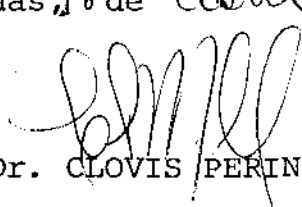


ESTUDO DOS PROBLEMAS DO CARTEIRO CHINÊS E DO
CAIXEIRO VIAJANTE

Este exemplar corresponde à redação final da tese devidamente corrigida e defendida por DENISE HELENA LOMBARDO e aprovada pela comissão julgadora.

Campinas, 18 de abril de 1986


Prof. Dr. CLOVIS PERIN FILHO

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

Aos meus pais e
minhas irmãs

AGRADECIMENTOS

Ao Professor Clóvis Perin Filho pela orientação constante.

Ao CNPq e FAPESP pelo apoio financeiro.

Aos colegas, professores e funcionários do Instituto de Matemática, Estatística e Ciência da Computação, pela colaboração.

Aos amigos pelo incentivo.

NOTAÇÃO

G um grafo ou uma rede

N conjunto de nós em um grafo (ou uma rede)

A conjunto de linhas (arcos ou arestas) em um grafo (ou rede)

(i,j) um arco em um grafo (ou rede)

$(i;j)$ uma aresta em um grafo (ou rede)

$\mathbb{R}$ conjunto dos números reais

$\mathbb{R}^n$ espaço vetorial real n -dimensional

$\|x\|$ norma euclideana de um vetor $x \in \mathbb{R}^n$

$$\text{Se } x = (x_1, \dots, x_n), \quad \|x\| = \sqrt{x_1^2 + \dots + x_n^2}$$

$|r|$ valor absoluto de um número real r

$|S|$ cardinalidade do conjunto S

$\lfloor r \rfloor$ maior inteiro menor ou igual ao número real r

$\lceil r \rceil$ menor inteiro maior ou igual ao número real r

Σ símbolo de somatória

∞ símbolo de infinito

$\in$ símbolo de pertinência

$\subset$ símbolo de subconjunto próprio

$\cup$ símbolo de união de conjuntos

$\cap$ símbolo de interseção de conjuntos

$\emptyset$ conjunto vazio

$\setminus$ símbolo de diferença de conjuntos

$<, \leq$ símbolo de relações de ordem

$\forall$ símbolo para todo

$[0,1]$ intervalo fechado $0,1$

$\equiv$ símbolo de congruência

$\implies$ símbolo de implicação

$\longrightarrow$ símbolo de tendência

$\longleftarrow$ símbolo de atribuição

$O(n^r)$ esforço computacional de um algoritmo com convergência finita de ordem n^r .

SUMÁRIO

PARTE I: O PROBLEMA DO CARTEIRO CHINÊS

CAPÍTULO 1: Introdução

CAPÍTULO 2: Redes Não-Orientadas

1. Algoritmo para rota do carteiro
2. Algoritmo para rota euleriana

CAPÍTULO 3: Redes Orientadas

CAPÍTULO 4: Redes Mistas

PARTE II: O PROBLEMA DO CAIXEIRO VIAJANTE

CAPÍTULO 1: Introdução

CAPÍTULO 2: Características e Propriedades

1. Conceitos Básicos
2. Rotas Hamiltonianas
3. Adjacência no Politopo do Problema da Designação
4. Politopo

CAPÍTULO 3: Métodos Exatos

1. Planos de Cortes
2. Programação Dinâmica
3. Método do Subgradiente
4. Particionar e Limitar
5. Cadeia Máxima

CAPÍTULO 4: Métodos Não Exatos (Heurísticos)

1. Cidade não visitada mais próxima
2. Inserção da cidade mais próxima
3. Método de Lin
4. Método de Christofides

CAPÍTULO 5: Aplicações

1. Sequenciamento de tarefas usando 1 máquina
2. Sequenciamento de tarefas usando m máquinas
3. m-caixeiros
4. Programação de Vôos
5. Planejamento de conexões elétricas em computadores
6. Agrupamento de Dados

CAPÍTULO 6: Casos Particulares

1. Sequenciamento de tarefas em fornos
2. Matriz de distância triangular superior
3. Redução de grandes Problemas
4. Coleta de itens em um almoxarifado retangular

CAPÍTULO 7: Problema da Cobertura Simétrica

CAPÍTULO 8: O Problema do Caixeiro Viajante tipo "Gargalo"

(Bottleneck).

APÊNDICE I: Definições

APÊNDICE II: Resultados Computacionais

PREFÁCIO

Esta dissertação envolve o estudo de dois problemas de otimização combinatória: O Problema do Caixeiro Viajante (PCV) e o Problema do Carteiro Chinês (PCC). Dada uma rede (ou grafo), o primeiro problema consiste em determinar uma rota circular mínima que passa em cada nó e o segundo em determinar uma rota circular mínima que passa em cada linha da rede. Embora ambos os problemas sejam da classe NP-"árduo" (NP-hard), o problema do Carteiro Chinês é apresentado na literatura como um problema menos "difícil" de ser resolvido.

O interesse em estudar o PCV e o PCC partiu do grande número de publicações em revistas e livros técnicos de Pesquisa Operacional a respeito destes problemas. Além disto, estes problemas são de importância no estudo da determinação de rotas de veículos onde se procura obter rotas que devem ser utilizadas por uma frota de veículos para satisfazer determinadas demandas (ou restrições) tanto nos nós quanto nas linhas; por exemplo, coleta de lixo de n cidades.

PARTE I : O PROBLEMA DO CARTEIRO CHINÊS

CAPÍTULO 1

INTRODUÇÃO

Considere uma região cortada por ruas com comprimentos conhecidos. O Problema do Carteiro Chinês (PCC) consiste em determinar uma rota de comprimento total mínimo que passa por todas as ruas e retorna ao ponto de origem. A denominação carteiro chinês deve-se ao primeiro trabalho a esse respeito publicado em um periódico chinês por Mei-Ko [Mk 1].

Considere um grafo $G = (N, A)$, onde N é um conjunto finito não vazio de n nós (ou esquinas) e A é um conjunto finito não vazio de m linhas (ou trechos de ruas entre duas esquinas). Cada linha de A pode ser uma aresta (isto é, par não-ordenado $(i; j)$ de nós distintos) ou pode ser um arco (isto é, par ordenado (i, j) de nós distintos). Se todas as linhas de G são arcos (arestas) então G é um grafo orientado (não-orientado); caso contrário G é um grafo misto.

Seja $C = (c_{ij})$ uma matriz de distância em G , isto é, c_{ij} é o comprimento (custo ou tempo) do arco (i, j) ou da aresta $(i; j)$ para cada linha de G ; então $G = (N, A, C)$ é uma rede se (N, A) é um grafo e $C = (c_{ij})$ é uma matriz de distância em G . Uma rede é orientada, não-orientada, ou mista de acordo com o grafo associado.

Uma *rota do carteiro* é uma sequência de nós e linhas $(i_1, l_1, \dots, l_{q-1}, i_q)$ onde $l_j = (i_j, i_{j+1})$ ou $l_j = (i_j; i_{j+1})$, $i_1 = i_q$, $\{l_1, l_2, \dots, l_{q-1}\} = A$ e $q-1 \geq m$. O PCC consiste em determinar uma rota do carteiro de comprimento mínimo. Se $C = (c_{ij}) \geq 0$ e a rede G possui uma *rota euleriana* (isto é, rota do carteiro com $q = m$), então esta é a solução ótima do PCC. Esta denominação é devido a Leonardo Euler que estudou o conhecido problema da ponte de Königsberg [Mu 1].

Para redes orientadas e não-orientadas o PCC pode ser resolvido em tempo polinomial [EJ 2], enquanto que para redes mistas é NP-árduo (NP-"hard") [GJ 1], isto é, não é conhecido nenhum algoritmo polinomial que resolva este problema. Os algoritmos desenvolvidos por Edmonds e Johnson são apresentados de modo mais simples por Minieka [Mi 1].

Guan [Gu1] trata do problema do carteiro com vento ("Windy postman"), onde $G = (N, A, C)$ é uma rede orientada com matriz de incidência simétrica (isto é, existe o arco (i, j) se e somente se existe o arco (j, i)) e com matriz de distância não necessariamente simétrica (isto é, o comprimento do arco (i, j) pode ser diferente do comprimento do arco (j, i)). Tal tratamento permite considerar ruas com subidas e descidas. Guan prova que este problema é NP-árduo (NP-"hard").

DEFINIÇÕES

Dada uma rede $G = (N, A, C)$, considere uma sequência finita $P = (i_1, \ell_1, i_2, \ell_2, \dots, i_{q-1}, \ell_{q-1}, i_q)$, onde $i_1, \dots, i_q$ são nós de G e cada ℓ_j é uma linha de G que une os nós i_j, i_{j+1} para $j = 1, \dots, q-1$.

P é um *caminho* conectando i_1, i_q se $i_1, \dots, i_q$ são nós distintos.

P é uma *cadeia* conectando i_1, i_q se $i_1, \dots, i_q$ são nós distintos e cada ℓ_j é um arco de i_j para i_{j+1} .

P é um *ciclo* se P é um caminho com $i_1 = i_q$.

P é um *circuito* se P é uma cadeia com $i_1 = i_q$.

G é uma *rede completa* se cada par de nós é unido por uma linha (aresta ou arco).

G é uma *rede conectada* se cada par de nós é conectado por um caminho.

G é uma *rede fortemente conectada* se para cada par de nós i, j existe uma cadeia de i a j e uma cadeia de j a i .

Um nó i de G é *isolado* se não existe nenhuma linha (aresta ou arco) incidente com ele.

$gr(i)$ denota o *grau do nó i* de G , isto é, o número de linhas de G incidentes com i .

$gr^+(i)$ denota o *grau externo do nó i* de G , isto é, o número de arcos de G incidentes de i .

$gr^-(i)$ denota o *grau interno do nó i* de G , isto é, o número

ro de arcos de G incidentes em i .

G é uma rede par se $gr(i)$ é par para todo i de G .

G é uma rede simétrica se $gr^+(i) = gr^-(i)$ para todo i de G .

CAPÍTULO 2

REDES NÃO-ORIENTADAS

TEOREMAS

Uma condição necessária e suficiente para a existência de uma rota do carteiro é que a rede G seja conectada, a menos de nós isolados [Be 1].

Uma condição necessária e suficiente para a existência de uma rota euleriana é que a rede conectada G seja par [Be 1].

Considere $G = (N, A, C)$ uma rede conectada, com $C = (c_{ij}) \geq 0$.

Se G não é par, então algumas arestas devem ser usadas mais do que uma vez na rota do carteiro. Observando que o número de visitas feitas pelo carteiro em um determinado nó é igual ao número de arestas usadas para chegar ao nó e que é igual também ao número de arestas usadas para sair do nó, resulta que:

- i) um *nó par* (isto é, nó com grau par) necessita que o número de arestas reutilizadas incidentes com ele seja par.
- ii) um *nó ímpar* (isto é, nó com grau ímpar) necessita que o número de arestas reutilizadas incidentes com ele seja ímpar.

Um método [EJ 2] para resolver o PCC consiste de três fases. Na primeira, determina-se o menor caminho conectando cada par de nós ímpares em G . Seja $G' = (N', A', C')$ uma rede comple

ta onde N' é o conjunto de nós ímpares de G e C' é a matriz das menores distâncias entre cada par de nós de N' . Na segunda fase determina-se o 1-emparelhamento perfeito (subconjunto de arestas tal que cada nó é incidente com exatamente uma aresta deste subconjunto) ótimo em G' [Ed2] ; as arestas deste 1-emparelhamento (caminhos em G) são acrescentadas em G transformando-a em uma rede par para que na terceira fase seja determinada a rota euleriana.

Edmonds e Johnson [EJ 2] desenvolveram um algoritmo que reúne as duas primeiras fases (determinação dos caminhos mínimos entre os nós ímpares e a obtenção do 1-emparelhamento perfeito ótimo). Este algoritmo e a formulação do PCC são apresentados na seção 1 e a rota euleriana é determinada na seção 2.

1. ALGORITMO PARA ROTA DO CARTEIRO

Seja $(1 + x_{ij})$ o número de vezes que a aresta $(i;j)$ é usada na rota do carteiro. Deseja-se determinar $X = (x_{ij})$ a solução ótima do problema

$$\text{Min} \quad \sum_{i,j} c_{ij} x_{ij}$$

$$\text{sujeito a} \quad \sum_j (1 + x_{ij}) \equiv 0 \pmod{2} \quad i \in N$$

$$x_{ij} = 0, 1, 2, \dots \quad (i;j) \in A$$

onde x_{ij} e x_{ji} denotam a mesma variável.

A solução do problema acima determina as arestas que são utilizadas mais do que uma vez. É possível mostrar que para $C = (c_{ij}) \geq 0$ a solução ótima é um vetor 0-1 [EJ 2].

Toda solução do problema acima é uma solução viável do programa linear abaixo, onde S é a família de todos os conjuntos ímpares de nós, isto é, conjuntos com um número ímpar de nós ímpares e um número qualquer de nós pares.

$$\text{Min} \quad \sum_{i,j} c_{ij} x_{ij}$$

$$\text{sujeito a} \quad \sum [x_{ij} : (i;j) \in \delta(S)] \geq 1 \quad S \in \mathcal{S}$$

$$x_{ij} \geq 0 \quad (i;j) \in A$$

onde $\delta(S) = \{(i;j) : i \in S, j \notin S \text{ ou } i \notin S, j \in S\}$

O problema dual associado e as condições de folga complementar são

$$\text{Max} \quad \sum_S \mu_S$$

$$\text{sujeito a} \quad \sum_S [(\mu_S : (i;j) \in \delta(S))] \leq c_{ij} \quad (i;j) \in A$$

$$\mu_S \geq 0 \quad S \in \mathcal{S}$$

onde μ_S é o preço associado ao conjunto ímpar S .

$$(\sum [x_{ij} : (i;j) \in \delta(S)] - 1) \mu_S = 0 \quad S \in \mathcal{S}$$

$$(c_{ij} - \sum_S [\mu_S : (i;j) \in \delta(S)]) x_{ij} = 0 \quad (i;j) \in A$$

CONCEITOS

Dada uma solução $\mu = (\mu_S)$ defina o *custo relativo da aresta* $(i;j)$, $d_{ij} = c_{ij} - \sum_S [\mu_S : (i;j) \in \delta(S)]$, o conjunto de *arestas-de-igualdade* $A^- = \{(i;j) : d_{ij} = 0\}$. Dada uma solução $X = (x_{ij})$ defina o conjunto de *arestas soluções* $A^* = \{(i;j) : x_{ij} = 1\}$.

Inicialmente $A^* = \emptyset$, $\mu_S = 0$. Durante a execução do algoritmo μ_S sempre é uma solução dual viável e (μ, X) satisfaz as condições de folga complementar. Ao final $X = (x_{ij})$ é uma solução primal viável, e portanto ótima.

Neste algoritmo, *árvores alternantes* são plantadas com raízes em nós expostos (isto é, nós ímpares não incidentes com arestas de A^*) e crescidas com a incorporação de novas arestas-de-igualdade. Este crescimento é feito de tal forma que o caminho predecessor de cada nó à raiz de sua árvore é um caminho alternante. Um *caminho alternante* é um caminho constituído por arestas-de-igualdade solução/não solução que se alternam ao longo deste caminho. A aresta de um caminho alternante inci-

dente com a raiz é sempre do tipo não solução.

Árvores alternantes são utilizadas com o objetivo de detectar caminhos aumentantes (isto é, caminhos alternantes que conectam dois nós expostos), para realizar ampliações. Uma *ampliação* consiste em trocar o papel solução/não solução das arestas no caminho aumentante e com isto diminui o número de nós expostos. A cada ampliação dois nós expostos tornam-se não expostos.

As árvores alternantes são mantidas associando a cada nó um rótulo com duas componentes:

- i) de tipo que pode ser *interno* se o número de arestas no caminho até a raiz é ímpar ou *externo*, caso contrário; e
- ii) de predecessor - próximo nó no caminho até a raiz. Onde i é um nó de uma árvore alternante, seja P_i o caminho predecessor deste nó obtido a partir dos rótulos predecessores. Os rótulos dos nós destes caminhos alternam-se em rótulos externos/internos.

Caso seja detetado um botão (subrede caracterizada abaixo), este é comprimido em um *pseudonó* criando uma nova rede corrente. Este procedimento pode ser repetido, e é possível que um pseudonó venha a conter outros pseudonós em seu interior.

As seguintes denominações são utilizadas: *nó original* - nó da rede G ; *nó corrente* - nó ou pseudonó que não se encontra no interior de um pseudonó; *nó ímpar* - nó original ímpar ou pseu-

donô (botão comprimido associado a conjunto ímpar); *nô exposto* - *nô ímpar* não incidente com nenhuma aresta solução; *nô emparelhado* - *nô original* ou *pseudonô* incidente com aresta solução ($x_{ij}=1$).

Subconjuntos de nós também recebem denominações especiais:

dupla emparelhada - dois nós ímpares unidos por uma aresta solução que é a única aresta solução incidente com cada um destes dois nós.

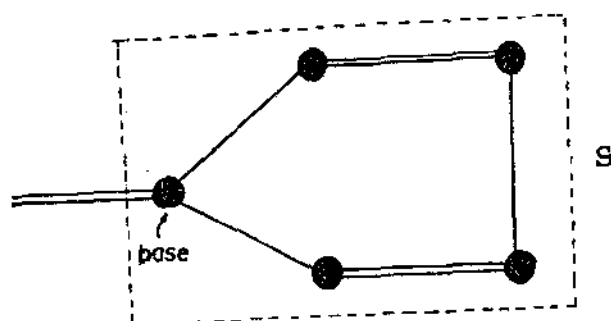
botão com ciclo ímpar - conjunto ímpar de nós formando um ciclo alternante ímpar com um *nô base* que é o único *nô* incidente com duas arestas não soluções deste ciclo.

botão sem ciclo ímpar - conjunto ímpar de nós formado por um *nô base* (que é ou um *nô original ímpar* com preço nulo ou um *nô original par*), emparelhado com um ou mais nós ímpares. É possível que haja no botão um *nô ímpar exposto* ou formando uma *dupla emparelhada* com um *nô* fora do botão, unido ao *nô base* por uma aresta-de-igualdade não solução.

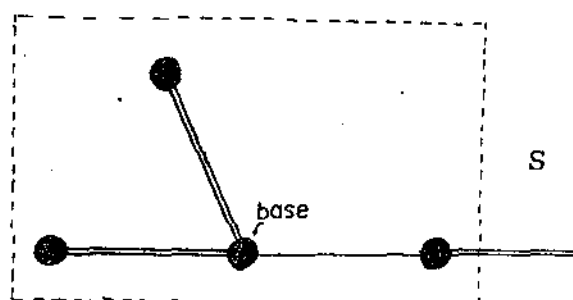
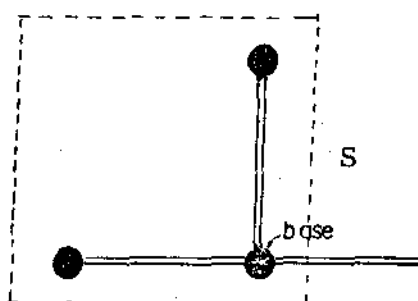
Apenas nós correntes expostos e nós correntes de dupla emparelhadas podem receber rótulos. As seguintes operações são realizadas durante a execução do algoritmo.



dupla emparelhada



botão com ciclo ímpar



botão sem ciclo ímpar (base é não original)

FIGURA 1. CLASSIFICAÇÃO DE SUBCONJUNTOS DE NÓS
 — aresta-de-igualdade não solução
 == aresta-de-igualdade solução

Plantação da Árvore - executada na inicialização e após cada am
pliação. Consiste em atribuir rótulo externo sem predecessor pa
ra cada nó corrente exposto.

Crescimento da Árvore (ou rotulação dos nós) - executada quando
uma aresta-de-igualdade não solução une um nó externo i a um
nó ímpar j (não original ou com preço não nulo) que forma uma
dupla emparelhada j,k . Atribua rótulos interno, externo a j,k ,
respectivamente.

Compressão tipo 1 (com ciclo ímpar) - executada quando uma are
sta-de-igualdade não solução une dois nós externos da mesma árvo
re alternante. $(P_i \cup P_j) \cup \{(i,j)\} \setminus (P_i \cap P_j)$ forma um ciclo al
ternante ímpar

Compressão tipo 2 - executada quando uma aresta-de-igualdade não
solução une um nó externo i a um nó original k que é ímpar
com preço nulo ou é par. O botão é constituído dos nós i,k e de
todos os nós emparelhados a k .

Compressão tipo 3 - executada quando uma aresta-de-igualdade não
solução une um nó externo i a um nó ímpar j emparelhado a
um nó original k que não formam uma dupla emparelhada. O botão é
constituído do nó k e de todos os nós emparelhados a k , ex
ceto o nó j , o qual recebe um rótulo interno.

OBSERVAÇÃO: Todos os nós de um botão detetado e todas as arestas unindo estes nós são comprimidos em um pseudonó. Assim, cada aresta que une um nó do botão a um nó fora do botão passa a ser incidente com este novo pseudonó. Atribua rótulo externo ao pseudonó (botão comprimido) e remova os rótulos dos nós do botão comprimido.

