

O PROBLEMA DO CARTEIRO CHINÊS

Este exemplar corresponde a redação final da tese de mestrado devidamente corrigida e defendida pelo Sr. Jaime de Mattos Taube e aprovada pela Comissão Julgadora.

Campinas, 26 de novembro de 1992

Prof. Dr.



Clóvis Perin Filho

Dissertação apresentada ao Instituto de Matemática, Estatística e Ciência da Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Matemática Aplicada.

UNICAMP
BIBLIOTECA CENTRAL

01100010

RESUMO

Nesta dissertação é feito um estudo do problema do carteiro chinês: as diferentes apresentações, formulações e métodos de resolução. Foi feita uma implementação do método de resolução do problema definido em redes não orientadas, que utiliza a teoria de emparelhamento. Finalmente, é feito o estudo de um problema de distribuição de jornal.

Dedicado a meus pais, Maria José e Miguel.

AGRADECIMENTOS

Quero agradecer especialmente ao Professor Clóvis Perin Filho pela sua constante orientação, tanto em relação ao trabalho de implementação computacional realizado, quanto à redação desta dissertação.

Agradeço também o Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio financeiro a mim concedido em forma de bolsa de mestrado.

Finalmente, quero agradecer a Viviane pelo incentivo e colaboração, tão importantes ao desenvolvimento deste trabalho.

I. O PROBLEMA DO CARTEIRO CHINÊS

I.1 INTRODUÇÃO

I.2 DEFINIÇÕES

I.2.1 REDE MISTA

I.2.2 REDE NÃO ORIENTADA

I.2.3 REDE ORIENTADA

I.3 O PROBLEMA DE EULER

I.4 A ROTA DO CARTEIRO CHINÊS

I.4.1 REDE NÃO ORIENTADA

I.4.2 REDE ORIENTADA

I.4.3 REDE MISTA

II. O PROBLEMA DE EMPARELHAMENTO

II.1 EMPARELHAMENTO MÁXIMO

II.2 EMPARELHAMENTO PERFEITO ÓTIMO

III. MÉTODO DE SOLUÇÃO DO PROBLEMA DO CARTEIRO CHINÊS EM REDE NÃO ORIENTADA

III.1 PROGRAMA PRINCIPAL

III.2 EMPARELHAMENTO PERFEITO

III.2.1 ESTRUTURA DE DADOS

III.2.2 ROTINAS

IV. ESTUDO DE CASO E CONCLUSÕES

IV.1 UM PROBLEMA DE DISTRIBUIÇÃO DE JORNAIS

IV.2 SOLUÇÃO RELAXADA

IV.3 SOLUÇÃO HEURÍSTICA

IV.4 AVALIAÇÃO DA ROTA

IV.5 CONCLUSÕES

IV.6 COMENTÁRIOS FINAIS

CAPÍTULO I
O PROBLEMA DO CARTEIRO CHINÊS

1.1 INTRODUÇÃO

Considere a atividade diária de um carteiro. No início do dia ele examina a correspondência e elabora um roteiro de entrega. Partindo da agência de correio, o carteiro deve percorrer as ruas de forma a cobrir todos os endereços que constam em sua lista de entrega e, finalmente, retornar à agência para devolver a correspondência que não pôde ser entregue. Observe que, ao elaborar o roteiro de entrega, o carteiro procura fazê-lo de modo a percorrer a menor distância total.

Existe uma diversidade de situações semelhantes ao problema de entrega do carteiro. Por exemplo, a distribuição de jornais para bancas de revistas em uma cidade e também a distribuição de jornais para seus assinantes. Outro problema semelhante é o de patrulhamento, em que um carro de polícia deve cobrir uma certa quantidade de ruas em um bairro.

O problema acima é conhecido como o problema do carteiro chinês (PCC), que de uma forma mais geral se enquadra na classe de problemas de roteamento e distribuição. Em modelos de roteamento, aparecem restrições relacionando quantidades a serem entregues e compromissos de entrega para determinados períodos ou janelas de tempo. Se, por exemplo, o agente de entrega é uma frota de caminhões, então o tamanho desta, assim como limitações de capacidade de cada veículo, representam também restrições a serem consideradas pelo modelo. Neste caso, o estudo do roteamento ou distribuição pode também incluir uma análise indicando o dimensionamento ideal da frota. Um bom exemplo contendo todos estes elementos é o problema de distribuição que os fabricantes de refrigerantes enfrentam.

Um outro problema de otimização combinatorial, também muito conhecido, é o problema do caixeiro viajante (PCV), que consiste em determinar uma rota de comprimento total mínimo passando por todos os vértices de uma dada rede; trata-se, portanto, de um problema diferente do problema do carteiro chinês (PCC). Tanto o (PCC) quanto o (PCV) pertencem à classe de problemas de roteamento. O estudo dos mesmos é fundamental no desenvolvimento de métodos e algoritmos para modelos mais gerais. Em alguns casos algoritmos para (PCC) e (PCV) servem como subrotinas no desenvolvimento de métodos para a solução de problemas de roteamento.

Nesta dissertação foi tratado o problema do carteiro chinês (PCC). Este é apresentado para três tipos de rua que o carteiro deve percorrer: ruas de mão dupla, ruas de mão única e ruas de ambos os tipos. Foi implementado o algoritmo de Edmonds

[2], para resolver o (PCC) em redes não orientadas, que se baseia na teoria de emparelhamento. Além disso, foi feito também um estudo de caso, tomando como base um problema de distribuição de jornais para assinantes na cidade de São Paulo. Para este problema foram analisadas duas soluções (rotas de entrega de jornais). Uma solução relaxada, que considera todas as ruas como sendo ruas de mão dupla e uma solução heurística que considera a orientação das ruas.

No capítulo I, o problema do carteiro chinês é apresentado como um problema de otimização combinatorial abordando os casos de ruas de mão dupla, mão única e mista. Além disso são discutidos métodos de solução do problema. No capítulo II é feita a descrição de um algoritmo polinomial para o problema de emparelhamento perfeito ótimo (EPO) baseado em [10]. A solução de (EPO) é parte do método de solução definido para ruas com mão dupla. No capítulo III é feita a descrição da implementação computacional para (PCC) com estas características. Finalmente, o capítulo IV é um estudo de caso a respeito de um problema de distribuição de jornais na cidade de São Paulo.

1.2 DEFINIÇÕES

Nesta seção são apresentadas definições e conceitos utilizados em teoria de grafos que são necessários para a apresentação deste trabalho.

1.2.1 REDE MISTA

Uma *rede* $R(V,A,E)$ é definida por um conjunto V não vazio e finito de *vértices*, um conjunto A de pares ordenados de vértices distintos, denominados *arcos*, e um conjunto E de pares não ordenados de vértices denominados *arestas*. Se (u,v) é um arco, então os vértices u e v são denominados *cauda* e *cabeça* do arco respectivamente. Se $\{u,v\}$ é uma aresta, então os vértices u e v são denominados *extremos* da aresta. Uma rede $R(V,A,E)$ com $A \neq \emptyset$, $E \neq \emptyset$ é chamada *rede mista*.

Uma sequência a_1, a_2, \dots, a_k de arcos e arestas, onde $a_i = (v_{i-1}, v_i)$ ou $a_i = \{v_{i-1}, v_i\}$, $i=1,2,\dots,k$ e nenhum vértice v_i é incidente em mais de dois arcos ou arestas da sequência, é um *caminho*. Note que este caminho passa pelos vértices v_0, v_1, \dots, v_k . Um *ciclo* é um caminho em que os vértices inicial e final v_0 e v_k são iguais.

Uma sequência a_1, a_2, \dots, a_k de arcos e arestas, onde $a_1 = (v_{1-1}, v_1)$ ou $a_j = (v_{j-1}, v_j)$, $1=1,2,\dots,k$ e $v_0 = v_k$, é dita ser uma *rota*.

O *comprimento* de um caminho, ciclo ou rota é igual ao número de arcos e arestas neles contido. A cada arco e aresta de uma rede é associado um *tamanho*. Dado um caminho, ciclo ou rota a_1, a_2, \dots, a_k , seu *tamanho* é definido por $\sum_{i=1}^k c_i$, onde c_i é igual ao tamanho do arco ou aresta a_i .

Seja a rede $R(V, A, E)$. Dado o subconjunto de vértices $V' \subset V$, a *subrede gerada* por V' é a rede maximal $R'(V', A', E')$ com $A' \subset A$ e $E' \subset E$. Por outro lado, dados os subconjuntos $A' \subset A$ e $E' \subset E$ de arcos e arestas, respectivamente, a *subrede gerada* por $A' \cup E'$ é a rede minimal $R'(V', A', E')$, com $V' \subset V$.

1.2.2 REDE NÃO ORIENTADA

Uma rede $R(V, A, E)$ com $A = \emptyset$ é dita uma rede *não orientada*. Neste caso omitimos o conjunto de arcos A e designamos a rede por $R(V, E)$.

Dizemos que uma rede $R(V, E)$ é *conexa* se, para cada par de vértices u e v , existe um caminho conectando u e v .

Um subconjunto T de arestas de uma rede não orientada é uma *árvore* se:

- a) T gera uma rede conexa.
- b) T não contém ciclos.

Um subconjunto de arestas que não contém ciclos é uma *floresta*. Uma *árvore geradora* relativa a uma rede $R(V, E)$ é uma árvore cuja rede por ela gerada contém todos os vértices em V .

Dado o subconjunto de vértices $V' \subset V$, definimos $\delta(V') \subset E$ como sendo o subconjunto maximal de arestas (v, u) tal que $|\{v, u\} \cap V'| = 1$. Ou seja, $\delta(V')$ é o subconjunto de arestas com exatamente um extremo em V' . Para o mesmo subconjunto V' definimos $\gamma(V') \subset E$ como sendo o subconjunto maximal de arestas (v, u) tal que $|\{v, u\} \cap V'| = 2$. Ou seja, $\gamma(V')$ é o subconjunto de arestas com dois extremos em V' .

O *grau* d_v de um vértice v é definido como sendo igual a $|\delta(v)|$. Um vértice sobre o qual incida um número ímpar de arestas é dito um *vértice ímpar* (o número de vértices ímpares em qualquer rede é par). Uma rede é dita *par* se todos seus vértices possuem grau par.

Uma rede $R(V, E)$ é dita *completa* se a todo par de vértices em V está associada uma aresta em E .

A rede $R(V,E)$ é dita *bipartida* se existe uma partição do conjunto de vértices, $V = V_1 \cup V_2$ e $V_1 \cap V_2 = \emptyset$, tal que $\delta(V_1) = \delta(V_2) = E$ (ou $\gamma(V_1) = \gamma(V_2) = \emptyset$).

1.2.3 REDE ORIENTADA

Uma rede $R(V,A,E)$, onde $E = \emptyset$ é dita uma *rede orientada*. Neste caso omitimos o conjunto E e designamos a rede por $R(V,A)$.

Dado o subconjunto de vértices $V' \subset V$, definimos $\delta^+(V') \subset A$ como sendo o subconjunto de arcos tal que para $(v,u) \in \delta^+(V')$ temos que $v \in V'$ e $u \notin V'$. Da mesma maneira, definimos $\delta^-(V') \subset A$ como sendo o subconjunto de arcos tal que para $(v,u) \in \delta^-(V')$ temos que $u \in V'$ e $v \notin V'$.

O *grau interno* d_v^- de um vértice v é igual a $|\delta^-(v)|$ e o *grau externo* d_v^+ de um vértice v é igual a $|\delta^+(v)|$. Uma rede $R(V,A)$ é dita *simétrica* se $d_v^- = d_v^+$, $\forall v \in V$.

Uma rede $R(V,A)$ é dita *biconectada* se para cada par de vértices u e v existe um caminho de u para v e um caminho de v para u .

1.3 O PROBLEMA DE EULER

O matemático suíço Leonhard Euler (1707-1783) escreveu em 1736 um dos primeiros artigos em teoria de grafos a respeito de um problema conhecido como o problema das pontes de Königsberg. A cidade de Königsberg, no leste da antiga Prússia, situava-se sobre a junção de dois rios e duas ilhas. As margens, assim como as ilhas, eram interligadas por sete pontes como mostra a figura I.1. Em seu artigo, Euler discutia a existência ou não de uma rota onde um cidadão, saindo de sua casa, percorresse uma única vez cada ponte e retornasse ao ponto de partida.

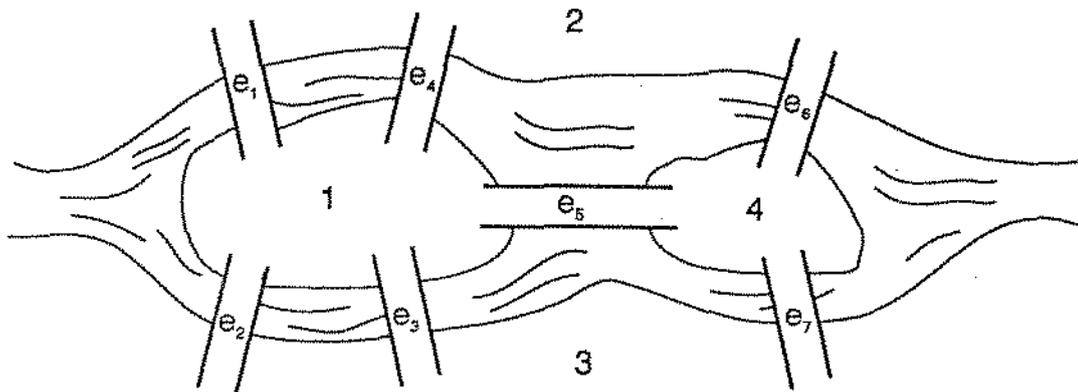


figura I.1

Este problema pode ser colocado em termos de uma rede. Considere a figura I.2 a seguir. A rota que Euler procurava corresponde a uma rota passando uma única vez por cada aresta desta rede. Não é difícil ver que neste caso não existe tal rota. Sobre o vértice 4, por exemplo, incidem três arestas. Utilizando a aresta e_7 para atingi-lo, a aresta e_6 para deixá-lo e retornando pela aresta e_5 , é necessário repetir uma das três arestas para abandonar o vértice 4 novamente.

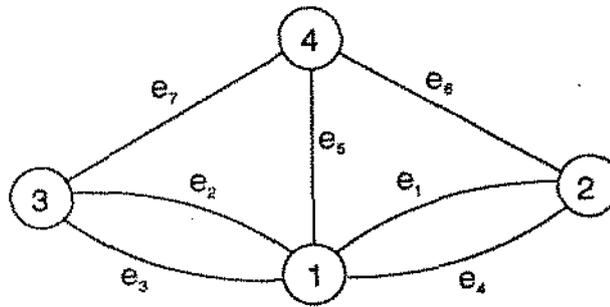


figura I.2

Determinação de uma rota de Euler

O algoritmo abaixo determina uma rota de Euler em uma rede $R(V,E)$. A sequência $L_v(1), L_v(2), \dots, L_v(k)$ associada ao vértice $v \in V$ representa a ordem em que os vértices adjacentes a v são atingidos a partir de v . Ou seja, quando passarmos por v pela primeira vez, seguimos para o vértice $L_v(k)$. Na segunda vez seguimos para o vértice $L_v(k-1)$, e assim por diante. Veja a descrição abaixo deste algoritmo.

- i) Seja r um vértice qualquer. Inicialmente n_0 e n são iguais a r . Faça $k_v = 0$ para todo vértice $v \in V$. Seja $a \in A$ adjacente ao vértice n . Inicialmente todas as arestas são não rotuladas.
- ii) Seja m tal que $a = (m,n)$. Faça $k_m = k_m + 1$ e $L_m(k_m) = n$. Rotule a aresta a . Se o vértice m incide sobre uma aresta ainda não rotulada vá para o passo iii). Caso contrário m deve ser igual a n_0 . Neste caso vá para o passo iv).
- iii) Faça $n = m$ e a igual a qualquer aresta adjacente a m não rotulada. Vá para o passo ii).
- iv) Seja n_0 qualquer vértice adjacente a pelo menos uma aresta não rotulada e pelo menos uma aresta rotulada. Faça a igual a uma aresta não rotulada, adjacente a n_0 e vá para o passo ii). Pare, se não houver tal n_0 .

