\ (\hat{A}^L)^t_{j+r''} & ;\ i = 1,\dots,m_h;\ i \neq r'' \\ 1\ /\ (\hat{A}^L)^t_{j+r''} & ;\ i = r'' \end{cases} \qquad (28)$$

$$j = \sum_{i=1}^{h-1} m_i \qquad e \qquad r'' = r' - j$$

Sub-caso 2.2: A coluna a entrar na base e a coluna a deixar a base pertencem ao mesmo sub-sistema.

Suponhamos, por exemplo, que a coluna $(A^B)^\rho$, com $\rho \in I_h$, deva deixar a base para que $(A^N)^s$, com $s \in N_h$, deva entrar na base.

No caso de existir uma coluna $(A^B)^\tau$, com $\tau \in L_h$, tal que seja possível executarmos a troca indireta, podemos aplicar o sub-caso anterior.

Esgotada esta possibilidade devemos executar a troca direta. Neste caso, tendo em conta a forma da inversa da matriz básica (apêndice), não é necessário atualizar a inversa da base de trabalho, basta que $(A^I)^{-1}$ seja atualizada a nível de I_h:

$$(A^{I_h'})^{-1} = MP\ (r',\ s)\ .\ (A^{I_h})^{-1} \tag{29}$$

onde $MP(r',s) = \begin{bmatrix} 1 & \cdots & \alpha_{j+1} & & \\ & \ddots & \vdots & & \\ & & \alpha_{j+r''} & & \\ & & \vdots & \ddots & \\ & & \alpha_{j+m_h} & \cdots & 1 \end{bmatrix}$

$$\alpha_{j+i} = \begin{cases} -(\bar{A}^N)^s_{j+i} \ /\ (\bar{A}^N)^s_{j+r''} & ;\ j = 1,\ldots,m_h\ ;\ i \neq r'' \\ 1\ /\ (\bar{A}^N)^s_{j+r''} & ;\ j = r'' \end{cases} \tag{30}$$

$$j = \sum_{i=1}^{h-1} m_i \quad e \quad r'' = r' - j$$

2.4 O MÉTODO G.G.U.B. PARA A FORMA DUAL DO PROBLEMA BLOCO ANGULAR

O dual do problema bloco-angular, apresentado nas seções anteriores, é definido por:

$$\text{Maximizar } \Phi = p_o b_o + p_1 b_1 + p_2 b_2 + \dots + p_q b_q \tag{31}$$

Sujeita às restrições:

$$\begin{array}{llll} p_o A_o^1 + p_1 A_1 & & & \leq c_1 \\ p_o A_o^2 & + p_2 A_2 & & \leq c_2 \\ \vdots & \ddots & & \vdots \\ p_o A_o^q & & + p_q A_q & \leq c_q \end{array} \tag{32}$$

onde $p_o, p_1, \dots, p_q$ são conhecidos por variáveis duais (ou parâmetros de Lagrange).

Dada uma base A^B para o problema primal, se a ela pudermos associar um vetor $P = (p_o, p_1, \dots, p_q)$ tal que:

$$\bar{C}^B = C^B - P\,A^B = 0 \tag{33}$$

$$\bar{C}^N = C^N - P\,A^N \geq 0 \tag{34}$$

então, P é solução admissível para o problema dual, e A^B é uma base dual admissível para o problema primal.

Para resolver o problema dual podemos trabalhar com as variáveis duais ou primais. Com o conhecimento do estudo feito na seção anterior, optaremos por trabalhar com as variáveis primais.

O método que se apresenta a seguir resume-se na procura de uma base de partida, dual admissível, que possua uma estrutura semelhante a (4), para depois tratar da otimização da função Φ .

Procura de uma base dual admissível:

Como a base procurada deve ter as características apropriadas à estrutura bloco-angular, o primeiro passo a ser dado para sua definição será a procura de um conjunto de colunas chaves.

Observando o problema dual definido em (31), (32), se escolhermos um vetor p_o qualquer e fizermos $C' = C-p_oA_o$, teremos em p_ob_o uma quantidade constante e poderemos dividir este problema em q problemas duais auxiliares do tipo:

$$\text{Maximizar } p_ib_i$$
$$\text{Sujeita a } p_iA_i \leq C_i' \tag{35}$$

A cada um destes problemas duais auxiliares temos associado um problema primal auxiliar do tipo:

$$\text{Minimizar } C_i'X_i$$
$$\text{Sujeita a } A_iX_i = b_i \tag{36}$$
$$X_i \geq o$$

Se, para cada problema auxiliar conseguirmos determinar uma base A^{I_i}, então para $p=(p_1, p_2,\ldots,p_q)$, teremos:

$$C'^I - p\,A^I = C^I - p_oA_o^I - pA^I = C^I - P.A^I = \bar{C}^I = 0 \tag{37}$$

Denotando por J o conjunto das variáveis não básicas, teremos ainda:

$$C'^J - pA^J = C^J - p_o A_o^J - pA^J = C^J - P.A^J = \bar{C}^J \geq 0 \tag{38}$$

Desta forma temos em P uma solução admissível para o problema dual mas ainda não temos caracterizada uma base dual admissível. Para tanto, passaremos agora a procurar um conjunto de colunas não chaves entre aquelas cujo índice pertença ao conjunto J. De acordo com (15), temos que p segue variações de p_o, desta forma garantimos que $\bar{C}^I = 0$ permanecerá sempre válido. Nestas condições o problema dual pode ser reescrito:

$$\begin{aligned} &\text{Maximizar} \quad p_o b_o + pb \\ &\text{Sujeita a} \quad p_o A_o^J + pA^J \leq C^J \end{aligned} \tag{39}$$

Substituindo (15) em (39), ficamos com:

$$\begin{aligned} &\text{Maximizar } p_o \tilde{b}_o \\ &\text{Sujeita a } p_o \tilde{A}_o^J \leq \tilde{C}^J \end{aligned} \tag{40}$$

$$\text{com } \tilde{b}_o = b_o - A_o^I \; (A^I)^{-1} b \tag{41}$$

$$\tilde{A}_o^J = A_o^J - A_o^I \; (A^I)^{-1} A^J \tag{42}$$

$$\tilde{C}^J = C^J - C^I \; (A^I)^{-1} A^J \tag{43}$$

O problema primal associado ao problema dual definido por (40), é dado por:

$$\text{Minimizar} \quad \tilde{C}^J X_J$$

$$\text{Sujeita a} \quad \tilde{A}_o^J X_J = \tilde{b}_o \tag{44}$$

$$X_J \geq 0$$

Note-se a semelhança entre este problema e P.R. apresentado em 2.2.

A busca de uma base $\tilde{A}_o^L$, gera um vetor p_o tal que:

$$\bar{C}^L = \tilde{C}^L - p_o \tilde{A}_o^L = 0 \tag{45}$$

Seja N tal que $J = L + N$, então teremos:

$$\bar{C}^N = \tilde{C}^N - p_o \tilde{A}_o^N \geq 0 \tag{46}$$

Como $\bar{C}^I = 0$, então $P = (p_o, p)$, com p definido em (15), constitui uma solução admissível para o problema dual e o conjunto das colunas com índice em $B = L \oplus I$ é um conjunto ordenado de $m_o + m$ colunas linearmente independentes, que constituem uma base A^B dual admissível para o problema primal.

A característica adicional de A^B é que, por construção, possui uma estrutura semelhante a (4).

Otimização do problema dual:

Seja X^+ a solução básica, dual admissível, para o problema primal, correspondente ao vetor P^+ corrente. Estes vetores guardam entre si a seguinte relação:

$$CX^{+}= C^{B}X^{B+}+ C^{N}X^{N+}= P^{+}A^{B}X^{B+}+(\bar{C}^{N}+P^{+}A^{N})X^{N+}= P^{+}(A^{B}X^{B+}+ A^{N}X^{N+})+ \bar{C}^{N}X^{N+}=P^{+}b \qquad (47)$$

Contudo as componentes de X^{+} podem ser positivas, negativas ou nulas e $X^{N+} = 0$.

Desta forma, trabalhando com as variáveis primais, quando encontrarmos uma base dual admissível tal que $X^{B*} \geq 0$ e $X^{N*} = 0$ teremos encontrado a solução ótima X^{*} para o problema primal e o vetor P^{*} correspondente, como solução ótima para o dual.

Portanto, toda coluna $(A^{B})^{\rho}$, com ρ ocupando a posição r em B, tal que $(X^{B+})^{\rho}$ seja negativa deverá deixar a base.

Como no método primal, temos dois casos a considerar:

Caso 1: $r \leq m_{o}$.

Para determinarmos a coluna que deve entrar na base, devemos antecipadamente expressar a linha r de A^{N} em termos da atual base:

$$(\bar{A}^{N})_{r} = ((A^{B})^{-1})_{r} \cdot A^{N} = ((\tilde{A}_{o}^{L})^{-1})_{r} \cdot \tilde{A}_{o}^{N} = (\bar{A}_{o}^{N})_{r} \qquad (48)$$

A coluna procurada é obtida através do seguinte coeficiente:

$$\theta_{s} = \frac{-(\bar{C}^{N})^{s}}{(\bar{A}_{o}^{N})_{r}^{s}} = \underset{j \in N}{\mathrm{Min}} \left\{ \frac{-(\bar{C}^{N})^{j}}{(\bar{A}_{o}^{N})_{r}^{j}} \; ; \; (\bar{A}_{o}^{N})_{r}^{j} < 0 \right\} \qquad (49)$$

Com $\bar{C}^{N}$ dada por (17).

Se $(\bar{A}_{o}^{N})_{r} \geq 0$, então o problema dual não tem solução finita, o que significa dizer que o problema primal não apresenta solução factível.

Caso contrário a coluna $(A^N)^s$, com $s \in N$ deve entrar na base, provocando a atualização da base de trabalho, como no caso 1 do método primal.

Caso 2: $r = m_o + r'$. Temos então dois subcasos a estudar:

Sub-caso 2.1: $(\hat{A}^L)_{r'} \neq 0$.

Esta condição satisfeita significa que existe uma coluna $(A^B)^\tau$, com $\tau \in L$, que pode ser permutada com $(A^B)^\rho$.

Como no sub-caso correspondente do método primal, devemos atualizar $(\tilde{A}_o^L)^{-1}$ e $(A^I)^{-1}$.

Tendo completada a permuta, recaímos no caso anterior.

Sub-caso 2.2: $(\hat{A}^L)_{r'} = 0$.

Sob esta condição temos que, obrigatoriamente, executar a troca direta com uma coluna não básica. A linha r de A^N, expressa em termos da base corrente é dada por:

$$(\bar{\bar{A}}^N)_r = ((A^B)^{-1})_r \, . \, A^N = (\hat{A}^N)_{r'} \qquad (50)$$

Devemos observar que a coluna que vai entrar para a base pertence, necessariamente, ao mesmo sub-sistema da que vai deixar a base.

