CONTRIBUIÇÃO PARA A RESOLUÇAO DO PROBLEMA DE STEINER NUM GRAFO DIRECIONADO: UM MÉTODO HEURÍSTICO

Oscar Iván Palma Pacheco

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENACAO DOS PROGRAMAS DE POSGRADUACAO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS (D.Sc.) EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Aprovada por:

Nelson Maculan Filho

Nelson Maculan Filho (UFRJ)
(Presidente)

Geraldo da Silva e Souza (EMBRAPA)

Hilton Vieira Machado (UnB)

Paulo Oswaldo Boaventura Netto (UFRJ)

Celso da Cruz Carneiro Ribeiro (PUC/RJ)

RIO DE JANEIRO, RJ - BRASIL

JUNHO DE 1985

PALMA PACHECO, OSCAR IVAN

Contribuição para a Resolução do Problema de Steiner num Grafo Direcionado: um Método Heurístico. [Rio de Janeiro] 1985.

IX, 132 p. 29,7 cm (COPPE/UFRJ, D.Sc., Engenharia de Sistemas e Computação, 1985)

Tese - Universidade Federal do Rio de Janeiro, COPPE.

1. Problema de Steiner I. COPPE/UFRJ II. Titulo (série).

A minha esposa
Georgina e aos meus
filhos Gonzalo e
Maria Paz, eu dedico.

AGRADECIMENTOS

- Ao meu orientador, Nelson Maculan Filho, pela dedicação, apoio e colaboração.
- Aos amigos e colegas do Programa de Engenharia de Sistemas e Computação, em especial ao pessoal do Laboratório, pela colaboração durante a utilização dos micro-computadores.
- À EMBRAPA, Empresa Brasileira de Pesquisa Agropecuária, da qual fui bolsista de 1981 até 1984, pelo apoio financeiro e pelo excelente ambiente de trabalho durante a etapa de redação desta tese.

Resumo da Tese Apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

CONTRIBUIÇÃO PARA A RESOLUÇÃO DO PROBLEMA DE STEINER NUM GRAFO DIRECIONADO: UM MÉTODO HEURÍSTICO

Oscar Iván Palma Pacheco

Junho de 1985

Orientador: Nelson Maculan Filho

Programa: Engenharia de Sistemas e Computação

No presente trabalho desenvolvemos um método heurístico para resolver o problema de Steiner num grafo direcionado. Para isto, consideramos o problema de Steiner como sendo um problema de arborescência geradora e de fluxo com custos fixos. Considerando uma definição particular de peso de uma arborescência, formulamos o problema como sendo um problema de arborescência geradora de peso mínimo, cuja solução contém a solução para o problema de Steiner.

O método heurístico, desenvolvido para resolver este problema, foi implementado, em FORTRAN IV, em duas versões: na primeira considera-se qualquer arborescência geradora inicial e na segunda utiliza-se uma arborescência geradora obtida a partir da solução dada pela heurística de Richard T. Wong, também implemetada. O método heurístico mostrou-se sensível à arborescência geradora inicial tanto no tempo empregado como na solução obtida. Os resultados obtidos foram considerados muito bons, especialmente pelas perspectivas que existem de se melhorar ainda mais o desempenho do método.

Abstract of Thesis presented to COPPE/UFRJ as partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

CONTRIBUTION TO SOLVE THE STEINER PROBLEM IN A DIRECTED GRAPH: A HEURISTIC METHOD

Oscar Iván Palma Pacheco

June, 1985

Chairman: Nelson Maculan Filho

Department: System Engineering and Computation

In this work we develop an heuristic method to solve the Steiner problem in a directed graph. To do this, the Steiner problem is considered as a spanning arborescence flow problem with fixed costs. Considering a proper definition of weight of an arborescence, the problem is formulated as a minimal spanning arborescence problem whose solution provides a solution for the Steiner problem.

The heuristic method, developed for solve this problem, has been implemented, in FORTRAN IV, in two versions: the first considers any initial spanning arborescence and the second uses a spanning arborescence provided by the Wong's heuristic method, also implemented. The heuristic method is sensitive to the initial spanning arborescence as well in relation to processing time as to the value of the aproximated solution. The results were judged as good, specialy considering other possible improvements of the method.

ÍNDICE

APÊNDICE C

APÊNDICE D

CAPÍTULO I

INTRODUÇÃO

No início de 1984 retornava ao Pais, depois de passar dois anos no Canadá, o Dr. Nelson Maculan Filho, quem fora o meu primeiro orientador no Programa de Doutorado em Engenharia de Sistemas e Computação na COPPE/UFRJ. Foi nesse momento que surgiu a possibilidade de, orientado novamente por êle, desenvolver algum método que permitisse obter soluções aproximadas para o Problema de Steiner, num grafo direcionado, em tempo computacional aceitável.

Assim, em Maio de 1984, começavam os primeiros testes para determinar a melhor forma de atacar o problema. No início, a idéia era dar continuidade ao trabalho desenvolvido por Claus e Maculan [5], no sentido de desenvolver uma boa implementação do método simplex orientado para resolver, especificamente, o problema de Steiner num grafo direcionado, trabalho já iniciado para grafos de pequeno e médio porte por Arpin, Maculan e Nguyen [2], aproveitando ao máximo as características da formulação proposta por eles: baixa densidade, diagonal de submatrizes, variáveis limitadas por variáveis ("variable upper bound"). Em seguida, tentou-se fazer uma analogia com o problema de transportes com transbordo, com algumas adaptações na escolha dos arcos para sempre passar de uma base a outra melhor.

Finalmente, chegou-se à conlusão de que era mais conveniente tentar descubrir arcos, ou grupo de arcos, que, de acordo com as estruturas de arcos das arborescências geradoras, que contêm subarborescências de Steiner, pudessem nos levar à obtenção de uma arborescêcia geradora de peso menor. Foi segundo esta orientação que desenvolvemos o presente trabalho.

No Capítulo II, como uma introdução ao Problema de Steiner, apresenta-se a sua definição original, no plano euclidiano, e a sua posterior adaptação em grafos, especialmente num grafo direcionado ao qual se refere este trabalho. Neste caso, apresentam-se comentários a respeito da literatura existente, com mais detalhes nos trabalhos considerados de maior importância.

No Capítulo III aparecem o desenvolvimento e justificativas do método proposto e no Capítulo IV apresenta-se a experiência computacional, cujos resultados foram satisfatórios.

No Capítulo V, considerando os resultados obtidos na experiência computacional, e imaginando novas estruturas de arcos nas arborescências geradoras, apresentamos as conlusões além de indicar onde é conveniente concentrar as pesquisas para melhorar ainda mais este método.

CAPITULO II

O PROBLEMA DE STEINER

2.1 - O PROBLEMA DE STEINER NO PLANO EUCLIDIANO

O problema de Steiner é um problema geométrico já antigo e foi definido, originalmente, sobre o plano euclidiano, no qual alguns pontos predeterminados deviam conectar-se mediante linhas formando uma árvore de comprimento mínimo, sendo permitida a interseção de linhas em qualquer ponto do plano. Nesta árvore, o comprimento de cada uma das linhas é igual à distância euclidiana entre os pontos extremos delas, e os pontos de interseção, utilizados para obter uma árvore de comprimento mínimo, são chamados de Pontos de Steiner.

Para três pontos quaisquer, o problema de Steiner está resolvido, sendo necessário um ponto de Steiner na árvore caso, no triângulo formado pelos três pontos, todos os ângulos sejam menores que 120 graus. Se um ângulo for maior ou igual que 120 graus, então nenhum ponto de Steiner é utilizado [8].

2.2 - O PROBLEMA DE STEINER NUM GRAFO

Do plano euclidiano passamos para um grafo no qual, no lugar de pontos, existem vértices e entre alguns pares de vértices existem arcos com pesos arbitrários que não correspondem à distância euclidiana. Alguns dos vértices do grafo, chamados **vértices obrigatórios**, devem ser conectados, utilizando arcos do grafo, para formar uma árvore de peso mínimo. Se for necessário, pode utilizar também alguns dos vértices optativos como sendo os **vértices de Steiner**.

Lawler [8] acha que este problema é mais fácil de resolver já que, em principio, poderia ser resolvido por simples enumeração. Lawler apresenta dois algoritmos para resolver o problema de Steiner num grafo não direcionado: o primeiro, resolve o problema de árvore geradora de peso mínimo para conjuntos de vértices formados pelos n vértices obrigatórios além de, no máximo, n-2 vértices optativos, para todas as possíveis combinações de vértices de Steiner; o segundo, devido a Dreyfus e Wagner [6], utiliza uma relação de recursividade no sentido de, se obtivermos árvores de Steiner ótimas para subconjuntos com 1,2,...,p-1 vértices, podemos construir árvores de Steiner ótimas para subconjuntos com p vértices.

Outros autores têm formulado modelos de programação inteira, com variáveis zero-um, e tentado a sua resolução utilizando métodos "branch-and-bound" ou decomposição de Benders.

Aneja [1] formula o problema como sendo um problema de recobrimento e, para contornar o problema derivado do crescimento exponencial do número de restrições, desenvolve um procedimento para gerar as restrições, mantendo as demais restrições em forma implícita. Também desenvolveu um procedimento heurístico para acelerar a convergência. Os resultados obtidos não foram muito alentadores.

Hakimi [7] define o problema de Steiner num grafo, relacionando-o com alguns problemas de recobrimento em grafos e, finalmente, apresenta um algoritmo para encontrar uma árvore de Steiner com uma topologia dada. Para resolver este problema, com este algoritmo, seria necessário considerar todas as topologias possíveis, o que daria um enorme trabalho. Hakimi [7], assim como Dreyfus e Wagner [6], sugere o emprego de programação dinâmica.

2.3 - O PROBLEMA DE STEINER NUM GRAFO DIRECIONADO

O problema que, em última instância, será o objetivo deste trabalho, é o problema de Steiner num grafo direcionado. Este problema é considerado mais geral que o problema de Steiner num grafo não direcionado, já que um grafo não direcionado, mediante a substituição de cada arco não direcionado por dois arcos direcionados de igual peso, pode ser transformado num grafo direcionado [10]. Neste caso, devemos considerar mais um elemento: a raiz do grafo.

Definição: Raiz do grafo G é um vértice, do próprio grafo, que alcança todos os demais.

Para efeitos deste trabalho, vamos supor que a determinação do vértice raiz do grafo é parte da formulação do problema.

Relacionado com o problema de Steiner num grafo direcionado, existem dois trabalhos, desenvolvidos recentemente, que merecem ser destacados: os trabalhos de Claus e Maculan [5] e de Wong [10].

2.3.1 - DEFINIÇÃO DO PROBLEMA

Seja G=(N,A) um grafo direcionado conexo onde N é o conjunto de vértices e A é o conjunto de arcos. Cada arco (i,j) em A tem associado um peso C(i,j) não negativo. O conjunto N de vértices está dividido em dois subconjuntos disjuntos V e S, sendo V o conjunto de vértices obrigatórios e S o conjunto de vértices optativos ou de Steiner. Seja r em V o vértive raiz do grafo.

O problema de Steiner, sobre um grafo direcionado, consiste em determinar uma arborescência de peso mínimo, com raiz em r, na qual r alcance todos os vértices obrigatórios, podendo utilizar os vértices optativos que forem necessários como sendo os pontos de Steiner.

2.3.2 - FORMULAÇÃO DE CLAUS E MACULAN

Armin Claus e Nelson Maculan [5] propõem uma nova formulação para o problema de Steiner num grafo direcionado, visando a sua resolução mediante uma versão primal do método simplex. Para isto, apresentaram um modelo de programação linear inteira, onde consideram o problema de Steiner como sendo um problema de fluxo de custo mínimo, no qual a raiz r envia uma unidade de fluxo a cada um dos demais vértices obrigatórios e, por sua vez, cada vértice obrigatório demanda exatamente uma unidade. O fluxo que passa através de cada arco é controlado individualmente para cada vértice obrigatório, ou seja, a cada arco é associada uma variável de fluxo para cada vértice obrigatório. Por exemplo, a variável Xijk representa o fluxo que passa pelo arco (i,j) e que está endereçado ao vértice obrigatório k.

Nesta formulação, Claus e Maculan definem, para cada vértice obrigatório k em V-{r}, um conjunto de restrições de balanço, do fluxo orientado para o vértice k, em cada um dos vértices do grafo. No vértice r, a diferença entre o fluxo que sai e o fluxo que chega deve ser igual a 1. No vértice k, esta diferença deve ser igual a -1. Nos demais vértices, a diferença deve ser zero.

Por outro lado, em cada arco, o peso é fixo, ou seja, se o arco é utilizado, o peso dele na arborescência independe da quantidade de fluxo que passa através dele. Por este motivo, o modelo utiliza variáveis zero-um, o que implica na necessidade de empregar métodos que considerem este tipo de variáveis.

Ao controlarem o fluxo em cada arco, individualmente para cada vértice obrigatório, estão garantindo que, por cada arco, em direção a cada vértice obrigatório, passe um fluxo de, no máximo, uma unidade. Sendo assim, as variáveis zero-um, que controlam o uso ou não de cada arco, servem para limitar o valor do fluxo que pode passar, por cada arco, em direção a cada vértice. Ou seja, se o valor da variável zero-um, associada a um determinado arco, tiver o valor zero, significa que o arco não pode ser utilizado e, portanto, todos os fluxos que passem por aquele arco, em direção a qualquer vértice obrigatório, deverão ser zero. Caso contrário, se tal variável tiver o valor um, o arco pode ser utilizado para transportar fluxo endereçado a qualquer vértice obrigatório e, como o fluxo que passa por cada arco, para cada vértice obrigatório, é de, no máximo, uma unidade, podemos limitar o fluxo pelo valor da variável zero-um.

Mas, o objetivo deles não era resolver este problema diretamente, que além de ter variáveis zero-um, o que já dificulta a sua resolução, o tamanho da matriz de coeficientes cresce muito rapidamente, impedindo a resolução de problemas grandes. Por exemplo, para um problema com 100 arcos e 50 vértices, dos quais apenas 10 obrigatórios, que podemos considerar como sendo um problema pequeno, seria necessário utilizar uma matriz com mais de um milhão de elementos, dos quais aproximadamente 3600 são diferentes de zero.

Para resolver este problema, optaram pelo relaxamento do modelo, substituindo todas as restrições referentes às variáveis zero-um por restrições de não negatividade. Com isto, eles conseguiram utilizar um método simplex padrão para resolver problemas pequenos e obtiveram resultados alentadores. Apesar de a matriz de coeficientes não ser totalmente unimodular, os problemas resolvidos sempre apresentaram soluções inteiras.

É claro que, nesta formulação, ainda é possível explorar a estrutura do problema e desenvolver um método simplex apropriado que considere, por exemplo, a baixa densidade da matriz de coeficientes e restrições do tipo "variável limitada por outra variável" (variable upper bound) [9], para tratar das restrições dos fluxos que, originalmente, estavam limitados pelo valor das variáveis zero-um, e que agora estão limitados por variáveis contínuas. Uma alternativa seria utilizar como base o trabalho de Ruy Campello [4], que trata do desenvolvimento e implementação de um programa para programação linear de médio porte, o qual está orientado para resolver problemas com matriz de coeficientes de baixa densidade, e incorporar nele a possibilidade de considerar variáveis limitadas por outras variáveis.

2.3.3 - MÉTODO DUAL ASCENDENTE E HEURÍSTICA DE WONG

Richard T. Wong [10] apresenta um método para obter limites inferiores para o valor da solução ótima do problema de Steiner num grafo direcionado e, a partir das informações geradas por este método, determinar, heuristicamente, uma solução aproximada para o problema. O método Dual Ascendente gera um grafo auxiliar G' o qual será formado por um subconjunto dos arcos do grafo original.

Definição: Corte de um vértice k qualquer é o conjunto de arcos cujos membros (i,j) satizfazem (1) (i,j) pertence ao grafo original; (2) (i,j) não pertence ao grafo auxiliar; (3) j alcança o vértice k no grafo auxiliar e i não alcança o vértice k no grafo auxiliar.

Basicamente, o Método Dual Ascendente pode ser resumido no algoritmo descrito a seguir. (Veja exemplo no Apêndice D.2)

ALGORITMO DUAL ASCENDENTE

Passo 0: Fazer S(i,j) = C(i,j) para todo arco (i,j). Formar o grafo auxiliar G' = (N,A') com A' = Ø. Fazer LI = Ø.

Passo 1: Selecionar, em G', uma componente fortemente conexa, que contenha algum vértice obrigatório, e que não seja alcançada por nenhum vértice obrigatório nem pela raiz. Se tal componente não existe, fim, e o valor de LI é o limite inferior para o valor da solução ótima do problema de Steiner.

Passo 2: Selecionar um vértice k, obrigatório, na componente. Seja S(i*,j*) = min{S(i,j)|(i,j) pertence ao corte de k}. Fazer LI = LI + S(i*,j*). Para cada arco (i,j) no corte de k, fazer S(i,j) = S(i,j) - S(i*,j*).

Passo 3: Atualizar o grafo auxiliar incluindo o arco (i*,j*). Retornar ao passo 1. **#####**

O método dual ascendente nos dá um limite inferior para o valor da solução ótima do problema de Steiner. Wong também descreve a seguinte heurística para encontrar uma sulução aproximada para o problema de Steiner.

(1) Utilizar o Método Dual Ascendente.

(2) Considerar o grafo auxiliar G' ao término do Método Dual Ascendente e determinar o conjunto Q de todos os vértices alcançados pela raiz.