1.4 A ROTA DO CARTEIRO CHINÊS

O primeiro trabalho a respeito do problema do carteiro chinês (PCC) [1], foi publicado em uma revista chinesa. Seu autor, o matemático chinês Kuan-Mei-Ko apresenta condições necessárias e suficientes para que uma dada rota, satisfazendo as condições do problema, seja ótima. Entretanto, a descrição do método sugerido para resolvê-lo não leva em conta aspectos como implementação computacional e análise de complexidade. Em 1972, Edmonds e Johnson [2] propuseram um algoritmo fortemente polinomial para (PCC) em uma rede não orientada baseado na teoria de emparelhamento desenvolvida por Edmonds na década de 60 [3].

O método de resolução do problema do carteiro chinês depende do tipo de rede utilizada. Se a rede é não orientada, ou seja, a rede contém somente arestas, então os algoritmos propostos baseiam-se na teoria de emparelhamento. Se a rede é orientada, (PCC) é reduzido a um problema de fluxo. Em redes mistas o problema (PCCP) é NP-completo.

1.4.1 REDE NÃO ORIENTADA

A condição necessária e suficiente para que o problema do carteiro chinês tenha solução é que a rede $R(V,E)$ seja conectada. Se a rede é par, então a solução para o problema do carteiro chinês é uma rota de Euler. Se a rede R não é par, a rota do carteiro chinês deverá, necessariamente, percorrer algumas arestas mais de uma vez. Seja $1 + x_a$ o número de vezes que a aresta $a \in E$ é percorrida na solução. Então a rede R' gerada pelo conjunto E contendo $1 + x_a$ cópias da aresta $a \in E$ é uma rede par. Resolver o problema do carteiro chinês é, portanto, o mesmo que determinar uma solução x_a para o problema de programação inteira (PCC).

(PCC) minimizar $\sum_a c_a x_a$

restrito a (1):

$$x(\delta(v)) - 2w_v = b_v, \quad v \in V.$$

$$x_a \geq 0$$

$$w_v \geq 0$$

$$x_a \text{ inteiro}$$

$$w_v \text{ inteiro}$$

onde $x(\delta(v)) = \sum_{a \in \delta(v)} x_a$ e b_v é igual a um se v é um vértice ímpar e zero se v for um vértice par.

Edmonds e Johnson [2], baseados na teoria de emparelhamento desenvolvida por Edmonds em 1965, apresentaram em 1973 dois métodos para resolver o problema (PCC). O primeiro método consiste basicamente em resolver um problema de caminho mínimo e um problema de emparelhamento perfeito. O segundo método se baseia em um algoritmo de emparelhamento mais geral e uma formulação linear (PE) equivalente a (PCC).

(PE) minimizar $\sum_a c_a x_a$

restrito a (2):

$$x(\delta(S)) \geq 1, \quad \forall S \subset V \text{ com } |S| \text{ ímpar e } |S| \geq 3$$

$$x_a \geq 0$$

onde $|S|$ representa a cardinalidade do conjunto.

Nesta dissertação foi implementado o primeiro método. Passemos a seguir à sua descrição.

Determina-se primeiro o caminho mínimo entre todos os vértices ímpares da rede. Para isso existem algoritmos de complexidade $O(|V|^3)$, onde $|V|$ é igual ao número de vértices da rede original R . Constrói-se, então, uma rede completa $R_I(V_I, E_I)$, cujos vértices são os vértices ímpares de R . O tamanho de uma aresta em R_I é igual ao tamanho do caminho mínimo ligando os dois vértices ímpares correspondentes em R . Determina-se um emparelhamento perfeito ótimo em R_I , ou seja, um subconjunto $M_I \subset E_I$ de arestas onde cada vértice de R_I pertence a exatamente uma aresta de M_I e a soma dos tamanhos de todas as arestas em M_I é mínima. Cada aresta $m \in M_I$ da rede R_I corresponde a um caminho mínimo ligando dois vértices ímpares em R . Faça $x_a = 1$ para

toda aresta $a \in E$ presente em um caminho mínimo correspondendo a uma aresta do emparelhamento M_1 . Construa a rede R' contendo $1 + x_a$ cópias da aresta a .

Veja o exemplo da figura I.3. Determina-se primeiro o caminho mínimo entre os vértices ímpares a, c, d e f da rede R . É construída, então, a rede R_1 contendo somente os vértices ímpares. Uma vez obtido o emparelhamento ótimo $\{(a,f);(c,d)\}$ em R_1 , constrói-se a rede R' com as arestas (a,f) e (c,d) duplicadas. A rede R' assim obtida é uma rede par. O caminho de Euler em R' corresponde à solução para (PCC) relativa à rede R .

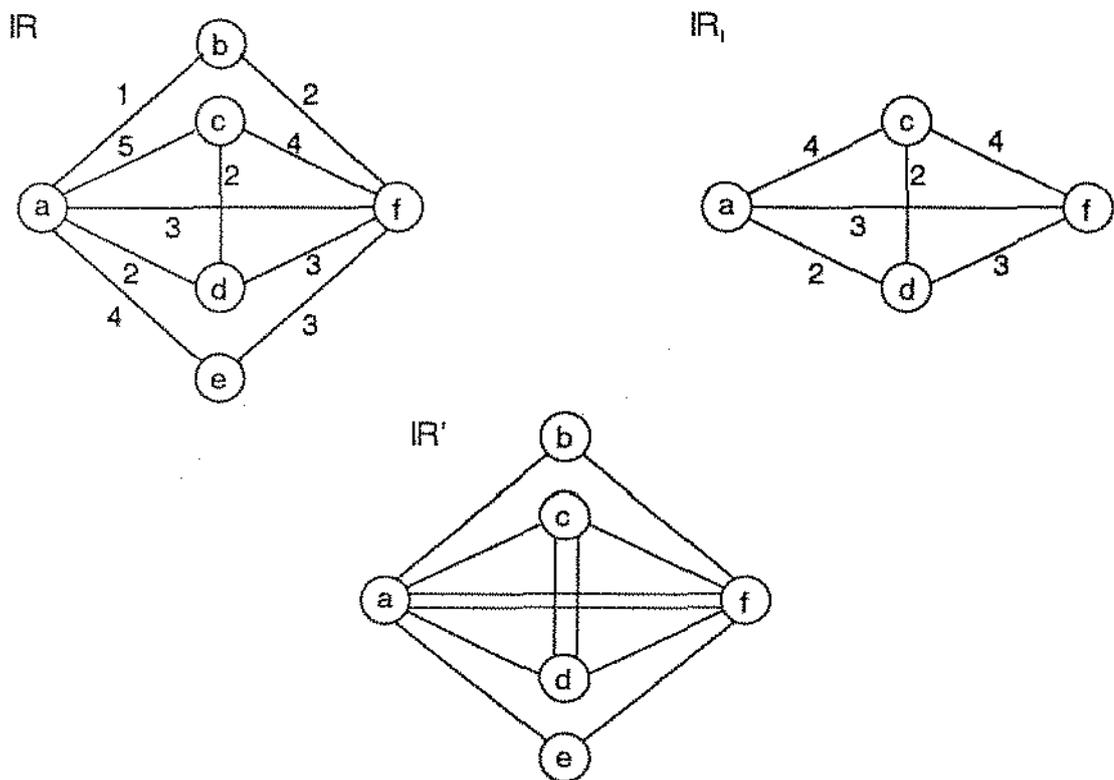


figura I.3

Considere a demonstração deste algoritmo. Em primeiro lugar, dois caminhos mínimos do emparelhamento M_1 têm que ser disjuntos e, portanto, é correto fixar $x_a = 1$ ou 0 . Para mostrar isto, suponha que existam dois caminhos mínimos do emparelhamento M_1 ligando os vértices v_1 e v'_1 e v_2 e v'_2 respectivamente, e uma aresta $a \in E$ pertencente a ambos. Veja que, removendo a , ficamos com dois caminhos entre v_1 e v_2 (ou v'_2) e v'_1 e v'_2 (ou v_2). O tamanho desses caminhos é menor que o tamanho dos caminhos ligando v_1 e v'_1 e v_2 e v'_2 . Portanto a soma das arestas $\{v_1, v_2\}$ e $\{v'_1, v'_2\}$

em R_1 é menor que a soma das arestas (v_1, v'_1) e (v_2, v'_2) . Poderíamos, então, criar um novo emparelhamento M'_1 substituindo as arestas (v_1, v'_1) e (v_2, v'_2) do emparelhamento M_1 pelas arestas (v_1, v_2) e (v'_1, v'_2) . O novo emparelhamento M'_1 teria um tamanho total menor que o emparelhamento M_1 , o que não é possível uma vez que, por hipótese, M_1 é um emparelhamento perfeito ótimo em R_1 .

Resta mostrar que existem inteiros não negativos w_v tal que a equação $x(\delta(v)) - 2w_v = b_v$, com $v \in V$ seja satisfeita. Mas veja que $x(\delta(v))$ é par se $v \in V$ for um vértice par, e ímpar caso contrário. Isto porque, em cada caminho mínimo correspondendo a uma aresta do emparelhamento M_1 , temos que $x(\delta(v))$ é par para todos os vértices pares do caminho e ímpar para os dois vértices ímpares na extremidade deste caminho. Mas cada vértice ímpar é extremidade de um único caminho mínimo correspondente a uma aresta do emparelhamento M_1 . Fica assim demonstrado que a solução x_a produzida pelo algoritmo é uma solução factível do problema (PCC).

A solução x_a produzida pelo algoritmo é também uma solução ótima de (PCC). Considere uma solução ótima x_a^* de (PCC). Vamos partir da hipótese que o tamanho das arestas é estritamente positivo ($c_a > 0$). Suponha $x_a^* = 2k + 1$, $k > 1$ para uma aresta $a \in E$ qualquer. Faça $x_a^* = 1$ e $w_v^* = w_v^* + 2k$ para todo v sobre o qual $a \in E$ incida. Com esta transformação obteríamos uma nova solução factível, porém com o valor da função objetivo estritamente menor, o que contraria a hipótese de que x_a^* é ótima. Portanto, x_a^* só pode assumir os valores zero e um. Isto implica que uma solução ótima x_a^* é composta de caminhos, conectando vértices ímpares, e ciclos. Pelo mesmo argumento utilizado anteriormente, estes caminhos têm que ser disjuntos e não existem tais ciclos, caso contrário, x_a^* não seria solução ótima. Como um emparelhamento perfeito ótimo representa a melhor combinação desses caminhos, a solução determinada pelo algoritmo é de fato uma solução ótima de (PCC).

1.4.2 REDE ORIENTADA

A condição necessária e suficiente para a existência de uma rota do carteiro chinês em uma rede $R(V, A)$ é que esta seja biconectada. Se a rede é simétrica, então a solução para o problema do carteiro chinês é uma rota de Euler (que obedece à orientação dos arcos). Caso a rede R não seja simétrica, a rota do carteiro chinês deverá percorrer alguns arcos mais de uma vez.

Seja a rede orientada $R(V,A)$. O problema do carteiro chinês definido para a rede assimétrica R , onde c_a é o custo da aresta a , pode ser colocado em termos do problema de fluxo abaixo.

$$\begin{aligned}
 \text{(PCCO)} \quad & \text{minimizar } \sum_a c_a x_a \\
 & \text{restrito a (3):} \\
 & x(\delta^+(v)) - x(\delta^-(v)) = b_v, \quad v \in V. \\
 & x_a \geq 0
 \end{aligned}$$

onde $b_v = d_v^- - d_v^+$, por definição.

Este Problema de minimização é conhecido como o problema de fluxo ótimo. Um vértice v é denominado fonte, sorvedouro ou intermediário se $b_v > 0$, $b_v < 0$, $b_v = 0$, respectivamente. Como os custos são valores inteiros e os b_v 's também o são, uma solução ótima básica x_a^* de (PCCO) é uma solução inteira não negativa (por causa da unimodularidade de A).

1.4.3 (PCC) PARA REDE MISTA

A condição necessária e suficiente para que exista uma rota do carteiro chinês em uma rede mista $R(V,A,E)$ é que esta seja uma rede biconectada. Neste caso, existem três situações distintas:

- a) R é uma rede par e simétrica.
- b) R é uma rede par não simétrica.
- c) R é uma rede não par.

Rede par e simétrica

Neste caso, a rota do carteiro é a uma rota de Euler e é determinada em dois estágios. No primeiro estágio são determinados ciclos passando por todas as arestas. No segundo estágio são determinados ciclos passando por todos os arcos. Os ciclos formados em cada estágio são, então, agregados para formar a rota de Euler da rede completa.

Rede par e não simétrica

O algoritmo para determinar uma rota do carteiro começa por fixar arbitrariamente uma orientação para cada aresta de E . Define-se, assim, uma rede orientada $R_0(V, A_0)$, onde $A_0 = A \cup A(E)$, sendo o conjunto $A(E)$ o conjunto dos arcos definidos a partir de arestas de E com orientações fixadas arbitrariamente. Resolve-se o problema de fluxo (PCCO), porém com algumas alterações devido à definição arbitrária de direções das arestas.

Se todos os vértices de R_0 são vértices intermediários, então R_0 é uma rede par e simétrica e, portanto, basta determinar a rota de Euler sobre R_0 aplicando o método dos dois estágios descrito para este caso. Caso os vértices de R_0 não sejam todos intermediários, construa uma rede $R'(V, A')$ da seguinte forma:

- a) Para cada arco $a \in A$, coloque este arco em A' e faça com que o mesmo tenha capacidade infinita e custo igual ao tamanho de a .
- b) Para cada aresta $a \in E$ coloque dois arcos em A' correspondendo às duas possíveis orientações de a . Faça com que cada um destes arcos tenha capacidade infinita e custo igual ao custo de a .
- c) Para cada aresta $a \in E$ coloque um arco em A' que seja um arco com direção contrária ao arco em A_0 associado à aresta a . Arcos deste tipo são chamados *arcos artificiais*. Faça com que o custo de cada arco artificial seja igual a zero e sua capacidade igual a dois.

Aplique o algoritmo de fluxo ótimo para a rede R' , considerando as demandas e suprimentos correspondentes à rede orientada R_0 .

Se o algoritmo termina sem encontrar uma solução factível, então não existe uma rota factível para o problema inicial. Caso contrário, seja x_a o valor do fluxo no arco a de R' (este valor será sempre inteiro positivo). Na prova deste algoritmo é

mostrado que o fluxo básico ótimo associado a um arco artificial é sempre igual a zero ou dois. Construa uma rede R^* da seguinte maneira:

- a) Coloque em R^* $x_a + 1$ cópias de cada arco de R' que não seja um arco artificial.
- b) Se o fluxo em um arco artificial é igual a dois, então coloque uma unidade deste arco em R^* .
- c) Se o fluxo no arco artificial é igual a zero, reverta a direção deste arco e coloque uma cópia do novo arco em R^* . Em outras palavras, se o fluxo de um arco artificial é zero, então a direção arbitrada na rede orientada R_0 é mantida, se é igual a dois sua direção é invertida.

A demonstração da validade deste algoritmo, assim como a solução para o problema de fluxo ótimo, estão além do escopo deste trabalho. Em caso de um estudo mais profundo, sugere-se [7,8] como referências.

Rede não par

Não é conhecido nenhum algoritmo polinomial para o problema do carteiro chinês definido em uma rede mista não par; ou seja, trata-se de um problema NP-completo. No entanto, Minieka [5] sugere uma heurística com complexidade $O(n^2)$ baseada em dois estágios: primeiro, torne a rede par, segundo, torne a rede simétrica utilizando o algoritmo descrito acima. Não existe nenhuma garantia de que esta estratégia conduza a uma solução ótima para o problema inicial. Como última referência a respeito do problema do carteiro chinês considere os artigos [5,6] a respeito do problema do carteiro chinês definido para redes não orientadas onde os custos de uma aresta quando percorrida nos dois sentidos são diferentes.

No capítulo IV é apresentado um método heurístico para resolver o problema de distribuição de jornais em um estudo de caso. O método empregado pode ser facilmente especializado para o (PCC).