Ampliação - executada quando uma aresta-de-igualdade não solução une dois nós externos de árvores alternantes diferentes. A operação consiste em trocar o papel das arestas solução/não solução no caminho aumentante $P_i \cup \{(i;j)\} \cup P_j$. Observe que os nós raízes tornam-se emparelhados.

Caso exista um pseudonó no caminho aumentante, sejam i, j os nós do botão associado que são incidentes com a aresta-de-igualdade não solução e com a aresta-de-igualdade solução do caminho aumentante, respectivamente. Caso o pseudonó seja exposto, j deve ser o nó exposto do botão. Se $i \neq j$ troque o papel solução/não solução das arestas no caminho alternante entre i, j no botão; para botões com ciclo ímpar o caminho considerado deve ser o caminho alternante par. Este procedimento deve ser repetido dentro de cada pseudonó deste caminho.

Após cada ampliação, os rótulos de todas as árvores devem ser eliminados.

Expansão de pseudonós em botões - executada quando existe um pseudonó S interno com preço nulo.

Seja i o nó externo cujo predecessor é S . Desta forma, $(i;S)$ é uma aresta-de-igualdade não solução e $(S;j)$ é uma aresta-de-igualdade solução. Substitua o pseudonó S pelo seu botão. Sejam i',j' os nós do botão de S que são incidentes com as arestas $(i;S), (S;j)$ e seja k a base do botão. Considere o caminho alternante entre i,j . Caso o botão não seja um ciclo ímpar, execute um dos itens abaixo.

i) Se o caminho alternante entre i,j possui quatro arestas e S possui mais de três nós (algum nó além de k,i',j') comprima o botão formado pelos nós de S (excluindo os nós i',j') em um novo pseudonó (Fig.2).

ii) Se este caminho possui três arestas, comprima o botão formado pelos nós de S excluindo j' e incluindo i em um novo pseudonó (Fig. 3).

iii) Se este caminho possui duas arestas e $k = i' = j'$ então comprima o botão formado pelos nós de S (incluindo i,j) em um novo pseudonó. Atribua rótulo externo a este pseudonó (Fig. 4).

iv) Se este caminho possui duas arestas e $k \neq i',j'$ ou se este caminho possui quatro arestas e S consiste de k,i',j' apenas, então não é necessário comprimir nenhum botão (Fig. 5)

Atribua rótulos internos, externos aos nós no caminho alternante entre i, j) (exceto caso iii)). Caso o botão S tenha sido um ciclo ímpar, considere o caminho alternante par entre i, j (Fig. 6).

A cada atribuição de rótulo interno a um pseudonó com preço nulo, repita este procedimento para expandi-lo. Se foi feita a atribuição de um rótulo interno a um nó ímpar com preço nulo, ou a atribuição de um rótulo a algum nó par, realize uma compressão do tipo 2.

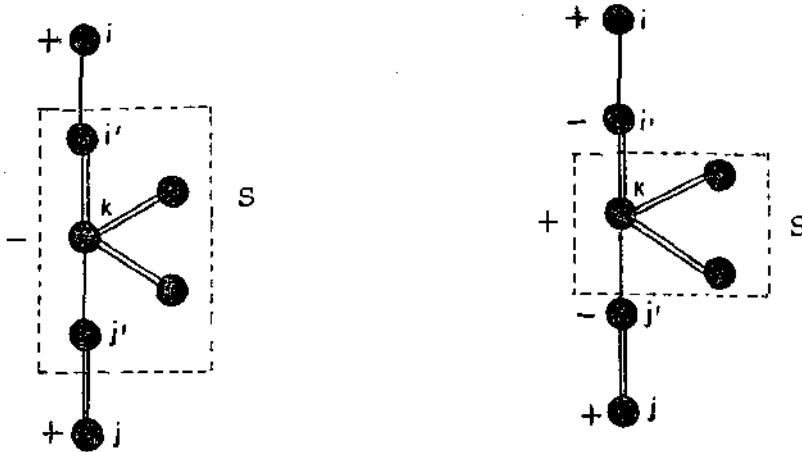


FIGURA 2.

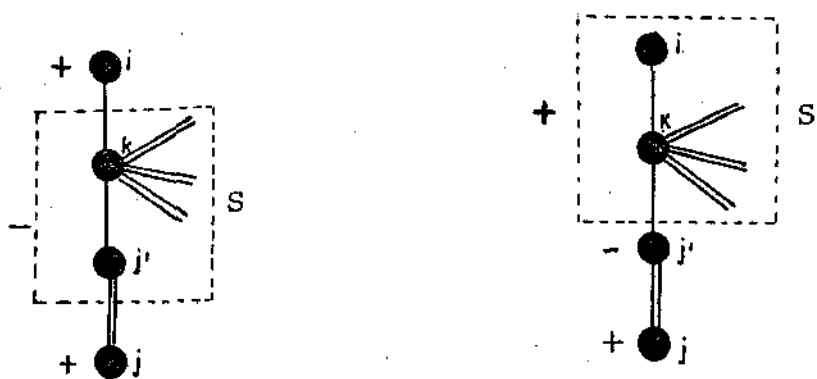


FIGURA 3



FIGURA 4

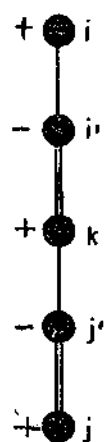
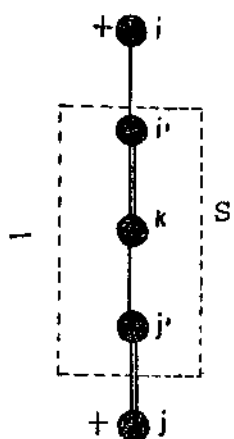
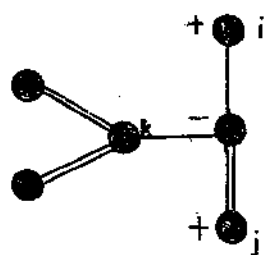
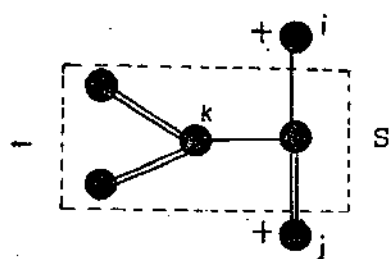


FIGURA 5

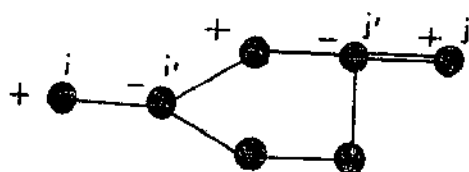
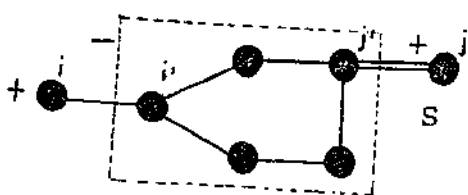


FIGURA 6

ALGORITMO DO BOTÃO

INICIALIZAÇÃO

0. Atribua

$$x_{ij} \leftarrow 0 \quad (i;j) \in A$$

$$u_s \leftarrow 0 \quad s \in S$$

1. Atribua rótulos externos aos nós correntes expostos. Se não há nós expostos, termine.

INVESTIGAÇÃO DA SOLUÇÃO PRIMAL

2. Se existe uma aresta-de-igualdade não solução unindo dois nós externos de árvores alternantes diferentes, execute uma ampliação e vá para 1.

3. Se existe uma aresta-de-igualdade não solução unindo dois nós externos da mesma árvore alternante, execute a compressão tipo 1 e vá para 2.

4. Se existe uma aresta-de-igualdade não solução unindo um nó externo i a um nó original k que é ímpar com preço nulo ou é par, execute a compressão tipo 2 e vá para 2.

5. Se existe uma aresta-de-igualdade não solução unindo um nó

externo i a um nó ímpar j emparelhado a um nó original k que não forma uma dupla emparelhada, execute a compressão tipo 3 e vá para 2.

6. Se existe uma aresta-de-igualdade não solução unindo um nó externo i a um nó ímpar j (não original ou com preço não nulo) que forma uma dupla emparelhada j,k , execute o crescimento e vá para 2.

7. Se existe um pseudonó interno com preço nulo, execute a expansão apropriada (i, ii, iii, iv ou v), e vá para 2.

MUDANÇA DA SOLUÇÃO DUAL

8. Seja $R(i)$ o rótulo do nó i , ou do pseudonó corrente que o contém.

$$\epsilon_1 = \min\{1/2 d_{ij} : R(i), R(j) \text{ são externos}\}$$

$$\epsilon_2 = \min\{d_{ij} : R(i) \text{ externo, } R(j) \text{ sem rótulo}\}$$

$$\epsilon_3 = \min\{\mu_S : R(S) \text{ interno}\}$$

$$\epsilon = \min\{\epsilon_1, \epsilon_2, \epsilon_3\}$$

Para cada $S \in S$, faça

$$\mu_S \leftarrow \begin{cases} \mu_S + \epsilon & \text{se } R(S) \text{ é externo} \\ \mu_S - \epsilon & \text{se } R(S) \text{ é interno} \\ \mu_S & \text{c.c.} \end{cases}$$

vá para 2.

Observe que as condições abaixo são sempre satisfeitas

$$x_{ij} = 0,1 \quad (i;j) \in A$$

$$\mu_S \geq 0 \quad S \in S$$

$$d_{ij} = c_{ij} - \sum_S [\mu_S : (i;j) \in \delta(S)] \geq 0 \quad (i;j) \in A$$

$$d_{ij} > 0 \implies x_{ij} = 0 \quad (i;j) \in A$$

$$\sum [x_{ij} : (i;j) \in \delta(S)] > 1 \implies \mu_S = 0 \quad S \in S$$

e quando algoritmo termina com $\sum [x_{ij} : (i;j) \in \delta(S)] \geq 1$ então as condições acima garantem que a solução é ótima.

Observe também que as condições abaixo são sempre satisfeitas imediatamente antes da execução de uma mudança da solução dual.

$$d_{ij} > 0 \quad (i;j) \in A \quad \text{tal que} \quad R(i), R(j) \quad \text{externos}$$

$$d_{ij} > 0 \quad (i;j) \in A \quad \text{tal que} \quad (R(i) \text{ externo}, R(j) \text{ sem rótulo}) \\ \text{ou} \quad (R(i) \text{ sem rótulo}, R(j) \text{ externo}).$$

$$\mu_S > 0 \quad S \in S \quad \text{tal que} \quad R(S) \quad \text{interno.}$$

Estas condições garantem que ϵ é estritamente positivo a cada execução do passo 8.

Uma rede com n nós e m arestas possui no máximo n nós

ímpares inicialmente; ou seja são executadas $O(n)$ ampliações. Após uma mudança da solução dual é executada uma ampliação, ou uma compressão ou uma expansão, ou um crescimento da árvore. Entre duas ampliações, é possível ocorrer $O(n)$ compressões, $O(n)$ expansões e $O(n)$ crescimentos. Desde que uma ampliação, ou uma compressão, ou uma expansão, ou um crescimento são realizados com esforço computacional $O(n)$ e a mudança da solução dual pode ser realizada com esforço $O(m+n)$ tem-se que a complexidade do algoritmo é $O(n) \cdot O(n) \cdot O(n+m) = O(n^2 m)$.

2. ALGORITMO PARA ROTA EULERIANA

Sejam $L(i, \cdot)$ = lista das arestas que serão utilizadas para sair do nó i .

k_i = número de arestas nesta lista (a ser determinado).

α_{ij} = número de vezes que a linha é utilizada na rota do carteiro (conhecido).

0. $k_i \leftarrow 0, \forall i; i' \leftarrow 1, i \leftarrow i'$

1. Se existe aresta $(i;j)$ com $\alpha_{ij} > 0$ então faça $\alpha_{ij} \leftarrow \alpha_{ij} - 1$, $k_j \leftarrow k_j + 1, L(j, k_j) \leftarrow (i;j), i \leftarrow j$, vá para 1.

2. Se $i' = n$, pare.

Senão $i' \leftarrow i' + 1$

$i \leftarrow i'$

vã para 1.

(construção da Rota Euleriana)

3. $j \leftarrow 1$

4. Se $k_j = 0$ então pare, senão faça:

$(i;j) \leftarrow L(j,k_j)$

imprima $((i;j))$

$k_j \leftarrow k_j - 1$

$j \leftarrow i$

vã para 4.

CAPÍTULO 3

REDES ORIENTADAS

TEOREMAS

Uma condição necessária e suficiente para a existência de uma rota do carteiro é que a rede orientada G seja fortemente conectada, a menos de nós isolados.

Uma condição necessária e suficiente para a existência de uma rota euleriana é que a rede fortemente conectada G seja simétrica [Be 1].

Se a rede fortemente conectada $G = (N, A, C)$ com $C \geq 0$ é não simétrica, então alguns arcos devem ser usados mais do que uma vez para formar uma rota do carteiro. Seja $(1 + x_{ij})$ o número de vezes que o arco (i, j) deve ser usado pela rota do carteiro. Deseja-se obter $X = (x_{ij})$, a solução ótima de

$$\text{Min} \quad \sum_{i,j} c_{ij} x_{ij}$$

$$\text{sujeito a} \quad \text{gr}^-(i) + \sum_j x_{ji} = \text{gr}^+(i) + \sum_j x_{ij} \quad i \in N$$

$$x_{ij} \geq 0 \quad (i,j) \in A$$

$$x_{ij} \text{ inteiro} \quad (i,j) \in A$$

Ao resolver este problema, obtêm-se uma rede simétrica construída a partir de G acrescentando $1 + x_{ij}$ cópias do arco

(i,j) . Uma formulação equivalente para o problema acima é:

$$\text{Min} \quad \sum_{i,j} c_{ij} x_{ij}$$

$$\text{sujeito a} \quad \sum_j x_{ij} - \sum_j x_{ji} = d_i \quad i \in N$$

$$x_{ij} \geq 0 \quad (i,j) \in A$$

$$x_{ij} \text{ inteiro} \quad (i,j) \in A$$

onde $d_i = gr^-(i) - gr^+(i)$.

A quantidade de cópias de cada arco a ser determinada é numericamente igual ao fluxo ótimo do Problema de Fluxo com custo mínimo em G onde a demanda de cada nó i é expressa por d_i .

O algoritmo para determinar a rota euleriana apresentado para redes não-orientadas (vide Capítulo 2, seção 2) pode ser adaptado para este caso.

CAPÍTULO 4

REDES MISTAS

TEOREMAS

Uma condição necessária e suficiente para a existência de uma rota do carteiro é que a rede mista G seja fortemente conectada, a menos de nós isolados. Observe que as arestas podem ser usadas com qualquer uma das direções possíveis.

Uma condição necessária e suficiente para a existência de uma rota euleriana é que a rede fortemente conectada G seja par e simétrica.

Em geral, para redes mistas fortemente conectadas, o PCC é NP-árduo (NP-"hard") [GJ 1], com algumas exceções:

i) $G = (N, A, C)$ com $C \geq 0$ par e simétrica.

A rota do carteiro ótima é uma rota euleriana.

ii) $G = (N, A, C)$ com $C \geq 0$ par e não simétrica.

Neste caso, G pode ser transformada em uma rede par e simétrica com o algoritmo abaixo, de tal forma que ambas as redes apresentem a mesma rota do carteiro ótima.

A partir da rede $G = (N, A, C)$ construa a rede $G' = (N, A', C')$ com $A' = A^0 \cup A^1 \cup A^2$, tal que $A^0 = \{(i, j) : \text{arco } (i, j) \in A\}$, $A^1 = \{[i, j], [j, i] : \text{aresta } (i, j) \in A\}$ e $A^2 = \{\langle i, j \rangle, \langle j, i \rangle : \text{aresta } (i, j) \in A\}$.

$(i;j) \in A$ de tal forma que A^0, A^1, A^2 são conjuntos distintos de arcos; isto é, os arcos $(i,j), [i,j], \langle i,j \rangle$ são distintos. Além disto,

$$c_{ij}^0 = c_{ij} \quad (i,j) \in A$$

$$c_{ij}^1 = c_{ij} \quad (i;j) \in A$$

$$c_{ij}^2 = 0 \quad (i;j) \in A$$

Sejam $x_{ij}^0, x_{ij}^1, x_{ij}^2$ o número de vezes que os arcos $(i,j) \in A^0$, $[i,j] \in A^1$, $\langle i,j \rangle \in A^2$ são atravessados, respectivamente. O problema a ser resolvido é

$$\text{Min} \quad \sum c_{ij}^0 x_{ij}^0 + \sum c_{ij}^1 x_{ij}^1$$

$$\text{sujeito a} \quad x_{ij}^0 \geq 0 \quad \text{e inteiro} \quad (i,j) \in A^0$$

$$x_{ij}^1 \geq 0 \quad \text{e inteiro} \quad [i,j] \in A^1$$

$$x_{ij}^2 = 0,1 \quad \langle i,j \rangle \in A^2$$

$$\sum_j x_{ij}^0 + \sum_j x_{ij}^1 + \sum_j x_{ij}^2 - \sum_j x_{ji}^0 - \sum_j x_{ji}^1 - \sum_j x_{ji}^2 = d_i \quad i \in N$$

onde $d_i = \text{gr}^-(i) - \text{gr}^+(i)$, $i \in N$

$$\text{gr}^-(i) = \sum_j x_{ji}^0$$

$$gr^+(i) = \sum_j x_{ij}^0$$

O algoritmo de Fluxo de Custo Mínimo resolve este problema determinando o número de reutilização das linhas de G . A rota euleriana pode ser determinada com uma adaptação do algoritmo apresentado para redes não-orientadas (Capítulo 2, seção 2).

EXTENSÕES

De maneira genérica, o Problema do Carteiro Chinês pode ser modificado para problemas em que:

- i) existem algumas linhas que podem ser utilizadas, mas que não necessariamente devem estar na solução, e/ou
- ii) existem algumas linhas que devem ser utilizadas exatamente uma única vez.

Além da determinação de uma rota para um carteiro, existem outras aplicações, tais como:

- coleta de lixos [BB 1]
- entrega de mercadorias [St 1].

PARTE II: O PROBLEMA DO CAIXEIRO VIAJANTE

CAPÍTULO 1

INTRODUÇÃO

Há 55 anos atrás [Me 1], o Problema do Caixeiro Viajante (PCV) já aparecia formulado, mas somente nestes últimos 30 anos é que seu estudo tem destacado-se. Dado um conjunto de cidades e a distância entre cada par de cidades, o PCV consiste em determinar uma rota de comprimento total mínimo. Uma rota do caixeiro viajante visita cada uma das cidades e tem início e término na mesma cidade.

Além desta aplicação clássica, o PCV é também utilizado na modelagem de diversos problemas: sequenciamento cíclico de tarefas em uma máquina [Ba 1]; sequenciamento cíclico de tarefas em m máquinas [Ba 1]; m -caixeiros [Ch 1]; programação de vôos [Ch 1]; planejamento de conexões elétricas em computadores [LR 1]; agrupamento de dados [LR 1]; etc. (vide Capítulo 5).

Uma rota hamiltoniana é uma rota do caixeiro que visita cada cidade exatamente uma vez. O problema de determinar se um dado grafo possui uma rota hamiltoniana é NP-completo [GJ 1], ou seja, não é conhecido nenhum algoritmo polinomial que resolva este problema. Além disto, o PCV restrito à rotas hamiltonianas é NP-arduo (NP-"hard") [GJ 1]. Entretanto, a literatura apresenta

algoritmos eficientes para casos particulares do PCV: sequenciamento de tarefas em fornos [CG 1]; matriz de distância triangular superior [La 1]; redução de grandes problemas [Ro 1, BN 1, Sz 1]; coleta de itens em almoxarifados [RR 1]; etc. (vide Capítulo 6).

Diversos métodos são utilizados para resolver o PCV na sua forma geral, dentre eles, destacam-se: Planos de Corte de Programação Inteira [GN 1]; Programação Dinâmica [DL 1]; Otimização com Subgradiente [BS 1]; e, principalmente, Particionar e Limitar ("Branch-and-Bound") [GN 1]. Neste último método a determinação de valores limitantes (mínimos) da rota hamiltoniana ótima é de grande importância para a sua eficiência e pode estar baseado em: designação [BC 1]; 2-emparelhamento [Ch 1]; 1-árvore [HK 1, HK 2]; 1-arborescência [Ch 1, PR 1]; n-rota [Ch 1]. Em termos de modelos para o PCV é conveniente salientar também a transformação do PCV em um problema da Cadeia Máxima [HN 1] (vide Capítulo 3).

Diversos métodos heurísticos (algoritmos aproximados ou não exatos) já foram desenvolvidos para o PCV. Tratam-se de algoritmos polinomiais que tem por objetivo obter soluções quase ótimas. Dentre eles, destacam-se: cidade não-visitada mais próxima; inserção da cidade mais próxima; Lin; árvore geradora mínima; Cristofides [Ch 1, PR 1] (vide Capítulo 4).

Condições necessárias ou suficientes para a existência de uma rota hamiltoniana para um dado PCV são estudadas em [Be 1, Ha 1, PR 1] (vide Capítulo 2).

O estudo do conjunto de soluções viáveis conduz à tentativa da caracterização do assim chamado politopo do caixeiro viajante. Os principais tipos de desigualdades contidas nesta caracterização são: canalização da variável, não-negatividade, eliminação de subrotas, pente ("comb") [CP 1, CP 2] e 2-emparelhamento [Ed 2] (vide Capítulo 2).