(3) Construir uma arborescência geradora de peso mínimo com os vértices do conjunto Q, e com todos os arcos do grafo original entre vértices do conjunto Q. Para isto, utilizar o próprio Método Dual Ascendente, o qual, neste caso, permite obter uma solução ótima.

(4) Finalmente, da arborescñcia geradora, eliminar todas as folhas correspondentes a vértices optativos, e os arcos incidentes a eles, até que as folhas sejam todas vértices obrigatórios.

O procedimento aqui descrito permite determinar uma arborescência de Steiner que é uma solução aproximada para o problema de Steiner num grafo direcionado. (Veja exemplo no Apêndice D.3)

CAPÍTULO III

UMA NOVA HEURÍSTICA PARA O PROBLEMA DE STEINER

3.1 - INTRODUÇÃO

Assim como o problema de Steiner pode ser formulado como um problema de fluxo, onde os custos (pesos) são fixos, ele também pode ser observado como um problema de arborescência geradora para um determinado conjunto de vértices formado por todos os vértices obrigatórios e, normalmente, por um subconjunto dos vértices optativos. Se fosse possível determinar o conjunto de vértices da solução ótima do problema de Steiner, seria relativamente fácil obter a solução dele utilizando algum algoritmo específico para contruir uma arborescência geradora. O método proposto neste trabalho baséia-se nestas duas orientações: fluxo com custos fixos e arborescência geradora.

Como o problema de Steiner num grafo não direcionado é NP-Hard, Wong [10] conclui que o problema de Steiner num grafo direcionado também é NP-Hard, motivo pelo qual achamos conveniente desenvolver o método heurístico descrito neste capítulo.

3.2 - O PROBLEMA DE STEINER COMO UM PROBLEMA DE ARBORESCÊNCIA GERADORA E FLUXO COM CUSTOS FIXOS

A solução ótima para o problema de Steiner será, invariavelmente, uma arborescência de peso mínimo na qual todas as folhas correspondam a vértices obrigatórios. Para efeitos deste trabalho, utilizaremos a seguinte definição de arborescência de Steiner.

Definição: Subarborescência de Steiner, num grafo direcionado G, é toda arborescência geradora T=(Nt,At) tal que $Nt \subset N$, $V \subset Nt$ e $At \subset A$, cujas folhas são todas vértices obrigatórios.

Proposição 1: Toda arborescência geradora Tg contém uma única subarborescência de Steiner. **#####**

Observação: Como a raiz envia uma unidade de fluxo para cada vértice obrigatório, e como cada vértice obrigatório absorve exatamente uma unidade de fluxo, podemos concluir que o fluxo que passa através de um arco qualquer, numa arborescência geradora Tg, é igual ao número de vértices obrigatórios que alcança.

Definição: F(i,j) representa o fluxo que passa através do arco (i,j).

Proposição 2: Numa arborescência geradora Tg, todos os arcos da subarborescência de Steiner têm fluxo positivo e os restantes, se houver, tem fluxo zero. **#####**

Como temos que encontrar uma arborescência de Steiner de peso mínimo, podemos resolver o problema procurando uma arborescência geradora que contenha uma subarborescência de Steiner de peso mínimo. Como, por outro lado, numa arborescência geradora, somente os arcos da subarborescência de

Steiner tem fluxo positivo, podemos tentar resolver o problema procurando uma arborescência geradora de peso mínimo, onde são considerados apenas os pesos dos arcos com fluxo positivo.

Definição: O peso de uma arborescência geradora Tg é igual ao peso da subarborescência de Steiner contida nela, ou seja, é igual à soma dos pesos dos arcos que têm fluxo positivo.

Esta última definição baseia-se no fato de estarmos apenas interessados nos arcos que alcançam vértices obrigatórios.

Assim, podemos tentar resolver o problema de Steiner procurando uma arborescência geradora de peso mínimo, de acordo com a definição de peso dada anteriormente.

Proposição 3: Considerando um grafo direcionado G, a subarborescência de Steiner contida numa arborescência geradora de peso mínimo é uma solução ótima para o problema de Steiner.

Demonstração: Seja r o vértice raiz do grafo G. Seja Ts uma solução ótima para o problema de Steiner. Então Ts é uma arborescência de Steiner e, portanto, contém todos os vértices obrigatórios e, possívelmente, alguns vértices optativos. Como r alcança todo vértice i em N, existem arcos em A que, se acrescentados à arborescência de Steiner, permitem transformá-la numa arborescência geradora do grafo G. Como também, ao incluir os arcos, só são incorporados vértices optativos, o peso da arborescência geradora é o mesmo que o da arborescência de Steiner Ts, já que os arcos incluidos só arcançam vértices optativos e, portanto, têm fluxo zero. Ao incrementarmos a arborescência de Steiner, da forma descrita anteriormente, estamos obtendo uma arborescência geradora na qual a arborescência de Steiner Ts é uma subarborescência. Por outro lado, se de uma arborescência geradora retiramos todas as folhas correspondentes a vértices optativos, e os respectivos arcos incidentes a eles, estaremos gerando uma arborescência de Steiner com o mesmo peso que a arborescência geradora. Portanto, a arborescência geradora de peso mínimo, ou seja, de peso igual ao da arborescência Ts, contém uma subarborescência de Steiner de peso mínimo, a qual resolve o problema. #####

Proposição 4: Numa arborescência geradora, o fluxo associado a um arco incidente a um vértice x é igual à soma dos fluxos associados aos arcos iniciados no vértice x, se x for um vértice optativo, ou é igual à mesma soma, acrescida de uma unidade, se x for um vértice obrigatório. #####

Observando os fluxos dos arcos, numa arborescência geradora, verifica-se que existem caminhos de arcos de igual fluxo. Então, baseados neste fato, podemos fragmentar uma arborescência geradora.

Definição: Fragmento de uma arborescência geradora é todo caminho de arcos de igual fluxo, tal que os arcos anteriores e posteriores ao caminho tem fluxo diferente ao dele.

Desta forma, fica clara a distribuição dos arcos pelos diversos fragmentos. No que diz respeito aos vértices, utilizaremos a seguinte definição.

Observação: Um vértice qualquer, numa arborescência geradora, pertence ao mesmo fragmento que o arco incidente a ele.

Esta observação descarta a possibilidade de um vértice, ao qual chega um fluxo positivo, pertencer a mais de um fragmento, como seria o caso dos vértices inicial e final dos fragmentos.

Na figura (III.1) temos uma arborescência onde os valores

```
                              ___
                             | 1 |
                              ---
                               |
                          3/3  |
                               V
                              ___
                             | 2 |
                              ---
                               |
                          3/7  |
                               V
                              ___
              _______________| 3 |_______________
              |               ---               |
         0/2  |                                 |  3/4
              V                                 V
             ___                               ___
      ______| 4 |______                 ______| 5 |______
      |      ---      |                 |      ---      |
 0/2  |               |  0/3       1/1  |               |  1/3
      V               V                 V               V
     ___             ___               ___             ___
    | 6 |           | 7 |             | 8 |           | 9 |
     ---             ---               ---             ---
```

Fig. III.1

ao lado de cada arco representam, respectivamente, o fluxo e o peso. Considerando estes fluxos, e a definição de fragmento dada anteriormente, podemos identificar os seguintes fragmentos:

1 - 2 - 3 - 5

3 - 4 - 6

3 - 4 - 7

5 - 8

5 - 9

Proposição 5: Todo arco de uma arborescência geradora, pelo qual esteja passando um fluxo positivo, pertence a um único fragmento.

Demonstração: Por contradição, suponhamos que um arco, pelo qual passa um fluxo positivo, pertença a dois fragmentos. Então, os fragmentos aos quais o arco pertence tem o mesmo fluxo. Seja (i,j) tal arco. Existem duas possibilidades:

a) existem dois arcos de igual fluxo, cada um deles incidente ao vértice i, o qual contradiz a definição de arborescência; e

b) existem dois arcos, ambos com o mesmo fluxo positivo do arco (i,j), saindo do vértice j, o que contradiz a definição de fluxo.

Portanto, um arco com fluxo positivo não pode pertencer, simultaneamente, a mais de um fragmento. **#####**

Os arcos com fluxo zero podem pertencer a mais de um fragmento já que do seu vértice final podem nascer vários arcos de igual fluxo zero.

Definição: Segmento de um vértice v qualquer, numa arborescência geradora, é o caminho, não vazio, de arcos, que começa no início do fragmento, ao qual v pertence, e termina no vértice v.

Na figura (III.1), encontramos os seguintes segmentos:

do vértice 1: não tem,

do vértice 2: 1 - 2,

do vértice 3: 1 - 2 - 3,

do vértice 4: 3 - 4,

do vértice 5: 1 - 2 - 3 - 5,

do vértice 6: 3 - 4 - 6,

do vértice 7: 3 - 4 - 7,

do vértice 8: 5 - 8, e

do vértice 9: 5 - 9.

Definição: PP(v) é o vértice inicial do segmento de v, para todo v em N, $v \neq r$. Para a raiz, PP(r) = r.

Definição: CS(v) é o peso acumulado desde a raiz r até o vértice v, independente da existência ou não de fluxo.

Desta forma, o peso CS(r), acumulado até a raiz, é igual a zero (CS(r) = 0).

Na figura (III.1), os valores de PP e CS, para cada um dos vértices, são os seguintes:

PP(1) = 1, CS(1) = 0,

PP(2) = 1, CS(2) = 3,

PP(3) = 1, CS(3) = 10,

PP(4) = 3, CS(4) = 12,

PP(5) = 1, CS(5) = 14,

PP(6) = 3, CS(6) = 14,

PP(7) = 3, CS(7) = 15,

PP(8) = 5, CS(8) = 15,

PP(9) = 5, CS(9) = 17.

Definição: O peso de um caminho de arcos x-y, numa arborescência geradora, é igual à soma dos pesos dos arcos compreendidos entre os vértices x e y, e corresponde ao valor da expressão

CS(y) - CS(x).

Proposição 6: O peso do segmento de um vértice x qualquer, numa arborescência geradora, é igual à soma dos pesos dos arcos compreendidos entre os vértices PP(x) e x, e corresponde ao valor da expressão

CS(x) - CS(PP(x)). **#####**

Por exemplo, de acordo com a figura (III.1), os pesos dos segmentos dos vértices são os seguintes:

para o vértice 1: CS(1) - CS(1) = 0 - 0 = 0,

para o vértice 2: CS(2) - CS(1) = 3 - 0 = 3,

para o vértice 3: CS(3) - CS(1) = 10 - 0 = 10,

para o vértice 4: CS(4) - CS(3) = 12 - 10 = 2, para o vértice 5: CS(5) - CS(1) = 14 - 0 = 14,

para o vértice 6: CS(6) - CS(3) = 14 - 10 = 4,

para o vértice 7: CS(7) - CS(3) = 15 - 10 = 5,

para o vértice 8: CS(8) - CS(5) = 15 - 14 = 1, e

para o vértice 9: CS(9) - CS(5) = 17 - 14 = 3.

Proposição 7: Se o vértice x for o último vértice de um fragmento qualquer, numa arborescência geradora, então o fragmento PP(x)-x coincide com o segmento do vértice x. #####

3.3 - DESCRIÇÃO GERAL DO MÉTODO

O método proposto neste trabalho considera, em todo instante, a existência de uma arborescência geradora. Esta arborescência geradora será modificada em cada iteração, visando a obtenção de outra arborescência geradora, de menor peso, se possível, ou de igual peso, se isto pode fazer com que, em iterações posteriores, seja conseguida uma arborescência geradora de menor peso.

Em linhas gerais, o método pode ser resumido no algoritmo descrito a seguir. (Veja exemplo no Apêndice D.4)

ALGORITMO GERAL

Passo 1: Começar com uma arborescência geradora qualquer. Inicializar PP e CS.

Passo 2: Determinar o arco, ou lista de arcos, que deva ser incorporado à arborescência geradora, em substituição a outro tanto, para que possa ser obtida uma outra arborescência geradora, de peso menor ou igual que o da arborescência geradora anterior, de acordo com o critério mencionado anteriormente. Se não existe, FIM, e a última arborescência geradora contém uma subarborescência de Steiner que é uma solução aproximada para o problema de Steiner num grafo direcionado. Caso contrário, vá ao passo 3.

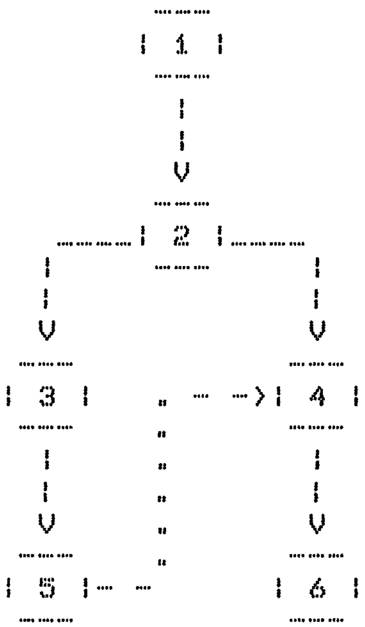
Passo 3: Modificar a arborescência geradora e retornar ao passo 2. **#####**

Obter uma arborecência geradora inicial, a ser utilizada neste método, não apresenta dificuldades já que estamos pensando em qualquer uma. Podemos considerar, inicialmente, uma arborescência geradora obtida da forma mais rápida possível ou mediante uma outra heurística (por exemplo, a heurística de Wong). Nossa maior preocupação será detetar arcos que, individualmente ou em conjunto, contribuam para obter arborescências geradoras de menor peso.

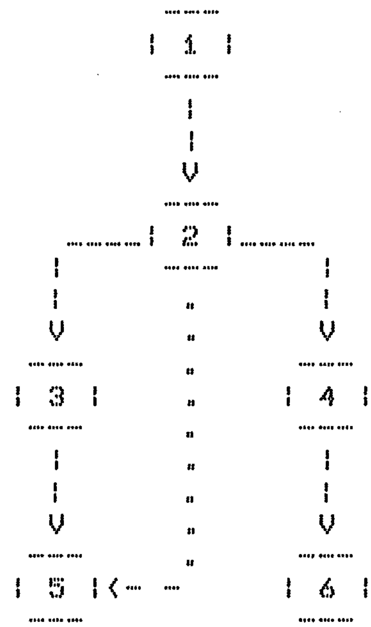
Para determinar se um arco, ou lista de arcos, pode ser incluido numa arborescência geradora, em substituição a outro tanto, tentaremos verificar, antecipadamente, o efeito de tais alterações numa arborescência.

3.4 - ATUALIZAÇÃO DE UMA ARBORESCÊNCIA GERADORA

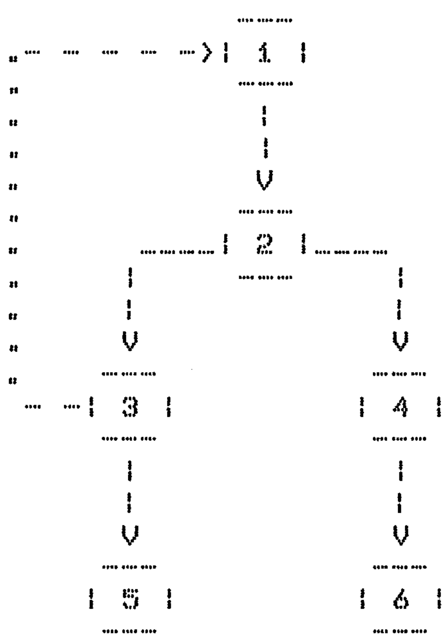
A atualização de uma arborescência geradora dar-se-á toda vez que for indicado um arco, ou lista de arcos, para substituirem outros já existentes nela. Para que seja executado o procedimento de substituição de arcos, é necessário apenas preparar uma lista contendo os arcos a serem incluidos na arborescência, já que os arcos a serem excluidos ficam automaticamente definidos. Mesmo que a lista contenha mais de um arco, as substuições são processadas uma de cada vez, sendo



Caso a



Caso b



Caso c

Fig. III.2

necessário que, após cada substituição, seja obtida uma outra arborescência geradora. Para isto, basta verificar que o arco a ser incluido não forme ciclo com os demais arcos da arborescência.

Na figura (III.2) podemos apreciar exemplos de arcos a serem incluidos numa arborescência, sendo válidos os casos "a" e "b". O caso "c" não é válido já que o arco (3,1) forma ciclo com os arcos (1,2) e (2,3).

Proposição 8: A inclusão de um arco (i,j), numa arborescência geradora, forma ciclo se, na arborescência, o vértice j alcança o vértice i. #####

O fato de não serem aceitos arcos que formam ciclo não implica que tais arcos estejam impedidos de substituir outros na arborescência. Apenas significa que, se for conveniente incluir um arco que forma ciclo, ele deverá vir acompanhado do correspondente arco que provoca a destruição do ciclo, sem formar outro, sendo este processado em primeiro lugar.

Como foi dito anteriormente, para cada arco e vértice de uma arborescência geradora, existem algumas informações que, devido à substitujão de arcos, deverão ser atualizadas.