CAPÍTULO II
O PROBLEMA DE EMPARELHAMENTO

Este capítulo foi elaborado com base nas referências [9,10]. É apresentado um algoritmo para resolver o problema de emparelhamento ótimo (EPO). Trata-se de um algoritmo primal-dual, baseado na teoria de emparelhamento desenvolvida por Jack Edmonds [3] em 1965. O algoritmo evolui por emparelhamentos não perfeitos (soluções primal infactíveis) e soluções dual factíveis satisfazendo às condições de folgas complementares. Em uma das fases do algoritmo é determinado um emparelhamento máximo (EM) em uma subrede. Desta forma, vamos discutir primeiro o problema de emparelhamento máximo.

II.1 EMPARELHAMENTO MÁXIMO

Seja a rede não orientada $R(V,E)$. Dado um subconjunto M de arestas de E , defina o vetor x , onde $x_a = 1$ se $a \in M$ e $x_a = 0$ caso contrário, como sendo o *vetor de incidência* de M . Dizemos que M (ou x) é um *emparelhamento* em R se $x(\delta(v)) \leq 1$ para todo $v \in V$ e, neste caso, dizemos que a é uma aresta emparelhada se $x_a = 1$. Em outras palavras, um subconjunto $M \subset E$ é um emparelhamento se sobre cada vértice $v \in V$ incidir no máximo uma aresta emparelhada. Um vértice $v \in V$ é chamado *vértice exposto* em relação ao emparelhamento M se sobre ele incidir somente arestas não emparelhadas ($x(\delta(v)) = 0$). Se $x(\delta(v)) = 1, \forall v \in V$, ou seja, se sobre cada vértice da rede incide exatamente uma aresta emparelhada, dizemos que o vetor x é um emparelhamento perfeito. A formulação abaixo (EM) corresponde ao problema de determinar um emparelhamento de cardinalidade máxima.

$$\begin{aligned}
 \text{(EM)} \quad & \text{maximizar } \sum_a x_a \\
 & \text{restrito a (1):} \\
 & x(\delta(v)) \leq 1, \forall v \in V \\
 & x_a \geq 0, \forall a \in E \\
 & x_a \text{ inteiro para todo } a \in E
 \end{aligned}$$

Um subconjunto de vértices $Y \subset V$ com $|Y|$ ímpar maior ou igual a três é dito ser um subconjunto *ímpar*. Se x é um emparelhamento, então vale a equação abaixo.

$$x(\gamma(Y)) \leq (|Y|-1)/2$$

Seja $F = \{V_1, V_2, \dots, V_k\}$ a família de todos os subconjuntos ímpares. Considere a seguinte formulação.

$$\begin{aligned}
 \text{(EM-P)} \quad & \text{maximizar } \sum_a x_a \\
 & \text{restrito a (2) :} \\
 & x(\delta(v)) \leq 1, \forall v \in V. \\
 & x(\gamma(Y)) \leq (|Y|-1)/2, \forall Y \in F \\
 & x_a \geq 0, \forall a \in E
 \end{aligned}$$

Qualquer vetor x satisfazendo (1) também satisfaz (2). Por outro lado, qualquer vetor x , inteiro, satisfazendo (2), satisfaz (1). Portanto, se a solução ótima de (EM-P) é uma solução inteira, é também solução ótima de (EM). É apresentado a seguir um algoritmo para (EM-P) que termina com uma solução ótima inteira, sendo portanto solução ótima de (EM) também.

Seja π_v a variável dual associada à restrição sobre o vértice v e μ_Y a variável dual associada à restrição sobre o conjunto ímpar Y , em (EM-P). Veja que o número de componentes do vetor μ depende exponencialmente de $|V|$. Considere as seguintes definições dadas a partir da solução dual (π, μ) :

$$\begin{aligned}
 \pi(a) &= \pi_v + \pi_u, \quad a = \langle v, u \rangle \\
 \mu(a) &= \sum (\mu_Y : a \in \gamma(Y)) \\
 d_a(\pi, \mu) &= \pi(a) + \mu(a)
 \end{aligned}$$

O dual de (EM-P) é

$$\begin{aligned}
 \text{(EM-D)} \quad & \text{minimizar } \sum_v \pi_v + \sum_{Y \in F} (|Y|-1)(\mu_Y)/2 \\
 & \text{restrito a (3) :} \\
 & d_a(\pi, \mu) \geq 1, \forall a \in E \\
 & \pi \geq 0, \mu \geq 0
 \end{aligned}$$

O algoritmo a ser apresentado para (EM-P) evolui por soluções primal e dual factíveis e atinge otimalidade quando as condições de folgas complementares são satisfeitas. Passemos a algumas definições relativas a emparelhamentos e, a seguir, à descrição do algoritmo.

Caminho Aumentante

Dado uma rede $R(V,E)$ e o emparelhamento M , dizemos que um caminho P é um *caminho alternado* em relação ao emparelhamento M , se suas arestas são alternadamente arestas emparelhadas e arestas não emparelhadas. Um *caminho aumentante* é um caminho alternado, cujos vértices inicial e final são expostos. Na figura II.1 a seguir, está representada a rede $R(V,E)$ e um emparelhamento M formado pelas arestas em destaque (reticuladas) $\{2,5\}$, $\{3,4\}$, $\{7,10\}$ e $\{8,9\}$. A figura II.2 representa um caminho alternado relativo a este emparelhamento, passando pelos vértices 1, 2, 5, 7 e 10. Finalmente a figura II.3 representa um caminho aumentante ligando os vértices expostos 1 e 6. O algoritmo que vamos apresentar para o problema (EM) baseia-se no teorema enunciado a seguir.

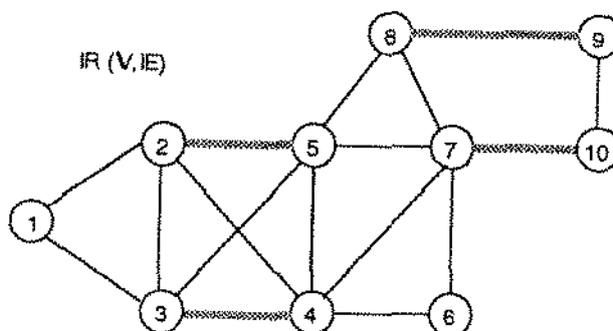


figura II.1

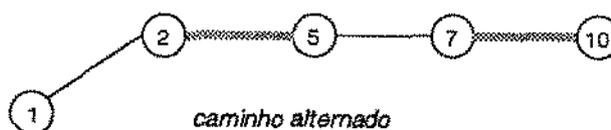


figura II.2

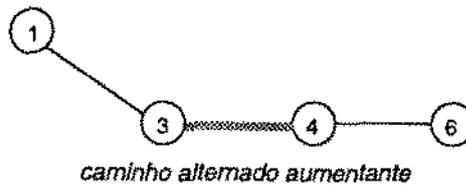


figura II.3

Teorema II.1 (Berge, 1957)

Um emparelhamento M em uma rede $R(V,E)$ é um emparelhamento máximo se e somente se não existe um caminho aumentante em R .

Demonstração: Se existe um caminho aumentante relativo a um emparelhamento M , então é possível criar um novo emparelhamento M' , onde $|M'| = |M| + 1$. Seja K_1 o conjunto de arestas emparelhadas de P e K_2 o conjunto de arestas não emparelhadas de P . Basta fazer $M' = K_2 \cup (M \setminus K_1)$.

Agora, seja um emparelhamento máximo M^* e um emparelhamento M , onde $|M^*| \geq |M| + 1$. Considere o subconjunto de arestas $K = (M \cup M^*) \setminus (M \cap M^*)$. Um vértice $v \in V$ incide no máximo em uma aresta de M . A mesma restrição vale para os vértices de R em relação às arestas do emparelhamento M^* . Portanto K é uma união disjunta de caminhos e ciclos. Cada ciclo contém alternadamente arestas em M e M^* e, portanto, o mesmo número de arestas de ambos. Por outro lado, cada caminho é formado também por arestas alternadamente em M e M^* . Como $|M^*| > |M|$, pelo menos um destes caminhos deverá ser um caminho aumentante em relação a M . Podemos concluir então que, se um emparelhamento qualquer M de uma rede não for um emparelhamento máximo, existe nesta rede um caminho aumentante em relação ao emparelhamento M . Fica, assim, demonstrado o teorema. ■

O teorema acima sugere um algoritmo para (EM). Partindo de um emparelhamento qualquer, realiza-se uma busca para detectar a existência de caminhos aumentantes. Uma vez encontrado um caminho aumentante, determina-se um novo emparelhamento. No exemplo da figura II.4 é realizado uma ampliação do número de arestas da rede R (rede da figura II.1) a partir do caminho aumentante ligando os vértices 1 e 6. A rede da figura II.5 contém o novo emparelhamento obtido com esta ampliação.

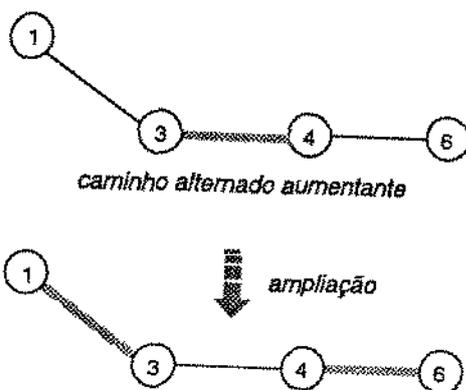


figura II.4

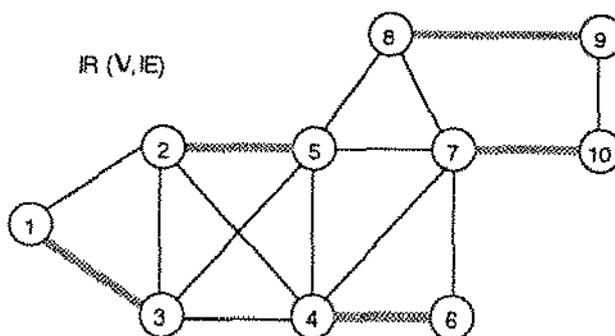


figura II.5

Este processo de busca de caminhos aumentantes e ampliações é repetido até que a transformação resulte em um emparelhamento em relação ao qual não haja nenhum caminho aumentante. Para uma implementação computacional deste processo é preciso definir um procedimento eficiente de busca de tais caminhos.

O algoritmo a seguir baseia-se no crescimento de uma floresta de caminhos alternados. Seja um emparelhamento M e uma rede $R(V,E)$. Uma *árvore alternada* designa uma árvore cuja raiz é um vértice exposto e , além disso, o caminho predecessor de qualquer de seus vértice é formado alternadamente por arestas em M e arestas não pertencentes a M . O algoritmo para (EM-P) consiste em gerar uma floresta de árvores alternadas, cujas raízes são vértices expostos da rede. A seguir é descrito o crescimento desta floresta.

O algoritmo evolui a partir de uma lista L contendo vértices já rotulados. Um

vértice rotulado possui três parâmetros: o primeiro parâmetro representa o antecessor do vértice na árvore alternada a qual pertence; o segundo parâmetro indica se o vértice é um vértice interno (I) ou externo (E); finalmente o último parâmetro é a raiz da árvore a qual pertence.

Primeiro, todo o vértice exposto é rotulado como sendo um vértice externo (E), raiz de uma árvore e colocado na lista L . Se não houver nenhum vértice exposto, o emparelhamento corrente é um emparelhamento máximo. Seja a rede $R(V,E)$ e $r(v)$ a raiz da árvore a qual v pertence. A sondagem dos vértices obedece à seguinte regra.

Sondagem de um vértice externo v . Retire v da lista. Se existe algum vértice externo w tal que $\{v,w\} \in E$, então os caminhos alternados antecessores de v e w , respectivamente, formam, juntamente com a aresta $\{v,w\}$, um caminho aumentante entre os vértices expostos $r(v)$ e $r(w)$. Caso contrário rotule todos os vértices w ainda não rotulados, onde a aresta $\{v,w\} \in E$ e $\{v,w\} \notin M$, sendo M é o emparelhamento corrente. Tais vértices recebem o rótulo $(v,I,r(v))$. Todos estes vértices incidem sobre arestas emparelhadas. Seja $\{w,u\} \in M$. O vértice u recebe o rótulo $(w,E,r(w))$.

Toda vez que um caminho aumentante é encontrado, determina-se um novo emparelhamento. O processo de busca é, então, reiniciado, ou seja, todos os vértices são feitos novamente não rotulados e $L = \emptyset$. Uma nova floresta é então plantada. As raízes das novas árvores, que são vértices expostos da rede corrente, são colocados na lista L . O algoritmo pára em duas circunstâncias. Se após fazer a transformação do caminho aumentante corrente não houver nenhum vértice exposto, então o emparelhamento encontrado é um emparelhamento máximo. Neste caso o emparelhamento encontrado é um emparelhamento perfeito, ou seja, $x(\delta(v)) = 1, \forall v \in V$. Outra situação de parada ocorre quando a floresta corrente não pode mais crescer, ou seja, não existe mais nenhum vértice na lista L para ser sondado e nenhum caminho aumentante é encontrado. Tal floresta é chamada *Floresta Húngara*.

Teorema II.2 (Nemhauser & Laurence, 1988)

O algoritmo acima produz um emparelhamento máximo em uma rede bipartida.

Demonstração: Seja M^* o emparelhamento obtido ao final do algoritmo e x^* o vetor de incidência associado. Vamos determinar uma solução dual factível, cujo valor da função objetivo correspondente é igual a $|M^*|$.

Quando o algoritmo, termina tem-se um conjunto de árvores alternadas

$T_i=(V_i,E_i)$, $i=1,2,\dots,s-1$, e também um conjunto de vértices não rotulados V_s . Como nenhum vértice em V_s é exposto, a subrede gerada por V_s , (V_s,E_s) , contém um emparelhamento perfeito, ou seja, $x^*(\delta(v)) = 1$, $\forall v \in V_s$. Veja figura II.6 a seguir. Seja (V_s^1,V_s^2) uma partição de V_s tal que nenhuma aresta emparelhada de M^* incida em V_s^1 ou V_s^2 . Seja também V_i^1 o conjunto de vértice internos da árvore T_i . Se π^* é o vetor de incidência do conjunto $W^* = (\bigcup_{i=1}^{s-1} V_i^1) \cup V_s^1$ (ou seja, π_v^* é igual a um se $v \in W^*$ e zero caso contrário), então o vetor (π^*,μ^*) , com $\mu_y^* = 0$ para todo conjunto ímpar Y , é uma solução dual factível. Para ver isto, basta observar os caso listados a seguir.

- a) Toda aresta pertencente a uma árvore incide em um vértice de W^* .
- b) Não existem arestas ligando vértices externos. Caso contrário, haveria um ciclo na rede $R(V,E)$ que, por hipótese, é uma rede bipartida.
- c) Não pode haver uma aresta ligando vértices externos de árvores diferentes. Neste caso, um caminho aumentante teria sido detectado.
- d) Não pode haver uma aresta ligando um vértice externo e um vértice não rotulado. Caso contrário, o vértice não rotulado teria sido rotulado a partir deste vértice.

