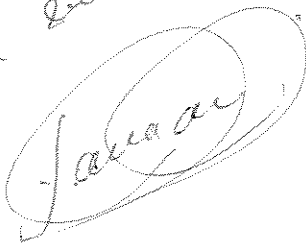


*e Exemplar correspondente
redação final da tese defendida
da por Hamilton Luchesi e aprovada
de Comissão Julgadora em 29/12/89*



UNIVERSIDADE ESTADUAL DE CAMPINAS

FACULDADE DE ENGENHARIA ELÉTRICA

DEPARTAMENTO DE TELEMÁTICA

"ALOCÇÃO DE MATERIAL RODANTE EM FERROVIAS"

AUTOR : HAMILTON LUCHESI

ORIENTADOR : PROF.DR. AKEBO YAMAKAMI

Tese apresentada à Faculdade de Engenharia Elétrica
como parte dos requisitos exigidos para a obtenção do título de
Mestre.

DEZEMBRO - 1989

Aos

meus pais

Alvemar e Luiza

Agradecimentos :

A Deus.

Ao Akebo, pela orientação e atenção.

Ao Christiano e ao Clovis, pelas sugestões.

A Capes, pela ajuda financeira.

A Fepasa, pelo fornecimento de dados.

Ao Miguel, Gorgonio, Sérgio e Paulo Correia, pela
ajuda técnica.

Aos colegas da Pós e especialmente aos da
Telemática, pelo companheirismo.

Aos meus irmãos : Aluemar, Marcelo, Márcio e
Eliane, pela força.

A Rô, que muito me ajudou.

ÍNDICE

Capítulo I . Introdução

I . 1. Apresentação do problema	1
I . 2. Definição da tese	4

Capítulo II . Modelagem do Problema

II . 1. Introdução	6
II . 2. Função objetivo	7
II . 3. Restrições	7
II . 3. 1. Equações de balanceamento	8
II . 3. 2. Restrições de limitação de tração	12
II . 3. 3. Equações de limitação de frota	14
II . 4. Comentários	20

Capítulo III . Método de Resolução

III. 1. Apresentação do método	22
III. 1. 1. Equacionamento	22
III. 1. 2. Caracterização da base	24
III. 1. 3. Especialização do método Primal Simplex	27
III. 1. 3. 1. Cálculo das variáveis duais	27
III. 1. 3. 1. a. Cálculo da partição π^2	29
III. 1. 3. 1. b. Cálculo da partição π^1	30

III. 1. 3. 2. Atualização da coluna que entra na base	32
III. 1. 3. 2. a. Cálculo da partição p^2	33
III. 1. 3. 2. b. Cálculo da partição p^1	34
III. 1. 3. 2. c. Cálculo da partição p^B	35
III. 2. Adaptação do método	35
III. 3. Apresentação de diagrama de blocos	36

Capítulo IV . Algoritmo e Implementação Computacional

IV . 1. Algoritmo	38
IV . 2. Implementação computacional	42
IV . 2. 1. Estrutura do programa Multipro	42
IV . 2. 1. 1. Programa principal	42
IV . 2. 1. 2. Sub-rotina : Artificial	43
IV . 2. 1. 3. Sub-rotina : Cálculo_do_fluxo_das_ variáveis_de_folga	45
IV . 2. 1. 4. Sub-rotina : Encontra_ π^2	45
IV . 2. 1. 4. 1. Sub-rotina : Inter	45
IV . 2. 1. 4. 2. Sub-rotina : Árvore_vezes_um_arco	45
IV . 2. 1. 4. 3. Sub-rotina : Verifica_leva	46
IV . 2. 1. 4. 4. Sub-rotina : Inverte	46
IV . 2. 1. 5. Sub-rotina : Encontra_ π^1	46
IV . 2. 1. 6. Sub-rotina : Encontra_candidato_a_base	46
IV . 2. 1. 7. Sub-rotina : Encontra_ p^2	47
IV . 2. 1. 8. Sub-rotina : Encontra_ p^1	47
IV . 2. 1. 9. Sub-rotina : Encontra_ p^B	47
IV . 2. 1.10. Sub-rotina : Cálculo_do_acrécimo_máximo	47
IV . 2. 1.11. Sub-rotina : Atualiza_o_fluxo_das_ variáveis_na_base	47
IV . 2. 1.12. Sub-rotina : Variável_a_sair_da_base	48
IV . 2. 1.13. Sub-rotina : Reestruturação_da_base	48
IV . 2. 1.13. 1. Sub-rotina : Reestrutura_árvore	48
IV . 2. 1.13. 2. Sub-rotina : Encontra_arco_corretor	48

Capítulo V . Aplicações

V . 1. Definição dos exemplos	84
V . 1. 1. Exemplo 1	85
V . 1. 2. Exemplo 2	90
V . 2. Apresentação dos resultados	93
V . 2. 1. Resultados do exemplo 1	93
V . 2. 2. Resultados do exemplo 2	94
V . 3. Comparações	95

Capítulo VI . Conclusões Finais

VI . 1. Conclusões	98
VI . 2. Propostas futuras	99

<u>Apêndice</u>	101
Apêndice A	101
Apêndice B	107

<u>Bibliografia</u>	133
---------------------------	-----

RESUMO

Modelamos um problema de alocação de carga em ferrovias sujeito a um conjunto de restrições. Esse tipo de problema é enfrentado diariamente pelas companhias ferroviárias.

O objetivo do problema é encontrar quais são os melhores caminhos a serem seguidos pelos produtos. O modelo prevê o caso da impossibilidade de se transportar todos os produtos, devido as restrições impostas, criando uma estrutura de caminhos alternativos para se transportar os produtos em excesso.

A modelagem envolve restrições de tração dos trechos de linha, de limitação das frotas de tipos de vagões e de balanceamento de produtos nas estações.

A utilização do método Simplex para a solução de problemas reais se torna inviável devido as proporções que o problema toma e por isso devemos buscar novas alternativas para resolvê-lo.

Fazemos um estudo dos métodos existentes na bibliografia para problemas com a estrutura do nosso modelo. Dentre estes, escolhemos um método de multifluxo em rede. Basicamente, este é uma especialização do método Primal-Simplex Revisado aplicado a uma rede de arcos e nós.

Mostramos o algoritmo do método e resultados de alguns exemplos simulados.

CAPÍTULO I

Introdução

I . 1. Apresentação do Problema

O transporte ferroviário de carga é um importante meio de transporte, sendo o mais adequado para produtos pesados e volumosos tais como grãos, minérios, etc.

Por estar em constante concorrência com os demais meios de transporte existe a preocupação em melhorar a sua produtividade e racionalizar suas decisões de planejamento.

A rede ferroviária consiste de várias estações e trechos de linha. Os produtos chegam nas estações para serem transportados para os seus destinos.

Nas últimas décadas tem aparecido diversos modelos voltados para os problemas das ferrovias. Basicamente existem modelos de Otimização e Simulação. Os dois tipos de modelos são importantes pois um pode vir a complementar o outro, ou mesmo trabalharem juntos.

Existem vários artigos sobre o assunto na literatura e alguns ([Assad, 1980] e [Haghani, 1987]) trazem uma sinopse dos melhores artigos sobre modelagem de problemas das ferrovias.

Os modelos de simulação visam testar a eficiência e a

factibilidade de novas políticas antes de implementá-las e fazer uma análise de soluções alternativas no caso de imprevistos no sistema ferroviário. Esses modelos são compostos por fórmulas matemáticas e regras lógicas especificadas pelo modelador baseado em dados concretos.

A longo prazo pode-se utilizar modelos de simulação para prever locais de melhoramentos. Esses modelos servem também para melhorar as decisões de planejamento da ferrovia.

A médio prazo tem-se buscado modelos que simulam linhas ou estações. Os modelos simuladores de estações estão preocupados em encontrar os pontos críticos nas operações, tais como custo operacional da estação e pontos de congestionamento. Esses modelos simulam as operações da estação.

As linhas merecem também modelos para simular movimentos e atividades de despachos. Esses simuladores investigam a capacidade das linhas, relatam pontos de congestionamentos e avaliam regras de prioridades em cruzamentos na linha.

Para as tarefas do dia a dia, muitas vezes são utilizados modelos de otimização. Essas tarefas diárias são roteamento de vagões pela rede, determinação da melhor alocação dos vagões vazios, etc. Esses modelos procuram apresentar o sistema da forma mais real possível. Os objetivos almejados podem ser minimização do custo do transporte, maximização do lucro ou minimização dos atrasos, dependendo do modelo.

Os modelos de otimização para rede são de especial interesse para esse trabalho, pois este focaliza um modelo de otimização voltado para minimização dos custos de roteamento de vagões.

A minimização do custo é conseguida alocando da forma mais econômica os recursos da ferrovia, por exemplo escolha dos melhores caminhos para os trens, melhor distribuição dos vagões vazios para as estações que precisem deles, melhor utilização da capacidade das estações, etc.

Na literatura encontramos modelos que tratam de diferentes aspectos do problema, tais como modelos para distribuição de vagões vazios ([Mendiratta, 1982], [Glickman, 1985] e [Philip, 1977]), modelos para classificação de trens que chegam numa estação intermediária ([Bodin, 1980] e [Turnquist, 1982]), etc.

O modelo de [Mendiratta, 1982] é representado por dois modelos: rede e estoque. O objetivo da rede é maximizar seus lucros, estando sujeita às restrições impostas pela demanda e o suprimento de vagões vazios. O modelo de estoque apresenta demandas estocásticas e prazos de entrega dos vagões vazios. Esses modelos são resolvidos alternativamente até encontrar a solução. O modelo da rede é formulado como um problema de multifluxo.

No modelo de [Glickman, 1985] uma melhor distribuição dos vagões é tentada para um grupo de ferrovias que compartilham os seus vagões. Esses modelos têm duas abordagens: a primeira, distribuir os vagões visando minimizar os custos e a segunda, beneficiar cada uma das ferrovias individualmente. A solução proposta é uma decomposição e um algoritmo para problemas de fluxo de custo mínimo numa rede.

Philip (1977), aplica um modelo de controle de estoque para distribuir os vagões vazios.

O modelo de [Bodin, 1980] procura determinar a melhor estratégia de classificação para todas as estações de classificação no sistema ferroviário em um prazo de tempo. O modelo pode ser visto como um problema de multifluxo não linear com muitas restrições laterais. Utilizou-se um grande pacote de programação linear para resolver o problema.

No modelo de [Turnquist, 1982] são propostos diversos modelos de filas para melhorar a eficiência das estações de classificação. O objetivo é diminuir os atrasos que advêm dessas etapas.

Os modelos de otimização de problemas de roteamento têm uma formulação de multifluxos. Encontramos em [Assad, 1978] um estudo dos principais algoritmos de multifluxo lineares e não lineares. Outra fonte de informação para o assunto é [Kennington, 1980] que também traz algoritmos de multifluxo.

Duas abordagens básicas têm sido empregadas para desenvolver técnicas especializadas em problemas de multifluxo, particionamento e decomposição.

A abordagem de decomposição coloca o problema de multifluxo numa forma onde um problema mestre de otimização coordena a solução de sub-problemas e cada um dos quais é um problema de fluxo de custo mínimo numa rede.

A abordagem de particionamento é uma especialização do método

simplex revisado ([Bazaraa, 1977] e [Kennington, 1980]) onde a base atual é particionada para explorar sua estrutura especial bloco angular.

Um dos pioneiros na abordagem de particionamento para problemas de multifluxo foi [Hartman, 1972]. Ele desenvolve um algoritmo que está baseado em problemas do tipo GUB (Generalized Upper Bounding), que são problemas que têm uma estrutura bloco angular e um conjunto de equações de acoplamento.

Outro trabalho importante sobre o assunto é o de [Kennington, 1977], que se baseia no algoritmo de [Hartman, 1972] para aplicá-lo a um problema de transporte com multifluxo. Kennington(1977) propõe um algoritmo para um problema de transportes multifluxo, comenta a estrutura de dados a ser utilizada, mostra quais simplificações podem ser feitas nas operações básicas do método simplex revisado e traz resultados computacionais para alguns exemplos.

Mais recentemente [Kennington, 1980] aborda o seu algoritmo de uma forma geral, podendo ser aplicado a uma rede qualquer.

I . 2. Definição da Tese

Trataremos com um modelo que está preocupado em encontrar os melhores caminhos para os vagões de carga numa ferrovia sujeita a restrições adicionais. O modelo trabalha com vários produtos que precisam ser levados para os seus destinos. O custo do transporte de um determinado produto por um determinado trecho da linha recebe um valor por tonelada transportada. Esses custos podem variar de produto para produto.

A frota de vagões é constituída de diversos tipos (por exemplo, tipo tanque, frigorífico, fechado e outros). A frota de cada um desses tipos é limitada em número de vagões.

A cada trecho de linha associamos uma capacidade máxima de carga que pode ser tracionada.

Estamos considerando que a rede ferroviária é estática, ou seja, não haverá variação nas demandas de produtos e não existe preocupação

com a duração das viagens.

O objetivo desta tese é criar um modelo para representar o problema de roteamento de vagões de carga numa ferrovia com restrições laterais.

Nós nos restringiremos na abordagem particionada, dada a extensão do trabalho envolvido na análise e desenvolvimento de cada método, mas os interessados na abordagem de decomposição podem obter maiores informações em [Assad, 1978] e [Kennington, 1980].

O outro motivo que nos levou a escolher este algoritmo em relação a outros existentes foi que esse segue os mesmos passos do bem conhecido algoritmo primal-simplex. Além disso o algoritmo trabalha muito com grafos, realizando quase que exclusivamente operações aditivas.

Segundo [Helgason, 1977] a abordagem particionada é o melhor meio de estender os limites no tamanho dos problemas de multifluxo que podem ser resolvidos. No nosso trabalho, objetivamos implementar algoritmos em microcomputadores, sem no entanto perder essa característica.

No capítulo II, apresentamos a modelagem do problema, introduzimos a função objetivo e as restrições do modelo.

Apresentamos o método primal-particionado e a adaptação do mesmo ao nosso modelo no capítulo III.

O algoritmo e o diagrama de blocos do programa são mostrados no capítulo IV.

No capítulo V, mostramos dois exemplos de problemas de multifluxo que ilustram o modelo e o algoritmo em termos computacionais.

Colocamos nossas conclusões finais e propostas futuras no capítulo VI.

CAPÍTULO II

Modelagem do Problema

II . 1. Introdução

Nosso objetivo é resolver um problema de escoamento de vários produtos numa mesma malha ferroviária. A cada um dos produtos associamos uma origem, um destino e a quantidade que se deseja transportar.

O transporte dos produtos é realizado por meio de vagões de carga, que compartilham a mesma malha ferroviária. Existem diferentes tipos de vagões para levar os produtos. Cada tipo de vagão leva apenas determinados tipos de produtos.

Pode haver mais do que um caminho para se transportar um produto da sua origem para o seu destino, de forma que o mais vantajoso é escolhido.

Além do problema de definir as rotas a serem tomadas pelos produtos, inserimos dois tipos de restrições: restrições de tração e de frota.

Modelamos o problema como um problema de otimização linear.

II . 2. Função Objetivo

A função objetivo do problema é minimizar o custo do transporte dos produtos pela malha.

Associamos um custo por tonelada a cada produto transportado em cada trecho da malha.

Analiticamente essa função pode ser escrita como:

$$\min \sum_{k=1}^{\mathcal{K}} \sum_{i=1}^{\mathcal{G}} c_i^k x_i^k$$
$$x_i^k, c_i^k \geq 0, \text{ para } \forall i, k$$

onde:

\mathcal{K} : número de produtos

\mathcal{G} : número de trechos da malha

c_i^k : custo do transporte de uma tonelada do produto k no trecho i

x_i^k : fluxo de carga (em toneladas) do produto k pelo trecho i

Alternativamente, no lugar de minimizar o custo do transporte dos produtos, podemos maximizar o lucro do transporte dos produtos.

II . 3. Restrições

São três os tipos de restrições envolvidas no problema: de

balanceamento, de limitação de frota e de limitação de tração. Essas restrições são descritas e analisadas nos itens seguintes.

II . 3. 1. Equações de Balanceamento

O balanço de fluxos de um produto numa estação é a soma dos fluxos que chegam e saem da estação. Convencionamos que os fluxos que chegam à estação recebem um sinal positivo e os fluxos que deixam a estação recebem um sinal negativo.

O resultado do balanço de fluxos de um produto numa estação tem apenas três possibilidades: $+q$, $-q$ ou zero, onde q é quantidade do produto transportado na malha.

Resulta em $+q$ quando a estação for a de origem, em $-q$ quando a estação for a de destino e zero quando não for nenhuma das anteriores (estação de passagem).

Essas equações garantem o balanço de cada um dos produtos nas estações.

O exemplo 1 ilustra o balanceamento de fluxos para uma malha ferroviária.

Exemplo 3.1 :

Vamos tomar como exemplo uma malha ferroviária com apenas 4 estações e 4 trechos de linha. Deseja-se transportar 100 toneladas de aço da estação 1 para a estação 4.

A figura (3.1) ilustra a malha (suponha que seja possível transportar o produto apenas no sentido indicado pelas setas).

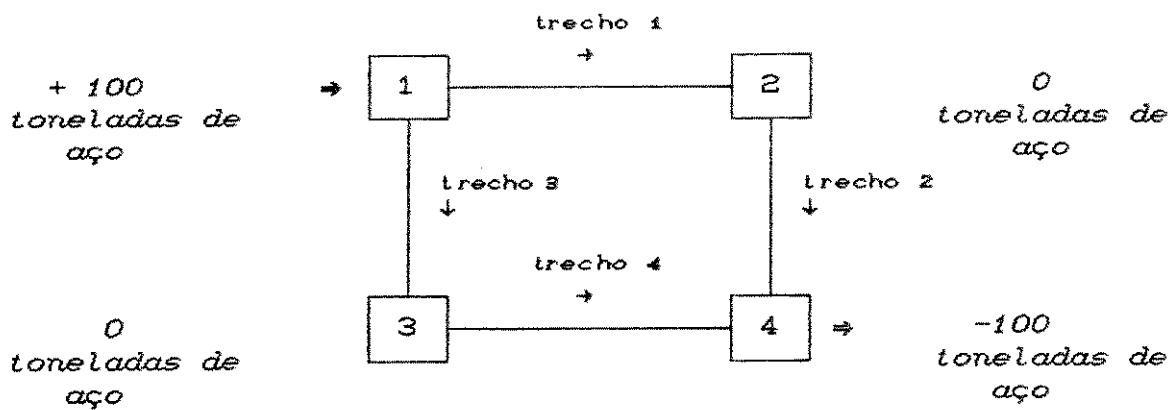


fig. 3.1 - malha ferroviária

As equações referentes aos balanceamentos de fluxos nas estações são:

$$\text{estação 1 : } x_1^{\text{aço}} + x_3^{\text{aço}} = +100$$

$$\text{estação 2 : } -x_1^{\text{aço}} + x_2^{\text{aço}} = 0$$

$$\text{estação 3 : } -x_3^{\text{aço}} + x_4^{\text{aço}} = 0$$

$$\text{estação 4 : } -x_2^{\text{aço}} - x_4^{\text{aço}} = -100$$

$$x_i^{\text{aço}} \geq 0 \quad , \quad \text{para } i=1, \dots, 4$$

Escrevendo as equações obtidas no exemplo 3.1 numa forma matricial, temos:

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & -1 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} x_1^{\text{aço}} \\ x_2^{\text{aço}} \\ x_3^{\text{aço}} \\ x_4^{\text{aço}} \end{bmatrix} = \begin{bmatrix} +100 \\ 0 \\ 0 \\ -100 \end{bmatrix}$$

que podemos representar da seguinte maneira:

$$A \cdot x^{\text{aço}} = r^{\text{aço}}$$

onde:

A : matriz dos coeficientes

$x^{\text{aço}}$: vetor coluna dos fluxos de aço pelos trechos

$r^{\text{aço}}$: vetor coluna dos requerimentos das estações

Suponha que temos mais um produto para ser transportado na mesma malha ferroviária do exemplo 3.1. Desejamos transportar 200 toneladas de café da estação 3 para a estação 4. As equações referentes aos balanceamentos dos fluxos de café nas estações são:

$$\begin{aligned} \text{estação 1} & : & x_1^{\text{café}} & & + x_3^{\text{café}} & & = & 0 \\ \text{estação 2} & : & -x_1^{\text{café}} & + & x_2^{\text{café}} & & = & 0 \\ \text{estação 3} & : & & & -x_3^{\text{café}} & + & x_4^{\text{café}} & = & 200 \\ \text{estação 4} & : & & & -x_2^{\text{café}} & & -x_4^{\text{café}} & = & -200 \end{aligned}$$

$$x_i^{\text{café}} \geq 0 \quad , \quad \text{para } i=1, \dots, 4$$

Colocando também na forma matricial temos:

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & -1 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} x_1^{\text{café}} \\ x_2^{\text{café}} \\ x_3^{\text{café}} \\ x_4^{\text{café}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 200 \\ -200 \end{bmatrix}$$

De uma forma mais compacta:

$$A \cdot x^{\text{café}} = r^{\text{café}}$$

As equações dos balanceamentos desses dois produtos nas estações da malha são:

$$A \cdot x^{\text{aço}} = r^{\text{aço}}$$

$$A \cdot x^{\text{café}} = r^{\text{café}}$$

$$x^{\text{aço}}, x^{\text{café}} \geq 0$$

Notemos que a matriz A é comum nas duas equações de balanceamento.

Podemos generalizar as equações de balanceamento dos fluxos nas estações, para uma malha ferroviária com \mathcal{K} produtos.

$$A x^k = r^k$$

$$x^k \geq 0, \quad k=1, \dots, \mathcal{K}$$

onde:

A : matriz comum às equações de balanceamento

x^k : vetor coluna $(x_1^k, \dots, x_g^k)^t$

r^k : vetor coluna dos requerimentos das estações do

produto k

Esse vetor tem a seguinte estrutura:

$$\text{produto } k \begin{cases} +q^k, & \text{se for estação origem} \\ -q^k, & \text{se for estação destino} \\ 0, & \text{caso contrário} \end{cases}$$

q^k : quantidade (em toneladas) do produto k que se quer transportar

II . 3. 2. Restrições de Limitação de Tração

Um outro tipo de restrição que aparece numa malha ferroviária é a limitação de tração nos trechos da malha. Os parâmetros que entram no cálculo da máxima tração num trecho de linha são: a potência da locomotiva que vai puxar os vagões, características topográficas do trecho, etc. Usamos *toneladas* como unidade de medida para a máxima tração num trecho.

Associamos um valor para a máxima tração em cada um dos trechos da linha, valor esse dado em toneladas. Devemos garantir que a soma dos fluxos dos produtos em qualquer trecho não exceda a sua capacidade de tração. Cada trecho de linha gera uma restrição de tração.

Tomamos a malha do exemplo 3.1 para exemplificar como são montadas as restrições de tração para os trechos da malha. Nesse exemplo temos quatro trechos de linha, o que implica em quatro restrições de tração.

As equações de limitação de trecho são:

$$\begin{array}{rcll} x_1^{\text{aço}} + x_1^{\text{café}} & \leq & b_1 & (\text{trecho } 1) \\ x_2^{\text{aço}} + x_2^{\text{café}} & \leq & b_2 & (\text{trecho } 2) \\ x_3^{\text{aço}} + x_3^{\text{café}} & \leq & b_3 & (\text{trecho } 3) \\ x_4^{\text{aço}} + x_4^{\text{café}} & \leq & b_4 & (\text{trecho } 4) \end{array}$$

onde:

b_i : máxima tração do trecho i , $i=1, \dots, 4$

que pode ser colocado na forma matricial dada abaixo:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1^{\text{açúcar}} \\ x_2^{\text{açúcar}} \\ x_3^{\text{açúcar}} \\ x_4^{\text{açúcar}} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1^{\text{café}} \\ x_2^{\text{café}} \\ x_3^{\text{café}} \\ x_4^{\text{café}} \end{bmatrix} \leq \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$$

De uma forma mais compacta:

$$I_4 \cdot x^{\text{açúcar}} + I_4 \cdot x^{\text{café}} \leq b$$

onde:

I_4 : matriz identidade de ordem 4

b : vetor coluna dos limites de tração dos trechos da malha ferroviária

Podemos generalizar as restrições que garantem que a soma dos fluxos dos produtos que passam por um trecho da linha não excede a sua capacidade de tração, para uma malha ferroviária de \mathcal{X} produtos e \mathcal{S} número de trechos.

Temos as seguintes equações:

$$\sum_{k=1}^{\mathcal{X}} I_{\mathcal{S}} x^k \leq b$$

$$0 \leq x^k \quad , \quad k=1, \dots, \mathcal{X}$$

onde:

$I_{\mathcal{S}}$: matriz identidade de ordem \mathcal{S}

b : vetor coluna das capacidades de tração dos trechos

Para trabalharmos com apenas equações de igualdades, inserimos variáveis de folga em cada uma das restrições de tração. Denotamos essas variáveis de folga por f_i , para $i=1, \dots, \mathcal{S}$, e por f o vetor coluna dessas variáveis de folga.

Procedendo desta forma obtemos:

$$\sum_{k=1}^{\mathcal{K}} I_{\mathcal{S}}^k x^k + f = b$$

$$x^k \geq 0 \quad , \quad k=1, \dots, \mathcal{K} \quad \text{e} \quad f \geq 0$$

II . 3. 3. Equações de Limitação de Frota

A outra restrição que aparece no modelo é a restrição de frota.

Criamos equações que garantem que para nenhum dos tipos de vagões seja ultrapassada a sua frota máxima:

Existe uma variedade de tipos de vagões e normalmente cada um dos produtos pode ser levado por mais do que um desses tipos.

Na nossa modelagem supomos que cada produto é levado por apenas um tipo de vagão. Mas cada tipo de vagão pode levar vários produtos. Numa situação real, um produto pode ser levado por vários tipos de vagões.

Marcamos os trechos que saem da estação origem de cada um dos produtos. É bom enfatizar que cada produto tem apenas uma estação origem e uma estação destino.

Na figura (3.2) ilustramos uma malha ferroviária com as estações e os trechos de linha. Os trechos marcados estão representados pelas linhas duplas.

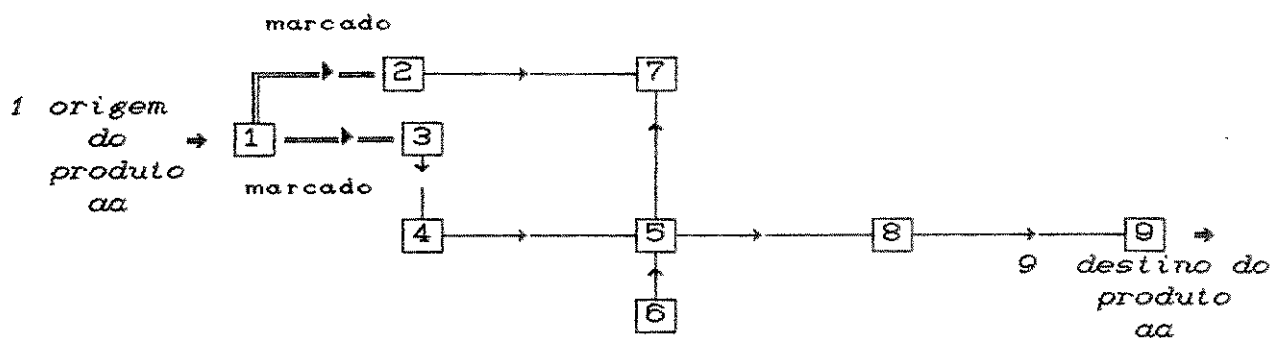


fig. 3.2 - malha ferroviária com trechos marcados

De modo análogo, marcamos todos os trechos que deixam a origem dos demais produtos. Para um determinado tipo de vagão, somando todos os fluxos dos trechos marcados dos produtos que ele leva, estamos calculando a quantidade de fluxo levada por esse tipo de vagão, que é aquilo que procuramos. Devemos impor que essa quantidade de fluxo seja menor do que a capacidade de carga da frota desse tipo de vagão. Essa capacidade de carga pode ser calculada multiplicando-se a frota de vagões desse tipo pela capacidade de carga de cada vagão. De modo análogo, podemos encontrar as restrições de frota para todos os tipos de vagões. Existe a possibilidade de considerarmos a diferença no volume de cada produto que é levada por um tipo de vagão, se atribuirmos coeficientes relativos a cada um dos produtos.

O exemplo 3.2 ilustra o equacionamento de limitação de frota de uma malha ferroviária.

Exemplo 3.2 :

Suponha que temos apenas dois tipos de vagões: vagão tipo 1 e vagão tipo 2. Desejamos transportar três produtos pela rede: petróleo, milho e arroz. Petróleo é levado pelo vagão tipo 1, enquanto que milho e arroz são levados pelo vagão tipo 2. A figura (3.3)

ilustra um malha ferroviária.

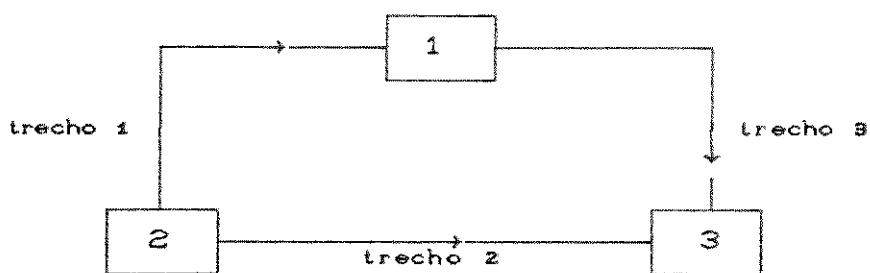


fig. 3.3 - malha ferroviária

O quadro 1 mostra dados a respeito dos produtos: origem do produto, destino do produto e tipo de vagão utilizado pelo produto. O quadro 2 mostra dados a respeito dos vagões: frota e capacidade de carga do vagão.

produto	origem	destino	tipo de vagão
petróleo	1	3	1
milho	1	3	2
arroz	2	1	2

quadro 1

tipo de vagão	frota (unidades)	capac. do vagão (toneladas)	frota x capac. do vagão (toneladas)
1	20	45	900
2	25	50	1250

quadro 2

Como assinalado acima, os trechos que deixam a estação origem de

cada produto são marcados e somados com os demais trechos marcados dos outros produtos que também são levados por aquele tipo de vagão.

A figura (3.4) esboça três malhas ferroviárias, uma para cada um dos produtos, com os seus respectivos arcos marcados.

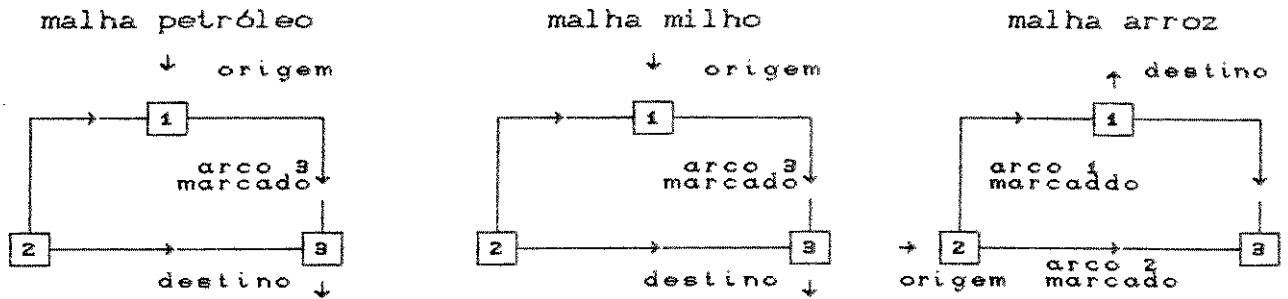


fig. 3.4 - arcos marcados

As equações de limitação de frota são:

$$\begin{aligned}
 0x_1^1 + 0x_2^1 + 1x_3^1 &\leq 900 \quad (\text{vagão 1}) \\
 0x_1^2 + 0x_2^2 + 1x_3^2 + 1x_1^3 + 1x_2^3 + 0x_3^3 &\leq 1250 \quad (\text{vagão 2})
 \end{aligned}$$

O sistema de equações acima pode ser colocado na seguinte forma matricial:

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1^1 \\ x_2^1 \\ x_3^1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1^2 \\ x_2^2 \\ x_3^2 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1^3 \\ x_2^3 \\ x_3^3 \end{bmatrix} \leq \begin{bmatrix} 900 \\ 1250 \end{bmatrix}$$

Colocando numa forma mais compacta:

$$F^1 \cdot x^1 + F^2 \cdot x^2 + F^3 \cdot x^3 \leq d$$

onde:

F^k : matriz dos trechos marcados do produto k ,
 $k=1, 2, 3$

d : vetor coluna do limitante de carga de cada uma das frotas de tipos de vagões

A forma geral para as equações de limitação de frota pode ser obtida da seguinte maneira:

$$\sum_{k=1}^{\mathcal{K}} F^k x^k \leq d$$

$$x^k \geq 0, \quad k=1, \dots, \mathcal{K}$$

Quando inserimos as variáveis de folga t_i , $i = 1, \dots, \mathcal{K}$, onde \mathcal{K} é o número de tipos de vagões, ao sistema acima convertemos todas as desigualdades em igualdades. Representamos estas variáveis de folga folga pelo vetor coluna t , $t = (t_1, \dots, t_{\mathcal{K}})^t$, onde $t \geq 0$. Após a adição destas variáveis de folga temos:

$$\sum_{k=1}^{\mathcal{K}} F^k x^k + t = d$$

$$x^k \geq 0, \quad k=1, \dots, \mathcal{K} \quad \text{e} \quad t \geq 0$$

II . 4. Modelo Matemático do Problema

Definida a função objetivo e todas as suas equações de restrição

II . 5. Comentários

Nos exemplos mostrados nesse capítulo, os trechos de linha estão orientados em apenas um sentido, ou seja, estão permitindo a passagem de fluxo em apenas um sentido do trecho. Na prática, os trechos de linha são bidirecionais, isto é, permitem o transporte de fluxo nos dois sentidos. Quando formos resolver problemas práticos, devemos levar isso em consideração.

A formulação final do nosso problema tem três características interessantes a considerar:

(a) a matriz de restrições apresenta uma parte bloco angular (equações de balanceamento) e uma parcela retangular (restrições de frota e de capacidade);

(b) a matriz de restrições é bastante esparsa;

(c) a matriz de restrições cresce rapidamente com a dimensão da rede e do número de produtos.

Estas três características sugerem que sejam utilizados métodos de decomposição para resolução deste problema, procurando explorar bem a característica (b) acima citada.

Além disso, como o problema é formulado sobre uma rede ferroviária, com trechos e estações, é conveniente utilizarmos a teoria dos grafos. Estas duas sugestões podem ser encontradas na literatura ([Kennington, 1980], [Assad, 1978], etc.), que em particular, recomendam os seguintes três métodos:

- primal particionado;

- "price-directive";
- "resource-directive".

CAPÍTULO III

Método de Resolução

III. 1. Apresentação do Método

III. 1. 1. Equacionamento

A forma que [Kennington, 1980] e outros equacionam matematicamente um problema de multifluxos numa rede sujeita a restrições de capacidade nos arcos é a seguinte:

$$\begin{array}{ll} \text{função objetivo} & \min \sum_{k=1}^{\mathcal{K}} \sum_{i=1}^{\mathcal{I}^k} c_i^k x_i^k \\ \text{sa} & A x^k = r^k \\ & \sum_{k=1}^{\mathcal{K}} D^k x^k \leq b \\ & 0 \leq x^k \leq u^k, \quad k=1, \dots, \mathcal{K} \end{array}$$

onde:

- \mathcal{K} : número de produtos
- \mathcal{S} : número de arcos da rede
- c_i^k : custo de transporte de uma unidade do produto k no arco i
- x^k : vetor dos fluxos do produto k pela rede
- A : matriz nó arco-incidência da rede
- D^k : matriz diagonal do k -ésimo produto
- b : vetor capacidade mútua dos arcos
- r^k : vetor requerimento do produto k
- u^k : vetor capacidade do produto nos arcos da rede

Esta formulação é semelhante à do nosso problema. Portanto, inicialmente analisamos o método primal-particionado aplicado a este problema e no item III.2 fazemos a adaptação para o nosso problema.

Para obtermos apenas restrições de igualdade, inserimos variáveis de folga nas restrições de capacidade mútua dos arcos.

A fim de trabalhar com as matrizes de nó-arco incidência dos produtos com rank completo, inserimos um arco artificial em cada matriz. Esse arco é chamado de arco raiz da matriz.

Após executar essas alterações, nossas equações ficam:

$$\text{função objetivo} \quad \min \sum_{k=1}^{\mathcal{K}} \sum_{i=1}^{\mathcal{S}} c_i^k x_i^k$$

$$\text{sa} \quad A x^k + \alpha e^l = r^k \quad (3.1)$$

$$\sum_{k=1}^{\mathcal{K}} D^k x^k + s = b \quad (3.2)$$

$$0 \leq x^k \leq u^k, \quad s \geq 0, \quad 0 \leq \alpha_k \leq 0, \quad k=1, \dots, \mathcal{K}$$

onde:

s : vetor das variáveis de folga das restrições de capacidade mútua dos arcos

e^l : arco raiz, vetor coluna com 1 na l -ésima posição e zero nas demais

α_k : variável associada ao arco raiz do produto k

O valor da variável α_k é mantido entre zeros para que essa não afete o problema.

As restrições (3.1) e (3.2) podem ser colocadas numa forma matricial, como segue:

$$\begin{bmatrix} A & e^l \\ D^1 & 0 \end{bmatrix} \dots \begin{bmatrix} A & e^l \\ D^k & 0 \end{bmatrix} \begin{bmatrix} I \end{bmatrix} \begin{bmatrix} x^1 \\ \alpha_1 \\ \vdots \\ x^k \\ \alpha_k \\ s \end{bmatrix} = \begin{bmatrix} r^1 \\ \vdots \\ r^k \\ b \end{bmatrix}$$

fig.3.1 - Forma bloco angular das restrições

III. 1. 2. Caracterização da Base

Segundo [Lasdon, 1970], qualquer base para um problema com uma estrutura bloco angular, mais especificamente para o problema com a estrutura da fig 3.1, pode ser colocado como na forma da fig.3.2.

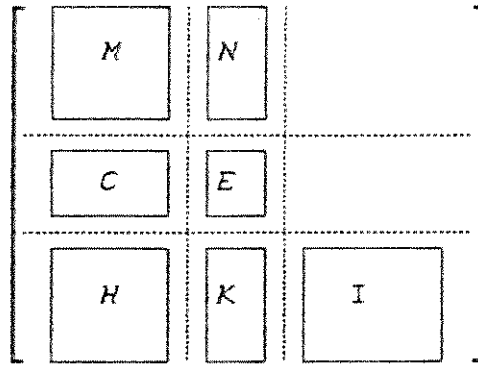


fig. 3.2 - Base para o problema de multifluxo em rede

A matriz M representa uma matriz composta de outras matrizes na sua diagonal. Cada uma dessas matrizes representa uma base para cada um das matrizes nó-arco incidência da matriz restrições. A dimensão de cada uma delas é a mesma, e é dada pelo número de nós da rede.

Uma base para a matriz nó-arco incidência recebe o nome de árvore enraizada. Muitas simplificações decorrem dessas bases serem árvores enraizadas. Denotamos essas árvores enraizadas por ϕ^k , $k=1, \dots, \mathcal{K}$.

A matriz N representa também uma matriz com matrizes em sua diagonal. Essas matrizes não são bases, mas representam caminhos alternativos de transporte dos produtos pela rede. Chamamos esses caminhos alternativos de arcos corretores. Não necessariamente todas as árvores têm arcos corretores. Os arcos corretores aparecem das limitações mútuas dos arcos, quando se deseja levar mais do que a capacidade do arco permite ([Tavares, 1984]).

Denotamos essas matrizes de arcos corretores por ψ^k , $k=1, \dots, \mathcal{K}$.

Reescrevendo a fig.3.2 com as matrizes ϕ^k e ψ^k , obtemos a fig.3.3.

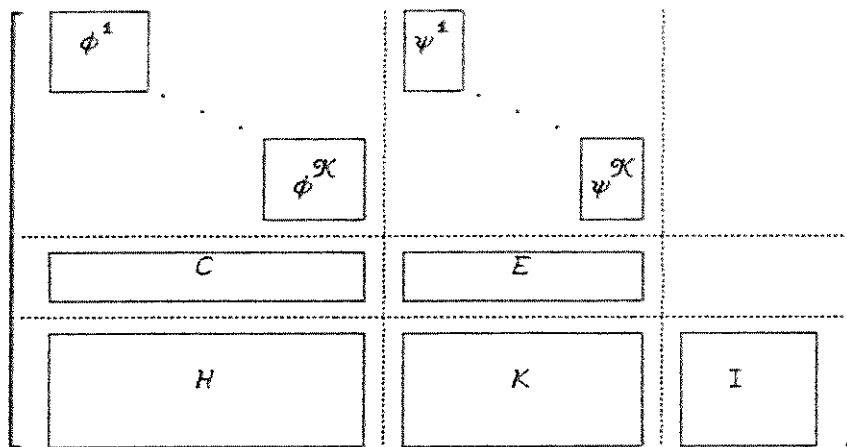


fig.3.3 - Particionamento da base

As matrizes C e E representam as restrições ativas. Restrições ativas representam restrições que estão no seu limite superior e com sua variável de folga fora da base. No caso representam arcos que estão cheios.

As matrizes H e K representam as restrições não ativas. Restrições não ativas são aquelas que têm sua variável de folga na base.

A soma do número de linhas ativas e não ativas é igual ao número de restrições de capacidade dos arcos e de frotas.

O número total de arcos corretores de todas as árvores é igual ao número total de restrições ativas ([Tavares, 1984]).