Na figura (III.3), ao lado de cada arco, está o valor do fluxo (número de vértices obrigatórios que alcança). Ao incluir o arco (7,10) deve ser excluido o único arco que, na arborescência, é incidente ao vértice 10, neste caso, o arco (9,10), e, devido a isto, modificam-se os fluxos de alguns arcos já que, com o desvio do fluxo que passava pelo arco (9,10) para o arco (7,10), alteram-se os fluxos dos arcos anteriores a eles. Neste caso em particular, após a substituição, os vértices obrigatórios que eram alcançados pelo arco (9,10), e também pelos anteriores a ele, agora são alcançados pelo arco (7,10), e pelos seus anteriores. Isto implica que, o fluxo que passava pelos arcos (3,8),(8,9) e (9,10), em direção aos vértices obrigatórios alcançados pelo arco (9,10), agora passará pelos arcos (3,4), (4,5), (5,6), (6,7) e (7,10). Como há uma alteração nos vértices obrigatórios que alguns arcos alcançam, nesses arcos o fluxo se altera de acordo com a quantidade de vértices obrigatórios que

```
                     ---
                    | 1 |
                     ---
                      |
                    5 |
                      V
                     ---
                    | 2 |
                     ---
                      |
                    5 |
                      V
                     ---
          ----------| 3 |-----------
          |          ---           |
        1 |                        | 3
          V                        V
         ---                      ---
        | 4 |                    | 8 |
         ---                      ---
          |                        |
        1 |                        | 3
          V                        V
         ---                      ---
        | 5 |               -----| 9 |-----
         ---                |     ---     |
          |                 |             |
        1 |               2 |             | 1
          V                 V             V
         ---               ---           ---
        | 6 |      . - - ->| 10|         | 12|
         ---       .        ---           ---
          |        .         |
        0 |        .       2 |
          V        .         V
         ---       .        ---
        | 7 | - - -        | 11|
         ---                ---
```

Fig. III.3

o arco (9,10) alcançava, e o arco (7,10) passa a alcançar os mesmos vérices que o arco (9,10) alcançava.

Na figura (III.4) pode ser observada a arborescência geradora obtida após a substituição do arco (9,10) pelo arco (7,10). Os valores de CS e PP, correspondentes aos vértices posteriores aos vértices 9 e 10, também serão modificados. O peso da nova arborescência é igual ao peso da arborescência

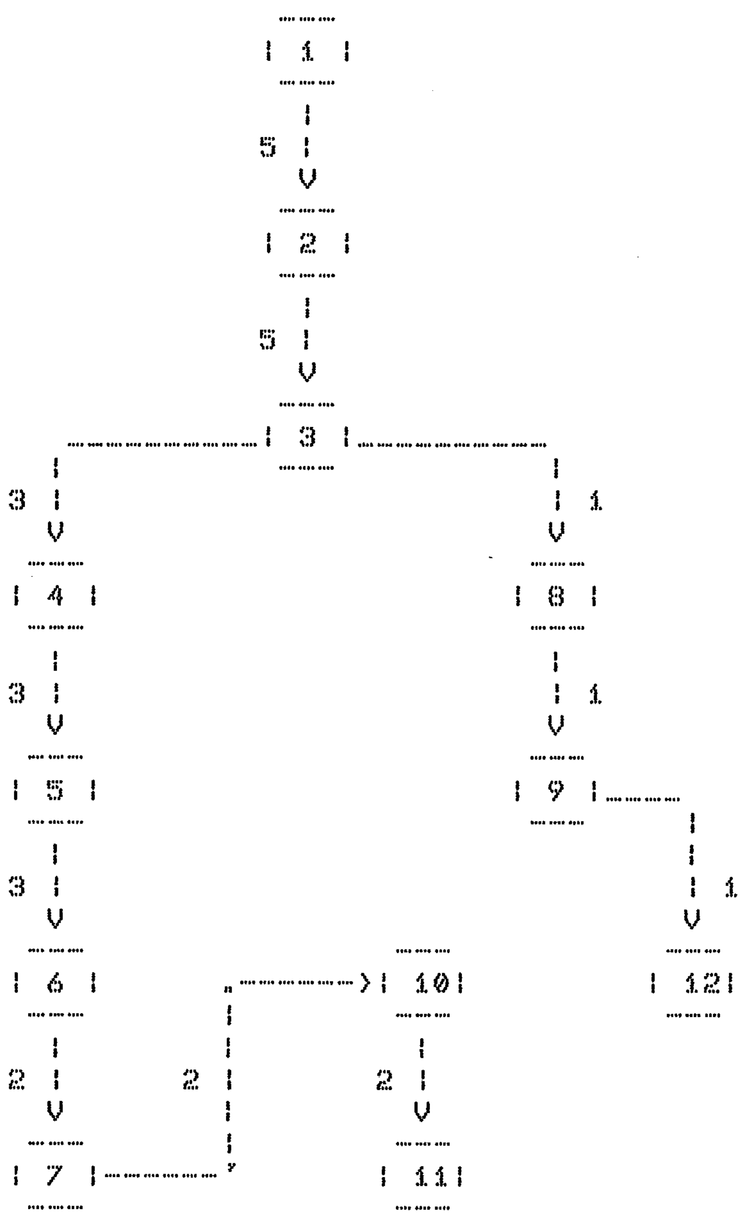


Fig. III.4

antes de qualquer alteração, mais o peso dos arcos (6,7) e (7,10), e menos o peso do arco (9,10).

Em geral, a atualização dos fluxos deve ser feita nos arcos precedentes ao arco a ser incluido e ao arco a ser excluido, até atingir a raiz da atualização.

Definição: Seja (i,j) o arco a ser incluido numa arborescência geradora qualquer. Vértice raiz da atualização é o vértice mais distante da raiz da arborescência que alcança, simultaneamente, os vértices i e j.

Por outro lado, o peso da nova arborescência geradora será igual ao peso da arborescência anterior, mais o peso do arco que foi incluido, menos o peso do arco excluido, mais os pesos dos arcos que passaram de fluxo zero para fluxo positivo, e menos os pesos dos arcos que alteraram o fluxo de positivo para zero. Na figura (III.4), os únicos arcos que alteraram o fluxo foram (3,4), (4,5), (5,6), (6,7), (3,8) e (8,9). A substituição deu-se entre os arcos (7,10) e (9,10). O vértice 3 é considerado o vértice raiz da atualização, a partir do qual podem ser atualizados o fluxo e os valores de CS e PP.

Na figura (III.5) estão representados os vértices i e j, do arco (i,j) a ser incluido, e o vértice a, raiz da atualização.

Recuando, antes da substituição, arco a arco, desde os vértices i e j, o vértice a, raiz da atualização, é o primeiro vértice que alcança, simultaneamente, i e j. Em outras palavras, podemos afirmar que, de todos os vértices que alcançam, simultaneamente, i e j, o vértice a é o que tem o maior valor de CS, o maior peso acumulado, e, portanto, é o que está mais distante da raiz da arborescência. Ao incluir o arco (i,j) devemos, obrigatoriamente, excluir o arco (k,j), o único arco incidente ao vértice j.

Para atualizar os fluxos dos arcos, não é necessário conhecer, antecipadamente, o vértice raiz da atualização, já que podemos fazer as alterações nos fluxos dos arcos, um de cada vez, no sentido contrário ao do fluxo, até atingirmos o vértice raiz da atualização. Existem dois caminhos a serem atualizados: o primeiro, alcança o arco a ser incluido, e o

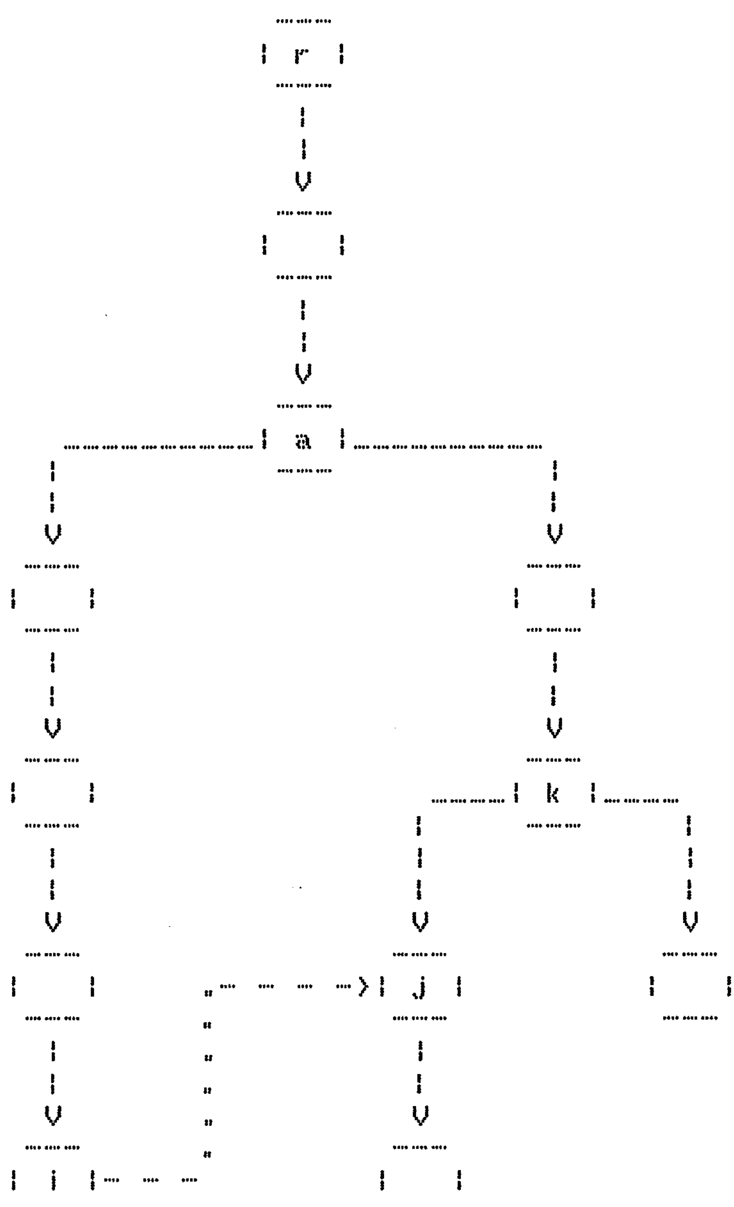


Fig. III.5

segundo, o arco a ser excluido. Recua-se, desde os vértices iniciais destes arcos, um arco de cada vez, considerando o vértice que estiver mais distante da origem (maior valor de CS), até que, nos dois caminhos o vértice considerado seja o mesmo.

Normalmente, após a atualização dos fluxos, podemos substituir o arco a ser excluido e, então, atualizar os valores de CS e PP para todos os vértices alcançados pela raiz da atualização. Mas, a atualização dos fluxos só será necessária quando o fluxo do arco a ser excluido for positivo. Não é necessário fazer a atualização dos valores de CS e PP a partir da raiz da atualização. Sendo (i,j) o arco incluido e (k,j) o arco excluido, é suficiente atualizar CS a partir do vértice j, e PP a partir dos vértices i e PP(j) (este último, obviamente, antes da atualização). Quando o fluxo do arco excluido for zero, só é preciso atualizar CS e PP a partir do vértice j.

Proposição 9: Toda vez que o arco excluido de uma arborescência geradora tiver fluxo zero, o peso da arborescência não será alterado. #####

Como numa arborescência, por definição, todo vértice têm exatamente um arco incidente, exceto a raiz, podemos descrever ela mediante os vértices iniciais dos arcos incidentes a cada vértice que não seja a raiz da arborescência.

Definição: PREC(v) é um vértice u tal que (u,v) é o único arco incidente ao vértice v na arborescência geradora.

ALGORITMO PARA ATUALIZAR UMA ARBORESCÊNCIA

Passo 0: Seja (i,j) o arco a ser incluido. Seja PESO o peso da arborescência geradora. Seja f = F(PREC(j),j) o fluxo do arco a ser substituido. Fazer p1 = PREC(j) e p2 = i.

Passo 1: Se $f > 0$, então fazer

$$PESO = PESO + C(i,j) - C(PREC(j),j).$$

Passo 2: Substituir, na arborescência, o arco (PREC(j),j) pelo arco (i,j) fazendo PREC(j)=i. Se $f = 0$, faça $x = i$ e vá ao passo 7.

Passo 3: Se $CS(p1) < CS(p2)$, vá ao passo 4. Se $CS(p1) > CS(p2)$, vá ao passo 5. Se $CS(p1) = CS(p2)$, vá ao passo 6.

Passo 4: Se $F(PREC(p2),p2) = 0$, então fazer

$$PESO = PESO + CS(p2) - CS(PREC(p2)).$$

Atualizar o fluxo do arco (PREC(p2),p2), fazendo

$$F(PREC(p2),p2) = F(PREC(p2),p2) + f.$$

Fazer $p2 = PREC(p2)$ e retornar ao passo 3.

Passo 5: Atualizar o fluxo do arco (PREC(p1),p1), fazendo

$$F(PREC(p1),p1) = F(PREC(p1),p1) - f.$$

Se, após a atualização, $F(PREC(p1),p1) = 0$, então fazer

$$PESO = PESO - CS(p1) + CS(PREC(p1)).$$

Fazer $p1 = PREC(p1)$ e retornar ao passo 3.

Passo 6: Se $p1 \neq p2$, retornar ao passo 4. Caso contrário, fazer $x = p1$, onde p1, neste caso, é a raiz da atualização.

Passo 7: Atualizar os valores de CS e PP a partir do vértice x.

#####

3.5 - ESCOLHA DOS ARCOS

A atualização descrita anteriormente deve ser feita considerando uma lista, contendo um ou mais arcos, escolhida de tal forma que, após as alterações necessárias, a nova arborescência geradora seja de peso inferior à anterior ou, se isto não for possível, seja de igual peso, desde que com isto melhorem as espectativas de, numa iteração posterior, obter uma arborescência geradora de peso menor.

Considerando o processo de atualização descrito anteriormente, podemos determinar com antecedência o efeito que um arco, ou lista de arcos, teria sobre o peso de uma arborescência geradora quando efetivada a inclusão.

Em outras palavras, quando um arco é incluido numa arborescência geradora, em substituição a outro, o fluxo pode ser mantido em todos os arcos, se o arco a ser excluido tem fluxo zero, ou o fluxo de alguns arcos pode ser alterado, caso o fluxo do arco a ser excluido for positivo. Isto acontece porque o fluxo do arco excluido da arborescência deve ser canalizado através de outros arcos e, desta forma, alimentar o fluxo do arco incluido. Portanto, o fluxo do arco excluido não é mais necessário e pode ser descontado dos arcos precedentes. Por outro lado, o arco incluido passa a demandar o fluxo antes demandado pelo arco excluido, o qual deve ser fornecido através dos arcos precedentes a ele. Neste processo, se (i,j) é o arco a ser incluido e (k,j) é o correspondente arco, com fluxo positivo, a ser excluido, o fluxo dos arcos do segmento do vértice j passa a ser zero e, portanto, o peso da arborescência é diminuido no valor correspondente à soma dos pesos destes arcos. Por outro lado, o peso será aumentado pela soma do peso do arco a ser incluido mais os pesos dos arcos do segmento do vértice i, se o fluxo nestes arcos for zero. Isto acontece porque estes arcos, se tiverem fluxo zero, deverão obrigatoriamente transportar um fluxo positivo para atender às necessidades do arco incluido.

Como no peso da arborescência só são considerados os arcos com fluxo positivo, devemos verificar quando o fluxo de um arco passa de zero para positivo, ou viceversa, para determinar corretamente a variação no peso da arborescência geradora. Como o fluxo de um arco é menor ou igual que o fluxo dos arcos precedentes, somente os arcos dos segmentos dos vértices i e j podem alterar o peso da arborecência, já que somente eles podem alternar o fluxo entre zero e positivo. Então, nesta fase, devemos identificar o vértice a partir do qual os arcos podem ter o seu fluxo alterado de zero para positivo e, também, o vértice a partir do qual os arcos podem ter o seu fluxo alterado de positivo para zero. Chamaremos estes vértices de pi e pj, respectivamente, e, em princípio, poderiam ser os prórios vértices iniciais dos segmentos de i e de j. Ou seja, pi = PP(i) e pj = PP(j). Mas, há uma excessão: quando, no segmento do vértice j, existe um arco incidente ao vértice i ou ao vértice PP(i), como pode ser observado na figura (III.6). Neste caso, pi = pj = k, onde k representa o vértice i ou o vértice PP(i), aquele que estiver no segmento de j. Quando o fluxo do segmento de i e o fluxo do segmento de j são ambos positivos, é impossível que PP(i) esteja no segmento de j, e quando são ambos iguais a zero, tanto faz considerar uma ou outra definição para pi e pj. Isto deve ser considerado na implementação para evitar cálculos desnecessários.

Proposição 10: Seja (i,j) o arco candidato a ser incluido na arborescência. Os vértices pi e pj, a partir dos quais o fluxo dos segmentos dos vértices i e j pode passar de zero para positivo, ou viceversa, são

$$pi = pj = i,$$

se o vértice i pertence ao segmento do vértice j, ou

$$pi = pj = PP(i),$$

se o vértice PP(i) pertence ao segmento do vértice j, ou pi = PP(i) e pj = PP(j),

em caso contrário. #####

Definição: Considerando o arco (i,j) e os vértices pi e pj acima, definimos a seguinte função

p(i,j) = (pi,pj).

Uma vez definidos os vértives pi e pj, fica fácil determinar o valor da alteração no peso da arborescência.