Uma análise geral dos estudos acima encontra-se em [Ch 1, BN 1, PR 1].

A maioria dos algoritmos apresentados na literatura fornecem rotas que visitam cada cidade exatamente uma vez. Nesta dissertação é desenvolvido um algoritmo do tipo Particionar e Limitar para o PCV, cuja rota ótima pode visitar uma cidade mais do que uma vez. Problemas de cobertura simétrica são utilizados para obter limitantes para os subproblemas gerados (vide Capítulo 7).

Finalmente, no Capítulo 8 é feita uma breve apresentação do Problema do Caixeiro Viajante tipo Gargalo ("Bottleneck") [CM 1].

CAPÍTULO 2

CARACTERÍSTICAS E PROPRIEDADES

1. CONCEITOS BÁSICOS

DEFINIÇÕES

$G = (N, A)$ é um *grafo* se N é um conjunto finito não-vazio de n *nós* (ou cidades) e A é um conjunto finito não-vazio de m *linhas* (ou ligação entre duas cidades), cada linha pode ser uma *aresta* (isto é, par não-ordenado $(i; j)$ de nós distintos) ou um *arco* (isto é, par ordenado (i, j) de nós distintos). Se (i, j) é um arco, i, j são a *cauda* e a *cabeça* do arco, respectivamente. Dois nós são *adjacentes* se existe uma linha unindo-os.

Se todas as linhas de G são *arcos* então G é um *grafo orientado*. Se todas as linhas de G são *arestas* então G é um *grafo não-orientado*. Caso contrário G é um *grafo misto*.

Seja $C = (c_{ij})$ uma matriz de distância em G , isto é, c_{ij} é o comprimento (custo ou tempo) associado ao arco (i, j) ou a aresta $(i; j)$ para cada linha de G . Se G é não-orientado o PCV associado é denominado *simétrico*, caso contrário é denominado *assimétrico*.

Dado um grafo $G = (N, A)$, considere uma sequência finita $P = (i_1, l_1, i_2, l_2, \dots, i_{q-1}, l_{q-1}, i_q)$ onde $i_1, \dots, i_q$ são nós de G e cada l_j é uma linha de G que une os nós i_j, i_{j+1} para $j = 1, \dots, q-1$.

P é um *caminho* conectando i_1, i_q se $i_1, \dots, i_q$ são nós distintos.

P é uma *cadeia* conectando i_1, i_q se $i_1, \dots, i_q$ são nós distintos e cada ℓ_j é um arco de i_j para i_{j+1} .

P é um *ciclo* se P é um caminho com $i_1 = i_q$.

P é um *circuito* se P é uma cadeia com $i_1 = i_q$.

P é uma *rota* se $i_1 = i_q$.

P é uma *rota do caixeiro*, se $i_1 = i_q$ e $\{i_1, i_2, \dots, i_q\} = N$.

P é uma *rota hamiltoniana* se $i_1 = i_q$, $\{i_1, i_2, \dots, i_q\} = N$ e $q = n$.

P é uma *subrota* se $i_1 = i_q$ e $\{i_1, i_2, \dots, i_q\} \subsetneq N$

P é uma *rota euleriana* se $i_1 = i_q$, $\{\ell_1, \ell_2, \dots, \ell_{q-1}\} = A$ e $q-1 = m$.

G é um *grafo hamiltoniano* se possui uma rota hamiltoniana.

G é um *grafo euleriano* se possui uma rota euleriana.

A denominação de rota hamiltoniana é devido Willian Rowan Hamilton, que foi o primeiro a estudar a determinação de uma rota visitando exatamente uma vez cada vértice de um dodecaedro.

2. ROTAS HAMILTONIANAS

Embora não seja conhecida nenhuma caracterização completa de grafos hamiltonianos (isto é, grafos que possuem pelo menos uma rota hamiltoniana), várias condições necessárias ou suficientes são disponíveis.

DEFINIÇÕES

Considere um grafo $G = (N, A)$

G é *completo* se cada par de nós é unido por uma linha (aresta ou arco).

G é *conectado* se cada par de nós é conectado por um caminho.

$\underline{G} = (\underline{N}, \underline{A})$ é um *subgrafo* de G se $\underline{N} \subset N, \underline{A} \subset A$.

$\underline{G} = (\underline{N}, \underline{A})$ é um *subgrafo maximal* de G com respeito a uma certa propriedade se:

- i) $\underline{G}$ é um subgrafo de G e apresenta a propriedade.
- ii) a inclusão de qualquer linha em $\underline{G}$ implica em $\underline{G}$ deixar de ser um subgrafo de G ou na perda da propriedade.

$\underline{G}$ é uma *componente* de G se $\underline{G}$ é um subgrafo maximal conectado de G .

Um nó de G é um *nó de corte* se a remoção deste nó e das linhas incidentes com ele aumenta o número de componentes de G .

G é *não-separável* se G é conectado e não possui nó de corte.

$\underline{G}$ é um *bloco* de G se $\underline{G}$ é um subgrafo maximal não-separável de G .

DEFINIÇÕES - GRAFO NÃO-ORIENTADO

Considere um grafo $G = (N, A)$ não-orientado.

$gr(i)$ denota o grau do nó i de G , isto é, o número de arestas de G incidentes com i .

G é um *grafo-teta* se G é um bloco com 2 nós não-adjacentes de grau 3 e todos os outros de grau 2 (Fig. 2.1).

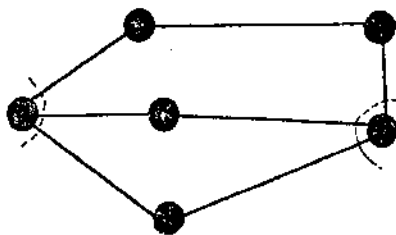


FIGURA 2.1 GRAFO-TETA

$G' = (N', A')$ é um *grafo-aresta* de G se $N' = A$ e $A' = \{(i, j) : i, j \text{ são arestas adjacentes em } G\}$, isto é, os nós de G são as arestas de G e dois nós de G' são adjacentes sempre que as arestas correspondentes de G o forem (Fig. 2.2).

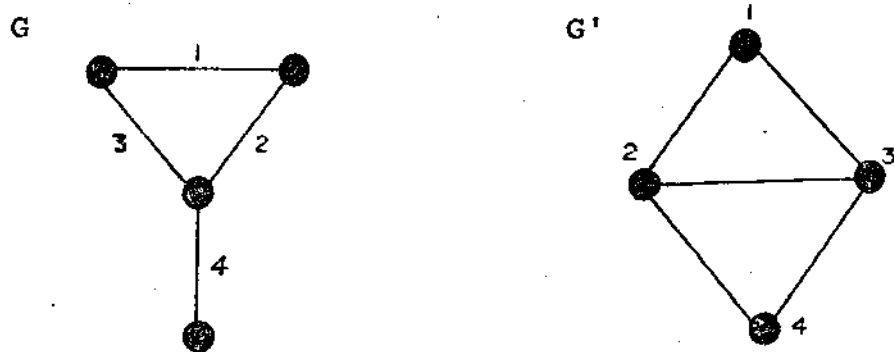


FIGURA 2.2 GRAFO E SEU GRAFO-ARESTA

$\bar{G}$ é um *supergrafo* de G se G é um subgrafo de $\bar{G}$.

$\bar{G}$ é um *supergrafo minimal* de G com respeito a uma certa propriedade se:

- i) $\bar{G}$ é um supergrafo de G e apresenta a propriedade;
- ii) se a exclusão de qualquer aresta de $\bar{G}$ implica em $\bar{G}$ deixar de ser um supergrafo de G ou na perda da propriedade.

$\bar{G} = (N, \bar{A})$ é o *fechamento* de $G = (N, A)$ se $\bar{G}$ é um supergrafo minimal de G e se

$$\forall (i; j) \notin A \implies \bar{gr}(i) + \bar{gr}(j) < n$$

onde $\bar{gr}$ denota o grau dos nós em $\bar{G}$.

TEOREMA [Hal]. Se o grafo G é hamiltoniano então G é não-separável.

PROVA: Seja $W = (i_1, i_2, \dots, i_1)$ uma rota hamiltoniana. Para qualquer par de nós i_p, i_q de W , existem dois caminhos distintos $i_p, i_{p+1}, i_{p+2}, \dots, i_q$ e $i_q, i_{q+1}, i_{q+2}, \dots, i_p$ conectando-os. Isto satisfaz a definição de grafo não-separável.

TEOREMA [Hal]. G' é um grafo-aresta de G se e só se as arestas de G' podem ser particionadas em subgrafos completos tais que nenhum nó esteja em três ou mais destes subgrafos.

TEOREMA [Ha 1]. Se um grafo G é euleriano então o seu grafo-aresta é hamiltoniano.

TEOREMA (Pósa [Po 1]) [Ha 1]. Um grafo G não-separável sem subgrafo - teta possui rota hamiltoniana.

Um grupo de condições suficientes para um grafo não orientado G ser hamiltoniano, respeitando a ordem de suficiência é apresentado abaixo [PR 1].

Suponha que os nós estão ordenados de tal forma que:

$$gr(1) \leq gr(2) \leq gr(3) \leq \dots \leq gr(n).$$

$$(i) \quad 1 \leq k \leq n \implies gr(k) \geq n/2$$

$$(ii) \quad (i;j) \notin A \implies gr(i) + gr(j) \geq n$$

$$(iii) \quad 1 \leq k < n/2 \implies gr(k) > k$$

$$(iv) \quad j < k, gr(j) \leq j, gr(k) \leq k-1 \implies gr(j) + gr(k) \geq n$$

$$(v) \quad gr(k) \leq k < n/2 \implies gr(n-k) \geq n-k.$$

$$(vi) \quad j < k, k \geq n-j, (j;k) \in A, gr(j) \leq j, gr(k) \leq k-1 \implies \\ \implies gr(j) + gr(k) \geq n.$$

$$(vii) \quad \bar{G} \text{ (o fechamento de } G) \text{ é um grafo completo.}$$

Algoritmo Polinomial que constrói $\bar{G} = (N, \bar{A})$ para obter uma rota hamiltoniana [PR 1].

0. Faça $\bar{A} \leftarrow A$

$$\overline{gr}(i) \leftarrow gr(i), \forall i \in N$$

$$p \leftarrow 2$$

$$r(i,j) \leftarrow \begin{cases} 1 & \text{se } (i;j) \in A \\ 0 & \text{c.c.} \end{cases}$$

1. Se existe (i,j) com $i < j$, $r(i,j) = 0$ e $\overline{gr}(i) + \overline{gr}(j) \geq n$, então faça:

$$r(i,j) \leftarrow p$$

$$\overline{gr}(i) \leftarrow \overline{gr}(i) + 1$$

$$\overline{gr}(j) \leftarrow \overline{gr}(j) + 1$$

$$p \leftarrow p + 1$$

vã para 1.

Se $\bar{G} = (N, \bar{A})$ não é completo, pare; pois é possível que não exista uma rota hamiltoniana.

$$\text{Faça } r(j,i) \leftarrow r(i,j), i < j$$

2. Escolha uma rota hamiltoniana $W = (i_1, i_2, \dots, i_q, i_1)$ em $\bar{G}$ tal que $r(i_q, i_1) \geq r(i_s, i_{s+1})$, $s = 1, 2, \dots, q-1$. Se $r(i_q, i_1) = 1$ pare, W é uma rota hamiltoniana de G .

Escolha um nó i_p em W tal que $r(i_1, i_{p+1}) < r(i_q, i_1)$ e $r(i_q, i_p) < r(i_q, i_1)$.

$$\text{Faça } W \leftarrow (i_1, i_{p+1}, i_{p+2}, \dots, i_q, i_p, i_{p-1}, \dots, i_1)$$

vã para 2.

DEFINIÇÕES - GRAFO ORIENTADO

Considere um grafo $G = (N, A)$ orientado.

G é *fortemente conectado* se para cada par de nós i, j existe uma cadeia de i para j e uma cadeia de j para i .

$gr^+(i)$ denota o *grau externo* do nó i de G , isto é, o número de arcos de G incidentes de i .

$gr^-(i)$ denota o *grau interno* do nó i de G , isto é, o número de arcos de G incidentes em i .

TEOREMA. Se o grafo G é hamiltoniano então G é fortemente conectado.

PROVA: Segue-se do fato que uma rota hamiltoniana contém uma cadeia entre cada par de nós de G .

TEOREMA [Be 1]. Se G é um grafo completo fortemente conectado então G é hamiltoniano.

TEOREMA [Be 1]. Se G é um grafo fortemente conectado e $gr(i) \geq n$, $\forall i$ então G é hamiltoniano

COROLÁRIO [Be 1]. Se G é um grafo fortemente conectado e $gr^+(i) \geq \frac{n}{2}$, $gr^-(i) \geq \frac{n}{2}$, $\forall i$ então G é hamiltoniano.

PROPRIEDADES

$G = (N, A, C)$ é uma rede se (N, A) é um grafo e $C = (c_{ij})$ é uma matriz de distância (tempo ou custo) no grafo (N, A) . Uma rede é orientada, não-orientada, conectada, fortemente conectada, não-separável, completa, de acordo com o grafo associado.

Para o PCV restrito a rotas hamiltonianas, pode-se supor $C = (c_{ij}) \geq 0$, pois o conjunto de soluções ótimas do PCV permanece inalterado ao adicionar uma constante arbitrária a cada linha ou coluna da matriz de distância C .

Considere $G = (N, A, C)$ uma rede orientada fortemente conectada.

1) Se a desigualdade triangular é satisfeita, isto é, para cada par de nós i, j de G ,

$$c_{ij} \leq c_{ik} + c_{kj} \quad \forall k \neq j \neq i$$

então a rota do caixeiro ótima é uma rota hamiltoniana mínima.

2) Se a desigualdade triangular não é satisfeita e cada arco pode ser utilizado no máximo uma vez na rota do caixeiro, então a rota do caixeiro ótima é uma cobertura simétrica em G , isto é, para cada nó de G o número de arcos que chega é igual ao número de arcos que sai.

3) Se a desigualdade triangular não é satisfeita e não há

restrições sobre o número de vezes que cada arco é atravessado na rota do caixeiro, então a rota do caixeiro ótima em G é uma rota hamiltoniana mínima em $G' = (N, A', C')$ onde c'_{ij} é o comprimento da cadeia mínima entre os nós i, j .

Considere a rede $G = (N, A, C)$ não-orientada não-separável.

1) Se a desigualdade triangular é satisfeita, então a rota do caixeiro ótima é uma rota hamiltoniana mínima.

2) Se não é satisfeita a desigualdade triangular e cada aresta pode ser utilizada no máximo uma vez na rota do caixeiro, então a rota do caixeiro ótima é uma cobertura par em G , isto é, para cada nó de G , o número de arestas incidentes é par.

3) Se não é satisfeita a desigualdade triangular e não há restrições sobre o número de vezes que cada aresta é atravessada na rota do caixeiro, então a rota do caixeiro ótima é a rota hamiltoniana mínima em $G' = (N, A', C')$ onde c'_{ij} é o comprimento do caminho mínimo entre os nós i, j .

Algoritmo para determinar uma rota hamiltoniana [PR.1]

(sem usar condições suficientes)

Seja $G = (N, A)$ o grafo para o qual se deseja determinar uma rota hamiltoniana. Crie o grafo orientado de trabalho $G_0 = (N, A_0)$ que possui um arco (i, j) para cada arco $(i, j) \in A$ e que possui os arcos $(i, j), (j, i)$ para cada aresta $(i; j) \in A$.

No algoritmo, $G_p = (N, A_p)$ é um grafo de trabalho no estágio p e A_p^* é um conjunto de arcos que forma cadeias que são requeridas na rota hamiltoniana.

0. Faça $p \leftarrow 0$; $A_p^* \leftarrow \emptyset$. Determine $G_0 = (N, A_0)$ a partir de $G = (N, A)$.

1. Para cada $(i, j) \in A_p^*$ elimine de A_p todos os arcos (i, k) com $k \neq j$ e todos os arcos (k, j) com $k \neq i$.

Para cada sequência de nós $(i_0, i_1, \dots, i_q)$ formando uma cadeia em A_p^* elimine de A_p todos os arcos (i_q, i_s) com $s = 0, 1, \dots, q-1$.

Se existe $(i, j) \notin A_p^*$ que é o único arco de A_p com cauda em i ou que é o único arco de A_p com cabeça em j , introduza (i, j) em A_p^* e vá para 1.

2. Se A_p^* forma uma rota hamiltoniana, pare.

Se existe $i \in N$ tal que não existe arco em A_p com cabeça em i ou se não existe arco em A_p com cauda em i , então G_p é não hamiltoniano. Neste caso, pare se $p = 0$ (G é não hamiltoniano), senão faça $p \leftarrow p + 1$ e vá para 1.

Seja $i \in N$ tal que não existe arco em A_p^* com cauda em i . Sejam $(i, j_1), \dots, (i, j_k)$ os arcos de A_p com cauda em i . Para $q = 1, 2, \dots, k$ crie $G_{p+q} = (N, A_{p+q})$, A_{p+q}^* com $A_{p+q} = A_p$, $A_{p+q}^* = A_p^* \cup \{(i, j_q)\}$. Faça $p \leftarrow p + k$ e vá para 1.

3. ADJACÊNCIA NO POLÍTOPO DO PROBLEMA DA DESIGNAÇÃO

Considere o problema de designação:

$$\text{Min} \quad \sum_{i,j} c_{ij} x_{ij}$$

$$\text{sujeito a} \quad \sum_j x_{ij} = 1 \quad i \in N$$

$$\sum_j x_{ji} = 1 \quad i \in N$$

$$x_{ij} = 0, 1 \quad (i,j) \in A$$

onde $C = (c_{ij})$ é uma matriz $n \times n$ de distância. Uma designação $X = (x_{ij})$ é uma matriz $n \times n$ de variáveis 0 - 1 onde cada linha e cada coluna apresenta exatamente uma variável com valor 1.

Claramente, qualquer rota hamiltoniana do Problema do Caixeiro Viajante ($x_{ij} = 1$ indica que da cidade i o caixeiro se dirige para a cidade j) satisfaz as restrições acima. Isto é, cada rota hamiltoniana é uma designação, entretanto, algumas designações não correspondem a rotas hamiltonianas.

Denominando por *laço* uma variável do tipo x_{ii} com valor 1 e denominando a designação com n laços por *designação diagonal*, é possível classificar as designações em:

- i) designação diagonal;
- ii) rotas hamiltonianas;
- iii) diversas subrotas envolvendo um total de n cidades;
- iv) k laços e diversas subrotas envolvendo um total de $n-k$ cidades.

Seja Q_D o politopo do Problema de Designação, caracterizado pelas desigualdades [Mu 2]:

$$\sum_j x_{ij} = 1 \quad i \in N$$

$$\sum_j x_{ji} = 1 \quad i \in N$$

$$x_{ij} \geq 0 \quad (i,j) \in A$$

É possível provar que:

- i) o posto da matriz de coeficientes das restrições de igualdade é $2n-1$;

ii) o sistema acima possui soluções básicas que são as designações;

iii) cada designação está associada a $2^{n-1} n^{n-2}$ bases.

DEFINIÇÃO: Duas designações distintas x^1 e x^2 são *adjacentes* se o segmento de reta que as une é uma *aresta* de Q_D , isto é, se cada ponto da forma $\alpha x^1 + (1 - \alpha)x^2$, $\alpha \in [0,1]$ tem uma única representação como uma combinação convexa de designações.

TEOREMA [He 1]. Uma designação é adjacente à designação diagonal se e somente se ela é:

- uma rota hamiltoniana
- ou uma subrota e um ou mais laços.

COROLÁRIO 1 [He 1] Atribuindo para todo i , $c_{ii} = \infty$ (para evitar laços), o PCV pode ser enunciado como: "A partir da designação diagonal, determinar uma designação adjacente cujo valor da função objetivo seja mínimo".

COROLÁRIO 2 [He 1] Se $(i_1, i_2), (i_2, i_3), \dots, (i_{n-1}, i_n)$ são os elementos da designação com valor 1 e o posto das colunas correspondente às variáveis

$$(x_{11}, \dots, x_{nn}, x_{i_1 i_2}, x_{i_2 i_3}, \dots, x_{i_{n-1} i_n})$$

é $2n-1$. Então $(i_1, i_2, \dots, i_n)$ é uma rota hamiltoniana e vice-versa.

4. POLITOPO

Considere um problema de programação inteira com um conjunto de soluções viáveis (inteiras) finito. O *politopo* Q deste problema é a envoltória convexa (conjunto de todas as combinações convexas) destas soluções viáveis.

A *dimensão* de um espaço vetorial é definida pelo número máximo de vetores linearmente independentes que ele comporta e a dimensão de um politopo Q , $\dim_Q$ é o número máximo q de pontos $x^0, x^1, \dots, x^q \in Q$, tais que $\{x^1 - x^0, \dots, x^q - x^0\}$ é um conjunto linearmente independente.