Supondo que $\rho \in I_h$, a coluna procurada é determinada através do coeficiente:

$$\Theta_s = \frac{-(\bar{C}^N)^s}{(\hat{A}^{N_h})^s_{r'}} = \operatorname*{Min}_{j \in N_h} \left\{ \frac{-(\bar{C}^N)^j}{(\hat{A}^{N_h})^j_{r'}} \;;\; (\hat{A}^{N_h})^j_{r'} < 0 \right\} \qquad (51)$$

Se $(\hat{A}^{N_h})_{r'} \geq 0$, então o problema dual não tem solução finita e o problema primal não tem solução factível.

Caso contrário a coluna $(A^N)^s$ deve ser permutada com $(A^B)^\rho$ provocando a atualização de $(A^{I_h})^{-1}$, como no método primal.

CAPÍTULO III

UM MÉTODO DUAL SIMPLEX PARA PROBLEMAS DE PROGRAMAÇÃO LINEAR COM VARIÁVEIS CANALIZADAS

3.1 INTRODUÇÃO

No capítulo anterior pudemos estudar os métodos de resolução de problemas de programação linear aplicáveis à estrutura bloco-angular. Porém, a aplicação de tais métodos só poderá ser levada a efeito quando as variáveis do problema devam respeitar apenas a restrição de não negatividade.

Problemas semelhantes ao apresentado no Capítulo I têm suas variáveis, ou parte delas, definidas em um intervalo de variação. Por este motivo, estas variáveis são chamadas de variáveis canalizadas.

Assim, para resolvermos problemas deste tipo devemos desenvolver uma especialização do método GGUB para atender às "restrições de canalização".

Naturalmente esta especialização do método GGUB deverá basear-se pelo caso mais simples de Programação Linear com variáveis canalizadas, ou seja, o caso em que a matriz de restrições não apresenta qualquer estrutura particular.

Mas, como mencionamos anteriormente, não conhecemos na literatura trabalhos que tratem da resolução de problemas de Programação Linear com variáveis canalizadas pelo método dual. Neste capítulo, esquecemos temporariamente a estrutura bloco- angular e propomos um método dual simplex para problemas de programação linear com variáveis canalizadas.

3.2 O MÉTODO

O problema de minimização de uma função linear sujeito a restrições lineares com variáveis "canalizadas", em sua forma primal, é definido por:

$$(P) \quad \begin{array}{l} \text{minimizar } C\,X \\ \text{sujeito a } AX = b \\ \qquad\qquad d^1 \leq X \leq d^2 \end{array}$$

Onde A é uma matriz de dimensão m x n, C é um vetor- linha com n elementos, b é um vetor coluna com m elementos e X, d^1 e d^2 são vetores-coluna com n elementos.

Se P é o vetor das variáveis duais (multiplicadores de Lagrange), podemos então definir a função Lagrangeana e a função dual para este problema:

$$L(X,P) = CX + P(b - AX) \tag{1}$$

$$\Phi(P) = \min_{d^1 \leq X \leq d^2} \{L(X,P)\} \tag{2}$$

O problema dual é, então, dado por:

$$(D) \quad \begin{array}{l} \text{maximizar } \Phi\,(P) \\ \text{sujeito a } P \in \mathbb{R}^m \end{array}$$

Denotaremos por B ao conjunto dos índices das colunas da matriz A que formam uma base do $\mathbb{R}^m$ e N ao seu complemento em relação ao espaço Euclidiano.

Se A^B é a matriz básica, as restrições do problema (P) podem ser escritas da seguinte forma:

$$A^B X_B + A^N X_N = b$$

$$\Rightarrow X_B = (A^B)^{-1} b - (A^B)^{-1} A^N X_N$$

$$\Rightarrow X_B = \hat{b} - \hat{A}^N X_N$$

Chamaremos $X^+ = [X_B^+, X_N^+]$ de solução básica de (P) se cada X_j, $j \in N$ é colocado em seu limite inferior d_j^1 ou em seu limite superior d_j^2:

$$X^+ = \left[\begin{array}{c} X_B^+ \\ \hdashline X_N^+ \end{array}\right] = \left[\begin{array}{c} \hat{b} - A^N d_N^k \\ \hdashline d_N^k \end{array}\right] \qquad (3)$$

para k = 1 ou k = 2.

Se além disto tivermos $d_B^1 \leq X_B^+ \leq d_B^2$, então X^+ é chamada solução básica factível de (P).

Dizemos que X^+ é uma solução básica degenerada de (P) se

$X_j^+ = d_j^1$ ou $X_j^+ = d_j^2$ para $j \in B$.

Dada uma base A^B, a função dual pode ser reescrita como se segue:

$$\Phi(P) = \text{Min} \{ (C^B - PA^B)\, X_B + (C^N - PA^N)\, X_N + Pb \} \tag{4}$$

$$\text{s/a} \quad d_B^1 \leq X_B \leq d_B^2$$

$$d_N^1 \leq X_N \leq d_N^2$$

Chamaremos $P = C^B (A^B)^{-1}$ de solução básica de (D).

$$\text{Fazendo } Z^B = PA^B \tag{5}$$

$$\text{e } Z^N = PA^N = C^B \hat{A}^N \tag{6}$$

temos que:

$$\Phi(P) = Pb + \text{Min } (C^N - Z^N)\, X_N \tag{7}$$

$$\text{s/a} \quad d_B^1 \leq X_B \leq d_B^2$$

$$d_N^1 \leq X_N \leq d_N^2$$

Nestas condições podemos observar que o problema de minimização acima independe das variáveis básicas, podendo estas assumirem,na solução ótima, qualquer valor no intervalo $[d_B^1, d_B^2]$.

Observamos ainda que, para $j \in N$, teremos na solução ótima:

$$X_j = d_j^1 \quad \text{se} \quad Z^J < C^j$$

$$X_j = d_j^2 \quad \text{se} \quad Z^J > C^j$$

X_j assume qualquer valor no intervalo $[d_j^1, d_j^2]$ se $Z^j = C^j$.

Portanto, uma solução X^* para (7), terá a seguinte forma:

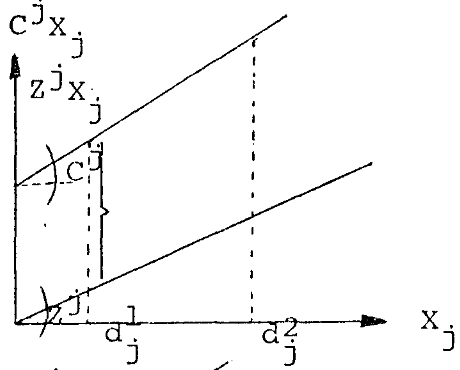
$$X^* = \left[\begin{array}{c} X_B^* \\ \hdashline X_N^* \end{array}\right] = \left[\begin{array}{c} \in [d_B^1,\ d_B^2] \\ \hdashline d_N^k \end{array}\right] \tag{8}$$

onde $d_j^k = d_j^1$ se $z^j < C^j$ (fig. 1)

$d_j^k = d_j^2$ se $z^j > C^j$ (fig. 2)

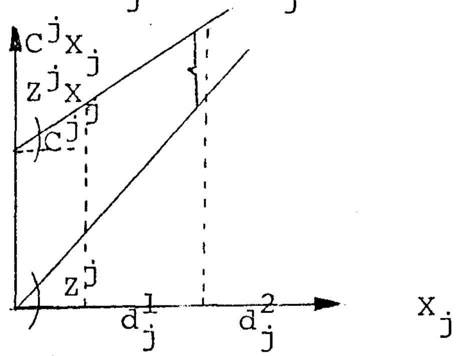
e $d_j^k = (d_j^1$ ou $d_j^2)$ se $z^j = C^j$ (fig. 3), para $j \in N$

Fig. 1



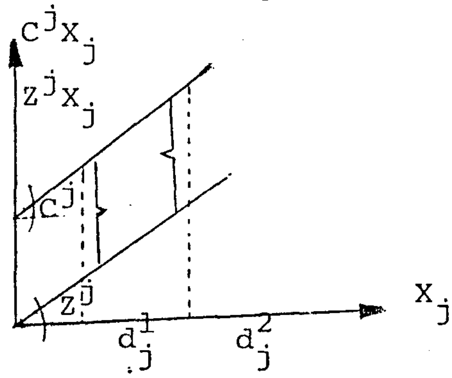
$z^j < C^j \rightarrow X_j^* = d_j^1$

Fig. 2



$z^j > C^j \rightarrow X_j^* = d_j^2$

Fig. 3



$z^j = C^j \rightarrow X_j^* = (d_j^2$ ou $d_j^2)$

À uma solução X^* definida como em (8), chamaremos solução MIN. Podemos então obter uma solução básica a partir de:

$$X^+ = \begin{bmatrix} X_B^+ \\ \hdashline X_N^+ \end{bmatrix} = \begin{bmatrix} \hat{b} - \hat{A}^N \, d_N^k \\ \hdashline d_N \end{bmatrix} = \begin{bmatrix} \hat{b} - \hat{A}^N \, X_N^* \\ \hdashline X_N^* \end{bmatrix} \tag{9}$$

Se, para o vetor P corrente, ou seja, para a matriz básica corrente, tivermos $X_B^+ \in [d_B^1, d_B^2]$ então a solução básica X^+ será uma solução MIN.

Neste caso X^+ será factível para (P). Pela teoria da dualidade, X^+ será então solução ótima de (P) e de (D), e como não temos "fosso" de dualidade: $\Phi(P) = CX^+$.

Temos então definido um teste de otimalidade para um Método do Simplex de resolução, tentaremos agora desenvolver seus passos intermediários na busca de uma solução.

Assim, após o cálculo de uma solução básica através de (9), e tendo sido verificado que tal solução não é factível para (P), devemos procurar uma nova solução básica que aumente $\Phi(P)$.

Para tanto, provocaremos uma perturbação em (P) obtendo

$$P' = P + \delta(A^B)^{-1} \tag{10}$$

Para $\delta = (\delta_1, \delta_2, \ldots, \delta_m)$, com os componentes δ_i suficientemente pequenos.