Observação: o valor da alteração no peso da arborescência geradora, devido a inclusão do arco (i,j), está dado pela

```
              ---
             PP(j)
              ---
               |
             5 |
               V
              ---
             |   |
              ---
               |
             5 |
               V
              ---
             |   |
              ---
               |
             5 |
               V
              ---
        -----PP(i)-----
        |     ---     |
        |             |
      0 |             | 5
        V             V
       ---           ---
      |   |         |   |
       ---           ---
        |             |
      0 |             | 5
        V             V
       ---           ---
      | i |- - - - >| j |
       ---           ---
        |             |
      0 |             | 5
```

Figura III.6

seguinte expressão:

$$ALTER = C(i,j) - CS(j) + CS(pj) + CC , \qquad (III-1)$$

onde

$$CC = \begin{cases} 0, \text{ se o fluxo do segmento de } i \text{ for positivo, ou} \\ CS(i) - CS(pi), \text{ caso contrário.} \end{cases}$$

e

$$(pi,pj) = p(i,j).$$

De acordo com a quantidade de arcos considerados nas listas, podemos ter listas simples ou compostos. Nas listas simples, apenas um arco é considerado. Nos listas compostas, mais de um.

3.5.1 - LISTA SIMPLES

A forma menos complicada de se melhorar o peso de uma arborescência geradora, é através de um único arco. Mas, melhorar não significa necessáriamente diminuir o peso dela, senão diminuir o peso ou mantê-lo se isto ajuda para abrir novas possibilidades de diminuir o peso posteriormente.

O peso de uma arborescência geradora não mudará se o arco a ser excluido tiver fluxo zero. Este será o único caso aceitável de não alteração do peso da arborescência. No caso de substituição de arcos com fluxo positivo, exigiremos que o peso da arborescência seja diminuido com a sua inclusão.

Para que um arco seja incluido numa lista simples, é necessário que ele não forme ciclo com os demais arcos da arborescência geradora, já que numa arborescência não podem existir ciclos.

3.5.1.1 - O ARCO A SER SUBSTITUIDO TEM FLUXO ZERO

No caso de estarmos testando um arco (i,j), onde o arco (k,j) que deve ser excluido tem fluxo zero, o objetivo será apenas tentar diminuir o peso do segmento de j na espectativa de, em iterações posteriores, compor um segmento que substitua um outro diminuindo o peso da arborescência geradora.

Proposição 11: A inclusão do arco (i,j), em substituição ao arco (k,j) de fluxo zero, implicará num segmento de peso menor para o vértice j se o valor de ALTER, definido pela expressão (III-1), for negativo. #####

Em outras palavras, ao determinar um segmento de peso menor para j, estamos fazendo com que a arborescência de Steiner, contida na arborescência geradora, alcance o vértice j utilizando um caminho de peso inferior. Isto permitirá que um arco alcançado pelo vértice j diminua o valor da alteração ao ser incluido, embora tal valor seja positivo.

ALGORITMO PARA SELECIONAR UM ARCO COM FLUXO ZERO

Passo 0: Começar com a lista vazia.

Passo 1: Procurar, no grafo, um arco (i,j) que não esteja na arborescência talque F(PREC(j),j) = 0. Se não existe, FIM, já que não é possível contruir este tipo de lista.

Passo 2: Determinar os vértices pi e pj, fazendo (pi,pj) = p(i,j). Calcular o valor de ALTER, definido pela expressão (III-1). Se ALTER < 0, incluir o arco (i,j) na lista e FIM, já que a lista, de um arco só, foi construida. Caso contrário, retornar ao passo 1. #####

3.5.1.2 - O ARCO A SER SUBSTITUIDO TEM FLUXO POSITIVO

Neste caso haverá, necessariamente, uma redistribuição dos fluxos e estaremos apenas interessados nos arcos que levem efetivamente a uma diminuição no peso da arborescência geradora.

Proposição 12: A inclusão do arco (i,j), em substituição ao arco (k,j) de peso positivo, implicará numa diminuição no peso da arborescência geradora se o valor de ALTER, na expressão (III-1), for negativo. **#####**

ALGORITMO PARA SELECIONAR UM ARCO COM FLUXO POSITIVO

Passo 0: Começar com a lista vazia.

Passo 1: Procurar, no grafo, um arco (i,j) que não esteja na arborescência tal que F(PREC(j),j) > 0. Se não existe, FIM, já que não é possível contruir este tipo de lista.

Passo 2: Determinar os vértices pi e pj, fazendo (pi,pj) = p(i,j). Calcular o valor de ALTER, (III-1). Se ALTER < 0, incluir o arco (i,j) na lista e FIM, já que a lista, de um arco só, foi completada. Caso contrário, retornar ao passo 1. **#####**

3.5.2 - LISTA COMPOSTA

O fato de um arco, individualmente, não diminuir o peso de uma arborescência geradora com a sua inclusão, não implica que deva ser descartada, definitivamente, a possibilidade de aproveitá-lo. Isto se deve a que existem outras formas de diminuir o peso de uma arborescência, através de listas de arcos.

Basicamente, existem três tipos de listas de arcos que podem levar a uma redução no peso de uma arborescência geradora: arcos cujos vértices iniciais pertencem a um mesmo segmento de fluxo zero, arcos cujos vértice finais têm segmentos iniciados num mesmo vértice optativo, e arcos onde um deles forma ciclo com os demais arcos da arborescência, sendo desfeito por outro arco da mesma lista.

3.5.2.1 - LISTA TIPO I: ARCOS CUJOS VÉRTICES INICIAIS PERTENCEM A UM MESMO SEGMENTO DE FLUXO ZERO

Seja (i,j) o arco testado e sejam pi e pj os vértices p(i,j), definidos anteriormente, a partir dos quais pode haver uma alteração no fluxo dos segmentos de i e de j, respectivamente. Para iniciar a contrução deste tipo de lista, a partir do arco (i,j), devem dar-se as seguintes condições:

a. o fluxo do segmento do vértice i deve ser zero e o do segmento do vértice j deve ser positivo; e

b. considerando o arco (i,j), e os vértices pi e pj correspondentes, deve acontecer que ALTER ≥ 0 e ALTER - CC < 0, considerando o valor de ALTER definido pela expressão (III-1).

Caso sejam satisfeitas as condições anteriores, podemos pensar na seguinte situação. Como (ALTER - CC) é negativo, se não existisse o incremento no peso devido ao segmento de i, o arco (i,j) conseguiria diminuir o peso da arborescência geradora. Então, através de arcos, cujos vértices iniciais estejam no segmento de i, devemos tentar neutralizar o peso deste segmento. Para isto, devemos analisar todos os arcos com i como vértice inicial tal que, para cada um deles, (ALTER -

CC) seja negativo. Se a soma destes valores mais o peso do segmento de i for negativo, então a lista consegue diminuir o peso da arborescência geradora. Caso contrário, devemos verificar se é conveniente analisar, da mesma forma anterior, o vértice x = PREC(i). Para que seja conveniente analisar o vértice x, é necessário que a soma dos valores (ALTER - CC), dos arcos já incluidos na lista, mais o peso do caminho x-i, seja negativa. Isto porque, se tal soma for positiva, os arcos já testados favoravelmente estariam prejudicando o teste dos arcos iniciados nos vértices do segmento de i e ainda não analisados. Se for conveniente, incluem-se na lista todos os arcos, iniciados no vértice x, para os quais o valor de (ALTER - CC) é negativo. Este procedimento é repetido para cada um dos vértices precedentes a x, fazendo x = PREC(x), até que aconteça uma das seguintes situações:

a) a soma dos valores (ALTER - CC), de todos arcos já incluidos na lista, mais o peso do segmento de i, é negativa, implicando na aceitação da lista;

b) a soma dos valores (ALTER - CC), de todos os arcos já incluidos na lista, mais o peso do caminho x-i, é positiva, o que nos leva a rejeitar a lista;

c) sendo necessário analisar mais um vértice, PREC(x), tal vértice não pertence ao segmento de i, o que implica na rejeição da lista. Mais especificamente, isto acontece quando PREC(x) é igual a PP(i), ou seja, quando atingimos o vértice inicial do segmento de i, o qual, por definição, não pertence ao segmento de i.

Observando o exemplo da fig. (III.7), onde os valores ao lado de cada arco representam o fluxo e o peso, respectivamente, e no qual os vértices 1, 2, 5, 10 e 12 são obrigatórios, vemos que a inclusão dos arcos (11,6) e (7,12), individualmente, aumentam o peso da arborescência em 9 e 2 unidades, respectivamente. Mas, os dois arcos juntos conseguem compensar o peso do segmento do vértice 11. A contribuição do arco (11,6) é de 2 - 5 - 3 = -6 unidades e a do arco (7,12) é de 3 - 7 - 6 = -10 unidades, fazendo um total de -16 unidades, quando o peso do segmento do vértice 11 é de 15 unidades. Portanto, os dois arcos, em conjunto, permitem reduzir o peso da arborescência em uma unidade.

Todo arco testado favoravelmente será, simplesmente, acrescentado à lista de arcos, ou seja, nenhum arco será substituido por outro na lista. A possibilidade de substituição será tratada posteriormente entre as sugestões para melhorar este método.

```
                            ---
            ...............| 1 |..............
            |               ---              |
            |                |               |
      2/5   |          1/5   |         1/9   |
            V                V               V
           ---              ---             ---
          | 2 |            | 3 |           | 4 |............
           ---              ---             ---            |
            |                |               |             |
      1/3   |          1/3   |        0/12   |       1/7   |
            V                V               V             V
           ---              ---             ---           ---
          | 5 |            | 6 |<- -.      | 7 |- -.     | 8 |
           ---              ---     .       ---    .      ---
            |                |      .        |     .       |
      0/3   |          1/8   |  -/2 .  0/3   | -/3 .  1/6  |
            V                V      .        V     .       V
           ---              ---     .       ---    .      ---
          | 9 |            | 10)|   - -| 11|     - ->| 12|
           ---              ---             ---           ---
```

Fig. III.7

No cálculo da contribuição de cada arco, para compensar o peso do segmento do vértice i, devemos ter especial cuidado para evitar que estas contribuições, ou parte delas, sejam consideradas mais de uma vez. Isto poderia acontecer quando o vértice final, do arco sendo testado, estiver no mesmo fragmento que o vértice final de alguns dos arcos já incluidos na lista. Como existe a possibilidade de existir, simultaneamente, mais de um arco na lista, cujos vértices finais estejam no mesmo fragmento, devemos identificar aquele cujo vértice final estiver mais distante da raiz da arborescência. Se o vértice final, do arco sendo testado, alcança tal vértice, então não há vantagem em utilizar este arco, já que a sua inclusão não melhoraria o valor da alteração. Se, no entanto, o vértice final é alcançado por aquele vértice, apenas a porção do segmento, do vértice final deste arco, compreendida entre os dois vértices, e o arco encontrado na lista, contribuem para compensar o peso do segmento do vértice i.

Na figura (III.8), o arco (i,j) está sendo testado e o arco (k,l) já está na lista. Quando o arco (k,l) foi incluido na lista, a sua contribuição foi

$$C(k,l) + CS(pl) - CS(l),$$

onde o vértice pl é tal que

$$(pk,pl) = p(k,l).$$

Agora, se fosse incluido o arco (i,j), a contribuição seria

$$C(i,j) + CS(pj) - CS(j) - C(k,l) + CS(l) - CS(pl),$$

onde o vértice pj é tal que

$$(pi,pj) = p(i,j),$$

já que o caminho compreendido entre os vértices pl e l tinha sido considerado anteriormente, e o arco (k,l) passaria a ter fluxo zero.

Como pj e pl representam o mesmo vértice, a contribuição devido à consideração, na lista, do arco (i,j), seria

$$C(i,j) + CS(l) - CS(j) - C(k,l).$$

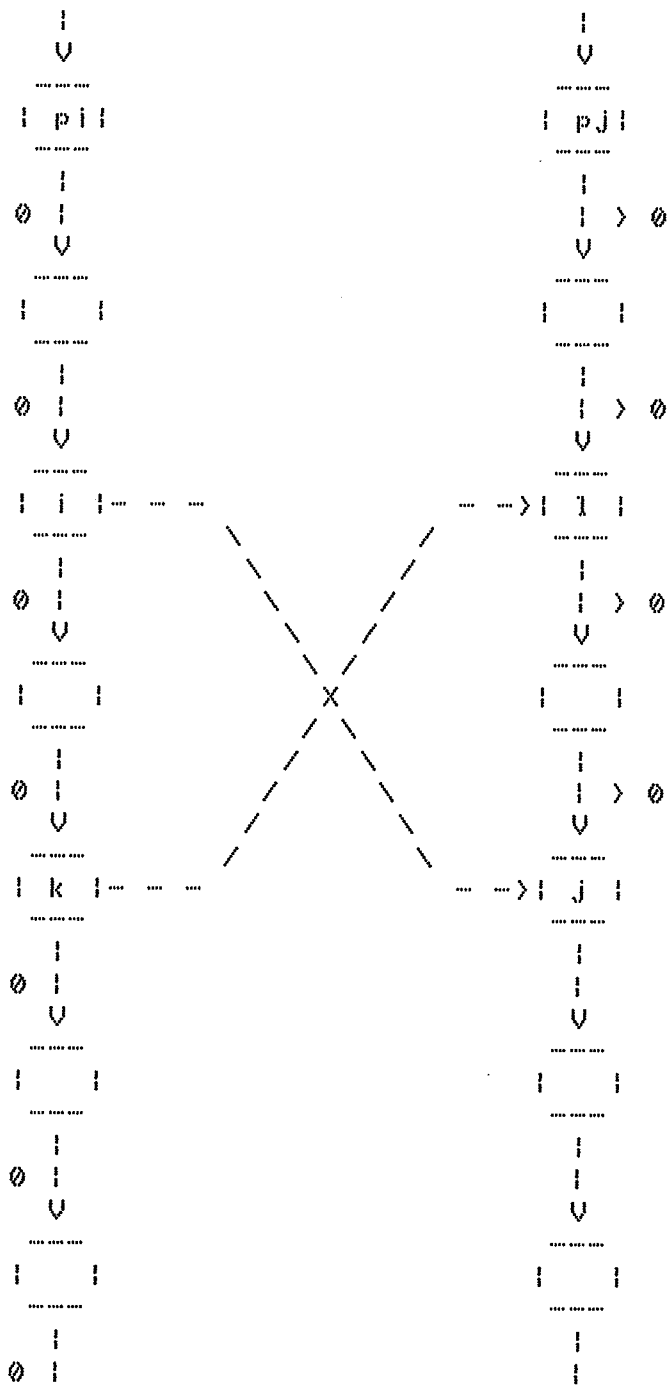


Fig. III.8

Proposição 13: O valor da contribuição devido à inclusão do arco (i,j) é

$$CONT = C(i,j) + CS(l) - CS(j) - C(k,l), \qquad (III-2.a)$$

se existe o arco (k,l) na lista, tal que os vértices j e l pertencem ao mesmo fragmento, sendo l o vértice alcançado mais distante da raiz no interior do fragmento, ou

$$CONT = C(i,j) - CS(j) + CS(pj), \qquad (III-2.b)$$

caso contrário. **#####**

Proposição 14: A inclusão de um arco (i,j) na lista contribuirá para compensar o peso de um segmento, no qual o vértice i está contido, se o valor de CONT, (III-2), for negativo. **#####**

Assim, um arco (i,j) só será incluido na lista se o valor de CONT, (III-2), for negativo.

O procedimento utilizado para determinar este tipo de lista é descrito a seguir.

ALGORITMO PARA CONTRUIR UMA LISTA TIPO I

Passo 0: Começar com a lista vazia.

Passo 1: Procurar um arco (i,j), que não forme ciclo, tal que o fluxo do segmento do vértice j seja positivo, e o do vértice i seja zero, e tal que, considerando $(pi,pj) = p(i,j)$,

$$C(i,j) - CS(j) + CS(pj)$$

seja negativo. Incluir o arco (i,j) na lista e fazer

$$V = C(i,j) - CS(j) + CS(pj).$$

Fazer, também, $x = i$.

Passo 2: Seja T o valor total da variação devida à lista,

$$T = V + CS(i) - CS(PP(i)).$$

Se $T < 0$, FIM, já que a lista de arcos consegue diminuir o peso da arborescência. Caso contrário, vá ao passo 3.

Passo 3: Procurar um arco (x,y), que não forme ciclo, tal que o fluxo do segmento do vértice y seja positivo. Se não for possível encontrar um arco com estas características, vá ao passo 5. Caso contrário, calcular o valor da contribuição deste arco

$$Q = C(x,y) + CS(l) - CS(y) - C(k,l),$$

se existe o arco (k,l), na lista, tal que l é o vértice que está mais distante da raiz, dentro do mesmo fragmento que o vértice y, ou, caso contrário,

$$Q = C(x,y) - CS(y) + CS(py),$$

onde $(px,py) = p(x,y)$.

Passo 4: Se $Q < 0$, incluir o arco na lista, atualizar a variação total fazendo

$$V = V + Q,$$

e retornar ao passo 2. Caso contrário, retornar ao passo 3.

Passo 5: Seja $w = PREC(x)$. Se $w = PP(i)$, FIM, já que o segmento não tem mais vértices e a lista não conseguiu diminuir o peso da arborescência geradora. Caso contrário, fazer $x = w$.

Se o valor

$$V + CS(i) - CS(w)$$

for negativo, retornar ao passo 3. Caso contrário, FIM, já que não é mais conveniente continuar expandindo esta lista. **#####**

3.5.2.2 - LISTA TIPO II: ARCOS CUJOS VÉRTICES FINAIS TÊM SEGMENTOS INICIADOS NUM MESMO VÉRTICE OPTATIVO

Neste tipo de lista pretendemos explorar o fato de que, quando um vértice optativo é início de vários fragmentos, com pelo menos dois deles tendo fluxo positivo, se for possível tornar zero o fluxo de um segmento em cada um destes fragmentos, também tornar-se-á zero o fluxo do fragmento terminado no vértice inicial dos outros, ou parte dele.