Sejam $a \in \mathbb{R}^n$, $a_0 \in \mathbb{R}$. Considere a desigualdade $ax \geq a_0$ e o hiperplano $ax = a_0$ associado. Uma desigualdade $ax \geq a_0$ é *válida* se $Q \subset \{x : ax \geq a_0\}$. Uma desigualdade válida $ax \geq a_0$ é *minimal* se $Q \cap \{x : ax = a_0\} \neq \emptyset$ e é *própria* se $Q \cap \{x : ax > a_0\} \neq \emptyset$. Se uma desigualdade válida $ax \geq a_0$ é minimal e própria então $H = \{x : ax = a_0\}$ é um *hiperplano suporte* e $F = Q \cap H$ é uma *face* de Q . Se além disto, $\dim_F = \dim_Q - 1$ então F é uma *faceta* de Q . Desta forma, o politopo Q fica caracterizado por um conjunto de desigualdades lineares que sejam minimais, próprias e geradoras de facetas; isto é, o politopo Q é o conjunto intersecção destas desigualdades.

Seja Q_C o politopo do PCV. Na caracterização de Q_C há

necessidade de considerar desigualdades especiais. Por simplicidade, é analisado apenas o PCV definido em uma rede não orientada completa e formulado a seguir

$$\text{Min } \sum_{i,j} c_{ij} x_{ij} \quad (1)$$

$$\text{Suj } \sum_j x_{ij} = 2 \quad i \in N \quad (2)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \quad S \subset N \quad (3)$$

$$x_{ij} = 0,1 \quad (i;j) \in A \quad (4)$$

Observe que x_{ij} e x_{ji} denotam a mesma variável associada a aresta $(i;j)$. As restrições (2,3) são restrições do 2-em-parelhamento e restrições de eliminação de subrotas (vide Capítulo 3, seção 4).

TEOREMA [GP 1]. Q_C tem dimensão $\dim_{Q_C} = \frac{1}{2} n(n-3)$. Observe que para $n=3$ existe uma única solução para o PCV.

Dentre todas as desigualdades geradoras de facetas do PCV, são conhecidas:

i) restrição de canalização [GP 1]

$$x_{ij} \leq 1; \quad (i;j) \in A \quad \text{gera faceta de } Q_C \text{ para todo } n \geq 4$$

ii) restrição de *não-negatividade* [GP 1]

$x_{ij} \geq 0; (i;j) \in A$ gera faceta de Q_C para todo $n \geq 5$.

iii) restrição de *eliminação de subrotas* [GP 2]

$\sum_{i,j \in S} x_{ij} \leq |S| - 1; S \subset N$ tal que $2 \leq |S| \leq \lfloor n/2 \rfloor$

gera faceta de Q_C para todo $n \geq 4$.

Observe que $|S| = 1$ ou $|S| = n - 1$ não devem ser considerados para esta restrição porque implica em redundância. Observe também que neste caso do PCV simétrico, as restrições provenientes de S são iguais as restrições de $\bar{S} = N \setminus S$.

iv) restrição de *botão* [Ed 2]

Dado $S \subset N$ defina

$\delta(S) = \{(i;j) : i \in S, j \notin S \text{ ou } i \notin S, j \in S\}$

$\gamma(S) = \{(i;j) : i,j \in S\}$

e seja $\delta(i) = \delta(\{i\})$.

Para cada $S \subset N$, $B \subset \delta(S)$ com $|B \cap \delta(i)| \leq 1, i \in S$, e

$$r = \frac{|B| + 2|S| - 1}{2} > 0 \text{ inteiro}$$

$\sum_{i,j \in \gamma(S) \cup B} x_{ij} \leq r$ gera faceta de Q_C para todo $n \geq 6$.

v) restrição de pente ("comb") [GP 2]

Considere um cabo $W_0 \subset N$ e um número ímpar $k \geq 3$ de dentes: $W_1, W_2, \dots, W_k \subset N$ tal que

$$|W_0 \cap W_p| \geq 1 \quad p = 1, \dots, k \quad (5)$$

$$|W_p \setminus W_0| \geq 1 \quad p = 1, \dots, k \quad (6)$$

$$W_p \cap W_q = \emptyset \quad 1 \leq p < q \leq k \quad (7)$$

A restrição de pente associada é expressa por

$$\sum_{p=0}^k \sum_{i \in W_p} \sum_{j \in W_p} x_{ij} \leq |W_0| + \sum_{p=1}^k (|W_p| - 1) - \lceil k/2 \rceil \quad (8)$$

e gera faceta de Q_C para todo $n \geq 6$.

Para ilustrar a necessidade das restrições tipo pente, considere o exemplo [GP 1] da rede (completa) de $n = 8$ nós com o seguinte pente (Fig. 4.1):

$$W_0 = \{1, 2, 3, 4\}$$

$$W_1 = \{1, 2, 5, 6\}$$

$$W_2 = \{3, 7\}$$

$$W_3 = \{4, 8\}$$

a restrição de pente associada é:

$$2x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{23} + x_{24} + x_{25} + x_{26} + x_{34} + x_{37} + x_{48} + x_{56} \leq 7$$

Usando as relações $\sum_{j=2}^8 x_{1j} = 2, x_{12} + \sum_{j=3}^8 x_{2j} = 2$, obtém-se:

OBSERVAÇÕES

- 1) Uma restrição de canalização é uma restrição de eliminação de subrotas para $|S| = 2$.
- 2) Uma restrição de eliminação de subrotas é uma restrição de pente para $|W_0| = 1, k = 1$ e $W_1 = S$.
- 3) Uma restrição de pente é a soma de restrições de eliminação de subrotas para os conjuntos de nós $W_0, W_1, \dots, W_k$, onde o termo $\lceil k/2 \rceil$ provém da estrutura do pente.
- 4) Uma restrição de botão é uma restrição de pente para $|W_0 \cap W_p| = 1$ e $|W_p \setminus W_0| = 1$.

CAPÍTULO 3

MÉTODOS EXATOS

1. PLANOS DE CORTE

Esta técnica é usada para programas lineares inteiros e consiste em aplicar o método simplex em um programa linear obtido a partir do programa inteiro ao relaxar as restrições de integrabilidade. O procedimento termina quando o método simplex detecta inviabilidade, ilimitabilidade, ou otimalidade (inteira). Sempre que é obtida uma solução ótima não-inteira, é introduzido um corte (restrição linear), gerado a partir da solução ótima atual, satisfazendo as seguintes propriedades:

- i) eliminar a solução ótima não-inteira do programa linear atual, e
- ii) não eliminar nenhuma solução inteira viável do programa original.

Este procedimento pode ser descrito como segue:

- 0. Designe por programa atual o programa inteiro relaxado em um programa linear.
- 1. Resolva o programa atual utilizando o método simplex (dual ou primal, o que for apropriado). Três casos podem ocorrer:
 - a) o programa atual não apresenta solução viável. O programa

inteiro original é inviável; termine.

b) a solução ótima do programa atual é inteira. Ela é a solução ótima do programa inteiro original; termine.

c) a solução ótima do programa atual não é inteira. Gere um corte satisfazendo as propriedades i) e ii). Adicione-o ao programa atual; vá para 1.

Dentre as regras de geração de cortes, destaca-se a de Gomory [GN 1], que utiliza apenas adições e subtrações em seu cálculo, e tem convergência finita garantida.

Um esquema que tem sido aplicado para o PCV consiste em utilizar o modelo de designação com restrições adicionais de eliminação de subrotas. São relaxadas as restrições de integrabilidade com as restrições de eliminação de subrotas; as primeiras para obter um programa linear e as últimas devido ao seu grande número. Desta forma, a rota hamiltoniana ótima é obtida somente quando todas estas restrições relaxadas forem satisfeitas. Além disto, os cortes a serem gerados podem ser provenientes das condições de integrabilidade (Cortes de Gomory, por exemplo) ou de eliminação de subrotas. Christofides [Ch 1] afirma que Dantzig, Fulkerson e Johnson [DF 1] foram os primeiros a sugerir este tipo de esquema. Outras variações são apresentadas em [CW1,Cr1,M21].

2. PROGRAMAÇÃO DINÂMICA [DL 1]

Considere $G = (N, A, C)$ uma rede orientada.

DEFINIÇÃO

Uma sequência finita $P = (i_1, \ell_1, i_2, \ell_2, \dots, i_{k-1}, \ell_{k-1}, i_k)$ é uma k -cadeia se $\{i_1, i_2, \dots, i_k\}$ é um conjunto de nós distintos e $\{\ell_1, \ell_2, \dots, \ell_{k-1}\}$ é um conjunto de arcos, onde ℓ_j é um arco unindo os nós $i_j, i_{j+1}, j = 1, \dots, k-1$ ($k \in N$).

A Programação Dinâmica é aplicável a problemas que possam ser expressos através de uma sequência de decisões inter-relacionadas recursivamente. O PCV pode ser modelado como um problema de programação dinâmica, onde a cada decisão é acrescentado um novo arco a uma cadeia que ao final de $n-1$ decisões se torna uma rota hamiltoniana.

2.1. FORMULAÇÃO

Considere o nó 1 como nó inicial e final da rota hamiltoniana. Seja

$$f_k(i, V)$$

o comprimento da k -cadeia mínima de 1 a i passando por todos os nós de V , sendo $|V| = k$ e $V \subset N \setminus \{1, i\}$

A formulação do PCV assimétrico é:

$$\text{Min}_{j \neq 1} \{f_{n-2}(j, N \setminus \{1, j\}) + c_{j1}\}$$

$$\text{Suj } f_k(i, V) = \text{Min}_{j \in V} \{f_{k-1}(j, V \setminus \{j\}) + c_{ji}\}, \quad k = n-2, \dots, 1$$

$$\text{com } f_0(i, \emptyset) = c_{1i}$$

Para o PCV simétrico, associado a uma rede não-orientada. $G = (N, A, C)$, a formulação torna-se mais simples, pois uma cadeia de i a j e a sua cadeia inversa de j a i possuem o mesmo comprimento.

Se n é par

$$\text{Min}_{j \neq 1} \left\{ \text{Min}_{j \in V} \left\{ f_{\frac{n-2}{2}}(j, V) + f_{\frac{n-2}{2}}(j, N \setminus \{1, j\} \setminus V) : V \subset N \setminus \{1, j\} |V| = \frac{n-2}{2} \right\} \right\}$$

$$\text{Suj } f_k(i, V) = \text{Min}_{j \in V} \{f_{k-1}(j, V \setminus \{j\}) + c_{ji}\} \quad k = \frac{n-2}{2}, \dots, 1$$

$$\text{com } f_0(i, \emptyset) = c_{1i}$$

Se n é ímpar

$$\text{Min}_{j \neq 1} \left\{ \text{Min}_{j \in V} \left\{ f_{\frac{n+1-2}{2}}(j, V) + f_{\frac{n-1-2}{2}}(j, N \setminus \{1, j\} \setminus V) : V \subset N \setminus \{1, j\} |V| = \frac{n-1}{2} \right\} \right\}$$

$$\text{Suj } f_k(i, V) = \text{min}_{j \in V} \{f_{k-1}(j, V \setminus \{j\}) + c_{ji}\}, \quad k = \frac{n-1}{2}, \dots, 1$$

$$\text{com } f_0(i, \emptyset) = c_{1i}.$$

2.2. OPERAÇÕES REQUERIDAS

Para cada estágio $k, f_k(i, V)$ deve ser calculado para $(n-1) \binom{n-2}{k}$ diferentes pares i, V . Em cada cálculo k adições e $k-1$ comparações são feitas. Então para todos os estágios o número de operações requeridas é:

CASO ASSIMÉTRICO

$$\text{nº adições: } (n-1) \sum_{k=1}^{n-2} k \binom{n-2}{k}$$

$$\text{nº comparações: } (n-1) \sum_{k=1}^{n-2} (k-1) \binom{n-2}{k}$$

CASO SIMÉTRICO (n par)

$$\text{nº adições: } (n-1) \sum_{k=1}^{(n-2)/2} k \binom{n-2}{k} + (n-1) \frac{1}{2} \binom{n-2}{(n-2)/2}$$

$$\text{nº comparações: } (n-1) \sum_{k=1}^{(n-2)/2} (k-1) \binom{n-2}{k} + (n-1) \frac{1}{2} \binom{n-2}{(n-2)/2} - 1$$

CASO SIMÉTRICO (n ímpar)

$$\text{nº adições: } (n-1) \sum_{k=1}^{(n+1-2)/2} k \binom{n-2}{k} + (n-1) \binom{n-2}{(n+1-2)/2}$$

$$\text{nº comparações: } (n-1) \sum_{k=1}^{(n+1-2)/2} (k-1) \binom{n-2}{k} + (n-1) \binom{n-2}{(n+1-2)/2} - 1$$

3. MÉTODO DO SUBGRADIENTE

Considere um conjunto $U \subset \mathbb{R}^n$

DEFINIÇÕES

U é um conjunto aberto se para todo ponto $x \in U$ existe uma bola centrada em x e contida em U . U é um conjunto fechado se $\mathbb{R}^n \setminus U$ (complementar de U) é um conjunto aberto. U é um conjunto convexo se para cada $\lambda^1, \lambda^2 \in U$, e para cada $\alpha \in [0, 1]$, $\alpha \lambda^1 + (1 - \alpha) \lambda^2 \in U$, isto é, se toda combinação convexa de dois pontos quaisquer do conjunto pertence ao conjunto também.

Uma função $w : U \rightarrow \mathbb{R}$ é côncava se, para todo $\lambda^1, \lambda^2 \in U$ e para todo $\alpha \in [0, 1]$,

$$w(\alpha \lambda^1 + (1 - \alpha) \lambda^2) \geq \alpha w(\lambda^1) + (1 - \alpha) w(\lambda^2)$$

Um vetor $\gamma^0 \in \mathbb{R}^n$ é um subgradiente da função côncava $w : U \subset \mathbb{R}^n \rightarrow \mathbb{R}$ no ponto $\lambda^0 \in U$, se

$$w(\lambda) \leq w(\lambda^0) + \gamma^0(\lambda - \lambda^0) \quad \text{para todo } \lambda \in U$$

Seja $\partial w(\lambda^0)$ o conjunto de todos os subgradientes de w em λ^0 .

Seja P_U um operador de projeção de $\mathbb{R}^n$ sobre um conjunto fechado e convexo U , isto é, $P_U : \mathbb{R}^n \rightarrow U$ e $\|P_U(\lambda) - \lambda^0\| \leq \|\lambda - \lambda^0\|$

para todo $\lambda \in U$ e $\lambda^0 \in \mathbb{R}^n$ onde $\|\cdot\|$ denota a norma euclidiana.

Considere o seguinte problema

$$\text{Max } \{w(\lambda) : \lambda \in U\}$$

onde $w : U \longrightarrow \mathbb{R}^n$ é uma função côncava, U é um conjunto convexo e fechado, $\lambda \in U$ é uma solução. Suponha que existe uma solução ótima $\lambda^* \in U$ tal que $w(\lambda^*) \geq w(\lambda)$, $\forall \lambda \in U$.

O método de otimização com o subgradiente é o seguinte.

0. Seja $\lambda^1 \in U$ uma solução arbitrária; faça $t \longleftarrow 1$
1. Determine um subgradiente $\gamma^t \in \partial w(\lambda^t)$ e um tamanho de passo $\rho_t \in \mathbb{R}$. Faça $\lambda^{t+1} \longleftarrow P_U(\lambda^t + \rho_t \gamma^t)$, $t \longleftarrow t+1$. Vá para 1.

Um critério de parada conveniente é $\|\gamma^t\| < \varepsilon$ onde ε é um número pré-fixado.

Na aplicação do método para o PCV, é comum utilizar os seguintes operadores de projeção:

- a) matriz identidade $P_U = I$, no caso $U = \mathbb{R}^n$
- b) $\lambda = (\lambda_j) = P_U(\lambda^1 = (\lambda_j^1))$ onde $\lambda_j = \max\{0, \lambda_j^1\}$ associado às restrições $\lambda \geq 0$.

Poljak [Pl 1] mostra que as condições

$$\rho_t \longrightarrow 0 \text{ e } \sum_{t=0}^{\infty} \rho_t = \infty$$

são suficientes para garantir que $\lambda^1, \lambda^2, \dots$ convergem para λ^* , embora os valores $w(\lambda^1), w(\lambda^2) \dots$ não sejam monotônicos.

Experiências mostram que a taxa de convergência deste método é muito pequena. Para superar esta dificuldade Poljak [Pl 1] propôs a seguinte regra: em cada estágio t , selecionar

$$\rho_t = \beta_t \left(\frac{w^* - w(\lambda^t)}{\|y^t\|^2} \right) \quad \beta_t \in (\epsilon_1, 2 - \epsilon_2)$$

onde $\epsilon_1, \epsilon_2 > 0$ são arbitrariamente escolhidos de tal forma que $(\epsilon_1, 2 - \epsilon_2)$ seja não vazio.

Esta regra apresenta o inconveniente de que w^* deve ser pré-fixado. Poljak [Pl 2] sugere que w^* seja substituído por $\underline{w} < w^*$ (o valor de uma solução primal viável já conhecida). Held, Wolf e Crowder [HW 1] sugerem que w^* seja substituído por $\bar{w} > w^*$ (uma solução dual viável já conhecida). Em ambos os casos, não há garantia que $w(\lambda^t)$ convirja para w^* . Bazaraa [BS 1] sugere uma regra de escolha do tamanho do passo ρ_t de modo a acelerar a convergência.

O método de otimização com subgradiente é aplicado com frequência na obtenção de limitantes para o método particionar e limitar (seção 4) utilizando um número pré-fixado de iterações.

As aplicações do método do subgradiente no PCV podem ser classificadas nos seguintes modelos:

1) Considere o problema

$$\{\text{Min } cx : a^t x \geq a_0^t, t = 1, 2, \dots, \bar{t}, x \in X\}$$

onde c é um vetor, a^t são vetores, a_0^t são números, e X é um conjunto de vetores binários, por exemplo, o conjunto de soluções viáveis do problema de designação (vide seção 4). A relaxação langrangeana das restrições "difíceis" " $a^t x \geq a_0^t$ " conduz ao problema

$$\text{Max}_{\lambda \geq 0} w^+(\lambda) = \min \{w(x, \lambda) = cx - \lambda(a^t x - a_0^t) : x \in X\}$$

onde λ é um vetor de multiplicadores de Langrange.

Deseja-se obter um valor limitante inferior para o valor da função objetivo do problema original com um pequeno esforço computacional. Desta forma são executadas apenas algumas iterações do método do subgradiente. Levando-se em consideração que os valores de $w^+(\lambda)$ convergem de modo não necessariamente monótono, considera-se como limitante o maior valor de $w^+(\lambda)$ obtido nestas iterações. Para provar que $w^+(\lambda)$ é um limitante para $\lambda \geq 0$, sejam x^* uma solução ótima do problema original e x^+ uma solução ótima do problema relaxado. Desta forma,

$$w^+(\lambda) = cx^+ - \lambda(a^t x^+ - a_0^t)$$

$$w^+(\lambda) \leq cx - \lambda(a^t x - a_0^t) \quad \text{para } x \in X$$

$$w^+(\lambda) \leq cx^* - \lambda(a^t x^* - a_0^t) \quad \text{pois } x^* \in X$$

$$w^+(\lambda) \leq cx^* \quad \text{pois } \lambda \geq 0 \text{ e } (a^t x^* - a_0^t) \geq 0$$

ii) Considere o problema

$$\min \{cx : a^t x = a_0^t, t = 1, 2, \dots, \bar{t}, x \in X\}$$

onde X é um conjunto de vetores binários, por exemplo, o conjunto de soluções viáveis do problema da 1-árvore mínima (vide seção 4).

A restrição lagrangeana das restrições "difíceis" " $a^t x = a_0^t$ " conduz ao problema:

$$\text{Max } w^+(\lambda) = \min\{w(x, \lambda) = cx - \lambda(a^t x - a_0^t) : x \in X\}$$

Para provar que $w^+(\lambda)$ é um limitante para qualquer λ , sejam x^* uma solução ótima do problema original e x^+ uma solução ótima do problema relaxado.

Desta forma,

$$w^+(\lambda) = cx^+ - \lambda(a^t x^+ - a_0^t)$$

$$w^+(\lambda) \leq cx - \lambda(a^t x - a_0^t) \quad \text{para } x \in X$$

$$w^+(\lambda) \leq cx^* - \lambda(a^t x^* - a_0^t) \quad \text{pois } x^* \in X$$

$$w^+(\lambda) \leq cx^* \quad \text{pois } a^t x^* - a_0^t = 0$$

4. PARTICIONAR E LIMITAR

O método particionar e limitar ("branch and bound") consiste em uma enumeração parcial das soluções viáveis e da eliminação das soluções não-ótimas com o uso de valores limitantes da função objetivo. Como o próprio nome indica, este método possui duas rotinas básicas.

- **PARTICIONAMENTO:** é o processo de particionar o conjunto de soluções viáveis de um problema em dois ou mais subconjuntos que são associados a subproblemas. Estes subproblemas por sua vez podem vir a ser particionados também.

- **LIMITAÇÃO:** é o processo de calcular um limitante (inferior no caso do PCV) do valor da solução ótima de um subproblema. Em geral, estes limitantes são obtidos relaxando algumas restrições "difíceis" do subproblema e que muitas vezes são incorporadas na função objetivo com o uso de multiplicadores de Lagrange.