Particionamos cada uma das matrizes C , E , H e K em novas matrizes, a fig.3.4 ilustra essa nova partição.

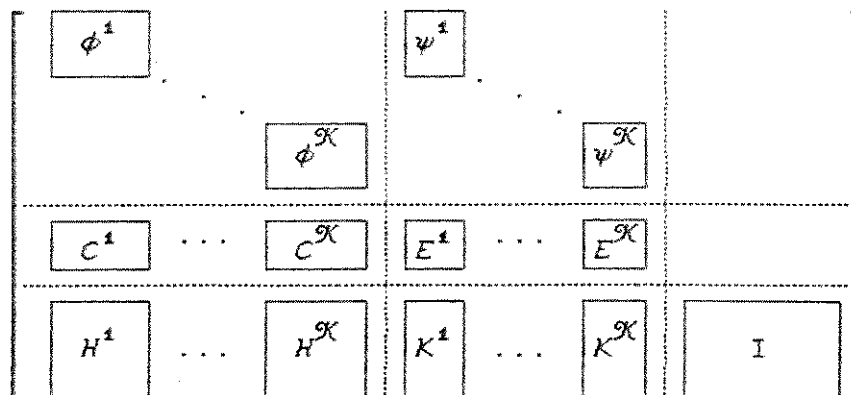


fig 3.4 - Novo particionamento da base

A fig.3.4 ilustra a forma definitiva da base para se aplicar a especialização do método primal simplex.

III. 1. 3. Especialização do Método Primal Simplex

A especialização do método acontece nas etapas do cálculo das variáveis duais e na atualização da coluna da variável a entrar na base.

Explicamos nos dois itens seguintes as modificações que podem ser feitas para explorar a estrutura da base.

III. 1. 3. 1. Cálculo das Variáveis Duais

Para saber se a i -ésima variável não básica é candidata a entrar na base, ela deve satisfazer uma das equações:

$$x_i^N = 0 \quad \text{e} \quad c_i^N - c^B \cdot B^{-1} \cdot N(i) < 0 \quad (3.3)$$

ou

$$x_i^N = u_i^N \quad \text{e} \quad c_i^N - c^B \cdot B^{-1} \cdot N(i) > 0 \quad (3.4)$$

onde x_i^N : fluxo da i -ésima variável não básica

c_i^N : custo da i -ésima variável não básica

c^B : vetor linha do custo das variáveis básicas

B^{-1} : inversa da base

$N(i)$: vetor coluna da i -ésima variável não básica

u_i^N : valor do limitante superior da i -ésima variável não básica

Fazendo $\pi = c^B \cdot B^{-1}$, e pré-multiplicando ambos os lados por B temos:

$$\pi \cdot B = c^B \quad (3.5)$$

onde π : vetor linha das variáveis duais

Substituindo na equação (3.5) a base da *fig.3.5* e particionando os vetores π e c^B nos seguintes vetores $(\pi^1 \mid \pi^2 \mid \pi^3)$ e $(c^1 \mid c^2 \mid 0)$ temos:

$$(\pi^1 \mid \pi^2 \mid \pi^3) \left[\begin{array}{|c|c|c|} \hline M & N & \\ \hline C & E & \\ \hline H & K & I \\ \hline \end{array} \right] = (c^1 \mid c^2 \mid 0) \quad (3.6)$$

Os vetores π e c^B são particionados segundo a estrutura da base.

O vetor c^1 representa o custo dos arcos das árvores enraizadas, o vetor c^2 o custo dos arcos corretores e o vetor nulo é o custo das variáveis de folga.

O vetor π^1 representa as variáveis duais das árvores enraizadas, π^2 as variáveis duais dos arcos corretores e π^3 as variáveis duais das variáveis de folga.

Resolvendo (3.6) encontramos as seguintes equações:

$$\pi^2 = (c^2 - c^1 M^{-1}N)(E - C M^{-1}N)^{-1} \quad (3.7)$$

$$\pi^1 = (c^1 - \pi^2 C) M^{-1} \quad (3.8)$$

$$\pi^3 = 0 \quad (3.9)$$

Fazendo a matriz $(E - C M^{-1}N) = Q$, onde Q recebe o nome de base trabalho ou matriz ciclo.

III. 1. 3. 1. 1. Cálculo da Partição π^2

Substituindo Q na equação (3.7) obtemos uma nova equação para o vetor π^2 .

$$\pi^2 = (c^2 - c^1 M^{-1}N) Q^{-1} \quad (3.10)$$

Os vetores c^1 e c^2 são particionados em $(c^{11} | \dots | c^{1\mathcal{K}})$ e $(c^{21} | \dots | c^{2\mathcal{K}})$. Os vetores c^{1k} e c^{2k} , $k=1, \dots, \mathcal{K}$, representam o custo dos arcos da árvore k e o custo dos arcos corretores da árvore k respectivamente.

Substituindo essas duas partições na equação (3.10) obtemos:

$$\pi^2 = [(c^{21} | \dots | c^{2X}) - (c^{11} | \dots | c^{1X}) M^{-1} N] Q^{-1} \quad (8.11)$$

A matriz M^{-1} tem a propriedade de ser a inversa de suas matrizes diagonais, como mostra a fig.3.5.

$$M^{-1} = \left[\begin{array}{c|c|c} \boxed{\phi^1} & & \\ \hline & \dots & \\ \hline & & \boxed{\phi^X} \end{array} \right]^{-1} = \left[\begin{array}{c|c|c} \boxed{\phi^1}^{-1} & & \\ \hline & \dots & \\ \hline & & \boxed{\phi^X}^{-1} \end{array} \right]$$

fig.3.5 - Matriz M^{-1}

Substituindo as matrizes M^{-1} e N na equação (3.11), e fazendo as devidas multiplicações obtemos a equação (3.12).

$$\pi^2 = [(c^{21} - c^{11}(\phi^1)^{-1}\psi^1 | \dots | c^{2X} - c^{1X}(\phi^X)^{-1}\psi^X)] Q^{-1} \quad (8.12)$$

O cálculo do produto $(\phi^k)^{-1}\psi^k$, $k=1, \dots, X$, é realizado sem precisar ter de calcular a inversa de ϕ^k . Esse cálculo é realizado utilizando-se da árvore que a matriz ϕ^k representa e os arcos corretores, que são as colunas da matriz ψ^k .

A matriz que resulta do produto dessas duas matrizes é simplesmente uma matriz que representa as orientações dos arcos da árvore em relação aos arcos corretores.

O particionamento da equação (3.12) é referenciado da seguinte maneira: $\pi^2 = (\pi^{21} | \dots | \pi^{2m})$.

III. 1. 3. 1. 2. Cálculo da Partição π^1

A equação (3.8) é reescrita de uma forma particionada quando substituimos as partições de c^1 e π^2 .

$$\pi^1 = [(c^{11} | \dots | c^{1m}) - (\pi^{21} | \dots | \pi^{2m}) C] M^{-1}$$

Realizando as devidas multiplicações entre as partições chegamos na seguinte equação para π^1 :

$$\pi^1 = ((c^{11} - \pi^2 C^1) (\phi^1)^{-1} | \dots | (c^{1\mathcal{K}} - \pi^2 C^{\mathcal{K}}) (\phi^{\mathcal{K}})^{-1}) \quad (3.13)$$

O particionamento da equação (3.13) é referenciado da seguinte maneira: $\pi^1 = (\pi^{11} | \dots | \pi^{1\mathcal{K}})$.

Onde cada uma das partições π^{1k} , $k=1, \dots, \mathcal{K}$, equivale ao sistema dado abaixo:

$$\pi^{1k} = (a^{1k} - \pi^2 C^k) (\phi^k)^{-1} \quad (3.14)$$

Pós-multiplicando ambos os lados da equação (3.14) por ϕ^k temos:

$$\pi^{1k} \phi^k = a^{1k} - \pi^2 C^k$$

Chamando o vetor $d^k = a^{1k} - \pi^2 C^k$, temos

$$\pi^{1k} \phi^k = d^k$$

Para se obter o valor de π^{1k} devemos resolver o sistema acima, que pode ser resolvido utilizando-se do fato que a matriz ϕ^k representa uma árvore.

Portanto deve-se calcular as variáveis duais de \mathcal{K} sistemas, para se obter o valor completo do vetor π^1 .

III. 1. 3. 2. Atualização da Coluna que entra na Base

Outra especialização do método é a atualização da coluna que vai entrar na base. A coluna é dividida em três partes e são desenvolvidas fórmulas para se atualizar cada uma das partes de modo mais eficiente, explorando a estrutura da base.

A operação de atualizar uma coluna é pré-multiplicar essa coluna pela inversa da base, ou seja precisamos resolver o sistema abaixo:

$$\underline{p} = B^{-1} p \quad (3.15)$$

onde:

\underline{p} : vetor coluna da variável não básica atualizado

p : vetor coluna da variável não básica a ser atualizado

Pré-multiplicando a equação (3.15) em ambos os lados por B , temos:

$$B \underline{p} = p \quad (3.16)$$

Particionamos os vetores \underline{p} e p da seguinte forma:

$$\underline{p} = (\underline{p}^1 \mid \underline{p}^2 \mid \underline{p}^3)$$
$$p = (p^1 \mid p^2 \mid p^3)$$

Substituindo a base da *fig.3.2* e os vetores particionados na equação (3.16), temos:

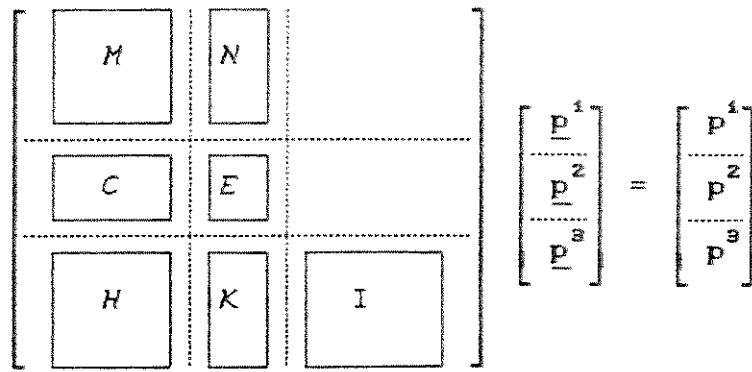


fig.3.6 - Atualização da coluna p

O particionamento é feito de modo adequado ao particionamento da base.

Resolvendo o sistema da fig.3.6 chegamos as seguintes equações para os vetores \underline{p}^1 , \underline{p}^2 e \underline{p}^3 .

$$\underline{p}^2 = Q^{-1} \cdot (\underline{p}^2 - C M^{-1} \underline{p}^1) \quad (3.17)$$

$$\underline{p}^1 = M^{-1} \cdot (\underline{p}^1 - N \underline{p}^2) \quad (3.18)$$

$$\underline{p}^3 = \underline{p}^3 - H \underline{p}^1 - K \underline{p}^2 \quad (3.19)$$

III. 1. 3. 2. 1. Cálculo da Partição \underline{p}^2

O vetor \underline{p}^2 , dado pela equação (3.17), é particionado novamente para obtermos uma forma que explora as características das matrizes M e C.

O vetor \underline{p}^2 e \underline{p}^2 são particionados em $(\underline{p}^{21} | \dots | \underline{p}^{2K})$ e $(\underline{p}^{21} | \dots | \underline{p}^{2K})$ respectivamente.

Do mesmo modo particionamos \underline{p}^1 e \underline{p}^1 em $(\underline{p}^{11} | \dots | \underline{p}^{1K})$ e $(\underline{p}^{11} | \dots | \underline{p}^{1K})$ respectivamente.

Então reescrevendo a equação (3.17), usando os particionamentos:

$$\begin{bmatrix} \underline{p}^{21} \\ \vdots \\ \underline{p}^{2K} \end{bmatrix} = Q^{-1} \left(\begin{bmatrix} \underline{p}^{21} \\ \vdots \\ \underline{p}^{2K} \end{bmatrix} - (C^1 | \dots | C^K) \begin{bmatrix} \phi^1 & & \\ & \ddots & \\ & & \phi^K \end{bmatrix}^{-1} \begin{bmatrix} \underline{p}^{11} \\ \vdots \\ \underline{p}^{1K} \end{bmatrix} \right)$$

Fazendo as operações obtemos a seguinte equação final para o vetor \underline{p}^2 .

$$\begin{bmatrix} \underline{p}^{21} \\ \vdots \\ \underline{p}^{2K} \end{bmatrix} = Q^{-1} \left(\begin{bmatrix} \underline{p}^{21} \\ \vdots \\ \underline{p}^{2K} \end{bmatrix} - \left[C^1 (\phi^1)^{-1} \underline{p}^{11} + \dots + C^K (\phi^K)^{-1} \underline{p}^{1K} \right] \right)$$

III. 1. 3. 2. 2. Cálculo da Partição \underline{p}^1

Escrevendo a equação (3.18) utilizando o particionamento já definido obtemos:

$$\begin{bmatrix} \underline{p}^{11} \\ \vdots \\ \underline{p}^{1K} \end{bmatrix} = \begin{bmatrix} \phi^1 & & \\ & \ddots & \\ & & \phi^K \end{bmatrix}^{-1} \left(\begin{bmatrix} \underline{p}^{11} \\ \vdots \\ \underline{p}^{1K} \end{bmatrix} - \begin{bmatrix} \psi^1 & & \\ & \ddots & \\ & & \psi^K \end{bmatrix} \begin{bmatrix} \underline{p}^{21} \\ \vdots \\ \underline{p}^{2K} \end{bmatrix} \right)$$

Fazendo as multiplicações obtemos a equação final de \underline{p}^1 :

$$\begin{bmatrix} \underline{p}^{11} \\ \vdots \\ \underline{p}^{1K} \end{bmatrix} = \begin{bmatrix} (\phi^1)^{-1} \underline{p}^{11} - (\phi^1)^{-1} \psi^1 \underline{p}^{21} \\ \vdots \\ (\phi^K)^{-1} \underline{p}^{1K} - (\phi^K)^{-1} \psi^K \underline{p}^{2K} \end{bmatrix}$$

III. 1. 3. 2. 3. Cálculo da Partição \underline{p}^3

Reescrevendo a equação (3.19) obtemos:

$$\underline{p}^3 = \underline{p}^3 - \left[\begin{array}{c|c} H^1 & \\ \hline & \dots \\ \hline & H^K \end{array} \right] \begin{bmatrix} \underline{p}^{11} \\ \vdots \\ \underline{p}^{1K} \end{bmatrix} - \left[\begin{array}{c|c} K^1 & \\ \hline & \dots \\ \hline & K^K \end{array} \right] \begin{bmatrix} \underline{p}^{21} \\ \vdots \\ \underline{p}^{2K} \end{bmatrix}$$

Fazendo as multiplicações acima temos:

$$\underline{p}^3 = \underline{p}^3 - H^1 \underline{p}^{11} - \dots - H^K \underline{p}^{1K} - K^1 \underline{p}^{21} - \dots - K^K \underline{p}^{2K}$$

III. 2. Adaptação do Método

Adaptamos o método descrito acima para o nosso modelo. O nosso modelo se encaixa inteiramente ao método sugerido por [Kennington, 1980], para problemas de multifluxo em uma rede.

Os trechos de linha são associados aos arcos, sendo que cada trecho de linha é representado por dois arcos, um arco num sentido e

outro em sentido contrário, pois um trecho de linha pode transportar carga nos dois sentidos.

As estações são associadas aos nós da rede.

Desse modo toda malha ferroviária fica associada a um rede de arcos e nós.

O modelo matemático do nosso problema e o equacionamento de um problema de multifluxo em rede são bem semelhantes.

A função objetivo de ambos os problemas é a mesma, ou seja, minimizar o custo do transporte dos produtos pela rede.

As restrições de balanceamento dos nós continuam iguais, não existem mudanças adicionais.

As restrições de trechos são compatíveis com as restrições da capacidade dos arcos. Trabalhamos com matrizes identidades ao invés de trabalharmos com as matrizes D^k , ou seja, as unidades de transporte são todas iguais, não há necessidade de se trabalhar com coeficientes diferentes de um na matriz D^k .

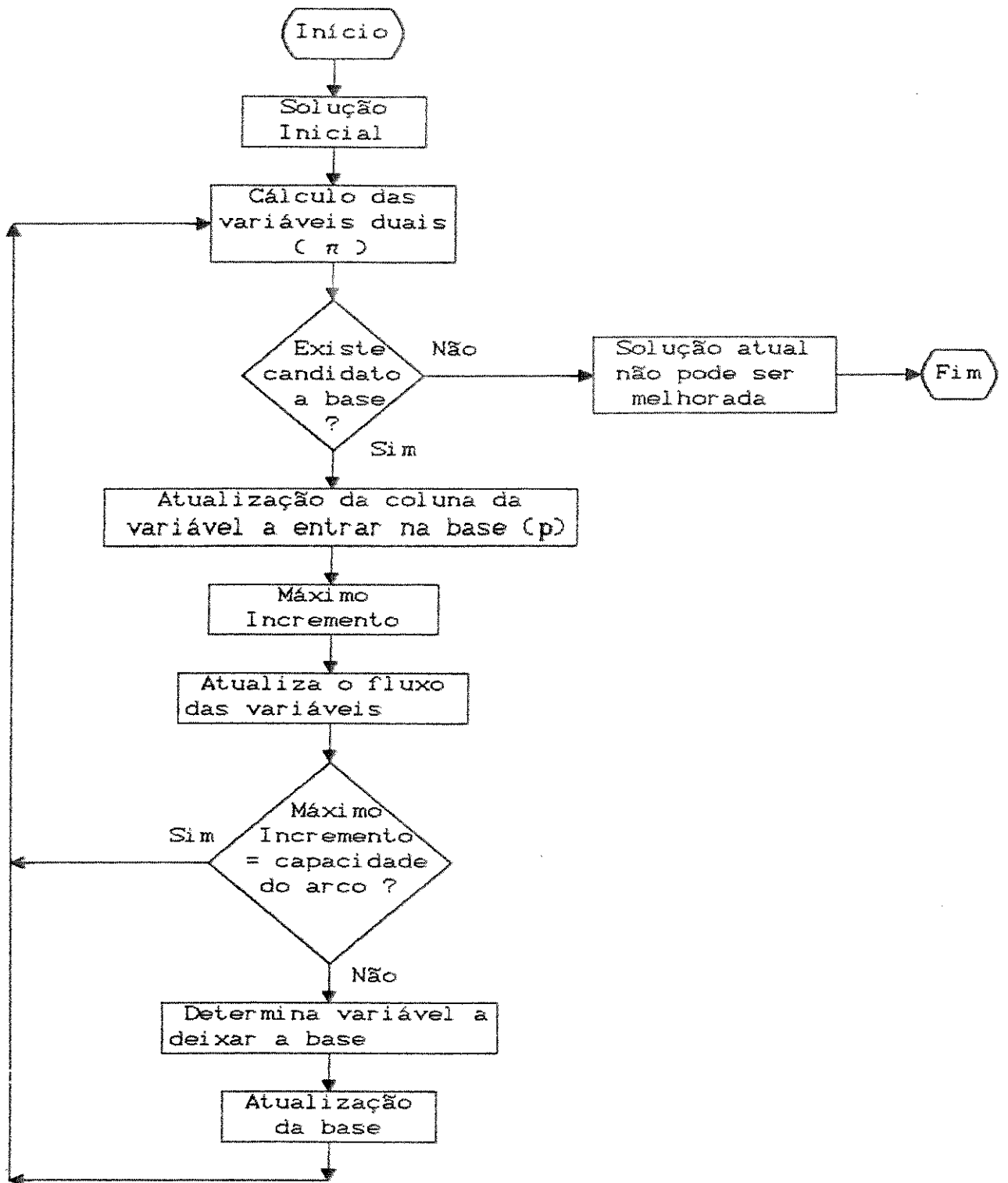
As restrições de frota não aparecem no equacionamento de [Kennington, 1980] e nós devemos inserir esse novo tipo de restrição ao equacionamento.

Basicamente a inserção desse novo tipo de equação não altera o método desenvolvido, apenas o algoritmo que implementa o método é alterado, pois devemos criar uma nova estrutura de identificação dos elementos dessas equações tanto a nível de elementos na base, quanto de elementos fora da base (ver apêndice A).

As fórmulas previamente desenvolvidas continuam válidas apesar do novo tipo de restrição que aparece.

III. 3. Apresentação de Diagrama de Blocos

O diagrama de blocos é basicamente o do método primal simplex revisado [Kennington, 1980]. O *fluxograma 3.1* ilustra as etapas do método.



fluxograma 3.1 - Etapas do Método

CAPÍTULO IV

Algoritmo e Implementação Computacional

IV. 1. Algoritmo

O algoritmo utilizado para resolver o problema está baseado no algoritmo Primal-Simplex-Revisado ([Kennington, 1980]).

As diferenças que ocorrem neste algoritmo, em relação ao algoritmo de um problema com apenas um produto na rede, estão no cálculo das variáveis duais e na atualização da coluna que entra na base.

Utilizamos as fórmulas deduzidas no capítulo III para efetuar essas alterações.

O Algoritmo 4.1 ilustra todos os passos a serem seguidos pelo novo algoritmo.

Algoritmo 4.1 - Especialização do Primal-Simplex-Revisado

Passo 1 Inicialização → **INÍCIO**

Gera-se uma solução básica factível para o problema.

Seja $[x^B | x^N]$ essa solução inicial.

Passo 2 Cálculo das variáveis duais : $\pi = (\pi^1 | \pi^2 | 0)$

Calcular π^2 , utilizar a equação (3.12).

Calcular π^1 , utilizar a equação (3.13).

Passo 3 Verificar se existem candidatos a entrar na base

Seja F_1 e F_2 dois conjuntos como definidos na equação (4.1):

$$\begin{aligned} F_1 &= \left\{ i : x_i^N = 0 \text{ e } (c_i^N - \pi N(i)) < 0 \right\} \\ F_2 &= \left\{ i : x_i^N = u_i^N \text{ e } (c_i^N - \pi N(i)) > 0 \right\} \end{aligned} \quad (4.1)$$

Se $F_1 \cup F_2 = \emptyset$ então termine o algoritmo: solução atual é ótima \rightarrow **FIM**.

Caso contrário, escolha um elemento pertencente ao conjunto $F_1 \cup F_2$, digamos k , para entrar na base.

Se k pertence a F_1 então associe o valor +1 à variável γ , caso contrário associe o valor -1. A variável γ guarda o sinal da variação do fluxo da variável não básica k , +1 indica que o fluxo será aumentado e -1 diminuído.

Passo 4 Atualização da coluna da variável que entra na base: k

Encontrar a coluna $\underline{p} = (\underline{p}^1 | \underline{p}^2 | \underline{p}^3)$:

- obter \underline{p}^2 pela equação (3.17)
- utilizar a equação (3.18) para calcular \underline{p}^1
- calcular \underline{p}^3 pela equação (3.19)

Passo 5 Cálculo da variação máxima do fluxo da variável k

A máxima variação, θ , é o valor mínimo entre as seguintes variáveis θ_1 , θ_2 e θ_3 , onde:

- a variável θ_1 representa a máxima variação das variáveis da base que terão o seu fluxo diminuído.

- a variável θ_2 representa a máxima variação das variáveis da base que terão o seu fluxo aumentado.

- a variável θ_3 representa a máxima variação que pode ser dada na variável k.

As variáveis que terão seu fluxo diminuído são aquelas que têm a sua variável \underline{p} com o mesmo sinal que γ . O máximo valor possível dessa variação é:

$$\theta_1 = \min \left\{ x_i^B / \underline{p}_i \right\}$$

As variáveis que terão seu fluxo aumentado são aquelas que têm a sua variável \underline{p} com sinal contrário a γ . O máximo valor possível dessa variação é:

$$\theta_2 = \min \left\{ (u_i^B - x_i^B) / | \underline{p}_i | \right\}$$

A variável θ_3 recebe o valor do máximo de incremento caso a variável não básica esteja no seu limite inferior (zero), ou recebe o máximo decremento caso a variável esteja no seu limite

superior, u_k^N :

$$e_3 = | u_k^N |$$

O valor de e é o seguinte:

$$e = \min \{ e_1, e_2, e_3 \}$$

Passo 6 Atualização dos fluxos das variáveis básicas e da variável não básica

Os fluxos das variáveis básicas são atualizadas do seguinte modo: para cada elemento i da base fazemos

$$x_i^B = x_i^B - e \gamma p_i$$

A variável não básica k é atualizada do seguinte modo:

$$x_k^N = x_k^N + e \gamma$$

Se o valor de e for igual a u_k^N , então a variável não básica x_k^N não entra na base, volte ao passo 2

Passo 7 Escolha de uma variável para sair da base

Definimos os seguintes conjuntos F_3 e F_4 :

$$F_3 = \left\{ i : x_i^B = 0 \text{ e o sinal de } p_i \text{ é igual ao de } \gamma \right\}$$

$$F_4 = \left\{ i : x_i^B = u_i^B \text{ e o sinal de } p_i \text{ difere ao de } \gamma \right\}$$

Seleciona-se um elemento pertencente ao conjunto $F_3 \cup F_4$ para deixar a base. A coluna desse elemento é substituída pela coluna do elemento que entra na base.

Retorne ao passo 2.

IV. 2. Implementação Computacional

Baseando-se no algoritmo 4.1 implementamos o programa na linguagem Pascal, mais especificamente em Turbo Pascal-Versão 4.0.

Chamamos o nosso programa de Multipro. O programa Multipro é constituído de um programa principal e várias sub-rotinas.

Os dados do problema são criados por programas que geram os devidos arquivos para serem lidos pelo programa principal. Esses arquivos são criados anteriormente ao se executar o programa Multipro.

IV . 2. 1. Estrutura do Programa Multipro

Como já mencionamos o programa Multipro é constituído de um programa principal e de várias sub-rotinas. Descreveremos a seguir o programa principal e suas sub-rotinas.

IV . 2. 1. 1. Programa Principal

Essa parte do programa é responsável pelo gerenciamento de todas as etapas do algoritmo, desde a criação de uma solução inicial

factível até o teste final de otimalidade.

O fluxograma 4.1 representa o programa principal e utiliza as seguintes sub-rotinas:

- Artificial
- Cálculo_do_fluxo_das_variáveis_de_folga
- Encontra_ π^2
- Encontra_ π^1
- Encontra_candidato_a_base
- Encontra_ p^2
- Encontra_ p^1
- Encontra_ p^3
- Cálculo_do_acrécimo_máximo
- Atualiza_fluxo_das_variáveis_na_base
- Variável_a_sair_da_base
- Reestruturação_da_base

A seguir mostramos os fluxogramas dessas dessas sub-rotinas e de outras que são utilizada por elas.

IV . 2. 1. 2. Sub-rotina : Artificial

Essa sub-rotina é responsável por várias tarefas: duplicação da rede, inicialização das variáveis, criação dos arcos artificiais e sobretudo pela geração de uma solução básica factível para o problema.

Os arcos lidos inicialmente são duplicados ou seja, para cada arco existente criamos um novo arco com mesmo custo e mesma capacidade, mas com origem e destino contrários.

Os arcos artificiais são criados do seguinte modo: primeiro criamos um nó artificial, então para cada nó i da rede criamos dois arcos artificiais, um com origem no nó i e destino no nó artificial e outro com origem no nó artificial e destino no nó i . Dessa forma estamos inserindo na rede mais duas vezes o número de nós arcos. A capacidade e o custo associados a esses arcos são bem maiores do que as capacidades e os custos dos arcos reais.

A solução básica e factível que geramos tem a estrutura mostrada na *fig 4.1*.

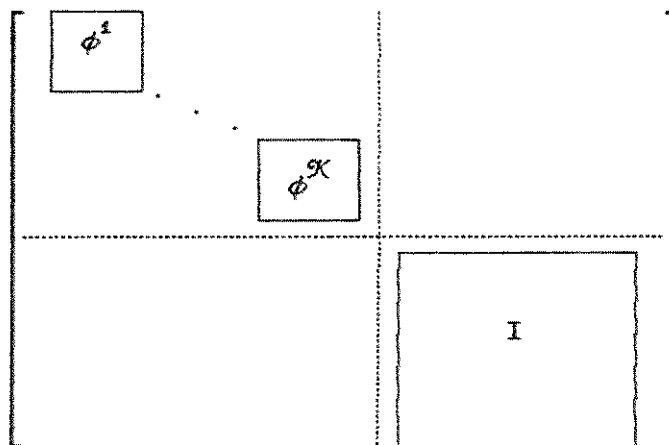


fig. 4.1 - Base inicial

As matrizes ϕ^i , $i=1, \dots, K$, representam árvores enraizadas formadas apenas com arcos artificiais. A matriz I representa as variáveis de folga das restrições. Como as árvores são formadas por apenas arcos artificiais não existe qualquer elemento sob as matrizes ϕ^i .

A forma pela qual geramos as árvores para cada produto é a seguinte: ligamos o nó origem do produto ao nó artificial pelo arco que tem origem no nó origem do produto e destino no nó artificial; ligamos o nó artificial ao nó destino do produto pelo arco que tem origem no nó artificial e destino no nó destino do produto e ligamos todos os demais nós da rede com o nó artificial por meio de arcos com origem nos nós e destino no nó artificial. Associamos ao fluxo dos dois arcos artificiais que ligam o nó origem ao nó destino do produto a quantidade total de carga que se deseja transportar e fluxo zero aos demais arcos artificiais da árvore.

As árvores geradas e a base são guardadas da forma explicada no apêndice A.

O fluxograma 4.2 mostra a seqüência das tarefas executadas pela sub-rotina.

IV . 2. 1. 3. Sub-rotina : Cálculo_do_fluxo_das_variáveis_de_folga

Esta sub-rotina é responsável pela inicialização das variáveis de folga das restrições. Além disso, testa se o fluxo gerado para os arcos das árvores está correto. O fluxograma 4.3 mostra as etapas da sub-rotina.

IV . 2. 1. 4. Sub-rotina : Encontra_ π^2

Calcula as variáveis duais associadas aos arcos corretores dos produtos. Segue basicamente a equação (3.12), mas com as simplificações decorrentes de estarmos trabalhando com grafos. O fluxograma 4.4 mostra as etapas da sub-rotina.

Outras 4 sub-rotinas que são utilizadas para encontrar π^2 são:

IV . 2. 1. 4. 1. Sub-rotina : Inter

Encontra o primeiro nó comum dos caminhos que partem da origem e do destino do produto em direção ao nó raiz. O fluxograma 4.5 examina essas etapas.

IV . 2. 1. 4. 2. Sub-rotina : Árvore_vezes_um_arco $(Q \times (D$

Utilizada para gerar o produto $(\phi^k)^{-1}.s$, onde s é um arco corretor do produto k ou um arco que vai entrar na base. O fluxograma

4.6 mostra como calcular.

IV . 2. 1. 4. 3. Sub-rotina : Verifica_leva

Simplemente verifica se o produto k é levado pelo tipo de vagão i . Fluxograma 4.7.

IV . 2. 1. 4. 4. Sub-rotina : Inverte

Esta sub-rotina inverte a matriz Q (matriz ciclo). Utilizamos um processo de escalonamento para gerar a matriz inversa. Fluxograma 4.8.

IV . 2. 1. 5. Sub-rotina : Encontra_ π^1

Calcula as variáveis duais de \mathcal{K} árvores. Antes de calcular π^1 é necessário ter calculado π^2 . O fluxograma 4.9 dá maiores detalhes de como calcular π^1 .

IV . 2. 1. 6. Sub-rotina : Encontra_candidato_a_base

Essa sub-rotina procura por candidatos à base, escolhe entre todos os arcos e variáveis candidatos a base qual é o que apresenta o custo relativo em módulo maior. Caso não encontre nenhum candidato o programa termina, a solução atual é ótima. O fluxograma 4.10 representa os passos para obter o candidato.

IV . 2. 1. 7. Sub-rotina : Encontra_p²

Atualiza uma parte da coluna p, que é candidata a entrar na base e é referente as restrições ativas da base. O *fluxograma 4.11* detalha o cálculo de p².

IV . 2. 1. 8. Sub-rotina : Encontra_p¹

Atualiza outra parte da coluna p, que é candidata a entrar na base e é referente as árvores dos produtos. O *fluxograma 4.12* mostra como calculamos p¹.

IV . 2. 1. 9. Sub-rotina : Encontra_p³

Atualiza a terceira e última parte da coluna p, que é candidata a entrar na base e é referente as equações não ativas da base. No *fluxograma 4.13* colocamos os passos no cálculo de p³.

IV . 2. 1. 10. Sub-rotina : Cálculo_do_acrécimo_máximo

Calcula o acréscimo máximo no fluxo dos arcos de todas as árvores para determinar o fluxo da variável que quer entrar na base. O *fluxograma 4.14* mostra como determinamos esse acréscimo máximo.

IV . 2. 1. 11. Sub-rotina : Atualiza_os_fluxos_das_
variáveis_na_base

Atualiza o fluxo de todas as variáveis da base com o valor da

máxima variação. O *fluxograma 4.15* mostra como atualizamos o fluxo de todas as variáveis da base.

IV . 2. 1. 12. Sub-rotina : Variável_a_sair_da_base

Procura por uma variável para deixar a base. Quando encontra uma sub-rotina volta para o programa principal. O *fluxograma 4.16* ilustra como é feita essa busca.

IV . 2. 1. 13. Sub-rotina : Reestruturação_da_base

Após identificada a variável que vai sair e a variável que vai entrar na base é preciso fazer as devidas alterações na base. No *fluxograma 4.17* relatamos todos os possíveis casos de alterações de entrada e saída de variáveis e as principais alterações.

Dependendo do caso, mais duas sub-rotinas podem vir a ser chamadas, são elas:

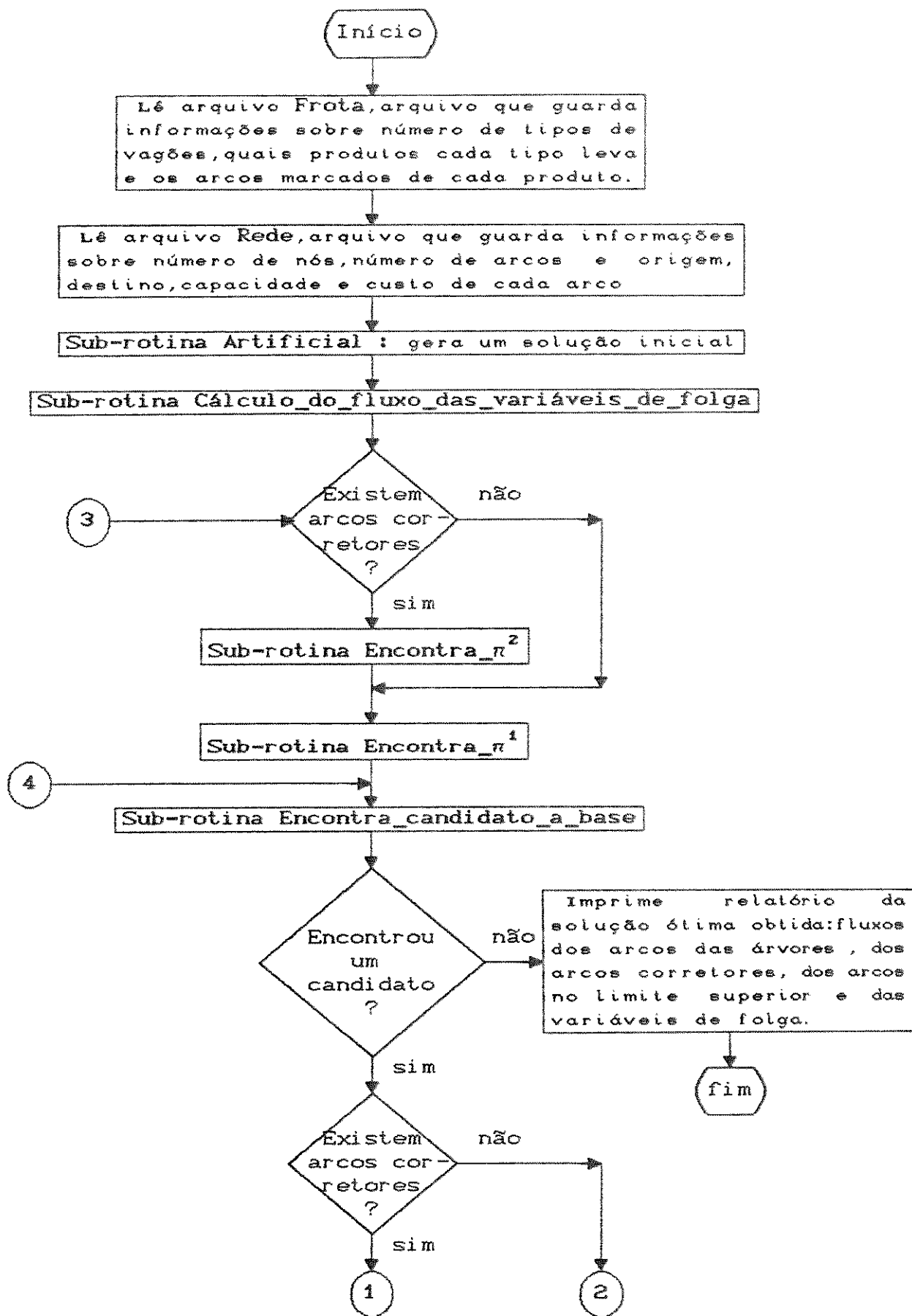
IV . 2. 1. 13. 1. Sub-rotina : Reestrutura_árvore

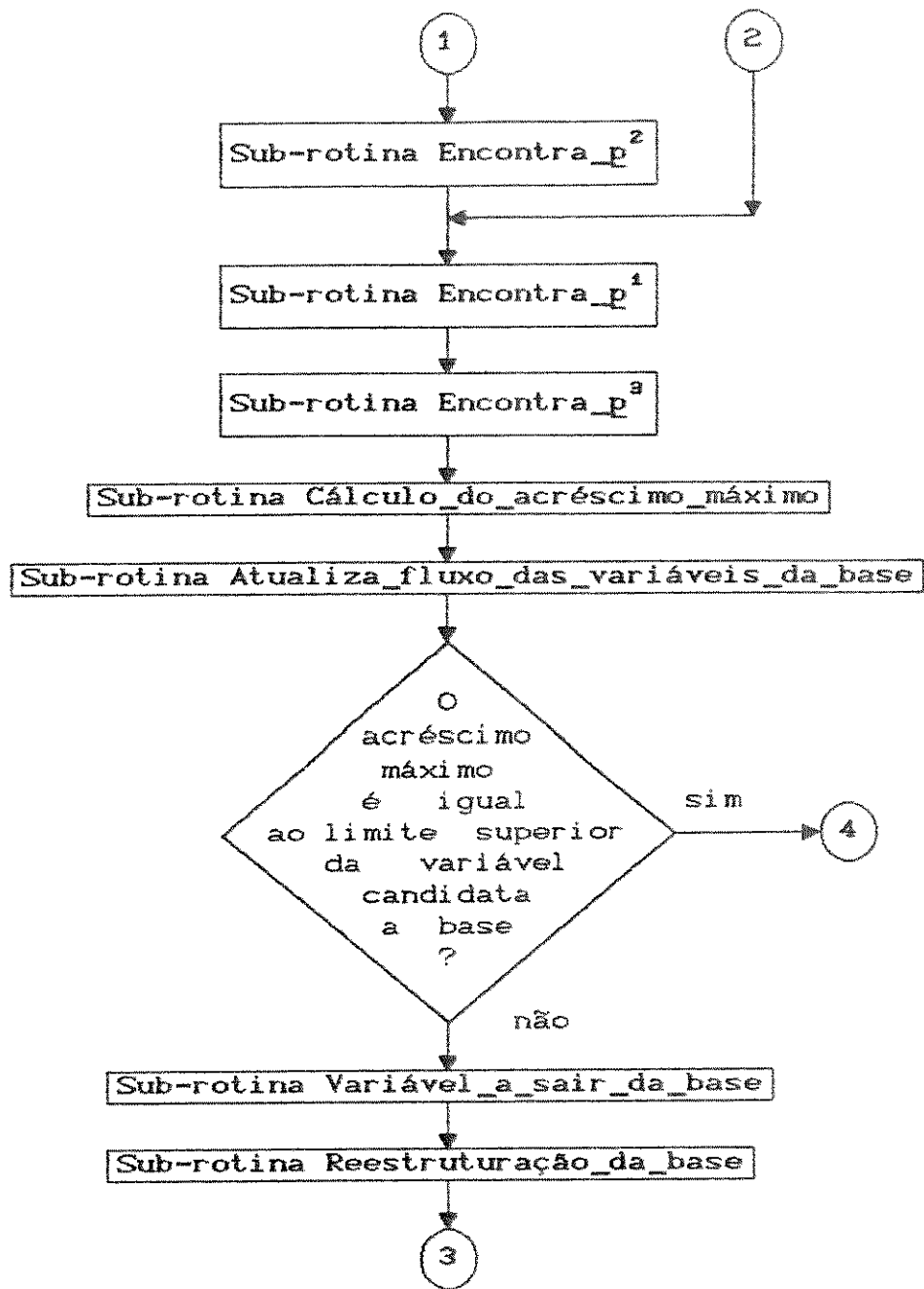
Essa sub-rotina é utilizada para fazer a troca de um arco por outro na árvore. O *fluxograma 4.18* mostra a alteração que é feita.