Em outras palavras, observando a figura (III.9), onde temos três fragmentos, k-l, l-s e l-w, se conseguimos encontrar dois arcos, incidentes a vértices de cada um dos fragmentos l-s e l-w, por exemplo os arcos (i,s) e (j,t), estaremos tornando zero o fluxo dos segmentos dos vértices s e t. Como o vértice l é optativo, o fluxo do fragmento terminado nele é igual à soma dos fluxos de todos os fragmentos iniciados nele. Sendo assim, o fluxo do fragmento k-l também tornar-se-á zero.

Na construção desta lista, assim como na do anterior, são desprezados todos os arcos que formam ciclo com os demais arcos da arborescência. Além destes, também são desprezados os arcos que formam ciclo com outros arcos da mesma lista.

Proposição 15: A lista Tipo II, associada a um vértice optativo l, deverá atingir, sempre, um número de arcos igual ao número de fragmentos com fluxo positivo iniciados no vértice l, para conseguir a liberação do fragmento terminado no vértice l, ou parte dele. **#####**

Para cada um dos fragmentos iniciados no vértice l deverá existir, na lista, exatamente um arco incidente a um de seus vértices, o qual fornecerá o fluxo que está sendo obtido do vértice l, liberando o segmento correspondente.

A possibilidade, mencionada anteriormente, de formar ciclo com outros arcos da mesma lista, só pode acontecer quando, ao testar um arco incidente ao vértice de algum fragmento iniciado no vértice l, tal arco se inicia num vértice pertencente a um outro fragmento também iniciado no vértice l.

Outro caso, possível de acontecer, é aquele em que o arco

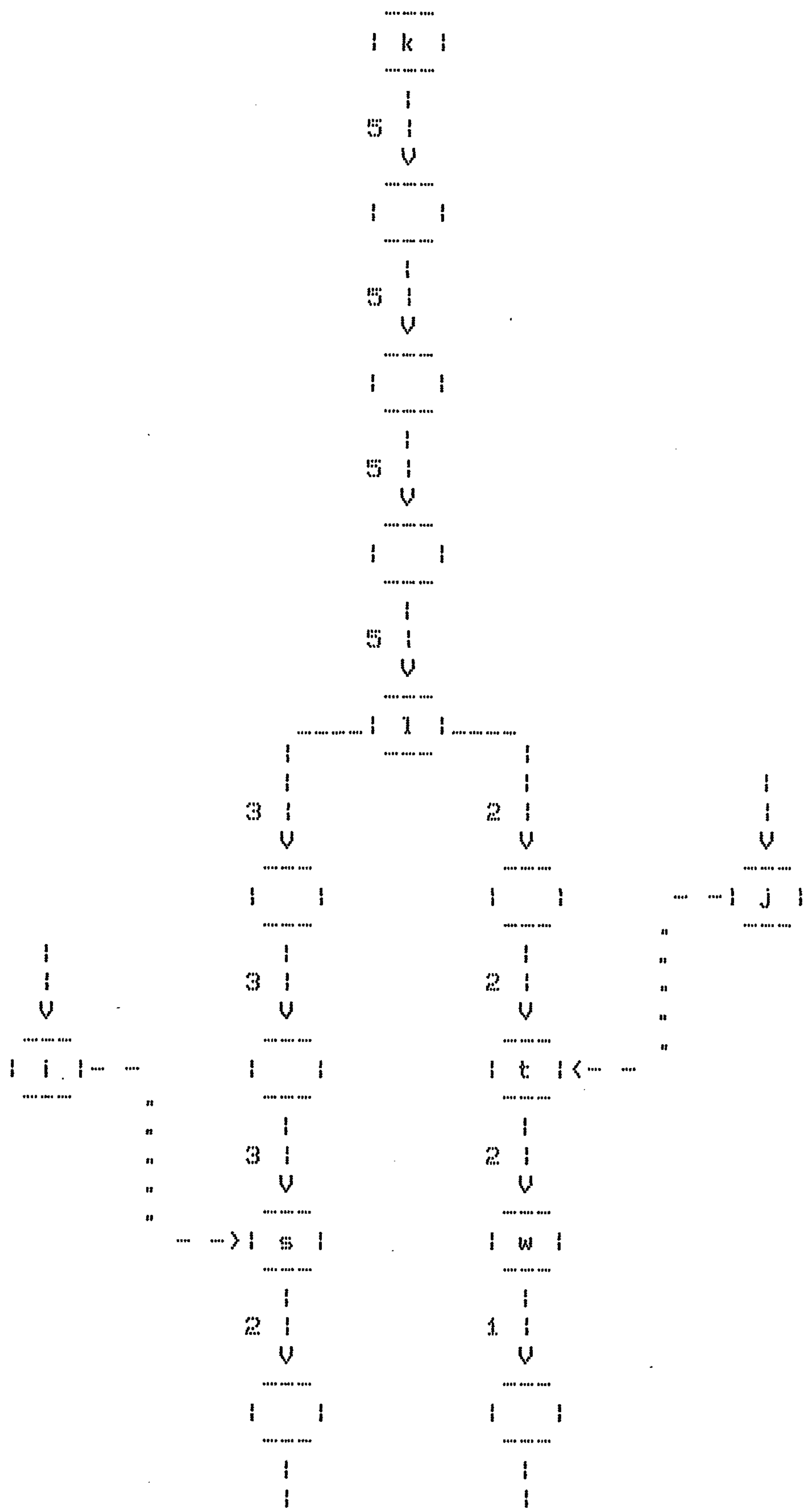


Fig. III.9

testado se inicia num vértice do fragmento terminado no vértice 1, ou se o segmento de seu vértice inicial se inicia num vértice desse fragmento. Isto deve ser considerado porque, neste caso, nem todo o fragmento, terminado no vértice 1, será zerado ao término deste processo.

Um exemplo, que ilustra este último caso, pode ser observado na figura (III.10), onde o fragmento k-1 não será totalmente zerado, já que o segmento k-u permanecerá com fluxo positivo para alimentar o fluxo do arco (i,s).

Isto nos permite concluir que, considerando os fragmentos em torno de um vértice optativo, o fragmento precedente a ele terá seu fluxo zerado a partir de um vértice determinado pelos arcos da lista. Toda vez que um arco é incorporado à lista, tal vértice pode mudar e, neste caso, também haverá uma alteração no cálculo do valor da contribuição do arco incluido. Assim, cada arco na lista tem associado um vértice a partir do qual o fluxo do fragmento PP(1)-1 pode ser zerado.

Também, como cada fragmento iniciado no vértice optativo só pode ter um arco incidente a um de seus vértices, todo arco, testado posteriormente, incidente a um vértice do mesmo fragmento, só poderá ser incluido na lista em substituição ao outro, se isto for conveniente, depois de considerar todas as variações devidas à possível substituição.

Os arcos que não são alcançados pelo fragmento terminado no vértice 1, têm associados o vértice PP(1), início do fragmento, como candidato a vértice inicial do caminho a ser zerado em tal fragmento. O vértice, a partir do qual o fragmento PP(1)-1 pode ser zerado, corresponde ao vértice que está mais distante da raiz, dentre os vértices associados aos arcos da lista.

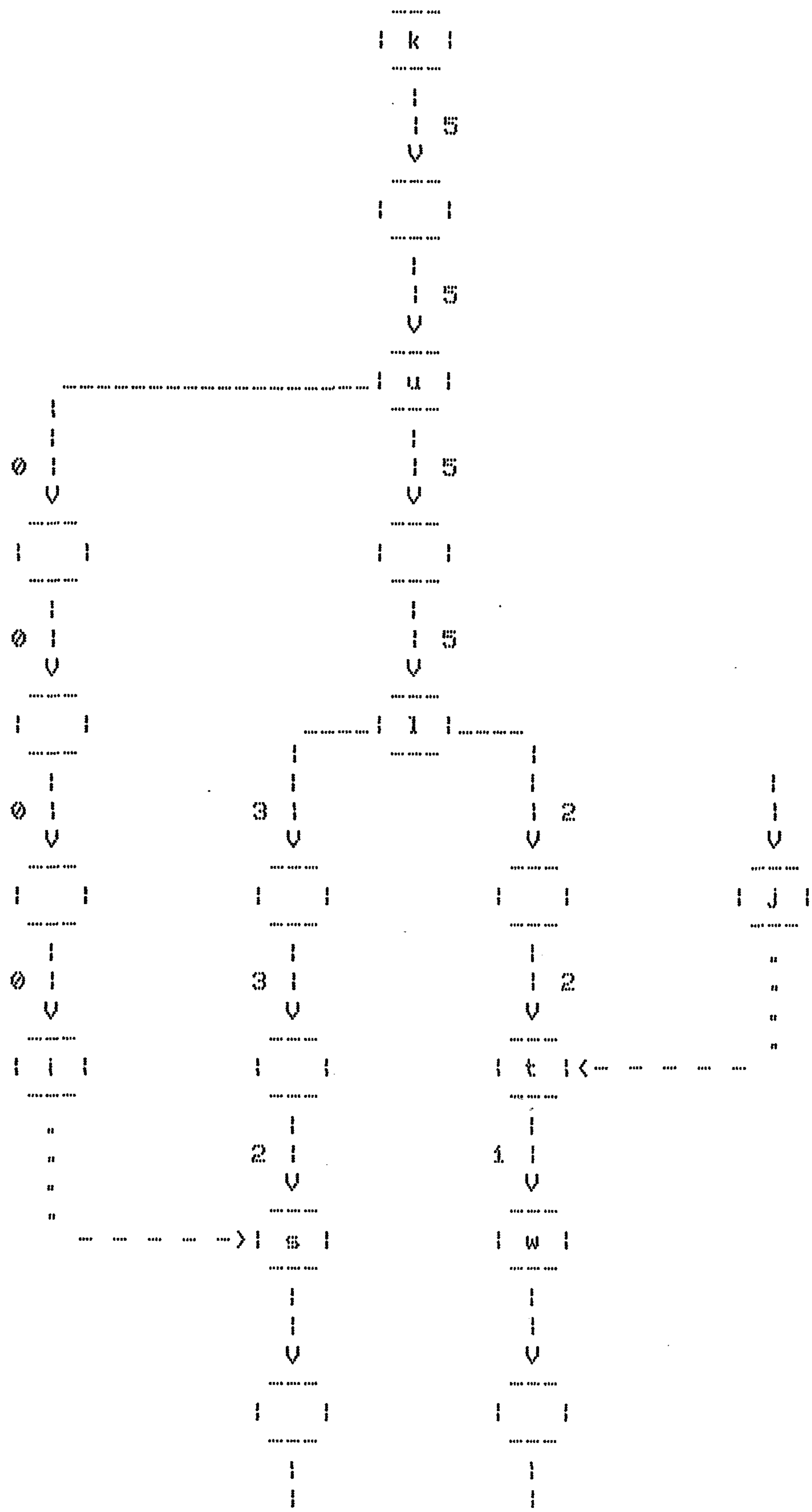


Fig. III.10

Os arcos, com vértices inicial e final em dois fragmentos iniciados no mesmo vértice l, serão considerados separadamente, já que existe a seguinte condição para serem utilizados: o vértice inicial destes arcos deve ser alcançado pelo vértice final de um dos arcos da lista, para garantir a sequência do fluxo e manter zerado algum segmento do fragmento comum aos dois vértices.

Então, numa primeira etapa, rejeitamos os arcos entre fragmentos que têm o mesmo vértice inicial.

Considerando uma lista de arcos, que pode ser vazia, seja l o vértice optativo onde se iniciam vários fragmentos de fluxo positivo. Seja k = PP(l) o vértice inicial do fragmento precedente ao vértice l. Seja pl o vértice, determinado pelos arcos da lista, a partir do qual o fragmento k-l é zerado. Finalmente, seja (i,j) o arco a ser testado. Inicialmente, a variação no peso da arborescência, associada ao vértice l, é

$$CV(l) = CS(PP(l)) - CS(l),$$

o qual representa um valor negativo, e que corresponde à liberação de todo o fragmento terminado no vértice l.

A primeira fonte de variação, no peso da arborescência, está na possibilidade de que exista um vértice x, no interior do fragmento k-l, que alcance o vértice i, e que esteja mais distante da raiz que o vértice pl considerado atualmente. Neste caso, a variação é no sentido de aumentar o peso e pode ser representada por

$$C1 = CS(x) - CS(pl).$$

Caso contrário,

$$C1 = 0.$$

O vértice x pode ser o próprio vértice i ou, se existe o segmento x-i, de fluxo zero, pode ser o vértice PP(i).

Se o segmento do vértice i tiver fluxo zero, o arco (i,j) pode acrescentar, à variação do peso, o valor

$$C2 = CS(i) - CS(PP(i)),$$

se não existe nenhum arco, na lista, cujo vértice inicial pertença ao mesmo fragmento que o vértice i, ou

$$C2 = CS(i) - CS(x),$$

se existe um arco (x,y), na lista, tal que seu vértice inicial x é o que está mais distante da raiz no mesmo fragmento que o vértice i, e x alcança i. Em qualquer outro caso,

$$C2 = 0.$$

Se não existe nenhum arco na lista, cujo vértice final esteja no mesmo fragmento que o vértice j, então o arco (i,j) deve ser incluido na lista e a variação no peso, devida a este arco, é

$$CT = C1 + C2 + C(i,j) - CS(j) + CS(1), \qquad \text{(III-3.a)}$$

e a variação total, associada ao vértice 1 e devida à lista toda, é

$$CV(1) = CV(1) + CT.$$

Se, no entanto, existe um arco (u,v) na lista, cujo vértice final v está no mesmo fragmento que o vértice j, então, se for decidida a inclusão do arco (i,j), haverá, necessariamente, uma substituição. Neste caso, além dos valores C1 e C2, definidos anteriormente, existem outras fontes de variação.

O arco a ser incluido gera uma variação no peso devido à alteração no fluxo do caminho compreendido entre os vértices j e v, variação que pode ser para mais ou para menos, e cujo valor é

$$C3 = C(i,j) - CS(j) + CS(v).$$

Se o segmento do vértice u tiver fluxo zero, então, quando o arco (u,v) foi incluido na lista, o peso desse segmento, ou parte dele, poderia ter sido adicionado à variação total da lista. Portanto, ao substituir o arco, essa variação, ou parte dela, deverá ser diminuida. Para efeitos deste cálculo, devemos considerar uma lista auxiliar, formada por todos os arcos da lista, mas com o arco (i,j) no lugar do arco (u,v). O valor da porção desta variação devida ao arco a ser substituido na lista é

$$C4 = CS(PP(u)) - CS(u) - C(u,v),$$

se não existe nenhum arco na lista auxiliar cujo vértice inicial pertença ao mesmo fragmento que o vértice u, ou

$$C4 = CS(x) - CS(u) - C(u,v),$$

se existe um arco (x,y) na lista auxiliar tal que o seu vértice inicial x pertence ao mesmo fragmento que o vértice u, sendo x o vértice mais distante da raiz, no fragmento de u, onde se inicia um arco da lista, tal que x alcança u. Em qualquer outro caso,

$$C4 = -C(u,v).$$

Finalmente, se dentre os vértices associados aos arcos da lista auxiliar, o vértice pp1 é o que está mais distante da raiz e, ao mesmo tempo, o vértice pp1 está mais perto da raiz que o vértice p1, então haverá uma alteração no vértice a partir do qual o fragmento k-1 é zerado, causando a seguinte variação, negativa, na variação total associada ao vértice 1,

$$C5 = CS(pp1) - CS(p1).$$

Em qualquer outro caso,

$$C5 = 0.$$

O valor total da variação, devida à substituição do arco (u,v) pelo arco (i,j), seria

$$CT = C1 + C2 + C3 + C4 + C5, \qquad \text{(III-3.b)}$$

e a substituição só será efetivada se o valor de CT for negativo.

Proposição 16: Um arco (i,j), tal que o segmento do vértice j se inicia no vértice optativo l, será incluido na lista correspondente ao vértice l se:

a. na lista não existe nenhum outro arco incidente a um vértice pertencente ao mesmo fragmento que o vértice j; ou

b. na lista existe outro arco incidente a um vértice pertenecente ao mesmo fragmento que o vértice j e CT, definido pela expressão (III-3), é menor que zero, sendo o arco (i,j) incluido em substituição ao outro. **#####**

Se o valor de CT, (III-3), for negativo, faz-se a substituição e atualiza-se o valor da variação total devida à lista fazendo

$$CV(l) = CV(l) + CT. \qquad \text{(III-4)}$$

Proposição 17: Uma lista de arcos Tipo II, associada a um vértice optativo l, conseguirá diminuir o peso de uma arborescência geradora se:

a. cada fragmento de fluxo positivo iniciado no vértice l tiver exatamente um arco incidente a um de seus vértices; e

b. a variação total CV(l), definida por (III-4), devida a tal lista, for negativa. **#####**

Cada vértice optativo l, que for início de mais de um fragmento com fluxo positivo, tem associado um valor CV(l), (III-4). Então, este procedimento é efetuado para cada um dos arcos do grafo, que não forme ciclo nem esteja entre dois fragmentos iniciados no mesmo vértice optativo, até que aconteça uma das alternativas seguintes:

a. não há mais arcos disponíveis, no grafo, para serem testados; ou

b. alguma lista, associada a um vértice optativo, foi considerada conveniente e aceita para ser incluida na arborescência geradora.