É prática comum durante a execução do método manter uma lista de subproblemas a resolver, assim como a melhor solução viável do problema original; esta será a solução ótima ao final. O

método termina quando a lista de subproblemas se torna vazia. Os subproblemas da lista são resolvidos um após o outro e a resolução de um subproblema termina em um dos casos abaixo:

- i) subproblema relaxado é inviável
- ii) solução ótima do subproblema relaxado é inviável para o problema original - se o valor desta solução ótima (limitante do subproblema) é menor do que o valor da melhor solução viável, o subproblema é particionado e os subproblemas gerados são incluídos na lista de subproblemas.
- iii) solução ótima do subproblema relaxado é viável para o problema original - se o valor desta solução ótima (limitante do subproblema) é menor do que o valor da melhor solução viável, esta é atualizada.

Dentre as regras mais utilizadas para a escolha do próximo subproblema da lista a ser resolvido, destacam-se:

- a) LIFO (seleciona o último subproblema gerado). Resulta em uma armazenagem eficiente, apresentando menor número de subproblemas candidatos na lista. Além disto, as soluções viáveis (não necessariamente ótimas) são alcançadas mais rapidamente.
- b) MENOR LIMITANTE INFERIOR. Tende a reduzir o número de

subproblemas a serem resolvidos, entretanto, apresenta dificulda
des de armazenagem.

De uma maneira geral, as partições apresentadas na literatura
para resolver o PCV associam a cada subproblema k , ou um con-
junto de linhas proibidas D_k^0 , ou um conjunto de linhas requeri-
das D_k^1 , ou ambos. A cada subproblema a ser particionado é deter-
minado um subconjunto $L_k = \{\ell_1, \ell_2, \dots, \ell_m\}$ de linhas livres (não
proibidas e não requeridas) ou um subconjunto $S_k = \{i_1, i_2, \dots, i_m\}$
de nós, a partir do qual são determinados os novos conjuntos de
linhas requeridas e/ou proibidas dos subproblemas gerados. Se os
subproblemas gerados são $p+1, p+2, \dots, p+m$, em geral são utili-
zados os seguintes esquemas de partição:

Partição orientada para linha - onde $L_k = \{\ell\}$ é um conjun-
to de uma única linha.

$$\begin{aligned} D_{p+1}^0 &= D_k^0 \cup \{\ell\} & D_{p+1}^1 &= D_k^1 \\ D_{p+2}^0 &= D_k^0 & D_{p+2}^1 &= D_k^1 \cup \{\ell\} \end{aligned}$$

Frequentemente um dos dois subproblemas gerados apresenta
o mesmo limitante do subproblema que o gerou.

Partição orientada para ciclo/circuito - onde L_k é um
conjunto de linhas formando uma subrota que é um ciclo ou circui-
to (válida apenas para o PCV restrito a rotas hamiltonianas).

$$D_{p+1}^0 = D_k^0 \cup \{\ell_1\}$$

$$D_{p+1}^1 = D_k^1$$

$$D_{p+2}^0 = D_k^0 \cup \{\ell_2\}$$

$$D_{p+2}^1 = D_k^1 \cup \{\ell_1\}$$

$$\vdots$$

$$\vdots$$

$$D_{p+m}^0 = D_k^0 \cup \{\ell_m\}$$

$$D_{p+m}^1 = D_k^1 \cup \{\ell_1, \ell_2, \dots, \ell_{m-1}\}$$

Partição orientada para corte - onde L_k é um conjunto de linhas formando um corte gerado a partir dos nós de uma subrota

$$D_{p+1}^0 = D_k^0$$

$$D_{p+1}^1 = D_k^1 \cup \{\ell_1\}$$

$$D_{p+2}^0 = D_k^0 \cup \{\ell_1\}$$

$$D_{p+2}^1 = D_k^1 \cup \{\ell_2\}$$

$$\vdots$$

$$\vdots$$

$$D_{p+m}^0 = D_k^0 \cup \{\ell_1, \ell_2, \dots, \ell_{m-1}\}$$

$$D_{p+m}^1 = D_k^1 \cup \{\ell_m\}$$

Partição orientada para nós - onde S_k é um conjunto de nós de uma subrota

$$D_{p+1}^0 = D_k^0 \cup \{(i_1, j) : j \in S_k\}$$

$$D_{p+2}^0 = D_k^0 \cup \{(i_1, j) : j \notin S_k\} \cup \{(i_2, j) : j \in S_k\}$$

$$\vdots$$

$$D_{p+m}^0 = D_k^0 \cup \{(i_1, j) : j \notin S_k\} \cup \dots \cup \{(i_{m-1}, j) : j \notin S_k\} \cup \{(i_m, j) : j \in S_k\}$$

tende a produzir um número menor de subproblemas a cada geração.

Com o objetivo de reduzir o número de subproblemas criados a cada geração, é comum escolher dentre os possíveis subconjuntos S_k ou L_k do subproblema k , o de menor cardinalidade. É possível aprimorar este procedimento para L_k considerando aquele subconjunto com menor número de linhas livres [CT2].

Em geral, elementos adicionais em D_k^0, D_k^1 são incluídos de acordo com as regras abaixo. As regras i), iv), v) aplicam-se apenas para o PCV restrito a rotas hamiltonianas.

- i) se o arco $(i,j) \in D_k^1$ então todos os arcos livres com cauda em i , e todos os arcos livres com cabeça em j são incluídos em D_k^0 , inclusive o arco (j,i) , se existir.
- ii) se (i,j) é o único arco livre com cabeça em j e não há nenhum arco requerido com cabeça em j então o arco (i,j) é incluído em D_k^1 .
- iii) se (i,j) é o único arco livre com cauda em i e não há nenhum arco requerido com cauda em i então o arco (i,j) é incluído em D_k^1 .
- iv) se o arco $(i,j) \in D_k^1$ então o arco (j,i) , se existir, é incluído em D_k^0 .
- v) se os arcos $(i_0,i_1), (i_1,i_2), \dots, (i_{t-1},i_t) \in D_k^1$ então o

arco (i_t, i_0) , se existir, é incluído em D_k^0 , exceto se (i_0, i_1) , $(i_1, i_2), \dots, (i_{t-1}, i_t), (i_t, i_0)$ formar uma rota do caixeiro.

Difícilmente a regra v) é integralmente aplicada. Além disso, observe que as regras acima são aplicáveis para o PCV assimétrico, e que podem facilmente ser adaptadas para o PCV simétrico.

Com relação aos esquemas de limitação empregados para o PCV pode-se afirmar que o simples relaxamento de restrições "difíceis" não conduz a bons limitantes. O método de particionar e limitar para ser eficiente requer a escolha de bons limitantes. A literatura apresenta muitas aplicações do método de otimização com o subgradiente na obtenção de limitantes (vide seção 3).

Deseja-se obter valores limitantes com pequeno esforço computacional e desta forma são executadas apenas algumas iterações do método do subgradiente para cada subproblema gerado. Além disso, os multiplicadores obtidos na melhor solução de um subproblema são usados como multiplicadores iniciais nos subproblemas gerados. Ao final de algumas iterações do método, um dos casos abaixo ocorre:

- i) subproblema relaxado é inviável
- ii) melhor solução do subproblema relaxado tem valor maior ou igual ao da melhor solução viável.
- iii) melhor solução do subproblema relaxado tem valor menor do que o da melhor solução viável e é uma solução inviável para o

problema original ou é uma solução viável que não satisfaz as condições de folga complementar - o subproblema é particionado e os subproblemas gerados são introduzidos na lista de subproblemas a resolver.

iv) melhor solução do subproblema relaxado tem valor menor do que o da melhor solução viável e é uma solução viável para o problema original que satisfaz as condições de folga complementar - trata-se de uma solução ótima para o problema original e o método termina.

A literatura apresenta diversos problemas clássicos de otimização aplicados em esquemas de limitação para o PCV, os quais são citados abaixo. A matriz de variáveis $X = (x_{ij})$ sempre é interpretada da seguinte forma: $x_{ij} = 1$ o arco (i,j) está na solução e $x_{ij} = 0$ caso contrário.

4.1. DESIGNAÇÃO

Considere a seguinte formulação para o PCV assimétrico (restrito a rotas hamiltonianas)

$$\text{Min} \quad \sum_{i,j} c_{ij} x_{ij}$$

$$\text{sujeito a} \quad \sum_j x_{ij} = 1 \quad i \in N$$

$$\sum_j x_{ji} = 1 \quad i \in N$$

$$x_{ij} = 0,1 \quad (i,j) \in A$$

$$X \in T$$

onde T denota o conjunto das designações que satisfazem as restrições de eliminação de subrotas.

Observe que a simples relaxação da restrição " $X \in T$ " na formulação acima produz o problema da designação. Ou seja, toda rota hamiltoniana é uma designação.

As restrições de eliminação de subrotas podem ser expressas de várias formas:

$$i) \quad \sum_{(i,j) \in K_t} x_{ij} \geq 1$$

onde $K_t = (S_t, \bar{S}_t)$, $\emptyset \neq S_t \subsetneq N$, $\bar{S}_t = N \setminus S_t$, $t \in T \subseteq N$

$$ii) \quad \sum_{i,j \in S_t} x_{ij} \leq |S_t| - 1 \quad \emptyset \neq S_t \subsetneq N, t \in T \subseteq N$$

$$iii) \quad \sum_{(i,j) \in K'_t \cup K''_t} x_{ij} \geq 1$$

onde $K'_t = (S_t, \bar{S}_t \setminus \{k\})$, $K''_t = (\bar{S}_t \setminus \{k\}, S_t)$; $k \in N$, $\emptyset \neq S_t \subsetneq N \setminus \{k\}$, $t \in T \subseteq N$

$$\text{iv)} \quad \sum_{(i,j) \in \varphi} x_{ij} \leq |S_t| - 1, \quad \varphi \in \Phi(S_t), \quad \varphi \neq S_t \subset N, t \in T \subseteq N$$

onde $\Phi(S_t)$ é a família de todos os circuitos formados com os nós de S_t .

$$\text{v)} \quad u_i - u_j + nx_{ij} \leq n - 1; \quad 1 \leq i, j \leq n - 1, \quad i \neq j.$$

As restrições do tipo iv) não são usadas em geral, pois são equivalentes às restrições dos tipos i) e ii). Embora, as restrições do tipo v) apresentem-se em número polinomial, não são de interesse por admitir mais soluções fracionárias quando comparadas com as demais restrições.

Murty [Mu 2] apresenta um esquema de partição orientada para linha. Belmore e Malone [BM 1] não apresentam um esquema de partição propriamente, pois uma mesma solução viável pode aparecer em dois ou mais subproblemas do mesmo nível. Garfinkel [Ga 2] utiliza uma partição orientada para nós. Carpaneto e Toth [CT 1] utilizam uma partição orientada para circuito. Carpaneto e Toth [CT 2] consideram um esquema de partição orientada para circuito onde o conjunto L_k escolhido possui o menor número de arcos livres e tais arcos são ordenados antes de ser feita a geração dos subproblemas. Outras partições mais sofisticadas encontram-se em Balas e Christofides [BC 1] que utilizam limitantes inferiores e superiores simultaneamente.

Vale salientar que o problema de designação pode ser usado

como limitante para o PCV simétrico, entretanto, apresenta o inconveniente de ter um número elevado de subrotas de cardinalidade 2.

4.2. 2-EMPARELHAMENTO

Considere a formulação abaixo para o PCV simétrico (restrito a rotas hamiltonianas).

$$\text{Min} \quad \sum_{i,j} c_{ij} x_{ij}$$

$$\text{sujeito a} \quad \sum_j x_{ij} = 2 \quad i \in N$$

$$x_{ji} = 0,1 \quad (i,j) \in A$$

$$X \in T$$

onde x_{ij} e x_{ji} denotam a mesma variável, e T denota o conjunto dos 2-emparelhamentos que satisfazem as restrições de eliminação de subrotas. Observe que a simples relaxação da restrição " $X \in T$ " na formulação acima produz o problema do 2-emparelhamento. Ou seja, toda rota hamiltoniana é um 2-emparelhamento.

As restrições de eliminação de subrotas apresentadas para o problema de designação podem ser adaptadas para o problema do 2-emparelhamento. Este limitante não é tão eficiente quanto ao

limitante da 1-árvore por motivos computacionais (vide seção 4.3).

4.3. 1-ÁRVORE

Seja $G = (N, A, C)$ uma rede não-orientada.

DEFINIÇÕES

Uma *árvore* de G é uma subrede conectada sem ciclos; portanto com $n - 1$ arestas.

Uma *1-árvore* (ou quase árvore) de G é uma subrede conectada com exatamente um ciclo; portanto com n arestas.

O PCV simétrico restrito a rotas hamiltonianas pode ser interpretado como o problema de determinar uma 1-árvore mínima que possui grau 2 em cada nó, pois uma rota hamiltoniana é uma 1-árvore cujos nós possuem grau 2.

Seja $A_V = \{1\text{-árvores de } G\}$. O PCV simétrico (restrito a rotas hamiltonianas) é

$$\text{Min } \sum_{i,j} c_{ij} x_{ij}$$

$$\text{sujeito a } x \in A_V$$

$$x \in T$$

onde T denota o conjunto das 1-árvores que satisfazem as res-

trições de eliminação de subrotas. Observe que a relaxação de " $X \in T$ " produz o problema da 1-árvore mínima.

As restrições " $X \in T$ " podem ser expressas por:

$$i) \quad \sum_{(i;j) \in \delta(i)} x_{ij} = 2, \quad i \in N$$

onde $\delta(i) = \{(i;j) \in A : j \in N\}$

$$ii) \quad \sum_{i,j \in S_t} x_{ij} \leq |S_t| - 1, \quad \emptyset \neq S_t \subsetneq N, \quad t \in T \subsetneq N$$

$$iii) \quad \sum_{(i;j) \in K_t} x_{ij} \geq 2$$

onde $K_t = (S_t; \bar{S}_t)$, $\emptyset \neq S_t \subsetneq N$, $t \in T \subseteq N$

$$iv) \quad \sum_{(i;j) \in K'_t} x_{ij} \geq 1$$

onde $K'_t = (S_t; \bar{S}_t \setminus \{k\})$, $\emptyset \neq S_t \subsetneq N$, $k \in N$, $t \in T \subseteq N$

observe que x_{ij} e x_{ji} denotam a mesma variável.

Held e Karp [HK 1] apresentam um esquema de partição orientada para linha. Held e Karp [HK 2] consideram um esquema de partição orientada para ciclo; para cada subproblema, as arestas livres do ciclo selecionado são ordenadas de acordo com a variação do limitante ao excluir a aresta.

Volgenant e Jonker [VJ 1] adotam um esquema de partição orientada para ciclo de três nós. Dado um nó com grau maior do que 2 na melhor 1-árvore do subproblema, o ciclo escolhido contém

este nó e outros dois nós adjacentes a este na 1-árvore.

4.4. 1-ARBORESCÊNCIA

Seja $G = (N, A, C)$ uma rede orientada.

DEFINIÇÕES

Uma *arborescência* é uma subrede conectada com $n-1$ arcos com cabeças distintas. Isto é, nenhum nó é a cabeça de mais de um arco, não há nenhum circuito e exatamente um dos nós não é cabeça de nenhum arco.

Uma *1-arborescência* é uma subrede conectada com n arcos com cabeças distintas. Isto é, nenhum nó é a cabeça de mais de um arco e há exatamente um circuito.

O PCV assimétrico restrito a rotas hamiltonianas pode ser interpretado como o problema de determinar uma 1-arborescência mínima que possui grau interno $(gr^-(.)) =$ grau externo $(gr^+(.)) = 1$ em cada nó; pois uma rota hamiltoniana é uma 1-arborescência cujos nós possuem $gr^-(.) = gr^+(.) = 1$.

Seja $A_b = \{1\text{-arborescências de } G\}$. O PCV assimétrico (restrito a rotas hamiltonianas) é

$$\text{Min } \sum_{i,j} c_{ij} x_{ij}$$

$$\text{sujeito a } x \in A_b$$

$$x \in T$$

onde T denota o conjunto das 1-arborescências que são rotas hamiltonianas. Observe que a relaxação de " $X \in T$ " produz o problema da 1-arborescência mínima. Entretanto, este limite não deve ser tão eficiente quanto ao problema da designação por motivos computacionais.

As restrições " $X \in T$ " podem ser expressas por

$$\sum_{(i,j) \in \delta^+(i)} x_{ij} = 1 \quad i \in N$$

$$\sum_{(j,i) \in \delta^-(i)} x_{ji} = 1 \quad i \in N$$

onde $\delta^+(i) = \{(i,j) \in A : j \in N\}$

$\delta^-(i) = \{(j,i) \in A : j \in N\}$

As demais restrições apresentadas na seção anterior podem ser adaptadas para este caso.

4.5. n-ROTA

A modelagem do PCV(restrito a rotas hamiltonianas) como uma Programação Dinâmica e o algoritmo não polinomial correspondente foram apresentados na seção 2. Ao desconsiderar o conjunto V (conjunto dos nós já visitados) naquela formulação, obtém-se o problema da n-rota. Este problema consiste em determinar uma subrota

de comprimento mínimo que visita n nós não necessariamente distintos; uma rota hamiltoniana é uma n -rota que visita n nós distintos.

Considere o nó 1 como nó inicial e final da rota hamiltoniana. Seja

$$f_k(i)$$

o comprimento da k -cadeia mínima de 1 a i .

Seja $G = (N, A, C)$ uma rede orientada, a formulação do problema da n -rota é

$$\begin{array}{ll} \text{Min} & \{f_{n-2}(j) + c_{j1}\} \\ \text{su}j & j \neq 1 \end{array}$$

$$\text{su}j \quad f_k(i) = \text{Min}_{j \neq 1, i} \{f_{k-1}(j) + c_{ji}\}, k = n-2, \dots, 1$$

com $f_0(i, \emptyset) = c_{1i}$.

Desta forma, o PCV assimétrico restrito a rotas hamiltonianas é

$$\begin{array}{ll} \text{Min} & \sum_{i,j} c_{ij} x_{ij} \\ \text{su}j & x \in R_n \end{array}$$

$$\sum_j x_{ij} = 1 \quad i \in N$$

$$\sum_j x_{ji} = 1 \quad i \in N$$

onde $R_n = \{n\text{-rotas de } G\}$.

Seja $G = (N, A, C)$ uma rede não-orientada. Neste caso, a cadeia de i a j e a sua cadeia inversa de j a i possuem o mesmo comprimento.

Se n é par

$$\text{Min}_{j \neq 1} \{f_{(\frac{n-2}{2})(j)} \times 2\}$$

$$\text{su}j \quad f_k(i) = \text{Min}_{j \neq 1, i} \{f_{k-1}(j) + c_{ji}\}, \quad k = n-2, \dots, 1$$

com $f_0(i, \emptyset) = c_{1i}$.

Se n é ímpar

$$\text{Min} \{f_{(\frac{n-3}{2})(j)} + f_{(\frac{n-1}{2})(j)}\}$$

$$\text{su}j \quad f_k(i) = \text{Min}_{j \neq 1, i} \{f_{k-1}(j) + c_{ji}\}, \quad k = n-2, \dots, 1$$

com $f_0(i, \emptyset) = c_{1i}$.

Desta forma, o PCV simétrico (restrito a rotas hamiltonianas)

é

$$\text{Min} \quad \sum_{i,j} c_{ij} x_{ij}$$

$$\text{su}j \quad x \in R_n$$

$$\sum_j x_{ij} = 2 \quad i \in N$$

onde x_{ij} e x_{ji} denotam a mesma variável e $R_n = \{n\text{-rotas de } G\}$

OBSERVAÇÃO

Uma pequena modificação no algoritmo permite eliminar um tipo de repetição de nó na solução. Acrescente um vetor $q_k(i)$ que armazena o penúltimo nó da k -cadeia de 1 a i , o qual deve ser proibido para a variável de controle j na determinação de $f_k(i)$; ou seja,

$$f_k(i) = f_{k-1}(j^*) + c_{j^*i} = \min_{j \neq 1, i} \{f_{k-1}(j) + c_{ji} : q_{k-1}(j) \neq i\}; k = n-2, \dots, 1$$

$$q_k(i) = j^*$$

onde $q_1(i) = 1$.

4.6. COBERTURA SIMÉTRICA

Considere a seguinte formulação para o PCV assimétrico (não restrito a rotas hamiltonianas)

$$\begin{aligned} \text{Min} \quad & \sum_{i,j} c_{ij} x_{ij} \\ \text{sujeito a} \quad & \sum_j x_{ij} \geq 1 \quad i \in N \\ & \sum_j x_{ji} \geq 1 \quad i \in N \\ & x \in T \end{aligned}$$

onde T denota o conjunto das coberturas simétricas que satisfazem as restrições de eliminação de subrotas. Observe que a simples relaxação da restrição " $X \in T$ " na formulação produz o problema da cobertura simétrica.

As restrições de eliminação de subrotas podem ser expressas por

$$i) \quad \sum_{(i,j) \in K_t} x_{ij} \geq 1$$

onde $K_t = (S_t, \bar{S}_t)$, $\emptyset \neq S_t \subsetneq N$, $t \in T \subseteq N$

$$ii) \quad \sum_{(i,j) \in K'_t \cup K''_t} x_{ij} \geq 1$$

onde $K'_t = (S_t, \bar{S}_t \setminus \{k\})$, $K''_t = (\bar{S}_t \setminus \{k\}, S_t)$; $k \in N$

$$\emptyset \neq S_t \subsetneq N \setminus \{k\}, t \in T \subseteq N.$$

No Capítulo 7 é desenvolvido um algoritmo para resolver o PCV (relaxado no problema da cobertura simétrica) e são apresentados experimentos computacionais de um algoritmo do tipo particionar e limitar que usa este limitante.