Nestas condições, teremos:

$$Z'^B = (P + \delta(A^B)^{-1}) \, A^B = PA^B + \delta = Z^B + \delta \tag{11}$$

$$Z'^{N} = (P + \delta(A^{B})^{-1})\, A^{N} = PA^{N} + \delta(A^{B})^{-1}\, A^{N} = Z^{N} + \delta\hat{A}^{N} \qquad (12)$$

O valor de $\Phi(P')$, semelhantemente ao valor de $\Phi(P)$ dado por (4), pode então ser obtido através de (10), (11) e (12):

$$\Phi(P') = Pb + \delta(A^{B})^{-1}\, b + \text{Min}'\{(c^{B}-Z^{B}-\delta)X_{B} + (C^{N}-Z^{N}-\delta\hat{A}^{N})X_{N}\}$$

$$\text{s/a} \quad d_{B}^{1} \le X_{B} \le d_{B}^{2}$$

$$d_{N}^{1} \le X_{N} \le d_{N}^{2}$$

ou ainda

$$\Phi(P') = Pb + \delta\hat{b} + \text{Min}'\{-\delta X^{B}\} + \text{Min}'\{(C^{N} - Z^{N} - \delta\hat{A}^{N})\, X_{N}\} \qquad (13)$$

$$d_{B}^{1} \le X_{B} \le d_{B}^{2} \qquad d_{N}^{1} \le X_{N} \le d_{N}^{2}$$

Os valores das variáveis básicas que minimizam $-\delta X_{B}$, no intervalo $[d_{B}^{1}, d_{B}^{2}]$, podem ser obtidos facilmente, verificando, para cada $j \in B$:

$$\text{se } \delta_{j} > 0 \Rightarrow X_{j} = d_{j}^{2}$$

$$\text{se } \delta_{j} < 0 \Rightarrow X_{j} = d_{j}^{1}$$

$$\text{se } \delta_{j} = 0 \Rightarrow X_{j} \in [d_{j}^{1}, d_{j}^{2}]$$

Assim, $\text{Min}'\{-\delta X_{B}\} = -\delta d_{B}^{k}$

$$d_{B}^{1} \le X_{B} \le d_{B}^{2}$$

Para as variáveis não básicas, se considerarmos δ suficientemente pequeno, de forma que o sinal de cada componente de

$$\hat{C}'^{N} = C^{N} - Z^{N} - \delta\hat{A}^{N} = \hat{C}^{N} - \delta\hat{A}^{N} \tag{14}$$

seja o mesmo que o sinal de cada componente correspondente de $\hat{C}^{N}$, então teremos:

$$\text{Min } \{(C^{N} - Z^{N} - \delta\hat{A}^{N})\ X_{N}\} = (C^{N} - Z^{N} - \delta\hat{A}^{N})\ d_{N}^{k}$$

$$d_{N}^{1} \leq X_{N} \leq d_{N}^{2}$$

Portanto (13) pode ser reescrita:

$$\begin{aligned}\Phi(P') &= Pb + (C^{N} - Z^{N})\ d_{N}^{k} + \delta(\hat{b} - \hat{A}^{N}\ d_{N}^{k} - d_{B}^{k}) \\ &= \Phi(P) + \delta(X_{B}^{+} - d_{B}^{k}) \\ &= \Phi(P) + \sum_{j\in B} \delta_{j}\ (X_{j}^{+} - d_{j}^{k})\end{aligned} \tag{15}$$

Esta última equação nos mostra que podemos aumentar Φ (P) da seguinte forma:

se $X_{j}^{+} > d_{j}^{2}$ então fazemos $\delta_{j} > 0$ e neste caso teremos $X_{j}^{*} = d_{j}^{2}$, $j\in B$

se $X_{j}^{+} < d_{j}^{1}$ então fazemos $\delta_{j} < 0$ e teremos $X_{j}^{*} = d_{j}^{1}$, $j\in B$

se $d_{j}^{1} \leq X_{j}^{*} \leq d_{j}^{2}$ então fazemos $\delta_{j} = 0$.

Suponhamos, então, que tenhamos uma variável X_{ℓ} com ℓ ocupando a posição r em B e tal que $X_{\ell}^{+} \notin [d_{\ell}^{1}, d_{\ell}^{2}]$. Queremos um procedimento simplex para solucionar (D).

Para tanto, fazemos $\delta = t\ (0,0,..0,1,0,...,0) = t.e_r$

Nestas condições e a partir de (11) e (12) teremos:

$Z'^{j} = Z^{j} \quad , \quad j \in B \quad e \quad j \neq \ell$

$Z'^{\ell} = Z^{\ell} + t$

$Z'^{N} = Z^{N} + t.e_r.\hat{A}^{N} = Z^{N} + t.\hat{A}^{N}_{r}$

e de (14):

$$\hat{C}'^{j} = \hat{C}^{j} - t\hat{A}^{B}_{r} \quad , \quad j \in N \tag{16}$$

Sabemos ainda que a equação (15) será válida enquanto o sinal de cada componente $\hat{C}'^{N}$ for igual ao sinal da respectiva componente de $\hat{C}^{N}$. Portanto podemos atribuir valores a t até que um dos custos relativos de $\hat{C}'^{N}$ se anule.

O valor de t pode ser obtido a partir de (16), porém temos dois casos a considerar:

(a) $X^{*}_{\ell} > d^{2}_{\ell}$. Neste caso deveremos ter $t > 0$.

Calculamos então $t_s = \text{Min}\ \{t_{s_1}, t_{s_2}\}$

onde $t_{s_1} = \underset{j \in N}{\text{Min}} \left\langle \left| \dfrac{\hat{C}^{j}}{\hat{A}^{j}_{r}} \right| \right\rangle$

$\hat{C}^{j} > 0 \quad e \quad \hat{A}^{j}_{r} > 0$

$t_{s_2} = \underset{j \in N}{\text{Min}} \left\langle \left| \dfrac{\hat{C}^{j}}{\hat{A}^{j}_{r}} \right| \right\rangle$

$\hat{C}^{j} < 0 \quad e \quad \hat{A}^{j}_{r} < 0$

(b) $X_\ell^+ < d_\ell^1$. Neste caso deveremos ter $t < 0$.

Calculamos então $t_s = \text{Max}\,\{t_{s_3}, t_{s_4}\}$

onde $t_{s_3} = \underset{j \in N}{\text{Max}} \left\{ \dfrac{\hat{C}^j}{\hat{A}_r^j} \right\}$

$\hat{C}^j > 0$ e $\hat{A}_r^j < 0$

$t_{s_4} = \underset{j \in N}{\text{Max}} \left\{ \dfrac{\hat{C}^j}{\hat{A}_r^j} \right\}$

$\hat{C}^j < 0$ e $\hat{A}_r^j > 0$

Com este procedimento, iremos alterar os custos relativos das variáveis não básicas, conforme (16):

$\hat{C}'^j = \hat{C}^j - t_s\, \hat{A}_r^j \quad ; \quad j \in N$

e em especial para $j = s$: $\hat{C}'^s = 0$

Então a variável X_s deverá substituir a variável básica X_ℓ.

Neste caso o novo custo relativo para X_ℓ será:

$\hat{C}'^\ell = -t_s$

Com estas alterações teremos uma nova solução básica.

Primeiramente devemos observar que, para $j \in N$, $j \neq s$, continuaremos tendo $X_j'^+ = X_j^+ = X_j^* = d_j^k$

Temos ainda que $X_\ell'^+ = d_\ell^k$.

Resta verificar os novos valores para as variáveis básicas, que podem ser obtidos a partir de (13) e sabendo-se que sofrerão uma alteração de ΔX_B:

$$X_B'^{+} = X_B^{+} - \Delta X_B = \hat{b} - \hat{A}^{N-\{s\}} . X_{N-\{s\}} - \hat{A}^s (X_s - \Delta_s)$$

assim concluimos que:

$$\Delta X_B = -\hat{A}^s \Delta X_s$$

mas a variação de X_ℓ é conhecida: $\Delta X_\ell = X_\ell^+ - d_\ell^k$, portanto, para a nova variável básica X_ℓ:

$$\Delta X_\ell = -\hat{A}_r^s \Delta X_s = X_\ell^+ - d_\ell^k$$

o que nos leva a concluir que

$$\Delta X_s = \frac{d_\ell^k - X_\ell^+}{\hat{A}_r^s}$$

segue-se que

$$X_s'^{+} = d_s^k - \frac{d_\ell^k - X_\ell^+}{\hat{A}_r^s}$$

e

$$X_j'^{+} = X_j^+ - \hat{A}_j^s \frac{d_\ell^k - X_\ell^+}{\hat{A}_r^s} \quad , \; j \in B \;\; e \;\; j \neq \ell \; .$$

Para que uma iteração do Simplex fique completada e caracterizada devemos efetuar a atualização da matriz básica, ou melhor, da inversa da matriz básica.

Isto pode ser facilmente feito, executando-se o pivoteamento em torno do elemento $\hat{A}^s_r$.

CAPÍTULO IV

UM MÉTODO DUAL SIMPLEX PARA PROBLEMAS DE PROGRAMAÇÃO LINEAR COM ESTRUTURA BLOCO-ANGULAR E VARIÁVEIS CANALIZADAS

4.1 INTRODUÇÃO

O método que aqui expomos resultou de adaptações da teoria exposta no Capítulo II e do método apresentado no capítulo anterior.

Trata-se, portanto, de uma especialização do método GGUB para problemas de programação linear que apresentem a estrutura bloco-angular e tenham variáveis do tipo canalizadas.

Optamos por trabalhar com a forma dual pelo fato da mesma apresentar maiores facilidades quanto a definição de uma base inicial, evitando uma Fase I no método de resolução. Esta opção foi feita ainda por termos encontrado dificuldades na procura de bibliografia que trate com a forma dual para problemas de programação linear que apresentem variáveis canalizadas.

4.2 O MÉTODO

INICIALIZAÇÃO:

Caracterizado um problema de programação linear que apresente estrutura bloco-angular e variáveis canalizadas, devemos inicialmente escolher um conjunto B de colunas da matriz de restrições para compor uma matriz A^B que tenha a propriedade de ser regular e que possa assumir a seguinte forma:

$$A^B = \left[\begin{array}{c|cccc} A_o^L & A_o^{I_1} & A_o^{I_2} & \cdots\cdots & A_o^{I_q} \\ \hline A^L & A^{I_1} & & & \\ & & A^{I_2} & & \\ & & & \ddots & \\ & & & & A^{I_q} \end{array}\right] \quad \text{com } B = L \oplus I \text{ e } I = I_1 \oplus I_2 \oplus \ldots \oplus I_q .$$

A partir desta estrutura podemos obter $(A^B)^{-1}$ a partir de $(\tilde{A}_o^L)^{-1}$, $(A^I)^{-1}$, $(A^{I_2})^{-1}$ ..., $(A^{I_q})^{-1}$ (vide apêndice), onde $(\tilde{A}_o^L) = A_o^L - A_o^I \hat{A}^L$ e $\hat{A}^L = (A^I)^{-1} A^L$.

Portanto, nosso trabalho fica reduzido à obtenção das inversas das matrizes A^{I_i}, ao cálculo da matriz $\hat{A}^L$ (será útil a mantermos para futuros cálculos), ao cálculo de $\tilde{A}_o^L$ (nossa "base de trabalho") e de sua inversa.