Como $\sum_v \pi_v = |W^*| = |M^*| = \sum_a x_a$, o emparelhamento M^* é o emparelhamento máximo e o teorema está demonstrado. ■

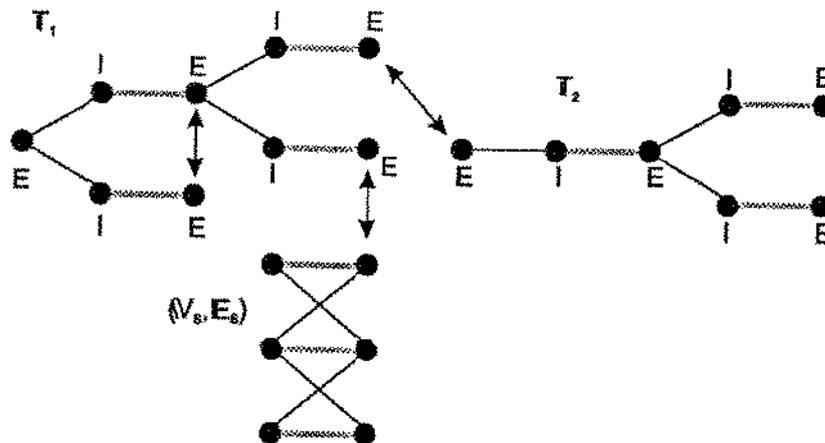
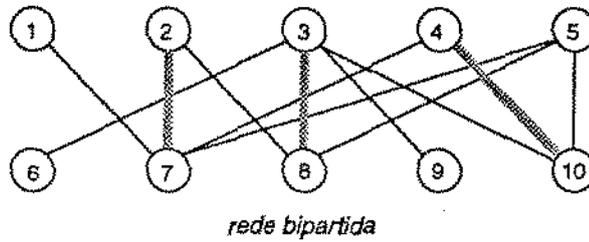


figura II.6

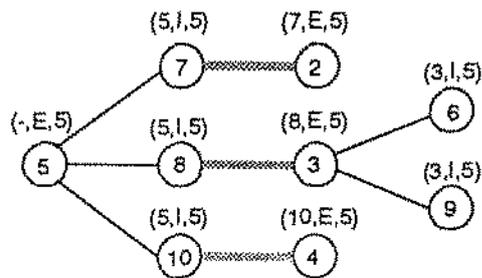
Aplicando o algoritmo descrito anteriormente à rede bipartida da figura II.7 obtém-se a árvore alternada da figura II.8. Nesta árvore existem dois caminhos

aumentantes possíveis, um ligando a raiz 5 ao vértice 6 e outro ligando a mesma raiz ao vértice 9.



rede bipartida

figura II.7



caminho aumentante (5, 8, 3, 6)

figura II.8

Realizando uma ampliação em relação ao caminho aumentante do vértice 1 ao 6, obtem-se o emparelhamento representado na figura II.9. Crescendo uma nova floresta, sendo a raiz de uma das árvores o vértice 5 e a raiz da outra igual ao vértice 9, obtem-se a floresta alternada da figura II.10. A partir desta floresta não é mais possível rotular nenhum vértice e, portanto, esta é dita ser uma floresta húngara. O emparelhamento corrente é um emparelhamento de cardinalidade máxima.

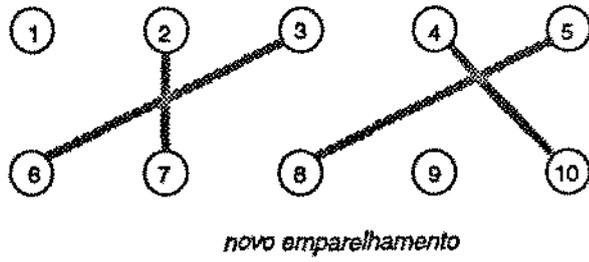


figura II.9

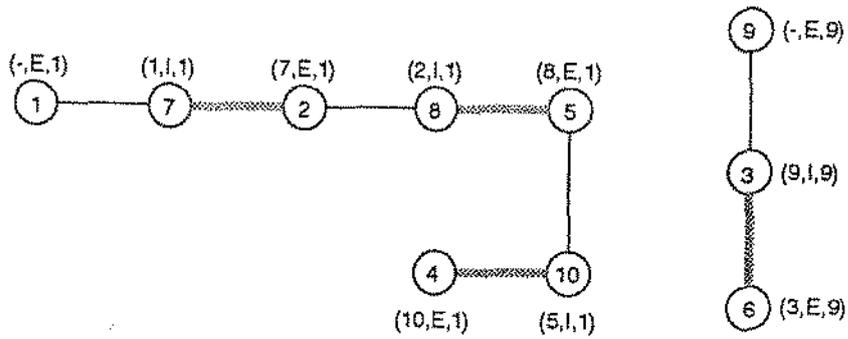


figura II.10

O que impede que o algoritmo acima seja aplicado a uma rede não bipartida é a possível existência de ciclos ímpares, ou seja, ciclos com um número ímpar de vértices. Considere este algoritmo aplicado à rede da figura II.11.

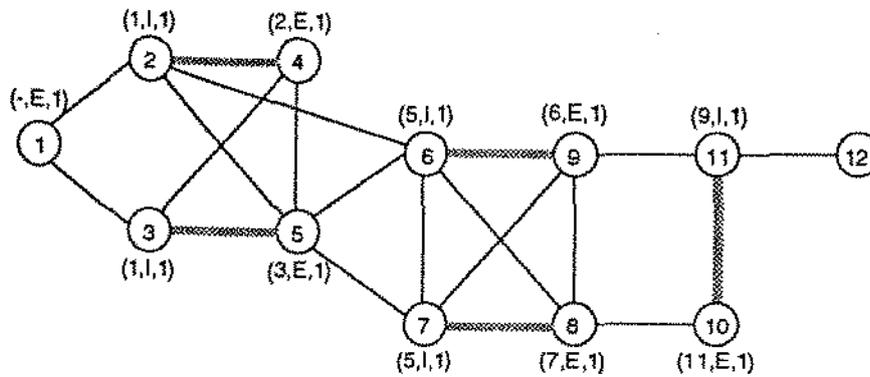


figura II.11

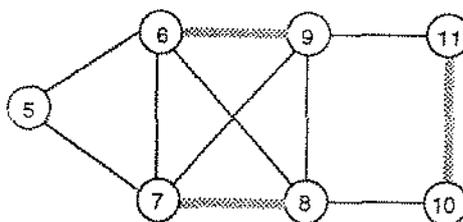
O algoritmo termina sem detectar o caminho aumentante ligando os vértices 1 e 12. Veja que neste ponto existem dois vértices externos, 10 e 8, ligados por uma aresta, o que não ocorreria se a rede fosse bipartida.

Considere a subrede gerada pelos vértices $\{5,6,7,8,9,10,11\}$ na figura II.12. Definiremos uma subrede com estas características como sendo um *anel simples*. Veja a definição a seguir.

Anel Simples. Seja uma rede não orientada $R(V,E)$ e um emparelhamento M sobre R . Um *anel simples* é uma subrede $B(V_b, E_b)$ onde E_b é gerado pelo subconjunto de vértices ímpar V_b com $|V_b| \geq 3$, satisfazendo às seguintes propriedades:

a) Existe exatamente um vértice, definido como sendo a base do anel, no qual não incide nenhuma aresta emparelhada de E_b .

b) Existe um ciclo em E_b que passa por todos os vértices em V_b , tal que as arestas incidentes na base são arestas não emparelhadas e as arestas restantes são, alternadamente, emparelhadas e não emparelhadas. A base do anel pode ser ou não incidente em uma aresta emparelhada de V .



anel simples

figura II.12

Suponha que a evolução do processo de sondagem dos vértices no algoritmo descrito anteriormente gere uma floresta de árvores alteradas em que hajam dois vértices externos pertencentes à mesma árvore. Sejam v e w estes dois vértices. Os caminhos alternados que os ligam a raiz se cruzam em um vértice b . Este vértice é, necessariamente, um vértice externo, pois dele emanam pelo menos duas arestas não emparelhadas. O vértice b pode ser ou não um vértice exposto. Se b for um vértice exposto, então b é também raiz da árvore a qual v e w pertencem. Os caminhos desta árvore que ligam v,b e w,b respectivamente, juntamente com a aresta $\{v,w\}$ formam um ciclo com um número ímpar de vértices. A subrede gerada pelos vértices deste ciclo é um anel simples, e o vértice b , no caso, é a base deste anel.

Seja a rede $R(V,E)$. Se, ao aplicar o algoritmo para (PCC) em R , um anel $B(V_b,E_b)$ for detectado a partir de dois vértices externos v e w da mesma árvore, é realizada uma operação chamada *contração*.

Contração

O procedimento de contração de um anel $B(V_b,E_b)$ consiste em criar uma nova rede $R'(V',E')$ a partir da rede corrente $R(V,E)$, onde os vértices do anel são substituídos por um único vértice, P , que chamaremos *pseudonó*. O subconjunto de vértices V' é constituído pelos vértices de V , exceto aqueles pertencentes a V_b , mais o vértice P . Para toda aresta $\{v,w\}$ de R com $w \in V_b$ e $v \notin V_b$, cria-se uma aresta em R' definida pelo par $\{v,P\}$. Além destas arestas, o conjunto de arestas E' da nova rede R' é formado pelas arestas da subrede gerada por V/V_b . As arestas geradas por V_b não existem em E' . O novo emparelhamento M' correspondente a M é formado pelas arestas de M que pertencem a R' e a M . Se a base b do pseudonó P incide sobre a aresta emparelhada $\{b,i\}$, então a aresta $\{b,P\}$ de R' também pertence a M' . Uma vez feita a contração do anel, o pseudonó P é colocado na lista L de vértices a serem sondados. Um pseudonó será sempre um vértice externo. Repare que, após a contração do anel B , temos $|V'| = |V| - |V_b| + 1$, e o número de vértices expostos em R' é igual ao número de vértices expostos em R , pois se a base b é um vértice exposto, então o pseudonó P será exposto.

A figura II.13 é um exemplo de contração. A subrede gerada pelas arestas e_1, e_2, e_3, e_4, e_5 e e_6 constitui o anel contraído $B(V_b,E_b)$. Com esta operação, o anel

simples \mathbb{B} é substituído pelo pseudonó P , que recebe o mesmo rótulo da base b de \mathbb{B} .

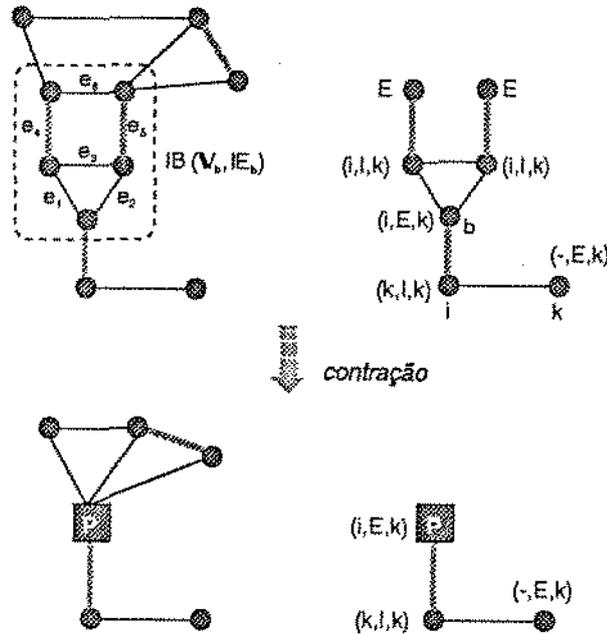


figura II.13

O crescimento da floresta continua após a contração de um anel. Neste processo, um novo anel \mathbb{B}' pode ser encontrado. Suponha que um dos vértices de \mathbb{B}' seja um pseudonó. Neste caso, quando \mathbb{B}' é contraído, o pseudonó criado terá um outro pseudonó dentro de si. Suponha, agora, que a base de \mathbb{B}' seja um pseudonó e que \mathbb{B}' seja contraído formando um pseudonó P' . Diante desta perspectiva vamos definir o que seja o *ápice* de um pseudonó. Se a base de um pseudonó for um vértice original, então o *ápice* deste pseudonó é igual a sua base. Por outro lado, se a base deste pseudonó for também um pseudonó, o *ápice* do pseudonó externo é igual ao *ápice* de sua base.

A definição de anel simples diz respeito apenas a vértices da rede original \mathbb{R} , no entanto a definição abaixo é mais genérica e corresponde à situação descrita anteriormente, ou seja, quando o anel a ser contraído possui pseudonós.

Anel. Seja uma rede não orientada $R(V,E)$ e um emparelhamento M sobre R . Um *anel* é uma subrede $B(V_b, E_b)$ onde E_b é gerado pelo subconjunto de vértices ímpares V_b com $|V_b| \geq 3$, satisfazendo às seguintes propriedades:

- a) O conjunto de arestas $M_b = M \cap E_b$ é um emparelhamento máximo em B .
- b) Exatamente um vértice é não emparelhado. Este único vértice, ap , é chamado ápice de B . Veja que o ápice ap pode ser um vértice emparelhado (em R), sendo que neste caso, a aresta emparelhada que nele incide não pertence a E_b .

Seja $R_k(V_k, E_k)$ a rede corrente obtida com a contração de k anéis. Vamos chamar os vértices do conjunto V_k de vértices correntes. Estes podem ser pseudonós ou simplesmente vértices da rede original R , que chamarei de vértices originais.

A seguir é descrito um procedimento recursivo que amplia a cardinalidade do emparelhamento M da rede original $R(V,E)$ a partir de um caminho aumentante encontrado na rede R_k .

Ampliação

Seja P um pseudonó pertencente a um caminho aumentante na rede corrente R_k e w o vértice adjacente a P por uma aresta não emparelhada do caminho aumentante. Seja R'_k a rede obtida a partir de R_k com a substituição de P pelo anel simples B a ele associado e b a base deste anel. Considere agora o vértice b' em B , adjacente ao vértice w . Por construção, existe um caminho alternante de comprimento par ligando b e b' . Substitua P por este caminho. O novo caminho é um caminho aumentante em R'_k . Este procedimento é repetido para cada pseudonó no caminho aumentante, inclusive os pseudonós que reaparecem quando um anel é substituído por um caminho alternado. O exemplo da figura II.14 representa a substituição do pseudonó P pelo caminho alternante, de comprimento par, ligando os vértices b e b' em P .

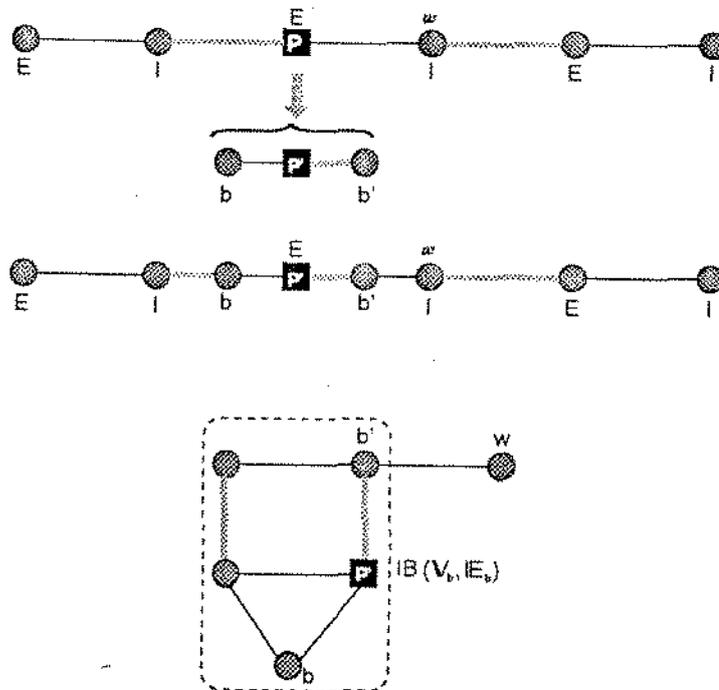


figura II.14

Em algoritmos de emparelhamento, a contração e ampliação são procedimentos executados repetidas vezes. Quando um pseudonó é formado, ele passa a ser um vértice corrente da rede naquele estágio. Posteriormente, este pseudonó pode ser absorvido por outro pseudonó e, neste caso, dizemos que o pseudonó interno se tornou um *pseudonó dormente*. Se um pseudonó é um vértice da rede corrente, dizemos que ele é um *pseudonó corrente*.

Vértices originais são vértices em V da rede original $R(V, E)$. Estes vértices são definidos como sendo *vértice de nível 0 (zero)*. Um pseudonó, cujos vértices são todos vértices de nível 0, é chamado *pseudonó de nível 1* (neste caso o anel correspondente é um anel simples). Generalizando, um pseudonó para o qual pelo menos um de seus vértices internos é um pseudonó de nível $s - 1$ ($s \geq 1$) e todos os demais são pseudonós de nível menor que $(s - 1)$ é dito ser um *pseudonó de nível s* .

Teorema II.3 (Katta G. Murty, 1992)

O número total de pseudonós de todos os níveis, que podem ser pseudonós correntes ou pseudonós dormentes, é sempre menor ou igual a $|V|/2$, onde $|V|$ é o número de vértices da rede original.