Se acontece a alternativa b), procede-se à atulização da arborescência geradora. No entanto, se acontece alternativa a), ainda é possível diminuir a variação associada a alguns vértices optativos, considerando agora só os arcos entre fragmentos iniciados num mesmo vértice optativo.

Então, se nenhuma das listas, associadas aos vértices optativos, início dos fragmentos com fluxo positivo, tiver variação negativa, devemos testar os arcos compreendidos entre dois fragmentos irmãos para tentar diminuir ainda mais os valores de algumas variações CV(1).

A primeira condição, que deve ser satisfeita por estes arcos, diz respeito aos seus vértices iniciais. Como foi mencionado anteriormente, para garantir a sequência do fluxo e manter zerado um segmento em cada fragmento iniciado em 1, é necessário que o vértice final v, do arco (u,v), na lista, alcance o vértice inicial i, do arco (i,j) que está sendo testado.

Por outro lado, isto pode levar à formação de um ciclo com os demais arcos da lista. Se isto acontecer, o arco deve ser rejeitado.

Para os arcos que satisfazem estas condições, devemos calcular os valores de C2, C3, C4, C5 e CT, já que C1 = 0 sempre, tal como foram definidos anteriormente, e efetuar a substituição do arco (u,v) pelo arco (i,j) se CT for negativo. Posteriormente, atualizar o valor de CV(1) e, se for negativo, aceitar a lista toda e atualizar a arborescência geradora.

Assim, o método para construir este tipo de lista pode ser resumido no seguinte algoritmo.

ALGORITMO PARA CONSTRUIR UMA LISTA TIPO II

Passo 0: Começar com todas as listas, associadas a vértices optativos, vazias.

Passo 1: Procurar um arco (i,j), ainda não testado, que não forme ciclo e tal que o fluxo do segmento do vértice j seja positivo. Se tal arco não existe, FIM.

Passo 2: Seja l = PP(j), o vértice inicial do segmento do vértice j. Se l for um vértice obrigatório, retornar ao passo 1. Caso contrário, seja k = PP(l) o vértice inicial do fragmento precedente ao vértice l. Também, seja pl o vértice, determinado pelos arcos da lista associada ao vértice l, a partir do qual o fragmento k-l é zerado. Se a lista estiver vazia, pl = PP(l) e o valor inicial da variação associada à lista do vértice l é

$$CV(l) = CS(pl) - CS(l).$$

Passo 3: Se existe o vértice x, no interior do fragmento k-l, tal que x = i ou x = PP(i), e tal que CS(x) > CS(pl), então

$$C1 = CS(x) - CS(pl).$$

Caso contrário,

$$C1 = 0.$$

Passo 4: Fazer

$$C2 = CS(i) - CS(PP(i)),$$

se não existe nenhum arco, na lista, cujo vértice inicial esteja no mesmo fragmento que o vértice i, ou

$$C2 = CS(i) - CS(x),$$

se existe um arco, na lista, tal que seu vértice inicial x esteja no mesmo fragmento que o vértice i, e tal que CS(x) < CS(i). Em qualquer outro caso,

$$C2 = 0.$$

Fazer, também,

$$CT = C1 + C2 + C(i,j) - CS(j) + CS(l).$$

Se não existe nenhum arco na lista, cujo vértice final esteja no mesmo fragmento que o vértice j, então vá ao passo 8.

Passo 5: Seja (u,v), o arco na lista, cujo vértice final v está no mesmo fragmento que o vértice j. Contruir uma lista auxiliar contendo todos os arcos da lista associada ao vértice l, mas com o arco (i,j) no lugar do arco (u,v).

Passo 6: Fazer

$$C3 = C(i,j) - CS(j) + CS(v).$$

Fazer, também,

$$C4 = CS(PP(u)) - CS(u) - C(u,v),$$

se não existe nenhum arco na lista auxiliar cujo vértice inicial pertença ao mesmo fragmento que o vértice i, ou

$$C4 = CS(x) - CS(u) - C(u,v),$$

se existe um arco na lista auxiliar tal que o seu vértice inicial x pertence ao mesmo fragmento que o vértice i, sendo x o vértice mais distante da raiz, no mesmo fragmento de i, onde se inicia um arco da lista, e tal que CS(x) < CS(i). Em qualquer outro caso,

$$C4 = -C(u,v).$$

Calcular, também,

$$C5 = CS(pp1) - CS(p1),$$

se a lista auxiliar determina o vértice pp1 como novo vértice inicial do caminho, contido no fragmento k-1, a ser zerado, tal que CS(pp1) < CS(p1). Caso contrário, fazer

$$C5 = 0.$$

Passo 7: Calcular

$$CT = C1 + C2 + C3 + C4 + C5.$$

Se CT ≥ 0, retornar ao passo 1.

Passo 8: Incluir o arco (i,j) na lista, substituindo o arco (u,v) se for o caso, e atualizar o valor total da variação devida à lista associada ao vértice 1 fazendo

$$CV(1) = CV(1) + CT.$$

Passo 9: Se existe algum fragmento, iniciado no vértice 1, que não tenha nenhum arco na lista incidente a um de seus vértices, retornar ao passo 1. Caso contrário, se CV(1) < 0, FIM, já que a lista associada ao vértice 1 permite diminuir o peso da arborescência geradora e, portanto, todos os seus arcos devem ser incluidos nela. Caso contrário, retornar ao passo 1. #####

Ao término deste método, se não for possível formar uma lista que permita diminuir o peso da arborescência geradora, devemos repetir os mesmos passos anteriores, com as duas seguintes alterações:

a) no passo 1, procuraremos arcos (i,j) que não formem ciclo, cujos vértices inicial e final pertençam a fragmentos diferentes, iniciados num mesmo vértice optativo, e tal que exista na lista um arco incidente a um vértice x no mesmo fragmento que o vértice i, e tal que CS(x) < CS(i);

b) no passo 3, o valor de Ci sempre será zero.

3.5.2.3 - LISTA TIPO III: ARCOS COM CICLO COMPENSADO

Até agora, na formação dos dois tipos de listas descritas anteriormente, os arcos que formam ciclos tem sido desprezados sistematicamente. Aqui, pela primeira vez, serão considerados arcos que formam ciclo com os demais arcos da arborescência geradora. Um arco com esta característica marcará o início da formação da lista.

Então, para começar a construir esta lista, precisamos de um arco que forme ciclo com os demais arcos da arborescência. Seja (ic,jc) um arco que, se incluido na arborescência, formaria ciclo com os demais arcos dela. Nesta versão do método, dentre os arcos que formam ciclo, consideraremos apenas os arcos onde o segmento do vértice final jc tem fluxo positivo.

Na figura (III.11) pode ser observado o arco (ic,jc) formando ciclo com arcos da arborescência.

O arco (ic,jc), se incluido na arborescência, provocaria uma variação no peso dela no valor

$$R = C(ic,jc) - CS(jc) + CS(PP(jc)) + CS(ic) - CS(PP(ic)),$$

se o segmento do vértice ic tiver fluxo zero, ou

$$R = C(ic,jc) - CS(jc) + CS(PP(jc)),$$

caso contrário.

Se o valor de R for negativo, o arco (ic,jc) é incluido na lista, o que nos leva à formação de um ciclo se não tomamos as devidas providências. Como uma arborescência não aceita a existência de ciclos, nossa primeira preocupação será no sentido de incluir na lista um outro arco que garanta a destruição do ciclo. Para isso, devemos procurar um arco que, sem formar outro ciclo nem manter o já formado, seja incidente a um dos vértices do ciclo, que não o vértice jc. Seja kc ≠ jc um vértice do ciclo. Para não manter o ciclo já formado será suficiente desprezar os arcos iniciados em vértices do mesmo ciclo.

Um arco (i,kc), incidente ao vértice kc no ciclo, altera o peso da arborescência no valor

$$R1 = C(i,kc) - CS(kc) + CS(PP(kc)),$$

```
                  ---
                 |   |
                  ---
                   |
                   |
                   V
                  ---
                 |   |
                  ---
                   |
                   |
                  ---
                 |   |
                  ---
                   |
                   |
                   V
                  ---
                 | jc|<- - -  "
                  ---         "
                   |          "
                   |          "
                   V          "
                  ---         "
                 |   |        "
                  ---         "
   |               |          "
   |               |          "
   V               V          "
  ---             ---         "
 | i |- - - - - ->| kc|        "
  ---             ---         "
                   |          "
                   |          "
                   V          "
                  ---         "
                 |   |        "
                  ---         "
                   |          "
                   |          "
                   V          "
                  ---         "
                 | ic|- - - - "
                  ---
                   |
                   |
                   V
                  ---
                 |   |
                  ---
                   |
```

Fig. III.11

se PP(kc) está mais distante da raiz que jc, ou

$$R1 = C(i,kc) - CS(kc) + CS(jc),$$

caso contrário. Por outro lado, se o segmento do vértice i tiver fluxo zero, haverá também uma variação no valor

$$R2 = CS(i) - CS(PP(i)).$$

Caso contrário,

$$R2 = 0.$$

Se o valor de R + R1 não for negativo, rejeita-se o arco (i,kc), e ténta-se procurar outro. Se o valor de R + R1 + R2 for negativo,então a lista formada por estes dois arcos é aceita para ser incorporada à arborescência, tendo o cuidado de inverter a ordem em que estes dois arcos são incluidos para evitar a efetiva formação do ciclo. Caso contrário, devemos tentar compensar o aumento de peso com arcos iniciados em vértices do segmento de i, se este segmento tiver fluxo zero. Se o fluxo do segmento do vértice i for zero, para compensar o peso do segmento do vértice i e dos efeitos no peso provocados pelos dois primeiros arcos, deveremos proceder da mesma forma que na costrução da lista de arcos com vértices iniciais no segmento de i, tratado na Seção (3.5.2.1). Se não for possível compensar o peso com o procedimento anterior, ainda poderemos tentar a compensação com arcos que, iniciados em vértices do caminho kc-ic, no ciclo, sem formar novos ciclos, consigam diminuir o valor da variação total até torná-la negativa. Se, contudo, não for possível, ainda, tornar negativa a variaão devida a esta lista, podemos tentar formar outras listas, para o mesmo ciclo, selecionando outros arcos para destruí-lo.

Em resumo, podemos utilizar o seguinte algoritmo.

ALGORITMO PARA CONSTRUIR UMA LISTA TIPO III

Passo 0: Selecionar um arco (ic,jc), tal que o fluxo do segmento do vértice jc seja positivo, e que forme ciclo com os demais arcos da arborescência. Se tal arco não existe, FIM.

Passo 1: Fazer

$$R = C(ic,jc) - CS(jc) + CS(PP(jc)) + CS(ic) - CS(PP(ic)),$$

se o segmento do vértice ic tiver fluxo zero, ou

$$R = C(ic,jc) - CS(jc) + CS(PP(jc)),$$

em caso contrário.

Se $R \geq 0$, retornar ao passo 0.

Passo 2: Inicializar a lista com o arco (ic,jc).

Passo 3: Escolher um arco (u,v), com $v \neq jc$ no caminho jc-ic, tal que u não esteja neste caminho, e que não forme ciclo. Se tal arco não existe, retornar ao passo 0.

Passo 4: Calcular

$$R1 = C(u,v) - CS(v) + CS(PP(v)),$$

se $CS(PP(v)) > CS(jc)$, ou

$$R1 = C(u,v) - CS(v) + CS(jc),$$

se $CS(PP(v)) \leq CS(jc)$.

Calcular, também,

$$R2 = CS(u) - CS(PP(u)),$$

se o segmento do vértice u tiver fluxo zero. Caso contrário,

$$R2 = 0.$$

Passo 5: Se $R + R1 \geq 0$, retornar ao passo 3. Caso contrário, incluir o arco (u,v) na lista.

Passo 6: Se $R + R1 + R2 < 0$, FIM, já que a lista formada permite diminuir o peso da arborescência geradora. Caso contrário, devemos utilizar o método definido na Seção 3.5.2.1, a partir do passo 3, com

$$V = R + R1,$$

e com x = i = u. Ao térmimo daquele método, retornar ao passo 7.

Passo 7: Selecionar um arco (k,l), que não forme nenhum tipo de ciclo, tal que o seu vértice inicial k esteja no caminho v-ic, e o fluxo do segmento do vértice l seja positivo. Se tal arco não existe, retornar ao passo 3. Caso contrário, calcular

$$R3 = C(k,l) - CS(l) + CS(PP(l)),$$

se não existe nenhum arco na lista cujo vértice final esteja no mesmo fragmento que o vértice l. No entanto, se existe o arco (s,t) na lista, tal que o vértice t está no mesmo fragmento que o vértice l e, ainda, CS(l) ≤ CS(t), então retornar ao passo 3. Caso contrário, recalcular

$$R3 = C(k,l) - CS(l) + CS(t) - CS(s,t).$$

Passo 8: Se R3 < 0, incluir o arco (k,l) na lista e fazer

$$V = V + R3.$$

Passo 9: Se V < 0, FIM, já que a lista consegue diminuir o peso da arborescência geradora. Caso contrário, retornar ao passo 7. **#####**

É bom destacar que, como o primeiro arco da lista forma ciclo com os demais arcos da arborescência, e que o segundo arco desfaz tal ciclo, é conveniente intercambiar a posição destes dois arcos. Desta forma, ao incluir o arco que deveria formar o ciclo, tal ciclo não seria formado.

3.6 - ACICLICIDADE DA NOVA HEURÍSTICA

O método heurístico apresentado garante que o valor do peso das arborescências geradas se aproxima do valor do peso da solução ótima em forma monotônica não decrescente. O que garante isto é o critério utilizado para determinar quando um arco ou lista de arcos deve ser incluido numa arborescência geradora.

Isto implica que o peso de uma arborescência geradora nunca aumenta. Mas, quando o peso dela não diminui, significa que a subarborescência de Steiner contida não mudou de uma arborescência geradora para outra. No entanto, quando isto acontece, o complemento desta subarborescência de Steiner diminui de peso, mesmo que não contribua na formação da subarborescência de Steiner. Isto nos leva a concluir que, ao considerarmos o peso de todos os arcos das arborescências geradoras, e não só os das subarborescências de Steier, este peso total sempre dimimui, sendo impossível obter uma arborescência geradora obtida em iterações anteriores.

CAPÍTULO IV

EXPERIÊNCIA COMPUTACIONAL

4.1 - IMPLEMENTAÇÃO

O método heurístico, descrito no Capítulo anterior, foi implementado em FORTRAN IV utilizando estruturas de dados apropriadas para armazenar grafos e arborescências. No armazenamento do grafo original, utilizou-se uma estrutura de dados na base de ponteiros, associando dois ponteiros a cada um dos vértices: um deles aponta para o início da lista de vértices adjacentes sucessores e o outro para o início da lista de vértices adjacentes antecessores. Cada um dos vértices das listas aponta para o próximo elemento da mesma lista, se existir. As listas de vértices foram dispostas de tal forma que num mesmo endereço estejam armazenados os vértices inicial e final de cada arco, facilitando desta maneira a associação dos pesos aos arcos, além das facilidades inerentes ao uso de ponteiros.

Para o armazenamento da arborescência geradora, como cada vértice só pode ter um arco incidente, bastou utilizar uma única lista contendo, para cada vértice do grafo, a identificação do vértice inicial do arco incidente a ele, exceto para a raiz que, por definição, não pode ter arco incidente numa arborescência.

Na implementação dos métodos para construir os diferentes tipos de listas, para aproveitar certas afinidades entre as listas Tipo I e Tipo III, o teste dos arcos para a construção destas listas foi feito em conjunto, determinando a construção de uma ou de outra lista, de acordo com o arco analisado. Isto porque, devido às características do primeiro arco de cada uma das listas Tipo I ou Tipo III, elas nunca poderiam ser construidas simultaneamente. Somente se não for possível construir nenhuma destas listas é que se tenta contruir a lista Tipo II.

O algoritmo implementado para escolher os arcos que devem ser incorporados a uma arborescência geradora é o descrito a seguir.

ALGORITMO PARA ESCOLHER OS ARCOS A SEREM INCLUIDOS

Passo 0: Escolher um arco para ser testado. Se tal arco não existe, vá ao passo 5. Se tal arco, por si só, consegue diminuir o peso da arborescência geradora ou diminuir o peso do segmento de um vértice ao qual chega um fluxo zero, então FIM, já que tal arco deve ser incluido na arborescência.

Passo 1: Se este arco satisfaz as condições necessárias para iniciar a formação de uma lista Tipo I, vá ao passo 3.

Passo 2: Se este arco satisfaz as condições necessárias para iniciar a formação de uma lista Tipo III, vá ao passo 4. Caso contrário, retornar ao passo 0.

Passo 3: Tentar a construção de uma lista Tipo I. Se for possível construir a lista Tipo I, FIM, já que a lista permite diminuir o peso da arborescência geradora. Caso contrário, tornar a lista vazia e retornar ao passo 0.

Passo 4: Tentar a contrução de uma lista Tipo III. Se for possível construir a lista Tipo III, FIM, já que a lista permite diminuir o peso da arborescência geradora. Caso contrário, tornar a lista vazia e retornar ao passo 0.

Passo 5: Tentar a construção de uma lista Tipo II. Se for possível, FIM, já que a lista consegue diminuir o peso da arborescência geradora. Caso contrário, FIM, já que não é mais possível diminuir o peso da arborescência geradora com este método. **#####**

O procedimento descrito anteriormente é utilizado apenas para determinar a lista de arcos a ser incluida numa arborescência geradora para diminuir o seu peso. Toda vez que uma lista, de qualquer tipo, for encontrada, após a atualização da arborescencia deverá ser reiniciado novamente o processo de construção de listas, já que as alterações feitas na arborescência podem fazer com que arcos anteriormente rejeitados passem a ser úteis à formação de alguma lista.