4.7. 2-COBERTURA

Considere a seguinte formulação para o PCV simétrico:
(não restrito a rotas hamiltonianas)

$$\text{Min} \quad \sum_{i,j} c_{ij} x_{ij}$$

$$\text{subj } \sum_j x_{ij} \geq 2 \quad i \in N$$

$$x_{ij} = 0,1 \quad (i;j) \in A$$

$$X \in T$$

onde x_{ij} e x_{ji} denotam a mesma variável, e T denota o conjunto das 2-coberturas que satisfazem as restrições de eliminação de subrotas. Observe que a simples relaxação da restrição " $X \in T$ " na formulação acima produz o problema da 2-cobertura.

As restrições de eliminação de subrotas apresentadas para o problema da cobertura simétrica podem ser adaptadas para o problema de 2-cobertura.

5. CADEIA MÁXIMA [HN 1]

O PCV pode ser transformado em tempo polinomial no problema da Cadeia Máxima através da duplicação de um dos nós. Embora não haja interesse prático nesta transformação, pois não se conhece nenhum algoritmo eficiente para resolver o problema da cadeia máxima, este artifício apresenta interesse teórico no sentido em que rotas hamiltonianas são transformadas em soluções básicas viáveis de um programa linear.

Seja $G = (N, A, C)$ uma rede orientada hamiltoniana com $N = \{1, 2, \dots, n\}$. Em casos de redes não-orientadas hamiltonianas, substitua cada aresta $(i;j)$ pelos arcos $(i,j), (j,i)$.

A partir de G construa a rede $G' = (N', A', C')$ com

$$N' = \{1, 2, \dots, n, 1'\}$$

$$A' = \{(i, j) \in A : j \neq 1\} \cup \{(i, 1') : (i, 1) \in A\}$$

$$c'_{ij} = \begin{cases} M - c_{ij} & \text{se } i \neq j, j \neq 1' \\ M - c_{i1} & \text{se } i \neq j, j = 1' \end{cases} \quad (i, j) \in A$$

onde $M > n \times \max \{|c_{ij}|\}$.

PROPRIEDADES:

- Qualquer cadeia máxima em G' contém n arcos.
- Existe uma correspondência biunívoca entre as rotas hamiltonianas de G e as cadeias de G' entre $1, 1'$ com n arcos.
- Existe uma correspondência biunívoca entre as subrotas de G que contêm o nó 1 e as cadeias de G' entre $1, 1'$ com menos de n arcos.
- Existe uma correspondência biunívoca entre as subrotas de G que não contêm o nó 1 e os circuitos de G' .
- Uma cadeia em G' é uma cadeia máxima se e só se a rota hamiltoniana em G correspondente é ótima.
- Se na rede $G = (N, A, C)$ toda a rota hamiltoniana ou subrota contém o nó 1 , então a cadeia máxima em G' pode ser obtida eficientemente através do algoritmo da cadeia máxima, (PERT-CPM) [Ma 1], pois a rede G' não apresenta circuitos nestas circuns-

tâncias.

Considere o programa linear abaixo:

$$\begin{aligned}
 &\text{Max} \quad \sum_{i,j} c'_{ij} x_{ij} \\
 &\text{suj} \quad \sum_j x_{ij} - \sum_j x_{ji} = \begin{cases} -1 & \text{se } i = l' \\ 0 & \text{se } i \neq l, l' \\ 1 & \text{se } i = l \end{cases} \\
 &\quad x_{ij} \geq 0 \quad (i,j) \in A
 \end{aligned}$$

PROPRIEDADES

- cada base deste problema constitui uma árvore geradora.
- as soluções básicas viáveis possuem valores 0,1 sendo que as componentes com valor 1 formam uma cadeia entre l, l' .
- as soluções homogêneas extremas correspondem aos circuitos de G' e caso existam o problema é ilimitado.

Desta forma, o programa linear acima é viável se e somente se existe uma cadeia entre l, l' , e se, além disto, existe um circuito em G' então o problema é ilimitado, pois $C' = (c'_{ij}) \geq 0$

Assim, uma cadeia máxima em G' corresponde à melhor solução básica viável do programa linear acima, e a sua determinação constitui em um problema bem definido.

CAPÍTULO 4

MÉTODOS NÃO EXATOS (ou heurísticos)

Devido ao desconhecimento de algoritmos exatos polinomiais para resolver o PCV, é comum a utilização de métodos heurísticos polinomiais para obter soluções quase ótimas. É conveniente que o estudo de um algoritmo heurístico inclua a determinação do valor máximo da taxa V^+/V^* onde V^+ é o comprimento da rota hamiltoniana determinada pela heurística e V^* é o comprimento da rota hamiltoniana ótima. Para o estudo das heurísticas apresentadas neste Capítulo, considere a rede $G = (N, A, C)$ não-orientada, completa e suponha que C satisfaça a desigualdade triangular (condição necessária para que a rota do caixeiro passe exatamente uma vez em cada nó).

Estimativas probabilísticas do valor máximo da taxa V^+/V^* necessitam uma análise técnica profunda, além de uma apropriada distribuição de probabilidade dos c_{ij} . Entretanto, um estudo de duas heurísticas particulares pode ser visto em Karp [Kal, Ka2]. Neste Capítulo são apresentadas as heurísticas mais conhecidas.

1. CIDADE NÃO - VISITADA MAIS PRÓXIMA

A partir de uma sequência de cidades $P = (i_1, i_2, \dots, i_k)$ formando um caminho, inclua a aresta $(i_k; i_{k+1})$, $k < n$ ou a aresta $(i_k; i_1)$, $k = n$, tal que:

$$c_{i_k, i_{k+1}} = \min_{i_j} \{c_{i_k, i_j} : i_j \notin P\}, \quad k < n$$

$$c_{i_k, i_1} = c_{i_k, i_1}, \quad k = n$$

Rosencrantz, Stearns e Lewis [RS 1, RS 2] mostram que

$$\frac{V^+}{V^*} \leq \frac{1}{2} (\lceil \log_2 n \rceil + 1)$$

Para $p = \lfloor n/2 \rfloor > 3$ existe um problema com $2^p - 1$ nós, para o qual é possível obter [RS 2]:

$$\frac{V^+}{V^*} \geq \frac{1}{3} (\log_2(n+1) + \frac{4}{3})$$

2. INSERÇÃO DA CIDADE MAIS PRÓXIMA

A partir de uma subrota S arbitrária, substitua sequencialmente uma aresta $(y; z)$ de S por duas arestas $(y; x)$ e $(x; z)$ com $x \notin S$ até obter uma rota hamiltoniana. São conhecidas três variantes:

(i) $(y; z) \in S$ é pré-fixada, seja:

$$c_{xy} + c_{xz} - c_{yz} = \min_i \{c_{iy} + c_{iz} - c_{yz} : i \notin S\}$$

(ii) $x \notin S$ é pré-fixado, sejam

$$c_{xy} = \min_j \{c_{xj} : j \in S\}$$

$$c_{xy} - c_{yz} = \min_k \{c_{xk} - c_{yk} : k \in S \text{ e adjacente a } y\}.$$

(iii) Seja

$$c_{xy} + c_{xz} - c_{yz} = \min \{c_{ij} + c_{ik} - c_{jk} : i \notin S, (j;k) \in S\}$$

Rosencrants, Stearns e Lewis mostram que:

$$\text{para (i)} \quad \frac{V^+}{V^*} \leq \lceil \log_2(n) \rceil + 1 \quad [\text{RS 2}]$$

$$\text{para (ii) e (iii)} \quad \frac{V^+}{V^*} < 2 \quad [\text{RS 1, RS 2}]$$

Além disso, para (ii), (iii) existem redes com $n \geq 6$, para as quais:

$$\frac{V^+}{V^*} = 2 \left(1 - \frac{1}{n}\right) \quad [\text{RS 1}]$$

3. MÉTODO DE LIN

A partir de uma rota hamiltoniana arbitrária faça uma k -mudança, que consiste em substituir k arestas por outras tantas de forma a obter uma rota hamiltoniana de comprimento inferior.

Pare quando não encontrar arestas que possibilitem esta mudança.

Para $n \geq 8$ e $k \leq n/4$,

$$\frac{V^+}{V^*} \leq 2 \left(1 - \frac{1}{n}\right) \quad [RS \ 2]$$

4. ÁRVORE GERADORA MÍNIMA

A partir de uma árvore geradora mínima A , considere o grafo $G = (N, A)$ com $A = \{(i, j), (j, i) : (i, j) \in A\}$ e seja $L = (i_1, i_2, \dots, i_s, i_1)$ uma rota euleriana em G . Observe que cada nó de G aparece no mínimo duas vezes em L . Para cada nó $i_p = i_q$ com $p < q$, substitua (i_{q-1}, i_q, i_{q+1}) por (i_{q-1}, i_{q+1}) em L , eliminando a segunda passagem em i_q . Desta forma pode ser obtida uma rota hamiltoniana para a qual

$$\frac{V^+}{V^*} \leq 2$$

5. MÉTODO DE CHRISTOFIDES

A partir de uma árvore geradora mínima A , determine o 1 - emparelhamento perfeito mínimo E no subgrafo completo de G formado pelos nós com grau ímpar em A . Seja $L = (i_1, i_2, \dots, i_s, i_1)$ uma rota euleriana

em A U E. Proceda como em 4. para obter uma rota hamiltoniana.
Christofides [Ch 2] mostra que para $n > 3$,

$$\frac{V^+}{V^*} < \frac{3}{2}$$

CAPÍTULO 5

APLICAÇÕES

Diversos problemas de programação inteira são modelados com base no PCV. Dentre eles destacam-se os seguintes:

1. SEQUENCIAMENTO DE TAREFAS USANDO 1 MÁQUINA [Ba 1]

Considere um conjunto de n tarefas independentes para serem processadas em uma única máquina. Seja p_j o tempo de processamento da tarefa j , $j = 1, \dots, n$ e c_{ij} , $i, j = 1, 2, \dots, n$ ($i \neq j$) o tempo de preparação da máquina para processar a tarefa j imediatamente após a execução da tarefa i . Deseja-se determinar uma sequência cíclica das tarefas que minimize o tempo total dispendido em preparação e processamento das tarefas. Se $(i_1, i_2, \dots, i_n, i_{n+1})$ denota uma permutação de $1, \dots, n$ correspondendo a uma sequência cíclica, isto é, a tarefa i_q é a q -ésima tarefa a ser executada, então o tempo total gasto em preparação e processamento pode ser expresso por

$$\sum_{q=1}^n c_{i_q, i_{q+1}} + p_{i_q} \quad \text{onde} \quad i_{n+1} = i_1$$

Desta forma, deseja-se resolver o problema

$$\text{Min}_{(i_q)} \left\{ \sum_{q=1}^n c_{i_q, i_{q+1}} \right\}$$

uma vez que $\sum_{q=1}^n p_{i_q}$ é constante para qualquer sequência. Entretanto, esta é uma das formulações possíveis do PCV, pois $(i_1, \dots, i_n, i_{n+1})$ pode ser interpretado como uma sequência de nós de uma rota hamiltoniana de uma rede e c_{ij} como o comprimento do arco (i, j) .

2. SEQUENCIAMENTO DE TAREFAS USANDO m MÁQUINAS [Ba 1]

Considere um conjunto de n tarefas independentes para serem processadas em uma sequência pré-fixada $\{1, 2, \dots, m\}$ de máquinas. São conhecidos os tempos p_{kj} de processamento da tarefa j na máquina k . Deseja-se determinar uma sequência de tarefas que minimize o tempo global dispendido no processamento.

Supondo que a tarefa j é executada imediatamente após a tarefa i , defina c_{ij} o atraso relativo de i para j , obtido através das expressões:

$$u_1 = p_{1i}$$

$$v_1 = u_1 + p_{1j}$$

$$\vdots$$

$$u_k = u_{k-1} + p_{ki}$$

$$v_k = \max \{u_k, v_{k-1}\} + p_{kj} \quad , \quad k = 2, \dots, m$$

$$c_{ij} = v_m - \sum_{k=1}^m p_{kj}$$

Observe que u_k, v_k representam os instantes de término do processamento das tarefas i, j na máquina k , respectivamente.

Para modelar este problema como um PCV, adicione uma tarefa fictícia $n+1$ com atrasos relativos:

$$c_{i,n+1} = \sum_{k=1}^m p_{ki} \quad v_i$$

$$c_{n+1,i} = 0 \quad v_i$$

Se $(i_1, \dots, i_n, i_{n+1})$ é uma permutação de $1, \dots, n+1$, representando um sequenciamento das tarefas (isto é, a tarefa i_q é a q -ésima tarefa a ser executada) então o tempo global é expresso por:

$$\sum_{q=1}^n c_{i_q, i_{q+1}} + \sum_{k=1}^m p_{k, i_{n+1}}$$

Desta forma, o problema torna-se

$$\text{Min}_{(i_q)} \left\{ \sum_{q=1}^n c_{i_q, i_{q+1}} + \sum_{k=1}^m p_{k, i_{n+1}} \right\}$$

Esta situação ocorre frequentemente em indústria de processamento de metais, onde todo o trabalho com o metal deve ser realizado à quente.

3. m-CAIXEIROS [Ch 1]

Considere um conjunto de $\{1, 2, \dots, n, n+1\}$ cidades. Suponha $n+1$ a cidade base. Deseja-se particionar o conjunto de cidades $\{1, \dots, n\}$ em m subrotas que possuem exatamente a cidade $n+1$ em comum, cujo comprimento total seja mínimo.

A solução deste problema é obtida duplicando $m-1$ vezes a cidade $n+1$, isto é, resolvendo o PCV para a matriz de distância $C' = (c'_{ij})$ de ordem $(n+m) \times (n+m)$ definida como segue:

$$c'_{ij} = \begin{cases} M & : n+2 \leq i, j \leq n+m, i \neq j \text{ e } 1 \leq i, j \leq n+m, i = j \\ c_{ij} & : 1 \leq i, j \leq n, \quad i \neq j \\ c_{nj} & : n+2 \leq i \leq n+m, \quad 1 \leq j \leq n \\ c_{in} & : 1 \leq i \leq n, \quad n+2 \leq j \leq n+m \end{cases}$$

onde M é um número positivo muito grande

4. PROGRAMAÇÃO DE VÔOS [Ch 1]

Considere uma companhia aérea que possui vôos com períodos idênticos (diários, mensais, etc.) conectando as cidades $1, 2, \dots, n$. Seja $m (m \geq 2)$ o número de tripulações. A cada tripulação deve ser atribuída uma rotaç o, ou seja, uma sequ ncia de vôos que inicia e termina com um v o fict cio (n  base) e que

faz conexões com outros vôos.

A cada conexão está associado um tempo de espera ou uma função de custo. Deseja-se obter m rotações que contenham todos os vôos e cujo tempo global seja mínimo.

Considere a rede $G = (N, A, C = (c_{ij}))$. Represente cada vôo por um nó e cada possível conexão entre dois vôos por um arco e seja c_{ij} o tempo de espera da conexão entre os vôos i, j .

5. PLANEJAMENTO DE CONEXÕES ELÉTRICAS EM COMPUTADORES [LR 1]

Este problema surge no projeto de interfaces de computadores. Um conjunto de pinos $N = \{1, \dots, n\}$ devem ser interligados na interface e cada pino pode receber no máximo dois fios de conexão por motivos tecnológicos. Deseja-se minimizar o comprimento total dos fios. Para modelar este problema, adicione um pino fictício $n+1$ com

$$c_{i,n+1} = c_{n+1,i} = 0, \quad \forall i$$

Desta forma, o problema consiste em resolver um PCV em uma rede completa não-orientada com $n+1$ nós (pinos), cuja solução ótima é uma rota hamiltoniana que especifica as interligações entre pinos.

6. AGRUPAMENTO DE DADOS [LR 1]

Considere uma matriz $0-1$ $C = (c_{ij})$ de medida de intensidade. Deseja-se determinar um agrupamento (uma permutação de linhas e uma permutação de colunas de C) que maximize uma medida de efetividade (ME), expressa pela soma de todos os produtos dos elementos de C adjacentes horizontalmente ou verticalmente.

Para modelar este problema como um PCV, sejam $P = \{1, \dots, p\}$, $Q = \{1, \dots, q\}$ os conjuntos de índices das linhas e das colunas de C ; e ρ, σ permutações de P, Q respectivamente.

Sejam

$$\rho(0) = \rho(p+1) = \sigma(0) = \sigma(q+1) = *$$

$$c_{i*} = c_{*j} = 0 \quad i \in P, j \in Q$$

assim,

$$ME(\rho, \sigma) =$$

$$= \frac{1}{2} \sum_{i \in P} \sum_{j \in Q} c_{\rho(i)\sigma(j)} (c_{\rho(i)\sigma(j-1)} + c_{\rho(i)\sigma(j+1)} + c_{\rho(i-1)\sigma(j)} + c_{\rho(i+1)\sigma(j)})$$

$$= \sum_{j=0}^q \sum_{i \in P} c_{i\sigma(j)} c_{i\sigma(j+1)} + \sum_{i=0}^p \sum_{j \in Q} c_{\rho(i)j} c_{\rho(i+1)j}$$

$$= ME(\sigma) + ME(\rho)$$

A solução do problema é obtida pela resolução de dois PCV'S:

$$\text{PCV}_c : c_{jk}^c = - \sum_{i \in P} c_{ij} c_{ik} ; j, k \in P \cup \{*\}$$

$$\text{PCV}_l : c_{ki}^l = - \sum_{j \in Q} c_{hj} c_{ij} ; h, i \in Q \cup \{*\}$$

Observe que ambos PCV's são simétricos.

CAPÍTULO 6

CASOS PARTICULARES

Embora não seja conhecido nenhum algoritmo polinomial para obter a solução ótima do PCV, existem casos particulares para os quais algoritmos eficientes são disponíveis.

1. SEQUENCIAMENTO DE TAREFAS EM FORNOS [GG 1]

Considere um problema de sequenciamento cíclico de tarefas (Capítulo 5, Seção 1) onde o processamento da tarefa i consiste em alterar a temperatura de um forno de s_i para t_i . Entre os processamentos das tarefas i, j é necessário preparar o forno, o que consiste em alterar a sua temperatura de t_i para s_j . Supõe-se que o custo de mudança de temperatura é dado por

$$c_{ij} = \begin{cases} \int_{t_i}^{s_j} f(x) \, dx & \text{se } t_i \leq s_j \\ \int_{s_j}^{t_i} g(x) \, dx & \text{se } t_i > s_j \end{cases}$$

onde f, g são funções integráveis, satisfazendo $f(x) + g(x) \geq 0$, $\forall x$.

O problema consiste em determinar uma permutação $(i_1, i_2, \dots, i_n, i_{n+1})$ que corresponde a uma sequência cíclica, isto

é, $i_{n+1} = i_1$, para a qual $\sum_{q=1}^n c_{i_q, i_{q+1}}$ seja mínima.

Em termos gerais, o algoritmo apresentado por Gilmore e Gomory [GG 1] para resolver este problema, constitui-se de três partes:

- 1) determinação de uma permutação de custo mínimo (designação usando algoritmo especializado), que não necessariamente corresponde a uma sequência cíclica.
- 2) determinação da árvore geradora mínima
- 3) com base na árvore geradora mínima, processar alterações na permutação de maneira a obter uma sequência cíclica de custo mínimo.

2. MATRIZ DE DISTÂNCIA TRIANGULAR SUPERIOR [La 1]

Seja $C = (c_{ij})$ a matriz de distância de ordem $n \times n$, tal que:

$$c_{ij} = 0 \quad \text{para} \quad i \geq j \quad i, j \in N$$

O algoritmo para resolver este problema, constitui-se de duas partes:

1) Resolução do problema de designação em C' de ordem $(n-1) \times (n-1)$ obtida deletando a primeira coluna e a última linha de C . A solução obtida compõe-se de uma cadeia P do nó 1 ao nó n e de subrotas. Caso a solução apresente apenas a cadeia do nó 1 ao nó n , introduza o arco $(n,1)$ para obter a rota hamiltoniana, senão vá para 2.

2) Remoção dos arcos (i,j) para $i \geq j$ (arcos com comprimento nulo) que não estão na cadeia P . Desta forma, as subrotas são desconectadas formando cadeias de i_1 a j_1, i_2 a $j_2, \dots, i_m$ a j_m onde $i_1 < i_2 < \dots < i_m$ e $i_1 \leq j_1, i_2 \leq j_2, \dots, i_m \leq j_m$. Uma rota hamiltoniana ótima com custo igual ao da designação é obtida ao introduzir os arcos de custo zero: $(n, i_m), (j_m, i_{m-1}), \dots, (j_2, i_1), (j_1, 1)$.