IV . 2. 1. 13. 2. Sub-rotina : Encontra_arco_corretor

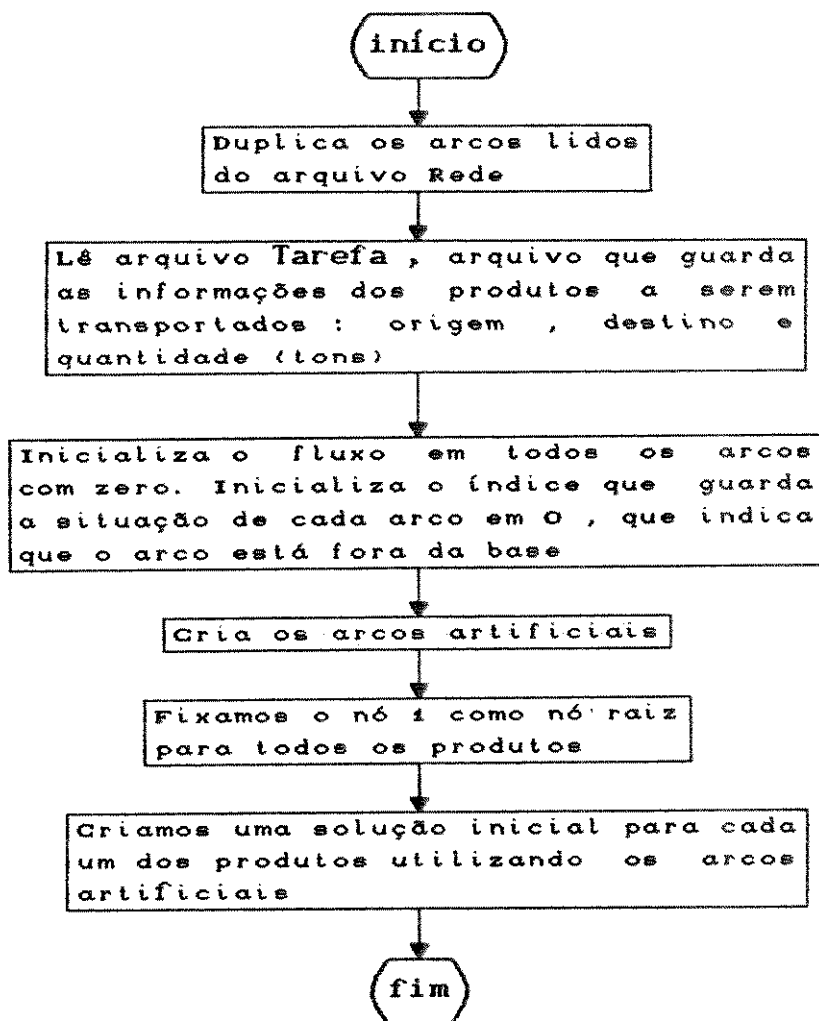
Essa sub-rotina é utilizada quando precisamos encontrar um arco corretor para inseri-lo na árvore no lugar de um arco que sai dela. O *fluxograma 4.19* mostra o procedimento para encontrar esse arco corretor.

fluxograma 4.1 - Programa Principal

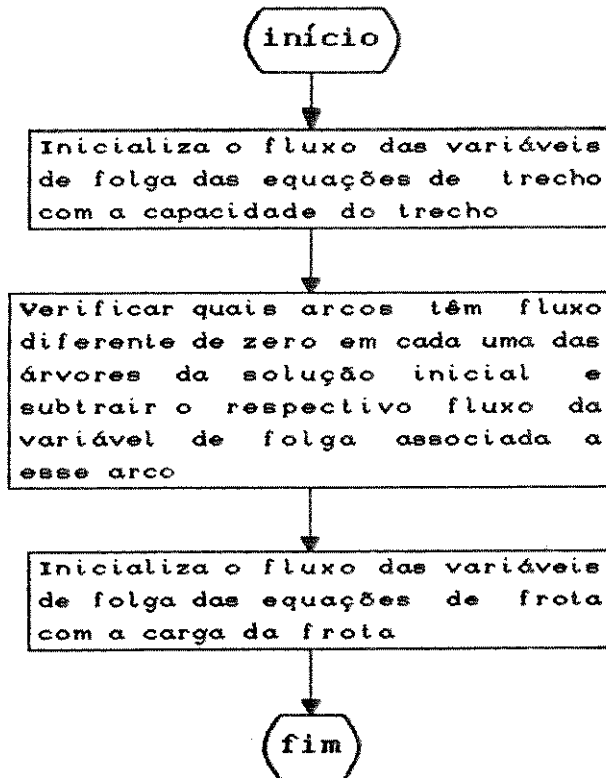




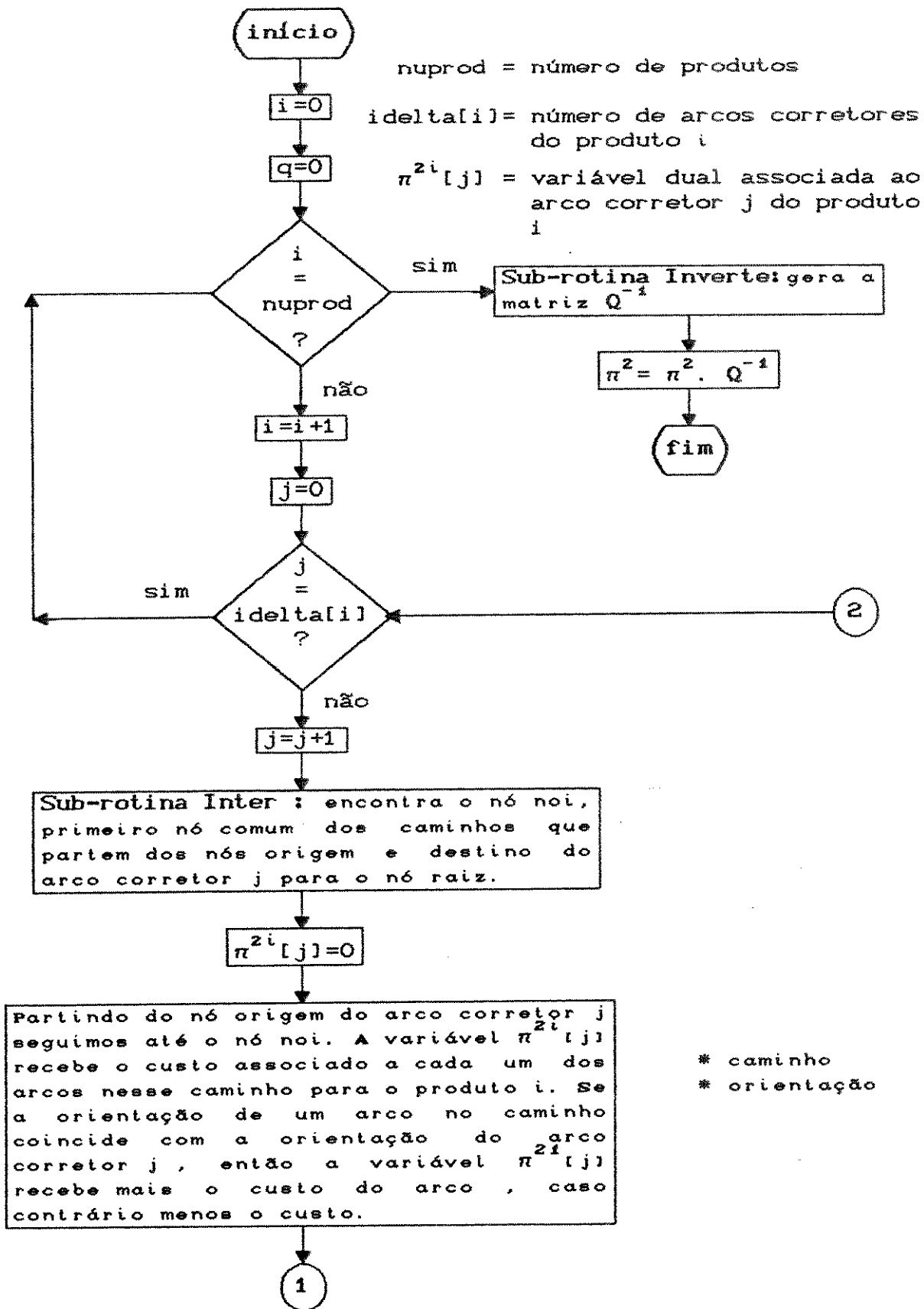
fluxograma 4.2 - Sub-rotina Artificial

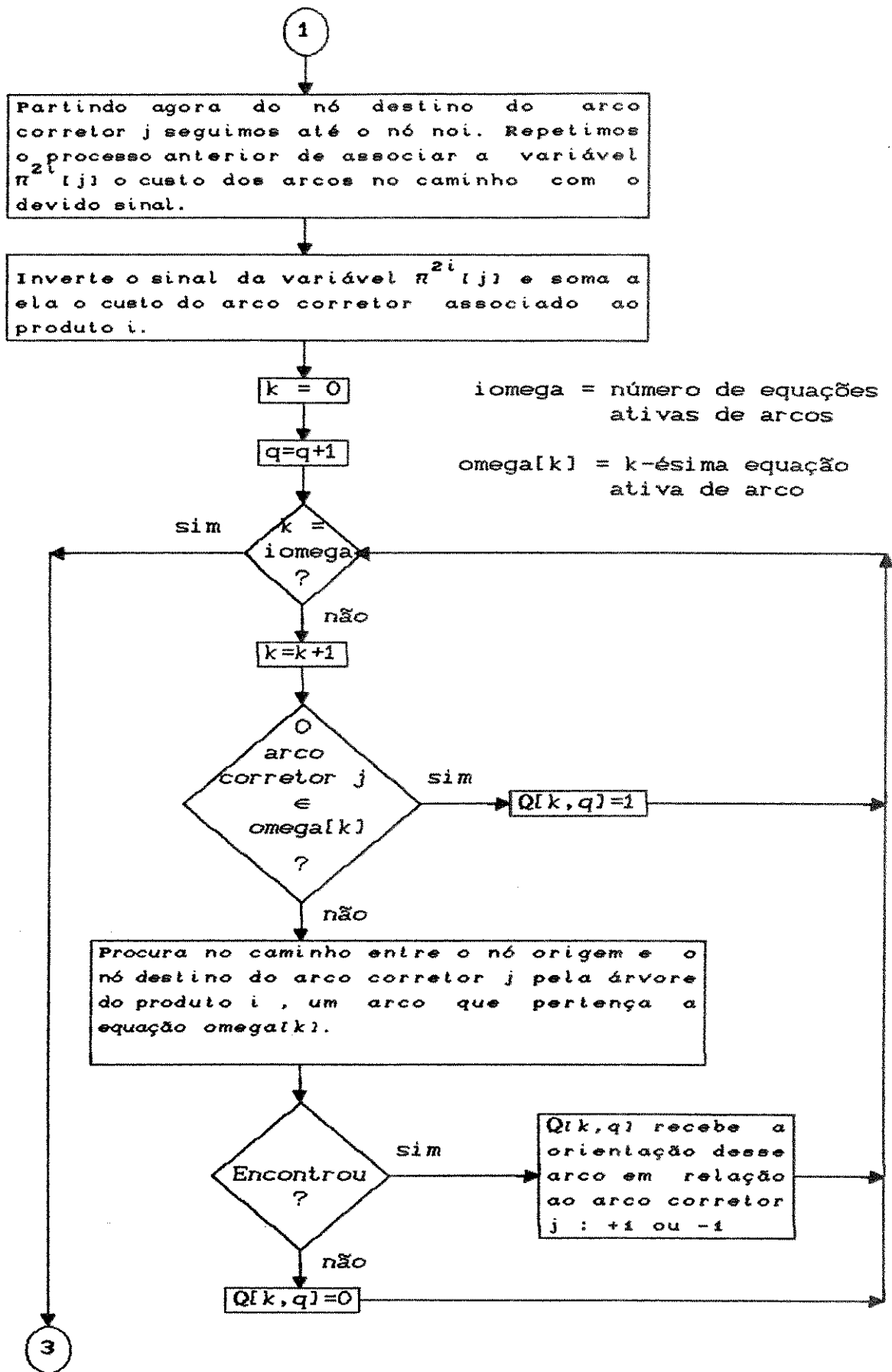


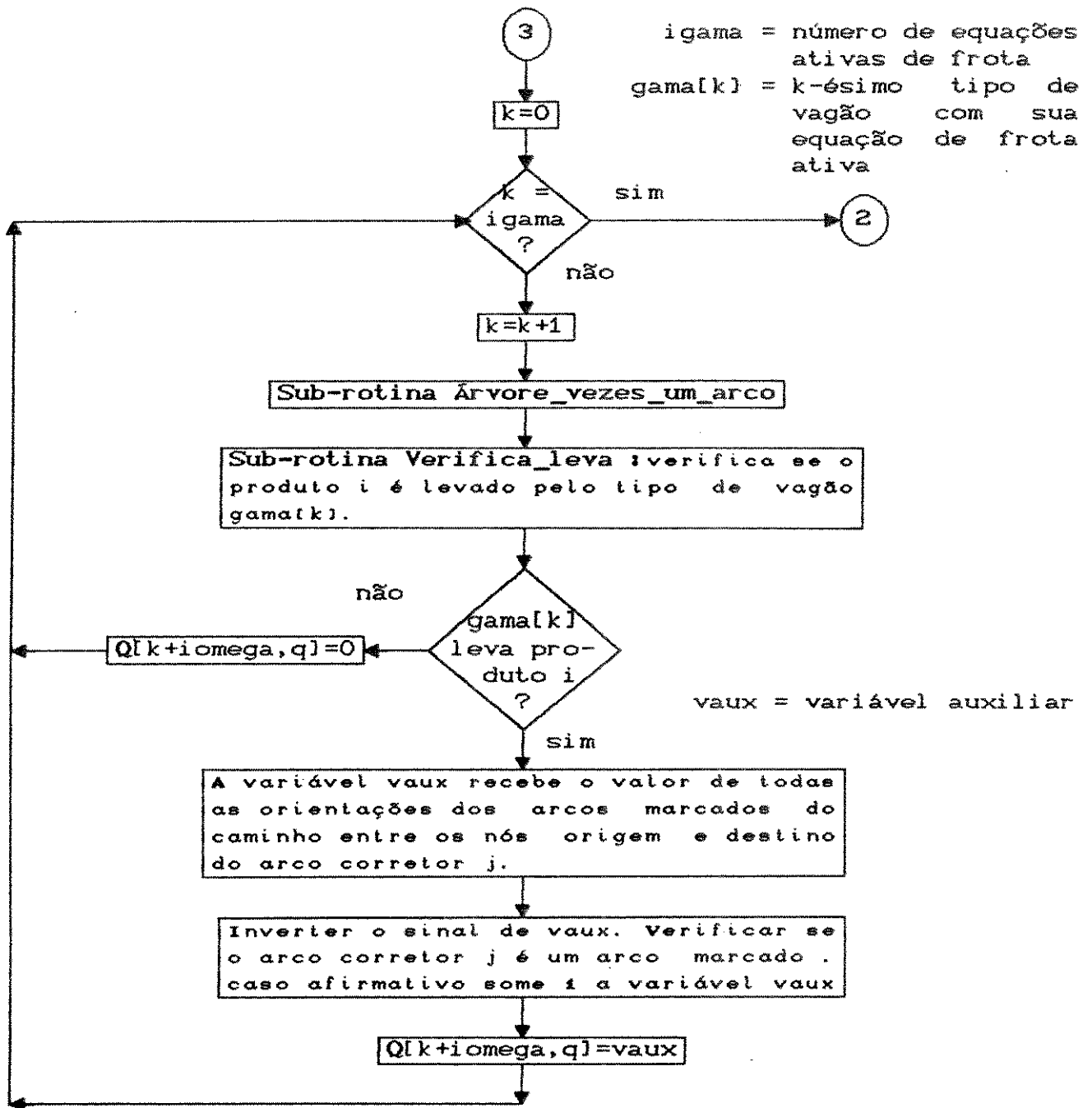
fluxograma 4.3 - Sub-rotina Cálculo_do_fluxo_das_variáveis_de_folga



fluxograma 4.4 - Sub-rotina Encontra_ π^2





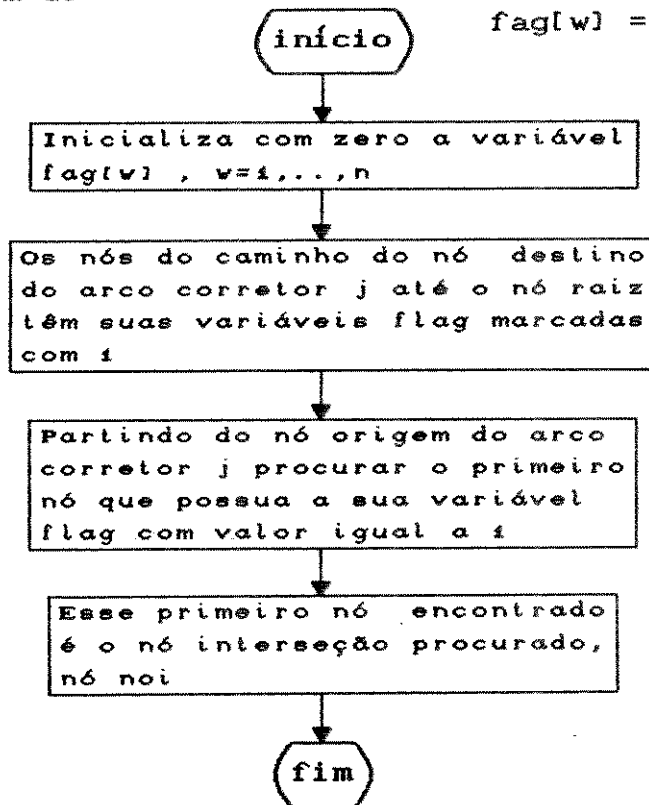


fluxograma 4.5 - Sub-rotina Inter

variáveis de passagem de
parâmetro:

i: produto
j: arco corretor do
produto i

n = número de nós
da árvore i
flag[w] = variável flag
associada ao
w-ésimo nó da
árvore



fluxograma 4.6 - Sub-rotina *Árvore_vezes_um_arco*

variáveis de passagem de parâmetro:

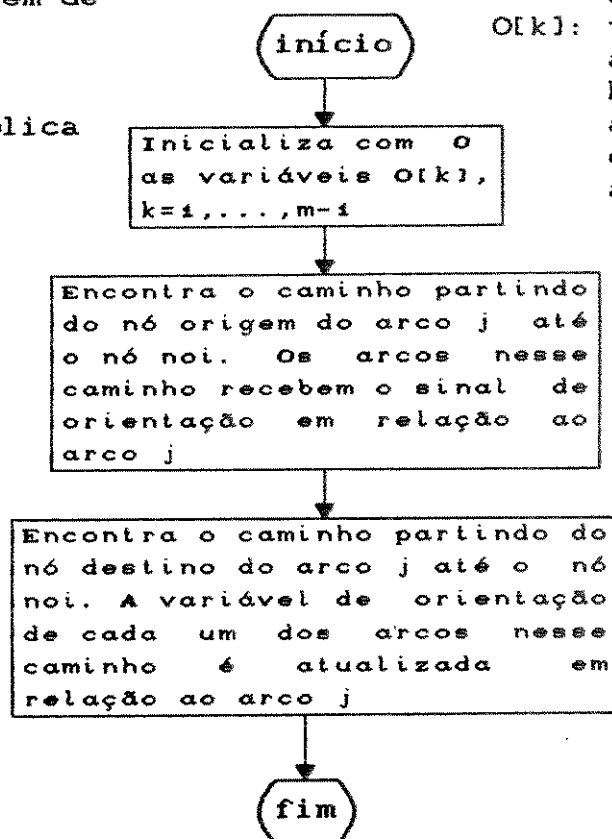
i : produto

j : arco que multiplica a árvore

noi: ver Sub-rotina Inter

m : número de nós da árvore

O[*k*]: variável que guarda a orientação do *k*-ésimo arco da árvore do produto *i* em relação a um arco fora da base



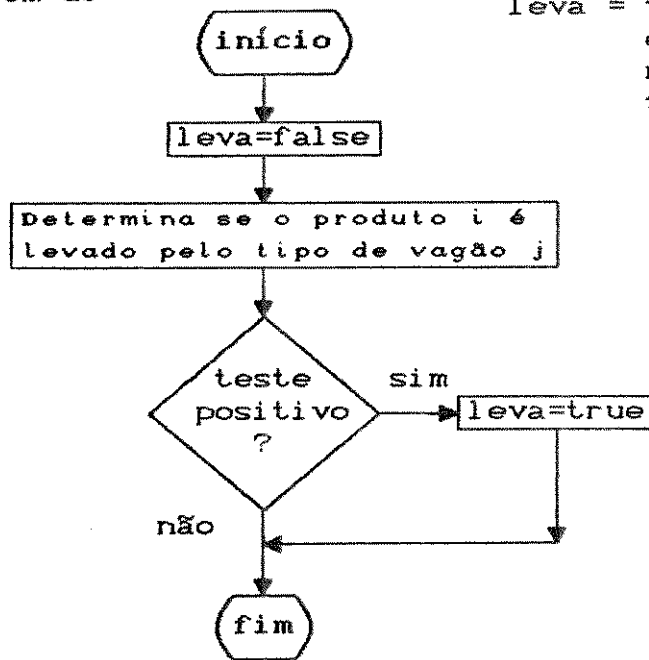
fluxograma 4.7 - Sub-rotina Verifica_leva

variáveis de passagem de parâmetro:

i: produto

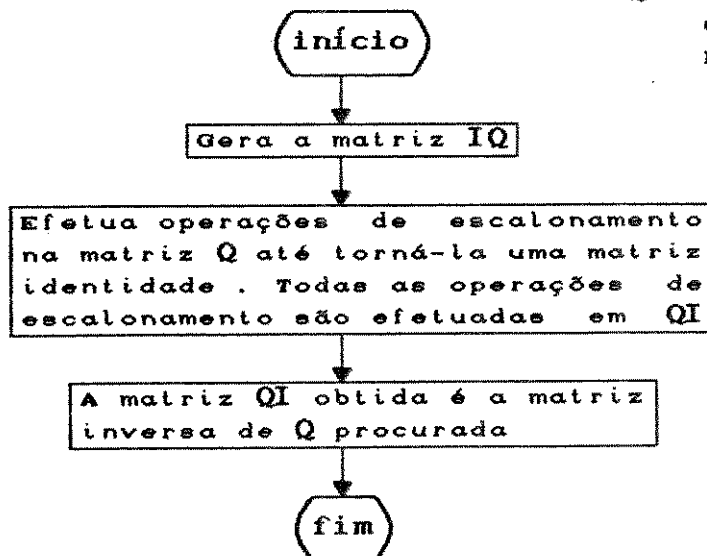
j: tipo de vagão

leva = variável booleana que recebe a resposta do teste, true ou false

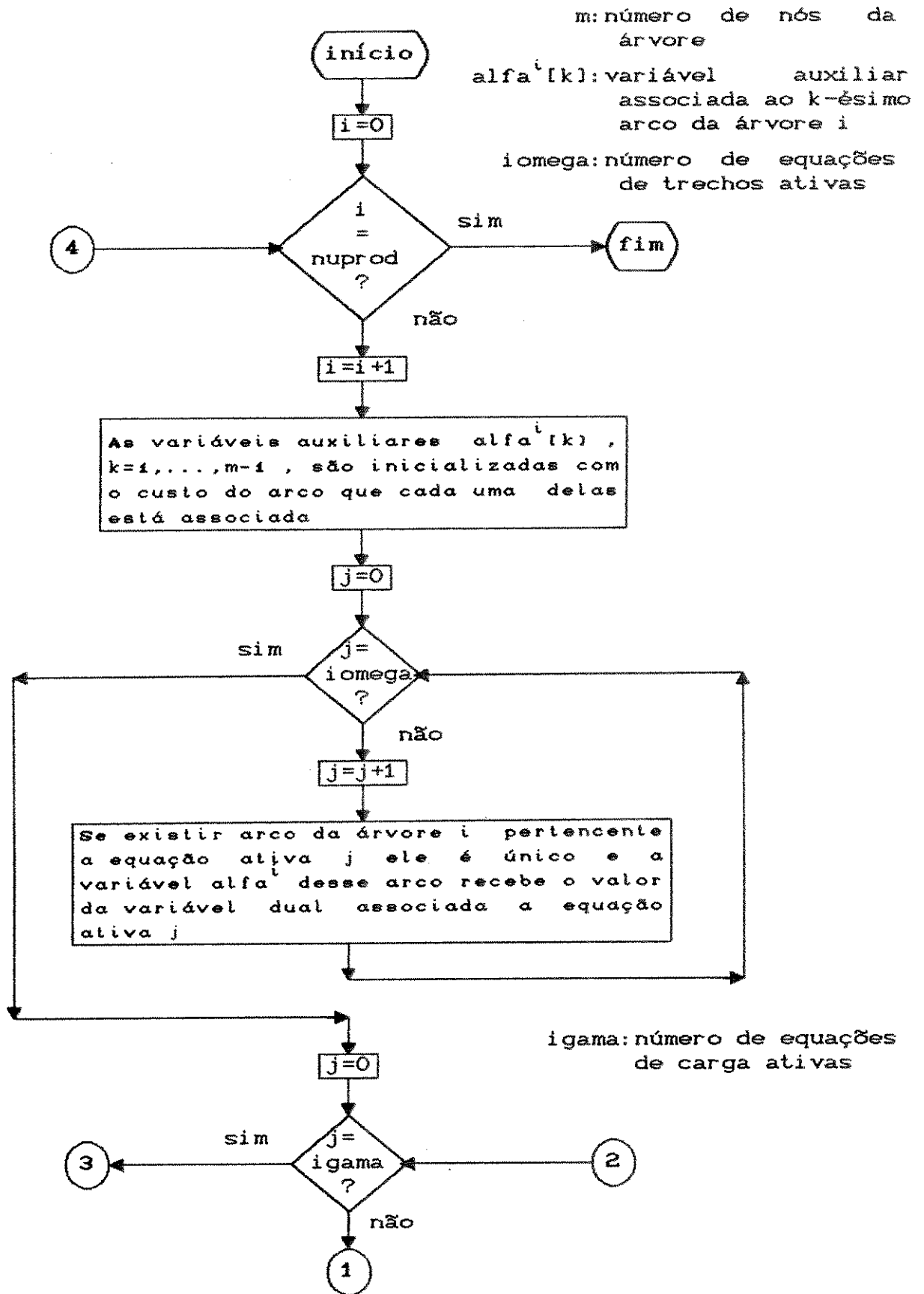


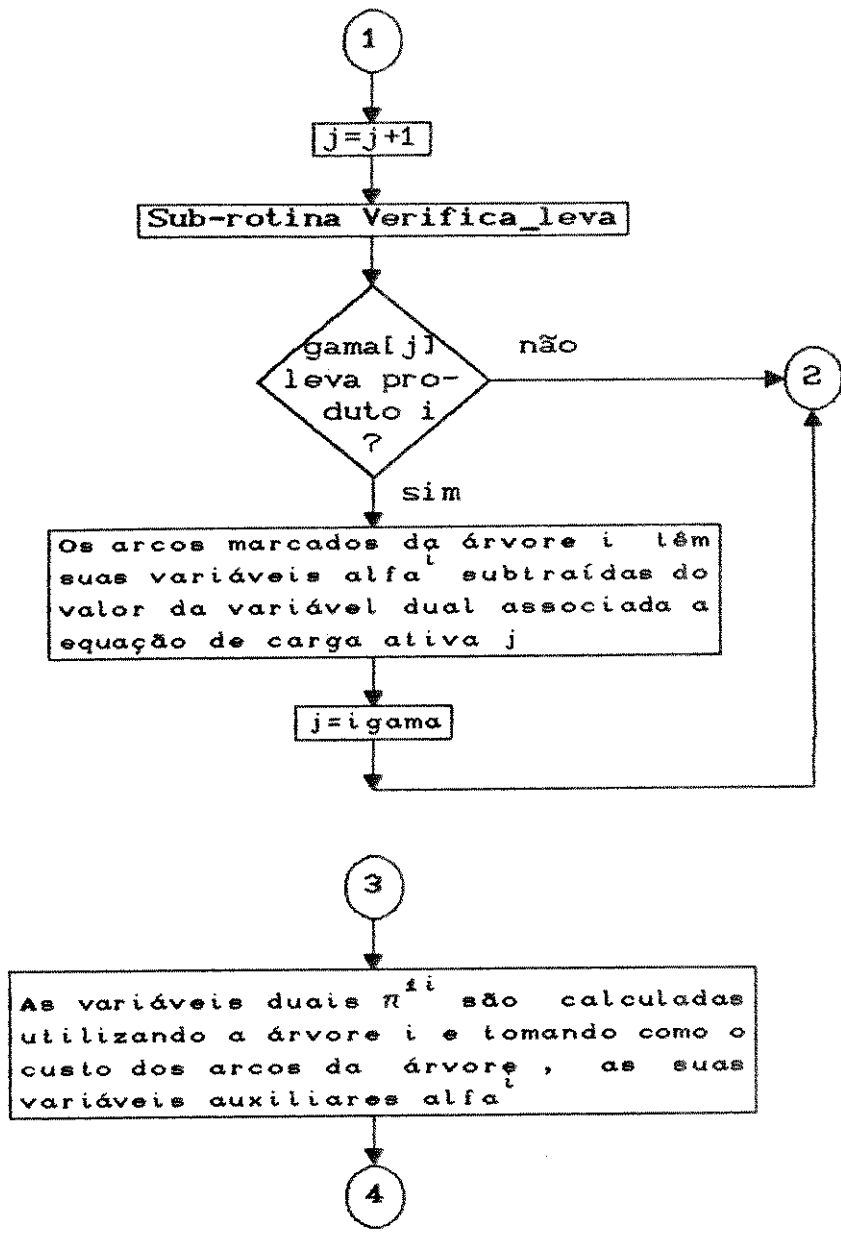
fluxograma 4.8 - Sub-rotina Inverte

IQ = matriz identidade de ordem igual à matriz Q

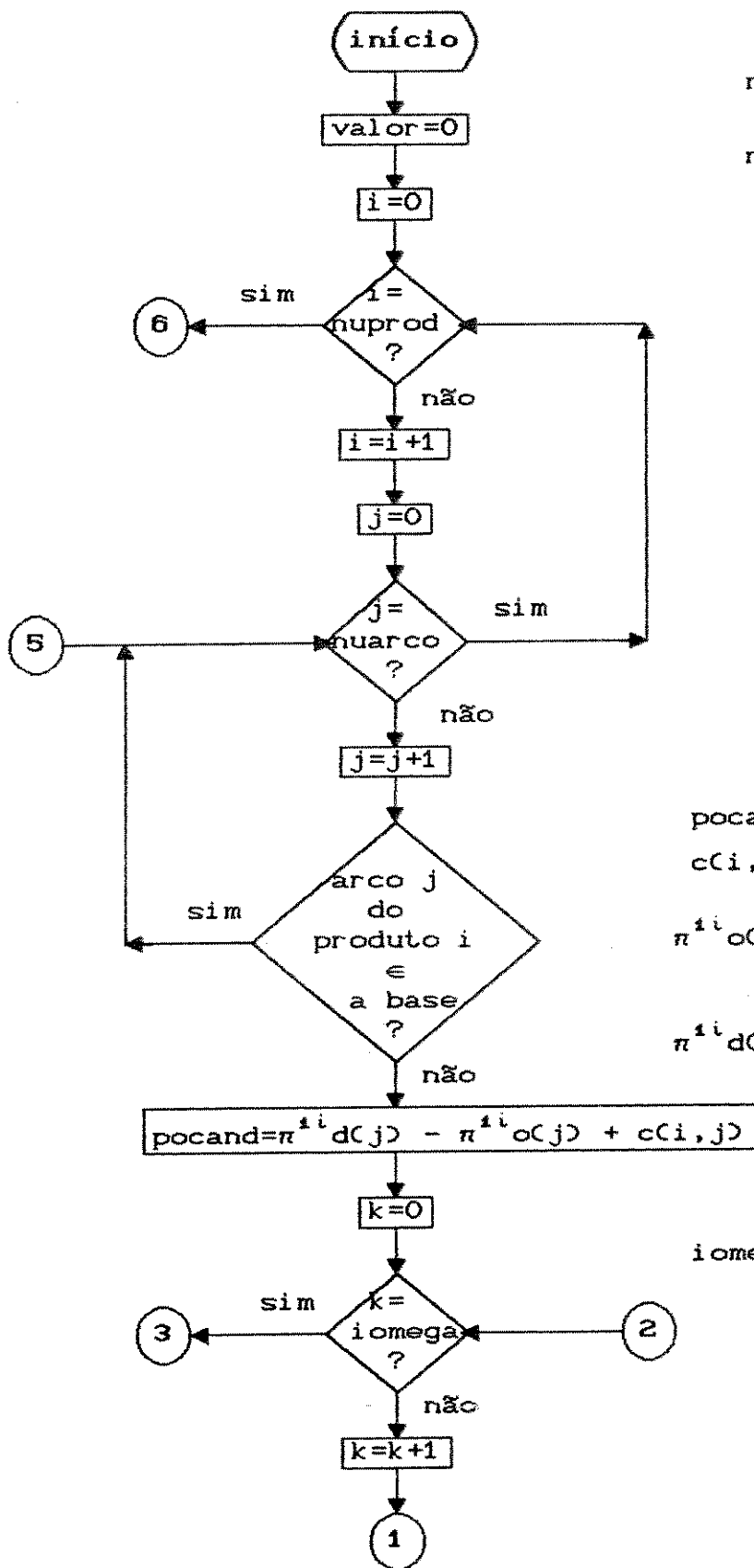


fluxograma 4.9 - Sub-rotina Encontra π^1





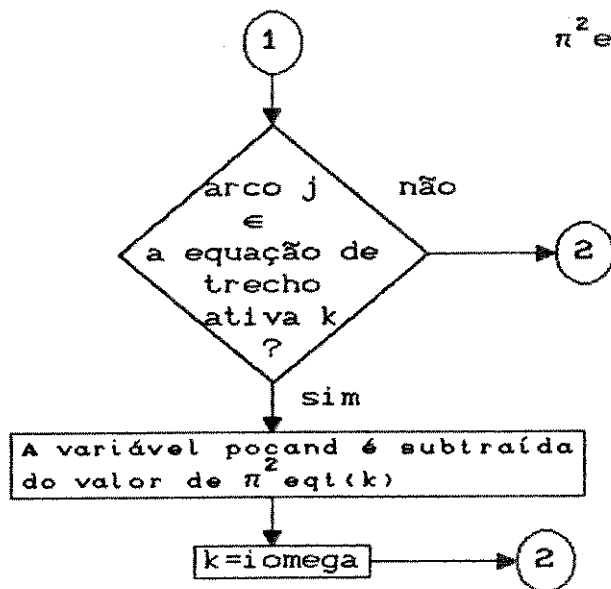
fluxograma 4.10 - Sub-rotina Encontra_candidato_a_base



valor : variável auxiliar
 nuprod : número de produtos transportados
 nuarco : número de arcos da rede

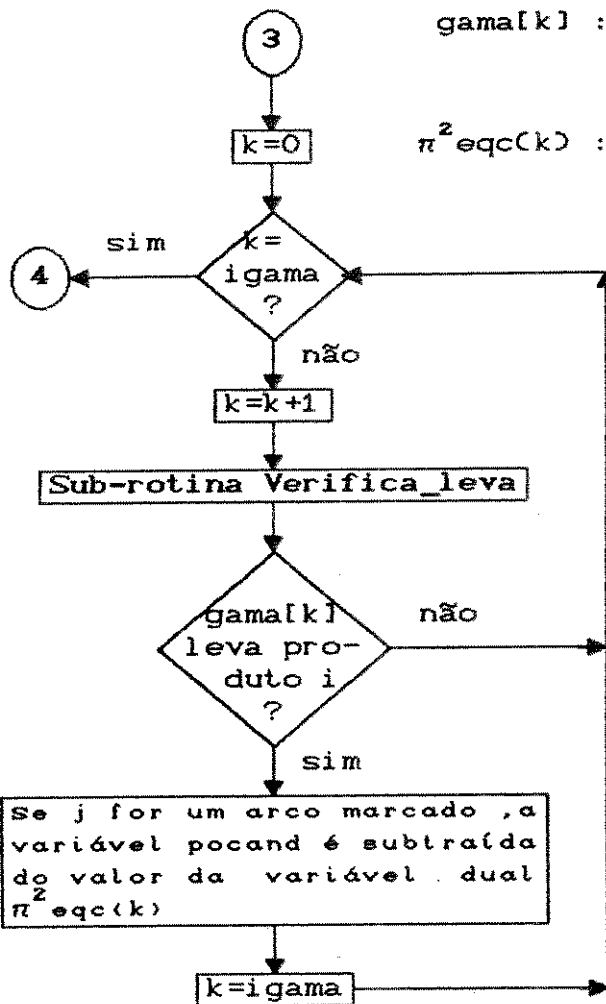
pocand : variável auxiliar
 $c(i,j)$: custo do transporte do produto i no arco j
 $\pi^{i_o}(j)$: variável dual do nó origem do arco j , produto i
 $\pi^{i_d}(j)$: variável dual do nó destino do arco j , produto i