Para os testes e comparações foi implementado, também, o método heurístico de Richard T. Wong, por ser um método recente e simples, além de ser rápido e eficiente.

Como o método heurístico desenvolvido neste trabalho precisa de uma arborescência geradora inicial, foram implementadas duas versões, cada uma delas utilizando procedimentos diferentes para construir uma arborescência geradora inicial, visando testar a sensibilidade do método em relação a ela.

Na primeira versão, tentando utilizar, no início, uma arborescência geradora obtida de uma forma rápida, utiliza-se um método que poderia ser classificado como de busca em profundidade. Neste método, a partir da raiz, consideram-se os arcos entre um vértice já alcançado, ou a raiz, e um vértice ainda não alcançado. Isto continua até que todos os vértices do grafo tenham sido alçançados, obtendo, assim, uma arborescência geradora. Neste caso, o vértice raiz, o qual não pode ter nenhum vértice incidente numa arborescência, é considerado, no início, com sendo o único vértice já alcançdo.

Considerando a necessidade de contar com uma arborescência geradora inicial, podemos tentar obtê-la mediante outros métodos heurísticos e, assim, verificar se o novo método consegue melhorar os resultados. Foi por isto que decidimos implementar a segunda versão, onde a arborescência geradora inicial é obtida a partir da arborescência de Steiner resultante da utilização da heurística de Wong. Para completar a arborescência geradora requerida, incluem-se, um a um, os vértices restantes, juntamente com um arco qualquer incidente a eles que se inicie num vértice já existente ou já acrescentado à arborescência.

É necessário destacar que todas as implementações utilizaram o mesmo tipo de estrutura de dados e o mesmo rigor na qualidade da programação.

Para os testes, realizados num computador IBM - 4341, foram gerados, aleatoriamente, 140 problemas, sendo 100 problemas com 100 vértices e 500 arcos, 10 problemas com 200 vértices e 1000 arcos, 10 problemas com 300 vértices e 1500 arcos, 10 problemas com 600 vértices e 3000 arcos e, finalmente, 10 problemas com 900 vértices e 4500 arcos.

Na geração de cada problema utilizou-se o seguinte procedimento:

a. construção de uma arborescência geradora, com n vértices, em forma aleatória;
b. determinação aleatória dos vértices do grafo a serem considerados como obrigatórios;
c. determinação aleatória dos arcos restantes; e
d. geração aleatória dos pesos no intervalo uniforme entre 1 e 50.

4.2 - RESULTADOS

No Apêndice A apresentamos os resultados obtidos pela primeira versão da nova heurística. Nos Quadros ali incluidos, estão contidas as seguintes informações: número do problema, número de vértices obrigatórios, o valor do peso da solução obtida e o tempo de CPU, em segundos, utilizado, sem considerar o tempo de entrada e saida de dados.

No Apêndice B podem ser observados os resultados obtidos pela segunda versão da nova heurística. Nos Quadros apresentados, também estão incluidos os resultados da heurística de Wong. As informações contidas nestes quadros são as seguintes: número do problema, número de vértices obrigatórios, limites inferior e superior para o valor da solução ótima do problema de Steiner obtido mediante a heurística de Wong, tempo CPU, em segundos, utilizado pela heurística de Wong, valor da solução obtida mediante a nova heurística, tempo CPU utilizado exclussivamente pela nova heurística, e tempo CPU total, incluido o tempo CPU utilizado pela heurística de Wong para gerar uma arborescência geradora inicial. No tempo CPU não está incluido o tempo de leitura de dados nem o tempo empregado na impressão de relatórios.

No Apêndice C foram incluidos quadros contendo o resumo dos resultados, no que se refere ao valor da solução obtida para cada problema, e os índices da variação havida nos intervalos onde se encontra a solução ótima.

No caso da heurística de Wong, o limite inferior, obtido mediante o método Dual Ascendente, é sempre menor ou igual que o valor da solução ótima [11]. Isto quer dizer que, em alguns casos, este valor poderia representar uma solução não viável. Por outro lado, a heurística de Wong nos permite obter uma solução aproximada para o problema de Steiner, solução viável que, nos Quadros, aparece como limite superior, já que por ser viável e aproximada, representa um valor que sempre será maior ou igual que o valor da solução ótima. No Apêndice C, as colunas identificadas como **limite superior** referem-se aos valores das soluções aproximadas obtidas pelos métodos utilizados.

Analisando os resultados obtidos pela primeira versão, podemos ver que os valores obtidos, para a solução dos problemas, foram melhores em 60 problemas e piores em, apenas, 25 problemas, o que nos dá uma boa idéia da eficácia deste método em relação ao metodo de Wong. Mas, o tempo de CPU utilizado é significativamente maior. No Apêndice C, para esta primeira versão, além do limite superior obtido, s"ao especificados mais dois índices: um de diminuição e outro de aumento do intervalo sugerido inicialmente pela heurística de Wong. Neste caso, como consideramos qualquer arborescência geradora inicial, é possível que a soluão obtida seja pior que a solução sugerida pela heurística de Wong, e sendo assim, ao considerarmos o mesmo limite inferior, podemos obter um intervalo pior que o de Wong.

Para a segunda versão da heurística, considerando inicialmente a solução obtida mediante o método de Wong, a nova heurística reduz significativamente o tempo de CPU requerido. Ao considerar o método híbrido, representado por esta segunda versão, podemos ver que a diminuição do tempo de CPU utilizado, em relação à primeira versão, continua sendo significativa a um nível mais baixo. Como foi utilizada a solução dada pelo método de Wong para determinar a arborescência geradora inicial, é interessante medir a eficiência do método verificando a obtenção de soluções melhores e o grau de redução do intervalo onde se encontra a solução ótima. Do total de 140 problemas, a nova heurística melhorou a solução da heurística de Wong em 72 problemas. É importante destacar que, dos 68 problemas restantes, com certeza absoluta, o método de Wong tinha atingido a solução ótima em 49 problemas, sendo, portanto, impossível obter soluções melhores. Assim, só restam 19 problemas para os quais existiria a possibilidade de se obter uma solução melhor que a nova heurística não obteve. Também é possível que, em alguns destes 19 problemas, a heurística de Wong tenha alcançado a solução ótima, sem ser evidente como nos outros 49 problemas. Para efeitos destas comparações, foram consideradas ótimas as soluções cujos valores coincidem com o limite inferior fornecido pelo método de Wong. Também, é conveniente destacar que, com certeza, em 60 problemas a nova heurística atingiu a solução ótima, contra 49 problemas pela heurística de Wong, sendo que destes 49 problemas, aqueles nos quais todos os vértices são obrigatórios, num total de 14 problemas, a heurística de Wong sempre atinge a solução ótima. Dos problemas restantes, nos quais existe a possibilidade de a heurística de Wong não atingir a solução ótima, a nova heurística resolveu em forma ótima 45 problemas, contra 34 da heurística de Wong.

Nesta segunda versão, no que diz respeito ao intervalo onde se encontra a solução ótima, podemos observar no Apêndice C as percentagens de diminuição dos intervalos. Como estamos considerando o próprio limite superior, obtido pela heurística de Wong, como ponto de partida na utilização da nova heurística, o intervalo a ser obtido não pode piorar. Por este motivo, no Apêndice C, no que diz respeito a esta versão, não existe o índice de aumento do intervalo, simplesmente porque ele não pode aumentar. Considerando todos os problemas de teste, para os intervalos sugeridos pela heurística de Wong, conseguiu-se uma diminuição média de 38,56%. Como existem vários problemas para os quais a heurística de Wong dá um intervalo de comprimento zero, ou seja, nos quais coincidem os limites inferior e superior, o que nos dá a certeza de termos uma solução ótima, não seria necessário utilizar a nova heurística nestes problemas. Assim, considerando apenas os problemas para os quais a heurística de Wong fornece um intervalo de comprimento positivo, ou seja, onde é possível obter um outro intervalo reduzido, foi conseguida uma diminuição média de 47,21%. Como a nova heurística gera sempre uma solução viável, o valor associado a elas pode ser considerado um limite superior para o intervalo onde se encontra o valor da solução ótima. Então, como só podemos modificar o limite superior, mantendo como limite inferior aquele fornecido pela heurística de Wong, concluimos que o grau de aproximação da solução obtida é ainda melhor do que os índices mostrados acima.

CAPÍTULO V

CONCLUSÕES E RECOMENDAÇÕES

Pelos resultados dos testes, expostos no Capítulo anterior, podemos concluir que o método heurístico, desenvolvido neste trabalho, atingiu plenamente o objetivo, o qual era o de desenvolver um método que permitisse obter boas soluções para o problema de Steiner em tempo razoável. É claro que esta primeira conclusão não é definitiva, no sentido de que o que acabamos de chamar de **tempo razoável** pode ser melhorado ainda mais, juntamente com as soluções geradas.

Depois de ter desenvolvido este método, depois de analisados os seus resultados, parece interessante recomendar algumas alterações que podem melhorar o método tanto na solução como no tempo de CPU utilizado.

A primeira alteração sugerida é para redefinir o método utilizado para gerar listas Tipo I, já que o procedimento utilizado atualmente não explora a possibilidade de que existam vários fragmentos, com fluxo zero, com um segmento comum a todos eles. Por exemplo, na Figura (V.1), para o arco (i,j) consideramos o segmento do vértice i, k-i, cujo peso deve ser compensado por outros arcos iniciados em vértices do prórpio segmento k-i, e não consideramos os arcos iniciados em vértices do caminho u-v que, em alguns casos, poderiam contribuir para compensar o peso do segmento k-i. Também poderiamos considerar a possibilidade de substituir arcos numa lista se isto contribui para diminuir ainda mais o peso de uma arborescência geradora. Isto poderia acontecer quando dois arcos iniciados em vértices, iguais ou não, pertencentes a um mesmo fragmento são incidentes a vértices pertencentes, também, a um mesmo fragmento tal que o vértice final do arco já existente na lista é alcançado pelo vértice final do arco que está sendo testado. Atualmente, tal arco seria sumariamente rejeitado sem considerar a possibilidade de este substituir o anterior.

Por outro lado, como a costrução de uma lista Tipo I está diretamente relacionada a um arco iniciado num fragmento de fluxo zero, pode acontecer que um mesmo teste, com resultado desfavorável, seja procesado múltiplas vezes.

Parece bastante claro que poderiamos melhorar o tempo computacional e, em alguns casos, a solução obtida se, ao invés de construir as listas Tipo I em torno de cada arco iniciado num fragmento de fluxo zero, optamos por construir estas listas em torno de arcos, na arborescência, que representem o arco inicial de um fragmento de fluxo zero, e tentar aproveitar tais fragmentos.

Isto nos levaria à necessidade de fazer algumas alterações na contrução das listas Tipo III e, por outro lado, permitiria a consideração de ciclos nas listas Tipo I, desde que, em conjunto com os demais arcos da lista, tais ciclos sejam destruidos.

Nas listas Tipo II, poderiamos ir mais longe e fazer alguns testes, de aparente custo computacional baixo, para

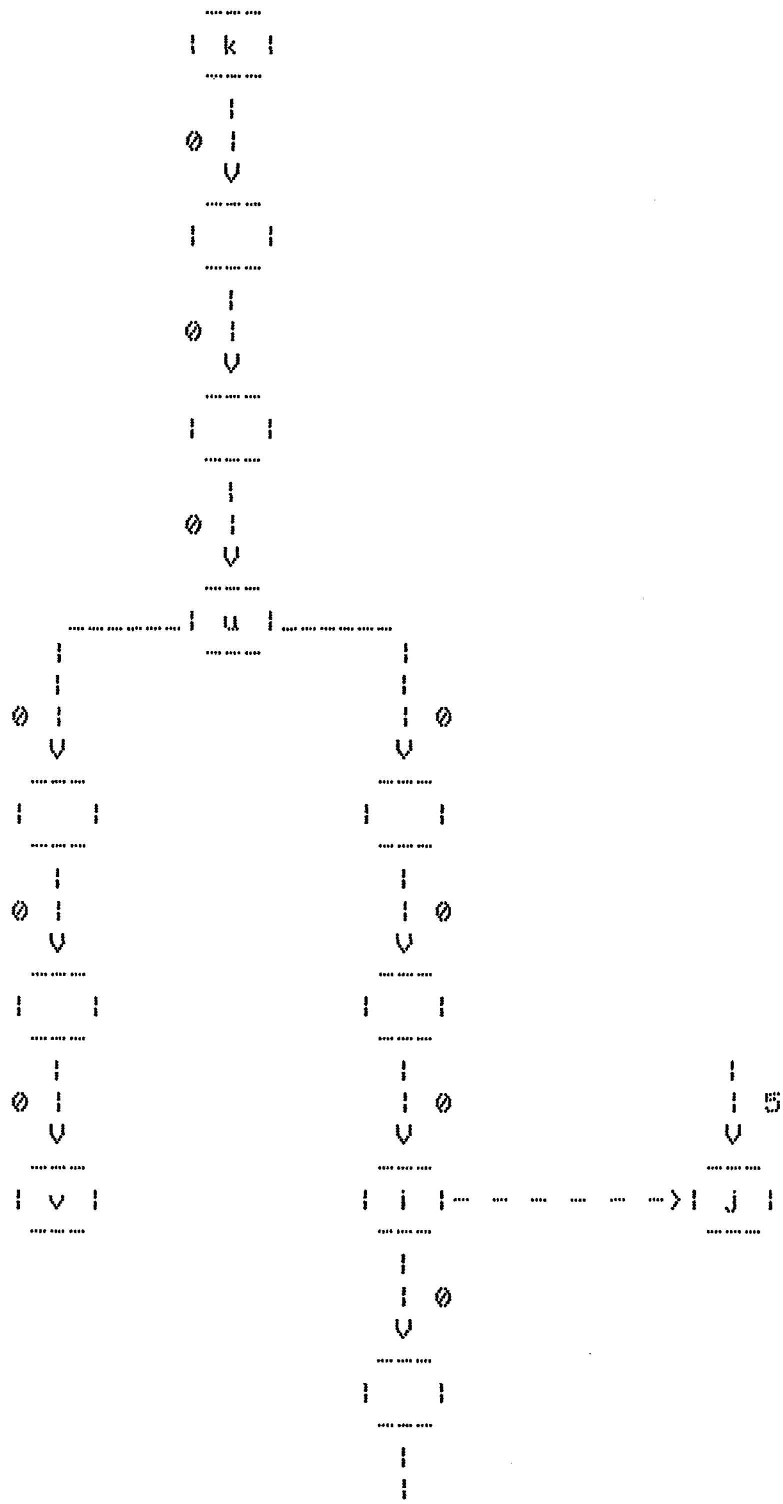


Figura V.1

verificar a liberação, conveniente ou não, de um fragmento terminado num vértice optativo, já que poderia existir, em alguns casos, uma interdependência entre duas, ou mais, estruturas deste tipo, de tal forma que em conjunto consigam diminuir o peso de uma arborescência geradora. Isto pode acontecer quando existe um fragmento cujos vértices inicial e final correspondem a vértices optativos. No exemplo da figura

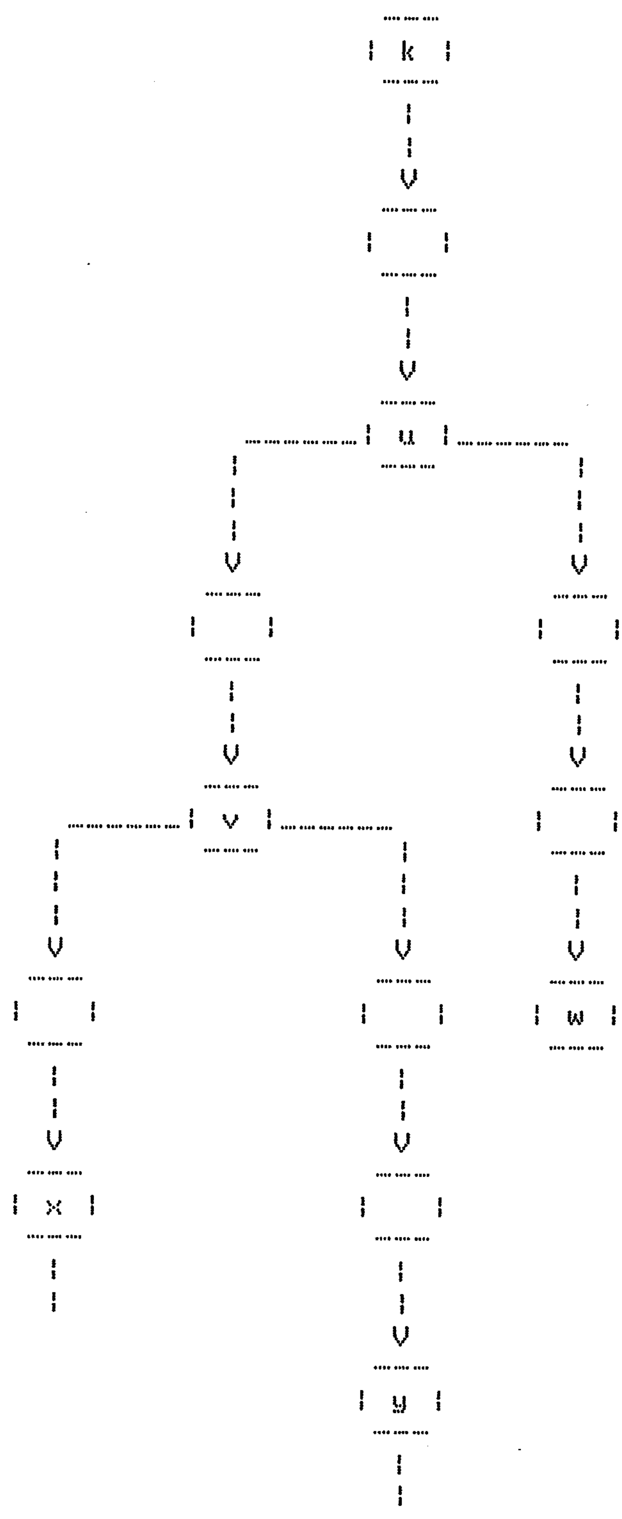


Figura V.2

(V.2), onde os vértices u e v são optativos, podemos liberar o uso do fragmento k-u mediante a substituição do fluxo em segmentos dos fragmentos u-v e u-w, ou, indiretamente, mediante a substituição do fluxo em segmentos dos fragmentos v-x, v-y e u-w. Isto se deve a que a substituição do fluxo em segmentos dos fragmentos v-x e v-y implica na liberação do fragmento u-v, já que o fluxo deste fragmento foi substituido.