Se denotarmos por N ao conjunto dos índices das variáveis não básicas, o custo relativo é dado por $\hat{C}^N = C^N - C^B (A^B)^{-1} A^N$

Considerando-se a estrutura de $(A^B)^{-1}$, este cálculo pode ser resumido a: $\hat{C}^N = C^N - \hat{C}^B A^N$ onde $\hat{C}^B = (\hat{C}^L, \hat{C}^I) = (C^L, C^I)(A^B)^{-1}$

Nestas condições teremos: $\hat{C}^L = (C^L - C^I \hat{A}^L)\ (\tilde{A}_o^L)^{-1}$

$$\hat{C}^I = (C^I - \hat{C}^L A_o^I)\ (A^I)^{-1}$$

Para maior eficiência na programação do método, podemos associar o cálculo do custo relativo ao cálculo do valor de uma variável não básica. Desta forma, para cada $J \in N$:

$$\hat{c}^j = c^j - \hat{c}^B A^j$$

$$\text{Se } \hat{c}^j > 0 \text{ fazemos } x^j = d_j^1 \tag{1}$$

$$\text{Se } \hat{c}^j < 0 \text{ fazemos } x^J = d_j^2$$

O valor das variáveis básicas é dado por:

$$x^B = (A^B)^{-1} (b - A^N x^N) \text{ ou ainda:}$$

$$\begin{bmatrix} x^L \\ \hdashline x^I \end{bmatrix} = (A^B)^{-1} \left[\begin{pmatrix} b_o \\ \hdashline b^I \end{pmatrix} - \begin{pmatrix} A_o^N \\ \hdashline A^N \end{pmatrix} x^N \right]$$

Lembrando mais uma vez a estrutura de $(A^B)^{-1}$, concluimos que

$$x^L = (\tilde{A}_o^L)^{-1} (b_o - A_o^N x^N - A_o^I (A^I)^{-1} (b^I - A^N x^N)) \tag{2}$$

$$\text{e } x^I = -(A^I)^{-1} A^L (\tilde{A}_o^L)^{-1} (b_o - A_o^N x^N) + ((A^I)^{-1} + (A^I)^{-1} A^L (\tilde{A}_o^L)^{-1} A_o^I (A^I)^{-1})(b^I - A^N x^N)$$

substituindo (2) nesta última equação, temos que:

$$x^I = (A^I)^{-1} (b^I - A^N x^N - A^L x^L) \tag{3}$$

Os demais passos deste método serão baseados no método GGUB, exposto anteriormente, e na seguinte estratégia:

Etapa 1: aplicaremos o teste de otimalidade nas variáveis não chaves (conjunto L) e só passaremos a etapa seguinte se todas estas variáveis satisfizerem o critério de otimalidade.

Etapa 2: aplicaremos o teste de otimalidade nas variáveis chaves (conjunto I). Se todas satisfizerem o critério de otimalidade sem que haja necessidade de se efetuar qualquer alteração na matriz básica, então temos em mãos a solução ótima, caso contrário procuraremos colocar nas condições de otimalidade o máximo de variáveis chaves que pudermos e em seguida retornamos a etapa 1 para que o processo se repita.

Etapa 1: Otimização das variáveis chaves

O critério de otimalidade é considerado satisfeito se, para todo $j \in L$, tivermos: $d_j^1 \leq X^j \leq d_j^2$.

Uma variável X^ρ, $\rho \in L$, tal que $X^\rho > d_\rho^2$ ou $X^\rho < d_\rho^1$, deve deixar a base. Suponhamos que X^ρ ocupe a posição $r \leq m_o$ em B.

A variável que deverá ocupar seu lugar será a variável X^s, cujo índice s tenha correspondência com o índice do coeficiente obtido por:

$$t_s = \text{Min}\ \{t_{s_1}, t_{s_2}\} \quad \text{se } X^\rho > d_\rho^2 \qquad (k=2)$$

$$\text{ou } t_s = \text{Max}\ \{t_{s_3}, t_{s_4}\} \quad \text{se } X^\rho < d_\rho^1 \qquad (k=1)$$

$$\text{para } t_{s_1} = \underset{\substack{j \in N \\ \hat{C}^j > 0,\ A_r^j > 0}}{\text{Min}} \left\{ \left| \frac{\hat{C}^j}{A_r^j} \right| \right\}$$

$$t_{s_2} = \underset{j \in N}{\text{Min}} \left\{ \frac{\tilde{C}^j}{\hat{A}_r^j} \right\}$$

$$\hat{C}^j < 0, \quad \hat{A}_r^j < 0$$

$$t_{s_3} = \underset{j \in N}{\text{Max}} \left\{ \frac{\hat{C}^j}{\hat{A}_r^j} \right\}$$

$$\hat{C}^j > 0, \quad \hat{A}_r^j < 0$$

$$t_{s_4} = \underset{j \in N}{\text{Max}} \left\{ \frac{\hat{C}^j}{\hat{A}_r^j} \right\}$$

$$\hat{C}^j < 0, \quad \hat{A}_r^j > 0$$

$\hat{C}^j$ é dado por (1) e $\hat{A}_r^j$ pode ser facilmente obtido, se lembrarmos que $r \leq m_o$:

$$\hat{A}_r^j = (\tilde{A}_o^L)_r^{-1} \, \tilde{A}_o^j \qquad \text{onde} \qquad \tilde{A}_o^j = A_o^j - A_o^I (A^I)^{-1} \, A^j$$

Os custos relativos devem ser recalculados:

$$\hat{C}'^N = \hat{C}^N - t_s \hat{A}_r^N = \hat{C}^N - \frac{\hat{C}^s}{\hat{A}_r^s} \; \hat{A}_r^N$$

$$\hat{C}^s = 0$$

$$\hat{C}^\rho = -t_s = -\frac{\hat{C}^s}{\hat{A}_r^s}$$

Também as variáveis abaixo terão seus valores recalculados:

$$X^{\rho} = d^{k}_{\rho}$$

$$X'^{B} = X^{B} + \frac{\hat{A}^{s}}{\hat{A}^{s}_{r}} \quad \Delta X^{\rho}$$

$$X^{s} = X^{s} + \frac{\Delta X^{\rho}}{\hat{A}^{s}_{r}}$$

Finalmente, devemos atualizar os conjuntos L e N e a inversa da matriz básica, bastando para tanto atualizar a inversa da base de trabalho, através de pivoteamento:

$$(\tilde{A}^{L'}_{o})^{-1} = MP(r,s) \; (\tilde{A}^{L}_{o})^{-1}$$

onde $MP(r,s) = \begin{pmatrix} 1 & & \alpha_1 & & \\ & \ddots & \vdots & & \\ & & \alpha_r & & \\ & & \vdots & \ddots & \\ & & \alpha_{m_o} & & 1 \end{pmatrix}$

$$\text{com } \alpha_j = \begin{cases} -A^{s}_{j} \,/\, A^{s}_{r} & j = 1,2\ldots,m_o \;;\; j \neq r \\ 1 \;/\; A^{s}_{r} & j = r \end{cases}$$

Etapa 2: Otimização das variáveis chaves.

Nesta etapa aplicamos o teste de otimalidade para toda variável chave (X^{j}, $j \in I$) para verificar se satisfaz: $d^{1}_{j} \leq X^{j} \leq d^{2}_{j}$.

Suponhamos uma variável X^{ρ}, $\rho \in I$, tal que $X^{\rho} > d^{2}_{\rho}$ ou

$x^{\rho} > d^{1}_{\rho}$. Esta variável deverá deixar a base.

Também aqui imitaremos os passos do GGUB, pois esta variável poderá deixar a base de duas formas: através de uma permutação com uma não chave ou diretamente com uma não básica.

Suponhamos, então, que x^{ρ} ocupe a posição $r = m_o + r'$ em B, e portanto, ocupando a posição r' em I. Desta forma $\rho \in I_h$.

Temos então dois casos a considerar:

Caso 1: existe $(\hat{A}^{L})^{t}_{r'} \neq 0$

Neste caso a coluna A^{τ}, que ocupa a posição t em B ($t \in L_h$), é escolhida para fazermos uma permutação com a coluna A^{ρ}, que ocupa a posição r' em I ($r' \in I_h$).

Isto nos leva à atualizações nos conjuntos L, I e na matriz básica:

$$(\tilde{A}^{L'}_{o})^{-1} = G(\tilde{A}^{L}_{o})^{-1} \quad \text{com } G = \begin{array}{c} \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ \hline & & -(\hat{A}^{L})_{r'} & & \\ \hline & & & 1 & \\ & & & & \ddots \\ & & & & & 1 \end{pmatrix} \end{array} \begin{array}{l} \\ \\ \\ \leftarrow \text{linha } t \\ \\ \\ \\ \end{array}$$

$$(A^{I'_h})^{-1} = \alpha (A^{I_h})^{-1} \quad \text{com } \alpha = \begin{pmatrix} 1 & \cdots & \alpha_{j+1} & & \\ & \ddots & \vdots & & \\ & & \alpha_{j+r''} & & \\ & & \vdots & \ddots & \\ & & \alpha_{j+m_h} & \cdots & 1 \end{pmatrix}$$

onde $r'' = r' - j$ e $j = \sum_{i=1}^{h-1} m_i$

$$\alpha_{j+k} = \begin{cases} -(\hat{A}^L)^{\tau}_{j+k} \;/\; (\hat{A}^L)^{\tau}_{j+r''} & k = 1,\dots,m_h \;;\; k \neq r'' \\ 1 \;/\; (\hat{A}^L)^{\tau}_{j+r''} & K = r'' \end{cases}$$

Caso 2: $(\hat{A}^L)_{r'} = 0$

Neste caso uma variável não básica X^s deverá ser permutada com X^ρ . Este índice s é obtido em correspondência com a obtenção do coeficiente t_s, dado por:

$$t_s = \text{Min}\ \{t_{s_1}, t_{s_2}\} \quad \text{se} \quad X^\rho > d_2^\rho \qquad (K = 2)$$

$$\text{ou } t_s = \text{Max}\ \{t_{s_3}, t_{s_4}\} \quad \text{se} \quad X^\rho < d_1^\rho \qquad (K = 1)$$

onde $t_{s_1} = \underset{j \in N}{\text{Min}} \left\{ \left| \dfrac{\hat{C}^j}{\hat{A}^j_{r'}} \right| \right\}$ $\quad$ $t_{s_2} = \underset{j \in N}{\text{Min}} \left\{ \left| \dfrac{\hat{C}^j}{\hat{A}^j_{r'}} \right| \right\}$

$\hat{C}^j > 0,\ \hat{A}^j_{r'} > 0$ $\quad$ $\hat{C}^j < 0,\ \hat{A}^j < 0$

$t_{s_3} = \underset{j \in N}{\text{Max}} \left\{ \left| \dfrac{\hat{C}^j}{\hat{A}^j_{r'}} \right| \right\}$ $\quad$ $t_{s_4} = \underset{j \in N}{\text{Max}} \left\{ \left| \dfrac{\hat{C}^j}{\hat{A}^j_{r'}} \right| \right\}$

$\hat{C}^j < 0,\ \hat{A}^j_{r'} > 0$ $\quad$ $\hat{C}^j > 0,\ \hat{A}^j_{r'} < 0$

C^j é dado por (1) e $\hat{A}^j_{r'}$ pode ser facilmente obtido, se lembrarmos que $(\hat{A}^L_{r'}) = 0$:

$$\hat{A}^j_{r'} = (A^B)^{-1}_{r'}\ A^j = (A^B)^{-1}_{r'} \begin{pmatrix} A^j_o \\ \hdashline A^j \end{pmatrix} = (A^I)^{-1}_{r'}\ A^j \quad ; \quad j \in N$$

Os custos relativos e os novos valores das variáveis básicas e não básicas devem ser recalculados de maneira análoga ao que foi apresentado na etapa 1.