3. REDUÇÃO DE GRANDES PROBLEMAS

Rothkopf [Ro 1] considera problemas onde um subconjunto de nós $S = \{i_k, i_{k+1}, \dots, i_n\}$ satisfazendo a propriedade:

$$c_{ij} = \beta \quad ; \quad i, j \in S$$

$$c_{ik} = c_{jk} \quad ; \quad i, j \in S, k \notin S$$

$$c_{ki} = c_{kj} \quad ; \quad i, j \in S, k \notin S$$

$$\beta + c_{pq} \leq c_{pj} + c_{jq} \quad ; \quad j \in S, p, q \notin S$$

onde $\beta \in \mathbb{R}$. Adicione um pseudonó s representando o conjunto S e seja $C' = (c'_{ij})$ a nova matriz de distância de ordem $k \times k$ definida por:

$$c'_{ij} = c_{ij} \quad ; \quad i, j \notin S$$

$$c'_{sj} = c_{i_n j} \quad ; \quad j \notin S$$

$$c'_{is} = c_{i i_k} \quad ; \quad i \notin S.$$

Se $(i_1, i_2, \dots, i_{k-1}, s)$ é uma rota hamiltoniana ótima do PCV sobre C' então uma rota hamiltoniana ótima do PCV original é $(i_1, i_2, \dots, i_{k-1}, i_k, \dots, i_n)$.

Problemas que apresentam as distâncias entre as cidades de S com valores pequenos se comparadas com distância de cidades de S para fora de S podem ser aproximadas de forma a satisfazer esta propriedade. Tais problemas ocorrem com frequência em sequenciamento de tarefas onde existem grupos de tarefas similares.

Bellmore e Nemhauser [BN1], Cosmadakis e Papadimitriou [CP1] e Szwarcfiter [Sz1] tratam de problemas similares.

4. COLETA DE ITENS EM ALMOXARIFADOS [RR1]

Um veículo está disponível em um ponto base para coletar os itens de uma lista pré-fixada de um almoxarifado retangular e transportá-los ao ponto base do veículo. O almoxarifado é dividido em alas paralelas que permitem o trânsito do veículo. As

extremidades das alas são unidas por dois corredores que permitem a passagem do veículo de uma ala para qualquer outra. O objetivo é minimizar a distância percorrida pelo veículo que coleta todos os itens, passando pelo menos uma vez no local de cada item.

Construa a rede não-orientada $G = (N, A, C)$, onde o conjunto de nós N representa os pontos:

- i) base de veículo
- ii) posição de cada item
- iii) cada uma das extremidades de cada ala.

Cada aresta de A indica a possibilidade do veículo se locomover entre os dois nós sem passar por nenhum outro nó intermediário. Desta forma o grau de cada nó associado a um item é 2. O comprimento de cada aresta é proporcional à distância entre os dois nós unidos pela aresta.

Para resolver este problema, Ratliff e Rosenthal [RR 1] utilizam um algoritmo da programação dinâmica com um estágio para cada ala. A solução corresponde a uma rota que passa por cada nó associado ao item pelo menos uma vez e pode vir a utilizar uma mesma aresta duas vezes. O esforço computacional requerido é linear em relação ao número de alas.

CAPÍTULO 7

PROBLEMA DA COBERTURA SIMÉTRICA

Para redes orientadas satisfazendo a desigualdade triangular, a rota do Caixeiro Ótima é uma rota hamiltoniana mínima; e neste caso um modelo adequado é o do problema de designação (cobertura exata) acrescido com restrições de eliminação de subrotas. No caso mais geral do PCV em uma rede que não satisfaz a desigualdade triangular, é possível que a rota do caixeiro Ótima visite um mesmo nó mais do que uma vez, e neste caso é natural considerar o modelo do problema de cobertura simétrica (para cada nó o número de arcos chegando e saindo é igual e diferente de zero), também acrescido com as mesmas restrições de eliminação de subrotas. Neste Capítulo é apresentado um algoritmo eficiente que foi desenvolvido para resolver o problema de cobertura simétrica.

Considere uma rede orientada $G = (N, A, C)$ com $N = \{1, 2, \dots, n\}$. O problema consiste em determinar um subconjunto de arcos de comprimento total mínimo que cubra todos os nós com a propriedade de que cada nó apresente grau interno ($gr^-(\cdot)$) igual ao grau externo ($gr^+(\cdot)$). Portanto, deseja-se obter $X = (x_{ij})$, a solução Ótima de:

$$\begin{array}{ll} \text{Min} & \sum_{i,j} c_{ij} x_{ij} \\ \\ \text{sujeito a} & \sum_j x_{ij} \geq 1 \quad i \in N \end{array}$$

$$\sum_j x_{ji} \geq 1 \quad i \in N$$

$$x_{ij} = 0,1 \quad (i,j) \in A$$

$$\sum_j x_{ij} = \sum_j x_{ji} \quad i \in N$$

As restrições de simetria (igualdade de grau interno e grau externo) implicam que as variáveis de folga das restrições de cobertura interna e cobertura externa de um mesmo nó sejam iguais.

Introduzindo estas variáveis de folga e relaxando as restrições de integrabilidade, obtêm-se o problema

$$\text{Min} \quad \sum_{i,j} c_{ij} x_{ij}$$

$$\text{sujeito a} \quad \sum_j x_{ij} - y_i = 1 \quad i \in N$$

$$\sum_j x_{ji} - y_i = 1 \quad i \in N$$

$$0 \leq x_{ij} \leq 1 \quad (i,j) \in A$$

$$y_i \geq 0 \quad i \in N$$

O problema dual associado é apresentado abaixo juntamente com as restrições de folga complementar,

$$\text{Max } \sum_i u_i + \sum_j v_j + \sum_{i,j} w_{ij}$$

$$\text{su}j \quad u_i + v_j + w_{ij} \leq c_{ij} \quad (i,j) \in A$$

$$u_i + v_i \geq 0 \quad i \in N$$

$$w_{ij} \leq 0 \quad (i,j) \in A$$

$$(c_{ij} - u_i - v_j - w_{ij})x_{ij} = 0 \quad (i,j) \in A$$

$$(u_i + v_i) y_i = 0 \quad i \in N$$

$$w_{ij}(1 - x_{ij}) = 0 \quad (i,j) \in A$$

Se o custo relativo da linha (i,j) é definido por $d_{ij} = c_{ij} - u_i - v_j$, então para toda solução ótima, $w_{ij} = \min \{0, d_{ij}\}$.
Desta forma,

$$d_{ij} < 0 \implies x_{ij} = 1 \quad (i,j) \in A$$

$$d_{ij} > 0 \implies x_{ij} = 0 \quad (i,j) \in A$$

$$y_i > 0 \implies u_i + v_i = 0 \quad i \in N$$

É fácil verificar que o problema acima é um programa linear com coeficientes inteiros definido em uma rede bipartida.

Portanto, suas soluções básicas são inteiras. Além disto, estas soluções básicas são as soluções de um problema de cobertura com arestas em uma rede bipartida $G' = (N', A', C')$ com $N' = \{1, 2, \dots, 2n\}$, $A' = \{(i; j') : (i, j) \in A\} \cup \{(i, i') : i \in N\}$ onde $i' = 2n + 1 - i$.

$$c'_{ij'} = \begin{cases} c_{ij} & , j \neq i \\ 0 & , j = i \end{cases}$$

Desta forma, cada nó de G é desdobrado em dois nós em G' . Uma linha de G é denominada *ligação* quando associada a uma variável y e *aresta* quando associada a uma variável x e portanto correspondendo a um arco em G .

Seja $A^- = \{(i; j) \in A' : c_{ij} < 0\}$. Dada uma solução primal viável $(X, y) = ((x_{ij}), (y_i))$ e uma solução dual viável $(u, v) = ((u_i), (v_j))$, defina o conjunto de *linhas-de-igualdade* $A^- = \{(i; j) \in A' : d_{ij} = 0\}$ e o conjunto de *linhas-soluções* $A^* = \{(i; j) \in A' : x_{ij} = 1\} \cup \{(i; i) \in A' : y_i > 0\}$.

Um nó $i \in N'$ é *exposto* se

- i) $\sum_j x_{ij} = 0$ ou $\sum_j x_{ji} = 0$
 ii) $\sum_j x_{ij} < \sum_j x_{ji'}$ ou $\sum_j x_{ji'} < \sum_j x_{ij}$

Um nó $i \in N'$ é *supercoberto* se

$$\sum_j x_{ij} > \sum_j x_{ji}, > 0 \quad \text{ou} \quad \sum_j x_{ji} > \sum_j x_{ij} > 0$$

Os conceitos de árvore alternante, caminho aumentante, ampliação são os já apresentados no Capítulo 2 da Parte I (PCC para redes não-orientadas). Neste algoritmo, um caminho alternante é um caminho constituído por arestas/ligações-de-igualdade tal que

- i) arestas ímpares são do tipo não solução ($x = 0$)
- ii) arestas pares são do tipo solução ($x = 1$)
- iii) ligações ímpares são do tipo solução ($y = 1, 2, \dots$)
- iv) ligações pares são de tipo qualquer ($y = 0, 1, 2, \dots$)

As seguintes operações são realizadas pelo algoritmo:

Plantação da Árvore-executada na inicialização e após cada ampliação. Consiste em atribuir rótulo externo para os nós expostos e atribuir rótulos internos aos nós supercobertos.

Crescimento da Árvore (ou rotulação dos nós)- executada quando uma das quatro situações abaixo ocorre:

- 1) há uma aresta-de-igualdade não solução unindo um nó externo i a um nó sem rótulo j .

- 2) há uma ligação-de-igualdade solução unindo um nó externo i a um nó sem rótulo j ($j = i'$).
- 3) há uma ligação-de-igualdade não solução unindo um nó interno j a um nó sem rótulo k ($k = j'$).
- 4) há uma aresta-de-igualdade solução unindo um nó interno j a um nó sem rótulo k .
- 5) há uma ligação-de-igualdade solução unindo um nó interno j a um nó sem rótulo k ($k = j'$).

Nas situações 1) e 2) atribuir ao nó j um rótulo interno com predecessor i . Nas situações 3) e 4) e 5) atribuir ao nó k rótulo externo com predecessor j .

Eliminação das Árvores - executada depois de cada ampliação; elimina-se o rótulo de todos os nós.

Ampliação - executada quando uma das três situações abaixo ocorre:

- 1) há uma aresta-de-igualdade não solução unindo dois nós externos i, j .
- 2) há uma aresta-de-igualdade solução unindo dois nós internos i, j .
- 3) há uma ligação-de-igualdade não solução unindo dois nós internos i, j ($j = i'$).

Uma ampliação consiste em alterar o papel solução/não solu-

ção das arestas do caminho aumentante $P = P_i \cup (i;j) \cup P_j$. Se $(p;q) \in P$ então o novo valor de x_{pq} será:

$$x_{pq} \leftarrow 1 - x_{pq}$$

Se $(p;p') \in P$ então o novo valor de y_p será

$$y_p \leftarrow \begin{cases} y_p + 1 & \text{se } (p;p') \text{ é uma ligação par de } P \\ y_p - 1 & \text{se } (p;p') \text{ é uma ligação ímpar de } P \end{cases}$$

ALGORITMO

INICIALIZAÇÃO

0. Atribua

$$u_i \leftarrow 0 \quad i \in N$$

$$v_{i'} \leftarrow 0 \quad i' \in N$$

$$x_{ij} \leftarrow \begin{cases} 1 & (i;j) \in A^- \\ 0 & (i;j) \in A \setminus A^- \end{cases}$$

$$\text{Max} \left\{ \min \left\{ \sum_j x_{ij}, \sum_j x_{ji} \right\} - 1, 0 \right\}$$

INVESTIGAÇÃO DA SOLUÇÃO PRIMAL

1. Atribua rótulos externos aos nós expostos e rótulos internos aos nós supercobertos. Estes nós são as raízes das árvores alternantes.
2. Se não existe nós expostos, termine.
3. Se existe uma aresta-de-igualdade não solução unindo um nó externo i a um nó sem rótulo j , execute o crescimento da árvore e vá para 2.
4. Se existe uma aresta-de-igualdade solução unindo um nó interno j a um nó sem rótulo k , execute o crescimento da árvore e vá para 2.
5. Se existe uma ligação-de-igualdade solução unindo um nó interno i a um nó sem rótulo i' , execute o crescimento da árvore e vá para 2.
6. Se existe uma ligação-de-igualdade não solução unindo um nó interno j a um nó sem rótulo j' , execute o crescimento da árvore e vá para 2.
7. Se existe uma aresta-de-igualdade não solução unindo dois nós externos i, j , execute a ampliação e vá para 1.

8. Se existe uma ligação-de-igualdade não solução unindo dois nós internos i, i' , execute a ampliação e vá para 1.

9. Se existe uma ligação-de-igualdade solução unindo um nó externo i a um nó sem rótulo i' , execute o crescimento da árvore e vá para 2.

10. Se existe uma aresta-de-igualdade solução unindo dois nós internos execute a ampliação e vá para 1.

MUDANÇA DA SOLUÇÃO DUAL

11. Sejam

$\epsilon_1 = \min \{d_{ij} : (i;j) \text{ é uma aresta não solução unindo um nó interno a um nó sem rótulo}\}$

$\epsilon_2 = \min \{-d_{ij} : (i;j) \text{ é uma aresta solução unindo um nó interno a um nó sem rótulo}\}$

$\epsilon_3 = \min \{\frac{1}{2} d_{ij} : (i;j) \text{ é uma aresta não solução unindo dois nós externos}\}$

$\epsilon_4 = \min \{-\frac{1}{2} d_{ij} : (i;j) \text{ é uma aresta solução unindo dois nós internos}\}$

$\epsilon_5 = \min \{ d_{ii'} : (i; i') \text{ é uma ligação não solução unindo um} \\ \text{nó interno a um nó sem rótulo} \}$

$\epsilon_6 = \min \{ \frac{1}{2} d_{ii'} : (i; i') \text{ é uma ligação não solução unindo dois} \\ \text{nós internos} \}$

$$\epsilon = \min \{ \epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4, \epsilon_5, \epsilon_6 \}$$

Se $\epsilon \longrightarrow \infty$, o problema é inviável, termine, caso contrário, para cada $i \in N$, faça:

$$u_i \longleftarrow \begin{cases} u_i + \epsilon & \text{se } i \text{ é nó externo} \\ u_i - \epsilon & \text{se } i \text{ é nó interno} \end{cases}$$

$$v_{i'} \longleftarrow \begin{cases} v_{i'} + \epsilon & \text{se } i' \text{ é nó externo} \\ v_{i'} - \epsilon & \text{se } i' \text{ é nó interno} \end{cases}$$

e vá para 2.

Observe que as condições abaixo são sempre satisfeitas:

$$x_{ij} \in \{0,1\} \quad (i,j) \in A$$

$$y_i \in \{0,1,2,\dots\} \quad i \in N$$

$$u_i + v_j + w_{ij} \leq c_{ij} \quad (i,j) \in A$$

$$u_i + v_i \geq 0 \quad i \in N$$

$$w_{ij} \leq 0 \quad (i,j) \in A$$

Seja $d_{ij} = c_{ij} - u_i - v_j$

$$d_{ij} < 0 \implies x_{ij} = 1 \quad (i,j) \in A$$

$$d_{ij} > 0 \implies x_{ij} = 0 \quad (i,j) \in A$$

$$y_i > 0 \implies d_{ii'} = 0 \quad i \in N$$

$$\sum_j x_{ij} = 0 \implies i \text{ é não exposto}$$

$$\sum_j x_{ji'} = 0 \implies i' \text{ é não exposto}$$

$$\sum_j x_{ij} < \sum_j x_{ji} \implies i \text{ é não exposto}$$

$$\sum_j x_{ij} > \sum_j x_{ji'} > 0 \quad \text{ou} \quad \sum_j x_{ji} > \sum_j x_{i'j} > 0$$

$$\implies i \text{ é não supercoberto}$$

e que quando o algoritmo termina com a inexistência de nós expostos e supercobertos, então as condições acima garantem que a solução é ótima.

É fácil verificar que quando o algoritmo termina com $\epsilon \rightarrow \infty$ que o vetor $(u, v) + \psi(\bar{u}, \bar{v})$ é uma solução dual viável para $\psi > 0$ sendo (u, v) a solução dual corrente e onde cada componente de $(\bar{u}, \bar{v})$ é 0, +1, -1 se o nó correspondente é sem rótulo, ou rótulo externo, ou rótulo interno, respectivamente. Fazendo $\psi \rightarrow \infty$ tem-se que a função objetivo dual tende para infinito, pois o número de nós com rótulo externo é maior que o número de nós com rótulo interno.

Além disto, a cada alteração da solução dual, tem-se $\epsilon > 0$ devido ao modo com que os rótulos são atribuídos.

TEOREMA: O esforço computacional requerido pelo algoritmo da cobertura simétrica é limitado superiormente por $O(n^2 m)$ para $C \geq 0$.

PROVA: Existirão no máximo n ampliações durante algoritmo, pois a cada ampliação dois nós expostos tornam-se cobertos e permanecem cobertos na execução restante do algoritmo. A cada execução da solução dual seguida por uma ampliação implica necessariamente no crescimento da árvore e no máximo podem ser executadas $O(n)$ crescimentos de árvores entre duas ampliações. O esforço requerido para realizar uma mudança da solução dual é $O(m)$ e os demais passos podem ser implementados com esforço computacional $O(n)$. Desta forma, o esforço computacional requerido pelo algoritmo antes do término é limitado superiormente por $O(n^2 m)$, supondo $m > n$.

É possível utilizar as técnicas descritas por Lawler [La 2] para obter uma implementação com esforço computacional $O(n^3)$.

CAPÍTULO 8

PROBLEMA DO CAIXEIRO VIAJANTE TIPO "GARGALO" (BOTTLENECK)

Considere uma rede $G = (N, A, C)$ orientada. O Problema do Caixeiro Viajante tipo Gargalo (PCVG) consiste em determinar uma rota do caixeiro em G , cujo comprimento do maior arco é mínimo.

A diferença entre o PCV e o PCVG reside apenas na função objetivo, pois o conjunto de soluções viáveis é idêntico. Assim como o PCV, o PCVG (restritos a rotas hamiltonianas) é um Problema NP-árduo (NP-"hard") [GJ 1].

Na obtenção de soluções exatas para o PCVG, o método mais empregado é o método de particionar e limitar (vide Capítulo 3, Seção 4). Os esquemas de limitação apresentados na literatura são baseados nos problemas abaixo.

a) Designação tipo Gargalo

Uma formulação do PCVG assimétrico (restrito a rotas hamiltonianas) é

$$\text{Min} \quad \text{Max} \quad \{c_{ij} : x_{ij} = 1\}$$

$$\text{Suj} \quad \sum_j x_{ij} = 1 \quad i \in N$$

$$\sum_j x_{ji} = 1 \quad i \in N$$

$$x_{ij} = 0, 1 \quad (i, j) \in A$$

$$X = (x_{ij}) \in T$$

onde T denota o conjunto das designações que satisfazem as restrições de eliminação de subrotas.

A relaxação das restrições de eliminação de subrotas " $X \in T$ " na formulação acima produz o Problema da Designação tipo Gargalo (PDG).

Burkard, Hahn e Zimmermann [BH 1] Cross [Cr 1] Garfinkel [Ga 1], Derigs e Zimmermann [Dz 1], Carpaneto e Toth [CT3] apresentam algoritmos polinomiais para resolver o PDG.

Se a solução ótima do PDG é uma rota hamiltoniana então esta é a solução ótima do PCVG. Caso contrário, a solução ótima do PDG é composta, digamos, das subrotas $\{S_1, S_2, \dots, S_m\}$ e constitui em um limitante inferior para o PCVG. Neste caso, um melhor limitante inferior obtido a partir das subrotas do PDG é dado por

$$\begin{array}{ll} \text{Max} & b_k \\ & 1 \leq k \leq m \end{array}$$

onde

$$b_k = \min \{ \max \{ c_{st}, c_{uv} \} : (s, t, u, v) \in K_k \}$$

e

$$K_k = \{ (s, t, u, v) \in N^4 : s \neq v, t \neq u, s, v \in S_k, t, u \notin S_k \}$$

b) 1 - Arborescência tipo Gargalo

Uma formulação do PCVG assimétrico (restrito a rotas hamiltonianas) é

$$\text{Min} \quad \text{Max} \quad \{c_{ij} : x_{ij} = 1\}$$

$$\text{su}j \quad X \in A_b$$

$$X \in T$$

onde $A_b = \{1\text{-arborescências de } G\}$ e T denota o conjunto das 1 - arborescências que satisfazem as restrições de eliminação de subrotas.

A relaxação das restrições de eliminação de subrotas " $X \in T$ " na formulação acima produz o Problema da 1 - Arborescência tipo Gargalo (PAG).

Uma 1 - arborescência sempre contém uma cadeia do nó raiz a qualquer outro nó e constitui um limitante inferior para o PCVG restrito a rotas hamiltonianas. Entretanto, este limitante (cadeias entre cada par de nós) é sobrepujado pelo PAG.

Gilmore e Gomory [GG 1] apresentam um algoritmo polinomial para um caso particular do PCVG.

Finalmente, considere o PCVB (Problema do Caixeiro Viajante Balanceado) que consiste em determinar uma rota do caixeiro cuja diferença entre o maior arco e o menor arco é mínima. Assim,

o PCV, o PCVG e o PCVB possuem o mesmo conjunto de soluções viáveis (isto é, rotas do caixeiro). Para determinar o valor de uma rota do caixeiro, considera-se o comprimento dos arcos da rota; no PCV, o valor da rota é a soma destes comprimentos, e no PCVG o valor da rota é o maior destes comprimentos, e no PCVB o valor da rota é a diferença entre o maior e o menor comprimento.

