UM MÉTODO HEURÍSTICO PARA A LOCALIZAÇÃO E DIMENSIONAMENTO DE ARMAZÉNS EM SISTEMAS DE GRANDE PORTE CONSIDERANDO ECONOMIAS DE ESCALA

Cesar Danuzio Baião Monterosso

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS (M.Sc.).

Aprovada por:

Cláudio Thomás Bornstein

Ronaldo C. Marinho Persiano

Nelson Maculan Filho

Ruy Neves Ribas

CIBRAZEM

RIO DE JANEIRO - BRASIL

DEZEMBRO DE 1977

AGRADECIMENTOS

Ao professor Claudio T. Bornstein, pela excelente orientação e pelo incentivo constante.

Ao professor Nelson Maculan Filho, que me apoiou e incentivou desde o início.

Ao professor Ronaldo C. Marinho Persiano, pelo seu interesse e sugestões apresentadas.

À CAPES, pelo apoio financeiro.

À Suely, pelo esforço no trabalho de datilografia.

À todos que colaboraram com sugestões e apoio

Obrigado.

SUMÁRIO

É feita uma apresentação geral do problema de localização de facilidades numa rede de distribuição. São apresentadas algumas variantes do problema e alguns métodos de solução são comentados. Em seguida, um método heurístico para a busca de uma solução de custo mínimo do problema é desenvolvido. O método é voltado para problemas de grande porte, onde economias de escala nos custos com armazéns são relevantes. As vantagens computacionais do método são mostradas e a flexibilidade na modelagem é explorada no estudo de alguns casos.

ABSTRACT

The general problem of warehouse location in a distribution network is presented. Side conditions are considered and some solution methods are reviwed. Then a heuristic method for the search of a minimum cost solution to the problem is developed. The method considers large systems, where economies of scale in warehouse costs are relevant. The computational advantages of the method are shown and flexibility in modelling is presented in some case studies.

ÍNDICE

1. - O PROBLEMA DE LOCALIZAÇÃO

1.1 - DEFINIÇÃO DO PROBLEMA

O problema básico de localização que examinaremos pode ser assim descrito:

Deseja-se construir ou ampliar um conjunto de plantas, que podem ser fábricas, centros de processamento, armazéns, etc.

Ao planejador é fornecido um conjunto finito de locais candidatos, isto é, locais préviamente selecionados como potencialmente mais atrativos para construção/ampliação de tais plantas.

É fornecido ainda um conjunto de centros de consumo, cujas demandas devem ser atendidas.

São conhecidos também os custos de transporte e os custos de contrução/ampliação de plantas, os últimos sendo em geral dados por funções não convexas.

O problema consiste em determinar:

1 - Quantos projetos de construção/ampliação devem ser executados.

2 - Em quais dos locais candidatos devem ser executados os projetos

3 - Que tamanho/capacidade deverá ter cada planta. (Este item não é considerado em problemas mais simples, onde os tamanhos são pré-fixados).

4 - O fluxo de distribuição de produtos entre plantas e centros de consumo.

O objetivo do planejador, é obter uma solução em que o custo total (custos de transporte + custos de construção/ampliação) seja mínimo.

Neste trabalho, estamos interessados em estudar um problema de localização onde as plantas são armazéns, ou seja, facilidades intermediárias entre centros de produção e centros de consumo. Nesse caso, são conhecidas também as ofertas de produtos em cada centro de produção, e os custos de transporte entre esses centros e os locais candidatos à centros de armazenagem.

Este tipo de problema surge em geral como parte de um problema maior, isto é, quando está sendo projetado um Sistema de Distribuição. Por esta razão o problema que vamos focalizar é freqüentemente tratado como "problema de distribuição" na literatura especializada.

De fato, armazéns exercem um papel importante num Sistema de Distribuição. Dentre as funções importantes de um armazem, podemos citar:

1 - Estocar produtos, para cobrir variações

na demanda e/ou produção.

2 - Receber e processar encomendas.

3 - Reduzir custos de transporte, permitindo carga mais racional dos veículos através do despacho de várias encomendas numa só viagem.

4 - Servir de base para a frota de veículos de transporte.

5 - Tornar as entregas mais rápidas, mantendo as mercadorias estocadas perto do ponto de consumo.

6 - Realizar o processamento (ou montagem) final e embalagem das mercadorias.

Este último item é particularmente relevante quando se trata de armazéns agrícolas. Estes atuam também como verdadeiros centros de processamento, executando tarefas como pesagem, lavagem e separação de grãos, secagem, ensacamento, etc.

Uma versão simples do problema de localização de armazéns pode ser descrita pelo seguinte modelo matemático(MM1):

$$\text{MIN } W = \sum_{i=1}^{\ell} \sum_{j=1}^{m} c_{ij}x_{ij} + \sum_{j=1}^{m} f_j(T_j) + \sum_{j=1}^{m} \sum_{k=1}^{n} d_{jk}z_{jk}$$

Sujeito a:

$$(1) \quad \sum_{j=1}^{m} x_{ij} = 0_i \qquad i = 1,\ldots\ldots \ell$$

$$(2) \quad \sum_{j=1}^{m} z_{jk} = D_k \qquad k = 1,\ldots n$$

$$(3) \quad \sum_{i=1}^{\ell} x_{ij} - \sum_{k=1}^{n} z_{jk} = 0 \qquad j = 1,\ldots m$$

$$x_{ij} \geq 0 \quad \forall i,j$$

$$z_{jk} \geq 0 \quad \forall j,k$$

onde:

c_{ij} - custo unitário de transporte do centro de produção $\underline{i}$ ar armazem j.

x_{ij} - quantidade de produto enviado do centro de produção $\underline{i}$ ao armazem j, por período.

$T_j = \sum_{i=1}^{\ell} x_{ij}$ - quantidade total de produto transitando pelo armazem j, por período.

$f_j(T_j)$ - função de custo do armazem no local j. A quantidade de produto transitando pelo armazem (T_j) é fator determinante do seu tamanho e portanto do custo.

O tipo de função f_j influenciará fortemente na escolha da técnica para se resolver o programa matemático.

d_{jk} - custo unitário de transporte do armazem j ao centro de consumo k.

z_{ij} - quantidade de produto enviado ao armazem j ao centro de consumo k, por período.

O_i - quantidade de produto ofertado para armazenagem no centro de produção i, por período.

D_k - quantidade de produto demandado no centro de consumo k, por período.

O período a que nos referimos acima é um espaço de tempo durante o qual o fluxo de distribuição é analizado (por ex. 1 ano), escolhido de maneira a englobar todos os padrões sazonais de escoamento (caso existam), e ao fim do qual esses padrões tendem a se repetir.

As equações (1), (2) e (3) do modelo refletem algumas hipóteses que são assumidas na maioria dos casos a saber:

. As equações do tipo (1) pressupõem que toda a oferta de produtos para armazenagem é encaminhada à rede de armazéns considerada. Além disso, os armazéns só recebem de centros de produção tratados no modelo.

. As equações do tipo (2) pressupõem que toda a demanda dos centros de consumo considerados será atendido pela rede de armazenagem em projeto. Na prática, os parâmetros D_k podem ser considerados não como a demanda do centro de consumo k, mas como a parcela da demanda desse centro a ser atendida pela rede de armazéns em projeto.

. As equações do tipo (3) são equações de conservação nos armazéns. Supõe-se que ao final de um período de análise, a quantidade contida em cada armazem é igual à que havia no início do período.

O modêlo simples que apresentamos é bastante geral. Na prática, este modêlo poderá ser consideravelmente modificado, para que sejam levadas em conta as condições particulares do problema real em estudo. Outras vezes, estas modificações são devidas à restrições impostas pelo usuário, provenientes de considerações políticas, sociais ou estratégicas.

Estas alterações no modelo serão expressas pela introdução de novas variáveis e/ou equações de restrição.

Devemos ressaltar aqui que a consideração dessas condições particulares muitas vezes se constituirá em fator determinante na escolha do método de resolução do problema. Em outras palavras, as variantes do modêlo tornarão sua estrutura mais apropriadas para esta ou aquela técnica de Pesquisa Operacional.

Mencionaremos a seguir algumas dessas condições particulares que, consideradas isoladas ou em conjunto, constituem variantes no problema de localização.

1 - Já existem armazéns instalados, que devem ser levados em conta no projeto da ampliação da rede de armazéns.

2 - Os armazéns já existentes devem ser considerados para eventual ampliação, retração ou mesmo desativação.

3 - O sistema de armazenagem opera com vários produtos, que devem ser tratados separadamente.

4 - São consideradas variações sazonais na oferta e/ou demanda.

5 - São consideradas elasticidades na oferta e/ou demanda. Como o preço é determinado pelo mercado, as ofertas e/ou demandas devem ser tratadas como variáveis aleatórias.
(Incertezas na demanda também podem ocorrer quando o tamanho dos lotes encomendados varia aleatóriamente).

6 - Existem limitações na rede de transporte, isto é, algumas vias de escoamento tem capacidade de fluxo limitada.

7 - É imposto um tamanho mínimo e/ou máximo para cada armazem aberto.

8 - O investimento em construção e/ou ampliação de armazéns é limitado.

9 - É imposto um vínculo armazem-ponto de consumo, isto é, cada ponto de consumo só pode ser suprido por um único armazém.

10 - São impostas prioridades geográficas. Por exemplo: exige-se que na solução final o local x contenha um armazem, ou exige-se que pelo menos 1 armazem seja aberto dentre os locais x, y e z.

11 - As capacidades dos armazéns não podem ser quaisquer devendo ser obrigatóriamente múltiplas de um módulo básico (ex:silos).

12 - Existem economias de escala nos custos relativos a cada armazém.

Muitas outras considerações podem surgir em casos específicos, atuando freqüentemente como fatores determinantes da abordagem a ser dada ao problema.

1.2. - CUSTOS NOS ARMAZÉNS

Já mencionamos que a forma da função de custo $f_j(T_j)$ (ver MM1) influenciará fortemente a escolha da técnica para se resolver o programa matemático.

A forma mais aceita para essa função é

$$f_j(T_j) = \begin{cases} F_j + G_j\, T_j & , \text{ se } T_j > 0 \\ 0 & , \text{ se } T_j = 0 \end{cases}$$

onde

F_j - custo fixo de construção e operação do armazém j, por período. Esta parcela do custo independe do tamanho do armazém, sendo em geral devida ao equipamento mínimo necessário, área mínima para carga/descarga, mínimo de pessoal para operar o armazém, etc.

G_j - custo marginal no armazém j. Trata-se de um custo variável de construção/operação do armazém, por período. Esta parcela depende da quantidade de produto em trânsito pelo armazém j (T_j), e portanto de sua capacidade de armazenamento.

Os custos F_j e G_j são tomados por período (exemplo: por ano) para que fiquem dimensionalmente compatíveis com os custos de transporte, que são computados por período. Nesse caso, a função $f_j(T_j)$ representaria a amortização anual do investimento em armazéns, acrescido dos custos anuais de operação.

Como $T_j = \sum_i x_{ij}$, este tipo de função de custo aparecerá na função objetivo como uma parcela da forma

$$\sum_{j=1}^{m} (F_j y_j + G_j \sum_{i=1}^{\ell} x_{ij})$$

onde y_j = 1 , caso o projeto do armazém j seja efetivado.

= 0 , caso contrário.

O uso da variável y_j sugere uma solução por programação inteira mista, que, na verdade, constitui a abordagem clássica para o problema de localização.

O fato de considerarmos que a capacidade do armazém j é determinada pela quantidade de mercadorias transitando por período (T_j), pode ser sustentato mesmo em presença de variações sazonais de oferta e/ou demanda. Isto porque, se for conhecido o padrão de consumo de todo o mercado em estudo, será possível obter uma relação entre o movimento anual de cada armazém e a capacidade necessária para que possa suportar picos de produção e/ou demanda.

Isto pode ser melhor entendido pelo seguinte exemplo ilustrativo: A produção de soja de certa região varia segundo um padrão sazonal bem conhecido. A demanda é considerada inelástica. Conhecidos os padrões de produção e consumo de toda a região, pode-se estabelecer a necessidade de armazenagem em cada época, e assim determinar o pico de demanda de armazenagem.

ÉPOCA	PRODUÇÃO TOTAL	DEMANDA TOTAL	QUANTIDADE ARMAZENADA
1º TRIMESTRE	50.000 t	25.000 t	25.000 t
2º TRIMESTRE	5.000 t	25.000 t	5.000 t
3º TRIMESTRE	35.000 t	25.000 t	15.000 t
4º TRIMESTRE	10.000 t	25.000 t	0
TOTAIS ANUAIS	100.000 t	100.000 t	75.000 t

Assim, durante o ano, pela rede de armazéns transitam 75.000 t, mas a capacidade estática da rede só precisa ser de 25.000 t, ou seja, 1/3 do volume em trânsito na rede.

Como o padrão sazonal é unico para toda a região, se um armazém em particular movimentou 9000 t durante o ano (T_j), sua capacidade estática mínima deverá ser de 3000 t.

Chama-se Índice de Rotação o fator que, multiplicado pela capacidade estática do armazém, fornece o volume de trânsito possível durante o ano. No exemplo, o índice de rotação é 3. É óbvio que este índice não será, em geral, obtido pelo método simples do exemplo mostrado. No entanto, para a maioria dos casos onde os padrões sazonais são bem conhecidos é possível obter índices de rotação com precisão razoável. Quando G_j é um parâmetro constante, o índice de rotação pode ser incorporado a ele, de modo que a parcela variável do custo no armazem j é obtida pela multiplicação de G_j pelo volume de trânsito T_j.

Outra consideração importante no estudo dos custos de armazenagem é a existência de economias de escala.

De maneira geral entende-se por economia de escala o fato dos custos unitários diminuirem à medida que aumenta o nível da atividade econômica, no nosso caso o volume de trânsito T_j.

Muitos autores consideram a existência de economias de escala nos custos de armazenagem mesmo quando estes são representados por uma função do tipo $f_j = F_j + G_jT_j$, isto é, quando o custo marginal G_j é constante. A figura 1.1 permi te vizualizar esta função

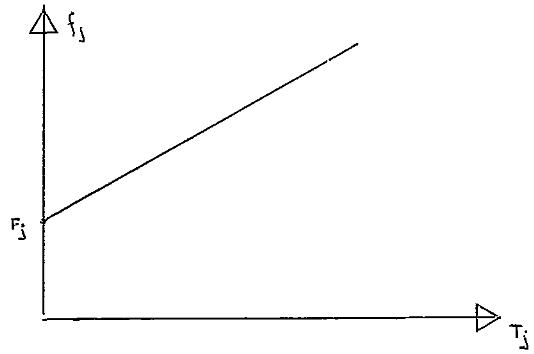


FIGURA 1.1

Respectivamente, a função de custo unitário teria a forma da figura 1.2

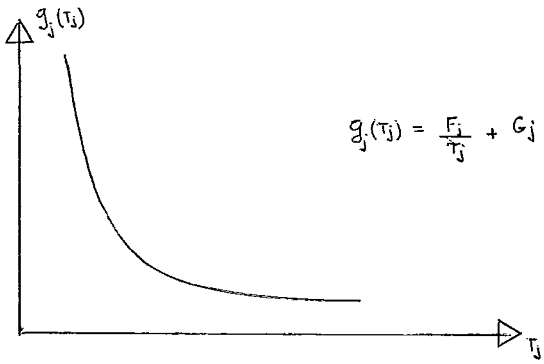


Figura 1.2

Vemos que apesar de G_j ser constante, a função de custo unitário g_j diminui com T_j. Isto deve-se à presença dos custos fixos F_j. Tomando-se custos unitários é claro que $\frac{F_j}{T_j}$ decresce com T_j. Este fenômeno pode ser considerado como economia de escala.

Outros autores consideram fundamentalmente a existência de economia de escala quando o custo marginal é uma função decrescente do volume de trânsito T_j. Teríamos neste caso uma função do tipo $f_j(T_j) = F_j + G_j(T_j) \cdot T_j$, que pode ser vizualizada na figura 1.3

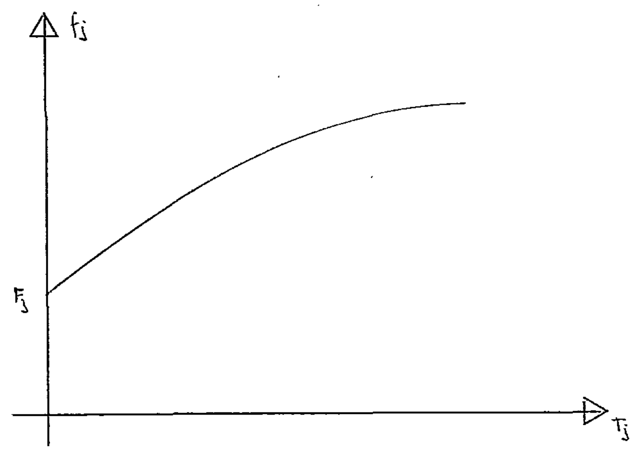


Figura 1.3

A função de custo unitário teria uma forma semelhante à do primeiro caso:

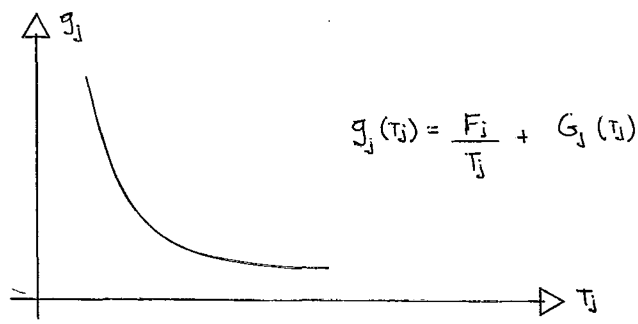


Figura 1.4

Como $G_j(T_j)$ é uma função decrescente em T_j, é lógico que a função de custo unitário da figura 1.4 decresce mais rápidamente que no primeiro caso (figura 1.2), o que de certa maneira legitima melhor esta interpretação para o problema de existência de economia de escala.

O algoritmo por nós desenvolvido permite a abordagem de ambos os casos acima descritos. Podemos tratar tanto o caso do custo marginal constante como o caso do custo marginal variável.

Embora estruturas de custo do tipo da figura 1.3 apareçam frequentemente na prática, a maioria dos analistas tem optado por uma aproximação linear, de modo a se recair no caso anterior (figura 1.1).

Entretanto, em alguns casos, onde a sensibilidade a custos de armazenagem é grande, os ganhos de escala devido a custos marginais decrescentes são tão relevantes que torna-se perigoso negligenciá-los. Um dos recursos mais freqüentes a que se recorre quando é inevitável levar em conta os custos marginais decrescentes é trabalhar com funções lineares por segmento ("piecewise linear").

Desta maneira, é possível tratar de economias de escalas por algoritmos de programação inteira mista, como veremos adiante.

Este recurso pode ser visualizado no gráfico abaixo.

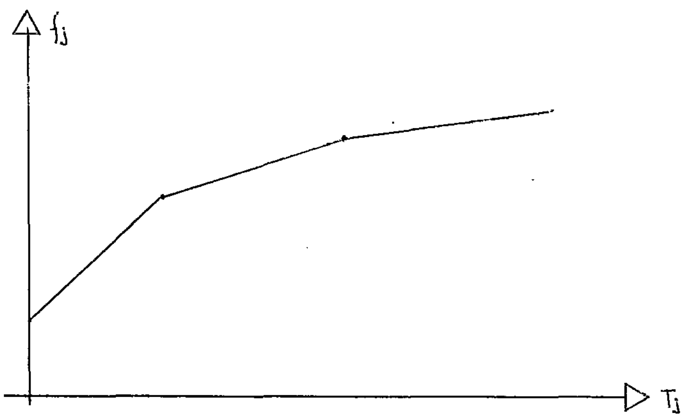


Figura 1.5

A consideração de economias de escala nos custos totais de armazenagem foi um fato importante na elaboração deste trabalho. Na realidade, um dos objetivos principais da pesquisa que deu origem a esta monografia foi o desenvolvimento de um método de solução que pudesse tratar eficientemente desse tipo de problema, além de outras condições particulares.

2. SOLUÇÕES PARA O PROBLEMA DE LOCALIZAÇÃO

2.1. MODELOS MATEMÁTICOS

Como já vimos o problema de localização pode apresentar diversas variantes pela consideração de uma ou mais condições particulares já referidas.

Mostraremos a seguir 3 modelos matemáticos levando em conta algumas variantes.

MM2 - Neste modelo, bastante simples, os custos nos armazéns tem uma parcela fixa, F_j, e o custo marginal G_j é constante

$$\text{MIN } W = \sum_{i=1}^{\ell} \sum_{j=1}^{m} c_{ij} x_{ij} + \sum_{j=1}^{m} (Y_j F_j + G_j \sum_{c=1}^{\ell} x_{ij}) + \sum_{j=1}^{m} \sum_{k=1}^{n} d_{jk} z_{jk}$$

sujeito à:

(1) $\sum_{j=1}^{m} X_{ij} = 0_i \qquad i = 1, \ldots\ldots \ell$

(2) $\sum_{j=1}^{m} z_{jk} = D_k \qquad k = 1, \ldots\ldots n$

(3) $\sum_{i=1}^{\ell} x_{ij} - \sum_{k=1}^{n} z_{jk} = 0 \qquad j = 1, \ldots\ldots m$

(4) $\sum_{i=1}^{\ell} x_{ij} \leq Y_j L_j \qquad j = 1, \ldots\ldots m$

$x_{ij} > 0 \qquad \forall_{i,j}$

$z_{jk} > 0 \qquad \forall_{j,k}$

$y_j = 0$ ou 1

Neste modelo as variáveis e parâmetros já usados em MM1 conservam suas significações.

As restrições do tipo (1), (2) e (3) também são mantidas.

É introduzida a variável bivalente Y_j que assumirá o valor 1, indicando que o projeto do armazém j deve ser efetivado, ou 0, caso contrário.

As restrições do tipo (4) servem para impor um limite (L_j) no tamanho do armazém j. Através das restrições tipo (4) impõe-se também que caso algum Y_j, digamos, Y_m, for zero, então $\sum_{i=1}^{\ell} x_{im} = 0$, o que garante que todos os custos no local m serão nulos.

Algumas condições particulares podem ser adicionadas ao modelo através de restrições simples nas variáveis bivalentes. Por exemplo, se é imposta a realização do projeto p basta adicionar a equação

$$Y_p = 1 \quad \text{ao modelo.}$$

Se ao menos um dos elementos de um conjunto de locais prioritários deve ter o projeto efetivado, basta adicionar

$$\sum_{j \in P} Y_j \geq 1$$

onde P é o conjunto de índices de locais prioritários.

MM3 - Neste modelo, as capacidades dos armazéns não podem ser quaisquer, mas devem ser múltiplas de um certo módulo básico. Este modelo aplica-se particularmente a armazéns agrícolas que são constituídos de um certo número de silos. Não consideramos aqui os custos marginais.

$$\text{MIN } W = \sum_{i=1}^{\ell} \sum_{j=1}^{m} c_{ij}x_{ij} + \sum_{j=1}^{m} (F_j Y_j + Gn_j) + \sum_{j=1}^{m} \sum_{j=1}^{m} d_{jk}z_{jk}$$

sujeito à:

Restrições (1),(2) e (3) dos modelos anteriores

e

(4) $\sum_{i=1}^{\ell} x_{ij} \leq Ln_j$

(5) $n_j \leq Ny_j$

$x_{ij} \geq 0 \quad \forall_{i,j}$

$z_{jk} \geq 0 \quad \forall_{j,k}$

$y_j \;\; = 0$ ou 1

n_j inteiro

As variáveis a parâmetros já usados nos modelos anteriores conservam suas significações. Temos ainda:

n_j - número de módulos no armazem j.

N - número máximo de módulos em um armazem.

G - custo de cada módulo.

L - capacidade de cada módulo.

MM4 - Economias de escala podem ser tratadas por programação inteira mista, embora nesse caso o número de variáveis inteiras seja drásticamente aumentado. Neste caso, os custos nos armazéns são funções lineares por segmento, onde os custos marginais decrescentes representam economias de escala, como ilustrado na figura 2.1 .

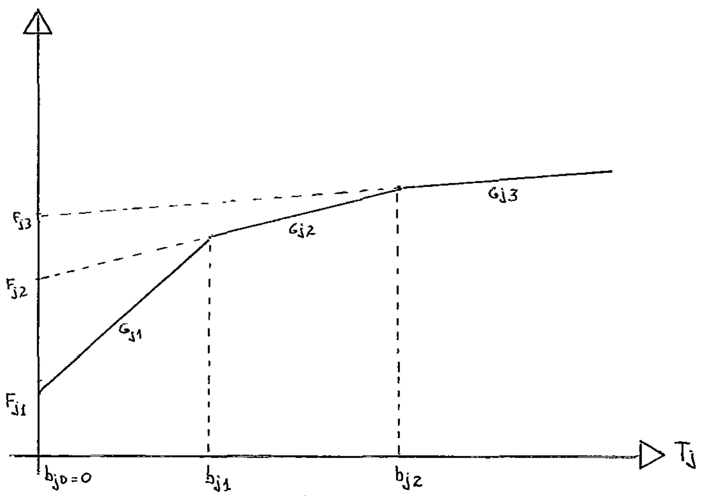


Figura 2.1

Um modelo matemático para este caso terá a seguinte forma básica:

$$\text{MIN } W = \sum_{i,j} c_{ij}x_{ij} + \sum_{j,\ell} (F_{j\ell}Y_{j\ell} + G_{j\ell}T_{j\ell}) + \sum_{j,k} d_{jk}z_{jk}$$

sujeito à:

$$(1) \quad \sum_{j} x_{ij} = O_i \qquad \forall i$$

$$(2) \quad \sum_j z_{jk} = D_k \qquad \forall k$$

$$(3) \quad \sum_i x_{ij} - \sum_k z_{jk} = 0 \qquad \forall j$$

$$(4) \quad \sum_\ell T_{j\ell} = \sum_i x_{ij} \qquad \forall j$$

$$(5) \quad b_{j\,\ell-1}\, y_{j\ell} \leq T_{j\ell} < b_{j\ell}\, y_{j\ell} \qquad \forall j,\ell$$

$$(6) \quad \sum_\ell y_{j\ell} \leq 1 \qquad \forall j$$

onde:

$$x_{ij} \geq 0 \quad \forall i,j$$

$$z_{jk} \geq 0 \quad \forall j,k$$

$$T_{j\ell} \geq 0 \quad \forall j,\ell$$

$$y_{j\ell} = 0 \text{ ou } 1$$

As variáveis e parâmetros já usados em modelos anteriores conservam suas significações. Além disso:

$y_{j\ell}$ - variável bivalente, assumindo valor 1 quando o projeto no local j for efetivado, operando a custo marginal $G_{j\ell}$; e valor 0 caso contrário.

$T_{j\ell}$ - Trânsito por período no armazém j, na faixa de nível ℓ.

$b_{j\ell}$ - limites das faixas de nível de trânsito no armazém j. Para cada nível ℓ existem um custo fixo $F_{j\ell}$ e um custo marginal $G_{j\ell}$.

2.2. ALGUNS MÉTODOS DE SOLUÇÃO

Os modelos MM2, MM3 e MM4 são de programação inteira mista, e podem ser bastante sofisticados, incorporando-se a eles diversas outras condições particulares. (Ver Elson |6|)

O emprego de "packages" de uso geral (MPSX, TEMPO, OPHELIE, etc) na solução de programas matemáticos mistos tem revelado que o tempo de processamento é excessivamente alto para este tipo de problema, pois os modelos típicos em usos reais têm um elevado número de variáveis inteiras.

O número de variáveis inteiras será ainda drásticamente aumentado se forem consideradas economias de escala nos custos de armazenagem, através de funções "piecewise linear".

Aproveitando-se a estrutura especial do problema, foram desenvolvidos alguns algoritmos do tipo "Branch-and-Bound" especializados em problemas de localização, possibilitando a redução do tempo de computação.(Ver referências |5|, |19| e |22|)

A estrutura do problema de localização torna-o também especialmente adequado ao tratamento pelo algoritmo de decomposição de Benders. Geoffrion e Graves |10| reportam excelentes resultados no projeto de um Sistema de Distribuição com vários produtos, usando decomposição.

Sweeney e Tathan |22| usam também o algoritmo de Benders para planejar a expansão de uma rede de armazéns ao longo de um horizonte de planejamento. A evolução de demandas e ofertas é suposta conhecida em cada estágio de tempo. Para cada estágio, são calculadas as R melhores soluções do problema estático de localização. Um política ótima de ampliação da rede no horizonte de planejamento é então determinada por programação dinâmica.

MÉTODOS ENUMERATIVOS

Valendo-se da estrutura especial do problema, Manne |16|, Stollsteimer |21|, Warrack |26| e outros desenvolveram métodos enumerativos especializados. Embora esses métodos permitam grande flexibilidade quanto à consideração de condições particulares (inclusive economias de escala), a maioria apresenta restrições quanto ao tempo de computação (muito elevado), ou não garante a obtenção de um ótimo global.

MÉTODOS HEURÍSTICOS

Como uma alternativa computacionalmente mais rápida para os métodos enumerativos, surgiram diversos trabalhos propondo métodos heurísticos como abordagem ao problema de localização. Entre estes, destacam-se os trabalhos de Feldman, Leher e Ray |7|, Kuehn e Hamburger |12| e Sá |19|.

A idéia básica desses métodos é o exame de várias configurações da rede de armazenagem através de exclusões

("DROP") ou adições ("ADD") sucessivas de armazéns em locais candidatos. O critério para escolha do próximo armazém a ser excluído ou adicionado à rede em cada estágio depende da heurística adotada.