Demonstração: Basta lembrar que a rede obtida com uma operação de contração tem um número par de vértice a menos que a rede contendo o anel simples contraído.■

O algoritmo abaixo resolve o problema (EM) para uma rede não orientada $R(V,E)$. Em relação ao algoritmo anterior descrito, foram acrescentados os procedimentos de contração e ampliação. A raiz da árvore a qual o vértice v pertence é denotada por $r(v)$ e o antecessor de v por $p(v)$.

Algoritmo de Emparelhamento para (EM).

Passo 1 Inicialização Escolha um emparelhamento inicial (pode ser \emptyset) em \mathbb{R} .

Passo 2 Plantar a Floresta Alternada Se não existem vértices expostos vá para o passo 7. Caso contrário, faça com que cada vértice exposto v seja raiz de uma árvore alternada, atribuindo a v o rótulo (v, E, v) . Coloque v na lista L .

Passo 3 Selecione um Vértice da Lista L Se $L = \emptyset$, vá para o passo 7. Caso contrário selecione um vértice de L e retire-o da lista.

Passo 4 Sondagem Seja v o vértice externo a ser sondado e $(p(v), E, r(v))$ seu rótulo. Para cada vértice w , $p(v) \neq w$ tal que $\{v, w\}$ é uma aresta corrente faça o seguinte.

1) Se $(p(w), E, r(w))$ é o rótulo de w e $r(w) \neq r(v)$, ou seja, w é um vértice externo não pertencente à mesma árvore de v , então um caminho aumentante foi encontrado. Vá para o passo 5.

2) Se $(p(w), E, r(w))$ é o rótulo de w e $r(w) = r(v)$, ou seja, w é um vértice externo pertencente à mesma árvore de v , então um anel foi detectado. Vá para o passo 6.

3) Se w for um vértice ainda não rotulado dê a ele o rótulo $(v, I, r(v))$. Sobre w necessariamente incide uma aresta emparelhada. Seja $\{w, u\}$ esta aresta. Faça o vértice u receber o rótulo $(w, E, r(w))$ e coloque-o na lista L .

Passo 5 Ampliação A partir do caminho aumentante detectado com a sondagem dos vértices da rede corrente, faz-se uma transformação sobre o emparelhamento corrente. A cardinalidade do novo emparelhamento é superior ao emparelhamento anterior. Volte para 2.

Passo 6 Contração Uma vez detectado um anel simples \mathbb{B} na rede corrente, aplicar o procedimento para fazer a contração do mesmo e criar, assim, um pseudonó. Utiliza-se o procedimento de contração descrito anteriormente. Uma vez criada a nova rede com o pseudonó P , dê a P o mesmo rótulo que possuía a base de \mathbb{B} e coloque P na lista L . Volte ao passo 3.

Passo 7 Fim Quando chegamos a este ponto, o emparelhamento corrente é um emparelhamento máximo. Se o emparelhamento não for um emparelhamento perfeito em \mathbb{R} , significa que a floresta alternada da rede corrente é uma Floresta Húngara. O emparelhamento obtido é um emparelhamento máximo. Pare.

Cada vez que um caminho aumentante é detectado e o passo 5 executado, a cardinalidade do emparelhamento aumenta de um. Portanto o passo 5 é executado pelo algoritmo no máximo $|V|/2$ vezes. Entre duas execuções do passo 5, o passo 6 é executado no máximo $|V|/2$ vezes, (teorema II.3), e o passo 3 é executado no máximo $O(|V|)$ vezes. Cada execução do passo 3 requer um esforço igual a $O(|V|)$. Cada execução dos passos 5 ou 6 requer o traçado do caminho predecessor até a raiz, o que equivale a um esforço proporcional a $O(|V|)$. Assim, o esforço feito entre duas execuções do passo 5 é proporcional a $O(|V|^2)$ e, portanto, a complexidade do

algoritmo para (EM), no pior caso é igual a $O(|V|^3)$.

11.2 EMPARELHAMENTO PERFEITO ÓTIMO (EPO)

Vamos considerar agora, o problema de determinar um emparelhamento perfeito ótimo (EPO) sobre uma rede $R(V,E)$. Seja c_a o custo associado à aresta $a \in E$ e x o vetor de incidência do emparelhamento M .

$$\begin{aligned} \text{(EPO)} \quad & \text{minimizar } \sum_a c_a x_a \\ & \text{restrito a (4) :} \\ & x(\delta(v)) = 1, \forall v \in V \\ & x_a \geq 0, \forall a \in E \\ & x_a \text{ inteiro } \forall a \in E \end{aligned}$$

Considere a seguinte formulação linear (EPO-P).

$$\begin{aligned} \text{(EPO-P)} \quad & \text{minimizar } \sum_a c_a x_a \\ & \text{restrito a (5):} \\ & x(\delta(v)) = 1, \forall v \in V. \\ & x(\gamma(Y)) \leq (|Y|-1)/2, \forall Y \in F \\ & x_a \geq 0, \forall a \in E \end{aligned}$$

onde F é a família de todos os subconjuntos ímpares.

Qualquer emparelhamento satisfazendo às condições em (4), satisfaz também às condições em (5). Portanto, se a solução ótima do problema (EPO-P) é uma solução inteira, será também solução ótima do problema (EPO). Vamos apresentar, a seguir, um algoritmo primal-dual que resolve (EPO), e mostrar que, se existe um emparelhamento perfeito na rede $R(V,E)$, então este algoritmo termina com um emparelhamento perfeito ótimo em R .

Seja π_v a variável dual associada à restrição sobre o vértice v na formulação (EPO-P) e μ_Y a variável dual associada à restrição sobre o conjunto ímpar Y . Veja que o tamanho do vetor μ depende exponencialmente de $|V|$. Veremos mais tarde, no entanto, que não mais do que $|V|/2$ de suas componentes terão valor diferente de zero.

Considere as seguintes definições dadas a partir da solução dual (π, μ) :

$$\mu(a) = \sum (\mu_y; a \in \gamma(Y_y))$$

$$d_a^-(\pi, \mu) = \pi(a) - \mu(a)$$

O dual de (EPO-P) é

$$(EPO-D) \quad \text{maximizar } \sum_v \pi_v - \sum_{Y \in F} (|Y|-1)(\mu_Y)/2$$

restrito a (6):

$$d_a^-(\pi, \mu) \leq c_a, \quad \forall a \in E$$

$$\mu \geq 0$$

Seja $E^*(\pi, \mu) = \{a \in E : d_a^-(\pi, \mu) = c_a\}$, ou seja, $E^*(\pi, \mu)$ é o conjunto das arestas associadas às restrições satisfeitas pela igualdade em (4). As condições de folgas complementares de (EPO-P) e (EPO-D) são

$$x_a > 0 \Rightarrow d_a^-(\pi, \mu) = c_a, \quad \forall a \in E$$

$$\mu_Y > 0 \Rightarrow x(\gamma(Y)) = (|Y|-1)/2, \quad \forall Y \in F$$

O algoritmo do problema (EPO-P) começa com uma solução dual (π^0, μ^0) com $\mu^0 = 0$. É aplicado, então, o algoritmo de emparelhamento máximo para a subrede R^* gerada pelas arestas em $E^*(\pi^0, \mu^0)$. Neste processo, alguns anéis podem ser detectados e contraídos, formando pseudonós. Se o emparelhamento encontrado satisfaz (4), então este é, também, um emparelhamento perfeito ótimo, pois satisfaz às condições de folgas complementares.

A fase primal do algoritmo de (EPO) corresponde à busca de uma solução primal x satisfazendo (1) sobre a subrede de igualdade R^* . Se este processo termina sem encontrar um emparelhamento perfeito, ou seja, com uma solução primal infactível, é feita uma transformação nas variáveis duais.

No algoritmo do problema (EM), sempre que um caminho aumentante é detectado, podemos eliminar todos os pseudonós e plantar uma nova floresta. Em outras palavras, voltamos à rede original, agora com um novo emparelhamento, e plantamos as árvores alternadas a partir dos vértices expostos relativos ao novo emparelhamento. Neste caso, todos os pseudonós são vértices externos. No algoritmo de (EPO), os pseudonós deverão ser mantidos até que a variável dual associada seja zero, caso contrário as

condições de folgas complementares seriam desrespeitadas. Neste sentido, a transformação do emparelhamento, a partir de um caminho aumentante, deve fazer a atualização da base de cada pseudonó contido no caminho. O exemplo da figura II.15 corresponde às situações em que, primeiro, a base de um pseudonó contido em um caminho aumentante não precisa ser atualizada e, segundo, a base de um pseudonó contido em um caminho aumentante precisa ser atualizada.

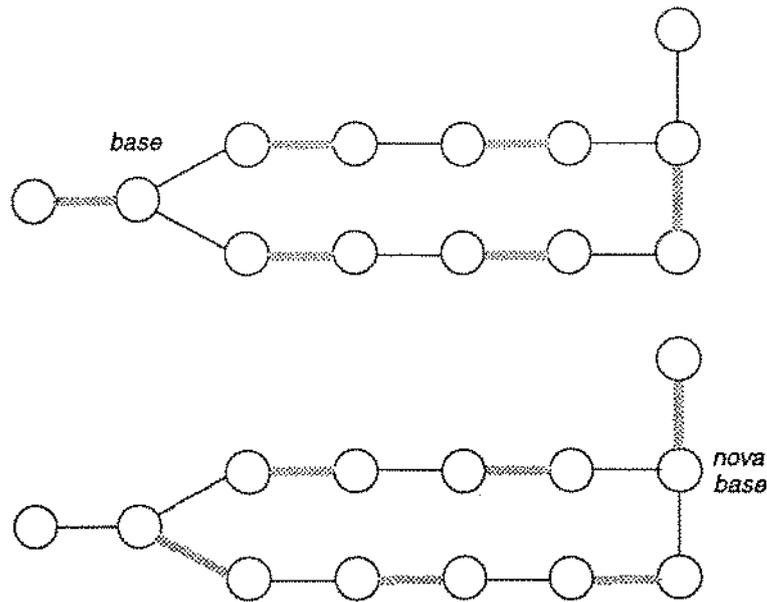
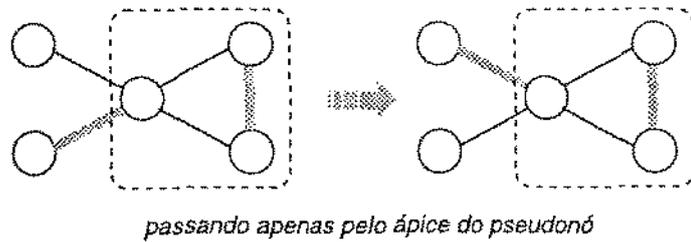


figura II.15

Ao crescer uma nova árvore, alguns pseudonós podem ser rotulados como vértices internos da rede corrente. No entanto, a presença de pseudonós internos não garante que a fase primal do algoritmo termine com uma solução ótima de (1). A transformação dual procura, entre outras coisas, fazer o custo reduzido dos pseudonós internos igual a zero.

Expansão de um Pseudonó

A operação de expansão de um pseudonó é realizada apenas sobre pseudonós internos associados a $\mu_y = 0$. Seja P o pseudonó com rótulo $(a, l, r(P))$ e \mathbb{B} o anel simples correspondente. Então, a é um vértice externo na floresta alternada da rede corrente e $\{a, P\}$ uma aresta não emparelhada. Seja $\{P, b\}$ a aresta emparelhada incidente em P . Existe um vértice v_1 em \mathbb{B} , tal que $\{a, v\}$ é uma aresta da rede original com custo reduzido igual a zero. Podem ocorrer duas situações: ou v é base de P e, neste caso, o pseudonó é desfeito e sua base recebe o rótulo de P , ou existe um caminho alternado de comprimento par começando a partir da aresta emparelhada incidente em v , ligando v à base de \mathbb{B} . Neste caso, a operação de expansão de P consiste em substituí-lo por este caminho, rotulando cada vértice nele contido, alternadamente, como um vértice interno (I) e um vértice externo (E). O exemplo da figura II.16 corresponde à expansão de um pseudonó interno P .

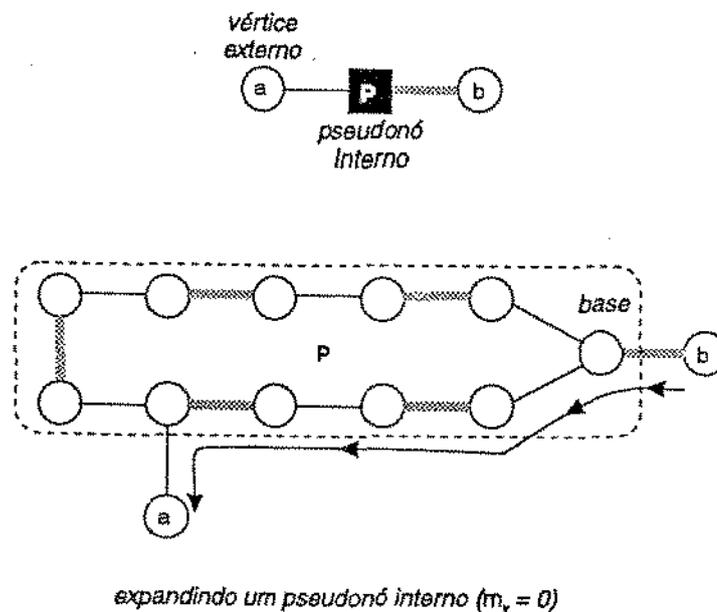


figura II.16

Transformação Dual (7)

$$\delta_1 = \min \{c_a - d_a^-(\pi, \mu) : a = \{v, w\} \text{ com } v \text{ dentro de um pseudonó corrente externo e um vértice dentro de um pseudonó não rotulado}\}$$

$$\delta_2 = \min \{(1/2)(c_a - d_a^-(\pi, \mu)) : a = \{v, w\} \text{ com } v \text{ e } w \text{ dentro de pseudonós correntes distintos}\}$$

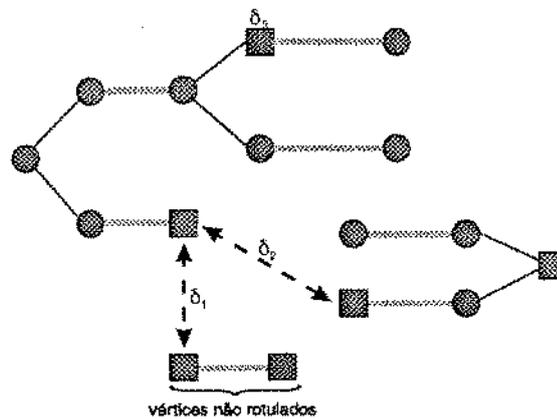
$$\delta_3 = \min \{(1/2)\mu_y : y \text{ é o conjunto de vértices originais dentro de um pseudonó interno}\}$$

$$\delta = \min \{\delta_1, \delta_2, \delta_3\}$$

Se $\delta = +\infty$ a rede original R não possui um emparelhamento perfeito e o problema é, portanto, infactível.

$$\pi_v = \begin{cases} \pi_v + \delta, & v \text{ dentro de um pseudonó externo.} \\ \pi_v - \delta, & v \text{ dentro de um pseudonó interno.} \\ \pi_v, & v \text{ dentro de um pseudonó não rotulado.} \end{cases}$$

$$\mu_y = \begin{cases} \mu_y + 2\delta, & \mu_y \text{ associado a um pseudonó externo.} \\ \mu_y - 2\delta, & \mu_y \text{ associado a um pseudonó interno.} \\ \mu_y, & \text{caso contrário.} \end{cases}$$



transformação dual

figura II.17

Algoritmo para o Emparelhamento Perfeito Ótimo (EPO)

Passo 1 Inicialização Comece com a solução dual factível (π^0, μ^0) tal que $\mu^0 = 0$ e $\pi_v^0 = (1/2)(\min_a \{c_a : a \in E\})$ para cada $v \in V$. Escolha um emparelhamento inicial qualquer (pode ser o emparelhamento \emptyset) para a subrede $E^*(\pi^0, \mu^0)$.

Passo 2 Plantar a Floresta Alternada Se não existem vértices expostos vá, para 10. Caso contrário, faça com que cada vértice exposto v seja raiz de uma árvore alternada e (v, E, v) seu rótulo. Coloque todos estes vértices na lista L .

Passo 3 Selecione um Vértice da Lista L Se $L = \emptyset$ vá para o passo 8. Caso contrário, selecione um vértice da lista L .