Devemos também atualizar os conjuntos I e N e a inversa da matriz básica, bastando para isto fazer a atualização a nível do conjunto I_h:

$$(A^{I_h'})^{-1} = MP\ (r',\ s)\ (A^{I_h})^{-1}$$

$$\text{com } MP(r',\ s) = \begin{pmatrix} 1 & \cdots & \alpha_{J+1} & & \\ & \ddots & \vdots & & \\ & & \alpha_{J+r''} & & \\ & & \vdots & \ddots & \\ & & \alpha_{J+m_h} & \cdots & 1 \end{pmatrix}$$

para $r'' = r' - j$ e $j = \sum_{i=1}^{h-1} = m_i$

$$\alpha_{j+k} = \begin{cases} -(\hat{A})^{s}_{j+k} \ / \ (\hat{A})^{s}_{j+r''} & k = 1,\ldots,m_h \ ; \ k \neq r'' \\ 1/ \ (\hat{A})^{s}_{j+r''} & K=r'' \end{cases}$$

CAPÍTULO V

UM PROGRAMA DE COMPUTADOR PARA PROGRAMAÇÃO DA PRODUÇÃO COM HORIZONTE DE QUATRO MESES

5.1 INTRODUÇÃO

Neste capítulo apresentamos a estrutura do programa de computador desenvolvido para solucionar o "problema de programação da produção com horizonte de quatro meses", exposto no Capítulo I.

Este programa tem suas bases no "método dual-simplex para problemas de programação linear com estrutura bloco-angular e variáveis canalizadas", exposto no capítulo anterior.

Para um melhor aproveitamento do espaço de armazenagem de dados, fazemos uso de alguns recursos de programação, que serão a seguir apresentados.

Este capítulo, por fim, reúne uma série de informações utilizadas para a implementação e melhor entendimento da estrutura computacional. Para tanto, apresentamos ainda a descrição do programa através de um algorítmo sintético.

5.2 ESTRUTURAS DE ARMAZENAMENTO

A formulação matemática do problema de programação da produção com horizonte de quatro meses, como já pudemos observar no Capítulo I deste trabalho, nos conduz a uma "matriz tecnológica " do problema de minimização, que assume a seguinte forma:

$$
\left[\begin{array}{cccccccccccccccccccccc}
I_N & & & & & I_N & & & & & I_N & & & & & I_N & & & & & & \\
I_N & & & & & I_N & & & & & I_N & & & & & & & & & & -I_N & \\
I_N & & & & & I_N & & & & & & & & & & & & & & & & -I_N \\
A & -I_M & -I_M & -I_M & +I_M & & & & & & & & & & & & & & & & & \\
 & & & & & A & -I_M & -I_M & -I_M & +I_M & & & & & & & & & & & & \\
 & & & & & & & & & & A & -I_M & -I_M & -I_M & +I_M & & & & & & & \\
 & & & & & & & & & & & & & & & A & -I_M & -I_M & -I_M & +I_M & &
\end{array}\right]
$$

A submatriz I_k representa a matriz identidade de ordem k e a submatriz A corresponde à matriz de armazenamento dos tempos de produção.

Para a aplicação do método de resolução, exposto no capítulo anterior, devemos "escolher" as colunas que comporão a matriz básica, formando dois conjuntos com os índices das mesmas: o conjunto I, com os índices das colunas chaves da matriz básica, e o conjunto L, com os índices das colunas não chaves da matriz básica. Teremos ainda um outro conjunto, N, que armazena os índices das colunas da matriz não básica.

A escolha das colunas que comporão a matriz básica deverão ser tais que as submatrizes A^{I_i} e $\tilde{A}_o^L$ (vide estrutura da matriz básica) deverão ser não singulares.

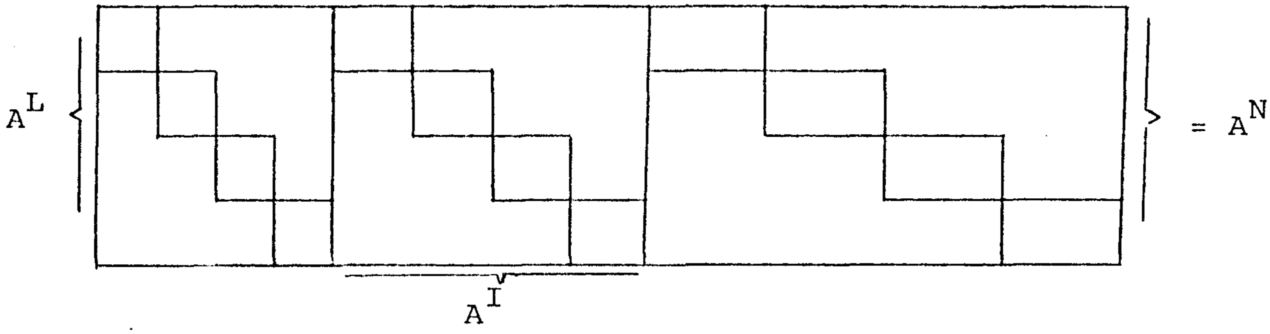
Como exemplo, uma estrutura que satisfaz as condições necessárias para aplicação do método, é a seguinte:

$$
\begin{array}{l|l|l|l}
\hline
I_N\ I_N\ I_N & & I_N & \\
I_N\ I_N & & I_N \qquad\qquad\qquad -I_N & \Bigg\} = A_O^N \\
I_N & & I_N \qquad\qquad\qquad\qquad -I_N & \\
\hline
A & I_M & -I_M\ -I_M\ -I_M & \\
 & \ \ I_M & \qquad A\ -I_M\ -I_M\ -I_M & \Bigg\} = A^N \\
\ \ A & \quad I_M & \qquad\qquad\qquad -I_M\ -I_M\ -I_M & \\
\qquad A & \qquad I_M & \qquad\qquad\qquad\qquad -I_M\ -I_M\ -I_M & \\
\hline
 & \underbrace{\qquad\quad}_{A^I} & &
\end{array}
$$

Como já pudemos observar esta estrutura apresenta grande esparsidade, em vista disso torna-se interessante a utilização de recursos de programação em busca de uma melhor forma de armazenamento.

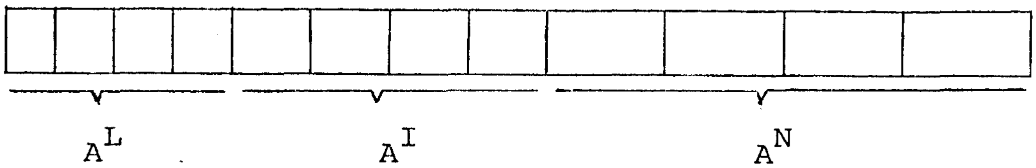
Armazenamento das matrizes A^L, A^I e A^N

Em qualquer estágio de processamento do método de resolução, é possível rearranjarmos as colunas destas matrizes, para melhor visualizarmos a seguinte estrutura.



Qualquer que seja a coluna que fixarmos, verifica-se que somente uma de suas quatro partições pode apresentar elementos não nulos.

Esta propriedade é suficiente para tomarmos uma primeira medida no sentido de economizar espaço de armazenamento: utilizar o tamanho de uma partição para armazenar tais colunas:



Por conveniência utilizaremos submatrizes A^{I_i} para indi - car que tal submatriz é característica da partição i. Porém, o mesmo não será aplicado às matrizes A^L e A^N, ou seja, é preferível abolir o uso de submatrizes A^{L_i} e A^{N_i}, tratando as colunas dos conjuntos L e N da mesma forma.

Porém isto nos obriga definir conjuntos L_i e N_i para armazenar os índices das colunas dos conjuntos L e N pertencentes à partição i.

Desta forma de tratamento, decorre que as demais estrutu -

ras obedecerão também à caracterização de suas colunas para as partições definidas.

Uma segunda medida para economizar espaço de armazenamento pode ser tomada ao observarmos que em uma determinada coluna podemos ter a mesma representação de uma coluna da matriz dos tempos de produção, ou a representação de uma coluna da matriz identidade (positiva ou negativa), ou ainda uma coluna nula.

Portanto podemos armazenar uma coluna utilizando a seguinte estrutura:

1		CÓDIGO
2		ÍNDICE
3		VALOR

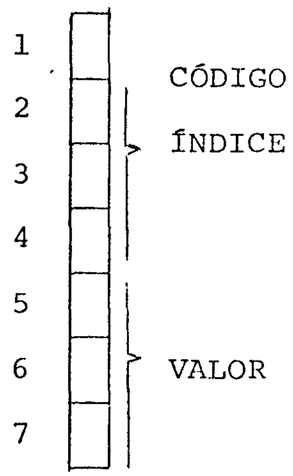
O código estabelecido na 1ª linha pode assumir os valores 0 ou 1: se CÓDIGO = 0 significa que a coluna em questão é uma coluna da matriz identidade (positiva ou negativa) ou é uma coluna nula. Para este caso, ÍNDICE indica a linha onde está armazenado o elemento não nulo da matriz identidade ou um zero se a coluna for nula. A informação armazenada em VALOR representa o elemento de interesse identificado pelo ÍNDICE.

Se CÓDIGO = 1, consideramos apenas a informação da linha 2, pois significa que estamos diante de uma coluna da matriz dos tempos de produção, identificada por ÍNDICE.

Armazenamento das matrizes A_o^L, A_o^I, A_o^N

Observamos que para uma dada coluna de qualquer uma destas matrizes podemos ter 0, 1, 2 ou no máximo 3 elementos não nulos.

Desta forma, decidimos armazenar estas matrizes por colunas, da seguinte forma:



O código estabelecido na 1ª linha pode assumir os valores 0 (coluna nula) ou 1 (coluna com até 3 elementos não nulos).

Quando CÓDIGO = 1 passamos a considerar as informações das demais linhas: as linhas 2, 3, 4 informam o índice das linhas que apresentam elementos não nulos, cujos valores estão armazenados nas linhas 5, 6 e 7. Quando a coluna apresentar 1 ou 2 elementos não nulos, os índices restantes podem apontar para elementos nulos.

Quando CÓDIGO = 0 as demais informações podem ser ignoradas, pois trata-se de uma coluna nula.

Armazenamento da Matriz dos Tempos de Produção

Por apresentar grande esparsidade decidimos por armazenar esta matriz em formato linearizado. Para tornar isto possível fizemos uso de três vetores:

MAT = armazena os elementos não nulos da matriz dos tempos de produção (por colunas).

POS = armazena os índices das linhas dos elementos não nulos da matriz dos tempos de produção, armazenados em MAT.

TAM = armazena os índices dos elementos do vetor MAT que são os primeiros elementos não nulos de cada coluna da matriz dos tempos de produção.

Se desejarmos saber quantos elementos não nulos ocorrem na coluna J da matriz dos tempos de produção, basta calcularmos:
TAM (J+1) - TAM (J)

Isto nos obriga a definir uma posição adicional neste vetor para que seja possível calcular a quantidade de elementos não nulos existente na última coluna da matriz dos tempos de produção.

Armazenamento de Outras Estruturas

As demais estruturas, próprias para o método de resolução, foram tratadas da forma tradicional, porém procuramos manter a compatibilidade com o armazenamento das matrizes anteriormente estudadas. Portanto a recuperação de elementos das demais estruturas, será feita através dos conjuntos de índices que caracterizam as partições existentes.