Como vantagens principais dos métodos heurísticos, podemos citar:

1 - Rapidez no processamento em computador, permitindo o tratamento de problemas de grande porte.

2 - Flexibilidade, podendo tratar de uma grande variedade de modelos e condições particulares.

Como desvantagens, apontaríamos:

1 - Não se garante a obtenção de um ótimo global, embora a maioria das experiências práticas tenha revelado resultados satisfatórios.

2 - A eficiência do método depende de estrutura do modelo, isto é, varia de caso em caso.

Um dos trabalhos mais importante nessa área é o de Kuehn e Hamburger |12|, que adotam a estratégia de adição de armazéns no seu programa heurístico.

As idéias básicas do programa de Kuehn e Hamburger são derivadas dos seguintes princípios:

1 - "Soluções quase-ótimas (ou ótimas) para o problema de localização podem ser obtidas fixando um armazém de cada vez. A cada estágio da análise é adiciona-

à rede o armazém que produzir a maior redução possível no custo total do sistema".

2 - "Somente um pequeno subconjunto de todos os locais candidatos precisa ser avaliado em detalhe a cada estágio da análise para se determinar o próximo armazém a ser adicionado à rede".

Seja M o número de locais candidatos. Suponhamos que o programa está no fim do estágio k, isto é, k armazéns já foram incorporados à rede e deseja-se determinar qual será o próximo. Para assegurar que este próximo armazém produzirá a maior redução possível no custo, seria necessário avaliar em detalhe as M-k alternativas restantes. Cada avaliação destas consistiria em resolver um problema de minimização de custo numa rede de fluxo composta de k+1 armazéns, por um algoritmo de programação matemática apropriado.

No entanto, o tempo necessário nesse tipo de avaliação é grandemente reduzido no programa, pela utilização do princípio 2. São colocados em um "buffer" os N "melhores" candidatos dentre os M-k restantes, e somente estas N alternativas são avaliadas em detalhe, a cada estágio.

A escolha dos N "melhores" candidatos a serem colocados no buffer é feita segundo um critério puramente heurístico: no uso, escolhem-se os candidatos de maior demanda local.

A cada estágio, N candidatos são incorporados à rede temporariamente, um de cada vez. Aquele que produzir a maior redução nos custos, será incorporado efetivamente. Se algum dos N locais considerados produzir uma elevação nos custos, será alijado de consideração futura.

O programa principal termina quanto nenhuma adição de candidatos puder produzir redução nos custos.

Ao fim do programa principal uma rotina especial tenta melhorar a solução obtida de 2 maneiras:

1 - Eliminação de armazéns que por acaso tenham se tornado anti-econômicas com a inclusão posterior de outros.

2 - Avaliação de possíveis melhorias no custo total pelo deslocamento de armazéns ora ativos para outros locais que são servidos por aqueles armazens.

Uma idéia sintética do programa heurístico pode ser obtida pelo exame do fluxograma na figura 2.3.

Embora o programa de Kuehn e Hamburger possa ser adaptado para tratar de problemas com economias de escala nos custos de armazenagem, não é mencionada no trabalho nenhuma técnica especial para o tratamento desse problema.

Posteriormente, Feldman, Leher e Ray |7| apresentaram um trabalho propondo um método heurístico semelhante, onde o tratamento de economias de escala foi destacado. Em contrapartida à técnica de adicionar armazéns à rede, este programa é iniciado com todos os armazéns ativos. A cada estágio é excluído da rede o armazém cuja desativação produzir a maior

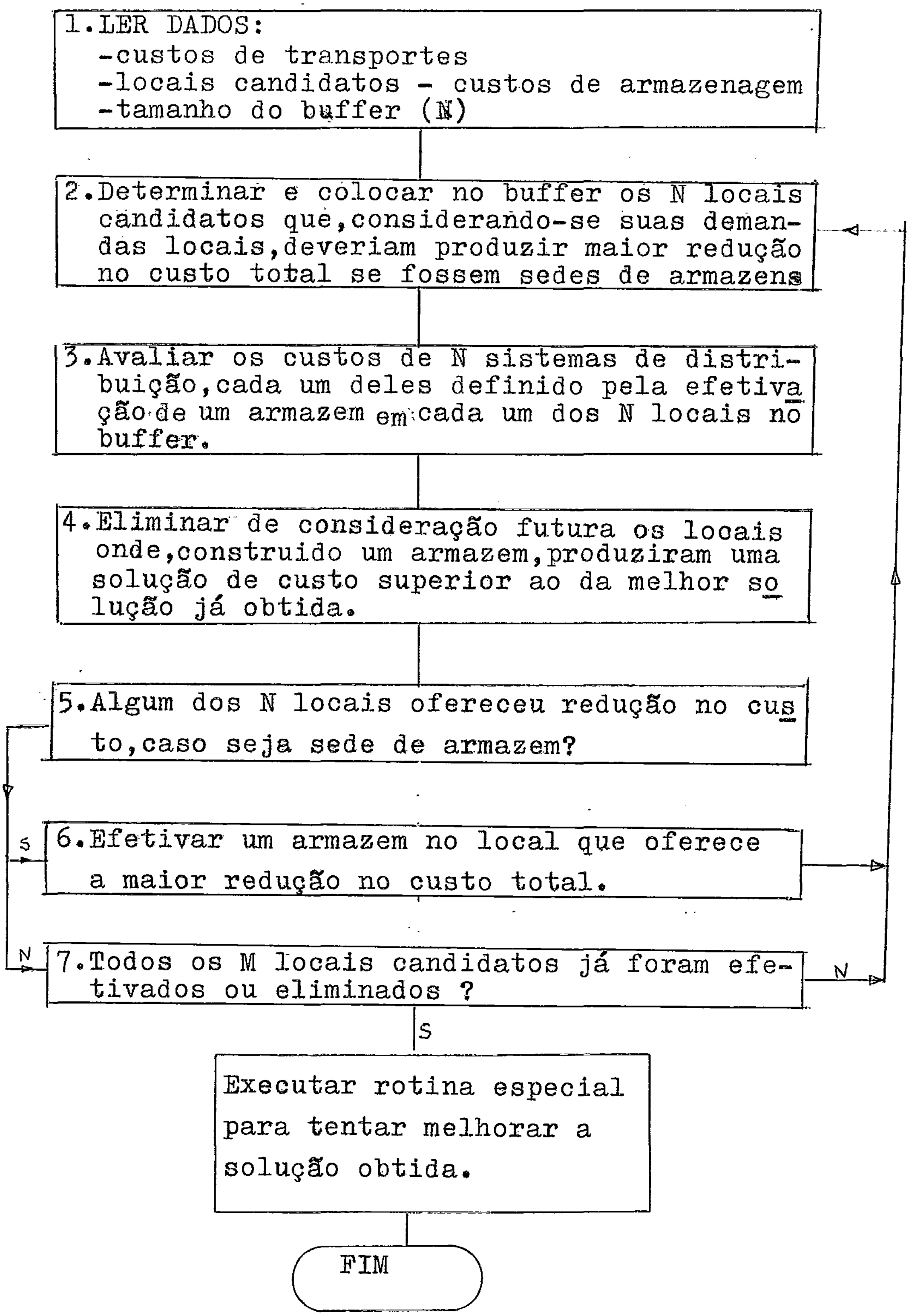

1. LER DADOS:
-custos de transportes
-locais candidatos - custos de armazenagem
-tamanho do buffer (N)
2. Determinar e colocar no buffer os N locais candidatos que, considerando-se suas demandas locais, deveriam produzir maior redução no custo total se fossem sedes de armazens
3. Avaliar os custos de N sistemas de distribuição, cada um deles definido pela efetivação de um armazem em cada um dos N locais no buffer.
4. Eliminar de consideração futura os locais onde, construido um armazem, produziram uma solução de custo superior ao da melhor solução já obtida.
5. Algum dos N locais ofereceu redução no custo, caso seja sede de armazem?
S
6. Efetivar um armazem no local que oferece a maior redução no custo total.
N
7. Todos os M locais candidatos já foram efetivados ou eliminados ?
N
S
Executar rotina especial para tentar melhorar a solução obtida.
FIM

Figura 2.3

redução nos custos.

UM MÉTODO ITERATIVO

King e Logan |13| apresentaram um método iterativo interessante para obter uma solução do problema de localização onde os custos de armazenagem são funções não lineares dos volumes de trânsito.

Em cada iteração, um problema de transporte com "transhipment" |18| é resolvido com custos de armazenagem avaliados a partir da iteração anterior. Devido ao uso de rede de transportes na modelagem, os custos de armazenagem são tratados na forma de custo unitário ($/tonelada de capacidade).

Resumidamente, o método iterativo segue os seguintes passos:

1 - Na primeira iteração, a todos os locais candidatos é dado o custo unitário mínimo, isto é, o custo unitário correspondente ao armazém de maior tamanho possível. O problema de distribuição é então resolvido pelo algoritmo apropriado, obtendo-se assim os volumes nos armazéns sob estas condições iniciais.

2 - A partir dos volumes de armazenagem obtidos na primeira iteração, são obtidos os custos correspondentes, através de uma função de custo unitário de armazenagem.

3 - Estes custos são comparados com os custos na iteração anterior. Nos locais onde houver discrepância, os custos são corregidos, e nova iteração começa. O processo iterativo continua até que se obtenha uma coerência entre os volumes de armazenagem e os custos usados para obte-los.

Desta maneira, é possível resolver um problema de programação não-linear por técnicas lineares. A prova de convergencia do método pode ser obtida em |9|.

É preciso destacar, no entanto que a solução de equilíbrio obtido pelo método iterativo depende do conjunto de custos iniciais. Isto quer dizer que existe mais de uma solução de equilíbrio, e portanto a solução obtida por este método pode ser apenas um ótimo local para o problema de localização.

Os autores, constatando este problema, sugeriram que deveria ser feita uma análise dos resultados obtidos, numa tentativa de, através de alterações nos custos de equilíbrio, corrigir imperfeições eventuais. O programa seria então rodado outra vez, obtendo-se possívelmente uma solução melhor.

A nosso ver, o método de King e Logan se constitui num bom instrumento de análise. No entanto, deve-se procurar desenvolver um método de busca automático onde várias soluções de equilíbrio (mínimos locais) possam ser examinadas, aumentando assim a probabilidade de obtenção de ótimo global.

PROGRAMAÇÃO CÔNCAVA

Quando as funções de custo nos armazéns tem a forma descrita na figura 1.2, temos um problema de minimização de uma função côncava em um politopo convexo.

Candler, Snyder e Faught |3| utilizam-se de um algoritmo desenvolvido por Tui |25|, para resolver problemas de localização com funções de custo côncavas.

Através deste algoritmo, é possível o exame de vários mínimos locais, na busca de uma solução ótima. Considerando-se que o mínimo de uma função côncava sempre ocorre em pelo menos um vértice de um politopo convexo, o algoritmo segue a seguinte seqüência:

1 - O algoritmo inicia-se numa solução básica primal viável, ou seja, num vértice do politopo.

2 - Através de pivotagens, são obtidas todas as soluções básicas viáveis correspondentes a vértices vizinhos do primeiro. Em cada vértice, o valor da função objetivo é calculado, até que se determine um mínimo local.

3 - Esta solução (ótima local) é guardada, e o vértice correspondente a ela é eliminado do politopo por um hiperplano de corte, que não exclui nenhuma solução melhor. A pesquisa de novo mínimo local inicia-se então no novo politopo.
Se não for encontrada nenhuma solução melhor, o últi

timo mínimo local é aceito como solução do problema.

Embora não se possa garantir a obtenção de solução ótima, trata-se de uma interessante aplicação de programação matemática a este tipo de problema.

3. DESENVOLVIMENTO DE UM MÉTODO HEURÍSTICO DE SOLUÇÃO

Neste capítulo são apresentados os resultados práticos das pesquisas realizadas nesse trabalho. Na seção 3.1 é feita uma análise detalhada das particularidades de um problema real, e um modelo matemático apropriado ao trato dessas condições particulares é adotado. Na seção 3.2, é proposta uma abordagem heurística ao problema, e é feita uma breve revisão das dificuldades provenientes da adoção de métodos heurísticos tipo "ADD-DROP" em presença de economias de escala. Uma solução para tais dificuldades é apresentada na seção 3.3, e uma idéia geral do método heurístico adotado é mostrado na seção 3.4. Esse método é descrito detalhadamente na seção 3.5, onde é adotado um enfoque computacional. A seção 3.6 trata de uma adaptação especial feita para que se pudesse tratar de armazéns agrícolas constituídos de silos com capacidade pré-determinada.

Finalmente, a seção 3.7 mostra os resultados obtidos em diversos testes de utilização do programa heurístico. A eficiência do algoritmo é discutida em termos de qualidade da solução e tempo de processamento.

3.1. ANÁLISE DE UM PROBLEMA

Como referência para a formulação de um modêlo matemático, consideramos o estudo feito pela CIBRAZEM |4|, para o planejamento de uma rede de armazem de arroz no Maranhão.

Embora o referido estudo tenha considerado custos de transporte, esses não foram tratados quantitativamente, tendo-se apenas procurado dar prioridade a locais candidatos com melhores vias de escoamento e maior produção local. A rede de armazéns foi então dimensionada de modo a atender à demanda de armazenagem e minimizar o investimento em armazéns, considerando-se as economias de escala existentes.

É preciso admitir que os custos de transporte não representam ônus para a CIBRAZEM. De fato, o estudo em questão trata de minimizar o ônus da companhia, o que é compreensível, procurando paralelamente melhorar as opções de transporte.

Numa segunda análise, no entanto, talvez fosse interessante encorporar custos de transporte a futuros estudos. Um dos objetivos básicos da CIBRAZEM é proporcionar ao produtor (principalmente pequeno e médio) a possibilidade de escapar à ação dos atravessadores e de ter acesso aos preços melhores da entressafra. Conforme o estudo da CIBRAZEM, "Devido as grandes distâncias e péssimas estradas da fonte de produção ao local de comercialização, os custos de fretes são tão elevados que tornam, muitas vezes, inviável para o produtor esse transporte, preferindo entregar sua produção ao intermediário".

Uma maneira de proporcionar a desejada segurança ao produtor seria construir uma extensa rede de pequenos armazéns situados próximos aos centros de produção, minimizando assim os custos de transporte. Esta solução, no entanto, seria demasiado onerosa, não só devido aos custos fixos na construção de cada armazém, como também na perda de economias de escala existentes em armazéns maiores. Devemos lembrar aqui que os recursos financeiros para a construção de armazéns são limitados.

Baseando-nos nas considerações acima, proporemos aqui um modelo matemático onde se deseja minimizar uma função de custo que inclui a despesa anual com transportes e a amortização anual do investimento em armazens(*).

Analisando o estudo da CIBRAZEM, que é bastante objetivo na descrição das facilidades e dificuldades existentes na região, concluimos que um modelo matemático deveria ter flexibilidade suficiente para levar em conta algumas particularidades do problema, a saber:

1 - A existência de armazéns já construídos (com possibilidade de ampliação/retração)

2 - A possibilidade de 2 ou mais meios de transporte entre as mesmas localidades. Isto porque algumas micro-regiões do Maranhão são servidas por estrada de ferro, embora a maior parte do transporte seja feita por rodovia.

3 - A produção é sazonal. No entanto, índices de rotação

(*) opcionalmente, também a despesa operacional dos armazéns pode ser incluida na função de custo.

podem ser calculados (o estudo da CIBRAZEM fornece os índices de rotação de cada micro-região).

4 - Existem limitações sérias na capacidade de escoamento da rede de transportes: a frota de caminhões na região é limitada e grande parte das rodovias fica com precárias condições de tráfego na época das chuvas, quando a demanda por transporte é grande.

5 - Pode ser imposta a utilização de certas vias, mesmo que sejam anti-econômicas. Isto porque, para não desamparar pequenos produtores, a CIBRAZEM pode impor que armazéns sejam construídos em locais anti-econômicos. Limitações desse tipo podem também vir expressas em restrições do tipo "pelo menos um armazém será construído dentre os locais A, B e C".

6 - As capacidades dos armazéns são modulares, isto é, cada armazém é constituído de um certo número de silos de tamanho fixo.

7 - Podem existir economias de escala nos custos relativos a cada armazém.

Ultimamente,o problema de localização de armazéns e/ou facilidades de beneficiamento para produtos agrícolas tem recebido substancial atenção por parte dos economistas envolvidos em planejamento agrário. Uma revisão nos trabalhos publicados nos últimos 10 anos pelo "American Journal of Farm Economics" por exemplo, revelará que raras vezes este tipo de problemas foi abordado por programação inteira/mista.

Provavelmente devido à necessidade de se lidar com custos não lineares e ao grande porte dos problemas, métodos de enumeração parcial ou programas heurísticos tem sido usados com bastante freqüência.

Como já foi dito, esses métodos constituem-se em geral no exame de várias configurações da rede de armazenagem, através da exclusão ou adição de armazéns na busca da configuração de menor custo. Denominamos "configuração" da rede à cada uma das possíveis combinações obtidas ao se fixar o <u>número</u> de armazéns e os <u>locais</u> para tais armazéns.

Vejamos agora como é possível determinar o custo associado a cada configuração.

Quando são fixados os locais a serem considerados, temos um problema (linear) de distribuição numa rede de fluxo como a da figura 3.1

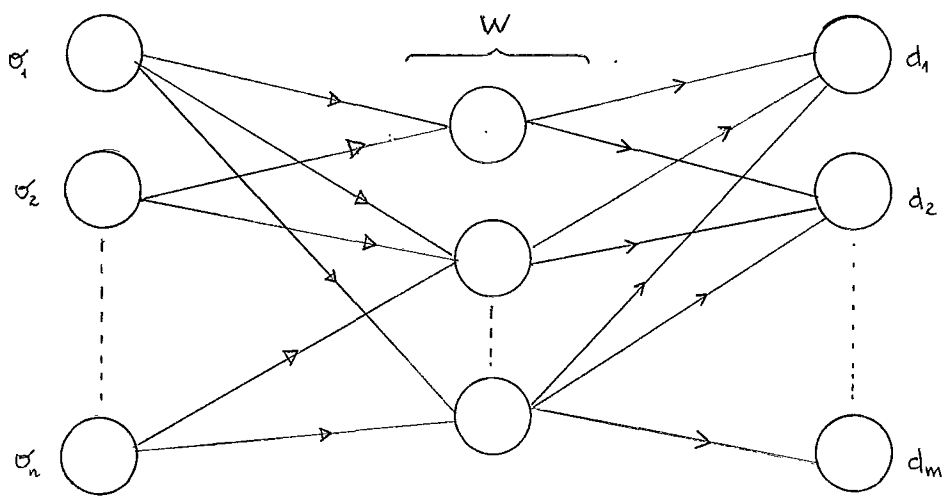


Figura 3.1

De maneira geral, podemos representar problemas de distribuição por uma rede orientada R(N,A) onde:

N : conjunto de nós

A : conjunto de arcos

P : conjunto de nós representando centros de produção $(P \subset N)$

D : conjunto de nós representando centros de consumo $(D \subset N)$

$W = N-P-D$: conjunto de nós representando armazéns
A cada configuração da rede corresponde um conjunto W

O_i : oferta no centro de produção i

d_j : demanda no centro de consumo j

c_{ij}: custo por unidade de fluxo no arco (i,j)

b_{ij}: capacidade máxima de fluxo no arco (i,j)

x_{ij}: fluxo no arco(i,j)

Em particular um problema de distribuição como o representado pela figura 3.1 pode ser escrito pelo seguinte modelo matemático:

$$\text{MIN} \quad z = \sum_{(i,j)\varepsilon A} x_{ij} c_{ij}$$

Sujeito à:

$$(1) \quad \sum_{j \varepsilon W} x_{ij} = O_i \qquad \forall i \; \varepsilon \; P$$

$$(2) \quad \sum_{i \varepsilon W} x_{ij} = d_j \qquad \forall j \; \varepsilon \; D$$

(3) $\sum_{i\varepsilon P} x_{ij} - \sum_{i\varepsilon D} x_{ji} = 0 \qquad \forall j \; \varepsilon \; W$

(4) $0 \leq x_{ij} \leq b_{ij} \qquad \forall (i,j) \; \varepsilon \; A$

As equações do tipo (1), (2) e (3) estão baseadas no mesmo conjunto de hipóteses usadas na formulação do modelo MM1 (capítulo 1), como é usual.

No caso, c_{ij} representa o custo unitário de transporte, se $j\varepsilon D$, e custo unitário de transporte + armazenagem, se $j\varepsilon W$.

O grafo da figura 3.1 é característica do problema de "transshipment". Este problema pode ser resolvido pela variante clássica do método simplex para o "problema de transportes", como mostra Orden |[18]|.

A maioria dos trabalhos na área de localização de armazens tem usado o método de Orden como instrumento de cálculo dentro de algoritmos voltados para problemas de localização, como King e Logan |[13]|, Ladd e Lifferth |[14]| e outros.

Neste trabalho,sugerimos o uso de um algoritmo "out-of-kilter" para avaliar o custo mínimo associado a cada configuração.

Uma análise das 6 condições particulares do nosso problema já citadas mostra que o modelo de transshipment é inadequado para o trato da maioria dessas condições. Em vista disso, propomos que cada configuração seja modelada por uma re de capacitada (onde os arcos tenham limites inferiores e superiores).

Uma rede capacitada, além de proporcionar grande flexibilidade na modelagem, nos parece mais adequada aos problemas de distribuição dos países em desenvolvimento. Por exemplo, a condição (4) dificilmente existirá em países desenvolvidos, sendo, no entanto, muito comum no Brasil. A rede correspondente ao modelo de "transshipment" não é muito flexível, e haveria certamente dificuldades para levar em conta as condições (2), (4) e (5) caso ela fosse usada.

Como veremos a seguir, o uso de um algoritmo "out-of-kilter" permitirá maior flexibilidade no desenho da rede. No entanto, como esse algoritmo minimiza custos em CIRCULAÇÕES (redes sem fontes nem semidouros), devemos adaptar nossa rede de modo que o fluxo neja seja uma circulação.

Isto pode ser conseguido adiconando-se à rede 2 nós fictícios $\underline{s}$ e $\underline{t}$, e associando a cada arco um limite mínimo de fluxo. Assim, cada arco da rede R(N,A) terá associados 3 parâmetros (a_{ij}, b_{ij}, c_{ij}), onde

a_{ij} - limite mínimo de fluxo

b_{ij} - limite máximo de fluxo

c_{ij} - custo por unidade de fluxo.

Podemos então, construir um grafo equivalente ao anterior onde todos os arcos pertençam a um circuito, como na figura 3.2.

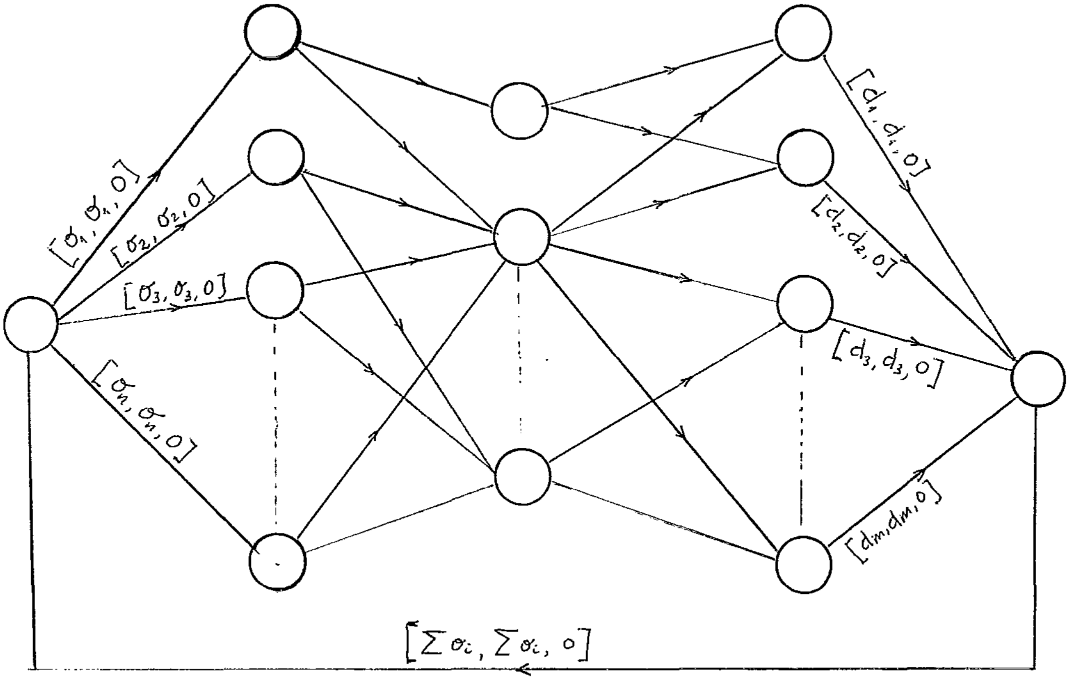


Figura 3.2

Observemos que os arcos fictícios (s,i), i ε P, (i,t), i ε D, e (s,t) possuem custo zero, de modo que sua introdução no modelo não causa alteração no custo total.

Além disso, os limites de fluxo nesses arcos impõem que pelos nós correspondentes a centros de produção passem fluxos correspondentes às respectivas ofertas, e pelos centros de consumo passem fluxos iguais às suas demandas.

Um modelo matemático (MM5) para esta rede será:

$$\text{MIN} \sum_{(i,j)\varepsilon A} c_{ij} x_{ij}$$

Sujeito à:

$$\sum_{j\varepsilon\Gamma^+(i)} x_{ij} - \sum_{j\varepsilon\Gamma^-(i)} x_{ji} = 0 \qquad \forall i \ \varepsilon \ N$$

$$a_{ij} \leq x_{ij} \leq b_{ij} \qquad \forall (i,j) \ \varepsilon \ A$$

onde $\Gamma^+(i) = \{j \in N \mid (i,j)\}$ - conjunto de nós sucessores de i.

$\Gamma^-(i) = \{j \in N \mid (j,i)\}$ - conjunto de nós antecessores de i.

O programa matemático descrito no modelo MM5 é resolvido com bastante eficiência pelo algoritmo "out-of-kilter" de Ford e Fulkerson.

Através deste modelo matemático bastante simples, podemos descrever diversas situações reais complexas, através do desenho apropriado da rede. Vejamos alguns caso:

(a) ARMAZÉNS - a representação de armazéns por um conjunto de nós $W \subset N$ (figura 3.2) pode trazer dificuldades para avaliar custos de armazenagem. Isto porque, quando existem economias de escala, estes custos são função do trânsito anual

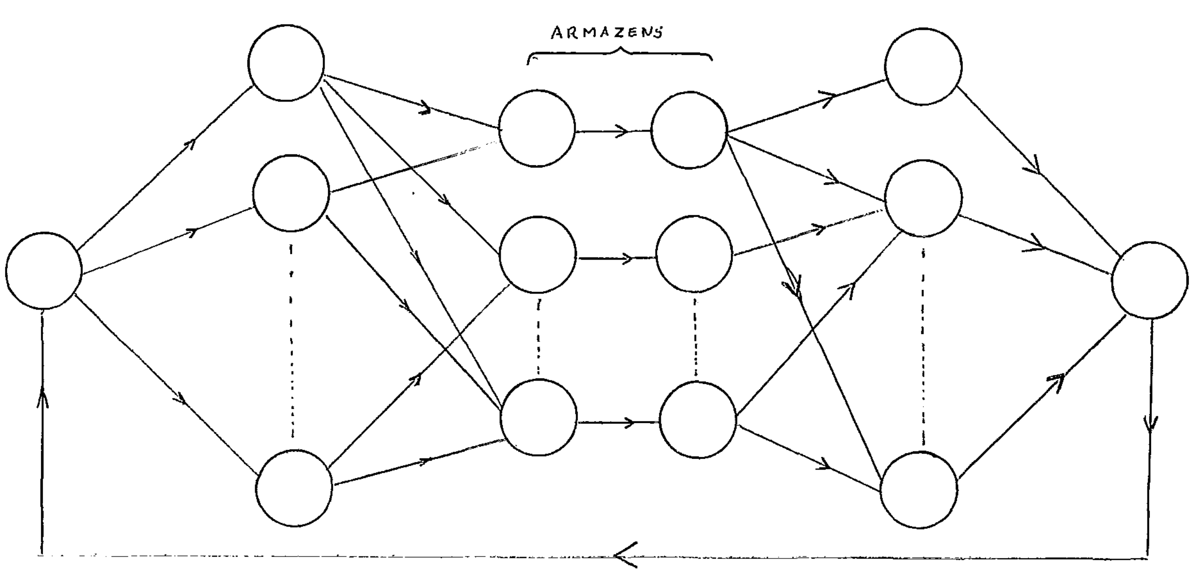
$T_j = \sum_{i \in \Gamma^-(j)} x_{ij}$ em cada armazém j (representado pelo nó j).

Assim, preferimos representar armazéns por um conjunto de arcos H obtidos pela introdução de nós artificiais na rede.

Seja k o índice do conjunto H. Então, o trânsito no k-ésimo armazém (T_k) será simplemente o fluxo x_{ij} no k-ésimo arco $(i,j) \in H$.

Em caso de custos não lineares, a cada arco $(i,j) \in H$ estará associado uma função de custo $g_k(T_k)$.

A rede teria então o aspecto da figura 3.3



Uma outra importante vantagem dessa contrução é a facilidade de excluir armazéns da rede. Isto pode ser feito tornando zero a capacidade de fluxo b_{ij} do arco correspondente, ou tornando o custo associado a este arco extremamente alto.