Passo 4 Sondagem Seja v o vértice externo a ser sondado e $(p(v), E, r(v))$ seu rótulo. Para cada vértice w com $p(v) \neq w$, tal que $\{v, w\}$ é uma aresta com custo reduzido zero, faça o mesmo que foi definido no passo 4 do algoritmo de (EM).

Passo 5 Ampliação A partir do caminho aumentante encontrado, determine o novo emparelhamento. A base de cada pseudonó neste caminho deve ser atualizada. Vá para o passo 2.

Passo 6 Contração Um anel foi detectado. Faça a sua contração e coloque o pseudonó recém criado na lista L . Vá para o passo 3.

Passo 7 Expansão Realize a expansão do pseudonó interno com custo reduzido zero. Vá para o passo 3.

Passo 8 Transformação Dual Determine uma nova solução dual a partir da transformação dual definida em (7). Se $\delta = +\infty$ vá para o passo 9. Se $\delta = \delta_3$ vá para o passo 7. Caso contrário vá para 3.

Passo 9 Infactibilidade Chega-se a este ponto quando $\delta = +\infty$. Neste caso, não existe um emparelhamento perfeito e, portanto, (EPO) é infactível.

Passo 10 Otimalidade Quando este ponto é atingido o emparelhamento corrente é um emparelhamento perfeito ótimo. Pare.

Validade do Método

Teorema II.4 (Katta G. Murty, 1992)

Se a solução dual corrente (π, μ) é factível, então o valor δ calculado no passo 8 é estritamente positivo ou igual a $+\infty$.

Demonstração: O passo 8 é executado somente quando a fase primal termina com uma Floresta Húngara. Este fato, juntamente com a factibilidade da solução dual corrente (π, μ) , implica que cada um dos conjuntos sobre os quais $\delta_1, \delta_2, \delta_3$ são mínimos, é vazio, ou consiste de elementos estritamente positivos. ■

Teorema II.5 (Katta G. Murty, 1992)

Se (π, μ) é uma solução dual factível, o vetor (π', μ') obtido com a transformação dual (7) é também dual factível.

Demonstração: Pelo teorema II.4, $0 < \delta \leq \delta_3$. Como $\mu \geq 0$, temos que $\mu' \geq 0$. Seja $a \in E$ uma aresta original e $a \in \gamma(E_b)$ onde E_b é o conjunto de vértices de um anel associado a um pseudonó corrente P . Independentemente de P ser rotulado ou não, valem as desigualdades $d_a^-(\pi', \mu') = d_a^-(\pi, \mu) \leq c_a$, sendo a última desigualdade válida porque (π, μ) é factível. Portanto, o conjunto $E^*(\pi, \mu)$ não se altera em relação a estas arestas.

Considere agora, a aresta $a \in E$, $a = \{v, w\}$ onde v e w são vértices originais contidos (ou iguais) a vértices correntes distintos V e W respectivamente. Se V é um vértice externo e W interno, então

$$d_a^-(\pi', \mu') = \pi'_v + \pi'_w = \pi_v + \pi_w = d_a^-(\pi, \mu) \leq c_a$$

Se ambos V e W são externos, então

$$d_a^-(\pi', \mu') = \pi'_v + \pi'_w = \pi_v + \pi_w + 2\delta = d_a^-(\pi, \mu) \leq c_a,$$

pois $\delta \leq \delta_2$ e (π, μ) é factível. Se V é externo e W não rotulado, então

$$d_a^-(\pi', \mu') = \pi'_v + \pi'_w = \pi_v + \pi_w + \delta = d_a^-(\pi, \mu) \leq c_a,$$

pois $\delta \leq \delta_1$ e (π, μ) é factível. Se V e W são ambos não rotulados ou ambos internos, então $d_2^-(\pi', \mu') = \pi'_v + \pi'_w$ é igual a $\pi_v + \pi_w - \delta$ ou $\pi_v + \pi_w - 2\delta$ respectivamente, e $\pi_v + \pi_w = d_2^-(\pi, \mu) \leq c_a$. Como $\delta > 0$ temos $d_2^-(\pi', \mu') \leq c_a$. ■

Do teorema II.4 concluímos que, partindo da solução (π^0, μ^0) , que é claramente dual factível, o algoritmo evolui por soluções dual factíveis. Veja também que as arestas incidentes em vértices correntes externos e internos, simultaneamente, não têm seu custo reduzido alterado. Além disso, as arestas pertencentes às árvores e arestas emparelhadas, permanecem em $E^*(\pi, \mu)$. Portanto, depois de cada transformação dual a floresta alternada permanece contida em $E^*(\pi, \mu)$ e o algoritmo retoma o processo de sondagem dos vértices correntes.

Neste ponto uma das seguintes situações ocorre:

- a) Se $\delta = \delta_1$, pelo menos uma aresta ligando um vértice corrente externo e um vértice corrente interno passa a ter custo reduzido igual a zero. Um vértice não rotulado é rotulado e o crescimento da floresta alternada continua.
- b) Se $\delta = \delta_2$, pelo menos uma aresta ligando dois vértices correntes externos passa a ter custo reduzido igual a zero. É feita a contração de um pseudonó e o crescimento da floresta alternada continua.
- c) Se $\delta = \delta_3$, o custo reduzido de um pseudonó interno passa a ser zero. É feita sua expansão e o crescimento da floresta alternada continua.
- d) Se $\delta = +\infty$ o problema (EPO) é infactível.

Análise de Complexidade

Considere que uma iteração no algoritmo de (EPO) ocorra entre duas ampliações consecutivas. Neste caso, ocorrem no máximo $|V|/2$ iterações. Vamos fazer a análise de complexidade das operações realizadas entre os limites de uma operação.

Apesar do algoritmo realizar a expansão de pseudonós, esta operação só é feita sobre pseudonós que completaram pelo menos uma iteração. Isto por que um pseudonó, antes de ser expandido, deverá primeiro se tornar não rotulado, o que só acontece após uma ampliação. Uma vez criado um pseudonó, ele pode ser absorvido por outros através de novas contrações, porém permanece como tal até o fim da iteração na qual foi criado. Portanto, pelo teorema II.3, a operação de contração é executada no máximo $|V|/2$ vezes em cada iteração.

Como somente pseudonós formados em iterações anteriores à iteração corrente podem ser expandidos, também pelo teorema II.3, a operação de expansão é executada no máximo $|V|/2$ vezes em cada iteração.

Seja K_E o conjunto de todos os vértices originais contidos em vértices externos e K_I o conjunto de vértices correntes ímpares. Neste caso, $|K_E| + |K_I| \leq |V|$.

Depois de cada transformação dual, o valor δ pode ser igual a $+\infty$, δ_1 , δ_2 ou δ_3 . Se δ for igual a $+\infty$, o algoritmo pára.

Se δ igual a δ_2 , pelo menos um pseudonó é criado. Todos os pseudonós internos dentro do anel contraído não pertencem mais a K_I , no entanto todos os vértices internos destes pseudonós passaram a pertencer ao conjunto K_E . Portanto, ou $|K_E| + |K_I|$ permanece igual, ou aumenta. Como podem ocorrer no máximo $|V|/2$ contrações durante uma iteração, o número de transformações duais com δ igual a δ_2 é, no máximo, igual a $|V|/2$.

Se δ igual a δ_3 , pelo menos um pseudonó interno será expandido. Depois da expansão este vértice não mais pertencerá a K_I . No entanto, pelo menos três vértices no anel simples correspondente serão rotulados, pelo menos dois como vértices internos e pelo menos um como vértice externo. Portanto, depois da expansão, $|K_I|$ aumenta em pelo menos um e $|K_E|$ aumenta em pelo menos um. Logo, $|K_E| + |K_I|$ aumenta em pelo menos dois.

Se δ igual a δ_1 na transformação dual, então, pelo menos um vértice corrente não rotulado será rotulado interno e $|K_I|$ incrementa no mínimo de um. O valor $|K_E|$ também incrementa de pelo menos um e, portanto, $|K_E| + |K_I|$ incrementa no mínimo de dois. Portanto, cada transformação dual em que δ é igual a δ_1 ou δ_3 , incrementa $|K_E| + |K_I|$ de pelo menos dois, e, logo, o número máximo de tais transformações duais em cada iteração é no máximo igual a $|V|$.

A complexidade das operações de contração, expansão e das transformações duais é claramente igual a $O(|E|)$. Logo, a complexidade das operações realizadas a cada iteração é $O(|V||E|)$. Assim, o esforço computacional do algoritmo é, no pior caso, proporcional a $O(|V|^2|E|)$.

CAPÍTULO III
MÉTODO DE SOLUÇÃO DO PROBLEMA DO CARTEIRO CHINÊS EM REDE NÃO ORIENTADA

Neste capítulo vamos discutir aspectos da implementação computacional do método de solução do problema do carteiro chinês para uma rede não orientada. Foi utilizada a linguagem de programação C++ versão 2.0 da AT&T e uma estação de trabalho da SUN.

III.1 PROGRAMA PRINCIPAL

Passamos a seguir à descrição das classes que fazem parte da estrutura de dados do programa.

Dois classes são utilizadas para a leitura e manipulação da rede: *vertex* e *graph*. A classe *vertex* é uma "hash table" linear usada para guardar o nome dos vértices da rede da maneira como são lidos. À medida em que é feita a leitura das arestas, o nome dos vértices é armazenado em *vertex* e a cada um associado um número. A classe *vertex* contém, também, um vetor com a relação inversa, ou seja, dado o número do vértice, este vetor retorna a posição correspondente na "hash table". A figura III.1 é um exemplo de "hash table" linear (a "hash function" é linear).

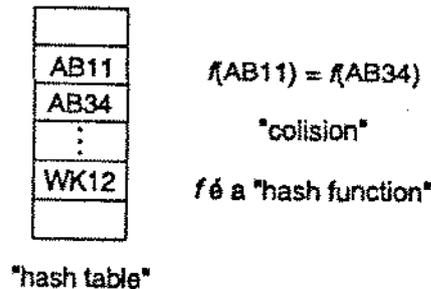


figura III.1

A classe *graph* é constituída por um vetor de listas, que a cada vértice associa uma lista contendo as arestas nele incidentes, e uma tabela que, para cada aresta associa quatro valores: o primeiro valor corresponde à sua cauda, o segundo, sua cabeça, o terceiro, seu tamanho, e, finalmente, um quarto valor que representa o número de vezes que a aresta é percorrida na solução; inicialmente este valor é igual a um, se a aresta deve necessariamente ser percorrida, e zero caso contrário. A figura III.2 é um exemplo da estrutura de dados implementada na classe *graph*.

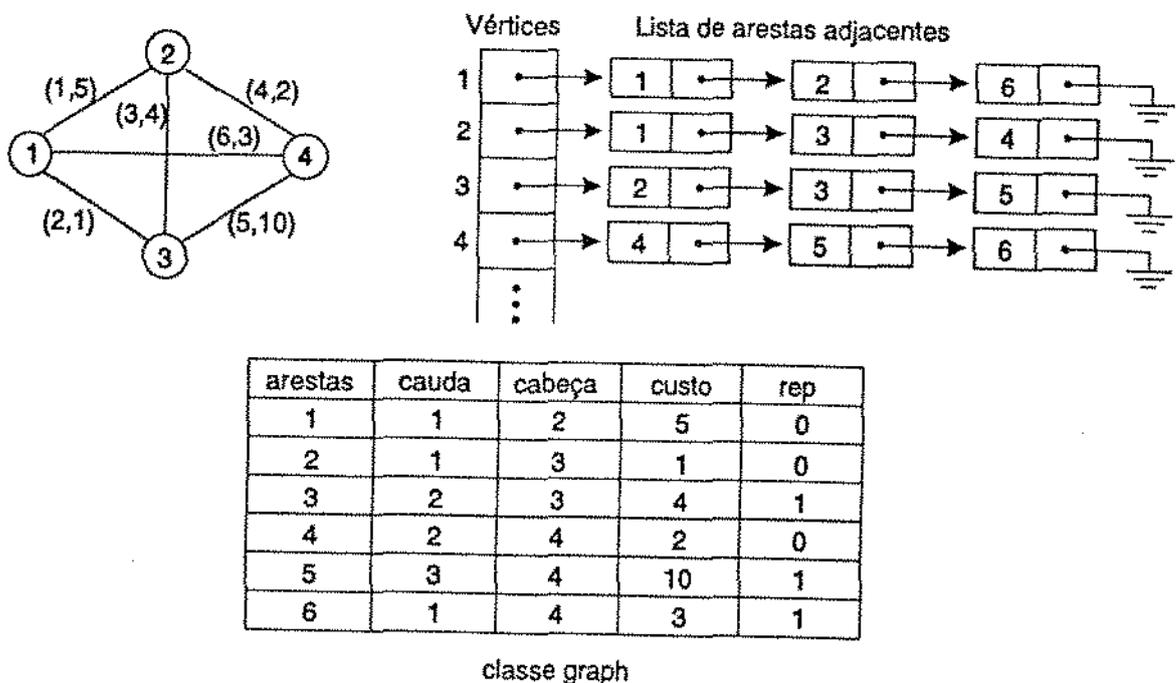


figura III.2

As rotinas que o programa principal chama são: *readgraph*, *odd_set*, *dijkstra* e *matching*. A rotina *readgraph* é chamada para fazer a leitura da rede, inicializando as classes *graph* e *vertex*. A rotina *odd_set* é chamada para determinar o conjunto dos vértices ímpares da rede, contabilizando para isto, apenas as arestas obrigatórias. A rotina *dijkstra* é chamada para determinar os caminhos mínimos entre todos os vértices ímpares da rede. Ela utiliza uma classe chamada *heap*. Trata-se de uma lista de prioridades cuja função é manter a lista de vértices a serem sondados. Esta classe contribui para diminuir a complexidade envolvida na determinação dos caminhos mínimos. Os caminhos mínimos determinados pela rotina *dijkstra* são armazenados em uma tabela.

Uma vez calculados os caminhos mínimos, o programa principal gera uma nova rede contendo, agora, somente os vértices ímpares. Como esta rede é completa, foi utilizada uma classe especial, chamada *matrix*, para armazená-la. Uma vez feita a inicialização dessa classe, a rotina *matching* calcula o emparelhamento perfeito ótimo dos vértice ímpares. Obtido este emparelhamento, calcula-se a rota de Euler na rede original, tendo em vista as arestas que devem ser repetidas. Estas correspondem às

arestas contidas nos caminhos mínimos ligando vértices ímpares emparelhados. O procedimento para a determinação da rota de Euler utiliza uma classe chamada *stack*. Trata-se de uma pilha que serve para armazenar a rota final obtida.

III.2 EMPARELHAMENTO PERFEITO

O emparelhamento perfeito ótimo é determinado com a chamada do procedimento *matching*. Este procedimento manipula uma estrutura de dados formada pelas classes *matrix*, *blossom*, *queue* e pelos vetores *mate*, *antc*, *root* e *flag*. As rotinas que o procedimento *matching* chama são: *exposed*, *pred*, *aum_path*, *shrink* e *unshrink*. Na seção III.2.1 é descrita a estrutura de dados utilizada pelo procedimento *matching* e na seção III.2.2 as rotinas acima listadas.

III.2.1 ESTRUTURA DE DADOS

A classe *matrix*, como já foi colocado, serve para armazenar a rede completa correspondendo aos vértices ímpares da rede original. A classe *blossom* é utilizada para armazenar os vértices da rede corrente e é constituída por um vetor. Se um vértice v está dentro de um anel simples correspondendo ao pseudonó P , então a posição de v neste vetor contém um apontador para a posição do pseudonó P . Se P também estiver contido dentro de um pseudonó P' , então, na posição que lhe cabe no vetor, existe um apontador para a posição de P' . Se um vértice ou pseudonó não está dentro de outro pseudonó, ou seja, se este for um vértice corrente, então o apontador indica sua própria posição.

O número de componentes do vetor de pseudonós que são utilizadas não ultrapassa o valor $3|V_1|/2$, onde V_1 é o conjunto de vértices ímpares da rede original. Isto porque, pelo teorema II.3, podem existir no máximo $|V_1|/2$ pseudonós na rede corrente. A figura III.3 representa um exemplo de como os vértices correntes são armazenados neste vetor.