$i\omega$: número de equações de trecho ativas



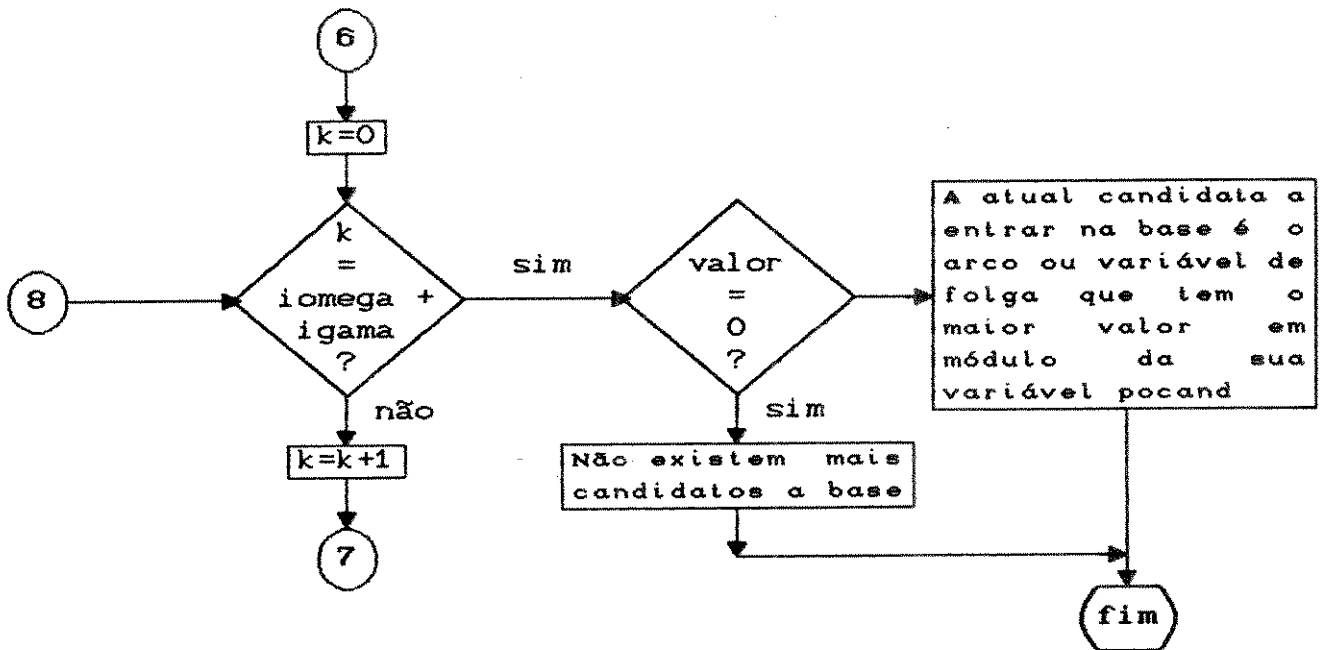
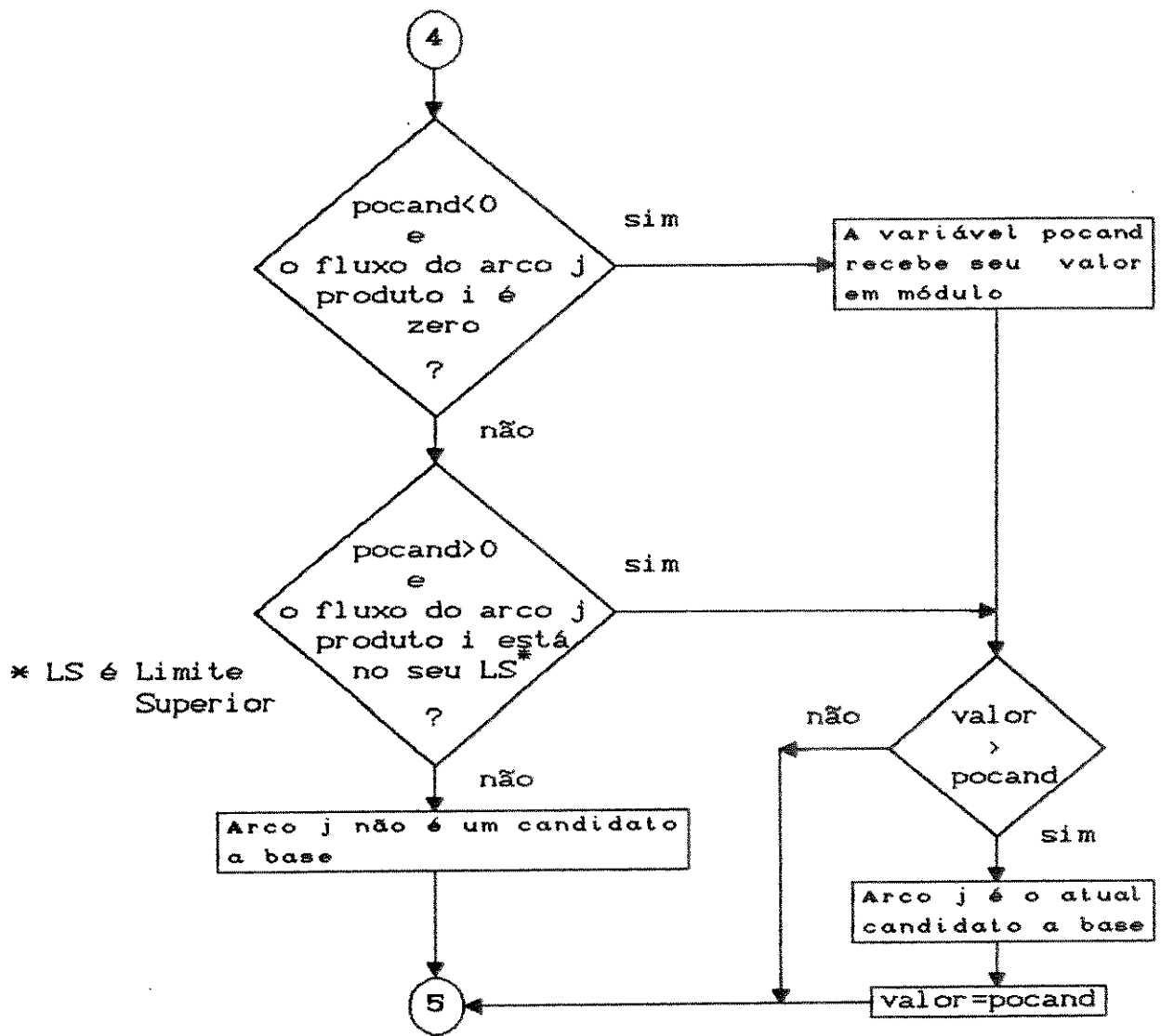
$\pi^2 eqt(k)$: variável dual referente a equação de trecho ativa k

igama : número de equações de carga ativas

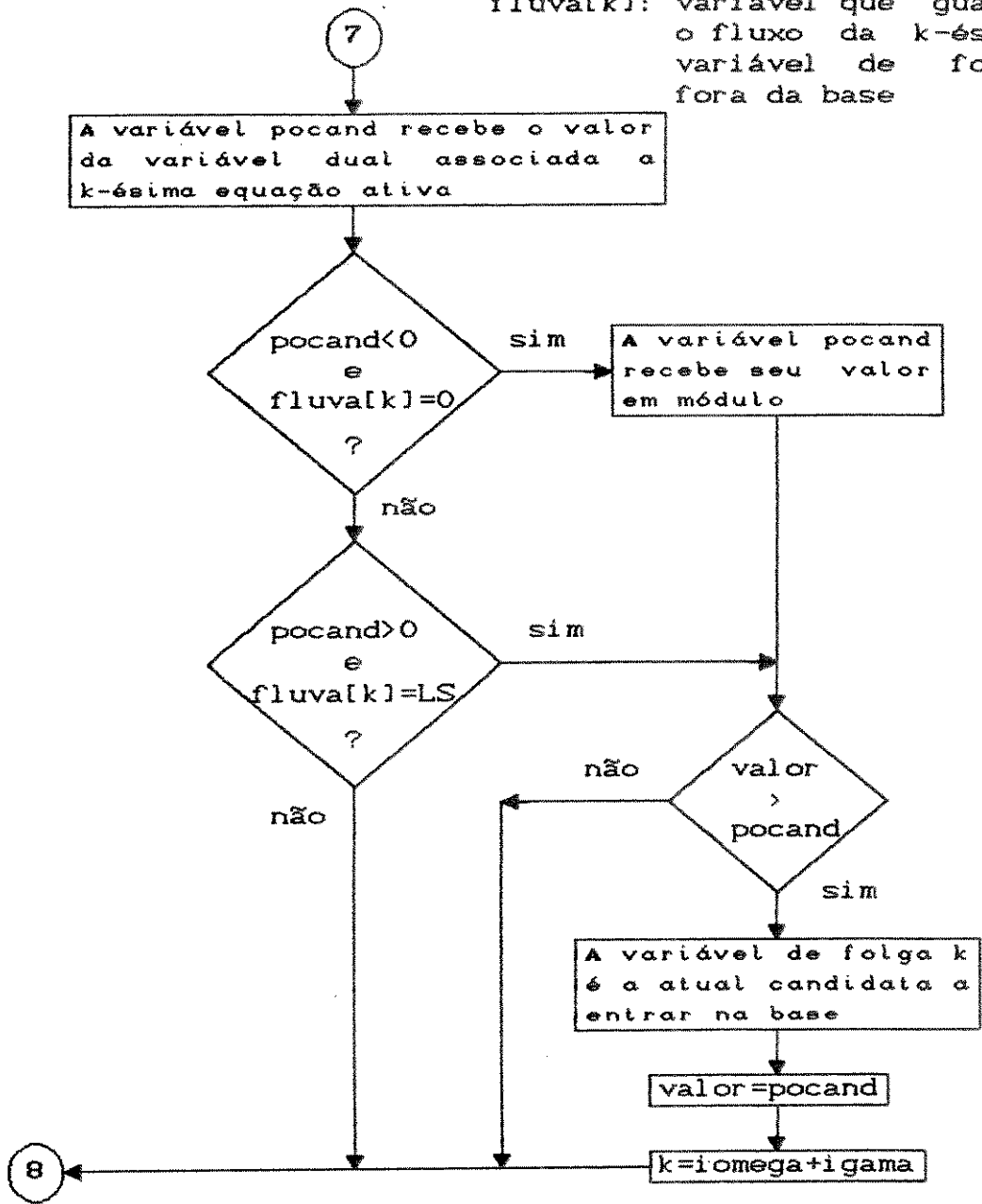


gama[k] : k-ésimo tipo de vagão com sua equação de carga ativa

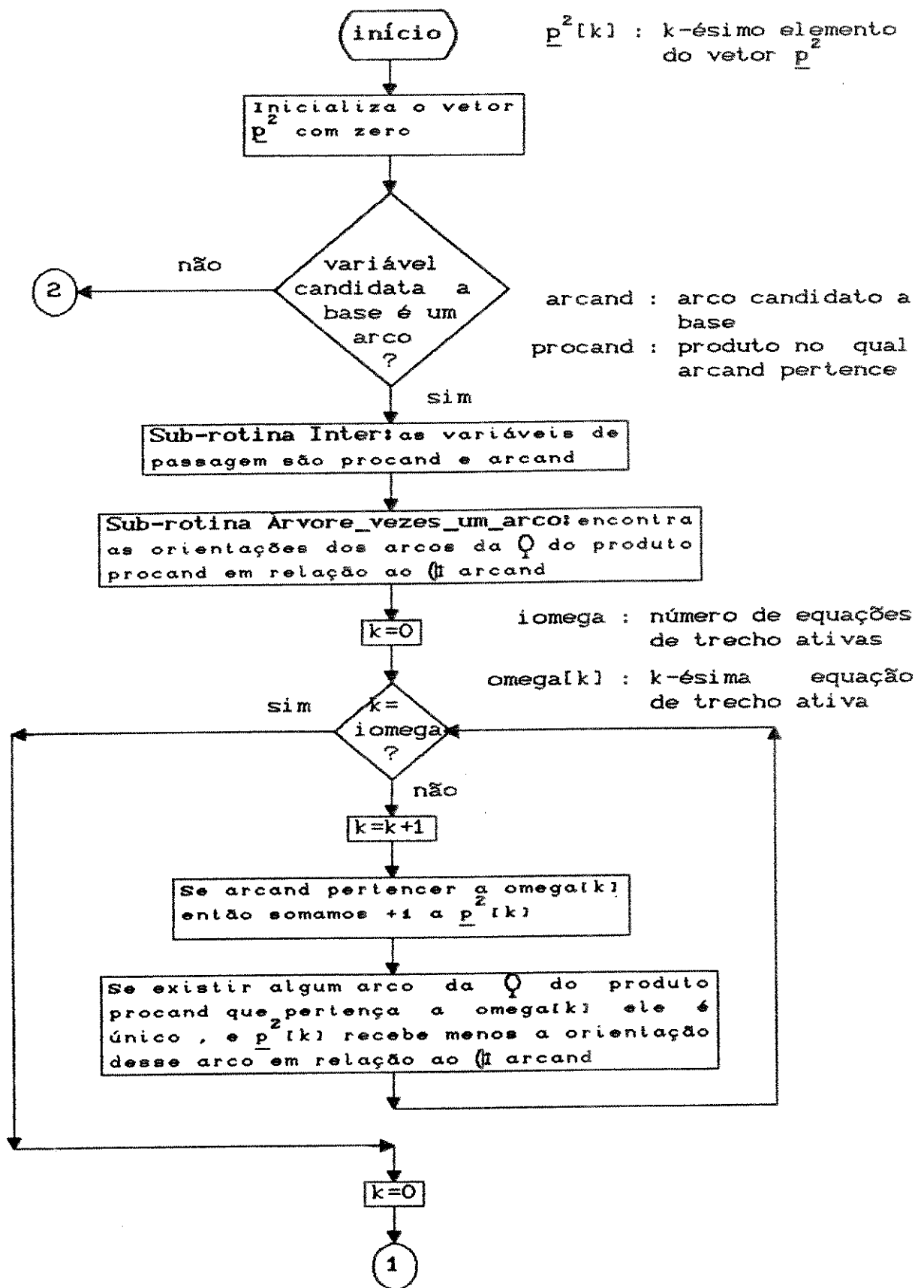
$\pi^2 eqc(k)$: variável dual referente a k-ésima equação de carga ativa

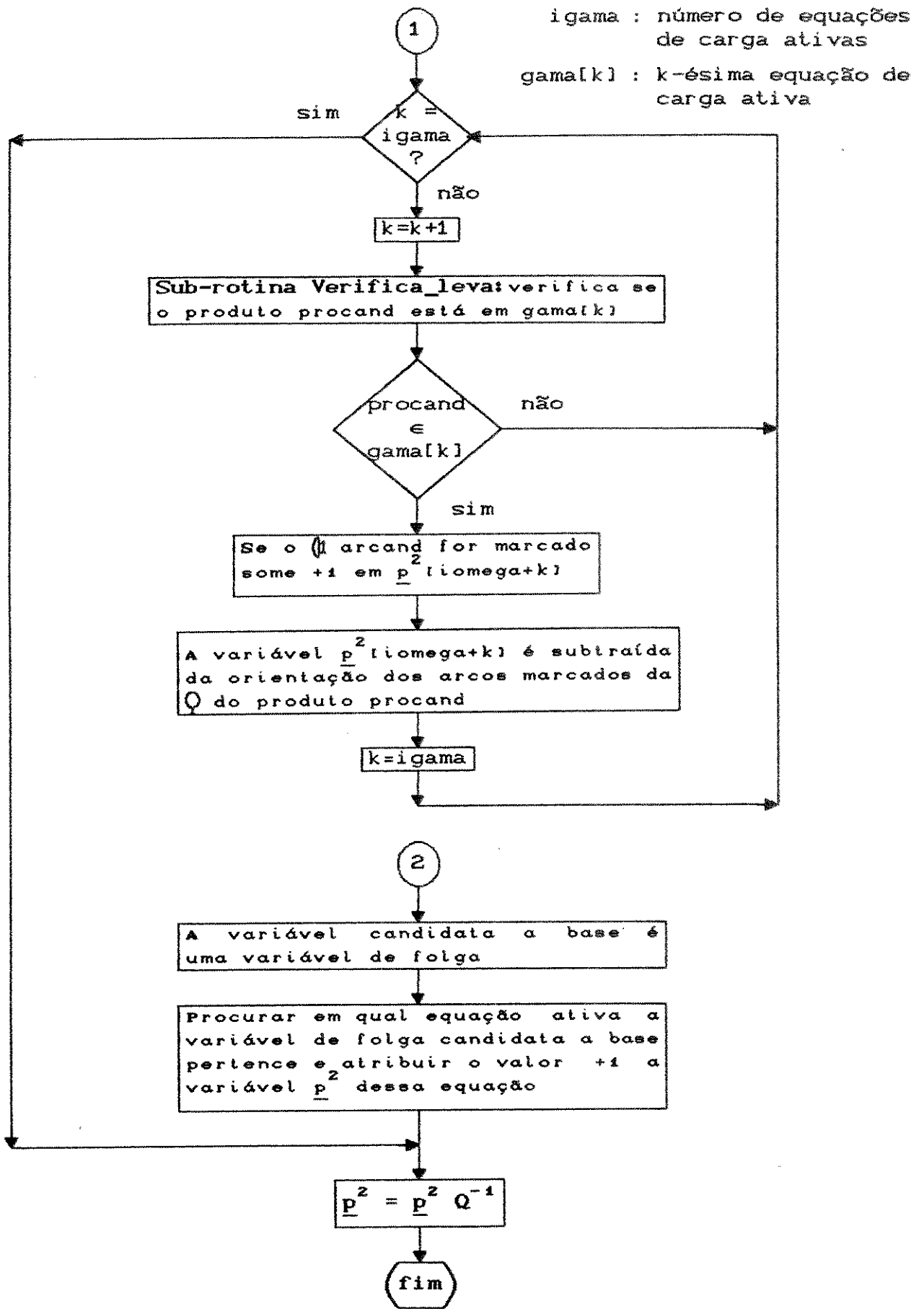


fluva[k]: variável que guarda o fluxo da k-ésima variável de folga fora da base

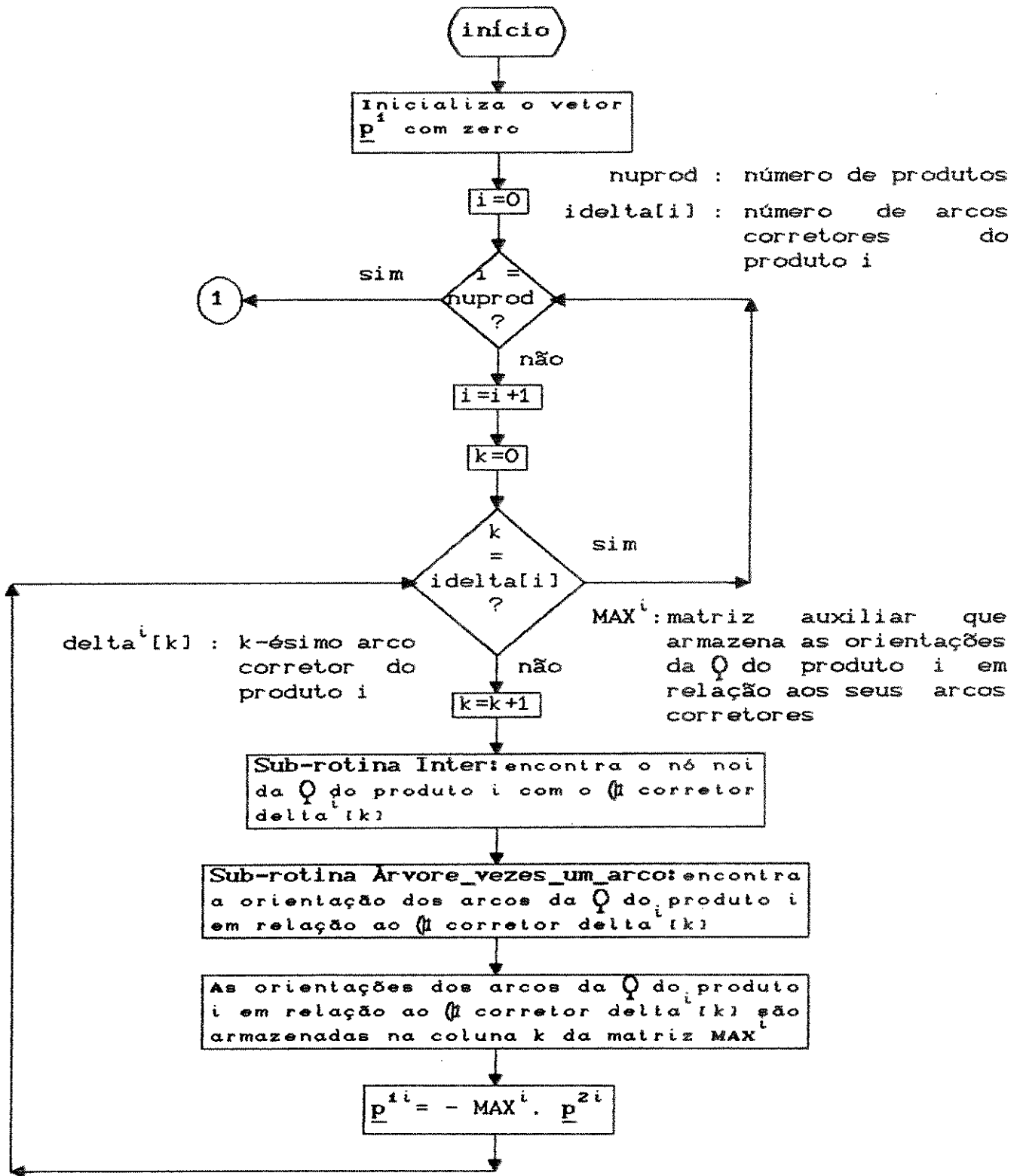


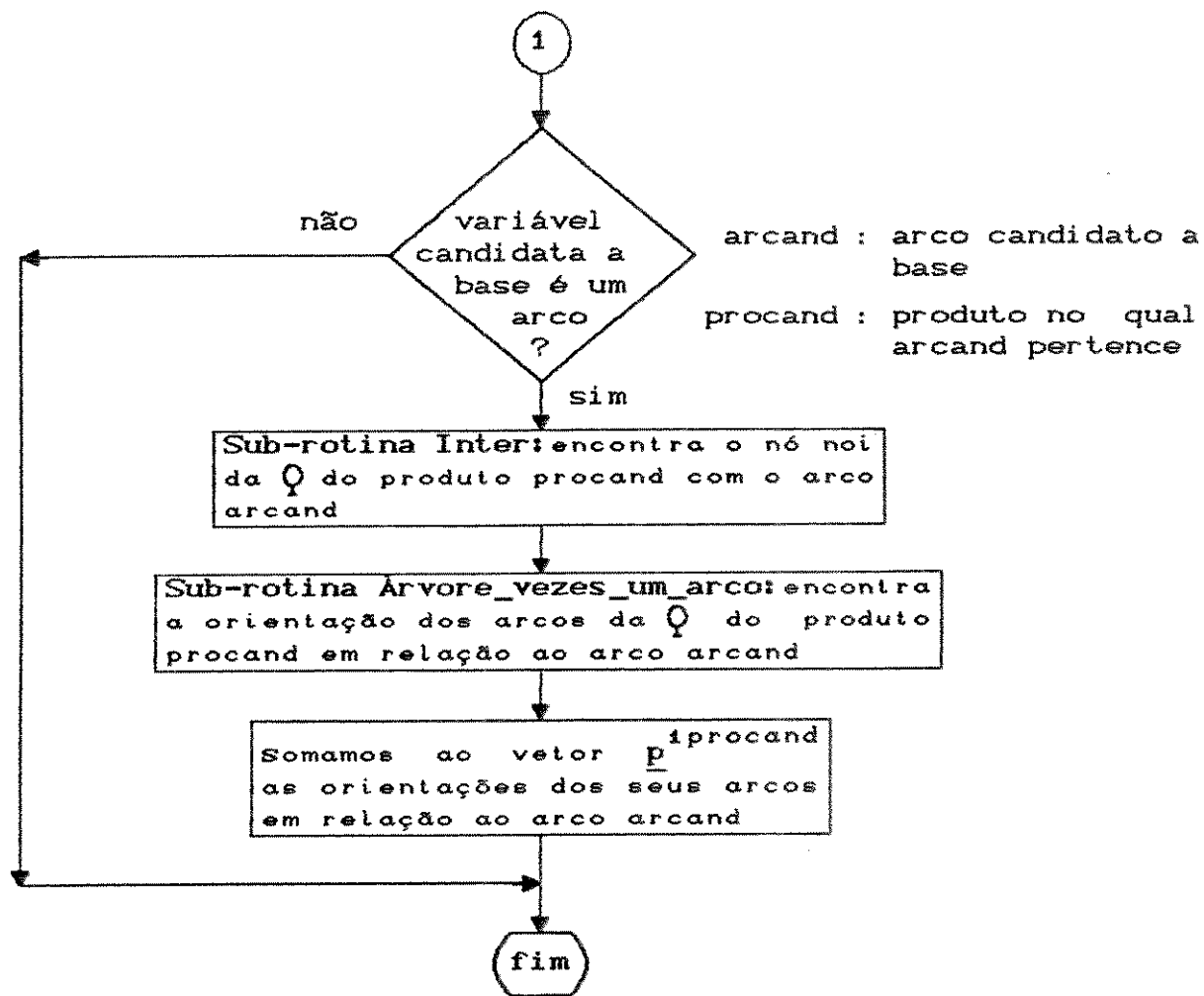
fluxograma 4.11 - Sub-rotina Encontra_p²



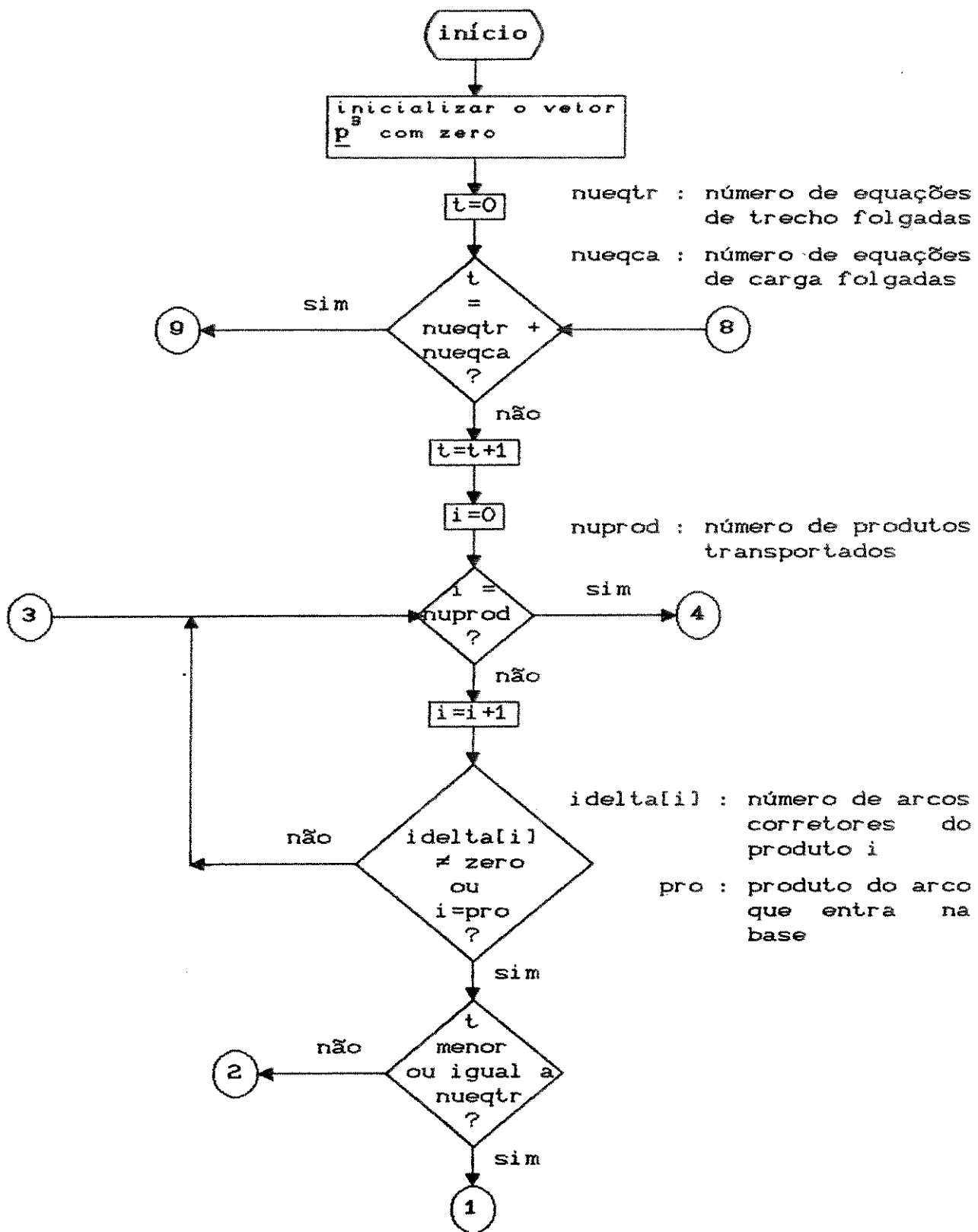


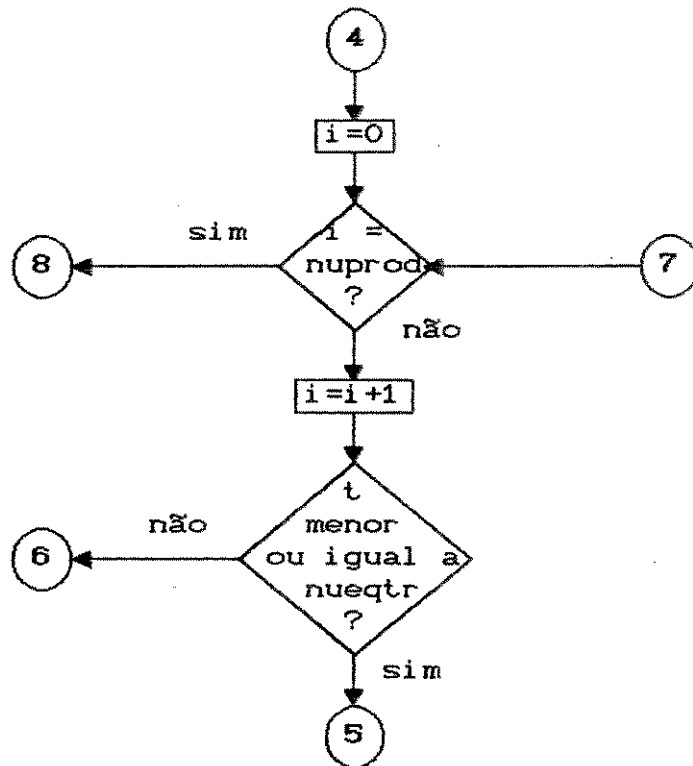
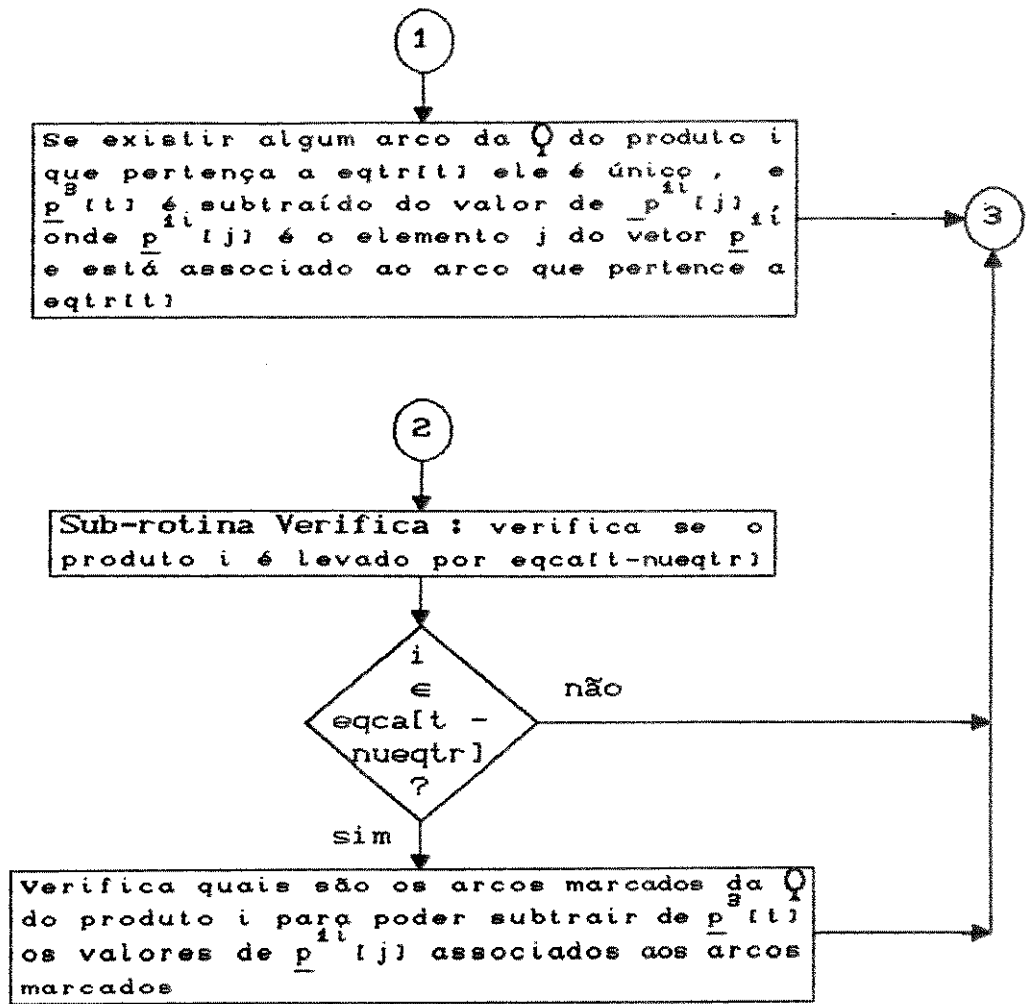
fluxograma 4.12 - Sub-rotina Encontra_pⁱ