(b) <u>ARMAZÉNS MODULADOS</u> - Quando as capacidades dos armazéns só podem ter certos valores pré-fixados, representamos um armazem por 2 nós ligados por um conjunto de arcos, como na figura 3.4

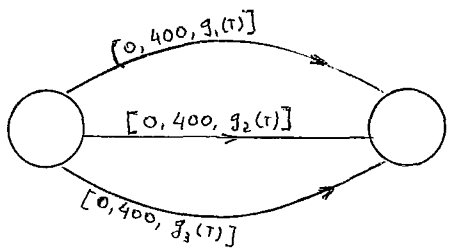


Figura 3.4

Devido à existência de economias de escala, as funções de custo g_i(T) podem ser diferentes para arcos de um mesmo armazém.

No exemplo, o armazém pode ter de 1 a 3 módulos de 400 toneladas de capacidade. Dificuldades especiais desta formulação serão comentadas mais tarde.

(c) ARMAZÉNS JÁ EXISTENTES OU IMPOSTOS - Se um armazém já existente deve ser considerado no estudo, ele pode ser representado por um arco cujo limite inferior de fluxo é igual à sua capacidade atual. Se a possibilidade de ampliação deste armazém deve ser considerada, uma função de custo especial deve ser encontrada (apenas custos de ampliação).

Da mesma forma deve ser tratado um local candidato onde se deseja, por motivos políticos criar um armazem, independente de considerações econômicas. Nesse caso, será dado ao arco correspondente um limite inferior de fluxo correspondente à capacidade mínima de armazenagem desejada no local.

(d) PRIORIDADES GEOGRÁFICAS - Algumas vezes, podem surgir imposições de ordem não econômica, do tipo: "pelo menos um armazém deve ser construído dentre os locais x, y e z". Esse tipo de restrição pode ser fàcilmente retratado no desenho da rede, como mostra a figura 3.5

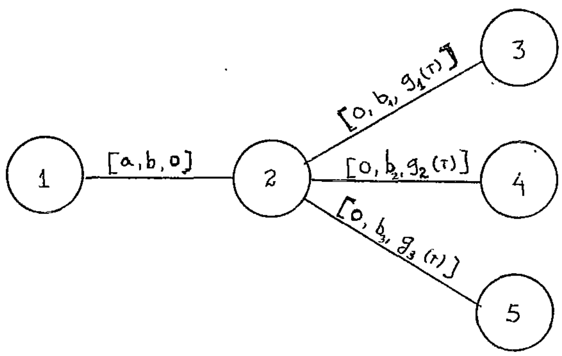


Figura 3.5

Os arcos (2,3), (2,4) e (2,5) representam locais candidatos a redes de armazéns (x,y e z). O arco (1,2) serve para garantir que um fluxo mínimo de a toneladas será armazenado na região definida pelos 3 locais candidatos.

(e) MÚLTIPLOS MEIOS DE TRANSPORTE - Se dois locais quaisquer da rede são servidos por mais de um meio de transporte com capacidades de fluxo e custos diferentes bastará ligar esses locais por mais de 1 arco, como mostra a figura 3.6

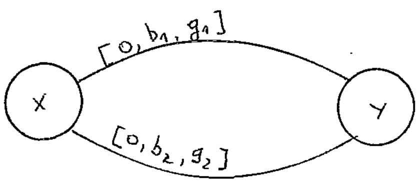


Figura 3.6

Esse caso poderia ser levado em conta num modelo de "transshipment", mas à custa da introdução de 2 nós artificiais aumentando a dimensão da matriz fluxos e o tempo computacional.

Como extenção deste caso, poderiam ser criados na rede alguns nós intermediários que representariam simplesmente pontos de baldeação, caso o transporte intermodal devesse ser considerado.

Nesse ponto, podemos estabelecer um conjunto de vantagens do uso do algoritmo "out-of-kilter" na avaliação do custo associado a cada configuração:

1 - Maior liberdade no desenho da rede, permitindo construir modelos que consideram grande variedade de con

dições particulares. A facilidade no uso de limites inferiores e superiores de fluxo contribui bastante para a flexibilidade de modelagem.

2 - Facilidade para alteração da rede. Por exemplo, na disponibilidade de nova estrada bastaria acrescentar um arco, enquanto que se o modelo de transshipment fosse usado, 1 nó e 1 arco artificiais teriam que ser criados.

3 - O algoritmo "out-of-kilter" não precisa iniciar com uma solução viável. Assim, quando limites inferiores ou superiores de fluxo forem alterados, uma solução anteriormente obtida (possivelmente inviável para a nova rede) pode ser usada como solução inicial. Este tipo de facilidade é bastante útil quando se deseja fazer estudos do tipo "o que aconteceria se ...".

Por exemplo, quando se deseja saber qual o impacto causado pelo aumento de 20% na produção de determinados centros, novo modelo será obtido pelo acréscimo de 20% em limites inferiores e superiores de alguns arcos. Mesmo assim, uma solução já disponível poderá ser usada como solução inicial (inviável) para o novo problema, proporcionando apreciável economia de tempo de computação.

ANÁLISE DE PÓS-OTIMIZAÇÃO

O algoritmo "out-of-kilter" utiliza-se, durante a computação da solução ótima de 2 tipos de variáveis:

1 - variáveis primais x_{ij}, que representam o fluxo em cada arco da rede

2 - variáveis duais u_k, associadas a cada nó da rede.

Durante a computação, utiliza-se também da medida $\bar{c}_{ij} = c_{ij} + u_i - u_j$, que pode ser interpretado como um tipo de custo marginal associado a cada arco.

Se na solução ótima temos que $x_{ij} = b_{ij}$ e $\bar{c}_{ij} < 0$, então o arco (i,j) é um ponto de estrangulamento na rede.

Isto quer dizer que se aumentássemos a capacidade deste arco haveria uma redução de $\bar{c}_{ij}$ cruzeiros no custo total por unidade de fluxo aumentado em (x,j).

Portanto, se o tal arco representa um armazém, a análise indica que a capacidade deste armazém deve ser ampliada. Se o arco representa uma via de transporte, então esta via é efetivamente um ponto de estrangulamento do sistema, e investimentos para aumentar sua capacidade de fluxo devem ser recomendados.

Por outro lado, se $x_{ij} = a_{ij}$ e $\bar{c}_{ij} > 0$, a utilização do arco (i,j) é anti-econômica. Nesse caso, $\bar{c}_{ij}$ representa o acréscimo que haveria no custo total se o fluxo nesse arco fosse aumentado de uma unidade.

Isto pode servir para uma previsão sôbre o aumento no custo do sistema no caso de haver aumentos na produção e/ou demanda de algum centro.

3.2. DIFICULDADES COM MÉTODOS HEURÍSTICOS

A utilização do modelo MM5 sugerida na seção 3.1 será util na avaliação do custo mínimo associado a cada configuração de rede de armazéns. Em outras palavras, o modelo MM5 será usado para determinar o custo mínimo de uma alternativa onde o número de armazéns e os locais foram fixados.

Para a determinação da melhor dessas alternativas adotamos nesse trabalho uma abordagem heurística, tendo em vista a rapidez e a flexibilidade proporcionadas.

Uma das principais dificuldades ao se usar métodos heurísticos do tipo "ADD" ou "DROP" consiste na determinação dos custos de armazenagem a serem usados durante a computação. Em presença de economias de escala, o custo marginal é uma função $G_j(T_j)$ do volume de trânsito em cada armazém. Uma primeira dificuldade reside em determinar quais os custos marginais de armazenagem iniciais, visto que não se conhece a priori quais serão os volumes de transito T_j que os determinarão.

Suponhamos agora que a técnica "DROP" esteja sendo aplicada. Ao se excluir um armazém, toda a produção que anteriormente por ele transitava vai se redistribuir entre os armazéns remanescentes. Isto quer dizer que, tendo maior volume, alguns armazéns vão passar a operar com custos marginais menores (caso existam economias de escala). Portanto, uma segunda dificuldade consiste em, ao se resolver o problema de minimização de custo na nova rede (com um armazém excluído), saber quais serão os novos custos nos armazéns remanescentes.

Feldman, Leher e Ray |7| adotaram a técnica de "DROP" onde os seguintes procedimentos são usados para contornar o problema de custos:

1 - No estágio zero (todos os armazéns ativos) é aplicado um método iterativo semelhante ao proposto por King e Logan |13|, descrito brevemente na seção 2.2 . Com isso, é obtida uma estrutura de custos iniciais coerente com os volumes de trânsito em cada armazém.

2 - Cada vez que um armazém é excluído temporariamente, os custos de armazenagem utilizados na nova configuração são os custos que existiam na última solução de equilíbrio encontrada. Embora este procedimento pareça causar imprecisão na avaliação da alternativa, não fica claro em |7| que uma avaliação mais precisa tenha sido adotada.

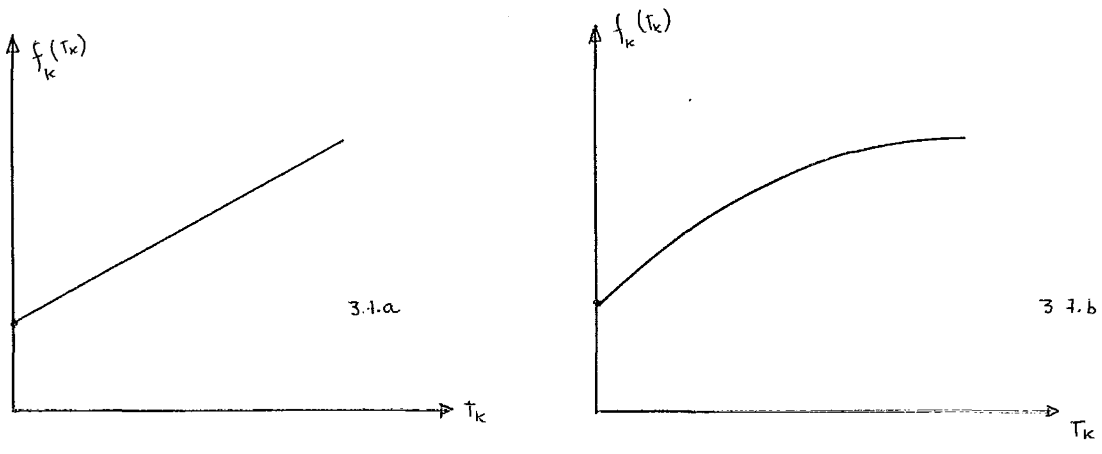
A partir da constatação dessas dificuldades, procuramos dar atenção ao problema de determinação de custos de armazenagem, adotando uma metodologia que será descrita nas seções 3.3 e 3.4.

3.3. UMA TÉCNICA DE AVALIAÇÃO DE CUSTOS

Como já vimos, o custo total de armazenagem é em geral uma função côncava $f_k(T_k)$ do trânsito anual T_k em cada armazém (figuras 3.7a e 3.7b).

Na seção 3.1, concluimos que um algoritmo "out-of-kilter" para minimização de custos em redes de fluxo seria o mais apropriado para otimizações intermediárias em nosso problema, por ser este algoritmo voltado para redes capacitadas (modelo MM5).

Entretanto, observamos que no modelo MM5 o custo c_{ij} associado a cada arco, e portanto a cada variável x_{ij}, é um custo unitário, ou seja, custo por unidade de fluxo.



Portanto, para que um algoritmo "out-of-kilter" possa ser usado, devemos conhecer a função de custo unitário em cada armazém. Esta função deverá ser convexa, como mostra a figura 3.8, sendo obtida pela divisão de funções do tipo das figuras 3.7a e 3.7b por T_k.

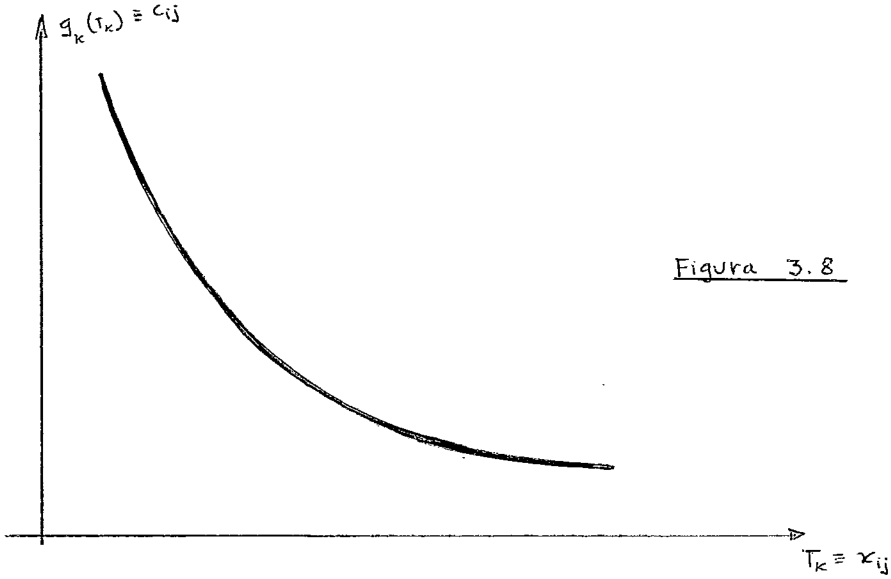


Figura 3.8

Se observarmos o formato da curva da figura 3.8, concluímos que enfrentaremos dificuldades semelhantes às descritas na seção 3.2, na avaliação de custos.

Isto pode ser melhor entendido da seguinte maneira: no modelo MM5 os custos c_{ij} são apresentados como parâmetros constantes. Por outro lado, os custos unitários correspondentes a arcos $(i,j)\varepsilon H$ (armazéns) são dados por funções do trânsito T_k. Em outras palavras, se associarmos a cada arco $(i,j)\varepsilon H$ um número k, então $c_{ij} = g_k(T_k)$ para todo $(i,j)\varepsilon H$.

Como é possível fornecer os parâmetro c_{ij} corretos para o algoritmo "out-of-kilter", se não se conhece, a priori, quais serão os T_k que os determinarão?

Uma das soluções para esta dificuldade está na utilização do método iterativo proposto por King e Logan |[13]|, que resumimos no capítulo 2. Este método permite que, ao final de um certo número de iterações, se obtenha uma estrutura de custos nos arcos $(i,j)\varepsilon H$ compatível com os trânsitos T_k nesses arcos.

Apresentamos a seguir uma descrição formal da rotina por nós desenvolvida para executar o método iterativo proposto por King e Logan. Doravante, nos referiremos a este método iterativo por rotina EQUILIBRAR(*).

Sejam

t - contador de iterações

c_{ij}^{t} - custo associado ao arco (i,j) usado pelo algoritmo "out-of-kilter" na iteração t.

x_{ij}^{t} - fluxo no arco (i,j) correspondente à solução ótima do "out-of-kilter" na iteração t.

k - índice do conjunto H, de arcos representando armazéns; a cada $(i,j) \varepsilon H$ associamos um número k.

T_k^t - trânsito anual no armazém k, calculado na iteração t.
Se este trânsito se refere ao arco $(i,j) \varepsilon H$, então $x_{ij}^{t} = T_k^t$.

I_k - índice de rotação (constante) do armzém k (ver capítulo 1)

V_k^t - capacidade estática no armazém k, necessária para suportar o trânsito anual T_k^t , calculado na itera-

---

(*) Este é o nome da rotina que executa o método iterativo no nosso programa, como pode ser visto no anexo 3.

ção t. Numèricamente, $V_k^t = T_k^t / I_k$.

$\overline{g}_k(V_k^t)$ - até agora, definimos uma função de unitário $g_k(T_k)$. Quando índices de rotação estão envolvidos, devemos calcular o custo unitário em função da capacidade estática do armazém, através de $\overline{g}_k(V_k^t)$.

$CR_{ij}^t = \overline{g}_k(V_k^t)$ - custo real unitário no arco $(i,j)\varepsilon H$, na iteração t. Este custo, ao contrário de C_{ij}^t, é sempre coerente com T_k^t.

$CTR_{i,j}^t$ - custo total no armazém representado pelo arco $(i,j)\varepsilon H$, na iteração t. $CTR_{i,j}^t = CR_{i,j}^t \times V_k$.

$CTA_{i,j}^t$ - custo total obtido quando custos unitários C_{ij}^t são usados. $CTA_{i,j}^t = C_{ij}^t \times V_k$.

$ERRO_{i,j}^t$ - erro relativo cometido quando se usa o custo C_{ij}^t para obter x_{ij}^t, mas o custo real é CR_{ij}^t.

TOL - Tolerancia. Uma solução será aceita pela rotina equilibrar se $ERRO_{i,j} \leq TOL \quad \forall (i,j)\varepsilon H$.

DESCRIÇÃO FORMAL DA ROTINA EQUILIBRAR

PASSO 0 Faça $t = 0$

Faça $C^t_{i,j}$ = valor mínimo possível(*), $\forall (i,j) \varepsilon H$

PASSO 1

Faça $t = t+1$

Se $t > 20$

Então PARE

Caso contrário,

obtenha uma solução $\{x^t_{ij} \mid (i,j) \varepsilon A\}$ para o problema descrito por MM5, através do algoritmo "out-of-kilter". Use C^t_{ij} para obter esta solução.

PASSO 2

Para todo $(i,j) \varepsilon H$, sendo $k \leftrightarrow (i,j)$

faça:

$V^t_k = x^t_{ij} / I_k$

$CR^t_{ij} = \bar{g}_k(V^t_k)$

$CTA^t_{i,j} = C^{t-1}_{ij} x V^t_k$

$CTR^t_{i,j} = CR^t_{i,j} x V^t_k$

PASSO 3

Para todo $(i,j) \varepsilon H$

(*) Como as funções g_k são decrescentes, o custo mínimo será o correspondente ao maior fluxo possível no arco $(i,j) \varepsilon H$ de maior capacidade.

faça $ERRO^t_{i,j} = \left| \dfrac{CTR^t_{i,j} - CTA^t_{i,j}}{CTR^t_{i,j}} \right|$

Se para algum $(i,j) \varepsilon H$, $ERRO^t_{i,j} > TOL$

então faça: $C^t_{ij} = CR^t_{ij} \quad \forall (i,j) \varepsilon H$

volte ao passo 1

caso contrário: PARE. A solução obtida é coerente com os custos usados para obtê-la.

Como se vê, o algoritmo "out-of-kilter" atua como subrotina da rotina EQUILIBRAR. O problema descrito em MM5 é resolvido várias vezes pelo algoritmo, utilizando-se de várias estruturas de custos, até que essa estrutura de custos esteja coerente com a solução obtida, dentro dos limites toleráveis.

No nosso programa, a rotina EQUILIBRAR permite até um máximo de 20 iterações, na busca da solução de equilíbrio(*).

Na nossa experiência computacional, a maioria das soluções de equilíbrio foi obtida após 3 ou 4 iterações.

Frank e Wolfe |9| mostram a convergência do método iterativo para funções de custo com segundas derivadas

(*) Aqui, nos referimos a um equilíbrio (ou coerência) entre os fluxos obtidos e os custos usados.

contínuas. No entanto, em nossas experiências pudemos obter soluções mesmo para funções não contínuas. A impressão que temos a partir de nossa experiência computacional é que o método iterativo é satisfatório para a maioria das funções com significado econômico prático.

Para melhor compreensão da rotina EQUILIBRAR, vejamos um exemplo de sua aplicação na localização de armazéns para arroz no centro do estado do Maranhão. São consideradas 13 cidades como locais candidatos a centros de armazenagem.

A função de custo unitário de armazenagem por período em cada local k é dada por:

$$\bar{g}_k(V_k) = \frac{1.222.000 + 365 \cdot V_k}{8V_k}$$

onde V_k é a capacidade do armazem k em toneladas, obtido pela multiplicação do trânsito anual no armazem (T_k) por um índice de rotação I_k.

$\bar{g}_k(V_k)$ - custo anual por tonelada de capacidade do armazém.

Na tabela 3.2.1 é mostrada a evolução dos custos e capacidades (V_k) ao longo das iterações.

Inicialmente, a todos os armazéns é dado o custo mínimo c_{ij} = Cr$ 25,00 por tonelada, isto é, todos os armazéns tem o mesmo custo unitário, independente do tamanho.

Mesmo ao custo mínimo, não foi atrativo armaze-

CONFIGURACAO N. 1 FASE 1 TOL: 0.002

LOCAL	ARCO	CUSTO UNIT. USADO	VOLUME DINAMICO	CAPACIDADE ESTATICA	CUSTO UNIT. REAL	CUSTO TOTAL OBTIDO	CUSTO TOTAL REAL	ERRO RELATIVO
BOM JARDIM	25- 38	25.00	21782.0	19801.8	53.34	495045.45	1056207.95	0.5313
LGO. PEDRA	26- 39	25.00	14872.0	13520.0	56.92	338000.00	769600.00	0.5608
MONCAO	27- 40	25.00	39459.0	35871.8	49.88	896795.45	1789401.70	0.4988
PINDARE MI	28- 41	25.00	0.0	0.0	351.13	0.00	0.00	0.0000
STA INES	29- 42	25.00	20198.0	18361.8	53.94	459045.45	990507.95	0.5366
STA LUZIA	30- 43	25.00	66000.0	60000.0	48.17	1500000.00	2890250.00	0.4810
VIT FREIRE	31- 44	25.00	30801.0	28000.9	51.08	700022.73	1430291.48	0.5106
BACABAL	32- 45	25.00	43052.0	40235.5	49.42	1005887.85	1988495.33	0.4941
IPIXUNA	33- 46	25.00	6644.0	6209.3	70.23	155233.64	436051.40	0.6440
OLHO DAGUA	34- 47	25.00	22747.0	21258.9	52.81	531471.96	1122686.33	0.5266
PEDREIRAS	35- 48	25.00	12842.0	12001.9	58.35	300046.73	700335.28	0.5716
P.DE PEDRA	36- 49	25.00	31748.0	29671.0	50.77	741775.70	1506490.65	0.5076
ST A LOPES	37- 50	25.00	22214.0	20760.7	52.98	519018.69	1099959.11	0.5281

DIAGNOSTICO: ERRO RELATIVO NAO TOLERADO.CUSTOS SERAO REAVALIADOS

CONFIGURACAO N. 1						FASE 1		TOL: 0.002
LOCAL	ARCO	CUSTO UNIT. USADO	VOLUME DINAMICO	CAPACIDADE ESTATICA	CUSTO UNIT. REAL	CUSTO TOTAL OBTIDO	CUSTO TOTAL REAL	ERRO RELATIVO
BOM JARDIM	25- 38	53.34	21782.0	19801.8	53.34	1056207.95	1056207.95	0.0000
LGO. PEDRA	26- 39	56.92	14872.0	13520.0	56.92	769600.00	769600.00	0.0000
MONCAO	27- 40	49.88	39459.0	35871.8	49.88	1789401.70	1789401.70	0.0000
PINDARE MI	28- 41	351.13	0.0	0.0	351.13	0.00	0.00	0.0000
STA INES	29- 42	53.94	20198.0	18361.8	53.94	990507.95	990507.95	0.0000
STA LUZIA	30- 43	48.17	66000.0	60000.0	48.17	2890250.00	2890250.00	0.0000
VIT FREIRE	31- 44	51.08	30801.0	28000.9	51.08	1430291.48	1430291.48	0.0000
BACABAL	32- 45	49.42	49696.0	46444.9	48.91	2295369.87	2271796.73	0.0104
IPIXUNA	33- 46	70.23	0.0	0.0	351.13	0.00	0.00	0.0000
OLHO DAGUA	34- 47	52.81	22747.0	21258.9	52.81	1122686.33	1122686.33	0.0000
PEDREIRAS	35- 48	58.35	12842.0	12001.9	58.35	700335.28	700335.28	0.0000
P.DE PEDRA	36- 49	50.77	31748.0	29671.0	50.77	1506490.65	1506490.65	0.0000
ST A LOPES	37- 50	52.98	22214.0	20760.7	52.98	1099959.11	1099959.11	0.0000

DIAGNOSTICO: ERRO RELATIVO NAO TOLERADO.CUSTOS SERAO REAVALIADOS

CONFIGURACAO N. 1 FASE 1 TOL: 0.002

LOCAL	ARCO	CUSTO UNIT. USADO	VOLUME DINAMICO	CAPACIDADE ESTATICA	CUSTO UNIT. REAL	CUSTO TOTAL OBTIDO	CUSTO TOTAL REAL	ERRO RELATIVO
BOM JARDIM	25- 38	53.34	21782.0	19801.8	53.34	1056207.95	1056207.95	0.0000
LGO. PEDRA	26- 39	56.92	14872.0	13520.0	56.92	769600.00	769600.00	0.0000
MONCAO	27- 40	49.88	39459.0	35871.8	49.88	1789401.70	1789401.70	0.0000
PINDARE MI	28- 41	351.13	0.0	0.0	351.13	0.00	0.00	0.0000
STA INES	29- 42	53.94	20198.0	18361.8	53.94	990507.95	990507.95	0.0000
STA LUZIA	30- 43	48.17	66000.0	60000.0	48.17	2890250.00	2890250.00	0.0000
VIT FREIRE	31- 44	51.08	30801.0	28000.9	51.08	1430291.48	1430291.48	0.0000
BACABAL	32- 45	48.91	49696.0	46444.9	48.91	2271796.73	2271796.73	0.0000
IPIXUNA	33- 46	351.13	0.0	0.0	351.13	0.00	0.00	0.0000
OLHO DAGUA	34- 47	52.81	22747.0	21258.9	52.81	1122686.33	1122686.33	0.0000
PEDREIRAS	35- 48	58.35	12842.0	12001.9	58.35	700335.28	700335.28	0.0000
P.DE PEDRA	36- 49	50.77	31748.0	29671.0	50.77	1506490.65	1506490.65	0.0000
ST A LOPES	37- 50	52.98	22214.0	20760.7	52.98	1099959.11	1099959.11	0.0000

CUSTOS EM TRANSPORTES POR PERIODO 142907176.00

AMORTIZACAO DO INVESTIMENTO EM ARMAZENAGEM: 15627527.20

CUSTO TOTAL POR PERIODO DA SOL. ENCONTRADA: 158534703.20

DIAGNOSTICO: SOLUCAO OTIMA PARA O PROBLEMA DE DISTRIBUICAO COMPATIVEL COM CURVA DE CUSTOS * SOLUCAO OTIMA LOCAL

nar em Pindaré Mirim que teve um volume nulo na solução da solução inicial gerada pelo "out-of-kilter".

Por outro lado, verifica-se, ainda na primeira solução, que os armazéns com baixo movimento (como Ipixuna) funcionam a custos unitários altos, enquanto os armazéns grandes (como Bacabal), devido as economias de escala funcionam com pequenos custos unitários.

Na segunda iteração, os custos reais $CR_{i,j}$ obtidos a partir da solução inicial são usados como custos unitários de armazenagem. Como resultado, Ipixuna, que tinha custos unitários muito altos, foi desativado, Bacabal, que se mostrava atrativo devido ao baixo custo, teve sua capacidade aumentada.

Em termos práticos, isso mostra que as economias de escala tornaram mais atraente pagar o custo de transporte da produção de Ipixuna para Bacabal, pois a economia em custos de armazenagem foi compensadora.

A terceira iteração não alterou a solução, mas foi executada automaticamente pelo programa de modo a obter uma solução onde todos os erros fossem toleráveis.

3.4 - O MÉTODO PROPOSTO

A solução obtida pela rotina EQUILIBRAR não é necessáriamente a solução ótima para o problema de localização.

Em geral, se iniciarmos a rotina com estruturas de custos diferentes, obteremos soluções de equilíbrio diferentes. Assim,seria interessante poder pesquisar novas soluções de equilíbrio na busca de uma solução ótima.

Por outro lado, a técnica de "DROP" empregada por Feldman, Lehn e Ray permite o exame de várias soluções,mas seu critério de avaliação de custos pode ser impreciso.

Para contornar essas dificuldades, foi desenvolvido neste trabalho um programa computacional com as seguintes características:

1 - A solução para a configuração inicial será obtida através da rotina EQUILIBRAR, descrita na seção 3.2. Desse modo, a estrutura de custos obtida nessa solução será em geral avaliada com precisão.

2 - A seguir, várias configurações da rede de armazenagem são examinadas, através da técnica de "DROP". Para se excluir o armazem k, basta que ao k-ésimo arco (i,j) do conjunto H seja dado um custo c_{ij} extremamente alto, e os demais custos de armazéns sejam mantidos como na solução de equilíbrio anterior.

3 - Cada vez que a estrutura de custos é alterada (exclusões são feitas por alteração de custo) a rotina EQUILIBRAR (que executa o método iterativo) é chamada para obter nova situação de equilíbrio entre fluxos e custos. Dessa maneira o programa pode avaliar com precisão o custo de uma configuração qualquer, e

compará-lo com o custo de outras configurações, na busca da melhor solução.

Como já foi mencionado, a solução obtida pela rotina EQUILIBRAR depende da estrutura de custos inicial, isto é, aquela usada no primeira iteração. Deste ponto de vista, nosso programa pode ser encarado como uma extensão do método de King e Logan, onde várias soluções de equilíbrio são obtidas a partir de diferentes estruturas de custo iniciais. Cada uma das estruturas de custos iniciais (exceto a primeira) forçará o fechamento de 1 ou mais armazéns.

De outro ponto de vista, o programa pode ser encarado como uma aplicação da técnica de "DROP", onde em cada configuração os custos são avaliados com precisão.