5.3 O PROGRAMA

Tendo em vista a metodologia e a forma de armazenamento anteriormente expostas, procuramos manter a notação dos capítulos anteriores e apresentar o programa em forma de algorítmo sintético para facilitar seu entendimento.

Obviamente este algorítmo aplica-se ao caso particular em que nos fixamos para estudo, porém acreditamos que com pequenas alterações pode-se dar a ele aspectos que o torne aplicável em outras situações.

O algorítmo que a seguir expomos não se prende a qualquer regra mais rígida de semântica, nem tampouco a qualquer linguagem de programação. Utiliza, sim, um vocabulário que nos parece apropriado para a finalidade a que se propõe.

Os passos assinalados por um asterístico, passarão por um refinamento, ou seja, serão posteriormente decompostos em níveis de detalhe que se caracterizam por conter instruções mais esclarecedoras.

Além das etapas de inicialização (A1 até A10) e apresentação da solução (A67), o algorítmo reune ainda duas etapas (Etapa 1 e Etapa 2) que interagem conforme a estratégia do método de resolução.

Na Etapa 1 (A11 até A33) procura-se colocar as variáveis não chaves em condições que satisfaçam o teste de otimalidade.

Quando todas as variáveis chaves satisfizerem esta condição passa-se, então, à Etapa 2 (A34 até A66).

A Etapa 2 procura colocar as variáveis chaves em condições que satisfaçam a otimalidade. Para tanto, nesta etapa pode - se lançar mão de dois recursos: a troca de variáveis chaves com não chaves (caso 1: A41 até A45) ou a troca de variáveis chaves diretamente com não básicas (caso 2: A47 até A63).

Se as variáveis chaves satisfizerem a condição de otimalidade sem que para isto tenha havido alterações no conjunto de variáveis não chaves, então teremos em mãos a solução do problema . Caso contrário retornamos à Etapa 1 e prosseguimos com o algorítmo conforme o já exposto.

Em seguida apresentamos o algorítmo para solução do "problema de programação da produção com horizonte de quatro meses " baseado no "método dual-simplex para problemas de programação linear com estrutura bloco-angular e variáveis canalizadas".

A1.* Entrada dos dados

A2.* Cálculo de $(A^{I_h})^{-1}$; $h = 1,\dots,4$

A3.* Cálculo de $\hat{A}^L$

A4.* Cálculo de $(\tilde{A}^L_o)^{-1}$

A5.* Cálculo de $\hat{C}^L$

A6.* Cálculo de $\hat{C}^I$

A7.* Cálculo de $\hat{C}^N$

A8.* Cálculo de X^N

A9.* Cálculo de X^L

A10.* Cálculo de X^I

A11. FLAG $\leftarrow$ "FALSE"

A12. para $1 \leq h \leq 4$ execute até o passo A32 em seguida vá para o passo A33

A13. para $r \in L_h$ execute até o passo A31 em seguida vá para o passo A32

A14. Se $d_r^1 \leq X_r \leq d_r^2$ vá para o passo A31

A15. FLAG $\leftarrow$ "TRUE"

A16.* Cálculo de $(\hat{A}_r^N)$

A17. Se $X_r < d_r^1$ vá para o passo A21

A18. $\Delta X_r \leftarrow X_r - d_r^2$

A19. $T_s = \text{Max} \{ T_j = \hat{C}^j / (\hat{A}^N)_r^j \; ; \; j \in N \text{ e } T_j < 0 \}$

A20. vá para o passo A23

A21. $\Delta X_r \leftarrow X_r - d_r^1$

A22. $T_s = \text{Min} \{ T_j = \hat{C}^j / (\hat{A}^N)_r^j \; ; \; j \in N \text{ e } T_j > 0 \}$

A23. $\hat{C}^j \leftarrow \hat{C}^j - T_s \, (\hat{A}^N)_r^j \; ; \; j \in N$

A24. $\hat{C}^r \leftarrow -T_s$

A25.* Cálculo de $(\hat{A}^N)^s$

A26.* atualização de X^L

A27.* atualização de X^I

A28. $X_s \leftarrow X_s + X_r / (\hat{A}^N)_r^s$

A29.* atualização de $(\tilde{A}_o^L)^{-1}$

A30. atualização dos conjuntos L e N

A31. continue o estabelecido pelo passo A13

A32. continue o estabelecido pelo passo A12

A33. Se FLAG = "TRUE" vá para o passo A11

A34. para $1 \leq h \leq 4$ execute até o passo A65 em seguida vá para o passo A66

A35. para $r \in I_h$ execute até o passo A64 em seguida vá para o passo A65

A36. Se $d_r^1 \leq X_r \leq d_r^2$ vá para o passo A64

A37. para $s \in L_h$ execute até o passo A39 em seguida vá para o passo A40

A38. Se $(\hat{A}^L)_r^s \neq 0$ vá para o passo A41

A39. continue o estabelecido pelo passo A37

A40. vá para o passo A47

A41. FLAG $\leftarrow$ "TRUE"

A42.* atualização de $(\tilde{A}_O^L)^{-1}$

A43.* atualização de $(A^{I_h})^{-1}$

A44.* atualização de $\hat{A}^L$

A45. atualização dos conjuntos I e L

A46. vá para o passo A64

A47. FLAG $\leftarrow$ "TRUE"

A48.* cálculo de $(\hat{A}^N)_r$

A49. Se $d_r^1 > X_r$ vá para o passo A53

A50. $\Delta X_r \leftarrow X_r - d_r^2$

A51. $T_s = \text{Max} \{ T_j = \hat{C}^j / (\hat{A}^N)_r^j \; ; \; j \in N \text{ e } T_j < 0 \}$

A52. vá para o passo A55

A53. $\Delta X_r \leftarrow X_r - d_r^1$

A54. $T_s = \mathrm{Min}\{ T_j = \hat{C}^j / (\hat{A}^N)^j_r \; ; \; j \in N \text{ e } T_j > 0 \}$

A55. $\hat{C}^j \leftarrow \hat{C}^j - T_s (\hat{A}^N)^j_r \; ; \; j \in N$

A56. $\hat{C}^r \leftarrow -T_s$

A57.* cálculo de $(\hat{A}^N)^s$

A58.* atualização de X^L

A59.* atualização de X^I

A60. $X_s \leftarrow X_s + \Delta X_r / (\hat{A}^N)^s_r$

A61.* atualização de $(A^{I_h})^{-1}$

A62.* atualização de $\hat{A}^L$

A63. atualização dos conjuntos I e N

A64. continue o estabelecido pelo passo A35

A65. continue o estabelecido pelo passo A34

A66. Se FLAG = "TRUE" vá para o passo A11

A67.* impressão dos resultados

A68. FIM

Passaremos, agora, à descrição das instruções do algorítmo que necessitam de maiores explicações.

A1. Entrada de dados:

Os dados necessários para o processamento das instruções do algorítmo são:

L_h, I_h, N_h = conjuntos dos índices das colunas não chaves, chaves e não básicas, respectivamente (h=1,...,4).

LX, IX, NX = conjuntos das variáveis do problema original relacionadas com as colunas não chaves, chaves e não básicas, respectivamente.

$A_o^L,\ A_o^{I_h},\ A_o^N$ = fornecidas por colunas, conforme o armazenamento exposto anteriormente.

$A^L,\ A^{I_h},\ A^N$ = fornecidas por colunas, conforme o armazenamento exposto anteriormente.

$c^L,\ c^{I_h},\ c^N$ = vetores dos custos das variáveis relacionadas com as colunas não chaves, chaves e não básicas, respectivamente.

$d_L^k,\ d_{I_h}^k,\ d_N^k$ = limitantes inferior (k=1) e superior (k=2) para os valores das variáveis relacionadas com as variáveis não chaves, chaves e não básicas, respectivamente.

$b_o,\ b_h$ = "R.H.S" das restrições do problema exposto (h=1,...,4)

TAM,POS,MAT = definem a matriz dos tempos de produção, conforme o armazenamento exposto anteriormente.

$A_2,\ A_3,\ A_4$: cálculo de $(A^{I_h})^{-1},\ \tilde{A}^L,\ (\tilde{A}_o^L)^{-1}$

Estes cálculos poderão ser simplificados pela escolha conveniente das submatrizes A^{I_h} e A_o^L.

Considerando o exemplo exposto em 5.2, podemos fazer:

$$A^{I_h} = I_m \text{ para } h = 1,\dots,4$$

$$A_o^L = \begin{pmatrix} I_n & I_n & I_n \\ I_n & I_n & o \\ I_n & o & o \end{pmatrix}$$

com esta escolha, teremos:

$$(A^{I_h})^{-1} = A^{I_h} \quad \text{para } h = 1,\ldots,4$$

$$\hat{A}^L = (A^I)^{-1} \cdot A^L = A^L$$

$$(\tilde{A}_o^L)^{-1} = (A_o^L - A_o^I \hat{A}^L)^{-1}$$

como $A_o^I = 0$, segue-se que $(\tilde{A}_o^L)^{-1} = (A_o^L)^{-1} = \begin{pmatrix} 0 & 0 & I_n \\ 0 & I_n & -I_n \\ I_n & -I_n & 0 \end{pmatrix}$

A_5. cálculo de $\hat{C}^L$

$$\hat{C}^L = (C^L - C^I \hat{A}^L)\ (\tilde{A}_o^L)^{-1}$$

A estrutura bloco-angular nos permite simplificar este cálcu lo se considerarmos cada partição h:

$$\hat{C}^{L_h} = (C^{L_h} - C^{I_h} \hat{A}^{L_h})\ (\tilde{A}_o^L)^{-1} \quad \text{para } h = 1,\ldots,4$$

A6. cálculo de $\hat{C}^I$

$$\hat{C}^I = (C^I - \hat{C}^L A_o^I)\ (A^I)^{-1}$$

considerando cada partição h podemos calcular

$$\hat{C}^{I_h} = (C^{I_h} - \hat{C}^L A_o^{I_h})\ (A^{I_h})^{-1} \quad \text{para } h = 1,\ldots,4$$

considerando ainda a forma de armazenamento de cada coluna de A_o^I teremos algumas facilidades adicionais.