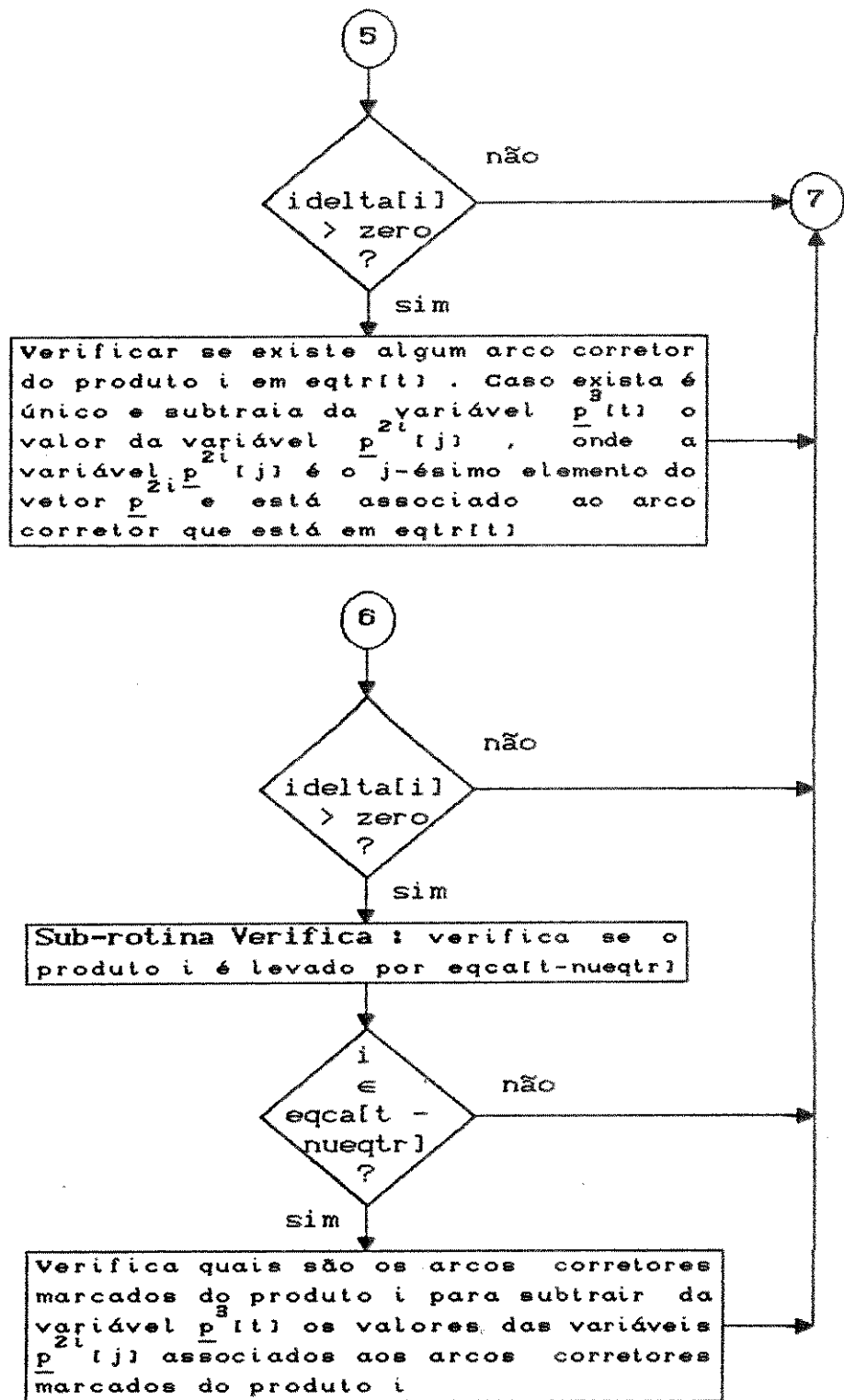


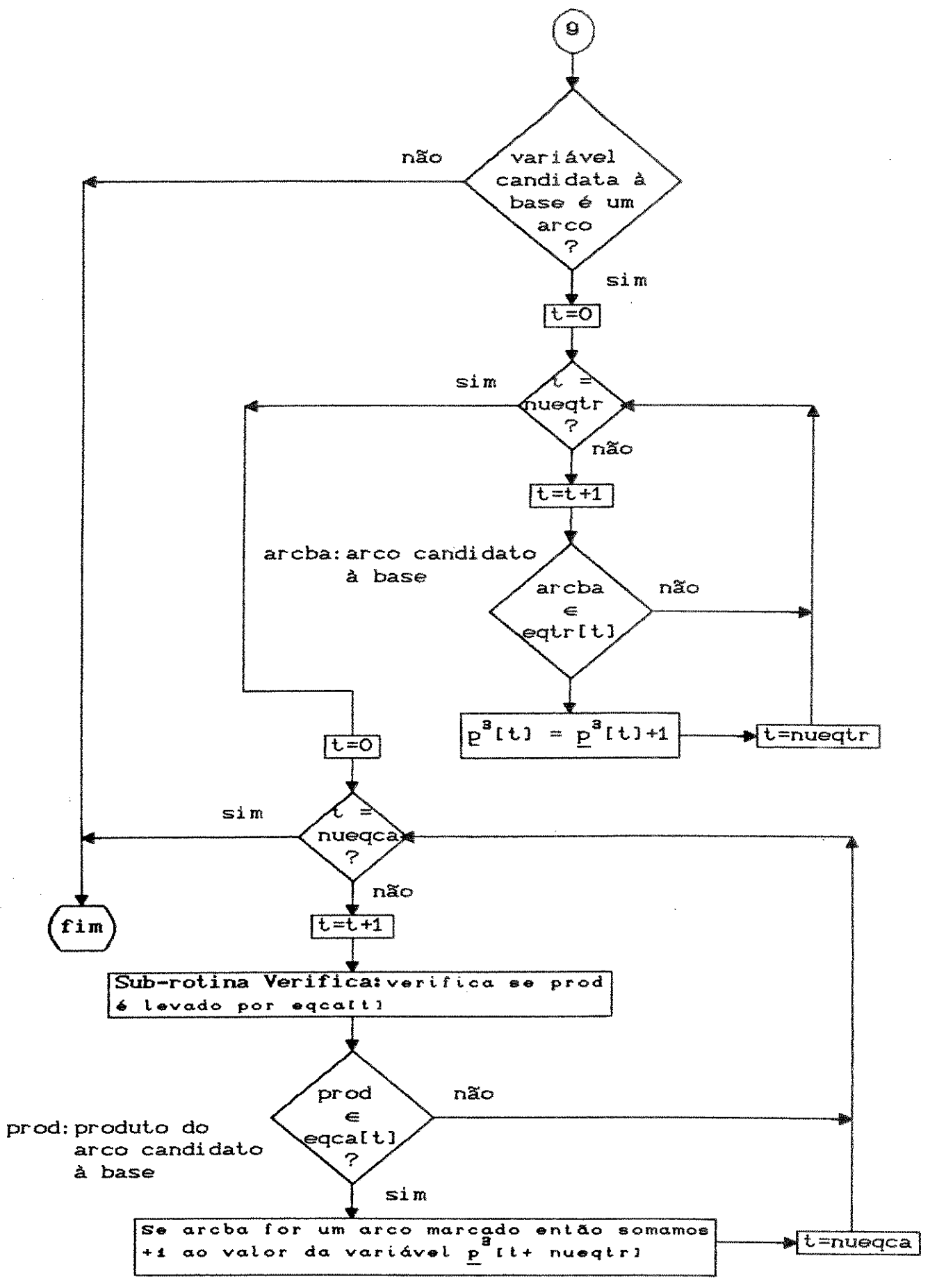


fluxograma 4.13 - Sub-rotina Encontra_p²

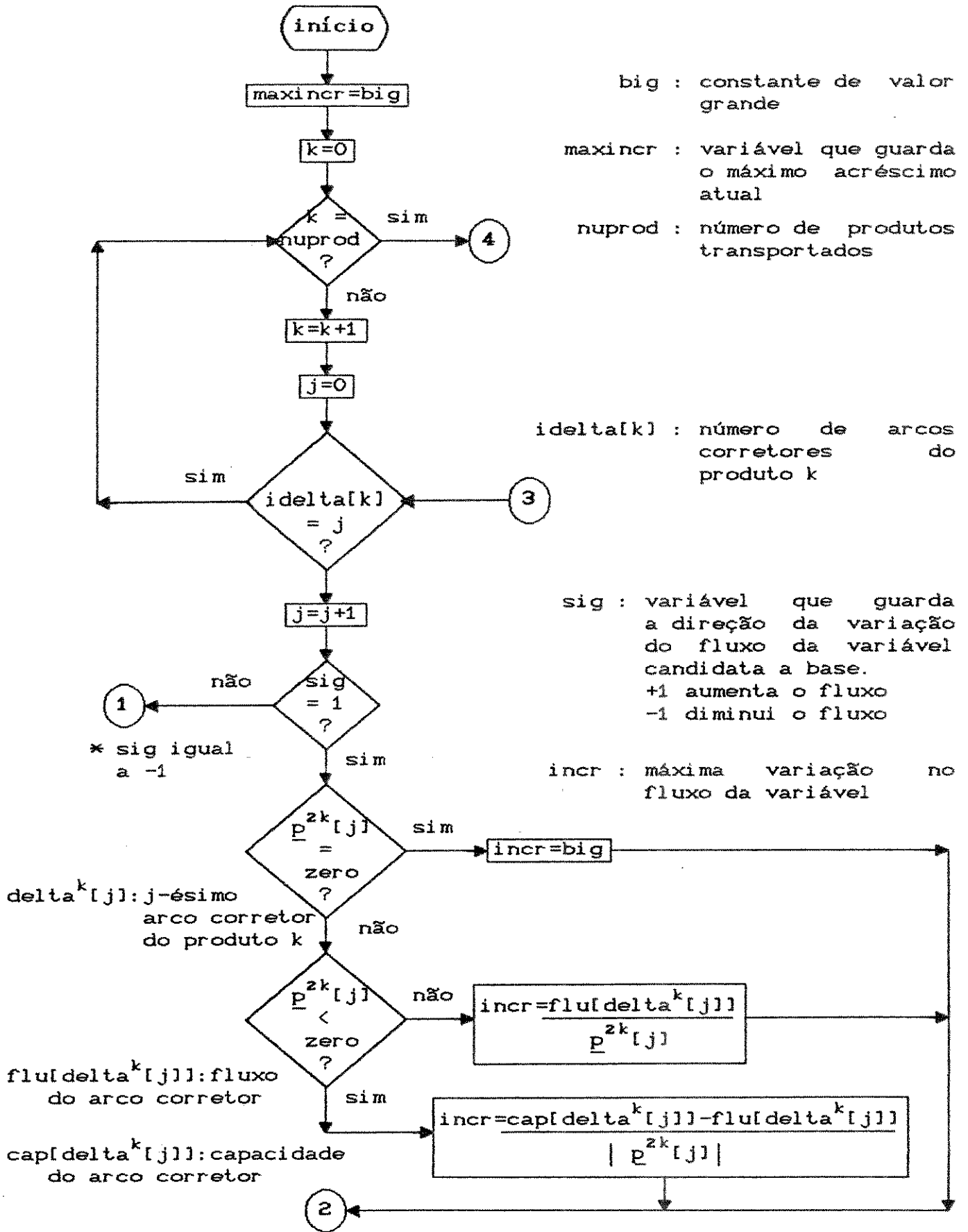


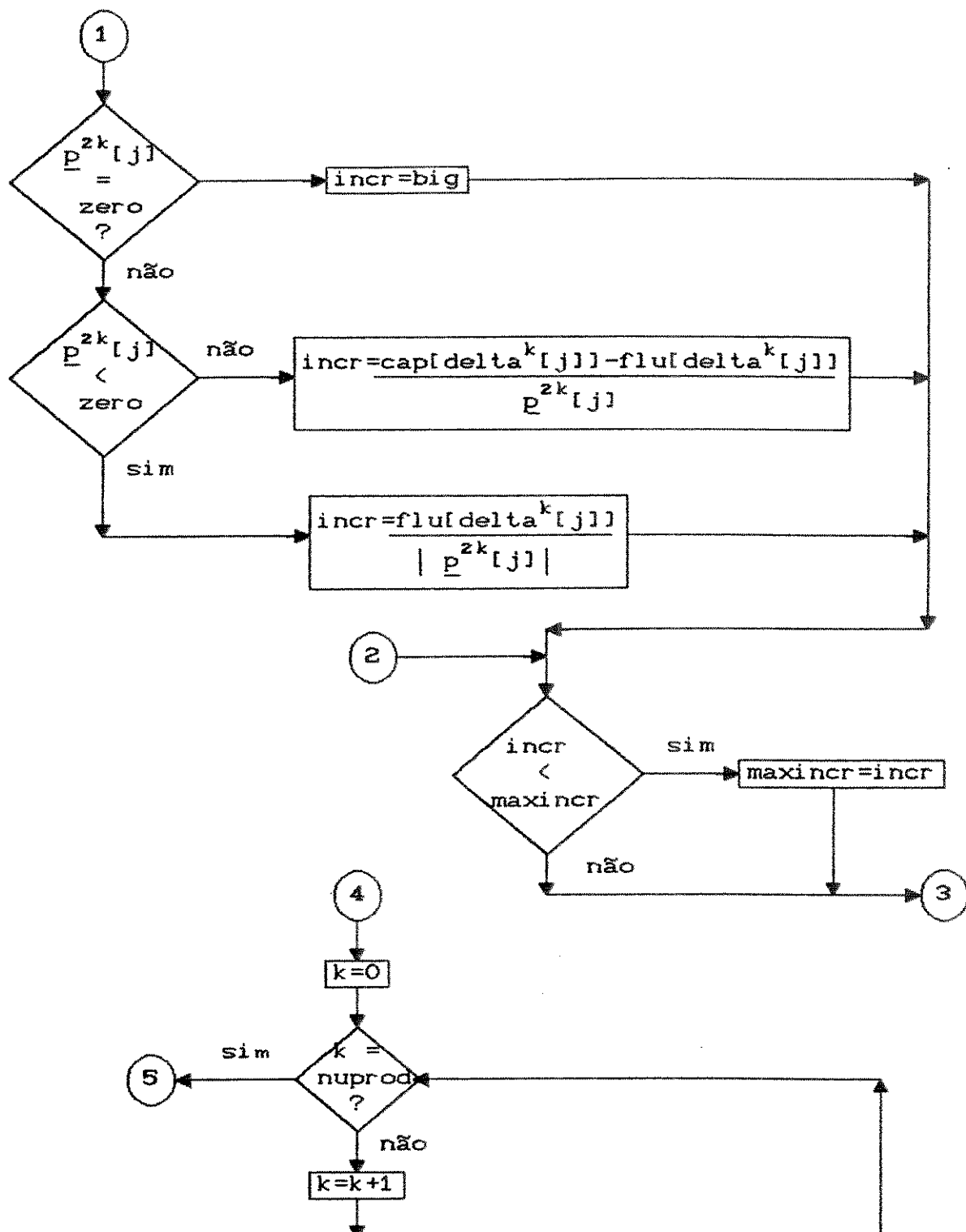




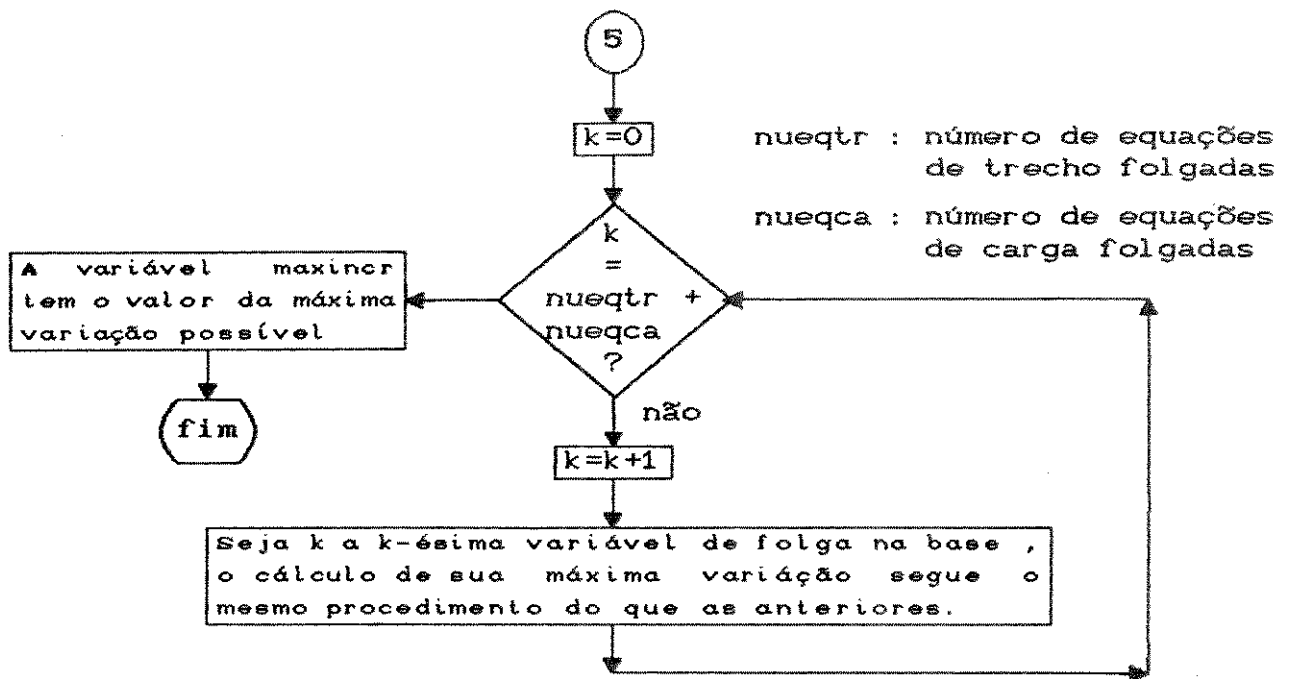


fluxograma 4.14 - Sub-rotina Cálculo_do_acrécimo_máximo

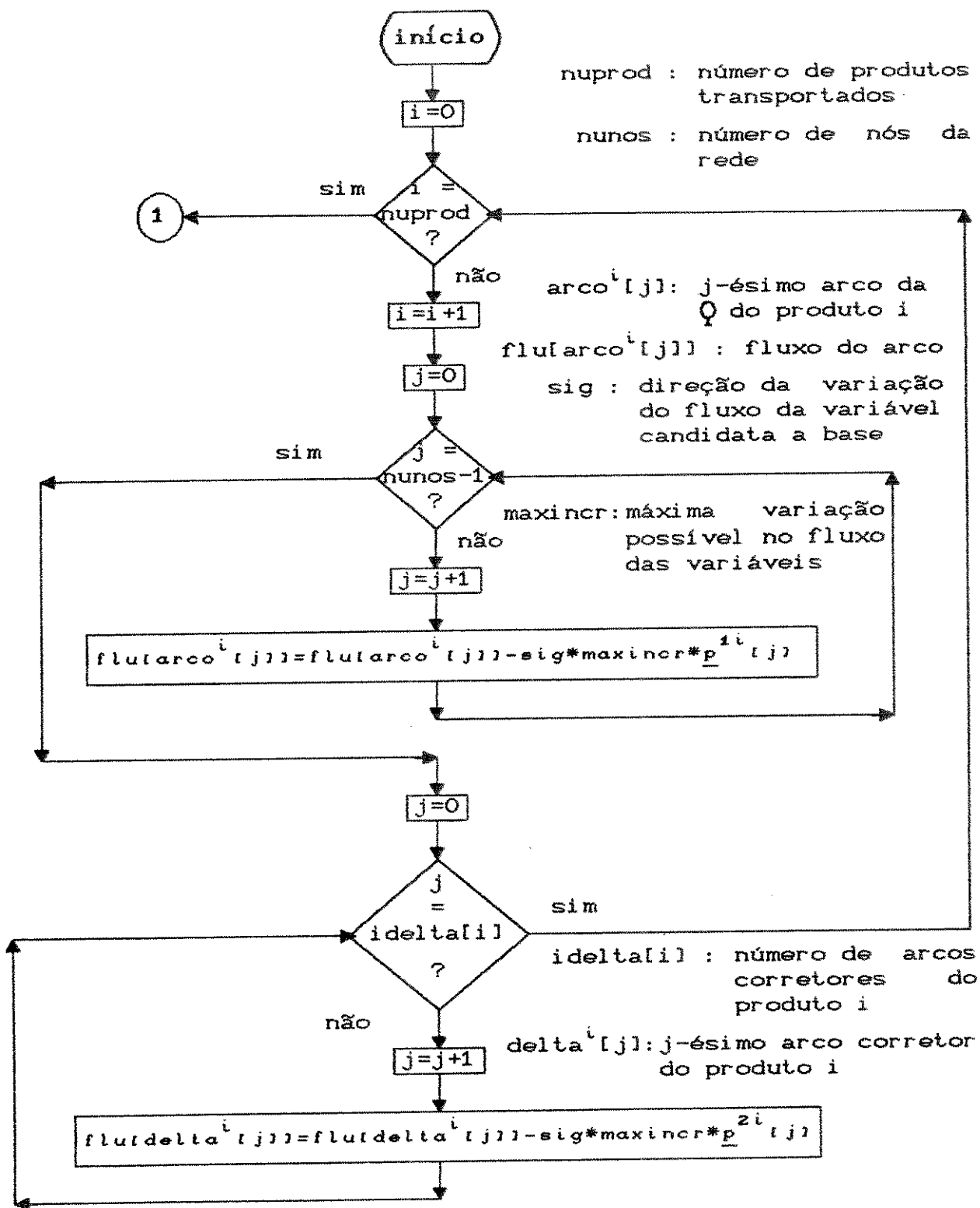


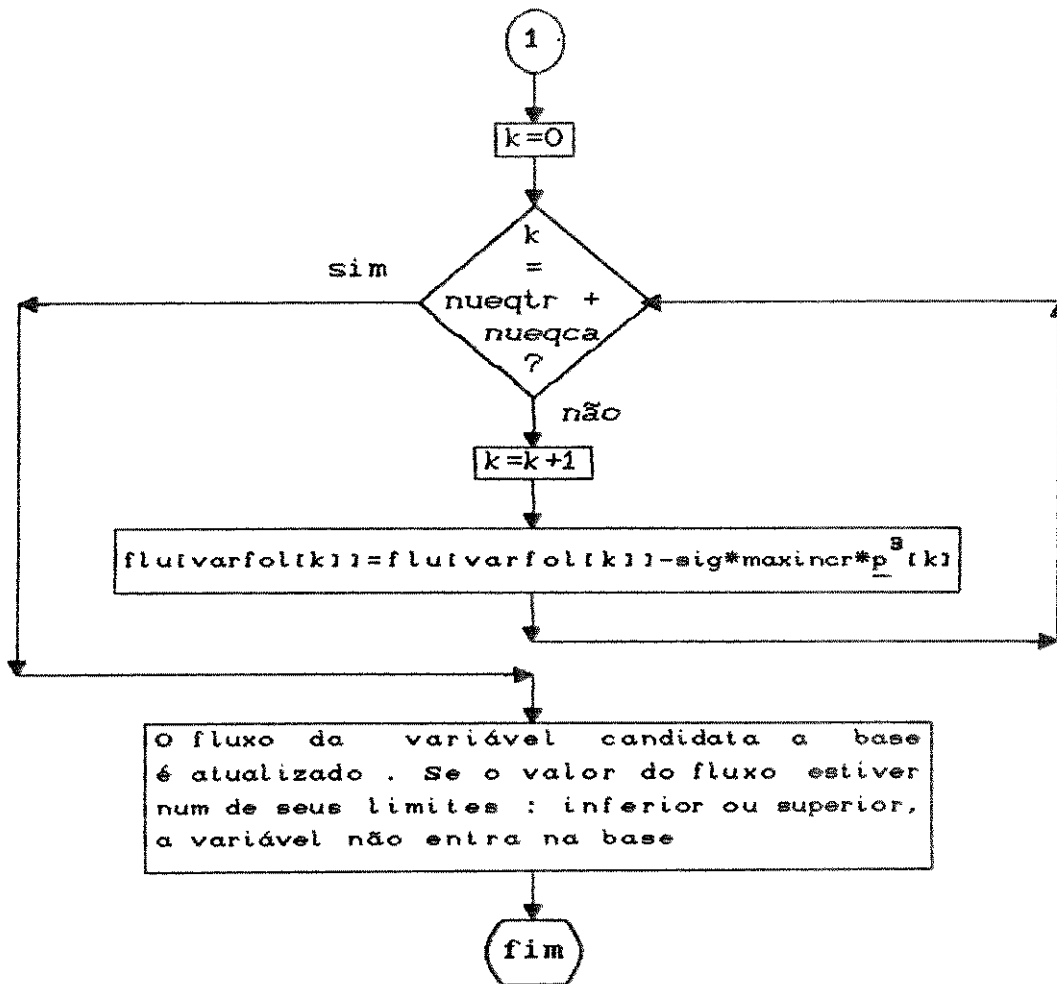


Para cada arco da Q do produto k estimar qual é a máxima variação possível no seu fluxo. Procedemos de maneira igual a anterior. Verificamos o valor da variável sig , verificamos o valor da variável do vetor p_{ik}^{2k} associado ao arco e calculamos a sua máxima variação pela fórmula correta. Sempre que o valor da variável $incr$ associada ao arco for menor que o valor da variável $maxincr$, atualizar o valor da variável $maxincr$ com o valor da variável $incr$.

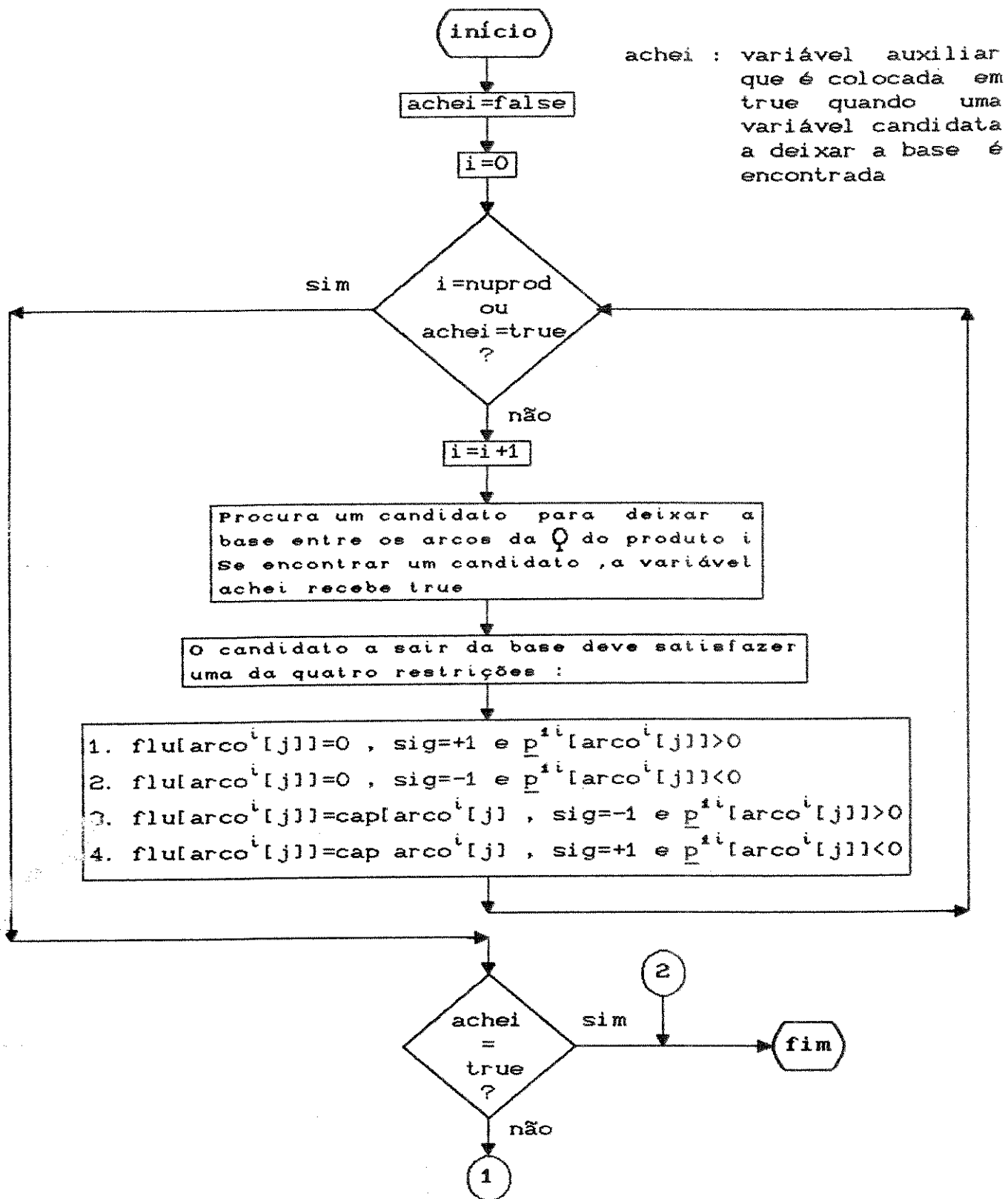


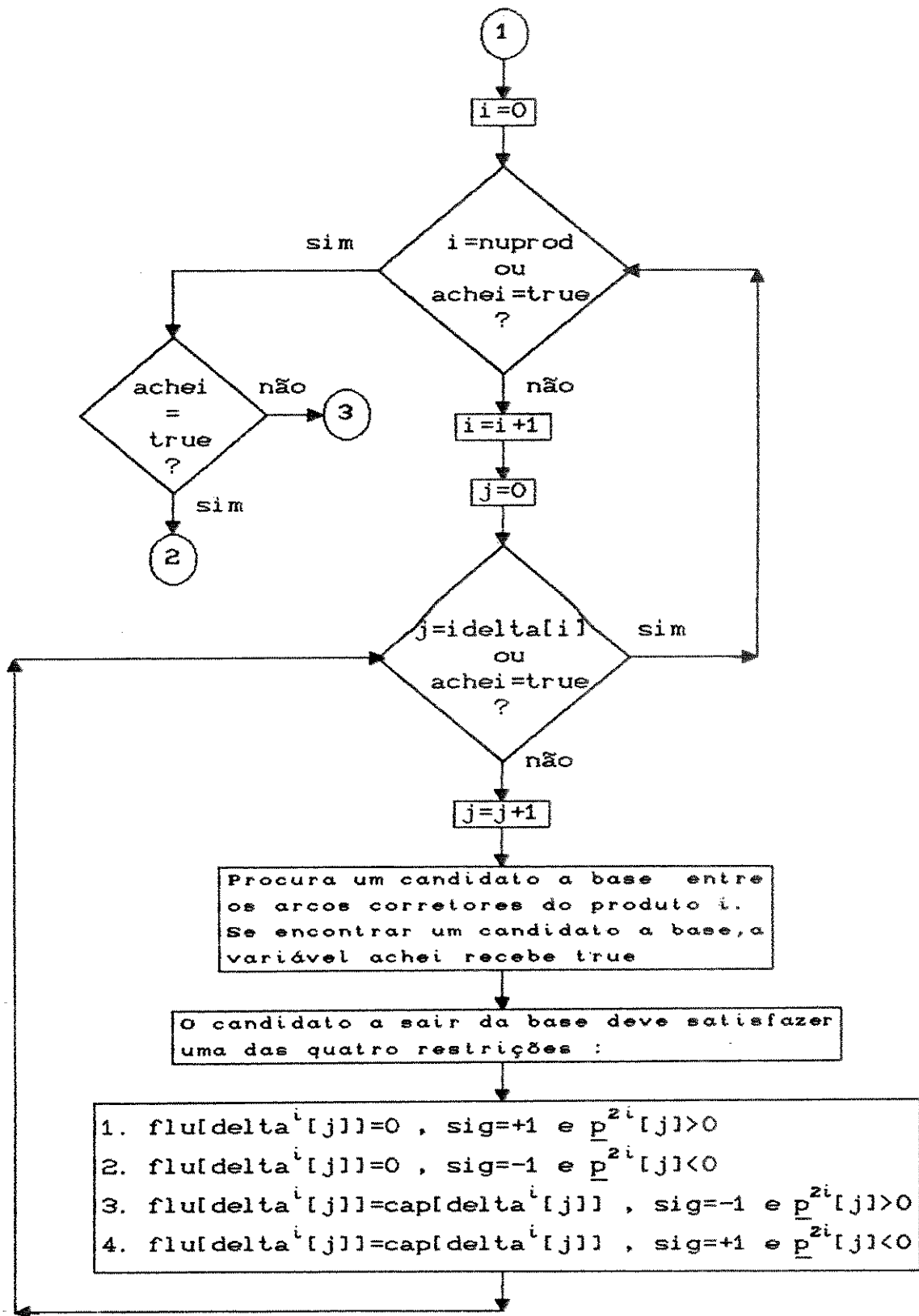
fluxograma 4.15 - Sub-rotina Atualiza_os_fluxos_das_variáveis_na_base

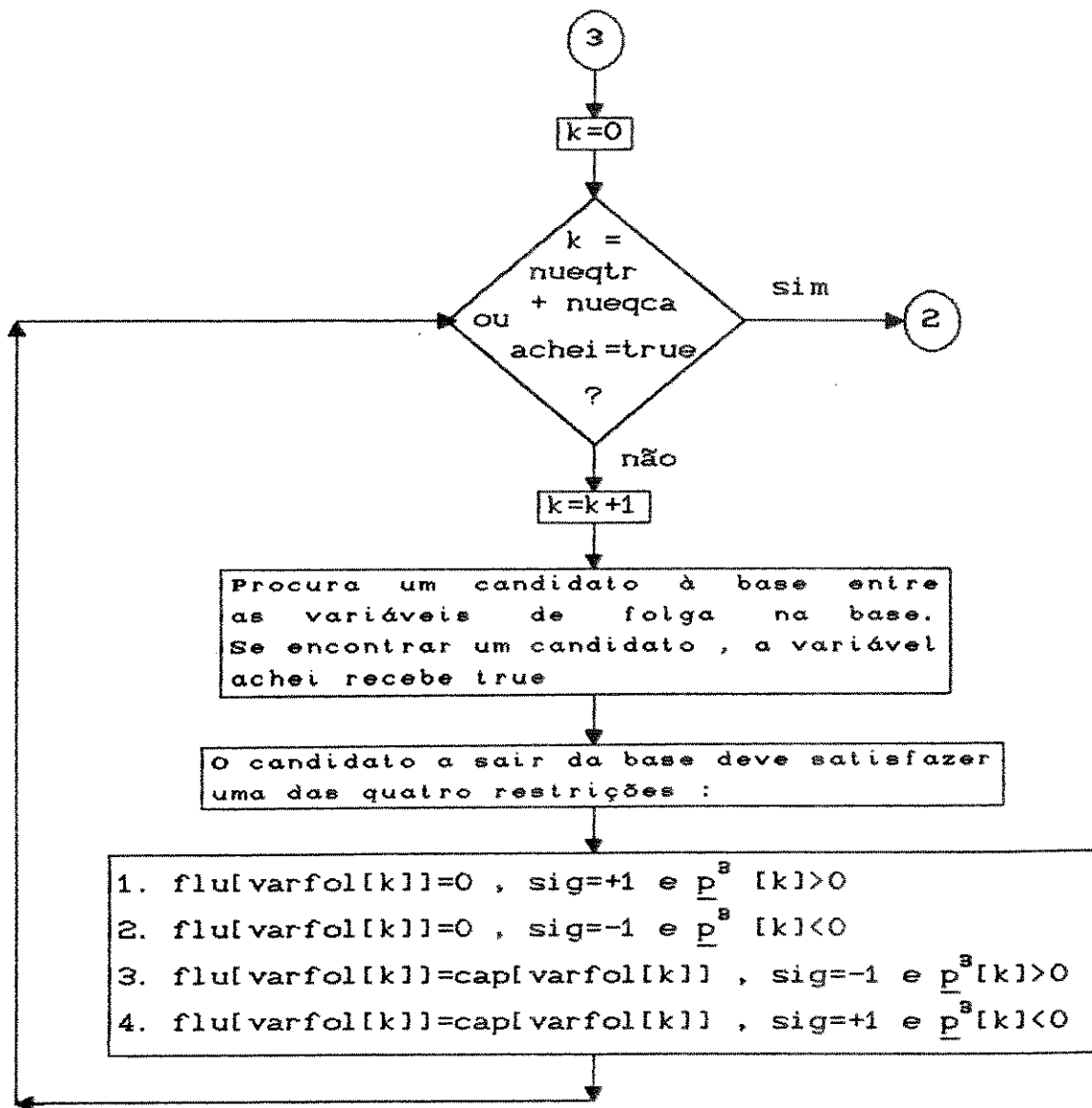




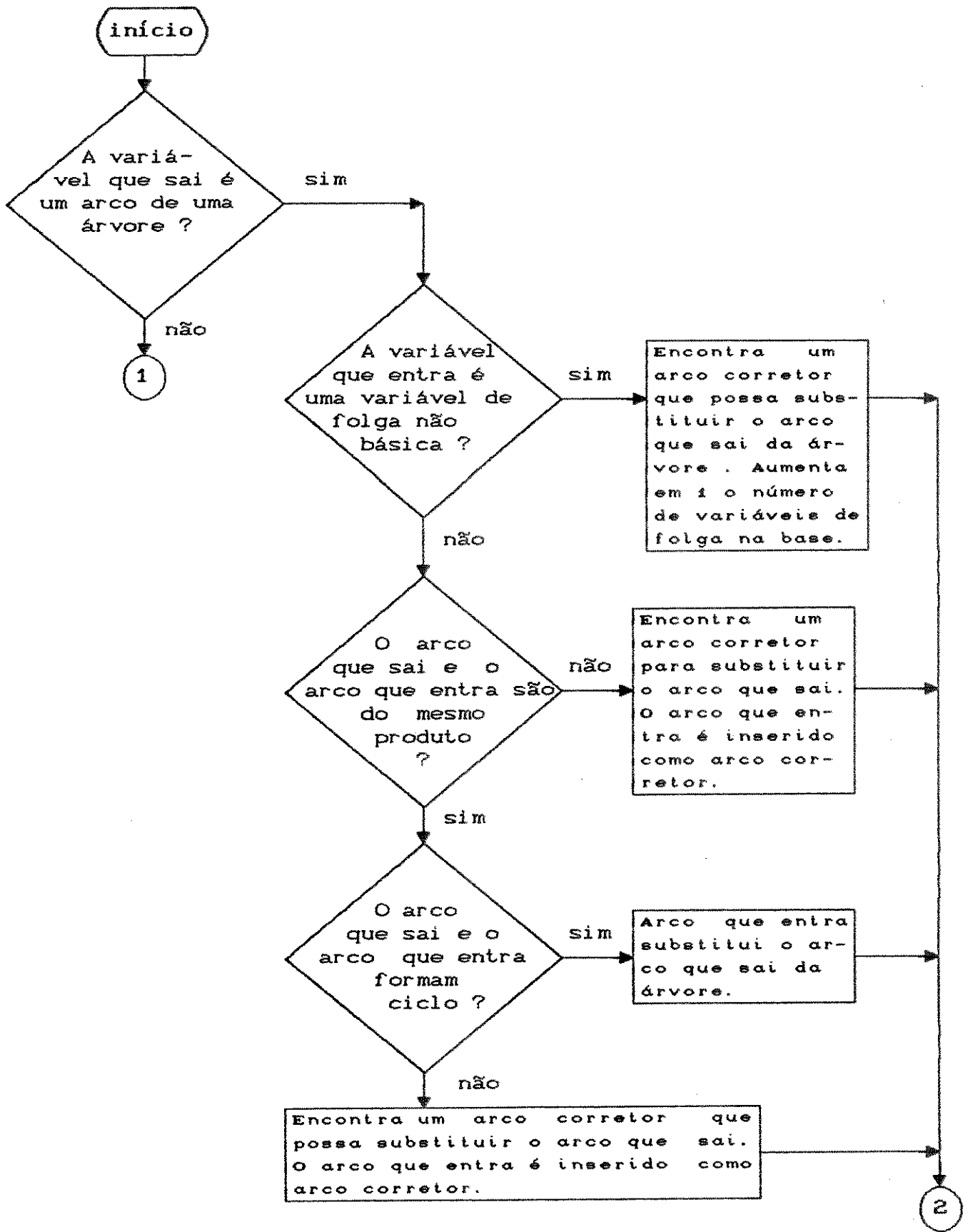
fluxograma 4.16 - Sub-rotina Variável_a_sair_da_base

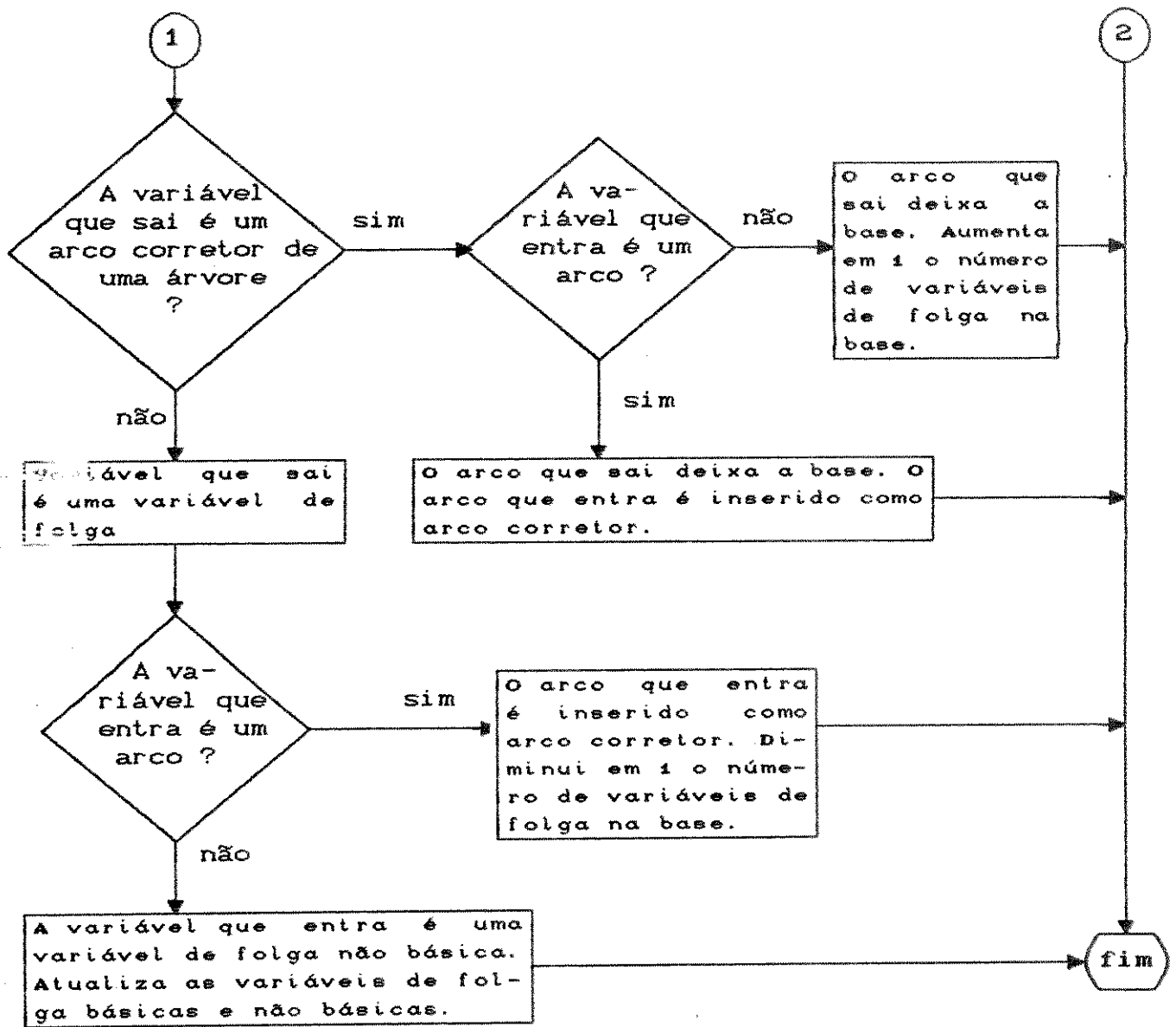






fluxograma 4.17 - Sub-rotina Reestrutura_a_base





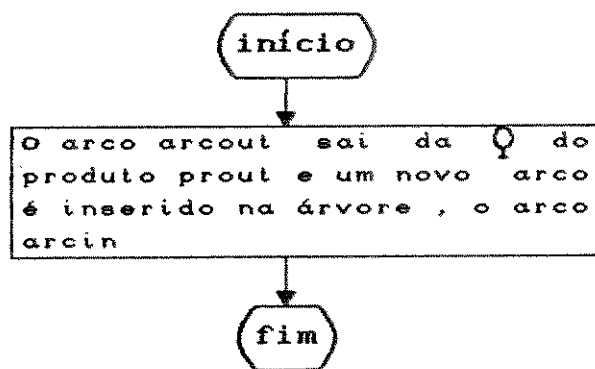
fluxograma 4.18 - Sub-rotina Reestrutura_árvore

variáveis de passagem de parâmetro:

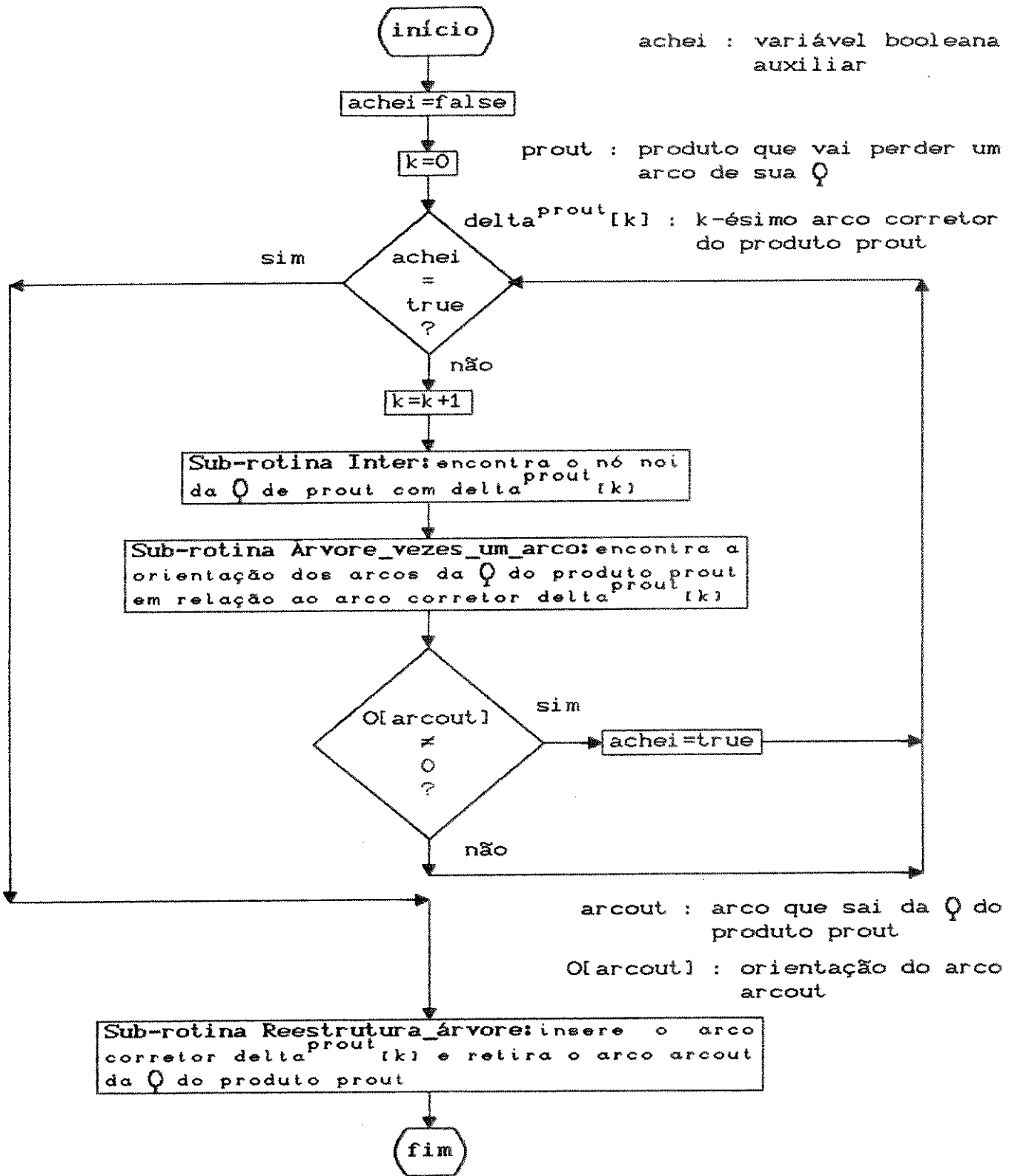
prout : produto que terá sua Q alterada

arcout: arco que deixa a Q

arcin : arco que entra na Q



fluxograma 4.19 - Sub-rotina Encontra_arco_corretor



CAPÍTULO V

Aplicações

V . 1. Definição dos Exemplos

Resolvemos a seguir dois exemplos que ilustram problemas de multifluxos numa malha ferroviária, sujeitos a restrições nos trechos e nas frotas de vagões, utilizando o programa Multipro.

Aos dois arcos que representam cada trecho da malha associamos o mesmo valor de capacidade, não haveria problemas em considerar capacidades diferentes.

Os dois exemplos diferem no tamanho de suas redes, número de produtos a transportar e número de tipos de vagões.

O primeiro exemplo trabalha com um rede de 10 nós e 48 arcos, transporta 20 produtos e trabalha com 2 tipos de vagões. O segundo exemplo tem uma rede de 89 nós e 358 arcos, transporta 10 produtos e trabalha com 4 tipos de vagões. Já estão incluídos os arcos artificiais e o nó artificial nas redes dos dois exemplos.

A rede utilizada no segundo exemplo representa a malha ferroviária de carga do Estado de São Paulo, obtida na Fepasa.

É necessário definir diversos tipos de dados para cada um dos exemplos. Representamos esses dados por meio de tabelas, algumas

delas estão colocadas no apêndice.

A forma na qual mostramos a resposta de cada um dos exemplos não representa a forma obtida na saída do programa Multipro, optamos por uma forma mais didática para facilitar a compreensão das mesmas. As soluções obtidas são colocadas sobre a própria rede, de forma individual para cada produto. Na solução para cada um dos produtos são indicados o nó origem, o nó destino, o custo do transporte, a carga que é transportada e a carga que não é transportada. O caminho ótimo do produto é marcado de forma destacada na rede. As soluções obtidas também estão colocadas no apêndice.

Após a entrada dos dados, o programa cria os arcos artificiais e o nó artificial para gerar uma solução inicial e factível para o problema. Essa solução inicial representa uma base para o problema e é composta pelas variáveis de folga das restrições e por árvores. Essas árvores são enraizadas e compostas por apenas arcos artificiais. O número de árvores é igual ao número de produtos transportados.

No exemplo 1 mostramos como ficam as árvores criadas pelo programa para gerar uma base. O custo que deve ser associado a cada um dos arcos artificiais precisa ser maior do que qualquer outro caminho que leve o produto de sua origem ao seu destino, caso contrário o programa escolheria um caminho artificial. A capacidade associada a cada um dos arcos artificiais é um valor que ultrapassa a soma total das quantidades de produtos que se quer transportar, para evitar o problema de saturação de um arco artificial.

A seguir definimos os exemplos 1 e 2.

V . 1. 1. Exemplo 1

O problema consiste em transportar 20 produtos diferentes por uma rede limitada, de maneira a transportar a maior quantidade de produtos possíveis e com o menor custo. A *Tabela 5.1* fornece os dados sobre os produtos a serem transportados: número do produto, estação origem, estação destino e a quantidade do produto que se deseja transportar.

produto	nó origem	nó destino (tons.)	carga
1	9	7	50
2	3	6	40
3	5	2	70
4	3	1	60
5	4	1	20
6	9	8	30
7	4	8	30
8	3	1	70
9	5	1	50
10	5	7	30

produto	nó origem	nó destino (tons.)	carga
11	4	6	30
12	9	6	60
13	9	8	40
14	3	2	30
15	9	7	20
16	5	7	20
17	4	1	40
18	4	8	30
19	5	6	50
20	5	8	30

Tabela 5.1 - Origem, destino e carga dos produtos

A tabela com os dados de todos os arcos reais e arcos não artificiais é a Tabela 5.2. Esses dados são: origem do arco, destino do arco e capacidade do arco.

núm. arco	nó origem	nó destino	capac. (tons.)
1	1	2	100
2	3	1	80
3	4	1	100
4	5	2	100000
5	3	2	50
6	3	6	90
7	4	6	70
8	4	8	60
9	5	6	40
10	5	7	50
11	9	5	40
12	5	8	80
13	9	6	70
14	9	8	70
15	9	7	100000

núm. arco	nó origem	nó destino	capac. (tons.)
16	2	1	100
17	1	3	80
18	1	4	100
19	2	5	100000
20	2	3	50
21	6	3	90
22	6	4	70
23	8	4	60
24	6	5	40
25	7	5	50
26	5	9	40
27	8	5	80
28	6	9	70
29	8	9	70
30	7	9	100000

Tabela 5.2 - Dados dos arcos reais do exemplo 1

A fig.5.1 ilustra a rede do exemplo 1 sem os arcos artificiais e o nó artificial. Não existe orientação nos arcos porque todos os arcos são bidirecionados.