Ao invés da técnica de "DROP" poderíamos também ter utilizado o "ADD", que consiste, como já visto na seção 2.2, em adicionar armazéns à rede de distribuição. A prática no entanto mostrou que a rotina EQUILIBRAR, que na primeira configuração começa com todos os armazéns ativos, não é capaz de eliminar por si só um número significativo de armazéns. Perdem se assim as vantagens advindas de uma economia de escala onde a tendência seria trabalhar com um pequeno número de armazéns de grande capacidade. Face a esta característica da rotina EQUILIBRAR, procuramos corrigir este fato adotando a rotina "DROP" ao invés da rotina "ADD".

3.5. DESCRIÇÃO DO PROGRAMA HEURÍSTICO

A figura 3.9 mostra a estrutura geral do programa em macro-blocos. Antes de descreve-la em detalhe, vamos definir os seguintes identificadores:

SBEST - identificador para a "melhor solução obtida até agora", em qualquer fase do programa. Entenda-se por "solução" um conjunto de fluxos x_{ij}, $(i,j)\varepsilon A$ e um conjunto de custos c_{ij}, $(i,j)\varepsilon H$. Portanto, SBEST não é uma variável, mas um nome dado a um conjunto de resultados.

ZBEST - nome da variável que contém o custo total da solução SBEST.

SCONF - Solução de equilíbrio obtida para determinada configuração, pela rotina EQUILIBRAR. Assim como SBEST, não se trata de uma variável, mas um conjunto de resultados.

CTT - nome da variável que contém o custo total associado a SCONF.

Numa primeira fase (bloco 1), é feita a leitura das opções da corrida (ver anexo 1) e dos dados necessários. Deve-se notar que os custos referentes a arcos representando armazéns (lidos nesta fase) são todos iguais a um valor mínimo, exceto para o caso descrito na seção 3.6. (armazéns modulados).

FLUXO DO PROGRAMA HEURISTICO

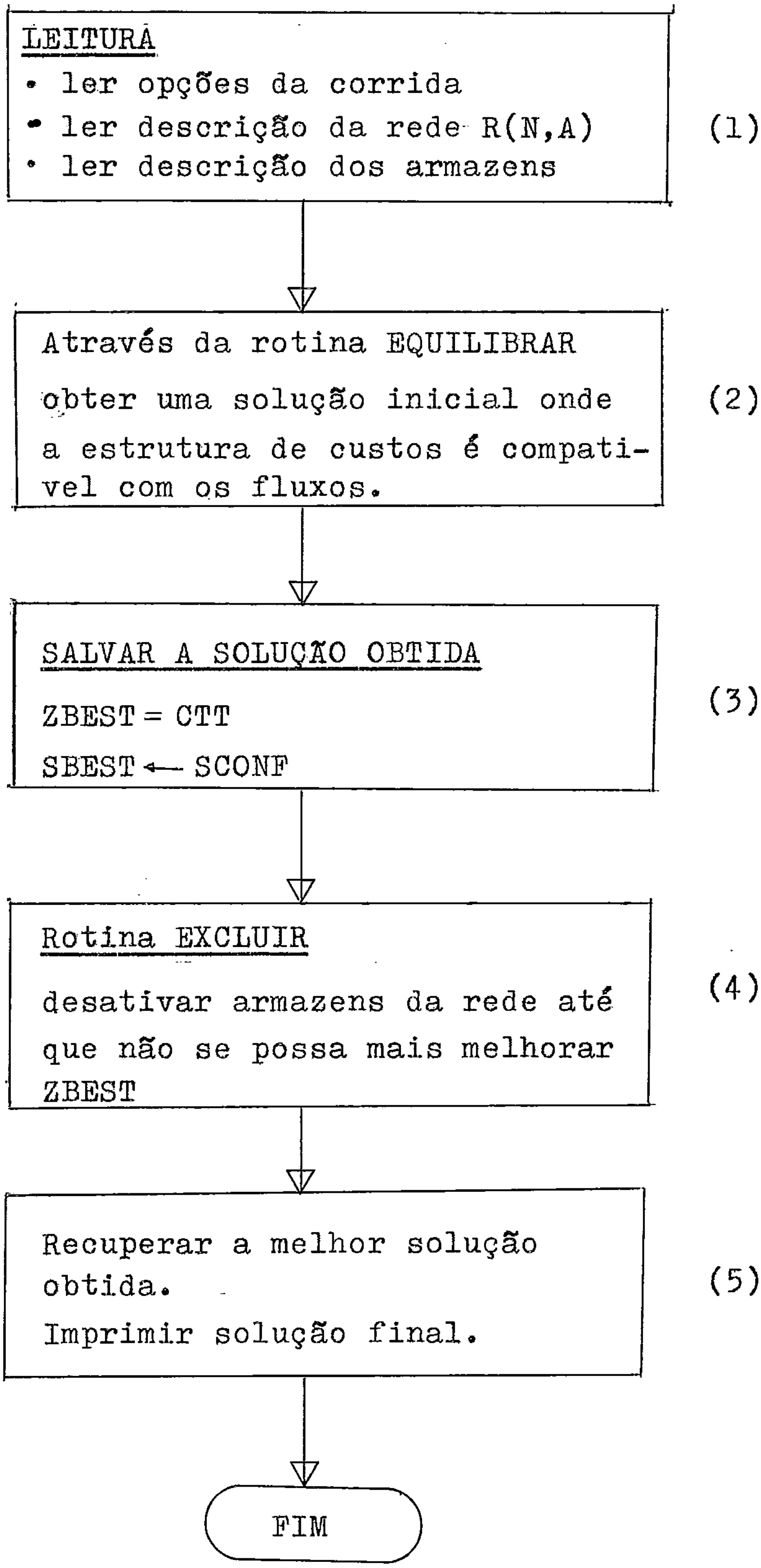


Figura 3.9

Na segunda fase (blocos 2 e 3), através da rotina EQUILIBRAR, o programa obtém uma solução inicial para o problema, onde os custos usados são coerentes com os custos usados para obte-la. Note-se que a rotina EQUILIBRAR inicia com custos unitários de armazenagem mínimos (lidos). A solução obtida pela rotina EQUILIBRAR é aceita como "melhor solução até agora", SBEST. O custo total dessa configuração é salvo, na variável ZBEST.

Na terceira fase (bloco 4), a rotina EXCLUIR é executada. Essa rotina tenta excluir armazéns da rede (o que equivale a desativar armazéns ativos na última solução aceita), um de cada vez, até que nenhuma exclusão possa diminuir o valor de ZBEST. Durante a série de exclusões, a escolha do próximo armazém a ser excluído é feita segundo princípios heurísticos, como veremos adiante.

Na quarta e última fase, o programa recupera a melhor solução de equilíbrio já obtida e esta solução é impressa.

A técnica de "DROP" é, portanto, executada através da rotina EXCLUIR (ver anexo 3). Devido a sua complexidade, damos a seguir uma descrição detalhada da mesma:

1 - No início de cada estágio, são selecionados dentre os armazéns ativos, os N melhores candidatos a uma exclusão. Entanda-se por "melhores candidatos" os armazéns ativos que se mostrem mais promissores quanto a uma redução no custo total, segundo a heurísti-

ca adotada. No nosso programa no início de cada estágio, são candidatos a exclusão os N armazéns de maior custo unitário. Estes candidatos são guardados em um "buffer".

2 - A seguir, cada um dos N candidatos é excluído da rede temporariamente. Essa exclusão é feita atribuindo-se um custo unitário extremamente alto ao arco correspondente ao armazém. Como isso significa uma alteração na estrutura de custos, a rotina EQUILIBRAR pode ser usada para obter uma solução de equilíbrio entre fluxos e custos para a nova configuração. Se a exclusão temporária de um determinado armzém produzir uma solução pior do que a última solução SBEST (que existia no início do estágio), esse armazém é marcado(*) e não será mais considerado para exclusão nos estágios seguintes.

3 - Ao fim de cada estágio, se alguma das configurações temporárias teve um custo total inferior a ZBEST, então a de menor custo é adotado como SBEST. Note-se que somente aqui uma exclusão é feita em carater definitivo. O novo ZBEST é o custo da nova SBEST.

Os ítens 1, 2 e 3 acima descrevem um estágio da rotina excluir. Ao fim de cada estágio, a rotina tenta selecionar novos candidatos a exclusão e novo estágio é iniciado. Em cada estágio, no máximo 1 armazém é excluído definitivamente.

(*) No fluxograma, este armazém é tratado como um armazém MARCADO.

A rotina excluir executa vários estágios, até que não seja mais possível colocar nenhum candidato no "buffer". Isto se dá quando todos os armazéns ativos estiverem marcados.

A figura 3.10 mostra um fluxograma da rotina EXCLUIR, onde

Sciclo - melhor solução já obtida no estágio

Zciclo - custo total associado a Sciclo

N - tamanho do "buffer". Este valor é escolhido pelo usuário e fornecido ao programa como uma das opções da corrida (ver anexo 1)

Alguns comentários sobre o fluxograma da rotina EXCLUIR:

bloco (1) - São colocados no buffer os N "melhores" candidatos à exclusão no próximo estágio.
A seleção dos candidatos é feita pelos critérios heurísticos já definidos.

bloco (2) - Se no bloco (1) não foi encontrado nehum candidato à exclusão, a rotina 'excluir' termina.

bloco (3) - Inicia-se um ESTÁGIO de exclusão: é atribuído a zciclo o valor da melhor solução até agora. Isto para que só sejam consideradas interessantes as configurações que, dentro desse estágio, melhorarem a solução atual(ver bloco (5)).

Rotina "EXCLUIR"

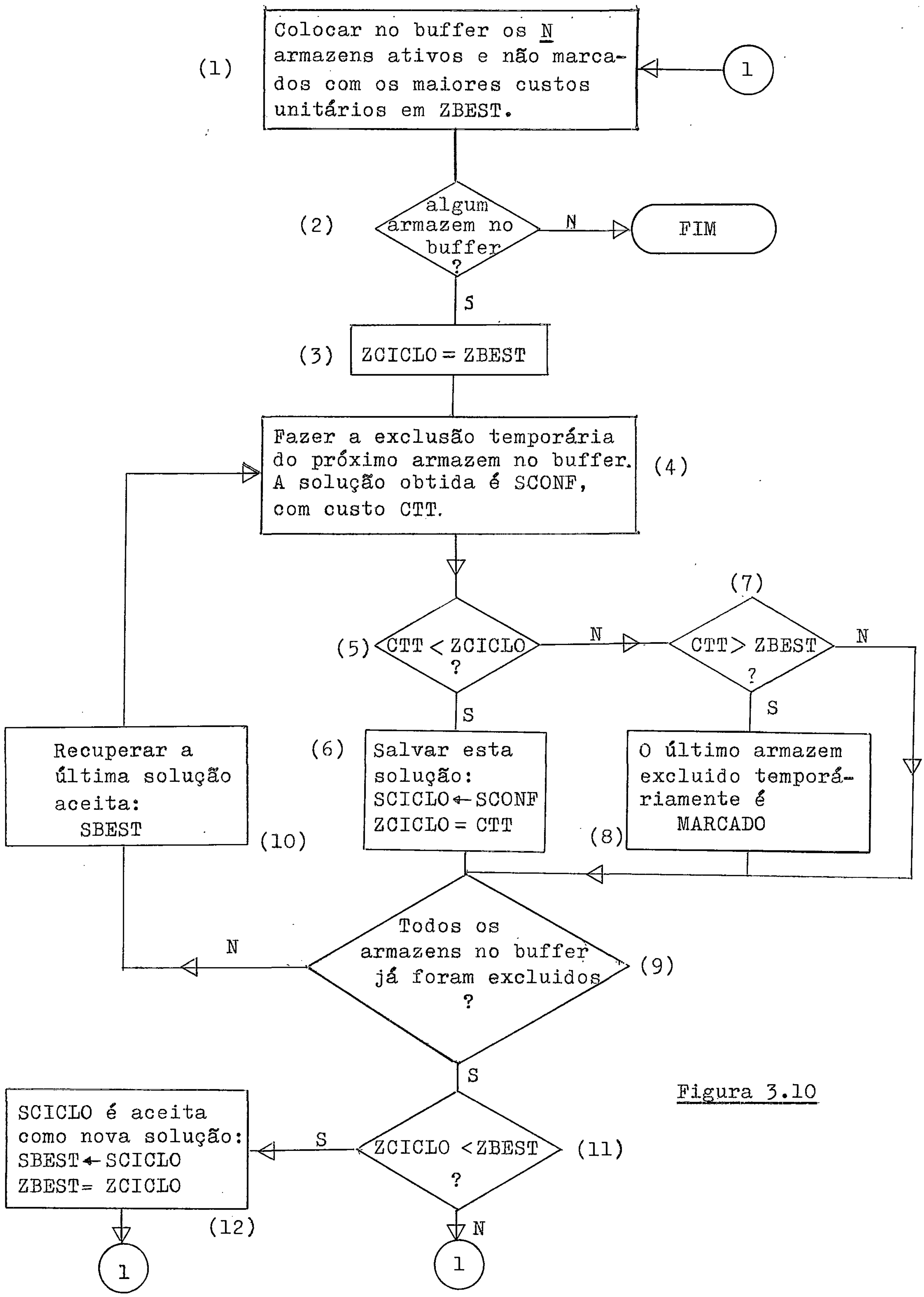


Figura 3.10

bloco (4) - Inicialmente, o custo unitário relativo ao armazém a ser excluído é aumentado, e todos os outros custos são mantidos como em ZBEST. (Logo, temos nova estrutura de custos)
Para obter uma solução (SCONF) para esta configuração o programa tem duas opções:

1 - Resolver, através da rotina "out-of-kilter", o problema descrito por MM5 segundo a estrutura de custos definida acima.

2 - Usar essa estrutura de custos como custos iniciais para a rotina EQUILIBRAR, e obter uma solução SCONF onde custos são compatíveis com fluxos.

blocos(5) e (6) - Se o custo da solução SCONF (CTT) é menor que o da melhor solução já obtida no estágio (sciclo), então SCONF é guardado como nova sciclo.

blocos(7) e (8) - Se SCONF é pior que SBEST, então o armazém que foi excluído temporariamente em SCONF é MARCADO, isto é, não será mais candidato à exclusão nos estágios seguintes.

bloco (9) - testa fim de um estágio de exclusão

bloco (10) - No bloco 4, a estrutura de custos de SBEST foi alterada. No entanto, a estrutura de custos da próxima configuração temporária deve ser obtida a partir da SBEST que existia no início do estágio. Portanto, SBEST é reconstituído nesse bloco, para que a partir dela sejam construídas configurações temporárias.

bloco (11) - Testa se ao fim do estágio de exclusão alguma configuração temporária teve custo menor que a solução que havia no início do estágio. Caso negativo, novo buffer será preenchido.

bloco (12) - Se alguma solução do estágio de exclusão foi melhor que SBEST, então a melhor delas é aceita como nova SBEST. Note que somente agora SBEST é alterada definitivamente.

Em sua primeira versão o programa chamava, antes da rotina EXCLUIR, uma outra rotina chamada AGREGAR.

Essa rotina, a exemplo de EXCLUIR, examinava novas configurações da rede de armazéns, usando a mesma técnica de avaliação de custos.

O funcionamento da rotina AGREGAR é resumidamente o seguinte:

1 - São definidos pelo usuário, através de cartões de dados, alguns conjuntos de locais vizinhos.

Esses conjuntos de vizinhos são determinados pela proximidade geográfica de um subconjunto de locais candidatos, ou por sua interdependencia econômica.

2 - A partir da primeira solução de equilíbrio obtida, o programa tenta agregar todos os armazéns ativos de um grupo em um só armazém, na expectativa de ganhos compensadores devido a economias de escala. Por exemplo, seja um grupo definido pelo locais A, B e C. Se os 3 locais tem armazéns ativos na primeira configuração, a rotina examinará os custos em 3 novas configurações. Em cada uma delas, serão dados custos de armazenagem elevados a 2 dos locais, obrigando o fluxo a se concentrar em 1 deles.

A rotina AGREGAR foi abandonada em nosso programa, por não estar proporcionando resultados melhores do que o uso da rotina excluir somente. Isto provavelmente ocorreu porque, nos problemas que examinamos, não havia um conjunto de locais candidatos próximos economicamente equivalentes.

Nos casos em que tais conjuntos existem, este procedimento pode se mostrar útil.

De modo geral, várias estratégias podem ser definidas para o exame de diversas configurações da rede, dependendo da heurística adotada.

A nosso ver, será sempre importante que o custo de cada configuração seja avaliado com precisão, e a rotina E-

QUILIBRAR se apresenta como um bom instrumento para tal.

3.6. UMA ADAPTAÇÃO PARA TRATAMENTO DE ARMAZÉNS MODULADOS

Armazéns agrícolas são frequentemente constituídos de silos. Muitas vezes, como no caso de silos metálicos, estes são pré-fabricados, e somente alguns tamanhos padronizados são disponíveis.

Suponhamos então um problema onde a demanda de armazenagem em certo local seja de 7112 toneladas mas só se dispõe de silos de 6000 ou 3000 toneladas. Nesse caso, seria necessário construir um armazém com 2 silos, um de 6000 e outro de 3000 toneladas, incorrendo-se num custo correspondente a 9000 toneladas. Entretanto, quando toda uma rede de armazéns está sendo dimensionada, é necessário avaliar a possibilidade de distribuir a produção por diversos armazéns, de modo a reduzir o investimento em capacidade ociosa de armazenagem, em contrapartida aos gastos com transportes.

Como geralmente existem custos fixos, o problema é complicado, podendo ser formalizado no modelo MM3, apresentado no capítulo 1.

Apresentamos a seguir uma adaptação do método heurístico desenvolvido a este tipo de condição.

Suponhamos que só é possível contruir armazéns com um número inteiro de silos de certo tipo, com b toneladas de capacidade para cada silo.

Para cada armazém, existe um custo fixo de F cruzeiros e mais S cruzeiros por cada silo instalado. Note-se que S independe do fluxo em cada silo, não sendo considerados os custos que variam com a quantidade armazenada. Levamos em conta somente o custo fixo F do armazém e o custo fixo S de cada silo.

Numa rede, cada armazém pode ser representado por 2 nós ligados por um conjunto de arcos com capacidade de fluxo b.

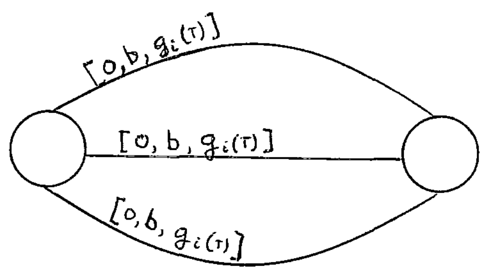


Figura 3.11

O número de arcos ligando os 2 nós depende do número máximo de silos que se permite em cada armazém.

A função de custo unitário no primeiro arco é $g_1(T_1) = \frac{F + S}{T_1}$ onde T_1 é o fluxo nesse arco (ignoramos índices de rotação)

Nos demais arcos, a função de custo é

$$g_j(T_j) = \frac{S}{T_j}$$

Ao adotarmos esta definição de custos, estamos supondo que só haverá fluxo no 2º arco depois que o primeiro es tiver saturado, ou seja, o custo fixo de um armazém é pago ao se utilizar o primeiro silo do armazém.

De fato, ao desenvolvermos um método para o trato deste problema, devemos ter em mente duas características necessárias a ele:

1 - Se a um armazém estão associados os arcos k, k+1,... k+n, ligando os nós i e j, então só será permitido haver fluxo no arco $k+\ell$ quando o arco $k+\ell-1$ estiver saturado. $(1 \leq \ell \leq n)$

2 - O método deve permitir o uso de uma estrutura de custos que reflita a desvantagem existente na utilização ociosa de um silo.

A estrutura de custo referida é determinada por funções de custo unitário da forma $g(T) = \frac{K}{T}$ para cada silo.

Na figura 3.12, é mostrado o aspecto de tais funções, para um silo de capacidade b.

Para evitar custos tendendo a infinito e os consequentes problemas computacionais, limitamos o domínio das funções de custo ao intervalo $[1,b]$. Dessa forma, o custo máximo que pode ocorrer em um arco-silo é aquele correspondente a um fluxo unitário no mesmo.

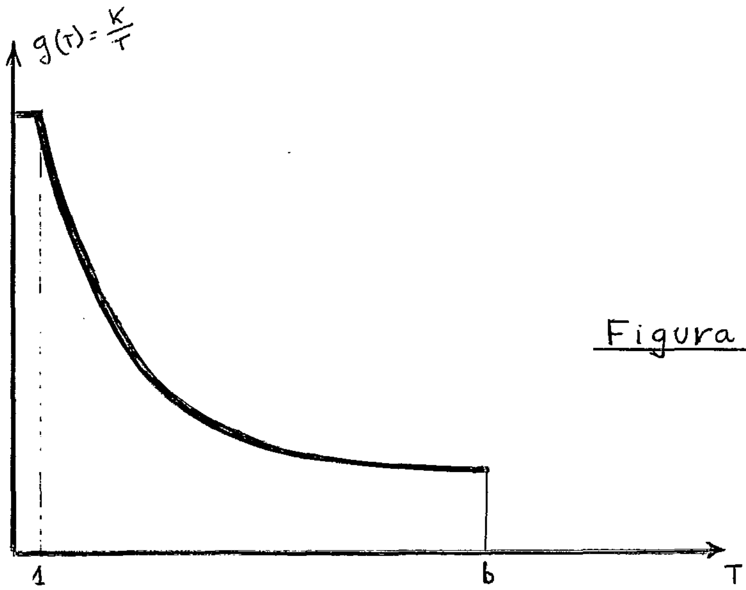


Figura 3.12

Da observação da figura 3.12 é fácil concluir que a utilização ociosa de um silo é desistimulada pelo alto custo unitário, ao passo que sua plena utilização torna-se atrativa em vista do baixo custo. Dessa forma, se um silo estiver sendo utilizado para operar com apenas 1/5 de sua capacidade, o custo unitário será suficientemente alto para que se pague o custo total do silo inteiro.

Portanto, funções de custo unitário do tipo $g(T) = \frac{K}{T}$ são suficientes para dar ao método a característica 2.

Quanto à primeira característica, algumas alterações são necessárias no programa. Seja $g_k^{\ell}(T_k^{\ell})$ a função de custo do ℓ-ésimo arco do k-ésimo armazém. Então devemos adaptar o programa para superar as seguintes dificuldades:

(a) $g_k^1(T_k^1) \geq g_k^{\ell}(T_k^{\ell})$ para $\ell \neq 1$, isto é, a função de custo no primeiro arco de um armazém será em geral maior que nos demais, devido ao custo fixo F do armazém. Isto significa que o algoritmo "out-of-kilter" deverá fornecer soluções onde os demais arcos de um armazém sejam utilizados antes do primeiro, o que é indesejável.

A figura 3.13 mostra as funções de custo unitário para os 2 primeiros arcos de um armazém, onde na primeira o custo fixo está incluído. Suponhamos que a rotina EQUILIBRAR esteja sendo aplicada ao problema. Seja nulo o fluxo no segundo arco, o que corresponderia a uma situação em que o segundo silo não estaria sendo utilizado. Suponhamos agora que o fluxo no primeiro arco é $T_k^1 < T_a$. Evidentemente o custo neste primeiro arco será superior ao do arco seguinte (ainda sem uso), o que certamente causará a transferência do fluxo do primeiro arco para o segundo, na próxima iteração.

Para contornar esta dificuldade, sugerimos iniciar o programa com uma estrutura de custos especial.

Ao primeiro arco de cada armazém é dado um custo inicial C^0. No nosso caso, escolhemos para C^0 o custo unitário inicial ao custo do arco de maior função de custo, operando com um fluxo de apenas uma tonelada, ou seja

$$C^0 = \frac{F + S}{1} = F + S$$

Para os demais arcos de cada armazém são dados custos unitário iniciais de C^0+E, C^0+2E,...C^0+nE, onde E é um valor pequeno qualquer(*).

(*) Em nossas experiências computacionais usamos E = 0,1 quando C^0 = 660.000,00.

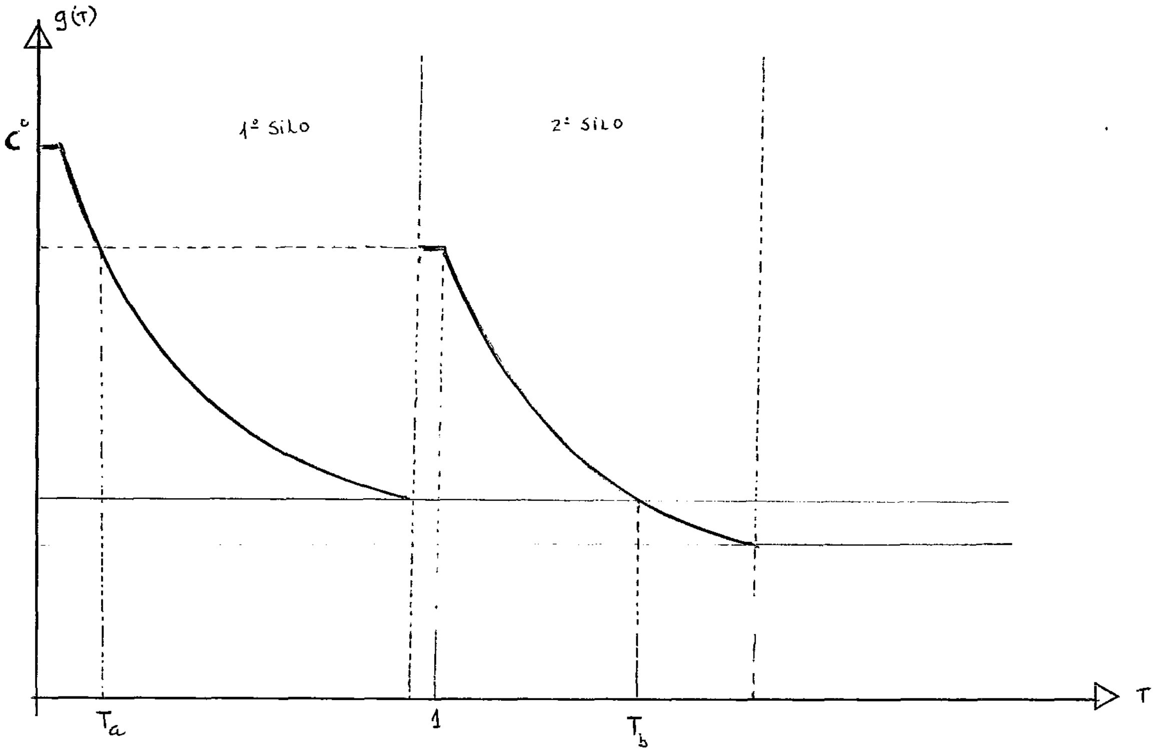


FIGURA 3.13

Dessa forma, garantimos que qualquer que seja o fluxo em um determinado silo, seu custo unitário será inferior ao custo em um silo de ordem superior no mesmo armazém e ainda não utilizado.

(b) Uma segunda dificuldade existente pode ser descrita pela seguinte situação: o primeiro arco está saturado e o segundo arco opera com um fluxo $T_k^2 > T_b$ (figura 3.13) ao fim de determinada iteração da rotina EQUILIBRAR.

Nesse caso, o custo unitário no segundo arco é inferior ao do primeiro, o que acarreta que na próxima iteração o algoritmo "out-of-kilter" deve transferir para o segundo arco parte do fluxo do primeiro, o que é indesejável.

Esta dificuldade pode ser contornada com algumas alterações da subrotina de avaliação de custos (passo 2 da rotina EQUILIBRAR). A principal alteração consiste em "trancar" os fluxos nos arcos saturados de um armazém onde algum arco tenha custo unitário maior que outro arco de ordem superior.

Uma nova subrotina de avaliação de custos foi codificada e agregada ao programa, a ser usada somente em caso de armazéns modulados. Um fluxograma desta subrotina é mostrado na figura 3.14, onde:

T_k^ℓ - fluxo no ℓ-ésimo arco (silo) do armazém k.

b_k^ℓ - capacidade máxima de fluxo em um silo (k,ℓ).

a_k^ℓ - limite inferior de fluxo no arco correspondente ao silo (k,ℓ).

L(k) - número de silos no armazém k

NA - número de armazéns.

Os demais símbolos usados tem o mesmo significado do usado na descrição da rotina EQUILIBRAR (seção 3.2) usando-se agora os índices k para armazém e ℓ para silo.