A7. cálculo de $\hat{C}^N$

$$\hat{C}^N = C^N - \hat{C}^B \mathcal{A}^N = C^N - (\hat{C}^L, \hat{C}^I) \begin{pmatrix} A_o^N \\ \hline A^N \end{pmatrix}$$

considerando a forma de armazenamento das matrizes A_o^N e A^N, este cálculo fica simples se o fizermos separadamente para cada $j \in N$:

$$(\hat{C}^N)^j = (C^N)^j - \tilde{C}^{L_h} (A_o^{N_h})^j - \tilde{C}^{I_h} (A^{N_h})^j \; ; \; j \in N_h, \; h=1,\ldots,4$$

A_8. cálculo de X^N

para cada $j \in N$: $X_j = d_j^1$ se $(\hat{C}^N)^j \geq 0$

$X_j = d_j^2$ se $(\hat{C}^N)^j < 0$

A_9. cálculo de X^L

$$X^L = (\tilde{A}_o^L)^{-1} (b^L - A_o^N X^N - A_o^I (A^I)^{-1} (b^I - A^N X^N))$$

Por conveniência este cálculo pode ser feito por partes:

$A_{9.1}^*$ baux $= b^I - A^N X^N$

$A_{9.2}^*$ Caux $= A_o^I (A^I)^{-1}$ baux

$A_{9.3}^*$ Caux $= b^L - A_o^N X^N -$ Caux

$A_{9.4}^*$ $X^L = (\tilde{A}_o^L)^{-1}$ Caux

Cada um destes refinamentos tem seu cálculo simplificado pela estrutura bloco-angular e a forma de armazenamento.

$A_{9.1}$ cálculo de baux $= b^I - A^N X^N$

considerando-se cada partição h e a forma de armazenamento de A^N, temos:

$$\text{baux}^h = b^{I_h} - A^{N_h} \cdot X^{N_h} \; , \; h = 1,\ldots,4.$$

$A_{9.2}$ cálculo de $Caux = A_o^I \, (A^I)^{-1}$ baux

Também este cálculo pode ser simplificado se conside - rarmos cada partição em particular:

$$Caux^h = A_o^{I_h} \, (A^{I_h})^{-1} \; baux^h, \; h = 1,\ldots,4$$

$A_{9.3}$ cálculo de $Caux = b^L - A_o^N \, X^N - Caux$

$$Caux^h = b^{L_h} - A_o^{N_h} - Caux^h, \; h = 1,\ldots,4$$

$A_{9.4}$ cálculo de $X^L = (\tilde{A}_o^L)^{-1}$ Caux

$$X^{L_h} = (\tilde{A}_o^L)^{-1} \; Caux^h, \; h = 1,\ldots,4.$$

$A_{10.}$ cálculo de X^I

$$X^I = (A^I)^{-1} \, (b^I - A^N \, X^N - A^L \, X^L)$$

Para maior facilidade, este cálculo pode ser particionado . Considerando-se que já temos calculado $baux = b^I - A^N \, X^N$ (vide $A_{9.1}$), temos:

$A^*_{10.1}$ $Caux = baux - A^L \, X^L$

$A^*_{10.2}$ $X^I = (A^I)^{-1} \, . \, Caux$

Estes refinamentos têm seu cálculo simplificado a partir de considerações da estrutura bloco-angular e da forma de armazenamento.

$A_{10.1}$ cálculo de $Caux = baux - A^L \, X^L$

considerando-se cada partição h, em particular, temos:

$$Caux^h = baux^h - A^{L_h} \, X^{L_h} \; ; \; h = 1,\ldots,4.$$

$A_{10.2}$ cálculo de $X^I = (A^I)^{-1} . Caux$

Também aqui temos facilidades se considerarmos cada partição h:

$$X^{I_h} = (A^{I_h})^{-1} . Caux^h, \quad h = 1,\ldots,4.$$

$A_{16.}$ cálculo de $(\hat{A}^N)_r$

$$(\hat{A}^N)_r = (A^B)_r^{-1} . A^N$$

Neste cálculo, $(A^B)_r^{-1}$ é a linha r da inversa da matriz básica e A^N é a matriz não básica.

Como estamos na etapa 1, temos que $r \leq m_o$ (onde m_o é a dimensão da submatriz A_o^L da matriz básica).

Considerando a estrutura da inversa da matriz básica, podemos escrever:

$$(\hat{A}^N)_r = (\tilde{A}_o^L)_r^{-1} . (A_o^N - A_o^I (A^I)^{-1} A^N)$$

Considerando cada elemento desta linha, este cálculo fica simplificado:

$$(\hat{A}^N)_r^j = (\tilde{A}_o^L)_r^{-1} . ((A_o^{N_h})^j - A_o^{I_h} (A^{I_h})^{-1} (A^{N_h})^j)$$

para cada $j \in N_h$, $h = 1,\ldots,4$.

É claro que o cálculo de $A_o^{I_h} (A^{I_h})^{-1}$ pode ser feito separadamente, ficando constante para cada partição h.

$A_{25.}$ cálculo de $(\hat{A}^N)^s$

$$(\hat{A}^N)^s = (A^B)^{-1} (A^N)^s = (A^B)^{-1} \begin{pmatrix} (A_o^N)^s \\ \hdashline (A^N)^s \end{pmatrix} = \begin{pmatrix} (\bar{A}_o^N)^s \\ \hdashline (\bar{A}^N)^s \end{pmatrix}$$

Como sabemos préviamente que $s \in N_h$, ficamos simplesmente com:

$$(\bar{A}_o^N)^s = (\tilde{A}_o^L)^{-1} \cdot ((A_o^{N_h})^s - A_o^{I_h}\,(A^{I_h})^{-1}\,(A^{N_h})^s)$$

$$(\bar{A}^N)^s = (A^{I_h})^{-1} \cdot ((A^{N_h})^s - A^{L_h}\,(\bar{A}_o^N)^s)$$

A_{26}. Atualização de X^L

Os valores das variáveis não chaves podem ser recalculados individualmente:

$$(X^L)_j \leftarrow (X^L)_j + \frac{(\bar{A}_o^N)_j^s}{(\hat{A}^N)_r^s} \cdot \Delta X_r \ , \ j \in L.$$

A_{27}. Atualização de X^I

Para recalcularmos os valores das variáveis chaves, podemos fazê-lo individualmente, desde que fique caracterizada a partição a qual a variával pertence:

$$(X^{I_h})_j \leftarrow (X^{I_h})_j + \frac{(\bar{A}^{N_h})_j^s}{(\hat{A}^N)_r^s} \cdot \Delta X_r \ ; \ j \in I_h \ , \ h = 1,\ldots,4$$

A_{29}. Atualização de $(\tilde{A}_o^L)^{-1}$

De acordo com o método de resolução exposto, podemos estabelecer os seguintes cálculos:

Os multiplicadores α_j , $j = 1,\ldots,m_o$ são dados por

$$\alpha_j = \begin{cases} -(\bar{A}^N_o)^s_j \,/\, (\bar{A}^N_o)^s_r & \text{para } j \neq r \\ 1 \,/\, (\bar{A}^N_o)^s_r & \text{para } j = r \end{cases}$$

Tendo estes valores, os elementos na nova matriz inversa da base de trabalho são dados por

$$((\tilde{A}^L_o)^{-1})^j_i = \begin{cases} ((\tilde{A}^L_o)^{-1})^j_i + \alpha_i \; ((\tilde{A}^L_o)^{-1})^j_r & \text{para } i \neq r \\ \alpha_r \; ((\tilde{A}^L_o)^{-1})^j_r & \text{para } i = r \end{cases}$$

$A_{42.}$ Atualização de $(\tilde{A}^L_o)^{-1}$

Neste passo do algorítmo devemos atualizar a inversa da matriz base de trabalho por termos aplicado o caso 1 da etapa 2.

De acordo com o método exposto a atualização desta matriz resume-se em recalcular a linha s desta matriz, da seguinte forma:

$$((\tilde{A}^L_o)^{-1})^j_s = -\,(\hat{A}^L)_r \,.\, ((\tilde{A}^L_o)^{-1})^j_s \quad ; \; j = 1,\ldots,m_o$$

$A_{43.}$ Atualização de $(A^{I_h})^{-1}$

Sabendo-se que r pertence à partição h, para a atualização desta matriz podemos estabelecer os seguintes cálculos:

Os multiplicadores α_J , $J = 1,\ldots,m_h$ são dados por

$$\alpha_j = \begin{cases} -(\hat{A}^L)^s_j \,/\, (\hat{A}^L)^s_r & \text{para } j \neq r \\ 1 \,/\, (\hat{A}^L)^s_r & \text{para } j = r \end{cases}$$

Tendo estes valores, os elementos da nova matriz $(A^{I_h})^{-1}$ são dados por

$$((A^{I_h})^{-1})_i^j = \begin{cases} ((A^{I_h})^{-1})_i^j + \alpha_i \; ((A^{I_h})^{-1})_r^j & \text{para } i \neq r \\ \alpha_r ((A^{I_h})^{-1})_r^j & \text{para } i = r \end{cases}$$

A_{44}. Atualização de $\hat{A}^L$

Como a matriz $(A^I)^{-1}$ foi atualizada somente a nível da partição h, também esta matriz deverá ser atualizada desta forma:

$$\hat{A}^{L_h} = A^{L_h} \; (A^{I_h})^{-1}$$

A_{48}. cálculo de $(\hat{A}^N)_r$

O cálculo desta linha deve ser feito aproveitando-se o fato de estarmos no caso 2 da etapa 2, ou seja, utilizar a informação que $(\hat{A}^L)_r = 0$ e r pertence a partição h.
Nestas condições, temos que

$$(\hat{A}^N)_r = (A^{I_h})_r^{-1} \; . \; (A^{N_h})$$

A_{57}, A_{58}, A_{59}. cálculo de $(\hat{A}^N)^s$ e atualização de X^L e X^I

O referido cálculo e atualizações são feitos de forma semelhante aos passos A25, A26 e A27 respectivamente, podendo, portanto, fazerem parte de uma rotina.

A_{61}. Atualização de $(A^{I_h})^{-1}$

Sabendo-se que r pertence à partição h, para a atualização da referida matriz calculamos primeiramente os multiplicadores α_j ; $j = 1,\ldots,m_h$

$$\alpha_j = \begin{cases} -(\hat{A}^N)^s_j \;/\; (\hat{A}^N)^s_r & \text{para } j \neq r \\ 1 \;/\; (\hat{A}^N)^s_r & \text{para } j = r \end{cases}$$

Tendo estes valores, os elementos da nova matriz $(A^{I_h})^{-1}$ são dados por

$$((A^{I_h})^{-1})^j_i = \begin{cases} ((A^{I_h})^{-1})^j_i + \alpha_i \; ((A^{I_h})^{-1})^j_r & \text{para } i \neq r \\ \alpha_r \; ((A^{I_h})^{-1})^j_r & \text{para } i = r \end{cases}$$

A_{62}. Atualização de $\hat{A}^L$

É feita de forma semelhante ao passo A_{44}.

A_{67}. Impressão dos Resultados

Os dados resultantes do processamento das instruções estão armazenados nas variáveis X^L, X^I e X^N, que representam os valores das variáveis do problema proposto. Para associarmos estes valores às variáveis do problema original utilizamos os conjuntos LX, IX e NX.