APÊNDICE I

DEFINIÇÕES

$G = (N, A)$ é um *grafo* se N é um conjunto finito não-vazio de n *nós* e A é um conjunto finito não-vazio de m *linhas* (pares de nós distintos).

Uma *aresta* é um par não-ordenado $(i; j)$ de nós distintos.

Um *arco* é um par ordenado (i, j) de nós distintos; (i, j) são a cauda e a cabeça do arco, respectivamente)

Um grafo $G = (N, A)$ é *não-orientado* se N é um conjunto de nós e A é um conjunto de *arestas*.

Um grafo $G = (N, A)$ é *orientado* se N é um conjunto de nós e A é um conjunto de *arcos*.

Um grafo $G = (N, A)$ é *completo* se cada par de nós é unido por uma linha (aresta ou arco).

Um grafo $\underline{G} = (\underline{N}, \underline{A})$ é um *subgrafo* de um grafo $G = (N, A)$ se $\underline{N} \subset N$ e $\underline{A} \subset A$.

Um nó i de G é *isolado* se não existe nenhuma linha (aresta ou arco) incidente com ele.

Dois nós são *adjacentes* se existe uma linha (aresta ou arco) unindo-os.

PCC - Problema do Carteiro Chinês

PCV - Problema do Caixeiro Viajante

PCV *simétrico* se G é um grafo não-orientado

PCV *assimétrico* se G é um grafo orientado

Dado um grafo $G = (N, A)$, considere uma sequência finita $P = (i_1, l_1, i_2, l_2, \dots, i_{q-1}, l_{q-1}, i_q)$ onde $i_1, \dots, i_q$ são nós de

G e cada ℓ_j é uma linha de G que une os nós i_j, i_{j+1} para $j = 1, \dots, q-1$.

P é um caminho conectando i_1, i_q se $i_1, \dots, i_q$ são nós distintos.

P é uma cadeia conectando i_1, i_q se $i_1, \dots, i_q$ são nós distintos e cada ℓ_j é um arco de i_j para i_{j+1} .

P é um ciclo se P é um caminho com $i_1 = i_q$.

P é um circuito se P é uma cadeia com $i_1 = i_q$.

P é uma rota se $i_1 = i_q$.

P é uma rota do caixeiro se $i_1 = i_q$ e $\{i_1, i_2, \dots, i_q\} = N$

P é uma rota hamiltoniana se $i_1 = i_q$, $\{i_1, i_2, \dots, i_q\} = N$

e $q = n$.

P é uma subrota se $i_1 = i_q$ e $\{i_1, i_2, \dots, i_q\} \subsetneq N$.

P é uma rota do carteiro se $i_1 = i_q$, $\{\ell_1, \ell_2, \dots, \ell_{q-1}\} = A$

e $q-1 \geq m$.

P é uma rota euleriana se $i_1 = i_q$, $\{\ell_1, \ell_2, \dots, \ell_{q-1}\} = A$

e $q-1 = m$.

G é um grafo hamiltoniano se possui uma rota hamiltoniana.

G é um grafo euleriano se possui uma rota euleriana.

$G = (N, A, C)$ é uma rede se (N, A) é um grafo e $C = (c_{ij})$ é a matriz de distância em G .

$\text{gr}(i)$ denota o grau do nó i de G , isto é, o número de linhas de G incidentes com i .

$\text{gr}^+(i)$ denota o grau externo do nó i de G , isto é, o número de arcos de G incidentes de i .

$gr^-(i)$ denota o grau interno do nó i de G , isto é, o número de arcos de G incidentes em i .

$G = (N, A, C)$ é uma rede par se $gr(i)$ é par para todo i de G .

$G = (N, A, C)$ é uma rede simétrica se $gr^+(i) = gr^-(i)$ para todo i de G .

$G = (N, A, C)$ é uma rede conectada se cada par de nós é unido por um caminho.

$G = (N, A, C)$ é uma rede fortemente conectada se para cada par de nós i, j existe uma cadeia de i a j e uma cadeia de j a i .

APÊNDICE II

RESULTADOS COMPUTACIONAIS

Foi feita uma implementação do método do tipo Particionar Limitar para resolver o PCV assimétrico utilizando FORTRAN-10 no computador DEC-10 da UNICAMP.

O método utiliza:

- i) procedimento de limitação-algoritmo de Cobertura Simétrica apresentado no Capítulo 7;
- ii) procedimento de partição-orientado para corte;
- iii) disciplina - LIFO

O programa foi executado para redes geradas aleatoriamente com as seguintes características:

- a) redes orientadas com 20 nós, 100 arcos e custos variando em $[0,10]$ - foram obtidos tempos de cpu de 1,51s a 116,78s.
- b) redes orientadas com 20 nós, 100 arcos e custos variando em $[0,100]$ - foram obtidos tempos de cpu de 1,35s a 119,14s.
- c) redes orientadas com 20 nós, 100 arcos e custos variando em $[0,1000]$ - foram obtidos tempos de cpu de 1,49s a 109,50s.
- d) redes orientadas com 20 nós, 100 arcos e custos variando em $[-10,10]$ - foram obtidos tempos de cpu de 1,03s a 8,94s.

- e) redes orientadas com 10 nós, 50 arcos e custos variando em $[0,100]$ - foram obtidos tempos de cpu de 0,55s a 6,79s.
- f) rede não-orientada com 20 nós, 100 arestas (200 arcos) e custos variando em $[0,10]$ - tempo de cpu excedido.
- g) rede não-orientada com 20 nós, 50 arestas (100 arcos) e custos variando em $[0,10]$ - foi obtido tempo de cpu de 1732,78s.
- h) redes não-orientadas com 10 nós, 25 arestas (50 arcos) e custos variando em $[0,100]$ - foram obtidos tempos de cpu de 8,52s a 26,38s.
- i) redes não-orientadas completas com 10 nós e 45 arestas (90 arcos) satisfazendo a desigualdade triangular. Os nós foram aleatoriamente gerados em um quadrado com vértices $(0,0), (0,1), (1,1), (1,0)$ - foram obtidos tempos de cpu de 162,81s a 1029,66s.

Foram feitas as seguintes observações:

- a) A transformação de um PCV simétrico (definido em rede não-orientada) em um PCV assimétrico (definido em rede orientada) através da substituição de cada aresta por dois arcos com sentidos contrários conduz a soluções com um número muito grande de subrotas de comprimento 2. Isto implica em um acréscimo muito grande no número de subproblemas

gerados e no aumento do tempo de execução do algoritmo.

- b) Para os problemas gerados em redes orientadas com 20 nós e 100 arcos, o maior número de subproblemas gerados foi 158, dos quais 111 com valor da solução não-melhor do que o valor da solução corrente foram eliminados, 40 necessitaram de aplicação do procedimento de partição, e 7 apresentaram como solução uma rota do caixeiro; isto é, ocorreram 7 soluções correntes. O número máximo de subproblemas na lista foi 42 sendo que a solução ótima foi obtida na resolução do 135º subproblema. Todos os 20 problemas gerados eram viáveis e apenas 3 apresentaram uma rota hamiltoniana como solução ótima.
- c) O problema gerado em rede não-orientada com 20 nós e 100 arestas (200 arcos) gerou mais de 5000 subproblemas e excedeu o tempo limite de execução.
- d) Para o problema gerado em rede não-orientada com 20 nós e 50 arestas (100 arcos) foram gerados 2897 subproblemas, dos quais 2 inviáveis, 2416 eliminados, 468 particionados e apenas 13 com novas soluções correntes. O número máximo de subproblemas na lista foi 32, e a solução ótima foi obtida na resolução do 1733º subproblema.
- e) O procedimento de limitação do PCV assimétrico em cobertu-

tura simétrica produziu limitantes inferiores com valores variando de 67% a 100% da solução ótima.

- f) Comparando os 5 problemas gerados em redes orientadas com 10 nós e 50 arcos com os 5 problemas gerados em redes não-orientadas com 10 nós e 25 arestas (50 arcos) e com os 5 problemas gerados em redes completas com 10 nós e 45 arestas (90 arcos) observou-se:

rede	orientada	não-orientada	completa
tempo de cpu	0,38 a 6,79	8,52 a 26,38	162,81 a 1029,66s
nº subproblemas gerados	1 a 35	42 a 193	852 a 6309
nº máx. subpr. na lista	1 a 13	11 a 17	45 a 49
nº problemas viáveis	3 em 5	5 em 5	5 em 5
nº rotas ótimas hamiltonianas	1 em 3	0 em 5	5 em 5
valor rota ótima	184 a 249	251 a 328	27 a 36
taxa 1ª solução/sol.ótima	0,97 a 1,00	0,67 a 0,94	0,75 a 0,92

REFERÊNCIAS

- [Al 1] Y.A.ALYAHYA. Matching/Edge-Covering Problems and Blossom Algorithms, proposta para tese de doutoramento. Department of Industrial and Operations Engineering. The University of Michigan, Ann Arbor (1981).
- [Ba 1] K.R.BAKER. Introduction Sequencing and Scheduling. New York, J. Wiley (1974).
- *[BB 1] E.J.BELTRAMI e L.D.BODIN. Networks and vehicle routing for municipal waste collection, Report Nº UPS 72-18, State University of New York, Stony Brook, New York (1972).
- [BC 1] E.BALAS e N.CHRISTOFIDES A Restricted Lagrangian Approach to the Traveling Salesman Problem. *Mathematical Programming*. 21(1981), pág. 19.
- *[BH 1] R.E.BURKARD, W.HAHN e U.ZIMMERMANN. An Algebraic Approach to Assignment Problems. *Mathematical Programming* 12(1977), pág. 318.
- [BM 1] M.BELLMORE e J.C.MALONE. Pathology of Traveling Salesman Subtour Elimination Algorithms. *Operations Research* 19 (1971), pág. 278.
- [BN 1] M.BELLMORE e G.L. NEMHAUSER. The Traveling Salesman Problem: a survey. *Operations Research* 16(1966), pág. 538.
- [BS 1] M.BAZARAA e H.D.SHERALI. On the Choice of Step Size in Subgradient Optimization. *European Journal of Operational Research* 7(1981), pág. 380.

- [Be 1] C.BERGE. *Graphes et Hypergraphes*. Paris, Dunod (1973).
- [Ch 1] N.CHRISTOFIDES. *Combinatorial Optmization*. New York, J.Wiley (1979).
- *[Ch 2] N.CHRISTOFIDES. Works - Case Analysis of a New Heuristic for the Traveling Salesman Problem. *Technical Report, Graduate School of Industrial Administration, Carnegie - Mellon University* (1976).
- [CM 1] G.CARPANETO, S.MARTELO e P.TOTH. An Algorithm for the Bottleneck Traveling Salesman Problem. *Operations Research* 32(1984), pág. 380.
- *[CP 1] S.COSMADAKIS e C.H.PAPADIMITROU. The Traveling Salesman Problem with Many Visits to Few Cities. *SIAM Journal Computing* 13(1984), pág. 99.
- [CR 1] O.CROSS. The Bottleneck Assignment Problem. P-1630, *The Rand Corporation, Santa Mônica, Ca.* (1959).
- *[CT 1] G.CARPANETO e P.TOTH. An Efficient Algorithm for the Asymmetric Salesman Problem, apresentado na *ORSA/TMS Joint National Meeting, Atlanta, GA* (1977).
- [CT 2] G.CARPANETO e P. TOTH. Branching and Bounding Criteria for the Asymmetric Traveling Salesman Problem. *Management Science* 26(1980), pág. 736.
- [CT 3] G.CARPANETO e P.TOTH. Algorithm for the Solution of the Bottleneck Assignment Problem. *Computing* 27(1981), pág. 179.

- [CV 1] V.CHVÁTAL. Edmonds Polytopes and Weakly Hamiltonian Graphs. *Mathematical Programming* 5(1973),pág. 29.
- *[CW 1] N.CHRISTOFIDES e C.WHITLOCK. An LP-Based Traveling Salesman Problem Algorithm. *Imperial College Report on* 78-14(1978).
- *[DF 1] G.DANTZIG, D.R. FULKERSON e S.JOHNSON. Solution of a Large Scale Traveling Salesman Problem. *Operations Research* 2(1954), pág. 393.
- [DL 1] S.E.DREYFUS e A.M.LAW. *The Art and Theory of Dynamic Programming*. London, Academic Press (1977).
- *[DZ 1] U.DERIGS e U.ZIMMERMANN. An Augmenting Path Method for Solving Linear Bottleneck Assignment Problems. *Computing*, 19(1978), pág. 285.
- [Ed 1] J.EDMONDS. Paths, Trees and Flowers. *Canadã J.Math.* 17(1965), pág. 449.
- [Ed 2] J.EDMONDS. Maximum Matching and a Polyedron With 0, 1-Vértices. *Journal of Research NBS.* 69(1965),pág. 125.
- [Ed 3] J.EDMONDS. The Chinese Postman Problem. *Operacional Research* 13 Suppl. 1(1965), pág. 373.
- [EJ 1] J.EDMONDS e E.L.JOHNSON. Matching: A Well-Solved Class of Integer Linear Programs. In: R. G.Guy, H. Hanani, N. Sauer e J.Schonhein, eds., *Combinatorial Structure and Their Applications* (Gordon and Breach, New York) (1970), pág. 89.

- [EJ 2] J. EDMONDS e E.L. JOHNSON. Matching, Euler Tours and the Chinese Postman. *Mathematical Programming* 5(1973), pág. 88.

- [Ga 1] R.S. GARFINKEL. An Improved Algorithm for the Bottleneck Assignment Problem. *Operations Research* 19(1971), pág. 1747.

- [Ga 2] R.F. GARFINKEL. On Partitioning the Feasible Set in a Branch-and-Bound Algorithm for the Asymmetric Traveling Salesman Problem. *Operations Research* 2(1973), pág. 340.

- [GG 1] P.C. GILMORE e R.E. GOMORY. Sequencing a One State-variable Machine: A Solvable Case of the Traveling Salesman Problem. *Operations Research* 12(1964), pág. 659.

- *[GG 2] R.S. GARFINKEL e K.C. GILBERT. The Bottleneck Traveling Salesman Problem: Algorithms and Probabilistic Analysis. *Journal Association Computation Mach.* 25(1978), pág. 435.

- [GJ 1] M.R. GAREY e D.S. JOHNSON. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. São Francisco, W.H. Freeman (1979).

- [GK 1] J.J. GREEN e B.A. KRÖTI. Matching Polyhedra. Tese de Mestrado, Department of Mathematics and Statics, Carleton University, Ottawa (1980).

- [GN 1] R.F. GARFINKEL e G.L. NEMHAUSER. *Integer Programming*. New York, J. Wiley (1972).

- [GP 1] M.GRÖTSCHEL e M.W.PADBERG. On the Symmetric Traveling Salesman Problem I: Inequalities. *Mathematical Programming* 16(1979), pág. 265.
- [GP 2] M.GRÖTSCHEL e M.W.PADBERG. On the Symmetric Traveling Problem II: Lifting Theorems and Facets. *Mathematical Programming* 16(1979), pág. 281.
- *[Gr 1] M.GRÖTSCHEL. An Optimal Tour Through 120 Cities in Germany. *Report 7770*, University of Bonn.
- *[Gu 1] M.GUAN: On the Windy, Postman Problem. *Discrete Applied Mathematics* 9(1984), pág. 41.
- [Ha 1] F.HARARY. *Graph Theory*. London, Addison-Wesley(1979).
- *[He 1] I.HELLER. On the Traveling Salesman's Problem. In: *Proceedings of the Second Symposium in Linear Programming* NBS. Washington, DC (1955), pág. 643.
- [HK 1] M.HELD e R.M.KARP. The Traveling Salesman Problem and Minimum Spanning Trees. *Operations Research* 18(1970) , pág. 1138.
- [HK 2] M.HELD e R.M.KARP. The Traveling Salesman Problem and Minimum Spanning Trees: Part II. *Mathematical Programming* 1(1971), pág. 6.
- [HN 1] W.W.HARDGRAVE e G.L.NEMHAUSER. On the Relation Between the Traveling Salesman and the Longest-Path Problems. *Operations Research* 10(1962), pág. 647.

- [HW 1] M.HELD e H.P.CROWDER. Validation of Subgradient Optimization. *Mathematical Programming* 6(1974), pág. 62.
- *[Ka 1] R.M.KARP. A Patching Algorithm for the Nonsymmetric Traveling Salesman Problem. *SIAM Journal of Computing* 8(1979), pág. 561.
- *[Ka 2] R.M.KARP. Probabilistic Analysis of Partitioning Algorithms for the Traveling Salesman Problem in the Plane. *Mathematical Operations Research* 2(1977), pág. 209.
- [La 1] E.L.LAWLER. A Solvable Case of the Traveling Salesman Problem. *Mathematical Programming* 1(1971), pág. 267.
- *[La 2] E.L.LAWLER. *Combinatorial Optimization: Networks and Matroids*. New York, Holt, Rinehart Winston (1976).
- [LR 1] J.LENSTRA e K.A.RINNOOY. Some Simple Applications of the Traveling Salesman Problem. *Operations Research Quarterly* 24(1975), pág. 717.
- [Ma 1] J.E.MAGALHÃES MOTTA. *Pert Custo Tempo*. Rio de Janeiro, Spencer(1969).
- *[Me 1] K.MENGER. Botenproblem. In: K.Menger, ed., *Ergebnisse eines Mathematischen Kolloquiums* (Wien 1930), Heft 2 (Leipzig, 1932), pág. 11.
- [Mk 1] M.MEI-KO. Graphic Programming Using Odd or Even Points. *Chinese Mathematics*. 11(1962), pág. 273.

- *[Ml 1] P.MILIOTS. Integer Programming Approaches to the Traveling Salesman Problem. *Mathematical Programming* 6 (1976), pág. 367.
- [Mi 1] E.MINIEKA. *Optimization Algorithms for Networks and Graphs*. New York, M.Dekker (1978).
- *[Mi 2] E.MINIEKA. The Chinese Postman Problem for Mixed Networks. *Management Science* 25(1979), pág. 643.
- [MP 1] K.G.MURTY e C.PERIN. A 1-Matching Blossom-Type Algorithm for Edge Covering Problems. *Networks* 12(1982), pág. 379.
- [Mu 1] K.G.MURTY. On the Tours of a Traveling Salesman. *SIAM Journal on Control* 7(1969), pág. 122.
- [Mu 2] K.G.MURTY. *Linear and Combinatorial Programming*. New York, J.Wiley (1976).
- [PC 1] W.PULLEYBLANK e G.CORNUÉJOLS. The Traveling Salesman Polytope and $\{0,2\}$ -Matching. *Management Science Research Report n°470* Carnegie-Mellon University, Pittsburgh (1981).
- [Pl 1] B.T.POLJAK. A General Method of Solving Extremum Problems. *Soviet. Mathematics* 8(1967), pág. 593.
- *[Pl 2] B.T.POLJAK. Minimization of Unsmooth Functionals. *U.S.S.R. Computational Mathematics and Mathematical Physics* 14-29 (Tradução de *Zurnal Vyčislitel'noi Matematiki i Matematiches Fiziki* 9(1969), pág. 509).

- *[Po 1] L.PÓSA. A Theorem Concerning Hamilton Lines. *Magyar Tud. Abad. Mat. Kutato Int. Kozl.* 7(1962), pág. 225.
- [PR 1] R.G.PARKER e R.L.RARDIN. The Traveling Salesman Problem: an Update of Research. *Naval Research Logistics Quarterly* 30(1983), pág. 69.
- [PS 1] G.H.PAPADIMITROU e K.STEIGHITZ. *Combinatorial Optimization: Algorithms and Complexity*, Englewood Cliffs, Prentice-Hall (1982).
- [Ro 1] M.ROTHKOPF. The Traveling Salesman Problem: On the Reduction of Certain Large Problems to Smaller Ones. *Operations Research* 14(1966), pág. 532.
- [RR 1] H.D.RATLIFF e A.S.ROSENTHAL. Order-Picking in a Rectangular Warehouse: A Solvable Case of the Traveling Salesman Problem. *Operations Research* 31(1982), pág. 507.
- *[RS 1] D.J.ROSENKRAUTZ, R.E.STEARNs e P.M.LEWIS. Approximate Algorithms for the Traveling Salesman Problem. *Proc. 15th IEEE Symp. on Switching and Automata Theory.*(1974), pág. 33.
- *[RS 2] D.J.ROSENKRAUTZ, R.E.STEARNs e P.M.LEWIS. II.Approximation Algorithms for the Traveling Salesperson Problem. *Report Nº 762 CRD266, Automation and Control Laboratory* (1976).
- *[St 1] R.STRICKER. *Public Sector Vehicle Routing: The Chinese Postman Problem.* Master Thesis, Massachusetts

Institute of Technology, Cambridge, Mass. (1970).

- [Sz 1] J.L.SZWARCFITER. *An ILP of the Many-Visits Traveling Salesman Problem*. Apresentado na School of Combinatorial Optimization. NEC/UFRJ, Brasil (1985).

- [VJ 1] T.VOLGENANT e R.JONKER. *A Branch and Bound Algorithm for the Symmetric Traveling Salesman Problem Based on the 1-Tree Relaxation*. *European Journal of Operational Research* 9(1982), pág. 83.

(*) Referências adicionais.