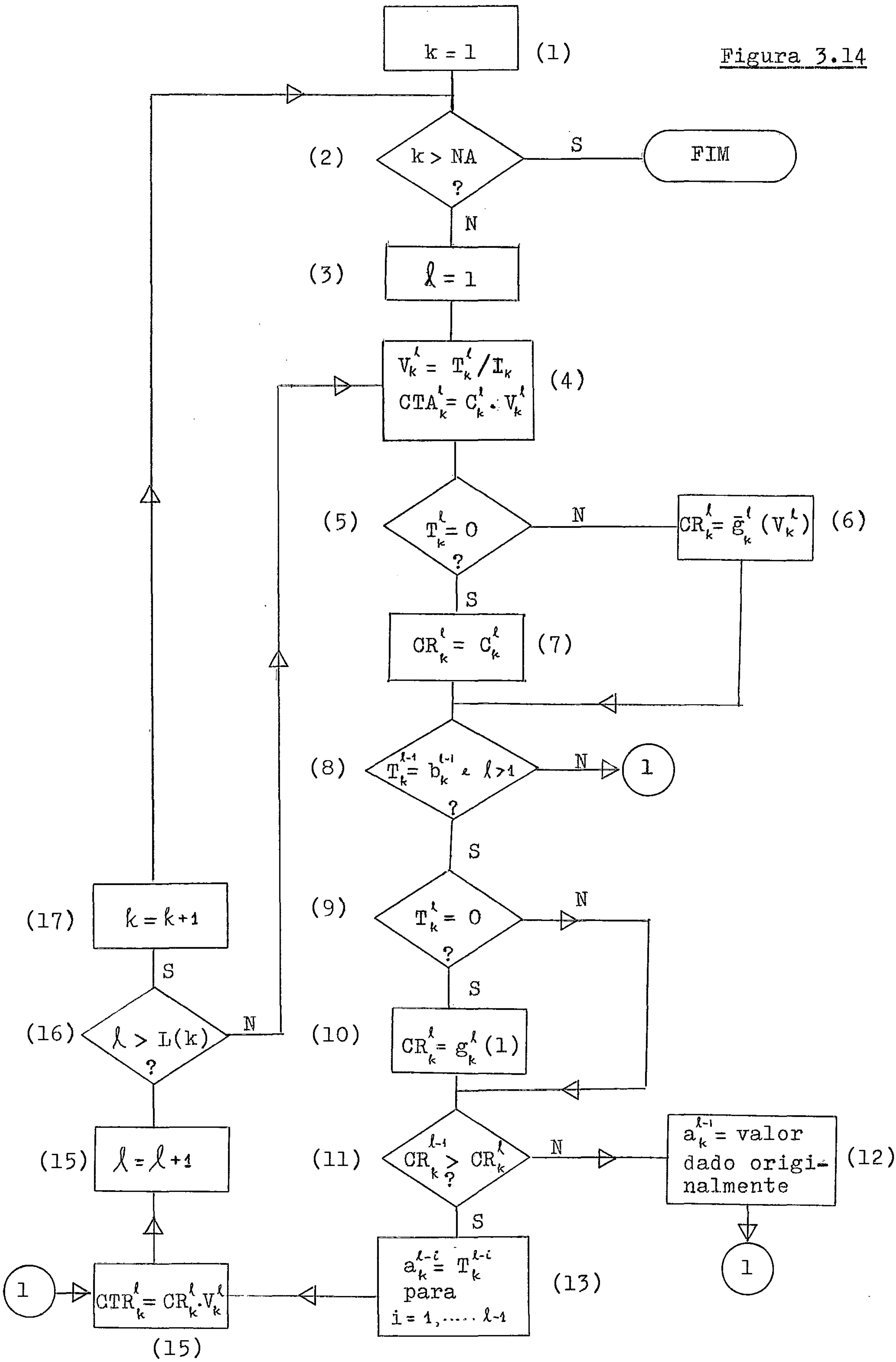
As principais modificações em relação ao passo 2 descrito na seção 3.2 são detalhadas a seguir:

blocos (5) e (6) e (7)

O custo em um arco só será revisado quando houver fluxo nele, ou quando o arco antecessor (do mesmo armazém) estiver saturado (blocos 8 e 9). Caso contrário, se um silo (k,ℓ) ainda não foi utilizado, o arco correspondente permanece

PASSO 2 MODIFICADO

Figura 3.14



com o custo inicial $C_k^{\ell} = C^0 + (\ell-1)E$

bloco (8)

Se o arco que antecede ao arco sendo examinado (do mesmo armazém) está saturado, então as seguintes medidas especiais são tomadas:

blocos (9) e (10)

Se o silo examinado está fora de operação, ao arco correspondente é dado o custo máximo "real" $\bar{g}_k^{\ell}(1)$, abandonando-se o custo máximo artificial $C^0 + (\ell-1)E$

blocos (11), (12) e (13)

Se o custo no arco examinado é inferior ao do arco antecessor (do mesmo armazém) então os fluxos em todos os arcos antecessores nesse armazém serão "trancados" através da imposição de um limite inferior de fluxo (bloco 12). Caso contrário, ao limite inferior de fluxo é dado o seu valor original, lido no início do programa (bloco 11). Com este procedimento, contorna-se a dificuldade (b).

No programa, o passo 2 da rotina EQUILIBRAR pode ser executado por duas subrotinas: A primeira, chamada REAVALIAR, é utilizada normalmente como "default". A segunda, chamada REVER, executa o passo 2 modificado, quando são considerados armazéns modulados. (ver anexo 3)

É evidente que no caso de armazéns modulados o conjunto H de arcos representando armazéns é extendido, passando a conter todos os arcos de todos os armazéns. Portanto, na rotina EQUILIBRAR, as operações efetuadas "para todo $(i,j)\varepsilon H$" incluem todos os arcos representando silos, e as variáveis anteriormente indexadas por k, passam a ter os índices k e ℓ.

Uma outra alteração necessária foi quanto ao conceito de "solução" obtida pelo programa (SBEST,sciclo,SCONF). Como já foi dito, uma solução é um conjunto de resultados obtidos para determinada configuração, incluindo fluxos e custos unitários de armazenagem. Assim quando era necessário salvar uma solução (isto é, guardar na memória para uso posterior), um conjunto de valores devia ser armazenado.

Como agora os limites inferiores de fluxo nos arcos correspondentes a silo podem ser alterados ao longo das iterações, tais limites de fluxo passam a fazer parte do conjunto de valores correspondente a uma solução.

As adaptações já mencionadas para o trato de armazéns modulares não causaram grandes mudanças no programa. No entanto, é necessário que o usuário informe, através do cartão de "opções da corrida" (ver anexo 1), que os armazéns devem ser tratados como modulares, para que as rotinas especiais de cálculo sejam acionadas.

3.7. RESULTADOS

Para o teste do programa, foi definido um problema de localização de armazéns para arroz em duas micro-regiões do Maranhão (números 34 e 35, segundo a classificação da CIBRAZEM). Os dados sobre saldos comercializáveis, armazéns existentes, índices de rotação e custos de armazéns foram obtidos de um estudo já feito pela CIBRAZEM no Maranhão |4|.

Custos de transportes foram inferidos a partir de dados da Tabela Nacional de Fretes, emitida pelo Sindicato das Empresas de Transporte de Carga do Rio de Janeiro. Eventuais imprecisões nos dados não nos causaram preocupação, uma vez que o objetivo principal dos testes foi verificar a eficiência computacional do programa.

Para que se obtivesse uma certa variedade de condições de teste, foram construidas tres redes de distribuição, todas com 23 centros de produção e 5 centros de demanda. Na primeira, foram realizados testes gerais de performance e convergencia sob diversas condições de custos de armazenagem e opções do programa.

Na segunda, foi testado o caso de armazéns modulares, com existência de custos fixos.

Na terceira, tentou-se estabelecer as mesmas condições utilizadas no estudo da CIBRAZEM, para uma comparação de resultados.

Como centros de produção, foram considerados 9 municípios da micro-região 34 e 14 municípios da micro-região 35. Como oferta em cada município, foi usada a projeção para 1976 dos saldos comercializáveis, obtido a partir de regressão linear com dados dos últimos anos. É claro que tais ofertas não fazem sentido num estudo feito em 1977, mas como já foi dito, o objetivo dos testes foi principalmente testar o programa.

A produção total oferecida é de 317487 toneladas/ano, que deve fluir, através dos armazéns, para 5 centros de demandas fixados em São Luiz (35%), Timon (30%), Candido Mendes (20%), Forto Franco (10%) e Imperatriz (5%). Esta distribuição da demanda foi arbitrada, por não haver informação precisa disponível.

Na primeira rede testada, foram escolhidos 13 locais candidatos, ou seja, locais considerados potencialmente atrativos para a localização de armazéns. Tais locais foram as redes dos municípios de Bom Jardim, Lago da pedra, Monção, Pindaré Mirim, Santa Ines, Santa Luzia, Vitorino Freire, Bacabal, Ipixuna, Olho d'água das cunhas, Pedreiras, Poção de Pedras e Santo Antonio dos Lopes.

Para esta primeira rede, foram executados 5 testes, sendo cada teste um problema diferente, obtido pela variação de custos de armazenagem, parâmetros do problema ou opções do programa.

A fim de verificar a convergencia da rotina EQUILIBRAR, foram construidas 4 funções de custo unitário diferentes, a saber:

1 - $\bar{g}_1(v) = \dfrac{1500000 + 650.V}{8V}$

2 - $\bar{g}_2(v) = \dfrac{1500000 + 95000.\sqrt{V}}{8V}$

3 - $\bar{g}_3(v) = \dfrac{1722000 + 52530.\sqrt{V}}{8V}$

4 - $$\bar{g}_4(v) = \begin{cases} \dfrac{300.000 + 50 \, . \, V}{V}, & \text{se } 0 < V \leq 8000 \\ \dfrac{600.000 + 12{,}5 \, . V}{V}, & \text{se } 8000 < V \leq 30.000 \\ \dfrac{800.000 + 5{,}7 \; V}{V}, & \text{se } 30.000 < V \leq 60.000 \end{cases}$$

Em todos os casos, V é a capacidade estática de armazenagem no local. Esta capacidade é obtida dividindo-se o fluxo anual no armazém pelo índice de rotação, obtido pela CIBRAZEM.

Tais índices são 1,1 e 1,07 respectivamente para as micro-regiões 34 e 35.

As funções de custo acima foram estipuladas de modo a serem razoávelmente coerentes com funções de custos mencionados em estudos da CIBRAZEM.

Foi arbitrada uma vida econômica de 8 anos para cada armazém, para que se pudesse obter custos anuais. Os cus tos de armazenagem devem ser anuais porque as ofertas e demandas são anuais e assim os fluxos e custos com transporte estarão em termos de custo por ano.

Não houve rigor na obtenção das funções de custo. No entanto, a obtenção de custos unitários que reflitam o investimento anual em cada armazém não será difícil para técnicos no assunto, embora fuja dos objetivos desse trabalho.

A função de custo unitário $\overline{g}_4(v)$ corresponde a uma função de custo total linear por segmento, e foi usada para que se pudesse comparar o resultado obtido pelo programa heurístico com o resultado obtido por um "package" de programação matemática.

REDE 1

Passamos agora à apresentação dos resultados dos testes feitos sobre a primeira rede.

Esta rede é constituída de 56 nós e 272 arcos. Uma representação esquemática é mostrada na figura 3.3.

Os resultados de cada problema resolvido são mostrados nas tabelas 3.7.1 a 3.7.4. A última linha de cada tabela dá o custo total anual de cada configuração (armazenagem + transportes). A melhor configuração obtida é marcada com (*).

PROBLEMA 1 - Neste problema, todos os armazéns tem custos unitários dados por $\overline{g}_1(v)$. Nessa função, o custo fixo é de Cr$ 1.500.000,00 e o custo marginal é de Cr$ 650,00 por tonelada. A capacidade máxima permitida em cada armazém é de 60.000 toneladas. Como opções da corrida, usou-se:

(a) tamanho do buffer = 3 tolerância: 0.0015

(b) armazéns não modulados

(c) custos de equilíbrio obtidos para todas as configurações, ou seja, cada vez que há uma exclusão, a rotina EQUILIBRAR é usada para obter custos compatíveis com fluxos.

A evolução da computação, que pode ser acompanhada na tabela 3.7.1, foi a seguinte:

1 - Solução de equilíbrio inicial obtida pela rotina EQUILIBRAR, na configuração 1. Embora a rotina inicie com todos os armazéns ativos, 3 deles foram desativados ao longo das iterações.

2 - Primeiro estágio de "DROP": Pedreiras, Santa Ines e Olho d'água são colocados no buffer. São feitas tres exclusões (configurações 2,3 e 4) sem melhoria no custo total.

3 - Segundo estágio de "DROP": Bom Jardim, Vit.Freire e St.A. Lopes são colocados no buffer, e excluídos, um de cada vez (configurações 5,6 e 7), sem melhoria da solução inicial.

4 - Terceiro estágio de "DROP": Monção, Lgo. Pedra e Bacabal no buffer. As tres exclusões (configurações 8, 9 e 10) não produziram melhoria na solução.

5 - Quarto estágio de "DROP": apenas Sta.Luzia é colocado no buffer, por ser o único armazém ativo e não marcado. Sua exclusão (configuração 11) não produz redução no custo e o programa termina. A primeira configuração é aceita como solução final e o esquema de distribuição (fluxo de transportes) correspondente a ela é impresso.

<u>PROBLEMA 2</u> - Neste problema, todos os armazéns tem função de custo unitário $\overline{g}_2(v)$. Nesse caso, economias de escala são relevantes, devido ao fator $\sqrt{V}$. A capacidade máxima permitida em cada armazém é de 60.000 t e as opções de corrida são idênticas as do problema 1.

A tabela 3.7.2 mostra os resultados parciais obtidos pelo programa, cuja evolução de processamento foi a seguinte:

1 - A solução inicial obtida pela rotina EQUILIBRAR, na configuração 1. Esta solução é aceita como SBEST, iniciando-se então a fase "DROP", com 3 estágios.

2 - Primeiro estágio: Olho d'água, Santa Ines e Bom Jardim no buffer. A exclusão de Olho d'água produz redução no custo e a configuração 2 é aceita como nova SBEST.

3 - Segundo estágio: Lgo Pedra. St.A.Lopes e Monção no buffer. Note-se que aqui são feitas a partir da configuração 2. A exclusão de Monção produz redução do custo na configuração 7 e esta é aceita como nova SBEST.

4 - Terceiro estágio: Vitorino Freire, Sta Luzia e Bacabal no buffer. Nenhuma das 3 exclusões (configurações 8, 9 e 10) produz redução no custo de SBEST.

5 - Como não há mais armazéns ativos e não marcados, o programa termina com a configuração 7 aceita como solução final.

PROBLEMA 3 - Todos os armazéns tem custos unitários dados por $\bar{g}_3(v)$ e capacidade máxima igual a 60000 t.

As opções de corrida usadas foram:

(a) Tamanho do buffer = 5

(b) armazéns não modulados

(c) Rotina EQUILIBRAR usada em todas as configurações

(d) Tolerancia para rotina EQUILIBRAR = 0.001

Os resultados obtidos para este problema podem ser vistos no anexo 2, no formato original de impressão do programa. A fase de "DROP", executada em 2 estágios, não melhorou a solução inicial obtida pela rotina EQUILIBRAR na configu ração 1. Note-se que a configuração 11 é a repetição da primeira, e é impressa pelo programa quando a "melhor solução já obtida" é recuperada.

PROBLEMA 4 - Em problemas reais, armazéns diferentes podem operar com diferentes funções de custo. Neste problema, aos armazéns da micro-região 34 foram atribuidas funções de custo $\bar{g}_3(v)$ e aos da micro-região 35, funções de custo $\bar{g}_2(v)$. Foram mantidas as capacidades máximas de 60.000 t e as opções de corrida do problema 3.

Os resultados obtidos são mostrados na tabela 3.7.3 onde se destaca:

1 - Solução inicial obtida pela rotina EQUILIBRAR na configuração 1.

2 - Primeiro estágio de exclusão com Bom Jardim, Santa Ines, St.A.Lopes, Lgo Pedra e Monção no buffer, sem melhoria. (configurações 2 a 6)

3 - Segundo estágio de exclusão com Vit.Freire, Sta Luzia e Bacabal no buffer, sem melhoria. (configurações 7 a 9)

O fato de não terem sido feitas exclusões após a configuração 1 pode ser explicado pelo apreciável número de armazéns inativos já na solução de equilíbrio inicial.

PROBLEMA 5 - Este problema foi montado para que se pudesse comparar a solução obtida pelo programa heurístico com a solução obtida por um "package" de programação matemática. A função de custo unitário $\overline{g}_4(v)$ usada no programa heurístico foi obtida a partir de uma função de custo total linear por segmento. As opções da corrida foram as mesmas utilizadas no problema 1, e a evolução do processamento pode ser acompanhada na tabela 3.7.4. A rotina "EXCLUIR" não melhorou a solução de equilíbrio inicial (configuração 1) que foi aceita como solução final.

O mesmo problema foi resolvido pelo "LARGE SYSTEMS TEMPO", da Borroughs, que pode resolver problemas de programação inteira-mixta pela técnica de "branch-and-bound". Utilizou-se para isso o modelo MM4, que para este problema é composto de 146 restrições e 308 variáveis,das quais 39 são inteiras do tipo 0-1. A solução ótima obtida pelo "TEMPO" teve um custo de Cr$ 133.117.000,00 contra Cr$ 133.133.352,54 do programa heurístico. A diferença de Cr$ 5.352,74 corresponde a um erro relativo de 0.0037%. Este pequeno erro torna-se irrelevante quando comparamos os tempos de computação de cada solução: 1848,118 segundos gastos pelo TEMPO contra 150,446 segundos gastos pelo programa heurístico.

Como se vê, em muitos casos será preferível obter uma solução um pouco pior em troca de um tempo de processamento 12 vezes menor. Isto será verdade principalmente se lembrarmos que problemas reais terão freqüentemente dimensões bem maiores que as deste.

Outra observação interessante é que a solução final do programa heurístico foi a primeira solução obtida pelo programa heurístico. Isso mostra que em alguns casos a rotina EQUILIBRAR é, por si só, um bom instrumento de cálculo.

CAPACIDADES EM TONELADAS DE CADA ARMAZEM NA CONFIGURAÇÃO EXAMINADA											
LOCAL	CONF.1	CONF.2	CONF.3	CONF.4	CONF.5	CONF.6	CONF.7	CONF.8	CONF. 9	CONF.10	CONF.11
BOM JARDIM	19801.8	19801.8	38163.6	19801.8	-	19968.2	19801.8	19968.2	19801.8	19801.8	19801.8
LGO. PEDRA	36341.8	36175.5	43292.7	35047.3	36341.8	36175.5	36341.8	36341.8	-	43292.7	43292.7
MONÇÃO	35871.8	35871.8	35871.8	35871.8	55673.6	35871.8	35871.8	-	35871.8	35871.8	35871.8
PINDARÉ	-	-	-	-	-	-	-	-	-	-	-
STA. INÊS	18361.8	18361.8	-	26436.4	18361.8	18361.8	18361.8	54233.6	53409.1	18361.8	18361.8
STA. LUZIA	46480.0	46480.0	46480.0	46480.0	46480.0	46480.0	46480.0	46480.0	46480.0	46480.0	-
VIT. FREIRE	28009.9	28009.9	28757.3	37002.7	28000.9	-	28000.9	28000.9	28000.9	28757.3	39362.7
BACABAL	46444.9	48518.7	46444.9	50148.6	46444.9	46444.9	46444.9	46444.9	46444.9	-	46444.9
IPIXUNA	-	-	-	-	-	-	-	-	-	-	-
OLHO D'ÁGUA	19928.0	20009.1	12004.7	-	19928.0	48714.0	19928.0	19928.0	20099.1	14758.9	48885.0
PEDREIRAS	7029.9	-	7029.9	7029.9	7029.9	7029.9	40302.8	7029.9	7029.9	50720.6	7029.9
P. DE PEDRA	-	-	-	-	-	-	-	-	-	-	-
ST. A. LOPES	33272.9	38299.0	33272.9	33272.9	33272.9	33272.9	-	33272.9	33272.9	33272.9	33272.9
CUSTOS COM TRANSPORTES EM CR$1000,00	123848.3	124549.4	125116.1	124638.4	126096.2	125202.9	126757.8	124598.0	129210.0	127328.4	130066.9
CUSTOS COM ARMAZENS EM CR$.1000,00	25562.1	25357.1	25357.1	25338.7	25374.6	25438.4	25374.6	25374.6	25375.0	25357.1	25438.8
CUSTO TOTAL EM CR$1000,00	* 149410.4	149924.4	150473.2	149977.1	151470.8	150641.3	152132.4	149972.6	154585.0	152685.5	155505.7

TABELA 3.7.1

CAPACIDADES EM TONELADAS DE CADA ARMAZEM NA CONFIGURAÇÃO EXAMINADA											
LOCAL	CONF.1	CONF.2	CONF.3	CONF.4	CONF.5	CONF.6	CONF.7	CONF.8	CONF. 9	CONF.10	CONF.11
BOM JARDIM	19801.8	19801.8	19801.8	—	19801.8	19801.8	19801.8	40528.2	32786.4	30965.5	
LGO. PEDRA	36175.5	35047.3	43292.7	36175.5	—	50222.7	35047.3	43292.7	43292.7	43292.7	
MONÇÃO	35871.8	35871.8	44416.4	55673.6	35871.8	38550.9	—	—	—	—	
PINDARÉ	—	—	—	—	—	—	—	—	—	—	
STA. INÊS	18361.8	19491.0	—	18361.8	60000.0	18361.8	60000.0	60000.0	60000.0	60000.0	
STA. LUZIA	46480.0	46480.0	57053.6	46480.0	47963.6	46480.0	46480.0	47263.4	—	46480.0	
VIT. FREIRE	35874.5	43949.1	28000.9	35874.5	37002.7	56843.6	39310.9	—	60000.0	60000.0	
BACABAL	51607.5	55311.2	51607.5	51607.5	55311.2	60000.0	55311.2	60000.0	60000.0	—	
IPIXUNA	—	—	—	—	—	—	—	—	—	—	
OLHO D'ÁGUA	12004.7	—	12004.7	12004.7	—	—	—	—	—	—	
PEDREIRAS	—	—	—	—	—	—	—	—	—	—	
P. DE PEDRA	—	—	—	—	—	—	—	—	—	—	
ST. A. LOPES	35140.2	35140.2	35140.2	35140.2	35140.2	—	35140.2	40302.8	35140.2	49299.0	
CUSTOS COM TRANSPORTES EM CR$1000,00	124584.6	125524.2	126057.0	126832.5	130736.7	134386.0	126171.6	128368.2	132247.8	135070.2	
CUSTOS COM ARMAZENS EM CR$.1000,00	20509.5	19369.6	19192.9	19203.8	18045.3	17885.6	18048.9	16768.9	16698.4	16697.9	
CUSTO TOTAL EM CR$1000,00	145094.1	144893.8	145249.9	146036.3	148782.0	152268.6	* 144220.5	145137.1	148946.2	151768.1	

TABELA 3.7.2

CAPACIDADES EM TONELADAS DE CADA ARMAZEM NA CONFIGURAÇÃO EXAMINADA											
LOCAL	CONF.1	CONF.2	CONF.3	CONF.4	CONF.5	CONF.6	CONF.7	CONF.8	CONF. 9	CONF.10	CONF.11
BOM JARDIM	19801.8	-	19801.8	19801.8	19801.8	22110.0	33153.6	36590.9	19801.8		
LGO. PEDRA	35047.3	35047.3	43292.7	50222.7	-	35047.3	43292.7	43292.7	43292.7		
MONÇÃO	35871.8	55673.3	44416.4	38550.9	35871.8	-	43292.7	43292.7	43292.7		
PINDARÉ	-	-	-	-	-	-	-	-	-		
STA. INÊS	26436.4	26436.4	-	26436.4	60000.0	60000.0	33520.0	60000.0	53910.0		
STA. LUZIA	46480.0	46480.0	57053.6	46480.0	47963.6	46480.0	47236.4	-	46480.0		
VIT. FREIRE	40605.5	40605.5	39678.2	48769.1	40605.5	40605.5	-	60000.0	38702.7		
BACABAL	51607.5	51607.5	51607.5	60000.0	51607.5	51607.5	60000.0	56088.8	-		
IPIXUNA	-	-	-	-	-	-	-	-	-		
OLHO D'ÁGUA	-	-	-	-	-	-	-	-	-		
PEDREIRAS	-	-	-	-	-	-	-	-	-		
P. DE PEDRA	-	-	-	-	-	-	-	-	-		
ST. A. LOPES	35140.2	35140.2	35140.2	-	35140.2	35140.2	38229.0	35140.2	49229.0		
CUSTOS COM TRANSPORTES EM CR$1000,00	125348.3	127596.8	127034.2	134181.4	130734.4	126303.8	127361.9	132358.6	133300.2		
CUSTOS COM ARMAZENS EM CR$.1000,00	13793.5	12959.9	12925.4	12005.9	12912.2	12927.8	13116.6	12113.5	11925.2		
CUSTO TOTAL EM CR$1000,00	* 139141.9	140556.7	139959.6	146187.2	143646.6	139231.6	140478.5	144472.1	145225.5		

TABELA 3.7.3

CAPACIDADES EM TONELADAS DE CADA ARMAZEM NA CONFIGURAÇÃO EXAMINADA											
LOCAL	CONF.1	CONF.2	CONF.3	CONF.4	CONF.5	CONF.6	CONF.7	CONF.8	CONF. 9	CONF.10	CONF.11
BOM JARDIM	19801.8	19801.8	19801.8	19801.8	-	19801.8	19801.8	33153.6	19801.8	19801.8	19801.8
LGO. PEDRA	36175.5	35175.5	35047.3	43292.7	36175.5	36341.8	35043.3	43292.7	-	43292.7	43292.7
MONÇÃO	35871.8	35175.5	35871.8	44416.4	55673.6	35871.8	-	35871.8	35871.8	35871.8	35871.8
PINDARÉ	-	-	-	-	-	-	-	-	-	-	-
STA. INÊS	18361.8	18361.8	26436.4	-	18361.8	18361.8	54233.6	18361.8	60000.0	18361.8	18361.8
STA. LUZIA	46480.0	46480.0	46480.0	57053.6	46480.0	46480.0	46480.0	47236.4	47963.6	-	46480.0
VIT. FREIRE	35874.5	35874.5	37000.2	28000.9	35874.5	35708.2	37002.7	-	37002.7	60000.0	28757.3
BACABAL	46615.9	51607.5	50148.6	46444.9	46615.9	46444.9	51607.5	60000.0	50148.6	46615.9	-
IPIXUNA	-	-	-	-	-	-	-	-	-	-	-
OLHO D'ÁGUA	12004.7	12004.7	-	12004.7	12004.7	12004.7	12004.7	13509.3	-	27669.2	14758.9
PEDREIRAS	6858.9	-	7029.9	7029.9	6858.9	40302.8	-	7029.9	7029.9	6858.9	50720.6
P. DE PEDRA	-	-	-	-	-	-	-	-	-	-	-
ST. A. LOPES	33272.9	35140.2	33272.9	33272.9	-	35140.2	33272.9	33272.9	33272.9	33272.9	33272.9
CUSTOS COM TRANSPORTES EM CR$1000,00	123886.4	124584.6	124638.4	125344.6	126134.3	126783.5	125322.5	125621.3	130024.4	130195.0	127328.4
CUSTOS COM ARMAZENS EM CR$.1000,00	9236.0	8633.0	8616.0	8507.8	8503.9	8633.0	7910.5	8523.9	7839.7	8542.9	8643.2
CUSTO TOTAL EM CR$1000,00	* 133122.4	133217.6	133254.4	133852.4	134638.2	135416.5	133233.0	134415.2	137864.1	138737.9	135971.6

TABELA 3.7.4

REDE 2

Uma segunda rede foi construída para um problema com armazéns modulados (silos). Nesta rede, foram utilizados os mesmos locais candidatos e mesmas vias de escoamento da primeira rede, aumentando-se apenas o número de arcos em cada armazém, que corresponde ao número máximo de silos permitido. Este número máximo de silos foi estabelecido a partir dos resultados observados nos problemas anteriores, e são mostrados entre parêntesis (em seguida ao nome do local) na tabela 3.7.5

Com esta rede, foi então resolvido o problema 6, utilizando-se a técnica desenvolvida na seção 3.6, onde a seguinte função de custo foi utilizada:

$$\bar{g}_5(v) = \begin{cases} \dfrac{800.000}{V} & \text{, no primeiro arco de um armazém} \\[2ex] \dfrac{366.000}{V} & \text{, nos demais arcos do armazém} \end{cases}$$

onde V é a capacidade estática de armazenagem.

Como é fácil notar, considerou-se que a amortização anual do custo fixo em cada armazém é de Cr$ 434.000,00 e cada silo utilizado tem uma amortização anual de Cr$ 366.000,00, independente de sua plena utilização. Aos arcos representando silos foram dadas capacidades máximas de fluxo, de maneira que só sejam utilizados silos de 6000 t cada. As opções de corrida utilizadas foram:

(a) tamanho do buffer = 5

(b) armazéns modulados

(c) rotina EQUILIBRAR utilizada em todas as configurações

(d) tolerância = 0.005

A rotina EXCLUIR não melhorou a solução inicial do programa e a configuração 1 foi aceita como solução final. Esta solução é mostrada na tabela 3.7.5, onde a segunda coluna mostra o número de silos de 6000 t alocados a cada armazém.

LOCAL	Nº DE SILOS	AMORTIZAÇÃO ANUAL EM CR$1.000,00
BOM JARDIM(5)	4	1898
LGO PEDRA (8)	6	2630
MONÇÃO(10)	6	2630
PINDARE(6)	0	0
SANTA INES(10)	3	1532
SANTA LUZIA(10)	8	3362
VIT.FREIRE(10)	5	2264
BACABAL(10)	6	2630
TPIXUNA(4)	1	800
OLHO D'ÁGUA(4)	3	1532
PEDREIRAS(4)	1	800
P.DE PEDRA(4)	2	1166
ST.A.LOPES(8)	4	1898

CUSTO ANUAL COM ARMAZÉNS : Cr$ 23.142.000,00

CUSTO ANUAL COM TRANSPORTE : CR$ 124.125.443,79

CUSTO TOTAL ANUAL : CR$ 147.267.443,79

TABELA 3.7.5

Na solução obtida, todos os silos foram utilizados plenamente, exceto:

1 silo em Bom Jardim, com 1801.8 t de capacidade estática.

1 silo em Pedreiras, com 5452.7 t de capacidade estática

1 silo em P. de Pedra, com 5370.0 t de capacidade estática.