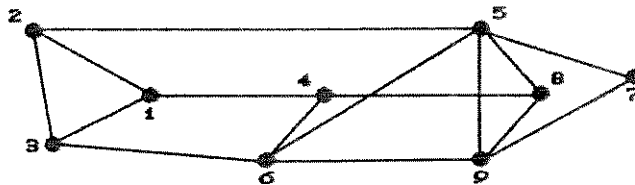


fig.5.1 - Rede do Exemplo 1

Existem apenas dois tipos de vagões para transportar os produtos: vagão tipo 1 e vagão tipo 2. A Tabela 5.3 mostra os produtos levados pelos tipos de vagões e a capacidade das frotas de vagões.

tipo de vagão	produtos levados	capacidade da frota (tons.)
1	1 a 7	290
2	8 a 20	510

Tabela 5.3 - Capacidade e produtos levados pelos tipos de vagões

Se somarmos a quantidade de carga necessária para transportar os produtos que utilizam o vagão tipo 1, verificamos que existe um excesso de 10 unidades em relação a capacidade de carga da frota. Esse excesso vai ficar nos arcos artificiais de algum produto e significa que é carga que não vai ser transportada, no caso, por falta de frota do tipo 1.

O custo do transporte de uma unidade de carga de um determinado produto por um arco varia de produto para produto. A tabela que contém o custo de cada produto em cada um dos arcos da rede está na Tabela 5.4.

número do arco	custos dos produtos (\$)																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	5	3	2	1	8	4	3	2	1	2	5	8	1	4	3	2	9	4	3	4
2	3	4	2	1	9	8	7	5	4	3	2	1	8	3	4	2	1	5	8	9
3	2	1	4	5	6	7	8	2	4	5	6	2	4	8	9	2	3	4	5	2
4	2	3	4	8	7	9	3	2	5	3	6	4	5	3	2	6	7	1	4	5
5	2	4	3	4	5	6	7	8	4	5	6	9	4	3	8	4	3	4	5	6
6	4	3	2	4	2	7	8	9	2	4	6	3	8	9	2	2	3	4	4	1
7	1	3	4	8	4	2	3	8	9	2	1	6	7	4	3	4	3	2	5	5
8	4	3	2	1	4	3	8	9	1	2	3	7	8	3	2	7	8	9	2	1
9	3	4	5	2	4	3	8	9	5	6	7	3	4	5	1	3	9	4	3	7
10	3	2	5	1	6	7	3	8	3	6	7	8	9	1	2	5	9	2	1	2
11	4	3	5	6	3	8	9	2	1	2	5	4	6	7	8	1	1	3	4	8
12	3	2	8	5	6	7	5	2	1	4	4	5	3	8	11	12	13	10	11	2
13	5	6	3	2	1	8	7	7	4	4	3	5	7	6	2	4	4	9	9	10
14	8	4	3	2	6	7	10	3	7	4	1	6	8	3	9	2	3	7	6	1
15	6	7	3	4	8	2	7	1	2	7	6	8	6	7	5	2	1	7	9	6
16	5	3	2	1	8	4	3	2	1	2	5	8	1	4	3	2	9	4	3	4
17	3	4	2	1	9	8	7	5	4	3	2	1	8	3	4	2	1	5	8	9
18	2	1	4	5	6	7	8	2	4	5	6	2	4	8	9	2	3	4	5	2
19	2	3	4	8	7	9	3	2	5	3	6	4	5	3	2	6	7	1	4	5
20	2	4	3	4	5	6	7	8	4	5	6	9	4	3	8	4	3	4	5	6
21	4	3	2	4	2	7	8	9	2	4	6	3	8	9	2	2	3	4	4	1
22	1	3	4	8	4	2	3	8	9	2	1	6	7	4	3	4	3	2	5	5
23	4	3	2	1	4	3	8	9	1	2	3	7	8	3	2	7	8	9	2	1
24	3	4	5	2	4	3	8	9	5	6	7	3	4	5	1	3	9	4	3	7
25	3	2	5	1	6	7	3	8	3	6	7	8	9	1	2	5	9	2	1	2
26	4	3	5	6	3	8	9	2	1	2	5	4	6	7	8	1	1	3	4	8
27	3	2	8	5	6	7	5	2	1	4	4	5	3	8	11	12	13	10	11	2
28	5	6	3	2	1	8	7	7	4	4	3	5	7	6	2	4	4	9	9	10
29	8	4	3	2	6	7	10	3	7	4	1	6	8	3	9	2	3	7	6	1
30	6	7	3	4	8	2	7	1	2	7	6	8	6	7	5	2	1	7	9	6

Tabela 5.4 - Custos dos produtos nos arcos reais da rede

Para cada um dos tipos de vagões precisamos identificar quais são os arcos marcados dos produtos que eles levam. Os arcos que devem ser marcados de um produto são os arcos que saem do nó origem. Por exemplo, tomando o produto 1, que é levado pelo tipo de vagão 1, seus arcos marcados são os seguintes arcos (fig 5.2):

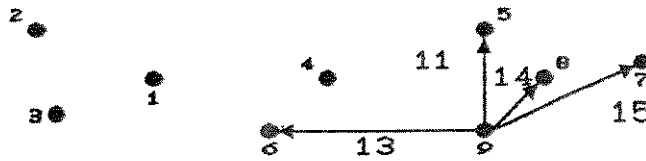


fig.5.2 - Arcos marcados do produto 1

Repetindo o processo de marcação dos arcos para todos os produtos chegamos a Tabela 5.5.

produto	arcos marcados
1	11,13,14,15
2	2, 5, 6
3	4, 9,10,12,26
4	2, 5, 6
5	3, 7, 8
6	11,13,14,15
7	3, 7, 8
8	2, 5, 6
9	4, 9,10,12,26
10	4, 9,10,12,26

produto	arcos marcados
11	3, 7, 8
12	11,13,14,15
13	11,13,14,15
14	2, 5, 6
15	11,13,14,15
16	4, 9,10,12,26
17	3, 7, 8
18	3, 7, 8
19	4, 9,10,12,26
20	4, 9,10,12,26

Tabela 5.5 - Arcos marcados dos produtos

Ao todo são inseridos na rede 18 arcos artificiais e um nó artificial, o nó de número 10. A tabela 5.6 mostra o número, a origem e o destino de cada um desses arcos. A fig.5.3 ilustra a rede com o nó artificial e os arcos artificiais.

núm. arco	nó origem	nó destino
31	1	10
32	2	10
33	3	10
34	4	10
35	5	10
36	6	10
37	7	10
38	8	10
39	9	10

núm. arco	nó origem	nó destino
40	10	1
41	10	2
42	10	3
43	10	4
44	10	5
45	10	6
46	10	7
47	10	8
48	10	9

Tabela 5.6 - Arcos artificiais do exemplo 1

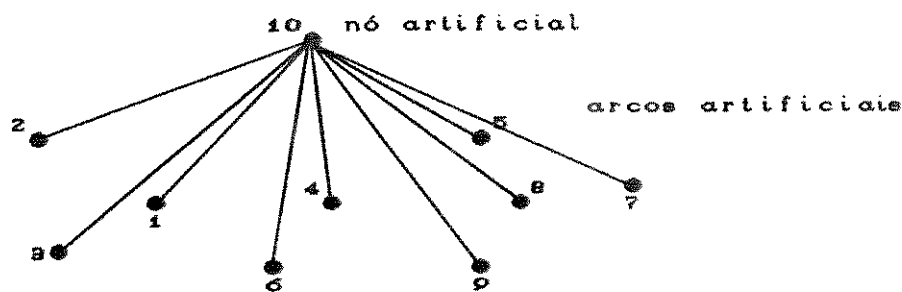


fig.5.3 - Rede com o nó artificial e os arcos artificiais

A solução inicial gerada pelo programa Multipro está colocada numa forma gráfica no apêndice, fig.B.1. Temos 20 árvores com apenas arcos artificiais que geram uma solução factível para cada um dos produtos a serem transportados.

V . 1. 2. Exemplo 2

Esse novo exemplo tem uma rede de dimensões maiores, mas trabalha com um número menor de produtos, 10 produtos. A Tabela 5.7 fornece o número do produto, sua origem, seu destino e a quantidade de carga que se quer transportar.

produto	estação origem	estação destino	carga (tons.)
1	1	2	106
2	7	3	70
3	1	2	44
4	40	14	40
5	13	27	200
6	19	69	120
7	6	15	55
8	13	28	201
9	2	77	28
10	2	80	18

Tabela 5.7 - Origem , destino e carga dos produtos

A tabela com os dados de todos os arcos reais está no apêndice, *Tabela B.1.*

A *fig.5.4* ilustra a rede do exemplo 2 sem os arcos artificiais e o nó artificial.

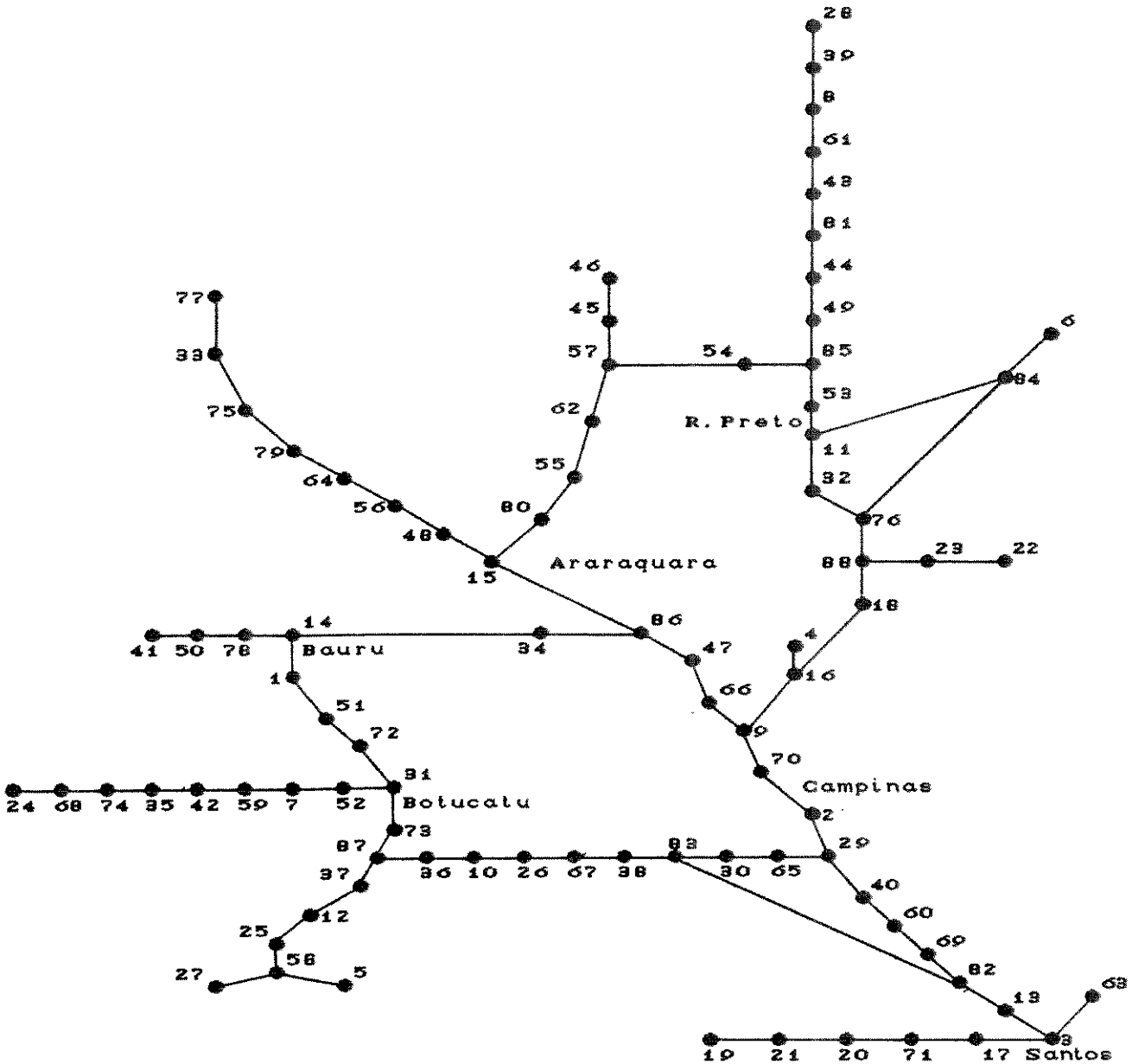


fig.5.4 - Rede do Exemplo 2

O número de tipos de vagões é 4, os produtos e a quantidade de carga que pode ser levada pelos tipos de vagões é dada pela Tabela 5.8.

<i>tipo de vagão</i>	<i>produtos levados</i>	<i>capacidade da frota (tons.)</i>
1	1,2,3	180
2	4,5	200
3	6,7	370
4	8,9,10	200

Tabela 5.8 - Capacidade e produtos levados pelos tipos de vagões

Os custos associados aos produtos que passam por um determinado arco da rede estão tabelados no apêndice, Tabela B.1.

Devemos fornecer os respectivos arcos marcados para cada um dos produtos que queremos transportar. A Tabela 5.9 fornece os arcos marcados de cada produto.

<i>produto</i>	<i>arcos marcados</i>
1	1 , 2
2	11 , 12
3	1 , 2
4	61 , 135
5	25 , 96

<i>produto</i>	<i>arcos marcados</i>
6	34
7	10
8	25 , 96
9	3 , 4
10	3 , 4

Tabela 5.9 - Arcos marcados dos produtos

Neste exemplo precisamos inserir na rede 176 arcos artificiais e 1 nó artificial. A lista completa desses arcos está no apêndice, Tabela B.2.

A solução inicial gerada pelo programa Multipro consta de 10 árvores com apenas arcos artificiais que são soluções factíveis para cada um dos produtos.

V . 2. Apresentação dos Resultados

A solução dos melhores caminhos para cada um os produtos está no apêndice. Representamos a solução dos produtos de forma individual e na própria rede.

Indicamos o nó origem do produto por uma seta entrando no nó e o nó destino por uma seta deixando o nó. O caminho que deve ser seguido pelos vagões está representado de forma mais destacada e com a devida orientação nos arcos. Indicamos ao lado do primeiro arco no caminho, a quantidade do produto que está sendo transportada. Alguns produtos podem ter mais do que um caminho para escoar os seus produtos. Isso ocorre porque algum arco da rede está com a sua capacidade tomada, obrigando os produtos a escolherem caminhos alternativos para escoarem uma parte ou toda a sua carga. Quando isso ocorre, indicamos também ao lado a quantidade de carga que está sendo levada.

Quando um produto possui um quantidade de carga que não pode ser levada, por esgotamento de caminhos alternativos ou por falta de frota de vagões, essa quantidade é desviada para um caminho artificial. A quantidade de carga é indicada ao lado do primeiro arco artificial desse caminho.

Na solução de cada produto apresentamos o custo total do transporte e as quantidades que devem e que não devem ser transportadas.

V . 2. 1. Resultados do Exemplo 1

A solução obtida para o exemplo 1 é constituída pelos melhores caminhos na rede para os produtos que têm a sua carga transportada.

A Tabela 5.10 ilustra a carga que cada produto deve transportar e custo total do seu transporte.

<i>produto</i>	<i>carga total</i>	<i>carga transp. (tons.)</i>	<i>carga não transp. (tons.)</i>	<i>custo (\$)</i>
1	50	50	-	300
2	40	40	-	120
3	70	70	-	280
4	60	50	10	130
5	20	20	-	120
6	30	30	-	210
7	30	30	-	240
8	70	70	-	630
9	50	50	-	300
10	30	30	-	180
11	30	30	-	30
12	60	60	-	300
13	40	40	-	320
14	30	30	-	90
15	20	20	-	100
16	20	20	-	60
17	40	40	-	120
18	30	30	-	270
19	50	50	-	250
20	30	30	-	60
total	800	790	10	4110

Tabela 5.10 - Cargas transportadas pelos produtos do exemplo 1

A solução com os caminhos que devem tomar as cargas está no apêndice e os dados computacionais do exemplo estão na *Tabela 5.11*.

<i>nós</i>	<i>arcos</i>	<i>produtos</i>	<i>custo total (\$)</i>	<i>iterações</i>	<i>tempo</i>	<i>máquina</i>
10	48	20	4110	142	42 seg	PC-AT

Tabela 5.11 - Dados computacionais do exemplo 1

V . 2. 2. Resultados do Exemplo 2

Como no exemplo 1, construímos uma tabela com a carga transportada e não transportada de cada produto juntamente com o seu custo. A *Tabela 5.12* ilustra esses dados.

<i>produto</i>	<i>carga total</i>	<i>carga transp. (tons.)</i>	<i>carga não transp. (tons.)</i>	<i>custo (\$)</i>
1	106	106	-	27168
2	70	30	40	14370
3	44	44	-	17160
4	40	40	-	13400
5	200	160	40	73780
6	120	120	-	40920
7	55	55	-	18480
8	201	154	47	130900
9	28	28	-	18004
10	18	18	-	5670
<i>total</i>	882	755	127	359852

Tabela 5.12 - Cargas transportadas pelos produtos do exemplo 2

A solução para o exemplo 2 está esboçada na *fig. B.3* do apêndice, os dados computacionais do exemplo estão na *Tabela 5.13*.

<i>nós</i>	<i>arcos</i>	<i>produtos</i>	<i>custo total (\$)</i>	<i>iterações</i>	<i>tempo</i>	<i>máquina</i>
89	358	10	359852	850	20 min	PC-AT

Tabela 5.13 - Dados computacionais do exemplo 2

V . 3. Comparações

Para podermos verificar se os problemas testados com o nosso programa eram confiáveis utilizamos, um pacote de programação linear para microcomputadores para resolver os mesmos problemas: utilizamos o Hiperlindo. Além de verificar se as soluções coincidiam, pudemos também comparar outros parâmetros entre os dois programas, tais como: tempo para preparar os problemas, tempos de execução e o número de iterações.

O pacote Hiperlindo resolve problemas de programação linear,

inteira e quadrática. A versão que utilizamos pode guardar até 32000 variáveis diferentes de zero respeitando o seu número máximo de linhas e colunas (3999 colunas e 1999 linhas).

Utilizamos esse pacote para resolver o exemplo 1. A forma de entrada das expressões foi pelo modo direto na tela. Tal forma é bem didática, mas cometido um engano na digitação de um dado torna-se difícil encontrá-lo, fazendo com que o tempo para preparar o problema aumente. O exemplo 1 equivale a um problema linear de 233 linhas (1 linha da função objetivo, 200 linhas de equações de balanceamentos dos nós, 30 linhas de equações de trecho e 2 linhas de equações de frota) e 992 colunas (960 arcos e 32 variáveis de folga). O tempo de execução foi de 60 segundos e teve 139 iterações.

O valor do custo total coincidiu com o resultado obtido com o programa Multipro, de todos os caminhos dos produtos apenas um ficou diferente. Para o produto 9 obtivemos uma solução um pouco diferente em cada um dos programas. Isso mostra que a solução desse problema não é única.

Reescrevemos os dados de ambos os programas na Tabela 5.14.

<i>programa</i>	<i>iterações</i>	<i>custo total (\$)</i>	<i>tempo de execução</i>	<i>máquina</i>
Multipro	142	4110	42 segs.	PC-AT
Hiperlindo	139	4110	60 segs.	PC-AT

Tabela 5.14 Comparações entre os programas Multipro e HiperLindo para o exemplo 1

O exemplo 2 equivale a um programa linear de 1077 linhas (1 linha da função objetivo, 890 linhas de equações de balanceamento de nós, 182 linhas de equações de trechos e 4 linhas de equações de frota) e 3766 colunas (3580 arcos e 186 variáveis de folga).

Nós não preparamos o exemplo 2 para o HiperLindo, pois [Sharda, 1986], que fez um estudo comparativo entre os pacotes de programação linear para microcomputadores, incluindo o HiperLindo, relatou que problemas do porte do exemplo 2 tomam um tempo computacional de 1 hora ou mais, que é superior ao tempo obtido

utilizando o programa Multipro (20 minutos). Outro motivo que nos fez abandonar a simulação foi que necessitaríamos muitas horas para prepararmos esse problema no HiperLindo.

CAPÍTULO VI

Conclusões Finais

VI. 1. Conclusões

Ao final deste trabalho, chegamos a algumas conclusões importantes sobre o método primal particionado aplicado a uma rede com multifluxo e restrições adicionais ao fluxo.

A listagem do programa ficou extensa, tornando-se difícil fazer alterações, pois cada alteração compromete um número grande de partes do programa. Gasta-se muito tempo para se fazer tais alterações e, portanto o programa exige uma boa documentação.

O método foi implementado na linguagem Pascal, Turbo-Pascal versão 4.0 e rodado num microcomputador PC-AT com DOS. Neste caso, o número máximo de bytes que o compilador reserva para as variáveis globais é de 65520 bytes, pois quando o programa é "linkado", as variáveis globais usadas pelo programa são colocadas no segmento de dados, que tem um tamanho máximo de 65520 bytes. Este fato limitou o teste do programa para problemas maiores. Num ambiente sem estas restrições, é possível resolver problemas bem maiores em micros.

VI . 3 Propostas Futuras

Para garantirmos que o nosso programa seja mais confiável, deveríamos simular uma grande quantidade de exemplos, dos mais variados, para que o programa passe por todas as situações possíveis.

Podemos citar algumas partes do programa Multipro que se melhoradas, diminuirão o tempo de convergência para a solução ótima:

- Cálculo da inversa da matriz ciclo (Q);
- Geração de uma melhor solução inicial;
- Estudar um melhor processo de escolha de candidatos a base;
- Simplificações em algumas sub-rotinas do programa.

O problema de melhorar o método de calcular a matriz inversa da matriz ciclo é tratado em [Kennington, 1980] e [Helgason, 1977]. Eles sugerem um método de atualização da matriz ciclo utilizando a matriz inversa da iteração anterior. O método cria casos e sub-casos para as alterações na matriz ciclo, dependendo das variáveis que entram e saem da base. Poderíamos então adaptar esse método ao nosso programa para melhorar sua eficiência. Kennington (1980) declara que é de extrema importância que o algoritmo possua uma boa forma de inversão da matriz ciclo quando os problemas possuem matrizes ciclo com dimensões grandes.

Para melhorar a geração de uma solução inicial poderíamos criar um método para gerar caminhos com arcos da rede ao invés de utilizarmos os arcos artificiais, muito embora, seja necessário introduzi-los na rede para garantir o seu balanceamento. Programas utilizando inteligência artificial seriam uma boa alternativa.

Kennington (1977) relatou três diferentes maneiras de estimar os candidatos a base: o primeiro candidato a base, o candidato que apresenta o maior valor em módulo do seu custo relativo e o candidato que apresenta o maior valor em módulo em uma certa quantidade de

candidatos. Das três opções a que obteve melhores resultados foi a segunda, ou seja, verificar entre todos os candidatos a base o que apresenta o maior valor em módulo de custo relativo. A primeira opção foi a que apresentou piores resultados, levando o autor a descartá-la para problema grandes. A terceira e última opção apresentou resultados quase tão bons quanto os da segunda. Acreditamos que deveríamos adotar a terceira opção para buscar resultados, porque conforme os problemas aumentam de dimensão fica custoso procurar entre todos os candidatos a base.

Uma melhor reestruturação nos algoritmos pode aumentar a velocidade das iterações, pois muitas operações desnecessárias são realizadas em cada uma das iterações. Por exemplo, no cálculo das variáveis duais não precisamos recalculá-las todas.

Abordamos uma das técnicas de resolvermos o problema. Então seria importante implementarmos outras metodologias para resolução do problema em questão, mais especificamente os métodos baseados na decomposição, e resolvermos problemas de dimensões e características distintas, procurando ressaltar as qualidades de cada metodologia para cada caso.

APÊNDICE

Apêndice A

Forma de Armazenamento da Base

A forma de armazenamento da Base para cada uma das iterações do método tem uma característica importante: não é necessário armazenar todos os elementos da base. Se tivéssemos que guardar todos os elementos da base, não teríamos espaço suficiente na memória do computador.

A *fig. A.1* ilustra a forma estrutural de uma base.

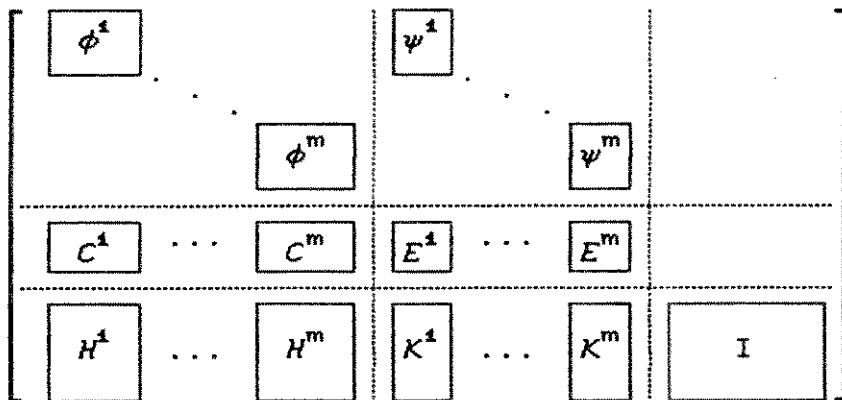


fig. A.1 - Estrutura de uma base

Apenas os blocos da *fig.A.1* são armazenados. Cada um desses blocos têm características que são exploradas para representá-los da forma mais compacta possível na memória do computador.

As principais características dos blocos são a grande quantidade de elementos iguais a zero, linhas e colunas com no máximo um elemento diferente de zero.

Apresentamos a seguir como cada um dos blocos da base são armazenados no computador.

Blocos ϕ^i , $i=1, \dots, m$

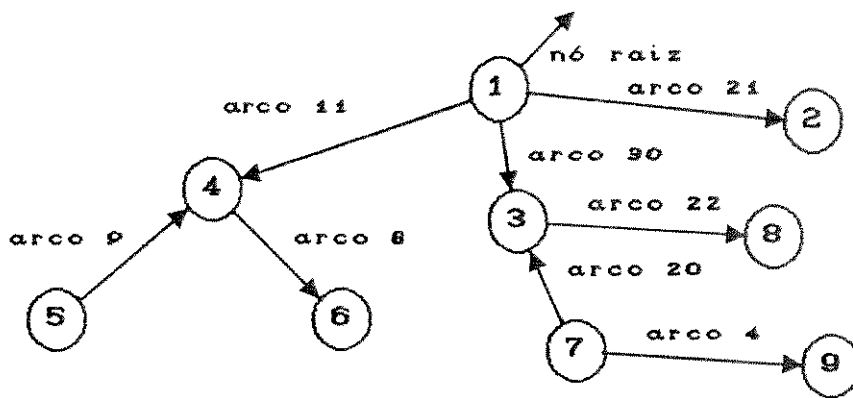
O bloco ϕ^i representa uma árvore enraizada para o i -ésimo produto a ser transportado. Utilizamos apenas três índices para caracterizar uma árvore, são eles FUX, KLP e PIA.

O índice FUX armazena todos os nós da árvore em pré-ordem a começar pelo nó raiz.

O índice PIA está associado a cada um dos nós da árvore e armazena o primeiro nó do caminho do nó em questão ao nó raiz.

O índice KLP também está associado a cada um dos nós da árvore e armazena o arco que liga o nó em questão com o nó PIA desse nó. Associamos um sinal a esse arco para indicar a sua orientação: + se ele tem origem no seu nó PIA e - caso contrário.

A *fig.A.2* ilustra uma árvore enraizada e a sua respectiva caracterização nos índices FUX, PIA e KLP.



FUX[1]= 1	PIA[1]= 0	KLP[1]= 0
FUX[2]= 4	PIA[4]= 1	KLP[4]= 11
FUX[3]= 5	PIA[5]= 4	KLP[5]= -9
FUX[4]= 6	PIA[6]= 4	KLP[6]= 8
FUX[5]= 3	PIA[3]= 1	KLP[3]= 30
FUX[6]= 7	PIA[7]= 3	KLP[7]= -20
FUX[7]= 9	PIA[9]= 7	KLP[9]= 4
FUX[8]= 8	PIA[8]= 3	KLP[8]= 22
FUX[9]= 2	PIA[2]= 1	KLP[2]= 21

fig.A.2 - Árvore enraizada e sua representação por índices

Notar que todos os arcos da árvore estão armazenados nos índices KLP e se desejarmos verificar se um determinado arco está na árvore basta verificar nos índices KLP.

Os índices FUX permitem obter todos os arcos que estão na árvore de forma ordenada. Essa ordenação de arcos é importante para a identificação dos elementos dos blocos C^i e H^i e é utilizada como referência de colunas.

Ocupamos apenas $1/3$ da memória que seria utilizada para armazenar toda a matriz nó-arco incidência de cada produto quando usamos essa estrutura de índices.

Blocos ψ^i , $i=1, \dots, m$

O bloco ψ^i representa os arcos corretores do produto i . Em vez de guardar todos os elementos da matriz ψ^i guardamos apenas os arcos corretores do produto i . Essa informação é suficiente para

caracterizar a matriz ψ^i .

A cada produto i associamos um índice $i\delta$ que guarda o número de arcos corretores do produto i e o índice $\delta^i[j]$ que guarda o j -ésimo arco corretor do produto i .

Bloco C^i , $i=1, \dots, m$

O bloco C^i armazena os elementos das equações ativas referentes aos arcos que estão na árvore do produto i . Trabalhamos com dois tipos diferentes de equações ativas, de trecho e de carga.

As matrizes C^i têm poucos elementos diferentes de zero e esses são todos iguais a um. Guardamos nos índices $i\omega$ e $i\gamma$ o número de equações de trecho e carga ativas, respectivamente. E nos índices $\omega^i[j]$, $j=1, \dots, i\omega$ e $\gamma^i[k]$, $k=1, \dots, i\gamma$ as equações de trecho e carga que estão ativas.

Com esses novos índices e com os índices FUX e KLP de cada árvore, as matrizes C^i estão caracterizadas.

Para identificar os valores dos elementos de uma linha da matriz C^i referente à j -ésima equação de trecho ativa, devemos comparar o índice $\omega^i[j]$ com os arcos da árvore do produto i , utilizando os índices FUX e KLP para obter os arcos de maneira ordenada. Caso encontre um arco que pertença a equação de trecho ativa, a posição desse elemento recebe o valor 1, caso contrário recebe o valor zero. É importante lembrar que essa linha tem no máximo um elemento diferente de zero, pois uma árvore não pode ter dois arcos iguais.

Para identificar os valores dos elementos de uma linha da matriz C^i referente à k -ésima equação de carga ativa, procedemos de uma forma um pouco mais complexa. Primeiro, deve-se verificar se o produto i pertence a equação de carga $\gamma^i[k]$. Caso o produto i não pertença a essa equação, todos os elementos dessa linha são zero. Caso contrário, verificar quais são os arcos marcados da árvore do produto i , para associar-lhes o valor 1 em suas posições e zero nos demais.

Dessa forma, as matrizes C^i são guardadas e lidas na memória do computador.

Blocos E^i , $i=1, \dots, m$

Nenhum índice novo precisa ser criado para armazenar os elementos das matrizes E^i , apenas com os índices anteriores já temos os seus elementos representados.

Os índices de equações ativas são os mesmos do que os definidos para as matrizes C^i .

Para identificar os valores de uma linha da matriz E^i , procedemos de maneira análoga à forma de identificação da matriz C^i , só que agora utilizamos os arcos corretores ao invés dos arcos das árvores.

A identificação da j -ésima linha de equação de trecho ativa da matriz E^i é feita verificando-se entre os arcos corretores do produto i , qual está na equação de trecho ativa para associar-lhe o valor 1 e zero nos demais. No máximo, a linha terá apenas um elemento diferente de zero e igual a 1, pois um produto não pode ter dois arcos corretores iguais.

Para identificar os elementos da k -ésima linha de carga ativa da matriz E^i , verificamos se o produto i é levado pela equação de carga $gama[k]$. Se o for, associe o valor 1 aos arcos corretores marcados do produto i e zero aos não marcados. Caso contrário, associe zero a todos os elementos dessa linha.

Bloco H^i , $i=1, \dots, m$

Dois novos índices são definidos para se poder representar as matrizes H^i . Esses dois índices são $nueqtr$ e $nueqca$. Eles guardam o número de equações de trecho e carga folgadas respectivamente. Precisamos de mais dois conjuntos de índices para guardar cada uma das equações folgadas: os índices $eqtr[j]$, $j=1, \dots, nueqtr$ e os índices $eqca[k]$, $k=1, \dots, nueqca$, que

representam respectivamente as equações de trecho e carga folgadas.

Para identificar elementos nas matrizes H^i , procedemos como no caso dos elementos das matrizes C^i : utilizamos os índices FUX e KLP de cada produto e em vez de se utilizar os índices das equações ativas, utilizamos os índices das equações folgadas. Todas as observações feitas para as Matrizes C^i continuam válidas.

Blocos K^i , $i=1, \dots, m$

Também aqui não é preciso definir mais nenhum índice, com os já definidos, as matrizes H^i estão representadas.

Para fazer a identificação dos elementos dessas matrizes, procedemos como no caso das matrizes E^i , só que agora utilizamos os índices das equações de trecho e carga folgadas.

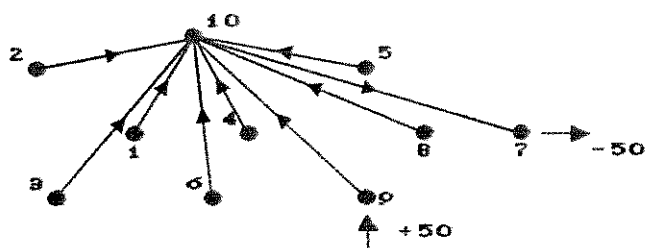
Bloco I

Esse bloco fica representado pelo conjunto de índices das equações de trecho e carga folgadas.

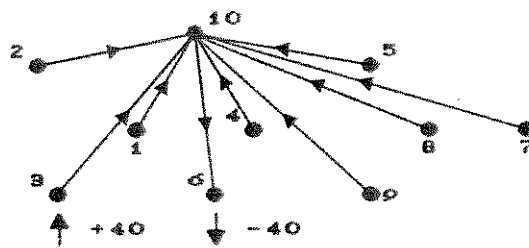
Apêndice B

fig.B.1 - Árvores iniciais geradas pelo programa Multipro para o exemplo 1, capítulo 5

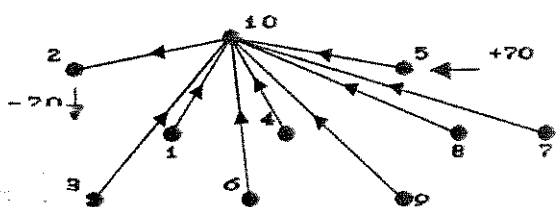
Produto 1



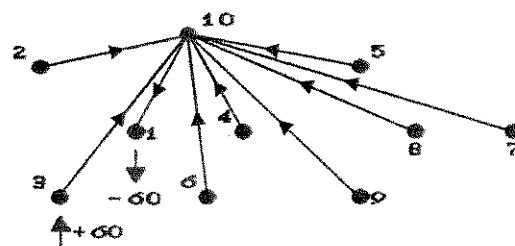
Produto 2



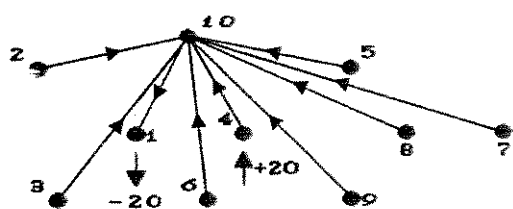
Produto 3



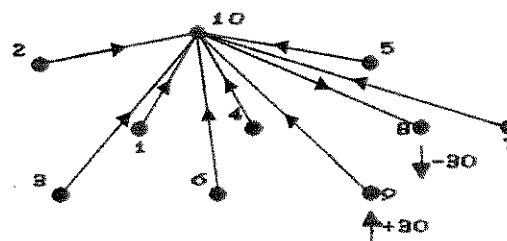
Produto 4



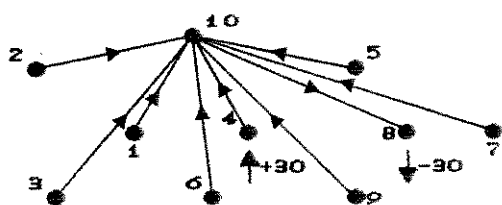
Produto 5



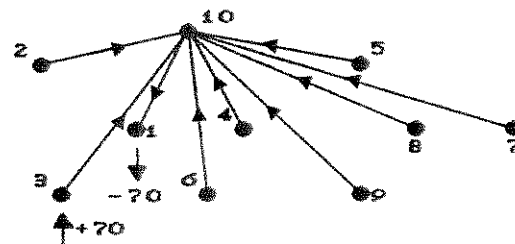
Produto 6



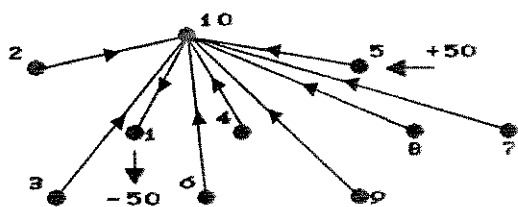
Produto 7



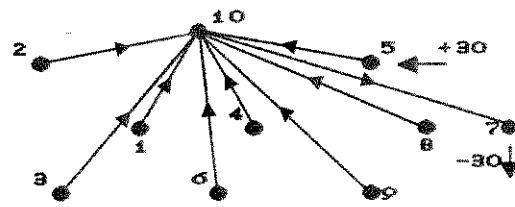
Produto 8



Produto 9

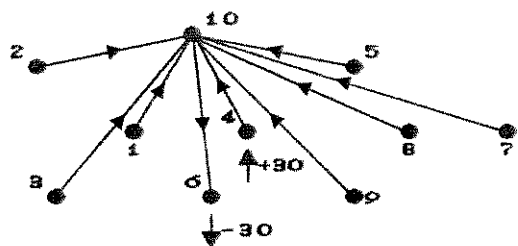


Produto 10

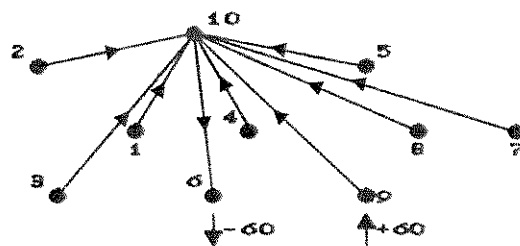


continuação da fig. B.1

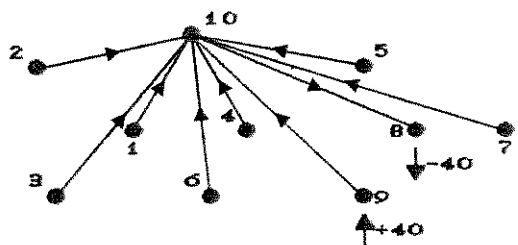
Produto 11



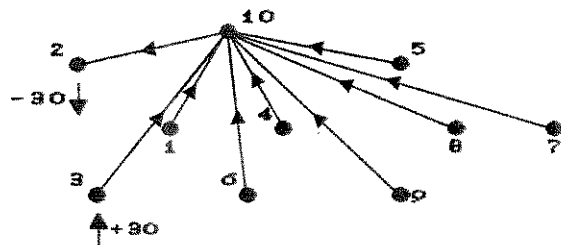
Produto 12



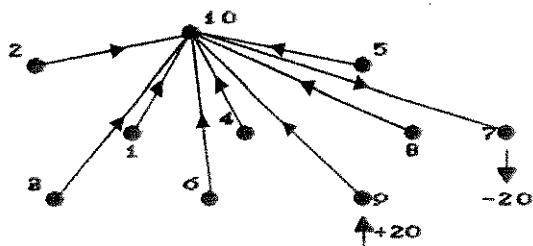
Produto 13



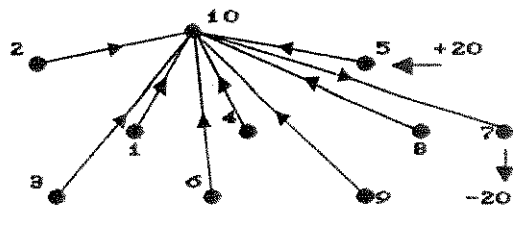
Produto 14



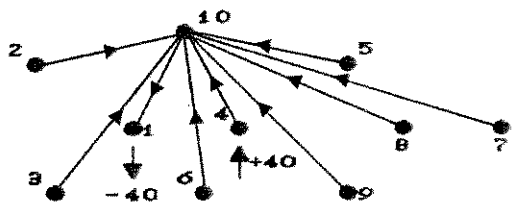
Produto 15



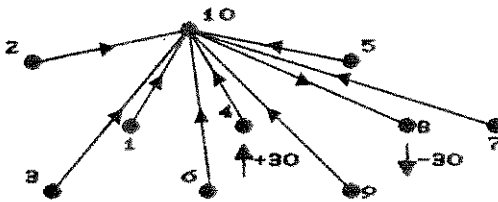
Produto 16



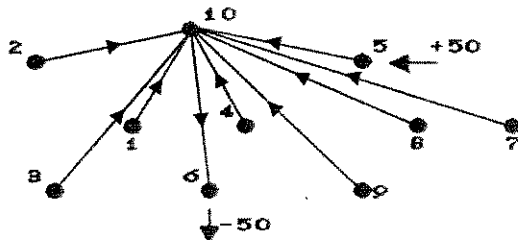
Produto 17



Produto 18



Produto 19



Produto 20

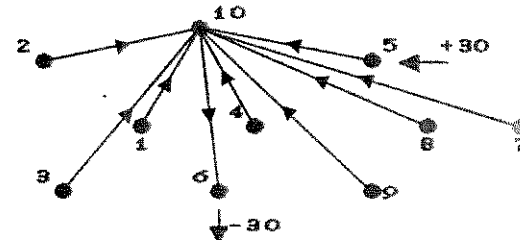
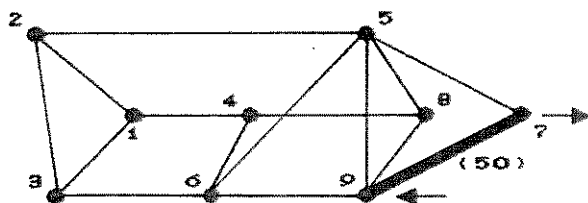


fig.B.2 - Solução ótima obtida pelo programa Multipro para o exemplo 1, capítulo 5