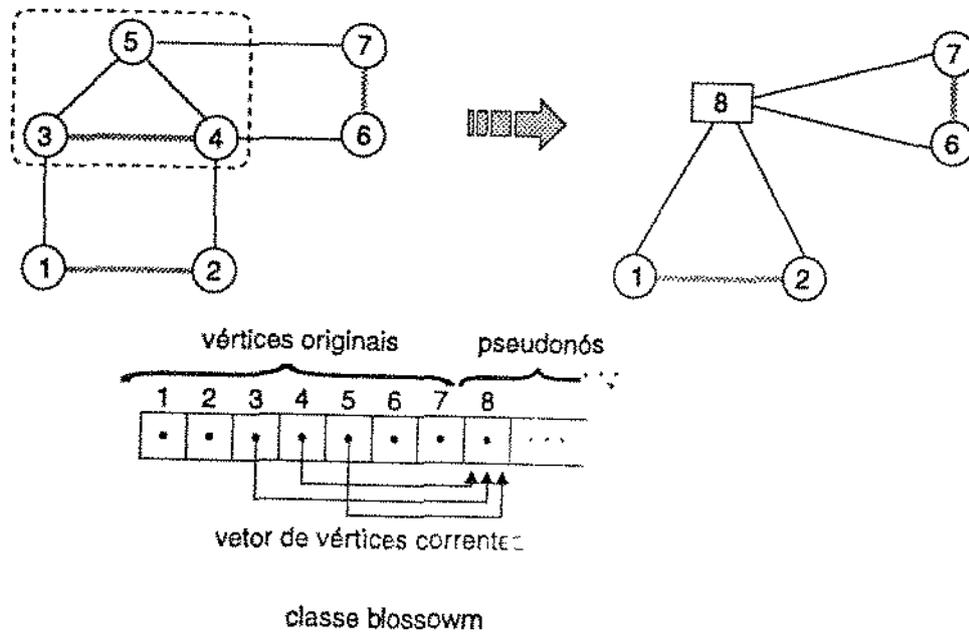


figura III.3

A classe *queue* é uma lista utilizada com duas funções no algoritmo. A primeira, para manter a lista de vértices a serem sondados. Cada vez que um pseudonó é retirado desta lista, coloca-se na mesma os vértices do anel simples a ele associado. A segunda função da classe *queue* é armazenar, logo após a transformação dual, os caminhos aumentantes detectados, os vértices que poderão ser rotulados e os pseudonós ímpares, com custo reduzido igual a zero, a serem expandidos.

Para armazenar as árvores alternadas, assim como os pseudonós, o procedimento *matching* utiliza os vetores *mate*, *antc* e *root*. Todos estes vetores são unidimensionais e no máximo $3|V_1|/2$ de suas componentes são utilizadas. Se $mate[v]=w$, então $mate[w]=v$ e a aresta (v,w) é uma aresta emparelhada. Se $mate[v]=0$, então v é um vértice exposto. O valor de $antc[v]$ é igual ao valor da aresta que liga v a seu antecessor imediato na árvore alternada. O valor de $root[v]$ é igual ao valor do ápice do vértice raiz da árvore a qual v pertence. A figura III.4 é uma representação gráfica de como é feita a armazenagem de uma árvore alternada utilizando os vetores *root*, *antc* e *mate*.

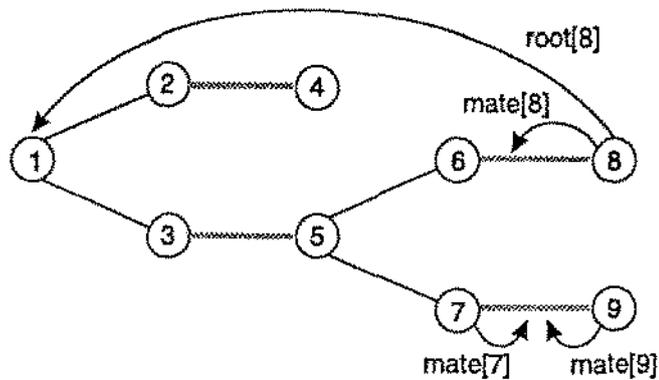
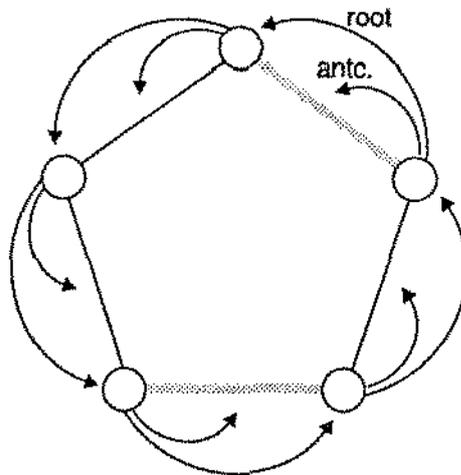


figura III.4

Os vetores *antc* e *root* são também utilizados para armazenar um anel. Neste caso, *root[v]* é o vértice antecessor de *v* no anel simples armazenado e *antc[v]* é a aresta que liga *v* a seu antecessor imediato no mesmo anel. Veja, na figura III.5, como um anel simples é armazenado utilizando os vetores *root* e *antc*.



armazenagem de um anel

figura III.5

O vetor *flag* é utilizado com dois propósitos. Quando dois vértices externos pertencentes à mesma árvore são encontrados, para que se possa fazer a contração do anel correspondente a esta situação, é preciso determinar a sua base. Esta base

corresponde ao vértice comum aos dois caminhos antecessores que ligam os vértices em questão à raiz da árvore a qual pertencem. Estes dois caminhos têm que ser percorridos até a raiz e, neste processo, as posições do vetor *flag*, correspondentes aos vértices destes caminhos, são marcados até que a posição relativa à base seja marcada pela segunda vez.

A outra função do vetor *flag* é determinar o caminho alternado, de comprimento par, que liga um vértice dentro de um anel simples à sua base. Este procedimento é realizado durante uma ampliação, toda a vez que um pseudonó é detectado em um caminho aumentante, e, também, quando é realizada a expansão de um pseudonó.

III.2.2 ROTINAS

A rotina *exposed*, após cada ampliação, identifica os vértices expostos (vértices originais ou pseudonós) e os rotula como sendo raízes de árvores alternadas. Em outras palavras, a cada ampliação, a rotina *exposed* planta uma nova floresta. A rotina *pred* retorna, para um determinado vértice, o seu predecessor relativo a rede corrente. A rotina *aum_path* realiza uma ampliação a partir de um caminho aumentante. No processo de transformação do emparelhamento corrente, a base dos pseudonós que se encontram no caminho aumentante são atualizadas. A rotina *aum_path* chama duas vezes a rotina *route* que, por sua vez, chama a rotina recursiva *set_apex*. A rotina *shrink* executa uma contração toda vez que um anel simples é detectado. Finalmente, a rotina *unshrink* expande um pseudonó ímpar, quando seu custo reduzido se torna zero.

CAPÍTULO IV
ESTUDO DE CASO E CONCLUSÕES

IV.1 UM PROBLEMA DE DISTRIBUIÇÃO DE JORNAIS

Neste capítulo é feita uma análise de um problema de distribuição de jornais. Os dados utilizados foram fornecidos pelo jornal **O Estado de São Paulo** com o objetivo de permitir um estudo do roteamento da entrega de jornais a seus assinantes na cidade de São Paulo. Esta foi mapeada em distritos que cobrem todos os assinantes. O processo de entrega é realizado independentemente em cada distrito. Ou seja, um único veículo é responsável pela entrega aos assinantes de um determinado distrito.

Foi feito um estudo em dois destes distritos. Em ambos, foram considerados os pontos de entrega e a distância aproximada de percurso de cada rua, assim com sua orientação, em caso de ser rua de mão única. É importante salientar que estes dados estão sujeitos a alterações tais como: entrada ou saída de assinantes, mudança na orientação de algumas ruas, ou mesmo interdição parcial de algumas delas por determinados períodos.

Neste capítulo, uma rua é definida como sendo a parte delimitada por duas esquinas, diferente, portanto, do significado real da palavra. Isto permite que trechos de mão única e mão dupla, sejam consideradas no programa. Além disso, estendendo o problema do carteiro chinês, são consideradas ruas obrigatórias e ruas não obrigatórias. Ou seja, ruas pelas quais a rota procurada deve necessariamente passar e ruas pelas quais a rota não precisa passar, mas que podem ser utilizadas. Desta forma, o grau dos vértices é calculado somente em relação às arestas obrigatórias que nele incidem, e os graus interno e externo de um vértice são calculados somente em relação aos arcos obrigatórios.

O distrito 1 (figura IV.1) possui 84 esquinas (vértices) e 131 ruas (67 arestas e 64 arcos) das quais 91 obrigatórias; ou seja, ruas onde se localiza pelo menos um assinante. As ruas obrigatórias somam 16,249 km.

O distrito 2 (figura IV.2) possui 169 esquinas e 283 ruas (100 arestas e 183 arcos) das quais 170 são obrigatórias, totalizando 21,276 km.

Em ambas as figuras as ruas obrigatórias estão marcadas com um "x". Além disto, há a presença de um viaduto em cada um dos distritos, o que dificulta a visualização do problema no plano.

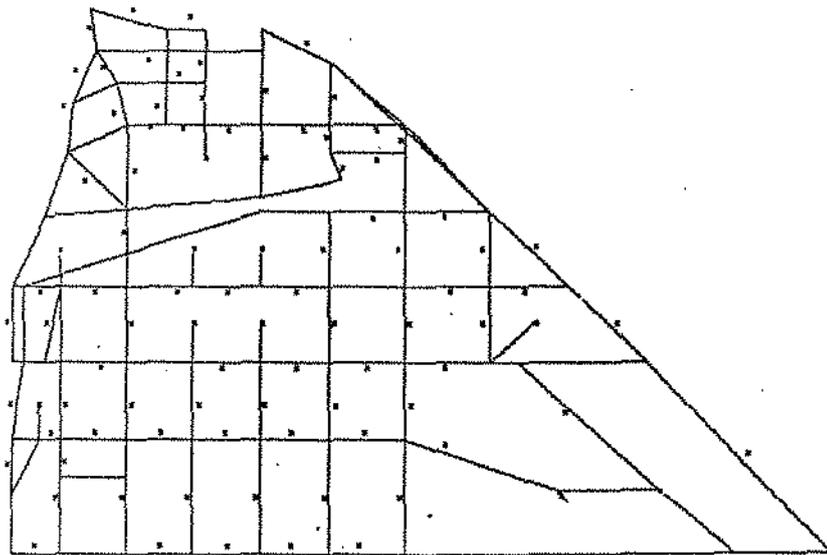


figura IV.1

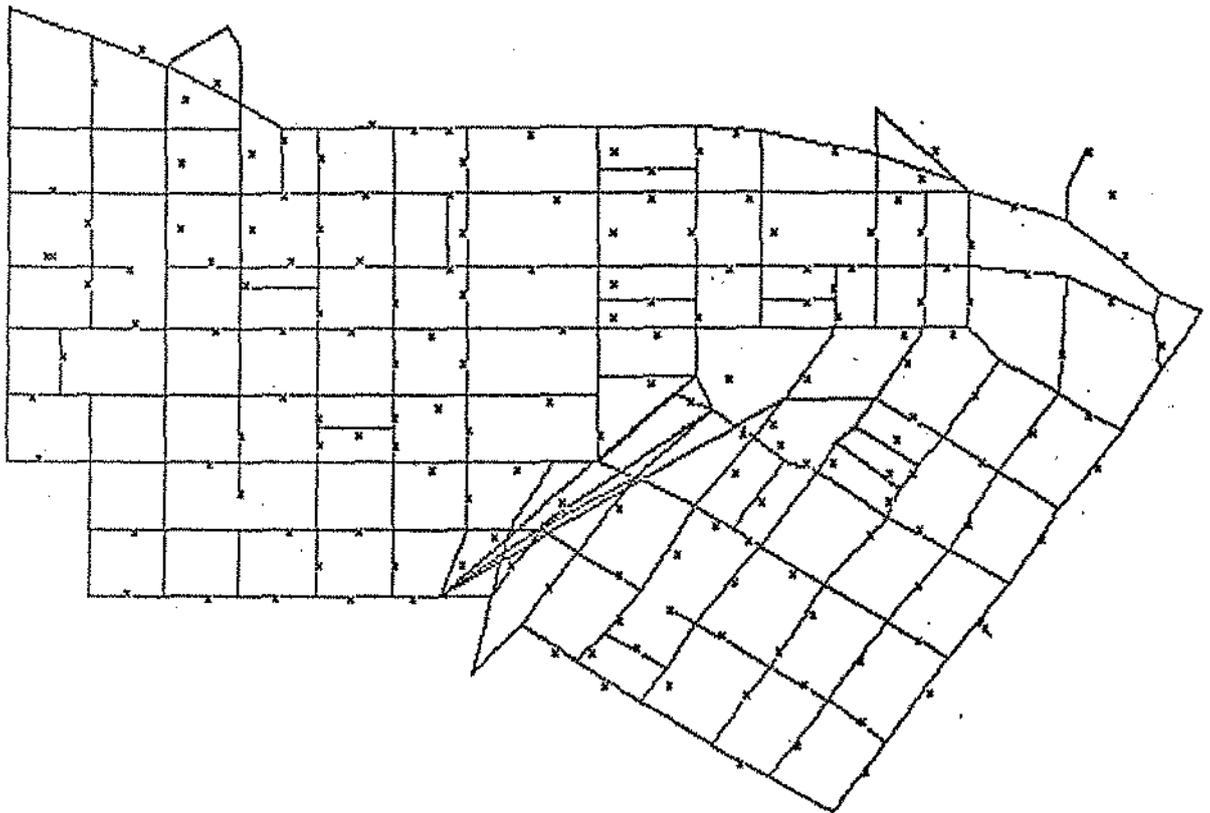


figura IV.2

A determinação das rotas ótimas para os distritos dados constitui um problema NP-completo. Apesar dos distritos 1 e 2 não serem muito grandes, a enumeração completa das soluções seria um problema certamente inviável. Foi utilizado um método heurístico baseado na escolha de orientações para ruas de mão dupla. Este método é iterativo e, a partir da fixação arbitrária de orientação das arestas obrigatórias, é determinado o fluxo ótimo associado à rede transformada. Além desta heurística, foi também implementado um algoritmo baseado na teoria de emparelhamento discutida no capítulo II, que calcula uma rota ótima para a rede, quando as ruas são todas consideradas ruas de mão dupla. O tamanho desta rota representa um limitante inferior para o valor da solução ótima e serve para avaliar o quão próxima a solução heurística encontra-se da solução ótima.

O algoritmo que calcula a rota para o problema relaxado foi implementado em uma estação de trabalho SUN (SPARK-STATION) e o método heurístico, baseado na teoria de fluxo, foi implementado em um microcomputador do tipo IBM-PC.

Foi implementado um terceiro programa, também em microcomputador, que reproduz em tela os mapas dos distritos e a simulação das rotas obtidas pelos dois métodos descritos anteriormente. O algoritmo de fluxo interage com este programa, de forma que é possível tomar as decisões de fixação de orientação das ruas obrigatórias de mão dupla a medida que as soluções parciais vão sendo apresentadas.

IV.2 SOLUÇÃO RELAXADA

Utilizando a implementação computacional, cuja estrutura de dados foi descrita no capítulo III deste trabalho, determinou-se as rotas ótimas dos distritos 1 e 2 considerando todas as ruas como sendo ruas de mão dupla. O tempo total de percurso neste caso, obtido para cada uma das rotas, é importante e serve como limitante inferior para o tempo total de percurso calculado em cada distrito obedecendo à orientação das ruas.