Pode parecer surpreendente o fato de não ter a exclusão de Pedreiras melhorado o custo da solução inicial. Um armazém operando com um único silo e mesmo assim ocioso, implicou num custo anual de armazenagem de Cr$ 146.72 por tonelada, bastante elevado em relação aos demais armazéns. Esta surpresa será acentuada se acrescentarmos que na configuração 2(com Pedreiras excluído), apenas 1 armazém funcionava ocioso, contra 3 armazéns ociosos na configuração 1(tabela 3.7.5)

No entanto, qualquer dúvida é esclarecida quando examinamos o resumo de custos na configuração 2:

CUSTO ANUAL COM ARMAZÉNS: CR$ 22.708.000,00

CUSTO ANUAL COM TRANSPORTES: CR$ 125.031.893,60

CUSTO TOTAL ANUAL: Cr$ 147.739.893,60

Como se vê, a apreciável redução nos custos com armazéns foi compensada com importante aumento nos custos de transporte,tornando a configuração 2 menos atrativa.

Isso mostra que o problema de localização de silos requer um estudo analítico preciso, e que custos de transporte (ao menos aqueles pagos pelo produtor) devem ser levados em conta.

REDE 3

Finalmente, foi construída uma terceira rede,na qual se procurou estabelecer as mesmas condições utilizadas no estudo da CIBRAZEM. Para isso, foi necessário ampliar a segunda rede, refletindo as seguintes considerações:

1 - Ao conjunto de locais candidatos, acrescentou-se Altamira(4), Bacabinha(4), Lago do Junco(2), Pio XII (2), São Mateus(2). Os números entre parênteses indicam o número máximo de silos permitido em cada local. Nos outros locais, este número foi mantido como no problema 6.

2 - Nas duas micro-regiões estudadas, já existe uma oferta de armazenagem disponível, não pertence à CIBRAZEM. Esta oferta está assim distribuída:

Bom Jardim - 1545 t	Bacabal - 4261 t
Lago da Pedra - 3000 t	Olho D'água - 1670 t
Pindaré-Mirim - 24570 t	Poção de Pedras - 29248 t
Sta Ines - 15076 t	St.A.Lopes - 6258 t
Vitorino Freire - 6900 t	

Para que se levasse esta oferta já existente em consideração, em cada um dos locais acima foi adicionado um arco de custo nulo, e com capacidade de fluxo correspondente à capacidade de armazenamento já existente no local.

O esquema de produção e demanda foi mantido como nas duas primeiras redes. Com isso, a terceira rede ficou constituída de 66 nós e 414 arcos.

A função de custo unitário foi obtida a partir da estrutura de custos utilizada pela CIBRAZEM, onde arbitramos a vida útil de cada silo em 8 anos:

$$\bar{g}_6(v) = \frac{666000}{V} \quad , \quad 0 \leq V \leq 6000$$

Esta função foi utilizada em todos os arcos (silos), refletindo a inexistencia de custos fixos, como no estudo da CIBRAZEM. Cada silo tem uma capacidade máxima de 6000 t.

Para tornar as condições do problema mais próximas às da CIBRAZEM, não consideramos os custos de transporte entre armazéns e centros de demanda, restringindo-os aos custos de transporte que afetam diretamente ao produtor, ou seja, entre os centros de produção e os armazéns.

A solução encontrada para este problema correspondeu à solução inicial de equilíbrio, na configuração 1. Esta solução é mostrada na tabela 3.7.6, ao lado da solução proposta pela CIBRAZEM. O dimensionamento da rede está expresso em termos do número de silos de 6000 t alocados a cada cidade.

NOME DO LOCAL	Nº DE SILOS DE 6000 t ALOCADOS	
	CIBRAZEM	PROGRAMA HEURÍSTICO
BOM JARDIM	3	3
LAGO DA PEDRA	5	5
MONÇÃO	5	3
PINDARÉ MIRIM	0	0
SANTA INES	0	0
SANTA LUZIA	8	8
VITORINO FREIRE	1	1
BACABAL	3	1
IPIXUNA	2	3
OLHO D'ÁGUA	1	2
PEDREIRAS	0	1
POÇÃO DE PEDRA	0	0
ST.ANTONIO DOS LOPES	0	1
ALTAMIRA	2	0
BACABINHA	2	4
LAGO DO JUNCO	1	1
PIO XII	1	0
S.MATEUS DO MARANHÃO	2	1
Nº TOTAL DE SILOS	36	34
INVESTIMENTO EM ARMAZÉNS(*)	Cr$ 191.808.000,00	Cr$ 181.152.000,00

TABELA 3.7.6

(*) valor presente do investimento

A diferença entre as soluções deve ser devida a:

1 - Omissão, em nosso modelo, de consideração levadas em conta no estudo da CIBRAZEM que não foram claramente explicitadas no relatório |4|

2 - Desconsideração de custos de transporte no estudo da CIBRAZEM

3 - Eventual imprecisão nos custos de transporte utilizados pelo programa heurístico

4 - O estudo da CIBRAZEM abrange outras micro-regiões. Alguns armazéns das micro-regiões 34 e 35 podem ter sido dimensionados para atender pequena parte da produção de municípios fora dessas duas micro-regiões, o que explica a maior capacidade da rede de armazéns sugerida pela CIBRAZEM.

Como medida de performance do programa, mostramos na tabela 3.3.7 os tempos de execução (em minutos) e o número de vezes que a rotina "out-of-kilter" for chamada, em cada um dos 7 problemas aqui descritos.

PROBLEMA Nº	TEMPO DE EXEC(MIN)	Nº DE CHAMADAS DA out-of-kilter
1	2.15	34
2	2.37	29
3	2.45	33
4	2.33	25
5	2.50	46
6	4.76	34
7	6.35	47

TABELA 3.3.7

Estes tempos de execução foram obtidos no computador BORROUGHS 6700 instalado no NCE/UFRJ.

Complementando informações sobre o programa, o anexo 1 descreve o input para o mesmo, e as opções para cada corrida.

No anexo 2 é mostrado o OUTPUT gerado pelo programa para o problema 3.

No anexo 3, é mostrada a listagem do programa fonte, codificado em ALGOL. A rotina BKILTE, que executa o algoritmo "out-of-kilter" é externa ao programa. Maiores informações sobre esta rotina podem ser obtidas na referência |17|, e uma listagem da mesma se encontra no anexo 4.

4. CONCLUSÃO

Programas heurísticos tem sido recomendados para a análise de problemas de localização de grande porte. A rapidez proporcionada por esse programas tem permitido a abordagem de problemas cuja dimensão tornaria praticamente inviável o uso de outras técnicas.

Uma outra importante vantagem dos programas heurísticos é a facilidade que se tem para tratar de funções de custo complexas. Por exemplo, a simples consideração de custos marginais de armazenagem não constantes causa uma multiplicação do número de variáveis inteiras, quando técnicas de programação inteira-mixta são utilizadas (ver modelo MM4). Isto em geral representa um ônus considerável em termos de tempo computacional, enquanto que esse ônus é irrelevante em programas heurísticos. Esta facilidade tem sua importância fundamentada no fato de ser a solução do problema muitas vezes bastante sensível em relação à forma da curva de custo, como reportam Feldman, Leher e Ray |7|.

Neste trabalho procuramos desenvolver um programa onde a avaliação dos custos de armazenagem em cada configuração fosse feita com precisão. O uso do método iterativo para a determinação de custos de equilíbrio em todas as configurações avaliadas permitiu melhor confiabilidade na confrontação de alternativas.

Além disso, procuramos desenvolver um modelo adaptado às condições dos países em desenvolvimento. O uso de redes capacitadas permitiu grande flexibilidade na consideração de uma variedade de condições encontradas principalmente em países menos desenvolvidos, tais como estrangulamentos nas vias de transporte, impossibilidade de escoamento em certas épocas do ano, etc. A adoção de redes capacitadas possibilitou também uma adaptação do programa para tratar de casos de armazéns modulares, o que reforça a idéia de flexibilidade proporcionada. O uso de um algoritmo "out-of-kilter" nas otimizações intermediárias nos pareceu bastante apropriado, tendo em vista a estrutura das redes usadas como modelos.

Embora não se possa garantir a obtenção de uma solução ótima para todos os modelos, o método apresentado permite que se obtenha soluções quase ótimas (em ótimas) para problemas cujos modelos matemáticos foram elaborados com maior precisão, como no caso de custo não lineares.

É evidente que nossos testes não cobriram todas as condições que podem surgir em problemas reais. É certo que em aplicações concretas diversas particularidades não mencionadas nesse trabalho deverão aparecer. Contudo, considerando a flexibilidade proporcionada pelo programa, confiamos que a maioria das eventuais dificuldades poderão ser contornadas.

De qualquer modo, acreditamos que o método aqui apresentado oferece boas alternativas para a solução de problemas complexos de localização de facilidades numa rede de distribuição.

ANEXO I

DESCRIÇÃO DO INPUT DO PROGRAMA

A entrada de dados para o programa se dá através de 4 tipos de cartão:

1 - CARTÃO DE CONTROLE, que contem as dimensões do problema e as opções da corrida, com o seguinte formato (campos limitados pela coluna inicial e final).

1-5 : número de arcos da rede

6-10: número de nós da rede

11-15: número de locais candidatos

16-20: número de arcos correspondendo a locais candidatos. Este número pode ser maior que o número de locais candidatos no uso de armazéns modulados (ex: vários silos por armazém).

OPÇÕES DA CORRIDA

21-25: tamanho do buffer

26-30: número de funções de custo diferentes

32: indica se armazéns são modulados (1) ou não (0)

34: indica se a rotina EQUILIBRAR será usada em todas as configurações (0) ou não (1). Na verdade aqui se trata de saber se a rotina EQUILIBRAR será ou não usada para obter uma nova estrutura de custos compatível cada vez com um armazém é excluído. Caso negativo, será utilizado o procedimento sugerido por Feldman, Leher e Ray, descrito na seção 3.2.

35-49: Tolerância para a rotina EQUILIBRAR.

2 - CARTÕES DESCRITORES DA REDE - Cada cartão desse tipo descreve um arco em detalhe, no seguinte formato:

1-5 : reservado para identificação numérica
6-10: nó de origem do arco
11-15: nó terminal do arco
16-25: limite inferior de fluxo
26-35: limite superior de fluxo
36-45: custo ou penalidade do arco

3 - CARTÕES DESCRITORES DE ARMAZÉNS - dão informações gerais sobre um local candidato a sede de armazém, com o seguinte formato:

1-10: nome (alfanumérico) do local
11-15: índice de rotação no local
16-20: número do primeiro arco correspondente a este local
21-25: número de arcos correspondentes a este local
41: indica se o local deve ser considerado para "DROP" (0) ou não (1)

4 - CARTÕES SILOS (ARMAZÉNS) - dão informações a respeito de arcos representando armazéns ou silos, com o seguinte formato:

1-5 : número do arco
6-10: número do armazém a que pertence
11-15: número que identifica o tipo de função de custo $\bar{g}(v)$ associado ao arco

ORDEM DE ENTRADA

1 - Inicialmente, é lido o cartão de controle

2 - A seguir, são lidos todos os cartões descritores da rede

3 - Os cartões tipo 3 e 4 são lidos alternadamente : após cada cartão tipo 3 segue um conjunto de cartões tipo 4. Se o armazém não é modulado, apenas um cartão tipo 4 seguirá cada cartão tipo 3.

ANEXO II

OUTPUT DO PROGRAMA

CONFIGURACAO N. 1 FASE 1 TOL: 0,001

LOCAL	ARCO	CUSTO UNIT. USADO	VOLUME DINAMICO	CAPACIDADE ESTATICA	CUSTO UNIT. REAL	CUSTO TOTAL OBTIDO	CUSTO TOTAL REAL	ERRO RELATIVO
BOM JARDIM	25- 38	57.53	21782.0	19801.8	57.53	1139245.70	1139245.70	0.0000
LGO PEDRA	26- 39	40.47	39793.0	36175.5	40.47	1464140.64	1464140.64	0.0000
MONCAO	27- 40	40.67	39459.0	35871.8	40.67	1458888.36	1458888.36	0.0000
PINDARE MI	28- 41	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
STA INES	29- 42	60.18	20198.0	18361.8	60.18	1105014.87	1105014.87	0.0000
STA LUZIA	30- 43	35.09	51128.0	46480.0	35.09	1630882.33	1630882.33	0.0000
VIT FREIRE	31- 44	40.67	39462.0	35874.5	40.67	1458935.63	1458935.63	0.0000
BACABAL	32- 45	33.08	55220.0	51607.5	33.08	1706923.33	1706923.33	0.0000
IPIXUNA	33- 46	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
OLHO DAGUA	34- 47	77.86	12845.0	12004.7	77.86	934686.68	934686.68	0.0000
PEDREIRAS	35- 48	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
P.DE PEDRA	36- 49	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
ST A LOPES	37- 50	41.15	37600.0	35140.2	41.15	1446140.58	1446140.58	0.0000

CUSTOS EM TRANSPORTES POR PERIODO 124584622,06

AMORTIZACAO OU INVESTIMENTO EM ARMAZENAGEM: 12344858,13

CUSTO TOTAL POR PERIODO NA SOL. ENCONTRADA: 136929480,19

DIAGNOSTICO: SOLUCAO OTIMA PARA O PROBLEMA DE DISTRIBUICAO COMPATIVEL COM CURVA DE CUSTOS * SOLUCAO OTIMA LOCAL

CONFIGURACAO N. 2 FASE 3 TOL: 0,001

LOCAL	ARCO	CUSTO UNIT. USADO	VOLUME DINAMICO	CAPACIDADE ESTATICA	CUSTO UNIT. REAL	CUSTO TOTAL OBTIDO	CUSTO TOTAL REAL	ERRO RELATIVO
BOM JARDIM	25- 38	57.53	21782.0	19801.8	57.53	1139245.70	1139245.70	0.0000
LGO PEDRA	26- 39	41.22	38552.0	35047.3	41.22	1444512.20	1444512.20	0.0000
MONCAO	27- 40	40.67	39459.0	35871.8	40.67	1458888.36	1458888.36	0.0000
PINDARE MI	28- 41	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
STA INES	29- 42	48.53	29080.0	26436.4	48.53	1282873.86	1282873.86	0.0000
STA LUZIA	30- 43	35.09	51128.0	46480.0	35.09	1630882.33	1630882.33	0.0000
VIT FREIRE	31- 44	39.95	40703.0	37002.7	39.95	1478339.95	1478339.95	0.0000
BACABAL	32- 45	31.81	59183.0	55311.2	31.81	1759522.77	1759522.77	0.0000
IPIXUNA	33- 46	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
OLHO DAGUA	34- 47	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
PEDREIRAS	35- 48	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
P.DE PEDRA	36- 49	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
ST A LOPES	37- 50	41.15	37600.0	35140.2	41.15	1446140.58	1446140.58	0.0000

CUSTOS EM TRANSPORTES POR PERIODO 125350757.44

AMORTIZACAO DO INVESTIMENTO EM ARMAZENAGEM: 11640405.75

CUSTO TOTAL POR PERIODO NA SOL. ENCONTRADA: 136991163.19

DIAGNOSTICO: SOLUCAO OTIMA PARA O PROBLEMA DE DISTRIBUICAO COMPATIVEL COM CURVA DE CUSTOS * SOLUCAO OTIMA LOCAL

CONFIGURACAO N. 3 FASE 3 TOL: 0.001

LOCAL	ARCO	CUSTO UNIT. USADO	VOLUME DINAMICO	CAPACIDADE ESTATICA	CUSTO UNIT. REAL	CUSTO TOTAL OBTIDO	CUSTO TOTAL REAL	ERRO RELATIVO
BOM JARDIM	25- 38	57.53	21782.0	19801.8	57.53	1139245.70	1139245.70	0.0000
LGO PEDRA	26- 39	36.53	47622.0	43292.7	36.53	1581483.36	1581483.36	0.0000
MONCAO	27- 40	36.00	48858.0	44416.4	36.00	1599099.67	1599099.67	0.0000
PINDARE MI	28- 41	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
STA INES	29- 42	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
STA LUZIA	30- 43	31.26	62759.0	57053.6	31.26	1783658.11	1783658.11	0.0000
VIT FREIRE	31- 44	46.93	30801.0	28000.9	46.93	1314011.62	1314011.62	0.0000
BACABAL	32- 45	33.08	55220.0	51607.5	33.08	1706923.33	1706923.33	0.0000
IPIXUNA	33- 46	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
OLHO DAGUA	34- 47	77.86	12845.0	12004.7	77.86	934686.68	934686.68	0.0000
PEDREIRAS	35- 48	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
P.DE PEDRA	36- 49	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
ST A LOPES	37- 50	41.15	37600.0	35140.2	41.15	1446140.58	1446140.58	0.0000

CUSTOS EM TRANSPORTES POR PERIODO 126056992.88

AMORTIZACAO DO INVESTIMENTO EM ARMAZENAGEM: 11505249.05

CUSTO TOTAL POR PERIODO NA SOL. ENCONTRADA: 137562241.93

DIAGNOSTICO: SOLUCAO OTIMA PARA O PROBLEMA DE DISTRIBUICAO COMPATIVEL COM CURVA DE CUSTOS * SOLUCAO OTIMA LOCAL

CONFIGURACAO N. 4 FASE 3 TOL: 0,001

LOCAL	ARCO	CUSTO UNIT. USADO	VOLUME DINAMICO	CAPACIDADE ESTATICA	CUSTO UNIT. REAL	CUSTO TOTAL OBTIDO	CUSTO TOTAL REAL	ERRO RELATIVO
BOM JARDIM	25- 38	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
LGO PEDRA	26- 39	40.47	39793.0	36175.5	40.47	1464140.64	1464140.64	0.0000
MONCAO	27- 40	31.69	61241.0	55673.6	31.69	1764573.86	1764573.86	0.0000
PINDARE MI	28- 41	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
STA INES	29- 42	60.18	20198.0	18361.8	60.18	1105014.87	1105014.87	0.0000
STA LUZIA	30- 43	35.09	51128.0	46480.0	35.09	1630882.33	1630882.33	0.0000
VIT FREIRE	31- 44	40.67	39462.0	35874.5	40.67	1458935.63	1458935.63	0.0000
BACABAL	32- 45	33.08	55220.0	51607.5	33.08	1706923.33	1706923.33	0.0000
IPIXUNA	33- 46	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
OLHO DAGUA	34- 47	77.86	12845.0	12004.7	77.86	934686.68	934686.68	0.0000
PEDREIRAS	35- 48	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
P.DE PEDRA	36- 49	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
ST A LOPES	37- 50	41.15	37600.0	35140.2	41.15	1446140.58	1446140.58	0.0000

CUSTOS EM TRANSPORTES POR PERIODO 126832524.46

AMORTIZACAO DO INVESTIMENTO EM ARMAZENAGEM: 11511297.93

CUSTO TOTAL POR PERIODO NA SOL. ENCONTRADA: 138343822.40

DIAGNOSTICO: SOLUCAO OTIMA PARA O PROBLEMA DE DISTRIBUICAO COMPATIVEL COM CURVA DE CUSTOS * SOLUCAO OTIMA LOCAL

CONFIGURACAO N. 5 FASE 3 TOL: 0.001

LOCAL	ARCO	CUSTO UNIT. USADO	VOLUME DINAMICO	CAPACIDADE ESTATICA	CUSTO UNIT. REAL	CUSTO TOTAL OBTIDO	CUSTO TOTAL REAL	ERRO RELATIVO
BOM JARDIM	25- 38	57.53	21782.0	19801.8	57.53	1139245.70	1139245.70	0.0000
LGO PEDRA	26- 39	34.76	51940.0	47218.2	34.76	1642079.38	1642079.38	0.0000
MONCAO	27- 40	40.67	39459.0	35871.8	40.67	1458888.36	1458888.36	0.0000
PINDARE MI	28- 41	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
STA INES	29- 42	60.18	20198.0	18361.8	60.18	1105014.87	1105014.87	0.0000
STA LUZIA	30- 43	35.09	51128.0	46480.0	35.09	1630882.33	1630882.33	0.0000
VIT FREIRE	31- 44	39.95	40703.0	37002.7	39.95	1478339.95	1478339.95	0.0000
BACABAL	32- 45	30.39	64200.0	60000.0	30.39	1823646.20	1823646.20	0.0000
IPIXUNA	33- 46	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
OLHO DAGUA	34- 47	48.74	28077.0	26240.2	48.74	1278905.22	1278905.22	0.0000
PEDREIRAS	35- 48	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
P.DE PEDRA	36- 49	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
ST A LOPES	37- 50	724.15	0.0	0.0	724.15	0.00	0.00	0.0000

CUSTOS EM TRANSPORTES POR PERIODO 132187288.76

AMORTIZACAO DO INVESTIMENTO EM ARMAZENAGEM: 11557002.02

CUSTO TOTAL POR PERIODO NA SOL. ENCONTRADA: 143744290.78

DIAGNOSTICO: SOLUCAO OTIMA PARA O PROBLEMA DE DISTRIBUICAO COMPATIVEL COM CURVA DE CUSTOS * SOLUCAO OTIMA LOCAL

CONFIGURACAO N. 6 FASE 3 TOL: 0.001

LOCAL	ARCO	CUSTO UNIT. USADO	VOLUME DINAMICO	CAPACIDADE ESTATICA	CUSTO UNIT. REAL	CUSTO TOTAL OBTIDO	CUSTO TOTAL REAL	ERRO RELATIVO
BOM JARDIM	25- 38	31.69	61241.0	55673.6	31.69	1764573.86	1764573.86	0.0000
LGO PEDRA	26- 39	40.47	39793.0	36175.5	40.47	1464140.64	1464140.64	0.0000
MONCAO	27- 40	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
PINDARE MI	28- 41	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
STA INES	29- 42	60.18	20198.0	18361.8	60.18	1105014.87	1105014.87	0.0000
STA LUZIA	30- 43	35.09	51128.0	46480.0	35.09	1630882.33	1630882.33	0.0000
VIT FREIRE	31- 44	40.67	39462.0	35874.5	40.67	1458935.63	1458935.63	0.0000
BACABAL	32- 45	33.08	55220.0	51607.5	33.08	1706923.33	1706923.33	0.0000
IPIXUNA	33- 46	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
OLHO DAGUA	34- 47	77.86	12845.0	12004.7	77.86	934686.68	934686.68	0.0000
PEDREIRAS	35- 48	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
P.DE PEDRA	36- 49	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
ST A LOPES	37- 50	41.15	37600.0	35140.2	41.15	1446140.58	1446140.58	0.0000

CUSTOS EM TRANSPORTES POR PERIODO 125452720.06

AMORTIZACAO DO INVESTIMENTO EM ARMAZENAGEM: 11511297.93

CUSTO TOTAL POR PERIODO NA SOL. ENCONTRADA: 136964018.00

DIAGNOSTICO: SOLUCAO OTIMA PARA O PROBLEMA DE DISTRIBUICAO COMPATIVEL COM CURVA DE CUSTOS * SOLUCAO OTIMA LOCAL

CONFIGURACAO N. 7 FASE 3 TOL: 0.001

LOCAL	ARCO	CUSTO UNIT. USADO	VOLUME DINAMICO	CAPACIDADE ESTATICA	CUSTO UNIT. REAL	CUSTO TOTAL OBTIDO	CUSTO TOTAL REAL	ERRO RELATIVO
BOM JARDIM	25- 38	42.55	36469.0	33153.6	42.55	1410842.00	1410842.00	0.0000
LGO PEDRA	26- 39	36.53	47622.0	43292.7	36.53	1581483.36	1581483.36	0.0000
MONCAO	27- 40	40.67	39459.0	35871.8	40.67	1458888.36	1458888.36	0.0000
PINDARE MI	28- 41	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
STA INES	29- 42	60.18	20198.0	18361.8	60.18	1105014.87	1105014.87	0.0000
STA LUZIA	30- 43	35.09	51128.0	46480.0	35.09	1630882.33	1630882.33	0.0000
VIT FREIRE	31- 44	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
BACABAL	32- 45	30.39	64200.0	60000.0	30.39	1823646.20	1823646.20	0.0000
IPIXUNA	33- 46	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
OLHO DAGUA	34- 47	64.49	17506.0	16360.7	64.49	1055133.38	1055133.38	0.0000
PEDREIRAS	35- 48	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
P.DE PEDRA	36- 49	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
ST A LOPES	37- 50	39.21	40905.0	38229.0	39.21	1499098.34	1499098.34	0.0000

CUSTOS EM TRANSPORTES POR PERIODO 126260970.58

AMORTIZACAO DO INVESTIMENTO EM ARMAZENAGEM: 11564988.84

CUSTO TOTAL POR PERIODO NA SOL. ENCONTRADA: 137825959.42

DIAGNOSTICO: SOLUCAO OTIMA PARA O PROBLEMA DE DISTRIBUICAO COMPATIVEL COM CURVA DE CUSTOS * SOLUCAO OTIMA LOCAL

CONFIGURACAO N. 8 FASE 3 TOL: 0.001

LOCAL	ARCO	CUSTO UNIT. USADO	VOLUME DINAMICO	CAPACIDADE ESTATICA	CUSTO UNIT. REAL	CUSTO TOTAL OBTIDO	CUSTO TOTAL REAL	ERRO RELATIVO
BOM JARDIM	25- 38	57.53	21782.0	19801.8	57.53	1139245.70	1139245.70	0.0000
LGO PEDRA	26- 39	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
MONCAO	27- 40	40.67	39459.0	35871.8	40.67	1458888.36	1458888.36	0.0000
PINDARE MI	28- 41	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
STA INES	29- 42	30.39	66000.0	60000.0	30.39	1823646.20	1823646.20	0.0000
STA LUZIA	30- 43	34.47	52760.0	47963.6	34.47	1653298.27	1653298.27	0.0000
VIT FREIRE	31- 44	39.95	40703.0	37002.7	39.95	1478339.95	1478339.95	0.0000
BACABAL	32- 45	31.81	59183.0	55311.2	31.81	1759522.77	1759522.77	0.0000
IPIXUNA	33- 46	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
OLHO DAGUA	34- 47	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
PEDREIRAS	35- 48	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
P.DE PEDRA	36- 49	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
ST A LOPES	37- 50	41.15	37600.0	35140.2	41.15	1446140.58	1446140.58	0.0000

CUSTOS EM TRANSPORTES POR PERIODO 130736744.84

AMORTIZACAO DO INVESTIMENTO EM ARMAZENAGEM: 10759081.83

CUSTO TOTAL POR PERIODO NA SOL. ENCONTRADA: 141495826.67

DIAGNOSTICO: SOLUCAO OTIMA PARA O PROBLEMA DE DISTRIBUICAO COMPATIVEL COM CURVA DE CUSTOS * SOLUCAO OTIMA LOCAL

CONFIGURACAO N. 9 FASE 3 TOL: 0.001

LOCAL	ARCO	CUSTO UNIT. USADO	VOLUME DINAMICO	CAPACIDADE ESTATICA	CUSTO UNIT. REAL	CUSTO TOTAL OBTIDO	CUSTO TOTAL REAL	ERRO RELATIVO
BOM JARDIM	25- 38	57.53	21782.0	19801.8	57.53	1139245.70	1139245.70	0.0000
LGO PEDRA	26- 39	36.53	47622.0	43292.7	36.53	1581483.36	1581483.36	0.0000
MONCAO	27- 40	40.67	39459.0	35871.8	40.67	1458888.36	1458888.36	0.0000
PINDARE MI	28- 41	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
STA INES	29- 42	60.18	20198.0	18361.8	60.18	1105014.87	1105014.87	0.0000
STA LUZIA	30- 43	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
VIT FREIRE	31- 44	30.39	66000.0	60000.0	30.39	1823646.20	1823646.20	0.0000
BACABAL	32- 45	33.08	55220.0	51607.5	33.08	1706923.33	1706923.33	0.0000
IPIXUNA	33- 46	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
OLHO DAGUA	34- 47	47.25	29606.0	27669.2	47.25	1307483.25	1307483.25	0.0000
PEDREIRAS	35- 48	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
P.DE PEDRA	36- 49	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
ST A LOPES	37- 50	41.15	37600.0	35140.2	41.15	1446140.58	1446140.58	0.0000

CUSTOS EM TRANSPORTES POR PERIODO 130885242.94

AMORTIZACAO DO INVESTIMENTO EM ARMAZENAGEM: 11568825.65

CUSTO TOTAL POR PERIODO NA SOL. ENCONTRADA: 142454068.59

DIAGNOSTICO: SOLUCAO OTIMA PARA O PROBLEMA DE DISTRIBUICAO COMPATIVEL COM CURVA DE CUSTOS * SOLUCAO OTIMA LOCAL

CONFIGURACAO N. 10 FASE 3 TOL: 0.001

LOCAL	ARCO	CUSTO UNIT. USADO	VOLUME DINAMICO	CAPACIDADE ESTATICA	CUSTO UNIT. REAL	CUSTO TOTAL OBTIDO	CUSTO TOTAL REAL	ERRO RELATIVO
BOM JARDIM	25- 38	57.53	21782.0	19801.8	57.53	1139245.70	1139245.70	0.0000
LGO PEDRA	26- 39	36.53	47622.0	43292.7	36.53	1581483.36	1581483.36	0.0000
MONCAO	27- 40	40.67	39459.0	35871.8	40.67	1458888.36	1458888.36	0.0000
PINDARE MI	28- 41	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
STA INES	29- 42	60.18	20198.0	18361.8	60.18	1105014.87	1105014.87	0.0000
STA LUZIA	30- 43	35.09	51128.0	46480.0	35.09	1630882.33	1630882.33	0.0000
VIT FREIRE	31- 44	46.93	30801.0	28000.9	46.93	1314011.62	1314011.62	0.0000
BACABAL	32- 45	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
IPIXUNA	33- 46	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
OLHO DAGUA	34- 47	33.56	53822.0	50300.9	33.56	1687920.00	1687920.00	0.0000
PEDREIRAS	35- 48	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
P.DE PEDRA	36- 49	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
ST A LOPES	37- 50	33.97	52675.0	49229.0	33.97	1672143.47	1672143.47	0.0000