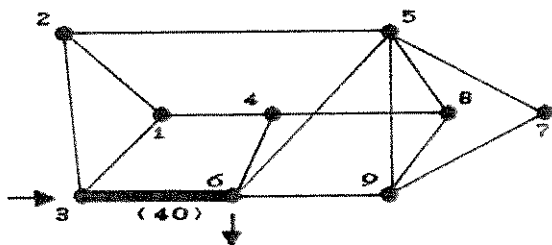
Produto 1

nó origem: 9
 nó destino: 7
 carga transportada: 50
 carga não transportada: 0
 custo: 300
 tipo de vagão: 1



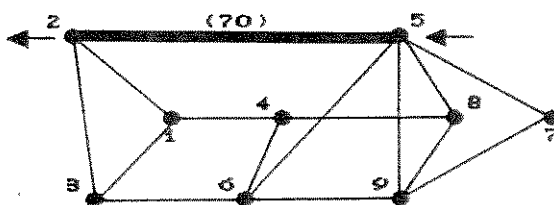
Produto 2

nó origem: 3
 nó destino: 6
 carga transportada: 40
 carga não transportada: 0
 custo: 120
 tipo de vagão: 1



Produto 3

nó origem: 5
 nó destino: 2
 carga transportada: 70
 carga não transportada: 0
 custo: 280
 tipo de vagão: 1

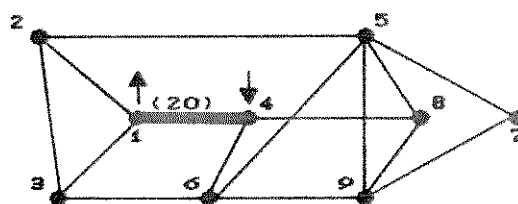
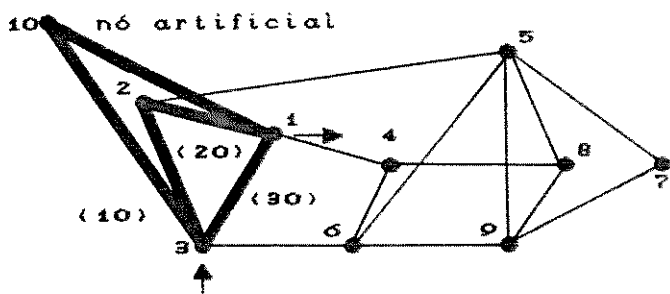


Produto 4

nó origem: 3
 nó destino: 1
 carga transportada: 50
 carga não transportada: 10
 custo: 180
 tipo de vagão: 1

Produto 5

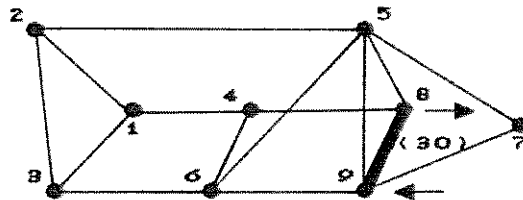
nó origem: 4
 nó destino: 1
 carga transportada: 20
 carga não transportada: 0
 custo: 120
 tipo de vagão: 1



Continuação dos resultados do exemplo 1 programa Multipro

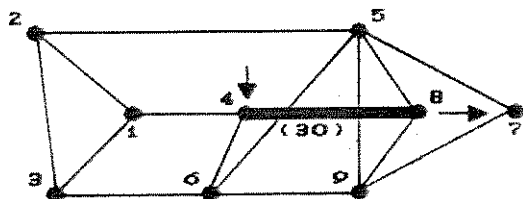
Produto 6

nó origem: 9
 nó destino: 8
 carga transportada: 30
 carga não transportada: 0
 custo: 210
 tipo de vagão: 1



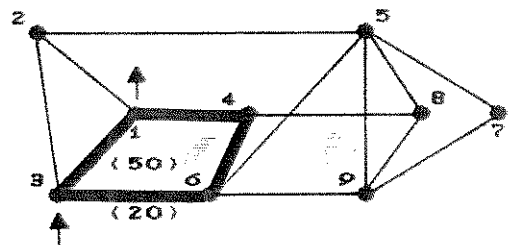
Produto 7

nó origem: 4
 nó destino: 8
 carga transportada: 30
 carga não transportada: 0
 custo: 240
 tipo de vagão: 1



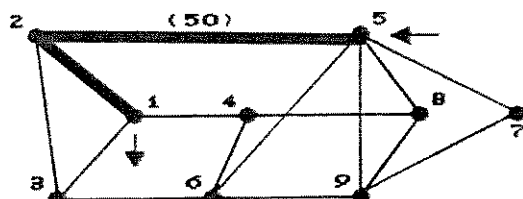
Produto 8

nó origem: 3
 nó destino: 1
 carga transportada: 70
 carga não transportada: 0
 custo: 630
 tipo de vagão: 2



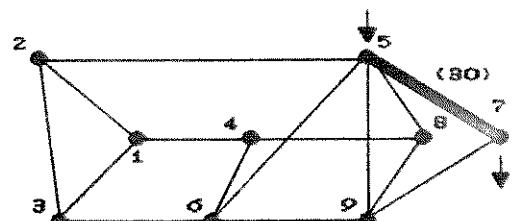
Produto 9

nó origem: 5
 nó destino: 1
 carga transportada: 50
 carga não transportada: 0
 custo: 300
 tipo de vagão: 2



Produto 10

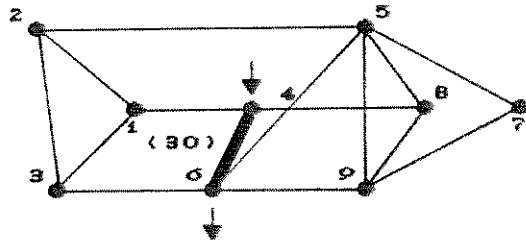
nó origem: 5
 nó destino: 7
 carga transportada: 30
 carga não transportada: 0
 custo: 180
 tipo de vagão: 2



Continuação dos resultados do exemplo 1 programa Multipro

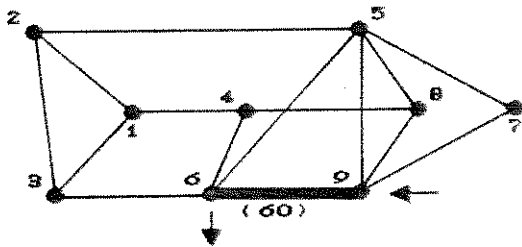
Produto 11

nó origem: 4
 nó destino: 6
 carga transportada: 30
 carga não transportada: 0
 custo: 30
 tipo de vagão: 2



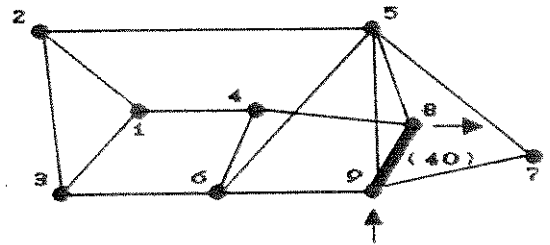
Produto 12

nó origem: 9
 nó destino: 6
 carga transportada: 60
 carga não transportada: 0
 custo: 300
 tipo de vagão: 2



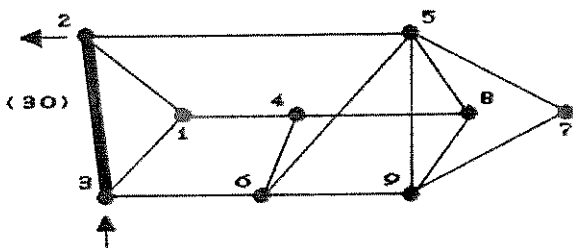
Produto 13

nó origem: 9
 nó destino: 8
 carga transportada: 40
 carga não transportada: 0
 custo: 320
 tipo de vagão: 2



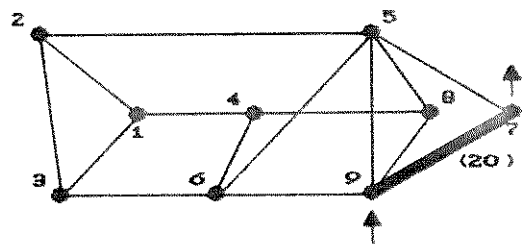
Produto 14

nó origem: 3
 nó destino: 2
 carga transportada: 30
 carga não transportada: 0
 custo: 90
 tipo de vagão: 2



Produto 15

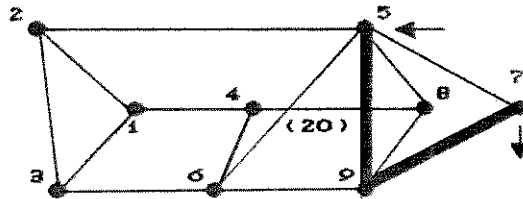
nó origem: 9
 nó destino: 7
 carga transportada: 20
 carga não transportada: 0
 custo: 100
 tipo de vagão: 2



Continuação dos resultados do exemplo 1 programa Multipro

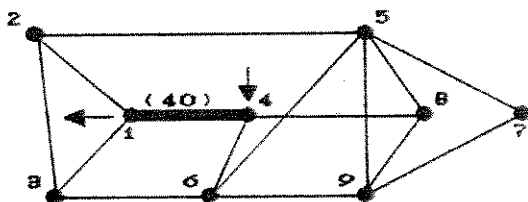
Produto 16

nó origem:5
 nó destino:7
 carga transportada:20
 carga não transportada:0
 custo:60
 tipo de vagão:2



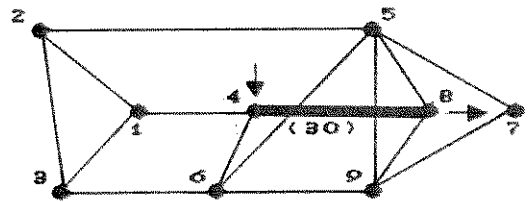
Produto 17

nó origem:4
 nó destino:1
 carga transportada:40
 carga não transportada:0
 custo:120
 tipo de vagão:2



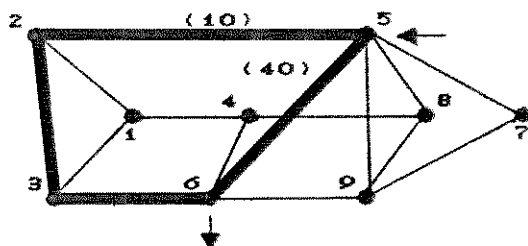
Produto 18

nó origem:4
 nó destino:8
 carga transportada:30
 carga não transportada:0
 custo:270
 tipo de vagão:2



Produto 19

nó origem:5
 nó destino:6
 carga transportada:50
 carga não transportada:0
 custo:250
 tipo de vagão:2



Produto 20

nó origem:5
 nó destino:8
 carga transportada:30
 carga não transportada:0
 custo:60
 tipo de vagão:2

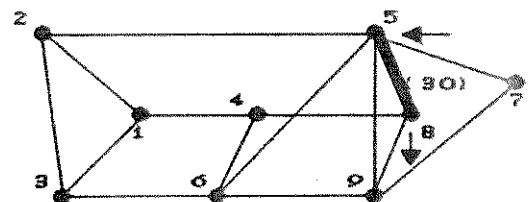
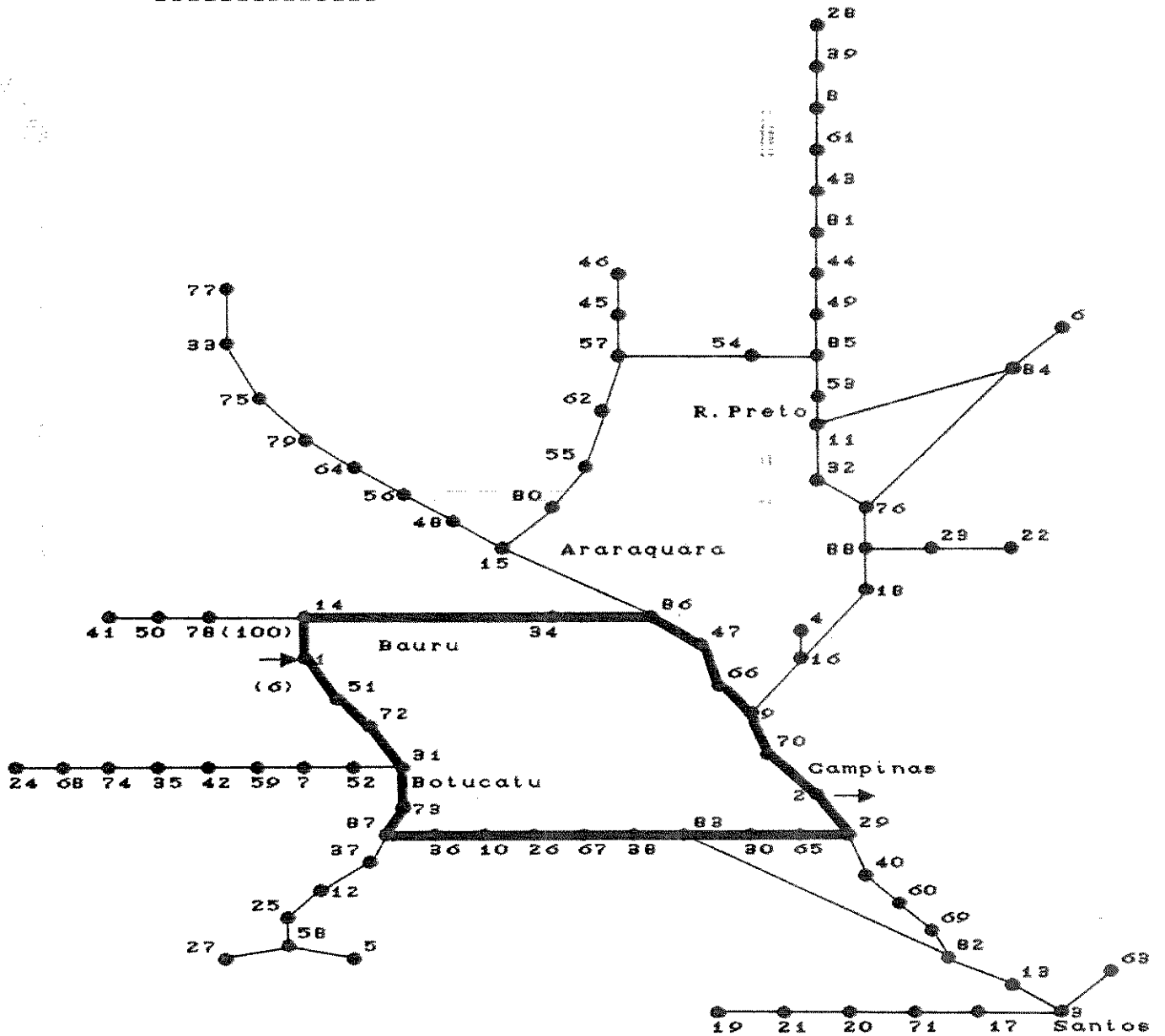


fig.B.3 - Solução ótima do exemplo 2 , capítulo 5 , utilizando o programa Multipro

Produto 1

nó origem : 1
 nó destino: 2
 carga transportada : 100
 carga não transportada : 0
 custo : 27168
 tipo de vagão : 1

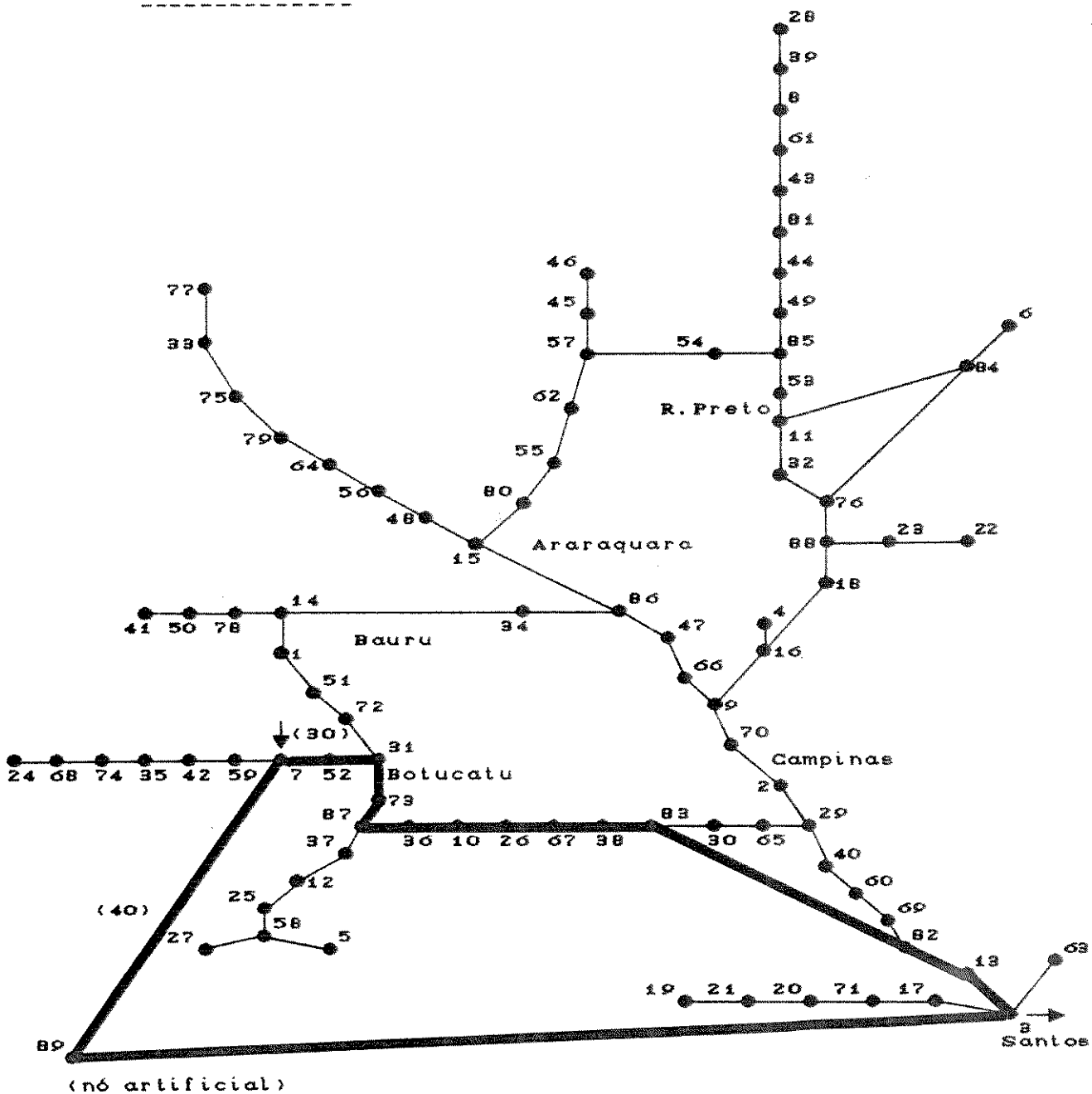
Melhor caminho



Produto 2

nó origem : 7
 nó destino: 8
 carga transportada : 30
 carga não transportada : 40
 custo : 14970
 tipo de vagão : 1

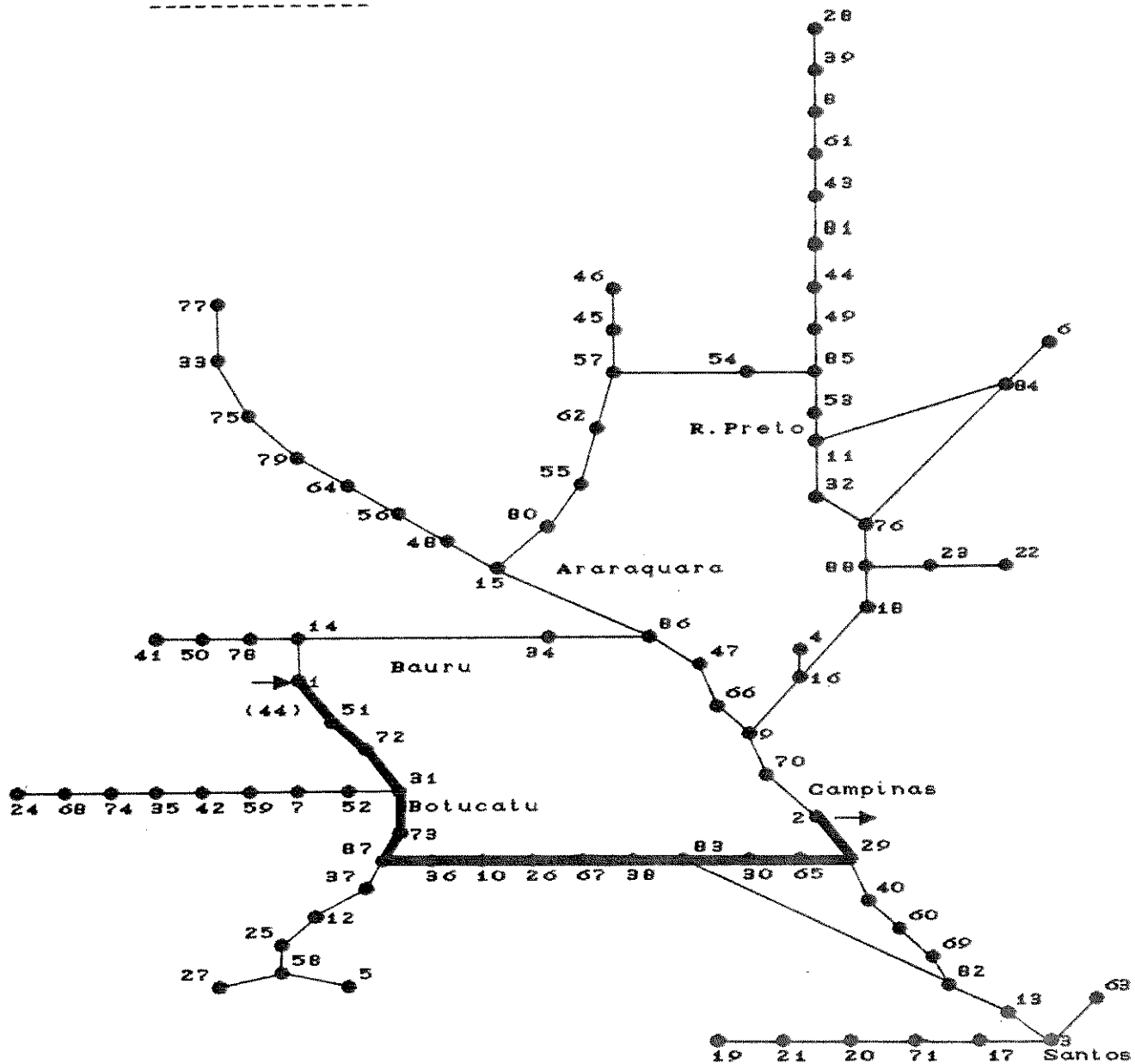
Melhor caminho



Produto 3

nó origem : 1
 nó destino: 2
 carga transportada : 44
 carga não transportada : 0
 custo : 17160
 tipo de vagão : 1

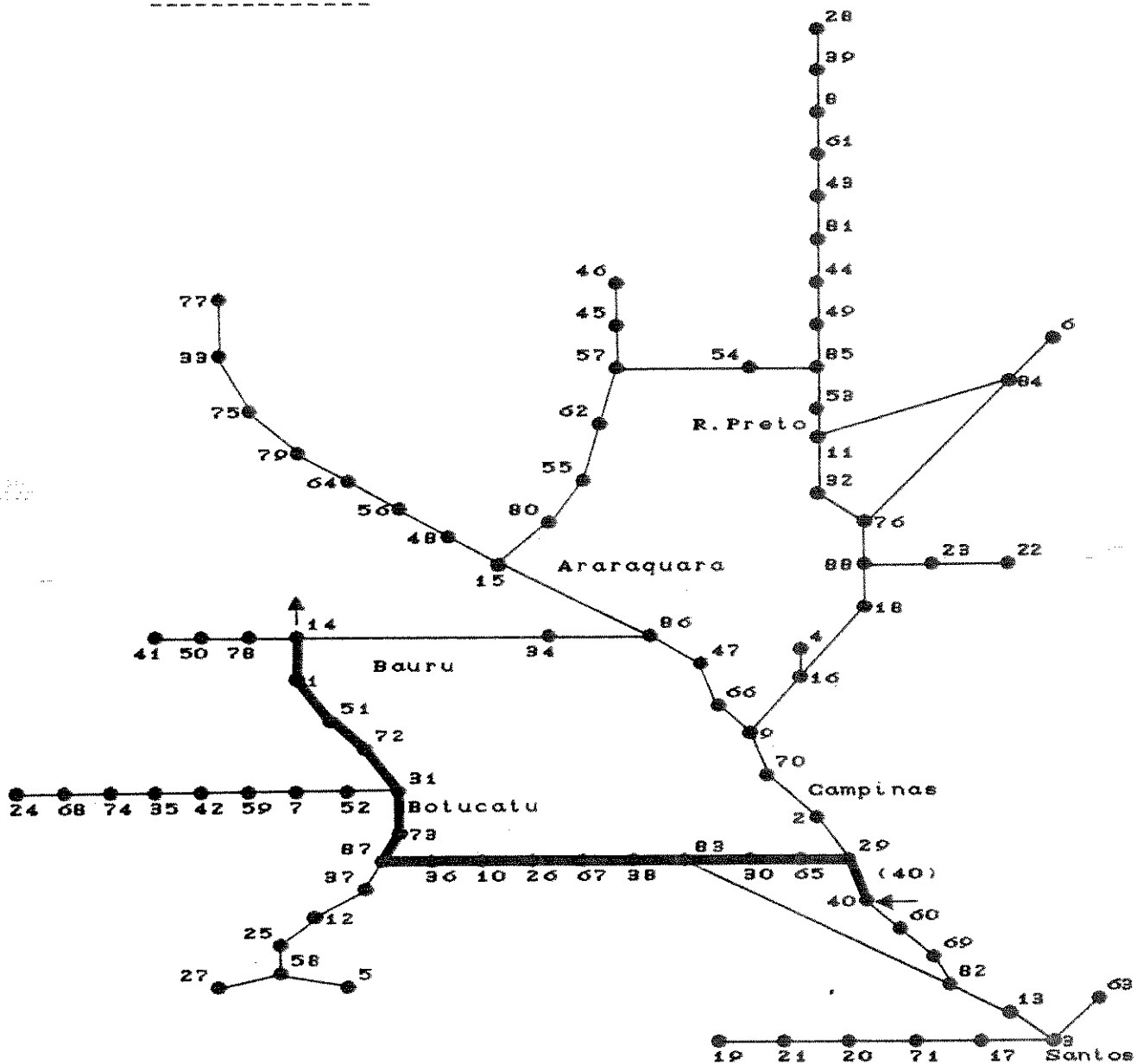
Melhor caminho



Produto 4

nó origem : 40
 nó destino: 14
 carga transportada : 40
 carga não transportada : 0
 custo : 19400
 tipo de vagão : 2

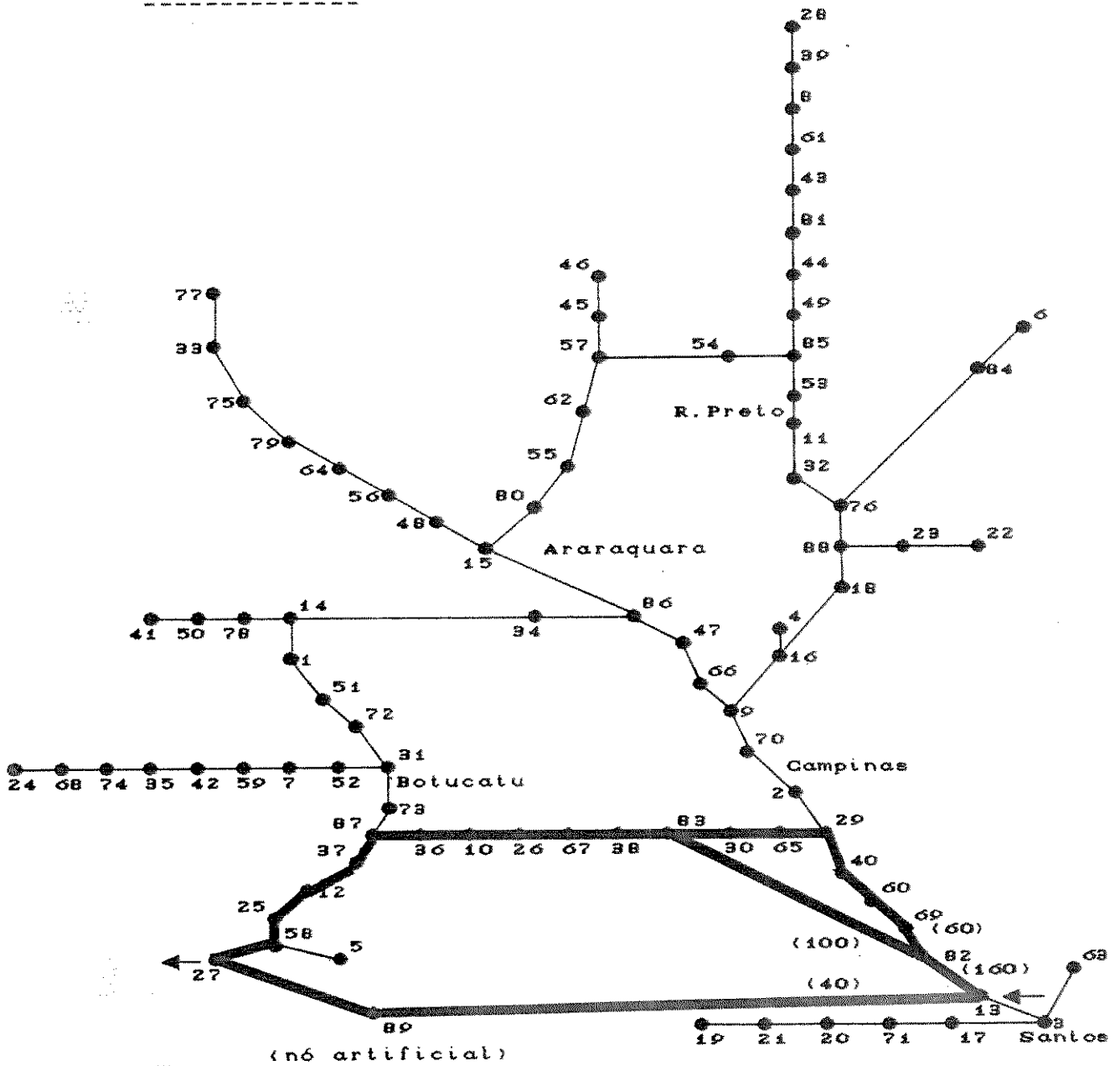
Melhor caminho



Produto 5

nó origem : 13
 nó destino: 27
 carga transportada : 160
 carga não transportada : 40
 custo : 73780
 tipo de vagão : 2

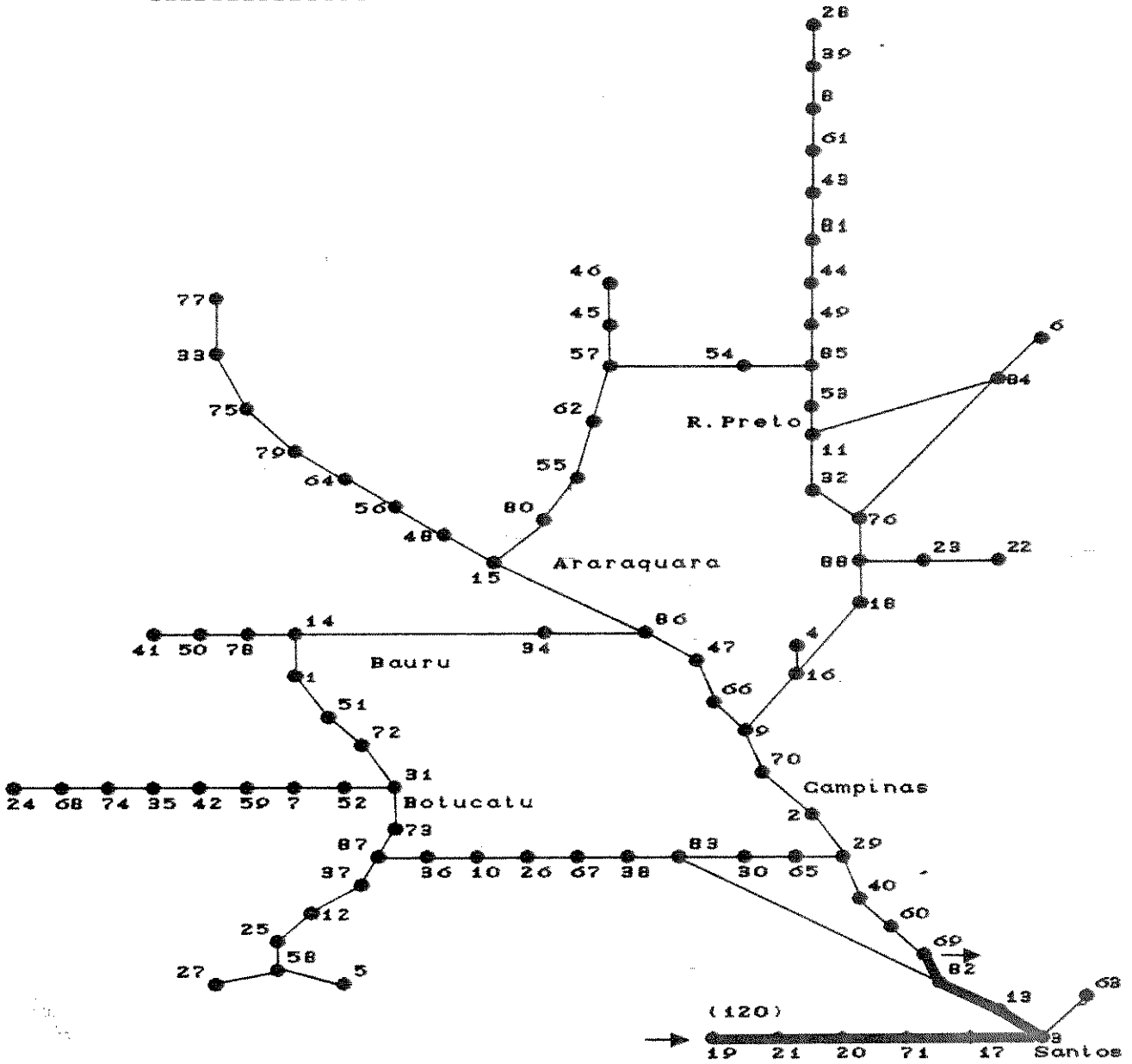
Melhor caminho



Produto 6

nó origem : 19
 nó destino: 69
 carga transportada : 120
 carga não transportada : 0
 custo : 40920
 tipo de vagão : 2

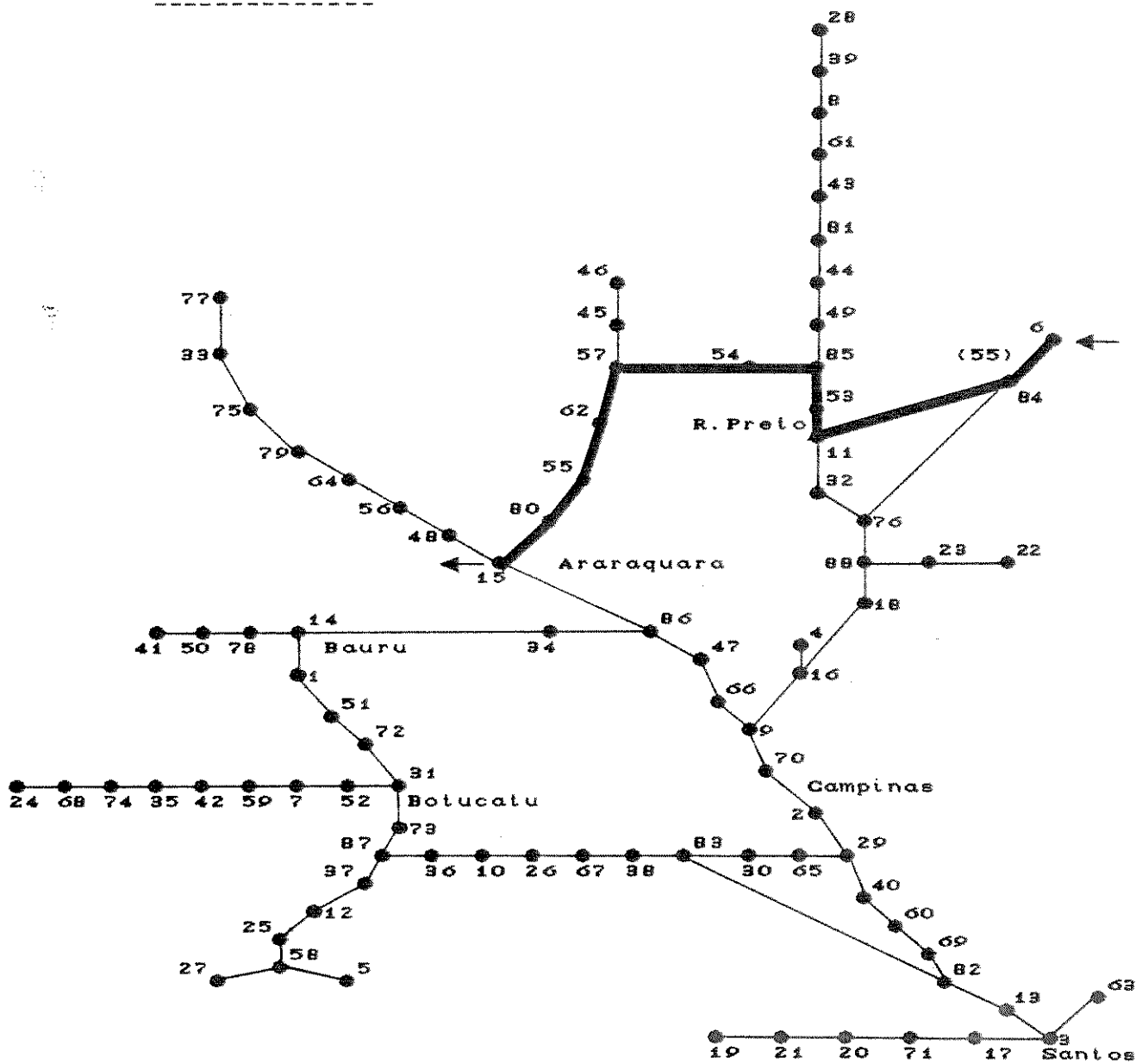
Melhor caminho



Produto 7

nó origem : 6
 nó destino: 15
 carga transportada : 55
 carga não transportada : 0
 custo : 18480
 tipo de vagão : 9

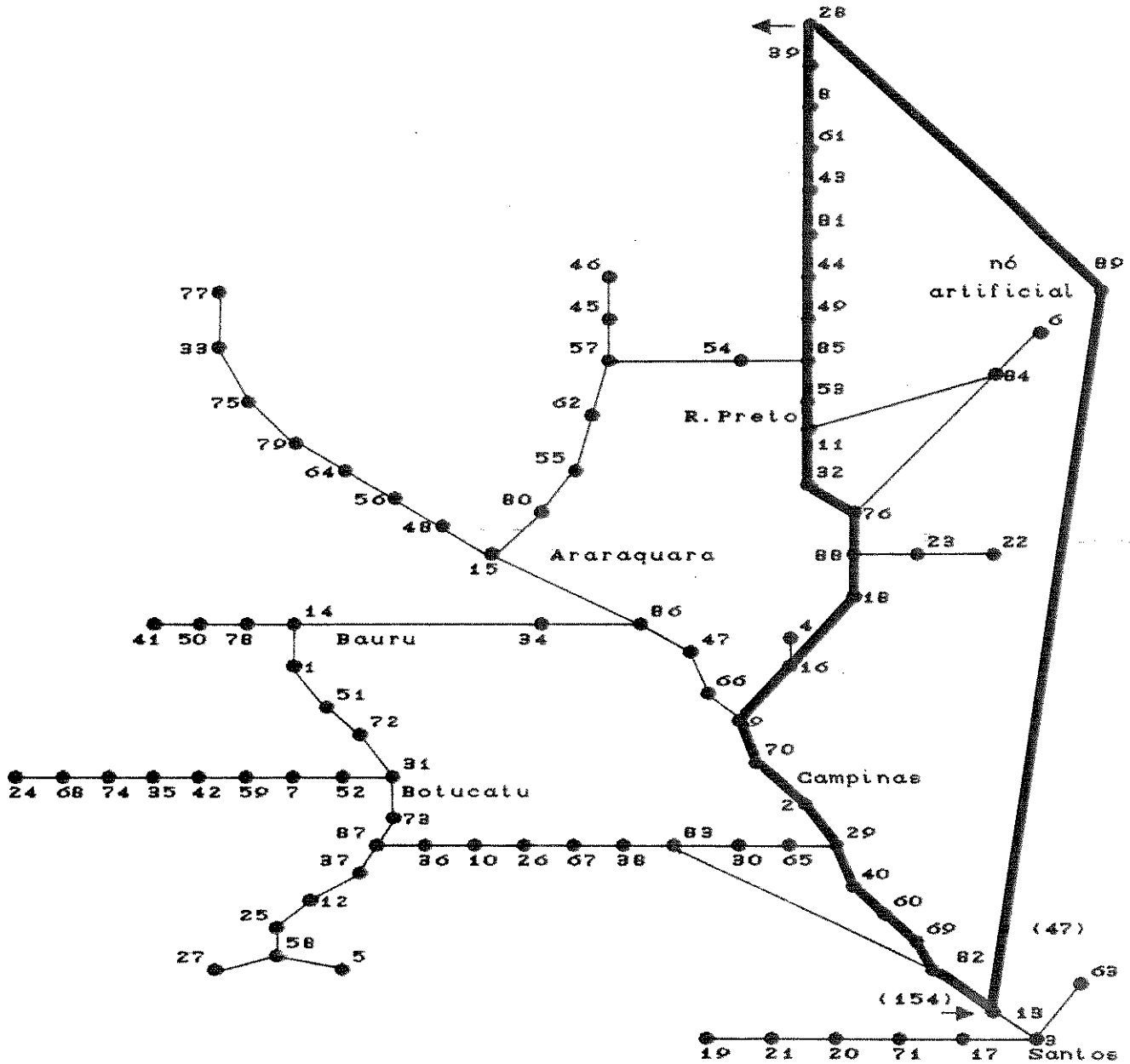
Melhor caminho



Produto 8

nó origem : 18
 nó destino: 28
 carga transportada : 154
 carga não transportada : 47
 custo : 130900
 tipo de vagão : 3