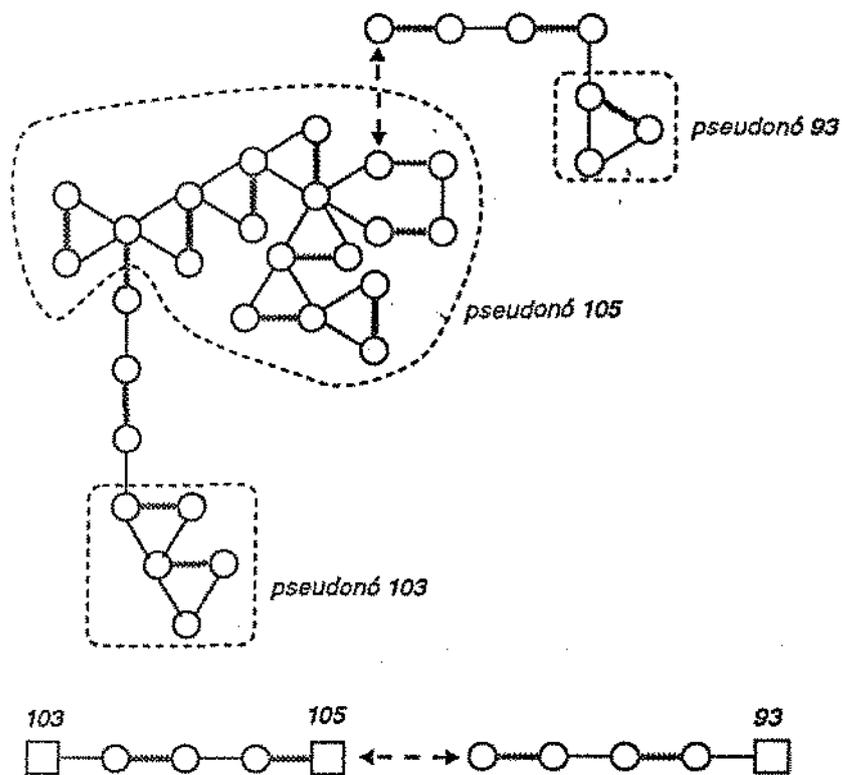
O algoritmo, aplicado ao distrito 1, detectou 44 vértices ímpares na rede original e, portanto, a rotina que resolve (EPO) é aplicada a uma rede com este número de vértices. Foram realizadas 22 ampliações em 20 iterações. Veja que pode ocorrer mais de uma ampliação por iteração, caso a transformação dual detecte mais de um caminho aumentante. No total, foram 35 transformações duais, 33 rotulamento e 4 operações de contração. Foi obtida uma rota passando por 123 arestas totalizando

20,495 km.

No distrito 2 foram detectados 82 vértices ímpares e realizadas 41 ampliações em 30 iterações. Ao todo, foram 81 transformações duais, 139 rotulamentos, 29 operações de contração e 7 operações de expansão. Foi obtida uma rota passando por 241 arestas totalizando 27,132 km.

Tanto para o distrito 1 quanto para o distrito 2, o tempo de processamento, necessário para a obtenção das rotas, não ultrapassou 2 min.

A tendência que se observa é de que, nas primeiras iterações, o número de caminhos aumentantes detectados é grande, sendo que as operações de contração e expansão, assim como rotulamentos, praticamente não ocorrem. Com o aumento do número de iterações, observa-se justamente o contrário. Isto indica que, com a evolução do algoritmo, os caminhos aumentantes detectados vão se tornando cada vez mais compridos e o tamanho das árvores alternadas aumenta. Ou seja, para se detectar um novo caminho aumentante é necessário crescer florestas com árvores cada vez maiores. A figura IV.3 corresponde a um caminho aumentante, encontrado na trigésima iteração, entre os pseudonós expostos 103 e 93 (vértices com valor acima de 82 são pseudonós). O comprimento deste caminho, em relação à rede corrente, é igual a nove e, em relação à rede original (a rede contendo 82 vértices), é igual a 25.



Caminho Aumentante ligando os Pseudonós 103 e 93

figura IV.3

O pseudonó 105, existente neste caminho, é um pseudonó de nível 5. Veja, na figura IV.4, a relação de heranças dos pseudonós relativas ao nincho de pseudonós para o qual o vértice 105 é o pseudonó corrente. Como se pode ver na figura, o pseudonó 105 foi obtido a partir da contração do anel simples contendo os vértices 102 (pseudonó), 100 (pseudonó) e 79. Dos pseudonós dormentes contidos no pseudonó 105, apenas um é o resultado da contração de um anel contendo mais de três vértices. O anel 94 é o resultado da contração dos vértices 90 (pseudonó), 24, 23, 74, e 73.

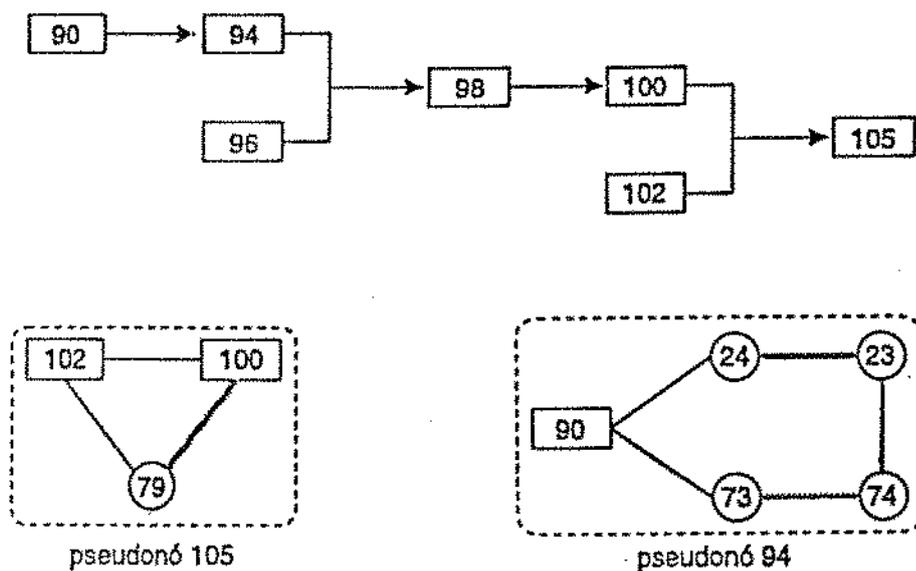


figura IV.4

IV.3 SOLUÇÃO HEURÍSTICA

O método heurístico utilizado consiste em fixar orientações a serem adotadas para as ruas obrigatórias de mão dupla, transformando o problema de roteamento em um problema de fluxo ótimo como descrito em I.4.2. Ou seja, cada aresta não obrigatória é transformada em dois arcos simétricos não obrigatórios e cada aresta obrigatória é transformada em dois arcos simétricos, sendo um obrigatório e o outro não obrigatório. A solução ótima é, então, determinada considerando a demanda em cada vértice (grau interno menos grau externo) computada somente em relação aos arcos obrigatórios do novo problema. A solução obtida neste problema pode ser apresentada sob a forma de um terço de números inteiros $(\alpha, \beta, \gamma)_a$ para cada arco ou aresta original a

- * para arcos obrigatórios: $(\alpha = 1, \beta, \gamma = 0)$
- * para arcos não obrigatórios: $(\alpha = 0, \beta, \gamma = 0)$
- * para arestas obrigatórias: $(\alpha = 1, \beta, \gamma)$ com $\beta\gamma = 0$
- * para arestas não obrigatórias: $(\alpha = 0, \beta, \gamma)$
com $\beta = \gamma = 1$ ou $\beta\gamma = 0$

Neste caso, o índice $\alpha = 0$ indica rua não obrigatória e $\alpha = 1$ indica arco obrigatório ou aresta obrigatória (com orientação fixada); o índice β representa o número de vezes que o arco ou aresta (com orientação fixada) é utilizado; o índice γ representa o número de vezes que a aresta é utilizada com orientação contrária à fixada. A rede obtida com estes índices é uma rede simétrica.

Em uma primeira etapa, a decisão heurística consiste em trocar a orientação fixada para as arestas obrigatórias com $\gamma \geq 2$ e resolver um novo problema de fluxo ótimo. Esta etapa reduz o tamanho da rota.

Em uma segunda etapa, devem ser efetuadas trocas tentativas em arestas obrigatórias com ($\alpha = 1, \beta = 0, \gamma = 1$), procurando reduzir o tamanho total da rota. Veja que esse tipo de alteração permite a diminuição do tamanho da rota ou, na pior das hipóteses, o tamanho desta permanece o mesmo.

A primeira etapa é executada uma única vez e produz uma redução no tamanho da rota. Por outro lado, a segunda etapa é realizada testando diversas combinações de orientação das arestas obrigatórias para, possivelmente, obter novas reduções no tamanho da rota. Observe que a solução ótima do problema pode ser obtida caso sejam testadas todas as combinações de orientação das arestas; isto corresponde a uma enumeração de todas as soluções factíveis para o problema. É importante salientar que esta etapa foi realizada manualmente, analisando a rota obtida em cada tentativa e sua complexidade corresponde a um problema de fluxo ótimo, ou seja $O(n^2 \times m \times \log(n^2/m))$ para cada tentativa.

Aplicando a heurística obteve-se uma rota contendo 158 ruas totalizando 25,130 km para o distrito 1 e uma rota contendo 300 ruas totalizando 35,130 km para o distrito 2. A solução para o problema de fluxo, resolvido cada vez que são fixadas novas orientações para as arestas, demanda um tempo de processamento da ordem de um minuto em um PC 386.

IV.4 AVALIAÇÃO DA ROTA

Foi desenvolvido um programa em Pascal com recursos gráficos para produzir a visualização dos mapas e as rotas dos distritos. Para cada distrito o programa apresenta:

- 1) tela com o mapa completo, inclusive com os assinantes (figuras IV.1 e

IV.2).

2) telas com o mapa contendo somente ruas obrigatórias e a marcação dos vértices ímpares e não simétricos.

3) telas com a construção das rotas sugeridas: solução relaxada e solução heurística.

O uso deste programa permite visualizar a execução de algumas conversões proibidas e outros defeitos que impedem a viabilidade da rota sugerida e que são eliminados com a alteração nos dados do problema.

IV.5 CONCLUSÕES

As rotas obtidas, aplicando o algoritmo de emparelhamento, totalizam 20 km e 27 km para os distritos 1 e 2, respectivamente. Aplicando a heurística foram obtidas rotas com os tamanhos 25 km e 35 km. Portanto, a solução relaxada representou uma economia de 23% e 25%. As rotas obtidas para os problemas 1 e 2 relaxados passam por 123 e 237 ruas, respectivamente. Por outro lado, as rotas obtidas pelo método heurístico passam por 158 e 300 ruas. É conveniente lembrar que nas redes originais há 91 ruas obrigatórias totalizando 16 km e 175 ruas obrigatórias totalizando 21 km para os distritos 1 e 2, respectivamente.

É bom lembrar que a entrega dos jornais é efetuada por volta das três horas da madrugada, quando praticamente não há tráfego, e que, por este motivo, os funcionários encarregados da entrega eventualmente não obedecem à orientação das ruas. Neste sentido a solução relaxada pode vir a ter uma importância maior do que simplesmente um limitante inferior para o valor da solução obtida com o método heurístico.

Ajustes na rede original

Eventualmente é necessário realizar alguns ajustes nos dados originais do problema para obter uma rota factível. A seguir são discutidos alguns casos.

A solução obtida pode ser formada por subrotas não conectadas; observe que, neste caso, a rede obrigatória (rede gerada pelas ruas obrigatórias), assim como a rede Euleriana obtida, são não conectadas. O ajuste adotado para conectá-las foi o de tornar obrigatórias algumas ruas inicialmente não obrigatórias. Estas ruas foram

escolhidas a partir da visualização das subrotas na tela do microcomputador.

Houve a necessidade de criação de esquinas e ruas artificiais para evitar conversões proibidas.

A sequência de percurso de algumas ruas foi alterada para evitar que o motorista encarregado da entrega percorresse duas vezes seguidas (em direções contrárias) a mesma rua. Neste caso ele teria que realizar uma manobra de 180 graus no final da rua, sendo que é preferível que tal rua seja percorrida em subrotas distintas.

Orientação das arestas obrigatórias

Foram consideradas apenas algumas combinações de arestas obrigatórias na fixação de suas orientações. Numa primeira etapa foram invertidas as direções fixadas para as arestas obrigatórias utilizadas mais de uma vez em sentido contrário ao fixado. Posteriormente, em uma segunda etapa, foram testadas algumas combinações "atraentes".

O uso da solução relaxada serve como parâmetro para o esforço (número de tentativas) empregado na heurística. Acredita-se que as soluções heurísticas estejam bem próximas do ótimo pelo número de combinações testadas. As diferenças entre as soluções relaxadas e heurísticas parecem refletir gargalos no tráfego que, na solução heurística, obrigam a repetir várias vezes subrotas necessárias para se alcançar determinadas ruas.

Tempo x distância

O tamanho das ruas, fornecido pelo jornal, foi dado em distância (medida em metros). É comum encontrar este tipo de problema com dados associados ao tempo de percurso de cada rua. Observe que em uma situação dessas o tempo de percurso de uma rua obrigatória envolve o tempo de parada para entrega em apenas uma das vezes que esta é percorrida. Ou seja, para cada rua obrigatória devem ser fornecidos tempos de percurso com entrega e sem entrega.

Os métodos discutidos também se aplicam ao problema com essas características, uma vez que o tempo total de parada para entrega independe da rota escolhida. Neste sentido, não é preciso computar o tempo de parada para realizar a otimização das

rotas, mas o tempo total de parada deve ser adicionado ao tempo total de percurso na rota final obtida.

IV.6 COMENTÁRIOS FINAIS

O Método proposto mostrou-se exequível, tendo em vista as exigências do pessoal do jornal **O Estado de São Paulo** que testou a viabilidade das rotas sugeridas. Na realidade, a interdição parcial e temporária de diversas ruas foi constatada apenas nestes testes, quando as rotas sugeridas foram postas em prática.

O ambiente obtido com o desenvolvimento de um programa em Pascal que mostra os mapas, as soluções obtidas, e realiza a simulação das rotas sobre o mapa produzido em tela, possibilita ao usuário obter rapidamente soluções satisfatórias.

Para finalizar, acreditamos ter alcançado o objetivo desta dissertação que, em última instância, constitui um estudo do problema do carteiro chinês. Foi considerado em maiores detalhes o caso de redes não orientadas. Neste sentido o capítulo II foi dedicado inteiramente ao problema de emparelhamento perfeito ótimo e o capítulo III a uma descrição da implementação computacional daquele caso, baseada no algoritmo proposto por Edmonds [2]. Além dos aspectos teórico (cap. I e II) e computacional (cap. III) abordados, no capítulo IV, procurou-se fazer uma descrição científica do desenvolvimento de um trabalho prestado ao jornal **O Estado de São Paulo**.

REFERÊNCIAS BIBLIOGRÁFICAS

1. Kwan Mei-Ko, "Graphic Programming using Odd and Even Points", Chinese Math., Vol. 1 (1962), pp. 237-277.
2. Jack Edmonds & Ellis L. Johnson, "Matching, Euler Tours and the Chinese Postman", Mathematical Programming, Vol 5 (1973) pp. 88-124.
3. Jack Edmonds, "The Chinese Postman Problema", Operational Research 13 Suppl. Vol 1 (1965), pp. 373.
4. Jack Edmonds, "Paths, Trees and Flowers", Canadian Journal of Mathematics, Vol 17 (1965), pp. 449-467.
5. E. Minieka, "The Chinese Postman Problem for Mixed Networks", Management Science Vol. 7 (1979), pp. 643-648.
6. M. Guan, "On the Windy Postman Problem", Discrete Applied Mathematics Vol. 9 (1984), pp. 41-46.
7. E. Minieka, "Optimization Algorithms for Networks and Graphs", (1978) Marcel Dekker, NY.
8. E. L. Lawler, "Combinatorial Optimization: Networks and Matroids", (1976) Holt, Rinerhart and Winston, NY.
9. G. L. Nemhauser & L. A. Wolsey, "Integer and Combinatorial Optimization", (1988) John Wiley & Sons.
10. K. G. Murty, "Network Programming", (1992) Prentice Hall, Englewood Cliffs, NJ.
11. Mark Allen Weiss, "Data Structures and Algorithm Analysis", (1992) The Benjamin/Cumming Publishing Company.

12. Bjarne Stroustrup, "The C++ Programming Language", (1991) AT&T Bell Laboratories, Murray Hill, NJ.