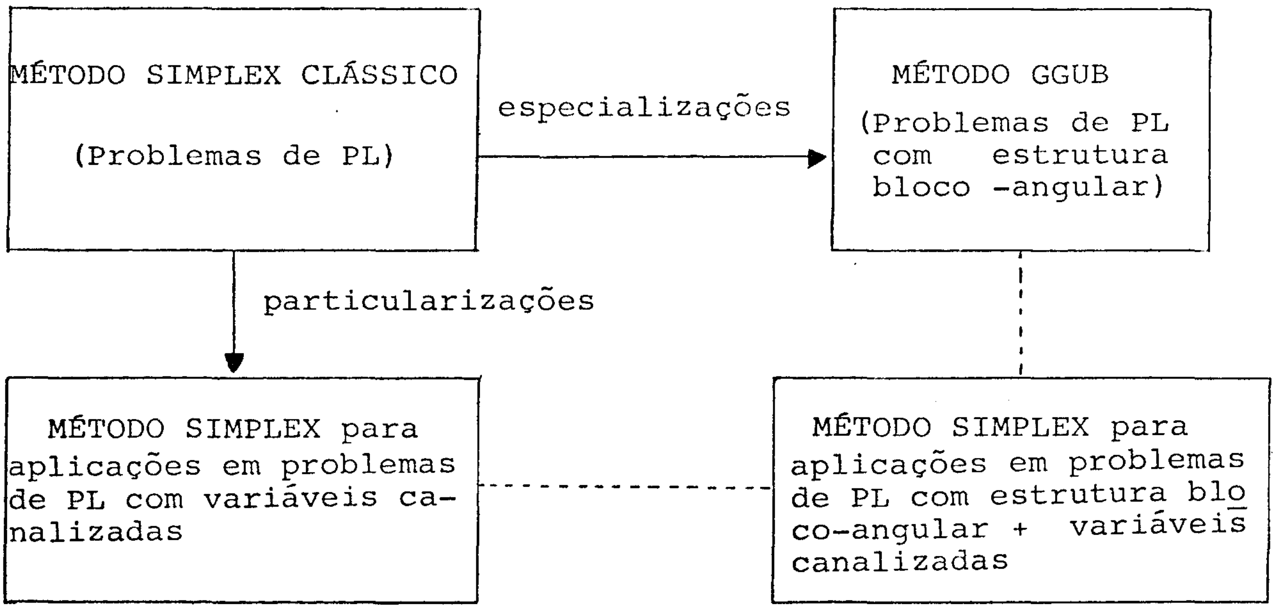
Desta forma, a variável do problema original identificada por $(LX)_i$ terá valor $(X^L)_i$.

CAPÍTULO VI

CONCLUSÕES

Através do histórico da Programação Linear, podemos observar que os métodos aplicáveis às mais diversas situações, surgiram como extenções do Método Simplex Clássico, sob formas de especializações e particularizações.

O método que neste trabalho apresentamos, com o objetivo de resolver problemas de Programação Linear que apresentem estrutura bloco-angular e variáveis canalizadas, é resultante da combinação de uma especialização e de uma particularização do Método Simplex Clássico. A primeira se aplica à problemas de Programação Linear que apresentem estrutura bloco-angular e a segunda se aplica à problemas de Programação Linear que apresentem variáveis canalizadas:



Para que esta combinação se tornasse possível fizemos uso do método G.G.U.B., que é aplicável à problemas de Programação Linear que apresentem estrutura bloco-angular, devendo ser ressaltado o fato de termos usado a forma dual.

Com a finalidade de prover a compatibilidade na forma de tratamento, apresentamos um Método Dual-Simplex para problemas de Programação Linear com variáveis canalizadas.

A alternativa da forma dual, para concepção do método proposto, foi escolhida com vistas voltadas às facilidades computacionais e pelo interesse de se seguir um caminho não usual.

O desenvolvimento deste método vem possibilitar o uso mais racional do computador para problemas de Programação Linear de grande porte que apresentem as características que aqui enfocamos.

Os "ganhos computacionais" se devem ao fato do método se incluir na categoria de "simplex revisado" e ao tratamento específico dado aos conceitos básicos como solução inicial, pivoteamento e custos relativos.

Uma análise mais cuidadosa do problema prático poderá aumentar ainda mais estes ganhos computacionais. Como exemplo disto podemos tomar o contido no Capítulo V, onde através de uma pequena análise da estrutura matemática do problema de programação da produção com horizonte de quatro meses, pudemos diminuir consideravelmente o espaço necessário para armazenagem das informações e aumentar a rapidez nos cálculos intermediários.

Uma possível continuação deste trabalho seria a comparação do método exposto com outros métodos aplicáveis à problemas de Programação Linear que apresentem as características de estrutura bloco-angular e variáveis canalizadas. Esta comparação poderia se traduzir na forma mais concreta de se obter as medidas de eficiência computacional.

BIBLIOGRAFIA

[1] RIBEIRO, R.V. - "Estudos em Programação Linear", Tese de Doutorado, FEC, UNICAMP, 1980.

[2] LASDON, L.S. - "Optimization Theory for Large Systems", Mac Millan, N.York, 1970.

[3] DANTZIG, G.B., VAN SLYKE, R.M. - "Generalized Upper Bounding Techniques for Linear Programming". Operations Research Center Tech. Repts. ORC 64-17 e ORC 64-18, University of California, California, 1964 e 1965.

[4] KAUL, R.N. - "An Extension of Generalized Upper Bounded Techniques for Linear Programming". Operations Research Center Tech. Rept 65-27, University of California, California, 1965.

[5] BAZARAA, M.S., JARVIS, J.J. - "Linear Programming and Network Flows", Wiley, N.York, 1979.

[6] ROSEN, J.B. - "Primal Partition Programming for Block Diagonal Matrices", Numerische Mathematik, 6, 250-260, 1964.

APÊNDICE

CÁLCULO DA INVERSA DE MATRIZ BÁSICA

No Capítulo II, com base nos resultados do Teorema 1, reescrevemos o problema bloco-angular para caracterizarmos dois sistemas: o Sistema de Restrições Globais (reunindo a função objetivo e as restrições de acoplamento) e o Sistema de Restrições Locais.

Em seguida efetuamos o pivoteamento interno no Sistema de Restrições Locais, colocando-o sob a forma canônica (RLC), para depois efetuarmos o pivoteamento externo no Sistema de Restrições Globais, definindo, então, o Problema Reduzido (PR):

$$\left[\begin{array}{c:c:c} \tilde{C}^L & 0 & \tilde{C}^N \\ \hdashline \tilde{A}^L_o & 0 & \tilde{A}^N_o \end{array}\right] \left[\begin{array}{c} X^L \\ \hdashline X^I \\ \hdashline X^N \end{array}\right] = \left[\begin{array}{c} \tilde{Z} \\ \hdashline \tilde{b}_o \end{array}\right] \qquad \text{(PR)}$$

$$\left[\begin{array}{c:c:c} \hat{A}^L & E & \hat{A}^N \end{array}\right] \left[\begin{array}{c} X^L \\ \hdashline X^I \\ \hdashline X^N \end{array}\right] = \left[\begin{array}{c} \hat{b} \end{array}\right] \qquad \text{(RLC)}$$

A submatriz E representa uma identidade de ordem $\sum_{i=1}^{q} m_i$.

Como $\tilde{A}^L_o$ é uma matriz não singular, podemos efetuar um pivoteamento interno em (PR) e um pivoteamento externo em (RLC)com esta matriz, que tratamos por "base de trabalho".

Após estas transformações, as restrições do problema bloco-angular assumiram a forma:

$$\left[\begin{array}{c:c:c} E_o & O & (\tilde{A}_o^L)^{-1}\tilde{A}_o^N \\ \hdashline O & E & \hat{A}^N - \hat{A}^L(\tilde{A}_o^L)^{-1}\tilde{A}_o^N \end{array}\right] \left[\begin{array}{c} X^L \\ \hdashline X^I \\ \hdashline X^N \end{array}\right] = \left[\begin{array}{c} (\tilde{A}_o^L)^{-1}\tilde{b}_o \\ \hdashline \hat{b} - \hat{A}^L(\tilde{A}_o^L)^{-1}\tilde{b}_o \end{array}\right]$$

A submatriz E_o representa uma identidade de ordem m_o.

A nova disposição das restrições nos permite calcular facilmente os valores das variáveis básicas:

$$X^L = (\tilde{A}_o^L)^{-1}\tilde{b}_o - (\tilde{A}_o^L)^{-1}\tilde{A}_o^N X^N$$

$$= (\tilde{A}_o^L)^{-1}(b_o - A_o^I(A^I)^{-1}b) - (\tilde{A}_o^L)^{-1}(A_o^N - A_o^I(A^I)^{-1}A^N)X^N$$

$$= [(\tilde{A}_o^L)^{-1} \;\vdots\; -(\tilde{A}_o^L)^{-1}A_o^I(A^I)^{-1}] \left[\begin{array}{c} b_o \\ \hdashline b \end{array}\right] - [(\tilde{A}_o^L)^{-1} \;\vdots\; -(\tilde{A}_o^L)^{-1}A_o^I(A^I)^{-1}] \left[\begin{array}{c} A_o^N \\ \hdashline A^N \end{array}\right] X^N$$

$$X^I = \hat{b} - \hat{A}^L(\tilde{A}_o^L)^{-1}\tilde{b}_o - (\hat{A}^N - \hat{A}^L(\tilde{A}_o^L)^{-1}\tilde{A}_o^N)X^N$$

$$= (A^I)^{-1}b - \hat{A}^L(\tilde{A}_o^L)^{-1}(b_o - A_o^I(A^I)^{-1}b) - ((A^I)^{-1}A^N - \hat{A}^L(\tilde{A}_o^L)^{-1}(b_o - A_o^I(A^I)^{-1}b))X^N$$

$$= [-\hat{A}^L(\tilde{A}_o^L)^{-1} \;\vdots\; (A^I)^{-1} + \hat{A}^L(\tilde{A}_o^L)^{-1}A_o^I(A^I)^{-1}] \left[\begin{array}{c} b_o \\ \hdashline b \end{array}\right] - [-\hat{A}^L(\tilde{A}_o^L)^{-1} \;\vdots\; (A^I)^{-1} + \hat{A}^L(\tilde{A}_o^L)^{-1}A_o^I(A^I)^{-1}] \left[\begin{array}{c} A_o^N \\ \hdashline A^N \end{array}\right] X^N$$

Porém, da estrutura original, temos que:

$$\begin{bmatrix} x^L \\ \hdashline x^I \end{bmatrix} = (A^B)^{-1} \cdot \begin{bmatrix} b_o \\ \hdashline b \end{bmatrix} - (A^B)^{-1} \cdot A^N \cdot x^N$$

Desta forma, deduzimos que a inversa de matriz básica, $(A^B)^{-1}$, é dada por:

$$(A^B)^{-1} = \left[\begin{array}{c:c} (\tilde{A}_o^L)^{-1} & -(\tilde{A}_o^L)^{-1} A_o^I (A^I)^{-1} \\ \hdashline -(A^I)^{-1} A^L (\tilde{A}_o^L)^{-1} & (A^I)^{-1} + (A^I)^{-1} A^L (\tilde{A}_o^L)^{-1} A_o^I (A^I)^{-1} \end{array}\right]$$

Assim, podemos verificar que não precisamos trabalhar diretamente com a inversa da matriz básica, pois podemos obtê-la facilmente a partir das inversas de $\tilde{A}_o^L$ e de A^{I_i} ; $i=1,2,\ldots,q$, de dimensões bem menores.