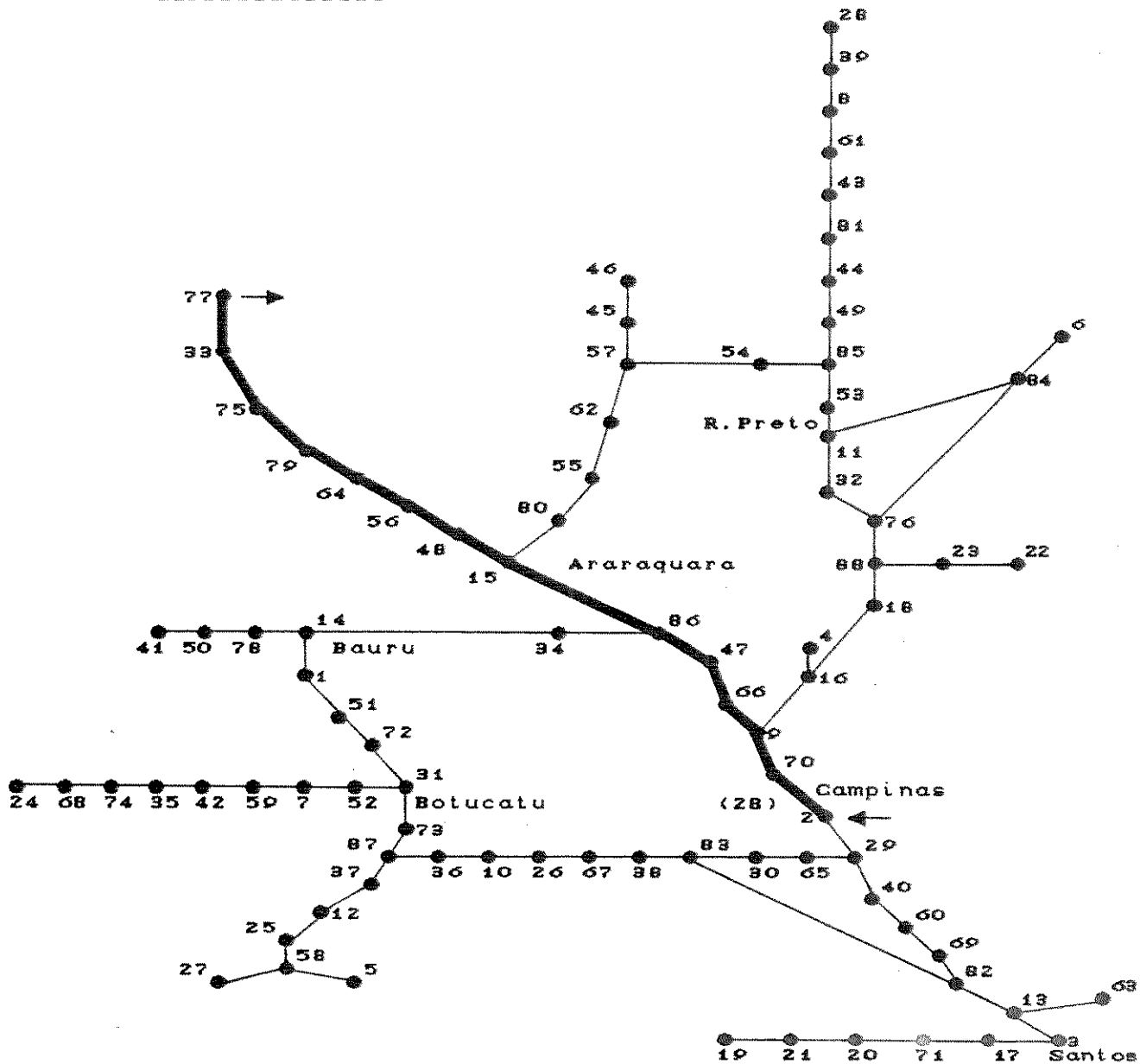
Melhor caminho



Produto 9

nó origem : 2
 nó destino: 77
 carga transportada : 28
 carga não transportada : 0
 custo : 18004
 tipo de vagão : 4

Melhor caminho



Produto 10

nó origem : 2
 nó destino: 80
 carga transportada : 18
 carga não transportada : 0
 custo : 5670
 tipo de vagão : 4

Melhor caminho

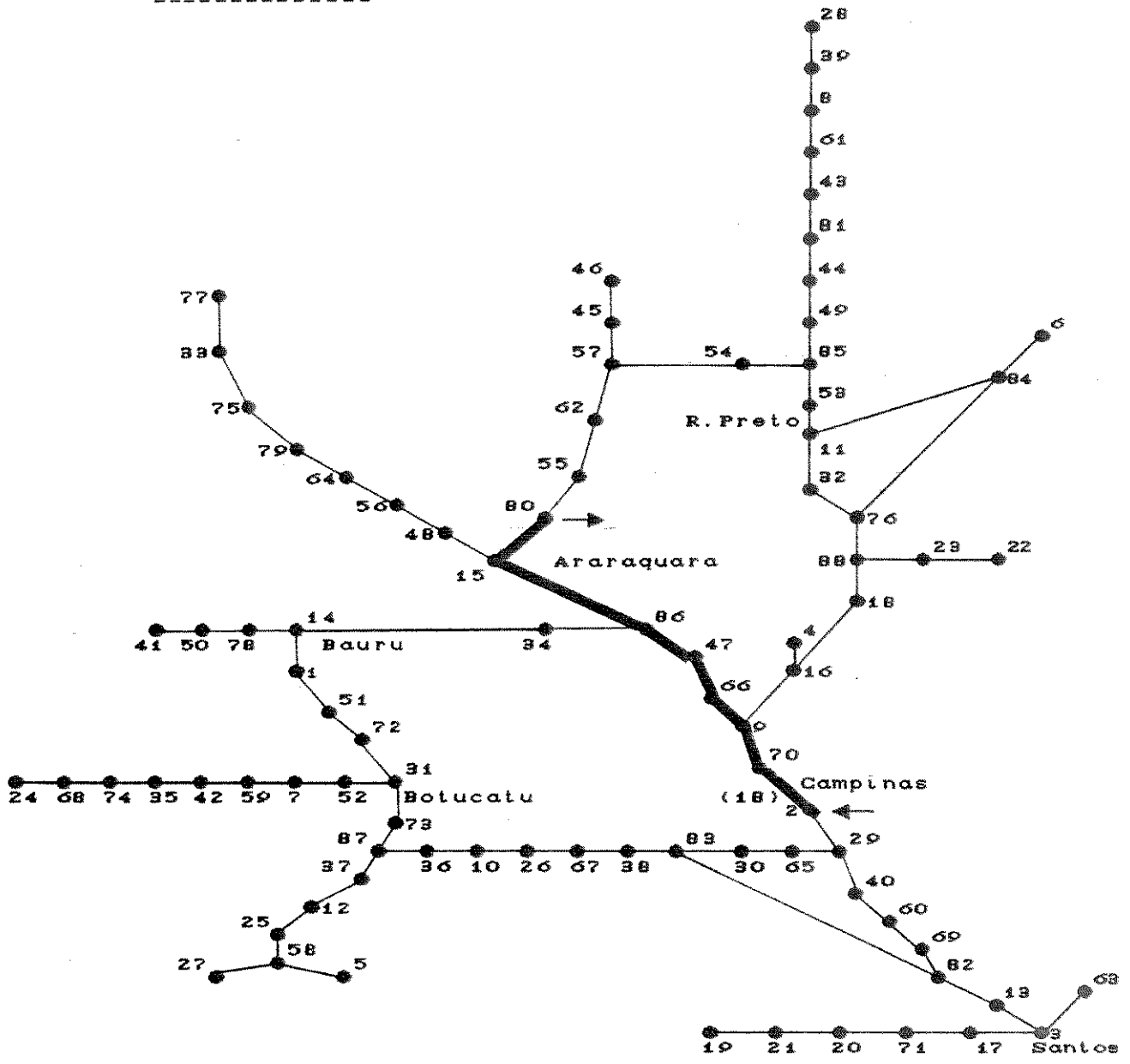
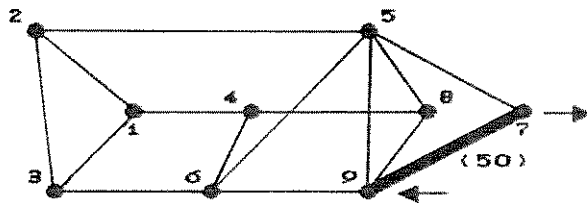


fig.B.4 - Solução ótima do exemplo 1, capítulo 5, utilizando o programa Hiperlindo

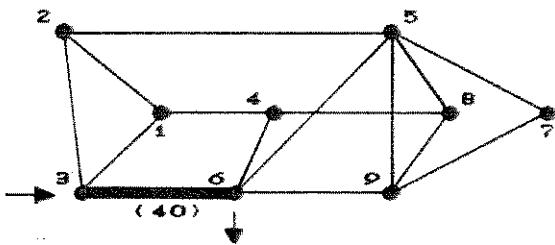
Produto 1

nó origem: 9
 nó destino: 7
 carga transportada: 50
 carga não transportada: 0
 custo: 300
 tipo de vagão: 1



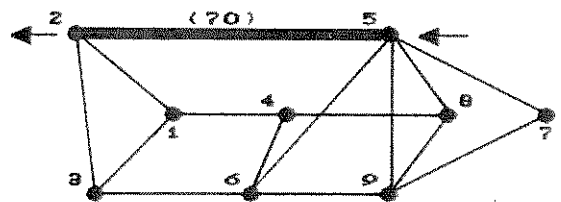
Produto 2

nó origem: 3
 nó destino: 6
 carga transportada: 40
 carga não transportada: 0
 custo: 120
 tipo de vagão: 1



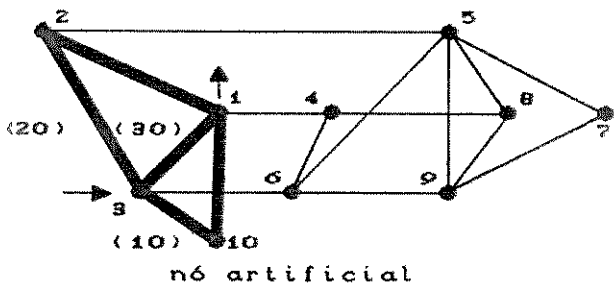
Produto 3

nó origem: 5
 nó destino: 2
 carga transportada: 70
 carga não transportada: 0
 custo: 280
 tipo de vagão: 1



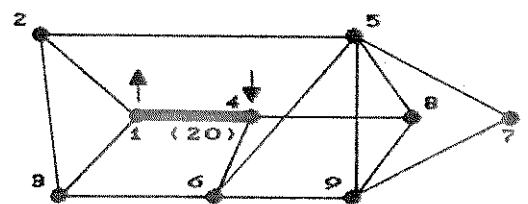
Produto 4

nó origem: 3
 nó destino: 1
 carga transportada: 50
 carga não transportada: 10
 custo: 130
 tipo de vagão: 1



Produto 5

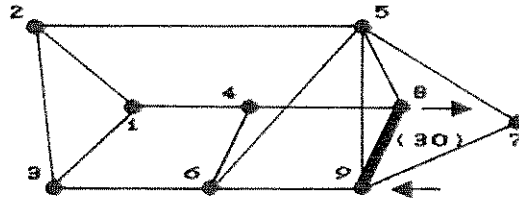
nó origem: 4
 nó destino: 1
 carga transportada: 20
 carga não transportada: 0
 custo: 120
 tipo de vagão: 1



Continuação dos resultados do exemplo 1 programa Hiperlindo

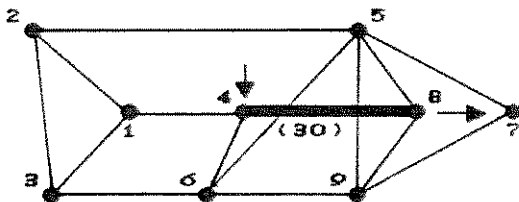
Produto 6

nó origem: 9
 nó destino: 8
 carga transportada: 30
 carga não transportada: 0
 custo: 210
 tipo de vagão: 1



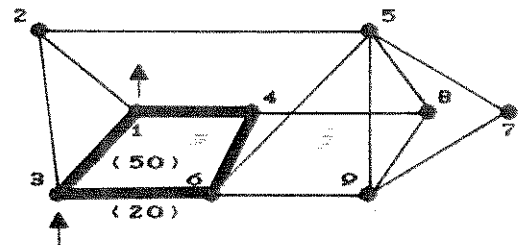
Produto 7

nó origem: 4
 nó destino: 8
 carga transportada: 30
 carga não transportada: 0
 custo: 240
 tipo de vagão: 1



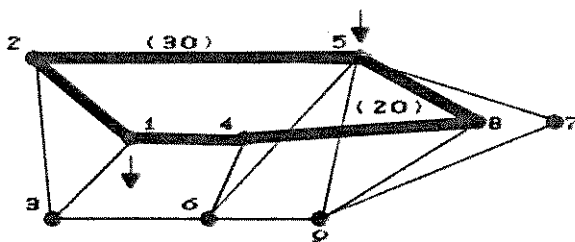
Produto 8

nó origem: 3
 nó destino: 1
 carga transportada: 70
 carga não transportada: 0
 custo: 630
 tipo de vagão: 2



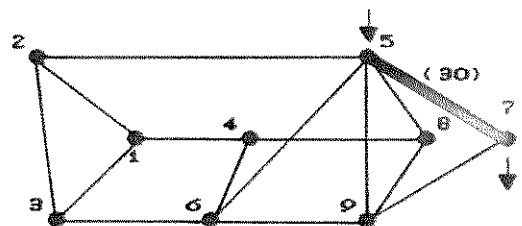
Produto 9

nó origem: 5
 nó destino: 1
 carga transportada: 50
 carga não transportada: 0
 custo: 300
 tipo de vagão: 2



Produto 10

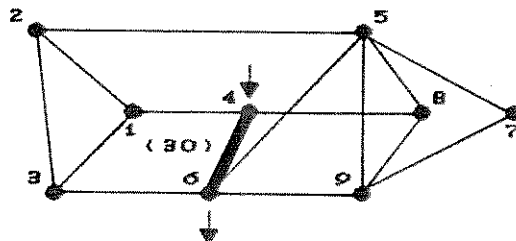
nó origem: 5
 nó destino: 7
 carga transportada: 30
 carga não transportada: 0
 custo: 180
 tipo de vagão: 2



Continuação dos resultados do exemplo 1 programa Hiperlindo

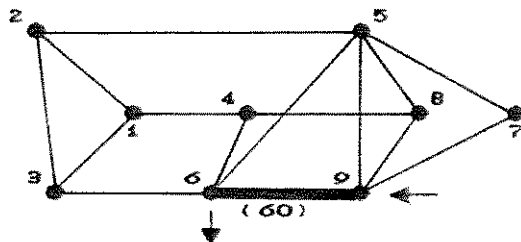
Produto 11

nó origem: 4
 nó destino: 6
 carga transportada: 30
 carga não transportada: 0
 custo: 90
 tipo de vagão: 2



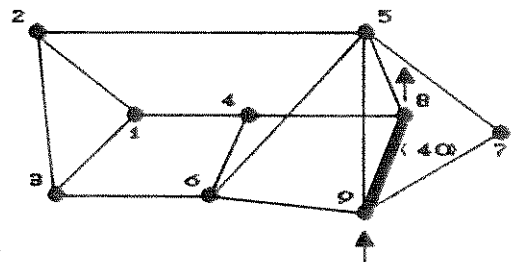
Produto 12

nó origem: 9
 nó destino: 6
 carga transportada: 60
 carga não transportada: 0
 custo: 300
 tipo de vagão: 2



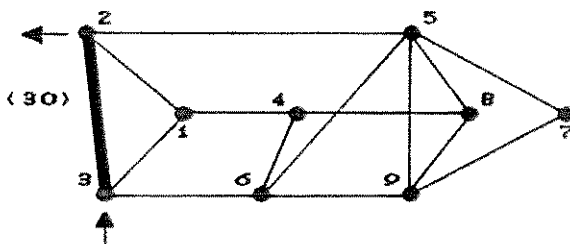
Produto 13

nó origem: 9
 nó destino: 8
 carga transportada: 40
 carga não transportada: 0
 custo: 320
 tipo de vagão: 2



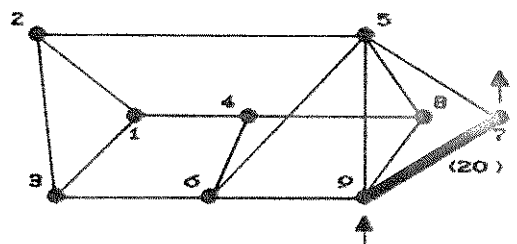
Produto 14

nó origem: 3
 nó destino: 2
 carga transportada: 30
 carga não transportada: 0
 custo: 90
 tipo de vagão: 2



Produto 15

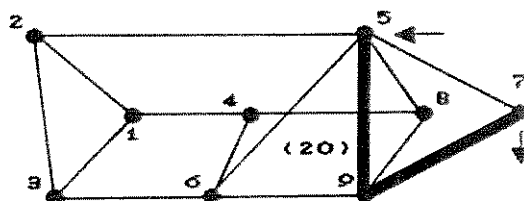
nó origem: 9
 nó destino: 7
 carga transportada: 20
 carga não transportada: 0
 custo: 100
 tipo de vagão: 2



Continuação dos resultados do exemplo 1 programa Hiperlindo

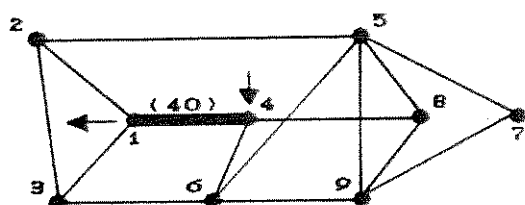
Produto 16

nó origem:5
 nó destino:7
 carga transportada:20
 carga não transportada:0
 custo:60
 tipo de vagão:2



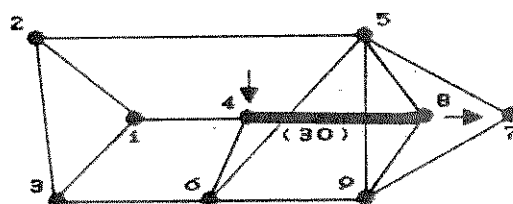
Produto 17

nó origem:4
 nó destino:1
 carga transportada:40
 carga não transportada:0
 custo:120
 tipo de vagão:2



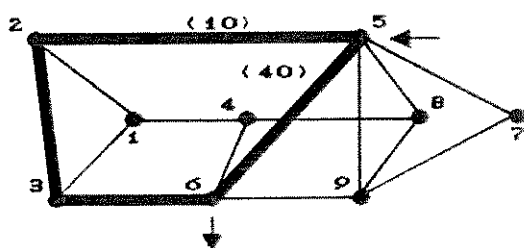
Produto 18

nó origem:4
 nó destino:8
 carga transportada:30
 carga não transportada:0
 custo:270
 tipo de vagão:2



Produto 19

nó origem:5
 nó destino:6
 carga transportada:50
 carga não transportada:0
 custo:250
 tipo de vagão:2



Produto 20

nó origem:5
 nó destino:8
 carga transportada:30
 carga não transportada:0
 custo:60
 tipo de vagão:2

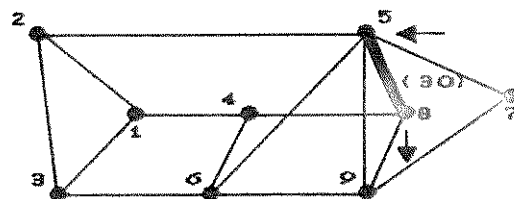


Tabela B.1 - Dados dos arcos do Exemplo 2, capítulo 5

n ^o arco	n ^o origem	n ^o dest. (tons.)	capac. (tons.)	custos dos produtos (\$)										
				1	2	3	4	5	6	7	8	9	10	
1	1	14	10 ⁴	1	1	1	1	1	1	1	1	1	1	1
2	1	51	10 ⁴	10	13	21	21	21	10	21	10	20	21	
3	2	29	10 ⁴	51	45	57	56	56	47	57	45	57	56	
4	2	70	10 ⁴	100	78	88	101	102	102	102	103	104	103	
5	3	13	10 ⁴	1	1	1	1	1	1	1	1	1	1	1
6	3	17	10 ⁴	21	23	23	23	23	23	23	23	23	23	23
7	3	63	10 ⁴	0	0	0	0	0	0	0	0	0	0	0
8	4	16	10 ⁴	4	4	4	4	4	4	4	4	4	3	4
9	5	58	10 ⁴	60	56	65	56	65	58	76	45	56	76	
10	6	84	10 ⁴	61	54	54	54	60	61	45	61	45	61	
11	7	52	10 ⁴	30	31	27	32	31	32	32	32	32	32	32
12	7	59	10 ⁴	32	32	28	23	32	31	32	32	33	33	
13	8	39	10 ⁴	100	111	111	116	119	118	110	109	109	109	
14	8	61	10 ⁴	30	54	45	45	45	56	56	56	56	45	
15	9	16	10 ⁴	10	19	18	3	17	18	16	18	19	19	
16	9	66	10 ⁴	11	17	13	10	13	13	14	12	14	16	
17	9	70	10 ⁴	4	4	2	4	5	6	5	5	5	6	
18	10	26	10 ⁴	2	11	12	2	12	12	12	12	12	12	12
19	10	36	10 ⁴	7	11	12	7	10	11	10	12	12	12	12
20	11	32	10 ⁴	23	39	39	52	43	43	53	39	53	53	
21	11	53	10 ⁴	1	1	1	1	1	1	1	1	1	1	1
22	11	84	10 ⁴	77	121	121	121	120	123	123	123	127	124	
23	12	25	10 ⁴	120	123	121	123	132	132	132	130	132	130	
24	12	37	10 ⁴	33	36	34	32	32	32	28	32	34	32	
25	13	82	10 ⁴	45	65	65	21	61	65	54	65	60	65	
26	14	34	10 ⁴	33	63	43	54	63	59	63	63	61	61	
27	14	78	10 ⁴	100	123	123	123	121	123	123	123	123	123	
28	15	48	10 ⁴	18	41	32	42	42	42	32	42	42	34	
29	15	80	10 ⁴	19	32	21	32	32	32	32	34	34	34	
30	15	86	10 ⁴	61	80	77	67	85	86	67	87	86	65	
31	16	18	10 ⁴	43	48	45	43	45	49	45	43	50	34	
32	17	71	10 ⁴	65	72	74	54	70	74	71	54	73	75	
33	18	88	10 ⁴	43	39	34	34	40	43	43	43	42	32	
34	19	21	10 ⁴	23	42	43	44	43	43	45	45	45	21	
35	20	21	10 ⁴	21	32	29	30	32	21	32	32	30	21	
36	20	71	10 ⁴	18	76	80	59	78	76	76	80	78	56	
37	22	23	10 ⁴	9	9	9	9	9	9	9	9	9	9	
38	23	88	10 ⁴	63	64	58	54	57	54	63	54	63	54	
39	24	68	10 ⁴	23	23	102	102	101	103	102	100	88	100	
40	25	58	10 ⁴	1	1	1	1	1	1	1	1	1	1	
41	26	27	10 ⁴	3	8	7	9	8	9	2	9	9	9	
42	27	58	10 ⁴	65	65	70	76	70	76	70	65	59	76	
43	28	39	10 ⁴	37	50	55	54	54	54	50	54	53	55	
44	29	40	10 ⁴	6	6	6	6	6	6	6	6	6	4	
45	29	65	10 ⁴	4	3	4	2	4	4	4	4	4	4	
46	30	65	10 ⁴	2	2	3	1	3	4	3	3	3	3	
47	30	83	10 ⁴	54	54	59	60	60	59	54	54	60	56	
48	31	52	10 ⁴	133	123	134	134	134	140	145	130	143	144	
49	31	72	10 ⁴	32	32	29	32	32	32	32	32	32	32	

Continuação da tabela B.1

nùm. arco	nó origem	nó dest. (tons.)	capac. (tons.)	custos dos produtos (\$)									
				1	2	3	4	5	6	7	8	9	10
50	31	73	10 ⁴	132	110	111	111	89	119	119	109	119	110
51	32	76	10 ⁴	56	65	56	71	49	74	70	65	54	59
52	33	75	10 ⁴	76	76	78	71	70	59	77	76	78	79
53	33	77	10 ⁴	135	151	149	132	132	134	150	147	145	151
* 54	34	86	100	12	100	88	104	109	99	100	102	104	109
55	35	42	10 ⁴	23	23	13	23	23	23	23	23	21	23
56	35	74	10 ⁴	14	10	8	12	13	13	13	12	14	14
57	36	87	10 ⁴	12	21	11	21	19	21	21	12	21	20
58	37	87	10 ⁴	17	13	18	14	12	13	14	13	13	16
59	38	67	10 ⁴	7	6	4	4	5	7	7	4	6	7
60	38	83	10 ⁴	1	1	1	1	1	1	1	1	1	1
61	40	60	10 ⁴	3	4	3	3	3	3	4	4	3	4
62	41	50	10 ⁴	120	120	89	99	120	117	120	121	112	113
63	42	59	10 ⁴	23	24	23	21	23	23	23	23	23	23
64	43	61	10 ⁴	32	37	34	34	34	32	32	32	34	34
65	43	81	10 ⁴	9	7	11	9	9	8	8	9	8	10
66	44	49	10 ⁴	12	9	17	12	14	12	14	12	13	13
67	44	81	10 ⁴	23	21	23	23	23	23	23	21	23	22
68	45	46	10 ⁴	34	54	49	54	45	45	54	34	45	54
69	45	57	10 ⁴	34	32	36	34	35	32	37	23	32	34
70	47	66	10 ⁴	31	31	29	31	32	32	35	34	33	33
71	47	86	10 ⁴	57	59	30	56	50	54	53	56	54	58
72	48	56	10 ⁴	19	19	17	19	18	17	17	18	19	19
73	49	85	10 ⁴	32	34	43	43	34	43	45	43	53	40
74	50	78	10 ⁴	100	119	117	87	109	87	118	108	120	120
75	51	72	10 ⁴	40	38	36	34	34	32	39	39	40	32
76	53	85	10 ⁴	9	9	9	9	9	8	9	9	9	10
77	54	57	10 ⁴	29	24	28	28	27	21	24	25	23	23
78	54	85	10 ⁴	27	27	31	27	30	30	28	24	21	21
79	55	62	10 ⁴	13	14	14	13	13	13	13	15	12	13
80	55	80	10 ⁴	39	38	39	41	41	38	40	41	32	23
81	56	64	10 ⁴	44	48	45	34	51	49	50	43	45	51
82	57	62	10 ⁴	18	21	21	21	21	21	21	21	21	21
83	60	69	10 ⁴	3	4	4	4	1	4	4	4	4	4
84	64	17	10 ⁴	15	13	13	13	13	15	14	13	12	13
85	68	74	10 ⁴	176	180	165	160	176	170	176	132	176	178
86	69	82	10 ⁴	42	42	32	41	38	38	32	32	42	32
87	73	87	10 ⁴	21	26	23	23	27	23	24	23	23	24
88	75	79	10 ⁴	6	5	4	3	4	5	6	4	6	6
89	76	84	10 ⁴	181	176	134	177	176	176	176	176	167	183
90	76	88	10 ⁴	43	42	34	34	42	43	44	32	32	44
* 91	82	83	100	54	65	67	78	88	87	67	88	43	99
92	14	1	10 ⁴	1	1	1	1	1	1	1	1	1	1
93	51	1	10 ⁴	10	13	21	21	21	10	21	10	20	21
94	29	2	10 ⁴	51	45	57	56	56	47	57	45	57	56
95	70	2	10 ⁴	100	78	88	101	102	102	102	103	104	103
96	13	3	10 ⁴	1	1	1	1	1	1	1	1	1	1
97	17	3	10 ⁴	21	23	23	23	23	23	23	23	23	23
98	63	3	10 ⁴	0	0	0	0	0	0	0	0	0	0

Continuação da tabela B.1

nùm. arco	nó origem	nó dest. (tons.)	capac. (tons.)	custos dos produtos (\$)									
				1	2	3	4	5	6	7	8	9	10
99	16	4	10 ⁴	4	4	4	4	4	4	4	4	3	4
100	58	5	10 ⁴	60	56	65	56	65	58	76	45	56	76
101	84	6	10 ⁴	61	54	54	54	60	61	45	61	45	61
102	52	7	10 ⁴	30	31	27	32	31	32	32	32	32	32
103	59	7	10 ⁴	32	32	28	23	32	31	32	32	33	33
104	39	8	10 ⁴	100	111	111	116	119	118	110	109	109	109
105	61	8	10 ⁴	30	54	45	45	45	56	56	56	56	45
106	16	9	10 ⁴	10	19	18	3	17	18	16	18	19	19
107	66	9	10 ⁴	11	17	13	10	13	13	14	12	14	16
108	70	9	10 ⁴	4	4	2	4	5	6	5	5	5	6
109	26	10	10 ⁴	2	11	12	2	12	12	12	12	12	12
110	36	10	10 ⁴	7	11	12	7	10	11	10	12	12	12
111	32	11	10 ⁴	23	39	39	52	43	43	53	39	53	53
112	53	11	10 ⁴	1	1	1	1	1	1	1	1	1	1
113	84	11	10 ⁴	77	121	121	121	120	123	123	123	127	124
114	25	12	10 ⁴	120	123	121	123	132	132	132	130	132	130
115	37	12	10 ⁴	33	36	34	32	32	32	28	32	34	32
116	82	13	10 ⁴	45	65	65	21	61	65	54	65	60	65
117	34	14	10 ⁴	33	63	43	54	63	59	63	63	61	61
118	78	14	10 ⁴	100	123	123	123	121	123	123	123	123	123
119	48	15	10 ⁴	18	41	32	42	42	42	32	42	42	34
120	80	15	10 ⁴	19	32	21	32	32	32	32	34	34	34
121	86	15	10 ⁴	61	80	77	67	85	86	67	87	86	65
122	18	16	10 ⁴	43	48	45	43	45	49	45	43	50	34
123	71	17	10 ⁴	65	72	74	54	70	74	71	54	73	75
124	88	18	10 ⁴	43	39	34	34	40	43	43	43	42	32
125	21	19	10 ⁴	23	42	43	44	43	43	45	45	45	21
126	21	20	10 ⁴	21	32	29	30	32	21	32	32	30	21
127	71	20	10 ⁴	18	76	80	59	78	76	76	80	78	56
128	23	22	10 ⁴	9	9	9	9	9	9	9	9	9	9
129	88	23	10 ⁴	63	64	58	54	57	54	63	54	63	54
130	68	24	10 ⁴	23	23	102	102	101	103	102	100	88	100
131	58	25	10 ⁴	1	1	1	1	1	1	1	1	1	1
132	27	26	10 ⁴	3	8	7	9	8	9	2	9	9	9
133	58	27	10 ⁴	65	65	70	76	70	76	70	65	59	76
134	39	28	10 ⁴	37	50	55	54	54	54	50	54	53	55
135	40	29	10 ⁴	6	6	6	6	6	6	6	6	6	4
136	65	29	10 ⁴	4	3	4	2	4	4	4	4	4	4
137	65	30	10 ⁴	2	2	3	1	3	4	3	3	3	3
138	83	30	10 ⁴	54	54	59	60	60	59	54	54	60	56
139	52	31	10 ⁴	133	123	134	134	134	140	145	130	143	144
140	72	31	10 ⁴	32	32	29	32	32	32	32	32	32	32
141	73	31	10 ⁴	132	110	111	111	89	119	119	109	119	110
142	76	32	10 ⁴	56	65	56	71	49	74	70	65	54	59
143	75	33	10 ⁴	76	76	78	71	70	59	77	76	78	79
144	77	33	10 ⁴	135	151	149	132	132	134	150	147	145	151
*145	86	34	100	12	100	88	104	109	99	100	102	104	109
146	42	35	10 ⁴	23	23	13	23	23	23	23	23	21	23
147	74	35	10 ⁴	14	10	8	12	13	13	13	12	14	14

Continuação da tabela B.1

núm. arco	nó origem	nó dest. (tons.)	capac.	custos dos produtos (\$)									
				1	2	3	4	5	6	7	8	9	10
148	87	36	10 ⁴	12	21	11	21	19	21	21	12	21	20
149	87	37	10 ⁴	17	13	18	14	12	13	14	13	13	16
150	67	38	10 ⁴	7	6	4	4	5	7	7	4	6	7
151	83	38	10 ⁴	1	1	1	1	1	1	1	1	1	1
152	60	40	10 ⁴	3	4	3	3	3	3	4	4	3	4
153	50	41	10 ⁴	120	120	89	99	120	117	120	121	112	113
154	59	42	10 ⁴	23	24	23	21	23	23	23	23	23	23
155	61	43	10 ⁴	32	37	34	34	34	32	32	32	34	34
156	81	43	10 ⁴	9	7	11	9	9	8	8	9	8	10
157	49	44	10 ⁴	12	9	17	12	14	12	14	12	13	13
158	81	44	10 ⁴	23	21	23	23	23	23	23	21	23	22
159	46	45	10 ⁴	34	54	49	54	45	45	54	34	45	54
160	57	45	10 ⁴	34	32	36	34	35	32	37	23	32	34
161	66	47	10 ⁴	31	31	29	31	32	32	35	34	33	33
162	86	47	10 ⁴	57	59	30	56	50	54	53	56	54	58
163	56	48	10 ⁴	19	19	17	19	18	17	17	18	19	19
164	85	49	10 ⁴	32	34	43	43	34	43	45	43	53	40
165	78	50	10 ⁴	100	119	117	87	109	87	118	108	120	120
166	72	51	10 ⁴	40	38	36	34	34	32	39	39	40	32
167	85	53	10 ⁴	9	9	9	9	9	8	9	9	9	10
168	57	54	10 ⁴	29	24	28	28	27	21	24	25	23	23
169	85	54	10 ⁴	27	27	31	27	30	30	28	24	21	21
170	62	55	10 ⁴	13	14	14	13	13	13	13	15	12	13
171	80	55	10 ⁴	39	38	39	41	41	38	40	41	32	23
172	64	56	10 ⁴	44	48	45	34	51	49	50	43	45	51
173	62	57	10 ⁴	18	21	21	21	21	21	21	21	21	21
174	69	60	10 ⁴	3	4	4	4	1	4	4	4	4	4
175	17	64	10 ⁴	15	13	13	13	13	15	14	13	12	13
176	74	68	10 ⁴	176	180	165	160	176	170	176	132	176	178
177	82	69	10 ⁴	42	42	32	41	38	38	32	32	42	32
178	87	73	10 ⁴	21	26	23	23	27	23	24	23	23	24
179	79	75	10 ⁴	6	5	4	3	4	5	6	4	6	6
180	84	76	10 ⁴	181	176	134	177	176	176	176	176	167	183
181	88	76	10 ⁴	43	42	34	34	42	43	44	32	32	44
*182	83	82	100	54	65	67	78	88	87	67	88	43	99

Tabela B.2 - Arcos artificiais gerados para o exemplo 2, cap.5

núm. arco	nó origem	nó dest.
183	1	89
184	2	89
185	3	89
186	4	89
187	5	89
188	6	89
189	7	89
190	8	89

núm. arco	nó origem	nó dest.
191	9	89
192	10	89
193	11	89
194	12	89
195	13	89
196	14	89
197	15	89
198	16	89

núm. arco	nó origem	nó dest.
199	17	89
200	18	89
201	19	89
202	20	89
203	21	89
204	22	89
205	23	89
206	24	89

Continuação da tabela B.2

núm.	nó	nó
arco	origem	dest.
207	25	89
208	26	89
209	27	89
210	28	89
211	29	89
212	30	89
213	31	89
214	32	89
215	33	89
216	34	89
217	35	89
218	36	89
219	37	89
220	38	89
221	39	89
222	40	89
223	41	89
224	42	89
225	43	89
226	44	89
227	45	89
228	46	89
229	47	89
230	48	89
231	49	89
232	50	89
233	51	89
234	52	89
235	53	89
236	54	89
237	55	89
238	56	89
239	57	89
240	58	89
241	59	89
242	60	89
243	61	89
244	62	89
245	63	89
246	64	89
247	65	89
248	66	89
249	67	89
250	68	89
251	69	89
252	70	89
253	71	89
254	72	89
255	73	89
256	74	89
257	75	89

núm.	nó	nó
arco	origem	dest.
258	76	89
259	77	89
260	78	89
261	79	89
262	80	89
263	81	89
264	82	89
265	83	89
266	84	89
267	85	89
268	86	89
269	87	89
270	88	89
271	89	1
272	89	2
273	89	3
274	89	4
275	89	5
276	89	6
277	89	7
278	89	8
279	89	9
280	89	10
281	89	11
282	89	12
283	89	13
284	89	14
285	89	15
286	89	16
287	89	17
288	89	18
289	89	19
290	89	20
291	89	21
292	89	22
293	89	23
294	89	24
295	89	25
296	89	26
297	89	27
298	89	28
299	89	29
300	89	30
301	89	31
302	89	32
303	89	33
304	89	34
305	89	35
306	89	36
307	89	37
308	89	38

núm.	nó	nó
arco	origem	dest.
309	89	39
310	89	40
311	89	41
312	89	42
313	89	43
314	89	44
315	89	45
316	89	46
317	89	47
318	89	48
319	89	49
320	89	50
321	89	51
322	89	52
323	89	53
324	89	54
325	89	55
326	89	56
327	89	57
328	89	58
329	89	59
330	89	60
331	89	61
332	89	62
333	89	63
334	89	64
335	89	65
336	89	66
337	89	67
338	89	68
339	89	69
340	89	70
341	89	71
342	89	72
343	89	73
344	89	74
345	89	75
346	89	76
347	89	77
348	89	78
349	89	79
350	89	80
351	89	81
352	89	82
353	89	83
354	89	84
355	89	85
356	89	86
357	89	87
358	89	88

B I B L I O G R A F I A

Assad, A.A. (1978). "Multicommodity Network Flows - A Survey ", Networks , Vol.8 , pp. 37-91.

Assad, A.A. (1980). "Modeling for Rail Transportation" , Transportation Research - A , Vol.14 , No.3, pp.205-220.

Bazaraa, M.S. and Jarvis, J.J. (1977) "Linear Programming and network flows" , John Wiley & Sons.

Bodin, L.D., Golden ,B.L., Schuster, A.D. and Romig, W. (1980) "A Model for blocking of trains", Transportation Research - B , vol.14 , No.1/2 , pp.115-120.

Glickman , T.S. and Sherali, H.D. (1985) "Large-scale network distribution of pooled empty freight cars over time , with limited substitution and equitable benefits " , Transportation Research - B , vol.19 , No.2 , pp.85-94.

Haghani, A.E. (1987). "Rail Freight Transportation: A Review of Recent Optimization Models for Train Routing and Routing" Journal of Advanced Transportation , Vol.21, No.2, pp.147-172.

Hartman, J.K. and Lasdon , L.S. (1972) "A generalized upper bounding algorithm for multicommodity network flow problems" , Networks , Vol.1 , pp.333-354.

Helgason , R.V. and Kennington ,J.L. (1977) "A product form

representation of the inverse of a multicommodity cycle matrix", Networks , Vol.7 , pp.297-322.

Kennigton , J.L. (1977) "Solving multicommodity transportation problems using a primal partitioning simplex technique", Naval Research Logistics Quartely , Vol.24 , pp.309-325.

Kennigton , J.L. and Helgason , R.V. (1980) " Algorithms for network programming ", John Wiley & Sons.

Lasdon , L.S. (1970) "Optimization theory for large systems", Macmillan Company, New York.

Mendiratta, V.B. and Turnquist, M.A. (1982) "Model for management of empty freight cars" , Transportation Research Record No.838 , pp.50-55.

Philip,C.E. and Sussman , J.M. (1977) "Inventory model of the railroad empty car distribution process" Transportation Research Record No.656 , pp.56-60.

Sharda,R. and Somarajan,C. (1986) "Comparative performance of advanced microcomputer LP systems", Comput. & Ops. Res., Vol.13, pp.131-147.

Tavares, H. (1984) Notas de aula do curso IA-511 "Otimização em grafos" , FEE-Unicamp.

Turnquist , M.A. and Daskin , M.S. (1982) " Queuing models of classification and connection delay in railyards" Transportation Science , Vol.16 , No.2 , pp.207-230.