CUSTOS EM TRANSPORTES POR PERIODO 130867243.30

AMORTIZACAO DO INVESTIMENTO EM ARMAZENAGEM: 11589589.71

CUSTO TOTAL POR PERIODO NA SOL. ENCONTRADA: 142456833.01

DIAGNOSTICO: SOLUCAO OTIMA PARA O PROBLEMA DE DISTRIBUICAO COMPATIVEL COM CURVA DE CUSTOS * SOLUCAO OTIMA LOCAL

CONFIGURACAO N. 11 FASE 4 TOL: 0.001

LOCAL	ARCO	CUSTO UNIT. USADO	VOLUME DINAMICO	CAPACIDADE ESTATICA	CUSTO UNIT. REAL	CUSTO TOTAL OBTIDO	CUSTO TOTAL REAL	ERRO RELATIVO
BOM JARDIM	25- 38	57.53	21782.0	19801.8	57.53	1139245.70	1139245.70	0.0000
LGO PEDRA	26- 39	40.47	39793.0	36175.5	40.47	1464140.64	1464140.64	0.0000
MONCAO	27- 40	40.67	39459.0	35871.8	40.67	1458888.36	1458888.36	0.0000
PINDARE MI	28- 41	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
STA INES	29- 42	60.18	20198.0	18361.8	60.18	1105014.87	1105014.87	0.0000
STA LUZIA	30- 43	35.09	51128.0	46480.0	35.09	1630882.33	1630882.33	0.0000
VIT FREIRE	31- 44	40.67	39462.0	35874.5	40.67	1458935.63	1458935.63	0.0000
BACABAL	32- 45	33.08	55220.0	51607.5	33.08	1706923.33	1706923.33	0.0000
IPIXUNA	33- 46	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
OLHO DAGUA	34- 47	77.86	12845.0	12004.7	77.86	934686.68	934686.68	0.0000
PEDREIRAS	35- 48	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
P.DE PEDRA	36- 49	724.15	0.0	0.0	724.15	0.00	0.00	0.0000
ST A LOPES	37- 50	41.15	37600.0	35140.2	41.15	1446140.58	1446140.58	0.0000

CUSTOS EM TRANSPORTES POR PERIODO 124584622.06

AMORTIZACAO DO INVESTIMENTO EM ARMAZENAGEM: 12344858.13

CUSTO TOTAL POR PERIODO NA SOL. ENCONTRADA: 136929480.19

DIAGNOSTICO: ESTA FOI A MELHOR SOLUCAO ENCONTRADA**

NUMERO DE OTIMIZACOES DA REDE DE FLUXO: 33

ESQUEMA DE FLUXO NA SOLUCAO ENCONTRADA

SEQ.	ARCO	FLUXO	CAPACIDADE	MINIMO	CUSTO	QBAR
1	1- 2	9902.0	9902.0	9902.0	0.00	2080.11
2	1- 3	16114.0	16114.0	16114.0	0.00	1989.03
3	1- 4	21782.0	21782.0	21782.0	0.00	2032.44
4	1- 5	38552.0	38552.0	38552.0	0.00	2021.06
5	1- 6	39459.0	39459.0	39459.0	0.00	1973.58
6	1- 7	10799.0	10799.0	10799.0	0.00	2013.06
7	1- 8	9399.0	9399.0	9399.0	0.00	1998.76
8	1- 9	51128.0	51128.0	51128.0	0.00	2029.67
9	1- 10	14687.0	14687.0	14687.0	0.00	1983.08
10	1- 11	10646.0	10646.0	10646.0	0.00	1938.98
11	1- 12	7377.0	7377.0	7377.0	0.00	2004.96
12	1- 13	3305.0	3305.0	3305.0	0.00	2058.98
13	1- 14	18463.0	18463.0	18463.0	0.00	1941.98
14	1- 15	10946.0	10946.0	10946.0	0.00	2004.96
15	1- 16	9551.0	9551.0	9551.0	0.00	1906.08
16	1- 17	2947.0	2947.0	2947.0	0.00	2015.42
17	1- 18	2219.0	2219.0	2219.0	0.00	2051.48
18	1- 19	3963.0	3963.0	3963.0	0.00	1976.27
19	1- 20	1998.0	1998.0	1998.0	0.00	2004.96
20	1- 21	8882.0	8882.0	8882.0	0.00	1999.57
21	1- 22	6011.0	6011.0	6011.0	0.00	1991.76
22	1- 23	11268.0	11268.0	11268.0	0.00	1913.96
23	1- 24	8089.0	8089.0	8089.0	0.00	2049.98
24	2- 25	0.0	********	0.0	222.75	85.08
25	2- 26	1241.0	********	0.0	113.05	0.00
[illegible]	[illegible]	[illegible]	********	[illegible]	[illegible]	[illegible]

26	2- 28	0.0	********	0.0	[illegible]	[illegible]
27	2- 29	0.0	********	0.0	198.00	62.66
28	2- 30	0.0	********	0.0	145.20	14.76
29	2- 31	8661.0	********	0.0	129.03	0.00
30	2- 32	0.0	********	0.0	198.00	11.88
31	2- 33	0.0	********	0.0	232.65	72.93
32	2- 34	0.0	********	0.0	156.00	34.16
33	3- 26	0.0	********	0.0	129.03	107.06
34	3- 27	0.0	********	0.0	198.00	122.55
35	3- 28	0.0	********	0.0	174.00	131.04
36	3- 29	0.0	********	0.0	150.00	105.74
37	3- 30	0.0	********	0.0	205.20	165.84
38	3- 31	16114.0	********	0.0	37.95	0.00
39	3- 32	0.0	********	0.0	144.00	48.96
40	3- 33	0.0	********	0.0	204.60	135.96
41	3- 34	0.0	********	0.0	108.00	77.24
42	4- 25	21782.0	********	0.0	90.00	0.00
43	4- 27	0.0	********	0.0	205.20	86.34
44	4- 28	0.0	********	0.0	148.50	62.12
45	4- 29	0.0	********	0.0	231.00	143.32
46	4- 30	0.0	********	0.0	180.00	97.23
47	4- 34	0.0	********	0.0	267.30	193.13
48	5- 25	0.0	********	0.0	237.60	158.98
49	5- 26	38552.0	********	0.0	54.00	0.00
50	5- 29	0.0	********	0.0	171.00	94.71
51	5- 30	0.0	********	0.0	163.35	91.96
52	5- 31	0.0	********	0.0	201.96	131.98
53	5- 34	0.0	********	0.0	181.50	118.71
54	5- 35	0.0	********	0.0	198.00	214.94
55	5- 36	0.0	********	0.0	188.10	150.13
56	6- 25	0.0	********	0.0	70.00	38.86
57	6- 27	39459.0	********	0.0	60.00	0.00
58	6- 28	0.0	********	0.0	62.54	35.02
59	6- 29	0.0	********	0.0	[illegible]	[illegible]

59	6- 29	0.0	********	0.0	[illegible]	[illegible]
60	6- 31	0.0	********	0.0	191.40	168.90
61	6- 34	0.0	********	0.0	178.20	162.89
62	7- 25	0.0	********	0.0	110.40	39.78
63	7- 26	0.0	********	0.0	238.00	191.99
64	7- 27	0.0	********	0.0	145.00	45.52
65	7- 28	0.0	********	0.0	67.00	0.00
66	7- 29	10799.0	********	0.0	68.30	0.00
67	7- 30	0.0	********	0.0	114.00	50.61
68	7- 31	0.0	********	0.0	189.00	127.01
69	7- 34	0.0	********	0.0	151.00	96.21
70	8- 25	0.0	********	0.0	113.85	57.53
71	8- 26	0.0	********	0.0	214.00	182.29
72	8- 27	0.0	********	0.0	132.00	46.82
73	8- 28	0.0	********	0.0	69.00	16.30
74	8- 29	9399.0	********	0.0	54.00	0.00
75	8- 30	0.0	********	0.0	134.00	84.91
76	8- 31	0.0	********	0.0	160.00	112.31
77	8- 33	0.0	********	0.0	225.00	146.62
78	8- 34	0.0	********	0.0	132.00	91.51
79	9- 25	0.0	********	0.0	264.00	176.77
80	9- 26	0.0	********	0.0	202.50	139.89
81	9- 28	0.0	********	0.0	252.00	168.39
82	9- 30	51128.0	********	0.0	80.00	0.00
83	9- 31	0.0	********	0.0	193.00	114.41
84	9- 34	0.0	********	0.0	217.80	146.40
85	10- 25	0.0	********	0.0	109.60	68.96
86	10- 26	0.0	********	0.0	112.20	96.18
87	10- 27	0.0	********	0.0	204.60	135.10
88	10- 28	0.0	********	0.0	132.00	94.99
89	10- 29	0.0	********	0.0	132.00	93.69
90	10- 31	14687.0	********	0.0	32.00	0.00
91	10- 32	0.0	********	0.0	166.50	77.41

92	10- 33	0.0	********	0.0	173.25	110.56
93	10- 34	0.0	********	0.0	110.40	85.59
94	11- 26	0.0	********	0.0	189.00	217.07
95	11- 27	0.0	********	0.0	202.50	177.09
96	11- 28	0.0	********	0.0	175.00	182.08
97	11- 29	0.0	********	0.0	162.00	167.78
98	11- 31	0.0	********	0.0	144.00	156.09
99	11- 32	10646.0	********	0.0	45.00	0.00
100	11- 33	0.0	********	0.0	66.00	47.40
101	11- 34	0.0	********	0.0	120.00	139.29
102	11- 35	0.0	********	0.0	112.20	211.21
103	11- 36	0.0	********	0.0	184.80	228.90
104	12- 26	0.0	********	0.0	227.70	189.79
105	12- 33	0.0	********	0.0	178.20	93.62
106	12- 35	0.0	********	0.0	178.20	211.23
107	12- 36	0.0	********	0.0	60.72	38.84
108	12- 37	7377.0	********	0.0	132.00	0.00
109	13- 26	0.0	********	0.0	198.00	106.07
110	13- 32	3305.0	********	0.0	165.00	0.00
111	13- 33	0.0	********	0.0	138.60	0.00
112	13- 35	0.0	********	0.0	91.08	70.09
113	13- 36	0.0	********	0.0	75.90	0.00
114	13- 37	0.0	********	0.0	191.40	5.38
115	14- 28	0.0	********	0.0	227.70	231.78
116	14- 29	0.0	********	0.0	195.00	197.78
117	14- 32	18463.0	********	0.0	48.00	0.00
118	14- 33	0.0	********	0.0	41.00	19.40
119	14- 34	0.0	********	0.0	162.00	178.29
120	14- 35	0.0	********	0.0	68.31	164.32
121	14- 36	0.0	********	0.0	204.60	245.70
122	15- 32	0.0	********	0.0	207.90	96.92
123	15- 33	0.0	********	0.0	214.50	129.92
124	15- 35	0.0	********	0.0	178.20	211.23

125	15= 36	0.0	********	0.0	106.26	84.50
126	15= 37	10946.0	********	0.0	132.00	0.00
127	16= 26	0.0	********	0.0	125.40	186.37
128	16= 31	0.0	********	0.0	183.15	228.14
129	16= 32	9551.0	********	0.0	12.10	0.00
130	16= 33	0.0	********	0.0	112.20	126.50
131	16= 34	0.0	********	0.0	191.40	243.59
132	16= 35	0.0	********	0.0	158.40	290.31
133	16= 36	0.0	********	0.0	132.00	209.00
134	16= 37	0.0	********	0.0	188.10	154.98
135	17= 26	0.0	********	0.0	186.00	137.63
136	17= 27	0.0	********	0.0	183.15	81.30
137	17= 28	0.0	********	0.0	167.40	98.04
138	17= 29	0.0	********	0.0	173.40	102.74
139	17= 30	0.0	********	0.0	227.70	161.95
140	17= 31	0.0	********	0.0	165.00	100.65
141	17= 32	2947.0	********	0.0	121.44	0.00
142	17= 33	0.0	********	0.0	145.20	50.16
143	17= 34	0.0	********	0.0	110.40	53.25
144	17= 35	0.0	********	0.0	187.00	209.57
145	17= 36	0.0	********	0.0	227.70	195.36
146	18= 32	2219.0	********	0.0	157.50	0.00
147	18= 33	0.0	********	0.0	132.00	0.90
148	18= 35	0.0	********	0.0	13.49	0.00
149	18= 36	0.0	********	0.0	198.00	129.60
150	18= 37	0.0	********	0.0	214.50	35.98
151	19= 25	0.0	********	0.0	198.00	164.17
152	19= 26	0.0	********	0.0	156.00	146.79
153	19= 27	0.0	********	0.0	178.20	115.51
154	19= 28	0.0	********	0.0	126.00	95.79
155	19= 29	0.0	********	0.0	114.00	82.49
156	19= 31	0.0	********	0.0	75.90	50.71
157	19= 32	0.0	********	0.0	126.00	43.71

158	19- 33	0.0	********	0.0	[illegible]	[illegible]
159	19- 34	3963.0	********	0.0	18.00	0.00
160	19- 35	0.0	********	0.0	178.20	239.92
161	20- 32	0.0	********	0.0	176.00	65.02
162	20- 33	0.0	********	0.0	102.00	17.42
163	20- 34	0.0	********	0.0	183.15	136.46
164	20- 35	0.0	********	0.0	41.00	74.03
165	20- 36	0.0	********	0.0	145.20	123.32
166	20- 37	1998.0	********	0.0	132.00	0.00
167	21- 25	0.0	********	0.0	213.18	156.05
168	21- 26	0.0	********	0.0	186.00	153.49
169	21- 27	0.0	********	0.0	132.00	46.01
170	21- 28	0.0	********	0.0	96.60	43.09
171	21- 29	0.0	********	0.0	75.90	21.09
172	21- 31	0.0	********	0.0	82.10	33.61
173	21- 32	0.0	********	0.0	156.00	50.41
174	21- 33	0.0	********	0.0	190.00	110.81
175	21- 34	8882.0	********	0.0	41.30	0.00
176	21- 35	0.0	********	0.0	198.00	236.42
177	22- 26	0.0	********	0.0	188.10	163.39
178	22- 32	0.0	********	0.0	185.00	87.22
179	22- 33	0.0	********	0.0	198.00	126.62
180	22- 35	0.0	********	0.0	138.60	184.83
181	22- 36	0.0	********	0.0	45.00	36.32
182	22- 37	6011.0	********	0.0	118.80	0.00
183	23- 32	0.0	********	0.0	268.95	248.97
184	23- 33	0.0	********	0.0	237.60	244.02
185	23- 35	0.0	********	0.0	132.00	256.03
186	23- 36	0.0	********	0.0	118.80	187.92
187	23- 37	11268.0	********	0.0	41.00	0.00
188	24- 25	0.0	********	0.0	316.80	209.26
189	24- 26	0.0	********	0.0	244.80	161.87
190	24- 32	8089.0	********	0.0	156.00	0.00

191	24- 33	0.0	********	0.0	[illegible]	[illegible]
192	24- 34	0.0	********	0.0	198.00	106.29
193	24- 35	0.0	********	0.0	189.00	177.01
194	25- 38	21782.0	66000.0	0.0	57.53	0.00
195	26- 39	39793.0	66000.0	0.0	40.47	0.00
196	27- 40	39459.0	66000.0	0.0	40.67	0.00
197	28- 41	0.0	66000.0	0.0	724.15	0.00
198	29- 42	20198.0	66000.0	0.0	60.18	0.00
199	30- 43	51128.0	66000.0	0.0	35.09	0.00
200	31- 44	39462.0	66000.0	0.0	40.67	0.00
201	32- 45	55220.0	64200.0	0.0	33.08	0.00
202	33- 46	0.0	64200.0	0.0	724.15	0.00
203	34- 47	12845.0	64200.0	0.0	77.86	0.00
204	35- 48	0.0	64200.0	0.0	724.15	0.00
205	36- 49	0.0	64200.0	0.0	724.15	0.00
206	37- 50	37600.0	64200.0	0.0	41.15	0.00
207	38- 51	0.0	********	0.0	417.60	150.27
208	38- 53	0.0	********	0.0	288.00	17.67
209	38- 54	0.0	********	0.0	440.00	102.50
210	38- 55	21782.0	********	0.0	324.00	0.00
211	39- 51	8045.0	********	0.0	309.00	0.00
212	39- 52	31748.0	********	0.0	430.00	0.00
213	39- 53	0.0	********	0.0	366.00	54.00
214	39- 54	0.0	********	0.0	417.00	37.83
215	39- 55	0.0	********	0.0	426.00	60.33
216	40- 51	0.0	********	0.0	475.20	219.87
217	40- 53	0.0	********	0.0	306.00	47.67
218	40- 54	0.0	********	0.0	423.00	97.50
219	40- 55	39459.0	********	0.0	312.00	0.00
220	41- 51	0.0	********	0.0	372.00	767.67
221	41- 53	0.0	********	0.0	280.80	673.47
222	41- 54	0.0	********	0.0	402.00	727.50
223	41- 55	0.0	********	0.0	348.00	687.00

224	42- 51	0.0	********	0.0	372.00	105.00
225	42- 53	20198.0	********	0.0	270.00	0.00
226	42- 54	0.0	********	0.0	390.00	52.83
227	42- 55	0.0	********	0.0	338.40	14.73
228	43- 51	7829.0	********	0.0	297.00	0.00
229	43- 52	0.0	********	0.0	495.00	77.00
230	43- 53	43299.0	********	0.0	300.00	0.00
231	43- 54	0.0	********	0.0	474.00	106.83
232	43- 55	0.0	********	0.0	363.00	9.33
233	44- 51	0.0	********	0.0	330.00	37.17
234	44- 52	0.0	********	0.0	510.00	96.17
235	44- 53	0.0	********	0.0	315.00	19.17
236	44- 54	39462.0	********	0.0	363.00	0.00
237	44- 55	0.0	********	0.0	366.00	16.50
238	45- 51	0.0	********	0.0	400.00	156.67
239	45- 52	0.0	********	0.0	545.00	180.67
240	45- 53	0.0	********	0.0	333.00	86.67
241	45- 54	5341.0	********	0.0	313.50	0.00
242	45- 55	49879.0	********	0.0	300.00	0.00
243	46- 51	0.0	********	0.0	420.00	841.35
244	46- 52	0.0	********	0.0	544.50	844.85
245	46- 53	0.0	********	0.0	354.00	772.35
246	46- 54	0.0	********	0.0	316.00	667.18
247	46- 55	0.0	********	0.0	321.00	685.68
248	47- 51	0.0	********	0.0	354.00	91.17
249	47- 52	0.0	********	0.0	534.00	150.17
250	47- 53	0.0	********	0.0	342.00	76.17
251	47- 54	12845.0	********	0.0	333.00	0.00
252	47- 55	0.0	********	0.0	342.00	22.50
253	48- 51	0.0	********	0.0	447.00	750.74
254	48- 52	0.0	********	0.0	491.40	674.14
255	48- 53	0.0	********	0.0	375.00	675.74
256	48- 54	0.0	********	0.0	310.20	543.77

257	48- 55	0.0	********	0.0	550.00	577.07
258	49- 51	0.0	********	0.0	580.00	938.65
259	49- 52	0.0	********	0.0	400.00	637.65
260	49- 53	0.0	********	0.0	423.00	778.65
261	49- 54	0.0	********	0.0	369.60	658.08
262	49- 55	0.0	********	0.0	372.00	673.98
263	50- 52	0.0	********	0.0	460.00	124.77
264	50- 53	0.0	********	0.0	510.00	292.77
265	50- 54	37600.0	********	0.0	284.40	0.00
266	50- 55	0.0	********	0.0	417.00	146.10
267	51- 56	15874.0	15874.0	15874.0	0.00	-1617.58
268	52- 56	31748.0	31748.0	31748.0	0.00	-1496.58
269	53- 56	63497.0	63497.0	63497.0	0.00	-1614.58
270	54- 56	95248.0	95248.0	95248.0	0.00	-1547.41
271	55- 56	111120.0	111120.0	111120.0	0.00	-1560.91
272	56- 1	317487.0	317487.0	317487.0	0.00	0.00

ANEXO III

LISTAGEM DO PROGRAMA FONTE

```
BEGIN
  FILE ENTRADA(KIND=READER);
  FILE SAIDA(KIND=PRINTER);
  INTEGER CONF,                % CONFIGURACAO
          M,                   % NUMERO DE ARCOS
          N,                   % NUMERO DE NOS
          LO,                  % NUMERO DE LOCAIS CANDIDATOS
          CA,                  % NUMERO DE ARCOS CANDIDATOS
          MOD,                 % INDICA SE ARMAZENS SAO MODULADOS
          FEO,                 % INDICA EQUILIBRIO EM CADA CONFIGURACAO
          TB,                  % TAMANHO DO BUFFER
          I,J,K,               % CONTADORES
          NT,                  % NUMERO DE TIPOS DE FUNCAO
          CHAMA,               % NUMERO DE CHAMADAS OUT-OF-KILTER
          FASE,
          VOLTA;               % STATUS DE RETORNO DA ACKILT
  REAL TOLERANCIA;
PROCEDURE BKILTE(DIM,NO,L,C,G,U,X,INDEX,STAT);
  VALUE DIM,NO;
  INTEGER DIM,NO,STAT;
  REAL ARRAY X[1],U[1],L[1],C[1],G[1];
  INTEGER ARRAY INDEX[1,1];
EXTERNAL; % EXECUTA ALGORITMO 'OUT-OF-KILTER'
```

```
READ(ENTRADA,<6I5,2I2,F10.4>,M,N,LC,CA,TE,NT,MOD,FED,
TOLERANCIA);
BEGIN
     LABEL FIM;
     INTEGER ARRAY NOME[1:CA,1:10], % NOME DO LOCAL
                   BUFER[1:TB+1],    % BUFFER
                   ARCOS[1:M,1:2],   % INDICES DOS NOS DO ARCO
                   TF[1:CA],         % TIPO DE FUNCAO DE CUSTO
                   SAINDO[1:N,1:N/2], % ARCOS QUE SAEM DO NO
                   CHEGA[1:N,1:N/2],  % ARCOS QUE CHEGAM AO NO
                   SAIN[1:N],        % NUMERO DE ARCOS SAINDO DO NO
                   CHEN[1:N],        % NUMERO DE ARCOS CHEGANDO AO NO
                   GRUP[1:LC,1:TE+1], % GRUPOS DE VIZINHOS
                   NVIZ[1:LC],       % NUMERO DE VIZ. NO GRUPO
                   ARCO[1:CA],        % NUMERO DO ARCO CANDIDATO
                   LOCAL[0:CA+1],    % LOCAL A QUE PERTENCE O ARCO
                   PRIMA[1:LC],      % PRIMEIRO ARCO DO LOCAL
                   NUMA[1:LC],       % NUMERO DE ARCOS NO LOCAL
                   PODE[1:LC];       % INDICA SE PODE EXCLUIR
     REAL ARRAY A[1:M],                   % LIMITE INFERIOR DO ARCO
                B[1:M],                   % CAPACIDADE DO ARCO
                C[1:M],                   % CUSTO NO ARCO(ATUAL)
                D[1:N],                   % VARIAVEIS DUAIS - NODE NUMBERS
                E[1:M],                   % CUSTO LIDO DO ARCO
                F[1:M],                   % FLUXO NO ARCO
                LS[1:10],                 % LIMITES DOS SEGMENTOS
                CR[1:CA],                 % CUSTO REAL NUM LOCAL
                CTA[1:CA],                % CUSTO TOTAL ATUAL NUM LOCAL
                CTR[1:CA],                % CUSTO TOTAL REAL NUM LOCAL
                CBEST[1:CA],
                VBEST[1:CA],
                IROT[1:LC],               % INDICE DE ROTACAO DO ARMAZEM
                VESTATICO[1:CA],          % VOLUME ESTATICO
                SAVEA[1:CA],              % SALVA F. MIN DURANTE DROP
                ALIDO[1:CA],              % GUARDA F. MINIMO LIDO
                SAVEC[1:CA],              % SALVA CUSTOS P/ PRINT
                ERRO[1:CA];
     REAL CTT,  % CUSTO DE UMA CONFIGURACAO
          ZBEST;% CUSTO DE UMA SOLUCAO ADOTADA
     BOOLEAN CONTINUAR;
```

```
PROCEDURE LEITURA;
BEGIN
     LABEL L1,L2,L3,FIM1;
     REAL SOMA,DIF;
     INTEGER I,J,SAVETIPC,TIPC,X,F,VIZ,Y,Z,V,W;
     % ********   LE CARTOES DESCRITORES DA REDE   ********
     FOR I:=1 STEP 1 UNTIL M DO
     BEGIN
          READ(ENTRADA,<3I5,3F10.2>,P,ARCOS[I,1],ARCOS[I,2],
          A[I],B[I],C[I]) [:L1:L1];
          IF ARCOS[I,1] > N OR ARCOS[I,2] > N THEN
          WRITE(SAIDA,<"ERRO EM N. DE NO- CARTAO ",I5>,P);
     END;
     J:=0;
     FOR I:=1 STEP 1 UNTIL LO DO
     BEGIN
          READ(ENTRADA,<10A1,F5.2,2I5,X15,I1>,NOME[I,*],
          IROT[I],PRIMA[I],NUMA[I],PODE[I]) [:L3:L3];
          FOR Y:=1 STEP 1 UNTIL NUMA[I] DO
          BEGIN
               J:=J+1;
               READ(ENTRADA,<3I5>,F,LOCAL[J],TF[J]) [ :L2:L2];
               ARCO[J]:=P;
               ALIDO[J]:=A[P];
              ALIDO[J]:=A[P];
          END;
     END;
     GO TO FIM1;
     L1: WRITE(SAIDA,<"ERRO NA LEITURA CARTAO-ARCO",I5>,F);
         GO TO FIM;
     L2: WRITE(SAIDA,<"ERRO NA LEITURA CARTAO-CANDIDATO",I5>,P);
         GO TO FIM;
     L3: WRITE(SAIDA,<"ERRO NA LEITURA ARCO CAND.",I5>,F);
FIM1:  WRITE(SAIDA,<"FIM NORMAL DO PROCEDIMENTO DE LEITURA">);
END;
```

```
REAL PROCEDURE CUSTO(ARG,TIPO);
  REAL ARG;  INTEGER TIPO;
  BEGIN
       CASE TIPO OF
       BEGIN
            0: CUSTO:=1000.0;
            1: CUSTO:=(1722000.0 + 52530*ARG**0.5)/(8*ARG);
            2: CUSTO:=(1500000+95000*ARG**0.5)/(8*ARG);
            3: CUSTO:=(1722000.0+52530.0*ARG**0.5)/(8*ARG);
            4: CUSTO:=(650.0*(ARG-9999))/ (8*(ARG-10000));
            5: CUSTO:=(2500000.0+650.0*ARG)/(8*ARG);
       END;
  END;
PROCEDURE REAVALIAR;
BEGIN
     COMMENT  ESTA ROTINA REAVALIA O CUSTO UNITARIO NUM ARCO DE ARNAZEM
     DE ACORDO COM O FLUXO.CORRESPONDE AO PASSO 2 DO METODO ITERATIVO;
     REAL ARGUMENTO;
     INTEGER K,          % CONTADOR DE LOCAIS
             I,          % FUNCAO
             J,          % ARCO CORRESPONDENTE AO LOCAL
             L;          % LOCAL
     FOR K:=1 STEP 1 UNTIL CA DO
     BEGIN
          I:=TF[K];    % TIPO DE FUNCAO DESTE LOCAL
          L:=LOCAL[K];
          J:=ARCO[K];
          VESTATICO[K]:=F[J]/IROT[L];
          CTA[K]:=C[J]*VESTATICO[K];
          ARGUMENTO:=MAX(VESTATICO[K],VMIN/2);
          CR[K]:=CUSTO(ARGUMENTO,I);
          CTR[K]:=CR[K]*VESTATICO[K];
     END;
END;
```

```
PROCEDURE REVER;
BEGIN
      COMMENT ESTA PROCEDURE E EQUIVALENTE A PROC. REAVALIAR
              MAS SO E USADA QUANDO OS LOCAIS CANDIDATOS SAO MODULARES;
      INTEGER I,J,K,L;
      PROCEDURE TRANCAR(L);
      VALUE L;  INTEGER L;
      BEGIN
      COMMENT  ESTA SUB-ROTINA TRANCA OS FLUXOS NOS ARCOS SATURADOS
               NO LOCAL L;
      INTEGER R;
      FOR R:=J-1 STEP -1 UNTIL PRIMA[L]
      DO A[R]:=F[R];
      END;
      FOR K:=1 STEP 1 UNTIL CA DO
      BEGIN
           L:=LOCAL[K];
           I:=TF[K];
           J:=ARCO[K];
           VESTATICO[K]:=F[J]/IROT[L];
           CTA[K]:=C[J]*VESTATICO[K];
           IF F[J]=0.0
           THEN CR[K]:=C[J]
           ELSE CR[K]:=CUSTO(VESTATICO[K],I);
           IF F[J-1]=B[J-1] AND LOCAL[K-1]=LOCAL[K]
           THEN BEGIN
                     IF F[J]=0.0
                     THEN CR[K]:=CUSTO(I,1);
                     IF CR[K-1] > CR[K]
                     THEN TRANCAR(L)
                     ELSE A[J-1]:=ALIDO[K-1];
                END;
                CTR[K]:=CR[K]*VESTATICO[K];
      END;
END;
```

```
PROCEDURE IMPRIMIR1;
BEGIN
      COMMENT ESTA ROTINA IMPRIME AS SOLUCOES DE EQUILIBRIO OBTIDAS;
      INTEGER I,J,K,L;
      FORMAT FMT1( "==============================================================
===================================================================================");
      FORMAT FMT2(X40,"CONFIGURACAO N.",I3,X12,"FASE",I2,X13,
      "TOL: ",F5.3);
      FORMAT FMT3(X4,"LOCAL",X7,"ARCO",X6,"CUSTO UNIT.  VOLUME      ",
      "CAPACIDADE  CUSTO UNIT.  CUSTO TOTAL     CUSTO TOTAL      ERRO");
      FORMAT FMT4(X28,"USADO       DINAMICO   ESTATICA       REAL",X9,
      "OBTIDO            REAL             RELATIVO");
      FORMAT FMT5(X2,10A1,X3,I3,"-",I3,X5,F7.2,X5,F8.1,X3,F8.1,X5,F7.2,
      X5,F12.2,X4,F12.2,X4,F6.4);
      FORMAT FMT6("  DIAGNOSTICO: ERRO RELATIVO NAO TOLERADO.",
      "CUSTOS SERAO REAVALIADOS");
      WRITE(SAIDA[SKIP 1]);
      WRITE(SAIDA,FMT1);
      WRITE(SAIDA,FMT2,CONF,FASE,TOLERANCIA);
      WRITE(SAIDA[SPACE(1)],FMT1);
      WRITE(SAIDA,FMT3);
      WRITE(SAIDA[SPACE(2)],FMT4);
      FOR K:=1 STEP 1 UNTIL CA DO
      BEGIN
           L:=ARCO[K];
           J:=LOCAL[K];
           WRITE(SAIDA[SPACE(2)],FMT5,NOME[J,*],ARCOS[L,1],ARCOS[L,2],
           SAVEC[K],F[L],VESTATICO[K],CR[K],CTA[K],CTR[K],ERRO[K]);
      END;
      IF CONTINUAR
      THEN BEGIN
                WRITE(SAIDA[SPACE(2)]);
                WRITE(SAIDA,FMT6);
           END;
END;
```

```
PROCEDURE IMPRIMIR2;
BEGIN
  COMMENT ESTA ROTINA IMPRIME O  ESQUEMA DE FLUXO CORRESPONDENTE
          A SOLUCAO FINAL OBTIDA PELO PROGRAMA;
  INTEGER I,J,K,L;
  REAL ARRAY QB[1:M];
  FORMAT FMTU("NUMERO DE OTIMIZACOES DA REDE DE FLUXO:",I5);
  FORMAT FMTX("ESQUEMA DE FLUXO NA SOLUCAO ENCONTRADA");
  FORMAT FMTY(X10,"SEQ.",X6,"ARCO",X16,"FLUXO       CAPACIDADE ",
  "     MINIMO       CUSTO         GEAR");
  FORMAT FMTZ(X10,I4,X7,I3,"-",I3,X13,F8.1,X5,F8.1,X6,F8.1,X3,F9.2,
  X5,F9.2);
  WRITE(SAIDA[SKIP 1 ]);
  WRITE(SAIDA[SPACE(2)],FMTU,CHAMA);
  WRITE(SAIDA[SPACE(2)],FMTX);
  WRITE(SAIDA[SPACE(2)],FMTY);
  FOR I:=1 STEP 1 UNTIL M DO
  BEGIN
     J:=ARCOS[I,1];  K:=ARCOS[I,2];
     QB[I]:=C[I]+D[J]-D[K];
     WRITE(SAIDA,FMTZ,I,ARCOS[I,1],ARCOS[I,2],F[I],B[I],A[I],C[I],
     QB[I]);
  END;
END;
```

```
PROCEDURE PRINTRES;
  BEGIN
  COMMENT ESTA ROTINA CALCULA OS CUSTOS ASSOCIADOS A UMA CONFIGURACAO
          E IMPRIME;
  INTEGER I, J;
  REAL CTP,CIA,CAR;
     FORMAT FMT7(" DIAGNOSTICO:  SOLUCAO OTIMA PARA O PROBLEMA DE ",
     "DISTRIBUICAO COMPATIVEL COM CURVA DE CUSTOS * "
     "SOLUCAO OTIMA LOCAL");
     FORMAT FMT8(X10,"CUSTOS EM TRANSPORTES POR PERIODO",X10,F13.2);
     FORMAT FMT9(X10,"AMORTIZACAO DO INVESTIMENTO EM ARMAZENAGEM:",
     F13.2);
     FORMAT FMT10(X10,"CUSTO TOTAL POR PERIODO NA SOL. ENCONTRADA:",
     F13.2);
  FORMAT FMT11(" DIAGNOSTICO: ESTA FOI A MELHOR SOLUCAO ENCONTRADA",
  "****************************************");
  CTT:=0.0;
  FOR J:=1 STEP 1 UNTIL CA DO
  BEGIN
       I:=ARCO[J];
       CIA:=CIA+CTR[J];
       CAR:=CAR+F[I]*C[I];
  END;
  FOR I:=1 STEP 1 UNTIL M DO CTT:=CTT+F[I]*C[I];
  CTP:=CTT-CAR;
  CTT:=CTP+CIA;
     WRITE(SAIDA[SPACE(3)]);
     WRITE(SAIDA[SPACE(2)],FMT8,CTP);
     WRITE(SAIDA[SPACE(2)],FMT9,CIA);
     WRITE(SAIDA[SPACE(2)],FMT10,CTT);
  IF FASE=4
  THEN WRITE(SAIDA,FMT11)
  ELSE WRITE(SAIDA,FMT7);
END;
```

```
PROCEDURE EQUILIBRAR;
  BEGIN
  COMMENT ESTA ROTINA APLICA O METODO ITERATIVO DE KING & LOGAN
        PARA OBTER UMA SOLUCAO DE EQUILIBRIO ENTRE CUSTOS E FLUXOS;
  INTEGER I,J,ITER;
     CONTINUAR:=TRUE;
     CONF:=CONF+1;
     WHILE CONTINUAR AND ITER < 20 DO
     BEGIN
        CHAMA:=CHAMA+1;
        BKILTE(M,N,A,B,C,D,F,ARCOS,VOLTA); %EXEC 'OUT-OF-KILTER'
        ITER:=ITER+1;
        IF VOLTA=1
        THEN BEGIN
             IMPRIMIR2;
             WRITE(SAIDA,<"PROBLEMA INFACTIVEL">);
             GO TO FIM;
        END;
        IF MOD=0
        THEN REAVALIAR
        ELSE REVER;
        CONTINUAR:=FALSE;
        FOR J:=1 STEP 1 UNTIL CA DO
        BEGIN
             I:=ARCO[J];
             SAVE[J]:=C[I];
             C[I]:=CR[J];
             IF CTR[J] > 0
             THEN ERRO[J]:=ABS(CTR[J]-CTA[J])/CTR[J]
             ELSE ERRO[J]:=0;
             IF ERRO[J] > TOLERANCIA
             THEN CONTINUAR:=TRUE;
        END;
     IF FED=1 AND FASE > 1
     THEN ITER:=20;
     END;
     IMPRIMIR1;
     PRINTRES;
END;
```

```
PROCEDURE EXCLUIR;
  COMMENT ESTA ROTINA APLICA A TECNICA DE "DFCP" PARA MELHORAR A
          SOLUCAO JA OBTIDA;
 BEGIN
 INTEGER I,J,K,L,Q,CAI;
 INTEGER ARRAY  NCRCPA[1:LO];
            GRUPC[1:CA],
            GRUPA[1:CA],
            VOLUME[1:LO];
 REAL BESTCICLO;
 COMMENT A PROCEDURE ENCHEBUFFER LISTADA A SEGUIR
 ATUA COMO SUB-ROTINA DA PROCEDURE EXCLUIR;
```

```
PROCEDURE ENCHEBUFER;   % INTERNA A ROTINA EXCLUIR
  BEGIN
  COMMENT ESTA SUB-ROTINA COLOCA NO BUFFER OS N "MELHORES"
          CANDIDATOS A EXCLUSAO EM CADA ESTAGIO;
  INTEGER ARRAY INTERBUF[1:LO+1];
  INTEGER I,J,K,L,M,MESC;
  FOR I:=1 STEP 1 UNTIL TB DO BUFER[I]:=0;
  FOR I:=1 STEP 1 UNTIL LO DO
  BEGIN
       FOR J:=1 STEP 1 UNTIL NUMA[I] DO
       BEGIN
            M:=M+1;
            VOLUME[I]:=VOLUME[I]+VBEST[M];
       END;
       IF VOLUME[I] > 0.0 AND NOFOPA[I]=0 AND PODE[I]=0
       THEN BEGIN
                 K:=K+1;
                 INTERBUF[K]:=I;
            END;
  END;
  FOR J:=1 STEP 1 UNTIL TB DO
  BEGIN
       M:=INTERBUF[J];
       FOR  I:=J+1 STEP 1 UNTIL K DO
       BEGIN
            L:=INTERBUF[I];
            IF VOLUME[L] < VOLUME[M]
            THEN BEGIN
                      MESC:=INTERBUF[J];
                      INTERBUF[J]:=INTERBUF[I];
                      INTERBUF[I]:=MESC;
                      M:=INTERBUF[J];
                 END;
       END;
       BUFER[J]:=INTERBUF[J];
  END;
```

```
   WRITE(SAIDA[SKIP 1 ]);
   WRITE(SAIDA[SPACE(2)],<"OS PROXIMOS CANDIDATOS A DROP SAO:">);
   FOR I:=1 STEP 1 WHILE INTERBUF[I] >0 AND I<=TB DO
   BEGIN
        M:=INTERBUF[I];
        WRITE(SAIDA,<10A1>,NOME[M,*]);
   END;
END;
```

```
 COMMENT          CORPO DA ROTINA EXCLUIR;
ENCHEBUFER;
WHILE BUFER[1] ¬= 0 DO
BEGIN
     BESTCICLO:=ZBEST;
     FOR L:=1 STEP 1 WHILE BUFER[L] ¬= 0 AND L <= TB DO
     BEGIN
          J:=BUFER[L];       % J -> LOCAL DE DROPP
          Q:=PRIMA[J]+NUMA[J]-1;
          FOR I:=PRIMA[J] STEP 1 UNTIL Q DO
          BEGIN
               A[I]:=0.0;
               C[I]:=CUSTO(VMIN,0);
          END;
          EQUILIBRAR;
          IF CTT < BESTCICLO
          THEN BEGIN          % SALVA SOLUCAO TEMPORARIA
                    CAI:=J;
                    BESTCICLO:=CTT;
                    FOR K:=1 STEP 1 UNTIL CA DO
                    BEGIN
                         GRUPC[K]:=CR[K];
                         I:=ARCO[K];
                         GRUPA[K]:=A[I];
                    END;
               END;
          IF CTT >= ZBEST
          THEN NDROPA[J]:=1;           % ARMAZEM  "MARCADO"
          FOR K:=1 STEP 1 UNTIL CA DO
          BEGIN
               I:=ARCO[K];
               C[I]:=CBEST[K];
               A[I]:=SAVEA[K];
          END;
     END;
```

```
      COMMENT  MELHOR SOLUCAO ATE AGORA E ACEITA COMO SOLUCAO ATUAL;
      IF BESTCICLO < ZBEST
      THEN BEGIN
                FOR J:=1 STEP 1 UNTIL CA DO
                BEGIN
                     CBEST[J]:=GRUPO[J];
                     SAVEA[J]:=GRUPA[J];
                     I:=ARCO[J];
                     C[I]:=CBEST[J];
                     A[I]:=SAVEA[J];
                END;
                ADROPA[CA]]:=1;
                ZBEST:=BESTCICLO;
           END;
           ENCHEBUFER;
 END;
END;




% **************************************************************************
%                          CORPO DO PROGRAMA
% **************************************************************************
      FASE:=1;
      LEITURA;
      FASE:=2;
      EQUILIBRAR;
      FASE:=3;
      ZBEST:=CTT;
      FOR J:=1 STEP 1 UNTIL CA DO
      BEGIN
            I:=ARCO[J];
            CBEST[J]:=C[I];
            SAVEA[J]:=A[I];
      END;
      EXCLUIR;
      FASE:=4;
      EQUILIBRAR;
      IMPRIMIR2;
   FIM: END;
END.
```

ANEXO IV

LISTAGEM DA ROTINA BKILIE

(algoritmo "out-of-kilter")

C D B M / B
= = = = = =

```
PROCEDURE BKILTE(DIM,NO,L,C,Q,U,X,INDEX,STAT);
  VALUE DIM,NO;
  INTEGER DIM,NO,STAT;
  REAL ARRAY X[1],U[1],L[1],C[1],Q[1];
  INTEGER ARRAY INDEX[1,1];
 BEGIN

      INTEGER ARRAY ESTADO[1:DIM];      % ESTADO DO ARCO- DE 1 A 9
      REAL ARRAY QBAR[1:DIM];           % QBAR(I,J)=Q(I,J)-U(I)+U(J)
      INTEGER ARRAY SAI[1:NO,1:NO/2],         % ARCOS QUE SAEM DO NO
                   ENTRA[1:NO,1:NO/2],    % ARCOS QUE ENTRAM NO NO
                   NSAI[1:NO] ,     NENTRA[1:NO];
      INTEGER Y,Z,W,V,T;
      INTEGER K,              %  VARIAVEL DE TRABALHO DE ARCO
              KT;             %  KT  ARCO SENDO PESQUISADO OUT-OF-KILTER
      LABEL RETORNO;
%  ================== A PROC STACALC CALCULA OS ESTADOS DOS ARCOS =====
  PROCEDURE STACALC(K);
  VALUE K;  INTEGER K;
   BEGIN
        INTEGER I,J;

        LABEL FIM;
        I:=INDEX[K,1];        J:=INDEX[K,2];
        QBAR[K]:=Q[K]+U[I]-U[J];
        IF QBAR[K] > 0
        THEN BEGIN
                IF X[K]= L[K]
                THEN ESTADO[K]:=1;   % ESTADO L   ** EM KILTER
                IF X[K] < L[K]
                THEN ESTADO[K]:=4;   % ESTADO L1 ** OUT-OF-KILTER
                IF X[K] > L[K]
                THEN ESTADO[K]:=5;   % ESTADO L2 ** OUT-OF-KILTER
                GO TO FIM;
            END;
        IF QBAR[K]= 0
        THEN BEGIN
                IF L[K] <= X[K] AND X[K] <= C[K]
                THEN ESTADO[K]:=2;   % ESTADO B ** EM KILTER
                IF X[K] < L[K]
                THEN ESTADO[K]:=6;   % ESTADO B1 ** OUT-OF-KILTER
                IF X[K] > C[K]
                THEN ESTADO[K]:=7;   % ESTADO B2 ** OUT-OF-KILTER
                GO TO FIM;
            END;
        IF QBAR[K] < 0
        THEN BEGIN
                IF X[K]= C[K]
                THEN ESTADO[K]:=3;   % ESTADO K ** EM KILTER
                IF X[K] < C[K]
                THEN ESTADO[K]:=8;   % ESTADO K1 ** OUT-OF-KILTER
                IF X[K] > C[K]
                THEN ESTADO[K]:=9;   % ESTADO K2 ** OUT-OF-KILTER
                GO TO FIM;
            END;
  FIM:
  END;  %  ================================================
```

```
PROCEDURE ROTULAT;
  BEGIN
  INTEGER ARRAY ROTULO[1:NO,1:4];

  COMMENT  - CADA ROTULO E UMA LINHA DE 4 ELEMENTOS:
           COLUNA 1- NO A PARTIR DO QUAL SE ROTULA
           COLUNA 2- NO ROTULADODO
           COLUNA 3- ARCO CORRESPONDENTE
           COLUNA 4- CONTEM 0 - SE AUMENTAR FLUXO
                            1 - SE DIMINUIR FLUXO
           O LABEL E COMPLETADO COM O ARRAY PSI
           QUE CONTEM O VALOR DO INCREMNETO;
  INTEGER ARRAY ISROT[1:NO];  COMMENT   INDICA SE NO ESTA ROTULADO
                                        1-SIM  0-NAO;
  INTEGER ARRAY ISCAN[1:NO];  % GUARDA NOS  JA VARRIDOS
  REAL ARRAY PSI[1:DIM];      % GUARDA INCREMENTO DE FLUXO NO ARCO
  INTEGER ICIC;               % CONTADOR DE NOS ROTULADOS
  REAL DELTA,INFINITO;
  LABEL FIMROTULAT;
% ==============PROC MUDA - ALTERA AS VARIAVEIS DUAIS ================
PROCEDURE MUDA;
  BEGIN
  INTEGER M,N,K;

  REAL DT1,DT2;
       DT1:=DT2:=INFINITO;
       FOR K:=1 STEP 1 UNTIL DIM DO
       BEGIN
            M:=INDEX[K,1];
            N:=INDEX[K,2];
            IF ISROT[M]=1 AND ISROT[N]=0 AND QBAR[K]>0 AND X[K] <=C[K]
            THEN IF QBAR[K] < DT1
                 THEN DT1:=QBAR[K];
            IF ISROT[M]=0 AND ISROT[N]=1 AND QBAR[K]<0 AND X[K]>=L[K]
            THEN IF -QBAR[K] < DT2
                 THEN DT2:=-QBAR[K];
       END;
       DELTA:=INFINITO;
       IF DT1 < DELTA
       THEN DELTA:=DT1;
       IF DT2 < DELTA
       THEN DELTA:=DT2;
       IF DELTA < INFINITO
       THEN BEGIN              % VAMOS ALTERAR VARIAVEIS DUAIS
                 FOR K:=1 STEP 1 UNTIL NO DO
                 BEGIN
                      IF ISROT[K]=0
                      THEN U[K]:=U[K]+DELTA;
                 END;
            END;
  END; % -----------------------------------------------------------

COMMENT      ************** CORPO DA PROCEDURE ROTULAT  **************;
      INTEGER META,D,KM,P,J;
      INFINITO:=1.0@10;
     IF ESTADO[KT]= 4         % ESTADO L1
      THEN BEGIN
                ROTULO[1,1]:=INDEX[KT,1];
                ROTULO[1,2]:=INDEX[KT,2];
                ROTULO[1,3]:=KT;
                ROTULO[1,4]:=0;         % AUMENTAR FLUXO
                PSI[1]:=L[KT]-X[KT];
                META:=INDEX[KT,1];
               P:=INDEX[KT,2]; ISROT[P]:=1;
           END;
```

```
IF ESTADO[KT]= 6 OR  ESTADO[KT]= 8          % ESTADOS B1 OU K1
THEN BEGIN
           ROTULO[1,1]:=INDEX[KT,1];
           ROTULO[1,2]:=INDEX[KT,2];
           ROTULO[1,3]:=KT;
           ROTULO[1,4]:=0;          % AUMENTAR FLUXO
           PSI[1]:=C[KT]-X[KT];
           META:=INDEX[KT,1];
          P:=INDEX[KT,2]; ISROT[P]:=1;
     END;
IF ESTADO[KT]= 5 OR ESTADO[KT]= 7     % ESTADOS L2 OU B2
THEN BEGIN
           ROTULO[1,1]:=INDEX[KT,2];
           ROTULO[1,2]:=INDEX[KT,1];
           ROTULO[1,3]:=KT;
           ROTULO[1,4]:=1;
           PSI[1]:=X[KT]-L[KT];
           META:=INDEX[KT,2];
          P:=INDEX[KT,1]; ISROT[P]:=1;
END;
IF ESTADO[KT]= 9                      % ESTADO K2
THEN BEGIN
           ROTULO[1,1]:=INDEX[KT,2];
           ROTULO[1,2]:=INDEX[KT,1];
           ROTULO[1,3]:=KT;
           ROTULO[1,4]:=1;          %  DIMINUIR FLUXO
           PSI[1]:=X[KT]-C[KT];
           META:=INDEX[KT,2];
          P:=INDEX[KT,1]; ISROT[P]:=1;
END;
COMMENT  JA ROTULOU-SE O PRIMEIRO NO. A SEGUIR PROCURA-SE
         NOVOS NOS A ROTULAR ATE QUE META SEJA ROTULADO;
ICIC:=1;
J:=1;
ISCAN[1]:=ROTULO[1,2];
WHILE(ROTULO[ICIC,2]¬=META AND ISCAN[J]¬=0) DO
BEGIN
     % PESQUISA ARCOS EMERGENTES
     Y:=ISCAN[J];
     FOR T:=1 STEP 1
     WHILE(T <= NSAI[Y] AND ROTULO[ICIC,2] ¬= META) DO
     BEGIN
           K:=SAI[Y,T];
           D:=INDEX[K,2];
           IF ISROT[D]=0
           THEN BEGIN
                     IF QBAR[K] > 0 AND X[K] < L[K]
                     THEN BEGIN
                               ICIC:=ICIC+1;
                               ROTULO[ICIC,1]:=INDEX[K,1];
                               ROTULO[ICIC,2]:=INDEX[K,2];
                               ROTULO[ICIC,3]:=K;
                               ROTULO[ICIC,4]:=0;
                               PSI[ICIC]:=MIN(PSI[J],L[K]-X[K]);
                               ISROT[D]:=1;
                          END;
                     IF QBAR[K] <=0 AND X[K] < C[K]
                     THEN BEGIN
                               ICIC:=ICIC+1;
                               ROTULO[ICIC,1]:=INDEX[K,1];
                               ROTULO[ICIC,2]:=INDEX[K,2];
                               ROTULO[ICIC,3]:=K;
                               ROTULO[ICIC,4]:=0;
                               PSI[ICIC]:=MIN(PSI[J],C[K]-X[K]);
                               ISROT[D]:=1;
                          END;
```

```
                    END;
        END;
                    % PESQUISA ARCOS INCIDENTES
                    Z:=ISCAN[J];
                    FOR T:=1 STEP 1 WHILE(T <= NENTRA[Z] AND
                    ROTULO[ICIC,2] ¬= META) DO
                    BEGIN
                         K:=ENTRA[Z,T];
                         D:=INDEX[K,1];
                              IF ISROT[D]=0
                              THEN
                              BEGIN
                                   IF QBAR[K] >=0 AND X[K] > L[K]
                                   THEN
                                   BEGIN
                                        ICIC:=ICIC+1;
                                        ROTULO[ICIC,1]:=INDEX[K,2];
                                        ROTULO[ICIC,2]:=INDEX[K,1];
                                        ROTULO[ICIC,3]:=K;
                                        ROTULO[ICIC,4]:=1;
                                        PSI[ICIC]:=
                                        MIN(PSI[J],X[K]-L[K]);
                                        ISROT[D]:=1;
                                   END;
                                   IF QBAR[K] < 0 AND X[K] > C[K]
                                   THEN
                                   BEGIN
                                        ICIC:=ICIC+1;
                                        ROTULO[ICIC,1]:=INDEX[K,2];
                                        ROTULO[ICIC,2]:=INDEX[K,1];
                                        ROTULO[ICIC,3]:=K;
                                        ROTULO[ICIC,4]:=1;
                                        PSI[ICIC]:=
                                        MIN(PSI[J],X[K]-C[K]);
                                        ISROT[D]:=1;
                                   END;
                              END;
                    END;
        J:=J+1;
        ISCAN[J]:=ROTULO[J,2];
    END;
    IF ISCAN[J]=0
    THEN BEGIN      % OCORREU BREAKTHROUGH
                    MUDA;  % ROTINA PARA MUDAR VARIAVEIS DUAIS
                    IF DELTA >= INFINITO
                    THEN BEGIN
                              STAT:=1;  % NAO HA SOL. VIAVEL
                              GO TO RETORNO;
                         END;
                              FOR K:=1 STEP 1 UNTIL DIM
                              DO STACALC(K); % RECALCULAR ESTADOS
                              GO TO FIMROTULAT;
        END;
    COMMENT AQUI,META FOI ROTULADO,OCORREU BREAKTHROUGH
            VAMOS AGORA ALTERAR O FLUXO NO CICLO ENCONTRADO;
    P:=META;
    FOR K:=ICIC STEP -1 UNTIL 1 DO
    BEGIN
        IF ROTULO[K,2]=P
        THEN BEGIN
                    KM:=ROTULO[K,3];
                    CASE ROTULO[K,4] OF
                    BEGIN
                         X[KM]:=X[KM]+PSI[ICIC];
                         X[KM]:=X[KM]-PSI[ICIC];
                    END;
```

```
                              P:=ROTULO[K,1];
                              STACALC(KM);
                         END;
          END;
      FIMROTULAT:
      END;  % ================================================================

      COMMENT  @@@@@@@@@@@    CORPO DA PROCEDURE AOKILT      @@@@@@@@@@@@@;
      FOR K:=1 STEP 1 UNTIL DIM DO
      BEGIN
           Y:=INDEX[K,1]; Z:=INDEX[K,2];
           NSAI[Y]:=W:=NSAI[Y]+1;
           NENTRA[Z]:=V:=NENTRA[Z]+1;
           SAI[Y,W]:=K;
           ENTRA[Z,V]:=K;
           STACALC(K);
      END;
      COMMENT
                @@@@@@@@   PESQUISA ARCOS OUT-OF-KILTER    @@@@@@@@@@@@;
      KT:=1;
      WHILE(KT <= DIM) DO
      BEGIN
           WHILE(ESTADO[KT] > 3) DO ROTULAT;  % ROTINA DE ROTULACAO
           KT:=KT+1;
      END;
      STAT:=0;  % SOLUCAO OTIMA ENCONTRADA
    RETORNO:
    END;
```

```
================================================================================
OF ERRORS DETECTED = 0.
OF SEGMENTS = 5.  TOTAL SEGMENT SIZE = 445 WORDS.  CORE ESTIMATE = 1012 WORDS.  STACK
 SIZE = 276 CARDS, 1818 SYNTACTIC ITEMS, 52 DISK SEGMENTS.
 FILE NAME:  CDBM/BKILTE.
TION TIME = 30.347 SECONDS ELAPSED;  4.706 SECONDS PROCESSING;  6.264 SECONDS I/O.
================================================================================
```

REFERÊNCIAS

1. BALINSKI, M.L., "Integer Programming: methods, uses and Computation"
Mgt.sci, 12:253-313 - nov., 1965

2. BEATTIE, D.W., "Improving the structure of a Distribution System".
Op.Res.Quarterly, 24:353-363 - set., 1973.

3. CANDLER, W., J.C. SNYDER e W. FAUGHT, "Convace Programming applied to rice mill location"
Amer. J. Agr.Econ., 54:126-13, 1972.

4. CIBRAZEM, "Alternativa de solução para o sistema de armazenagem a meio ambiente no estado do Maranhão".

5. EFROYMSON, M.A. e T.L. RAY, "A Branch-and-Bound algorithm for plant location"
Or. Res. 14:361-368, 1966.

6. ELSON, D.G., "Site location via mixed-integer programming"
Op. Res. Quarterly 23:31-43 , Março, 1972.

7. FELDMAN, E., F.A. LEHER e T.L. RAY, "Warehouse location under continuous economie of scale"
Mgt. Sci. 12:670-687, Maio, 1966.

8. FORD, L.R. e D.K. FULKERSON, "Flows in Networks".
Princenton University Press, 1962.

9. FRANK, M. e P. WOLFE, "An Algorithm for quadratic Programming"
Naval Res.Log. Quart 3:95-110, 1956.

10. GEOFFRION, A.M. e G.W. GRAVES, "Multicommodity Distribution System design by Benders decomposition"
Mgt. Sci, 20:822-844 - Jan, 1974.

11. HU, T.C., "Integer programming and network flows"
Addinson-Wesley, 1969.

12. KUEHN, A.A. e M.J. HAMBURGER, "A heuristic Program for locating warehouses"
Mgt. Sci. 9:643-666 , 1963.

13. KING, G.A. e S.H. LOGAN, "Optimum location, number and size of processing plants with row product and final product shipments".
J.Farm.Econ. 46:94-108, Fev. 1964.

14. LADD, G.W. e D.R. LIFFERTH, "An Analysis of alternative grain Distribution Systems"
Amer.J.Agr.Econ., 57:420-430 - Agosto, 1975.

15. LASDON, L.S., "Optimization Theory for Large Systems".
MacMillan, 1970.

16. MANNE, A.S., "Plant location under economies of scale - decentralization and computation"
Mgt.Sci. 11:213-235 - 1964.

17. MONTEROSSO, C., - "O algoritmo primal-dual out-of-kilter" COPPE-UFRJ, 1977.

18. ORDEN, A., "The transhipment problem" Mgt.Sci 2:276-285 - 1956

19. SĀ, G., "Branch-and-bound and approximate solutions to the capacited plant location problem" Op.Res. 17:1005-1016 - 1969.

20. SIMMONNARD, M., "Linear Programming" Prentice-Hall, 1966.

21. STOLLSTEIMER, J.F., "A working model for plant number and locations" J.Farm.Econ. 45:631-645 , Ago, 1965.

22. SPIELBERG, K., "Algorithms for the simple plant location with some side conditions" Op.Res. 12:85-11 - 1969.

23. SWEENEY, D.J. e R.L. TATHAN, "An improved long-run model for multiple warehouse location" Mgt. Sci. 22:748-758 - 1976.

24. TOFT, H.I., P.A. CASSIDY e W.O.McCARTHY, "Sensivity testing and the plant location problem" J.Farm.Econ. 52:403-412 - 1970.

25. TUI, H., "Concave programming under linear constraints" Soviet Math. 5:1437-1440 , Nov-Dez, 1964.

26. WARRACK, A.A. e FLETCHER, L.B., "Plant location model suboptimization for large problems" J.Farm.Econ., 52:587-590 , Nov-Dez, 1970.