BIBLIOGRAFIA

[1] **ANEJA, Y.P.**, "An Integer Linear Programming Approach to the Steiner Problem in Graphs", Networks, vol. 10, pp. 167-178, (1980).

[2] **ARPIN, D., MACULAN, N. e NGUYEN, S.**, "Le Problème de Steiner sur un Graphe Orienté: Formulations et Relaxations", Publication #315, Centre de Recherche sur les Transports, Université de Montréal, (1983).

[3] **BEASLEY, J.E.**, "An Algorithm for the Steiner Problem in Graphs", Research Report, Department of Management Science, Imperial College, London, England, (1982).

[4] **RESENDE, M. e CAMPELLO, R.**, "Software para Programação Linear de Médio Porte: Desenvolvimento e Implementação", Informe Técnico AMM-L-001-82, Assessoria de Métodos e Modelos, Coordenação de Planejamento, Diretoria de Planejamento, FURNAS: Centrais Elétricas S.A., 1982.

[5] CLAUS, A. e MACULAN, N., "Une Nouvelle Formulation du Problème de Steiner sur un Graphe", Publication #280, Centre de Recherche sur les Tranports, Université de Montréal, (1983).

[6] DREYFUS, S.E. e WAGNER, R.A., "The Steiner Problem in Graphs", Networks, vol. 1, pp. 195-207, (1972).

[7] HAKIMI, S.L., "Steiner's Problem in Graphs and its implications", Networks, vol. 1, pp. 113-133, (1971).

[8] LAWLER, E.L., Combinatorial Optimization: Networks and Matroids, New York, Holt, Rinehart and Winston, (1976).

[9] SCHRAGE, L., "Implicit Representation of Generalized Variable Upper Bounds in Linear Programming", Mathematical Programming, vol. 14, pp. 11-20, (1978).

[10] WONG, R.T., "A Dual Ascent Approach for Steiner Tree Problems on a Directed Graph", Mathematical Programming, vol. 28, pp. 271-287, (1984).

APÊNDICE A

QUADROS CONTENDO OS RESULTADOS OBTIDOS, NA RESOLUÇÃO DOS PROBLEMAS DE TESTE, PELA VERSÃO INDIVIDUAL DA NOVA HEURÍSTICA

Prob No.	Vert. obr.	Solução	Tempo CPU
1	10	214	1,160
2	10	157	1,774
3	10	254	2,230
4	10	166	1,280
5	10	190	1,640
6	10	143	1,159
7	10	236	1,260
8	10	115	1,035
9	10	235	1,744
10	10	206	1,450
11	20	269	1,221
12	20	282	1,831
13	20	274	2,076
14	20	282	1,652
15	20	324	1,785
16	20	298	1,719
17	20	332	1,229
18	20	341	1,995
19	20	245	1,332
20	20	282	1,502

Tabela A.1.- Problemas com 100 vértices e 500 arcos.

Prob No.	Vert. obr.	Solução	Tempo CPU
21	30	354	1,667
22	30	350	1,316
23	30	339	1,436
24	30	549	2,377
25	30	433	1,806
26	30	384	1,196
27	30	490	1,323
28	30	414	1,191
29	30	516	1,159
30	30	462	1,387
31	40	484	1,429
32	40	564	2,632
33	40	595	1,392
34	40	487	1,719
35	40	461	1,383
36	40	441	1,500
37	40	562	1,762
38	40	513	1,393
39	40	566	1,225
40	40	527	1,382

Tabela A.1 (Continuação)

Prob No.	Vert. obr.	Solução	Tempo CPU
41	50	653	1,463
42	50	644	1,840
43	50	755	1,678
44	50	757	1,666
45	50	634	1,475
46	50	664	1,227
47	50	643	1,253
48	50	647	1,440
49	50	671	2,299
50	50	634	1,935
51	60	688	1,416
52	60	827	1,828
53	60	684	1,398
54	60	678	1,678
55	60	717	1,229
56	60	728	1,168
57	60	828	1,710
58	60	749	1,835
59	60	593	1,519
60	60	706	1,612

Tabela A.1 (Continuação)

Prob No.	Vert. obr.	Solução	Tempo CPU
61	70	695	1,394
62	70	752	1,676
63	70	818	1,483
64	70	727	1,208
65	70	902	1,681
66	70	712	1,502
67	70	712	1,296
68	70	808	1,714
69	70	794	1,421
70	70	694	1,353
71	80	909	1,403
72	80	856	1,573
73	80	755	1,295
74	80	749	1,289
75	80	917	1,649
76	80	905	1,332
77	80	902	2,361
78	80	827	1,424
79	80	873	1,310
80	80	978	1,410

Tabela A.1 (Continuação)

Prob No.	Vert. obr.	Solução	Tempo CPU
81	90	830	1,615
82	90	920	1,515
83	90	911	1,686
84	90	960	1,510
85	90	903	1,222
86	90	925	1,319
87	90	878	1,243
88	90	933	1,555
89	90	921	1,218
90	90	881	1,160
91	100	1009	1,243
92	100	1103	1,353
93	100	896	1,214
94	100	1065	1,166
95	100	982	1,444
96	100	1068	1,425
97	100	1114	1,126
98	100	807	1,330
99	100	986	1,354
100	100	895	1,539

Tabela A.1 (Continuação)

Prob No.	Vert. obr.	Solução	Tempo CPU
1	20	403	3,831
2	40	746	5,223
3	60	901	4,428
4	80	1103	5,407
5	100	1348	5,872
6	120	1337	4,647
7	140	1703	3,775
8	160	1734	4,544
9	180	1856	3,990
10	200	1834	4,290

Tabela A.2.- Problemas com 200 vértices e 1000 arcos.

Prob No.	Vert. obr.	Solução	Tempo CPU
1	30	524	7,512
2	60	986	8,769
3	90	1161	8,401
4	120	1520	11,637
5	150	1914	14,811
6	180	2111	12,063
7	210	2467	9,946
8	240	2465	8,692
9	270	2748	10,364
10	300	2866	7,306

Tabela A.3.- Problemas com 300 vértices e 1500 arcos.

Prob No.	Vert. obr.	Solu-ção	Tempo CPU
1	60	1313	29,938
2	120	2012	35,350
3	180	2680	40,948
4	240	3287	48,922
5	300	3758	40,689
6	360	4250	37,382
7	420	4647	34,959
8	480	5187	40,365
9	540	5424	34,362
10	600	5589	33,099

Tabela A.4.- Problemas com 600 vértices e 3000 arcos.

Prob No.	Vert. obr.	Solu-ção	Tempo CPU
1	90	1864	41,637
2	180	3085	76,245
3	270	4108	78,022
4	360	4908	71,431
5	450	5914	99,980
6	540	6166	81,969
7	630	6900	86,642
8	720	7741	91,037
9	810	8012	75,607
10	900	8633	68,075

Tabela A.5.- Problemas com 900 vértices e 4500 arcos.

APÊNDICE B

QUADROS CONTENDO OS RESULTADOS OBTIDOS, NA RESOLUÇÃO DOS PROBLEMAS DE TESTE, PELA VERSÃO HÍBRIDA DA NOVA HEURÍSTICA

Prob No.	Vert. obr.	HEURÍSTICA DE WONG: Limites infer.	HEURÍSTICA DE WONG: Limites super.	HEURÍSTICA DE WONG: Tempo CPU	NOVA HEURIST.: Solução	NOVA HEURIST.: Tempo CPU	CPU Total
1	10	211	228	0,258	214	0,631	0,889
2	10	155	160	0,425	157	0,908	1,333
3	10	246	251	0,259	249	0,621	0,880
4	10	166	166	0,212	166	0,724	0,936
5	10	179	198	0,351	184	0,829	1,180
6	10	141	143	0,187	143	0,598	0,785
7	10	228	231	0,241	231	0,548	0,789
8	10	115	115	0,191	115	0,587	0,778
9	10	226	235	0,303	235	0,902	1,205
10	10	206	234	0,607	206	0,962	1,569
11	20	265	276	0,304	269	0,525	0,829
12	20	282	282	0,309	282	0,583	0,892
13	20	273	274	0,335	274	0,715	1,050
14	20	268	287	0,441	274	0,889	1,330
15	20	324	324	0,369	324	0,626	0,995
16	20	284	298	0,314	298	0,751	1,065
17	20	314	328	0,390	320	0,613	1,003
18	20	331	331	0,288	331	0,584	0,872
19	20	240	245	0,241	245	0,751	0,992
20	20	274	282	0,401	279	0,618	1,019

Tabela B.6.- Problemas com 100 vértices e 500 arcos.

Prob No.	Vert. obr.	HEURÍSTICA DE WONG: L i m i t e s infer.	super.	Tempo CPU	NOVA HEURIST.: Solu- ção	Tempo CPU	CPU Total
21	30	353	354	0,354	354	0,540	0,894
22	30	346	352	0,391	348	0,765	1,156
23	30	337	348	0,404	339	0,635	1,039
24	30	536	586	0,602	549	1,756	2,358
25	30	430	433	0,441	433	0,620	1,061
26	30	374	385	0,377	384	0,532	0,909
27	30	457	467	0,629	463	1,461	2,090
28	30	414	414	0,376	414	0,595	0,971
29	30	512	524	0,443	516	0,492	0,935
30	30	453	455	0,387	455	0,620	1,007
31	40	484	484	0,430	484	0,512	0,942
32	40	538	575	0,586	571	1,083	1,669
33	40	583	599	0,622	595	0,791	1,413
34	40	477	488	0,494	487	0,597	1,091
35	40	461	462	0,455	461	0,628	1,083
36	40	437	457	0,482	441	0,565	1,047
37	40	551	562	0,551	555	0,895	1,446
38	40	495	512	0,531	512	0,597	1,128
39	40	563	573	0,449	566	0,667	1,116
40	40	519	533	0,478	527	0,589	1,067

Tabela B.6 (Continuação)

		HEURÍSTICA DE WONG			NOVA HEURIST.		
Prob No.	Vert. obr.	L i m i t e s infer.	super.	Tempo CPU	Solu-ção	Tempo CPU	CPU Total
41	50	629	635	0,604	635	0,627	1,231
42	50	631	642	0,744	635	0,589	1,333
43	50	751	764	0,564	755	0,604	1,168
44	50	748	763	0,582	760	1,095	1,677
45	50	631	651	1,353	633	0,836	2,189
46	50	664	664	0,541	664	0,555	1,096
47	50	643	643	0,509	643	0,641	1,150
48	50	646	646	0,621	646	0,493	1,114
49	50	665	674	0,605	671	0,677	1,282
50	50	634	637	0,555	634	0,872	1,427
51	60	684	692	0,627	688	0,560	1,187
52	60	803	809	0,660	803	0,641	1,301
53	60	684	684	0,650	684	0,496	1,146
54	60	674	685	0,699	678	0,554	1,253
55	60	717	717	0,592	717	0,501	1,093
56	60	714	714	0,573	714	0,510	1,083
57	60	823	830	0,742	830	0,694	1,436
58	60	742	759	0,701	749	0,858	1,559
59	60	593	593	0,624	593	0,510	1,134
60	60	706	706	0,589	706	0,516	1,105

Tabela B.6 (Continuação)

Prob No.	Vert. obr.	HEURÍSTICA DE WONG Limites infer.	super.	Tempo CPU	NOVA HEURIST. Solução	Tempo CPU	CPU Total
61	70	695	696	0,629	695	0,554	1,183
62	70	752	752	0,675	752	0,443	1,118
63	70	818	836	0,720	818	0,516	1,236
64	70	721	729	0,697	726	0,605	1,302
65	70	890	898	0,897	893	0,504	1,401
66	70	712	712	0,698	712	0,526	1,224
67	70	712	712	0,634	712	0,507	1,141
68	70	808	810	0,733	808	0,806	1,539
69	70	780	794	0,752	794	0,441	1,193
70	70	694	694	0,675	694	0,496	1,171
71	80	909	909	0,763	909	0,507	1,270
72	80	856	856	0,913	856	0,461	1,374
73	80	755	755	0,745	755	0,467	1,212
74	80	746	749	0,767	749	0,444	1,211
75	80	917	917	0,965	917	0,458	1,423
76	80	904	912	0,934	904	0,765	1,699
77	80	901	905	0,969	903	0,853	1,822
78	80	822	826	0,791	826	0,495	1,286
79	80	873	873	0,741	873	0,549	1,290
80	80	977	978	0,751	978	0,549	1,300

Tabela B.6 (Continuação)

Prob No.	Vert. obr.	HEURÍSTICA DE WONG: Limites infer.	HEURÍSTICA DE WONG: Limites super.	HEURÍSTICA DE WONG: Tempo CPU	NOVA HEURIST.: Solução	NOVA HEURIST.: Tempo CPU	CPU Total
81	90	821	831	1,201	830	0,761	1,962
82	90	920	920	0,930	920	0,425	1,355
83	90	911	911	0,887	911	0,449	1,336
84	90	960	960	0,821	960	0,484	1,305
85	90	903	903	0,804	903	0,432	1,236
86	90	925	925	0,829	925	0,432	1,261
87	90	878	878	0,833	878	0,446	1,279
88	90	933	933	0,894	933	0,475	1,369
89	90	901	904	1,193	901	0,647	1,840
90	90	881	881	0,883	881	0,410	1,293
91	100	1009	1009	0,905	1009	0,273	1,178
92	100	1103	1103	0,893	1103	0,281	1,174
93	100	896	896	1,038	896	0,279	1,317
94	100	1065	1065	0,891	1065	0,267	1,158
95	100	975	975	1,170	975	0,308	1,478
96	100	1068	1068	0,932	1068	0,289	1,221
97	100	1110	1110	1,107	1110	0,303	1,410
98	100	807	807	0,905	807	0,271	1,176
99	100	986	986	0,905	986	0,279	1,184
100	100	895	895	0,953	895	0,306	1,259

Tabela B.6 (Continuação)

Prob No.	Vert. obr.	HEURÍSTICA DE WONG: L i m i t e s infer.	HEURÍSTICA DE WONG: L i m i t e s super.	HEURÍSTICA DE WONG: Tempo CPU	NOVA HEURIST.: Solu- ção	NOVA HEURIST.: Tempo CPU	CPU Total
1	20	369	435	2,052	399	1,605	3,657
2	40	718	739	1,165	725	2,149	3,314
3	60	885	887	1,279	887	1,527	2,806
4	80	1084	1090	2,171	1088	2,051	4,222
5	100	1329	1371	2,242	1348	2,500	4,742
6	120	1331	1337	2,302	1337	1,396	3,698
7	140	1700	1709	2,600	1703	1,344	3,944
8	160	1728	1728	2,993	1728	1,215	4,208
9	180	1856	1856	3,210	1856	1,019	4,229
10	200	1834	1834	3,845	1834	0,588	4,433

Tabela B.7.- Problemas com 200 vértices e 1000 arcos.

Prob No.	Vert. obr.	HEURÍSTICA DE WONG: L i m i t e s infer.	HEURÍSTICA DE WONG: L i m i t e s super.	HEURÍSTICA DE WONG: Tempo CPU	NOVA HEURIST.: Solu- ção	NOVA HEURIST.: Tempo CPU	CPU Total
1	30	510	530	1,775	523	3,013	4,788
2	60	946	1001	2,866	989	4,410	7,276
3	90	1147	1195	3,228	1161	2,562	5,790
4	120	1496	1541	4,062	1519	3,191	7,253
5	150	1896	1945	4,744	1914	3,243	7,987
6	180	2100	2118	4,967	2103	2,062	7,029
7	210	2462	2467	5,401	2467	1,657	7,058
8	240	2464	2476	6,655	2468	2,271	8,926
9	270	2744	2758	7,722	2747	2,695	10,417
10	300	2866	2866	7,204	2866	0,872	8,076

Tabela B.8.- Problemas com 300 vértices e 1500 arcos.

Prob No.	Vert. obr.	HEURÍSTICA DE WONG: Limites infer.	HEURÍSTICA DE WONG: Limites super.	HEURÍSTICA DE WONG: Tempo CPU	NOVA HEURIST.: Solução	NOVA HEURIST.: Tempo CPU	CPU Total
1	60	1225	1390	6,690	1324	10,652	17,342
2	120	1955	2038	9,449	2006	5,429	14,878
3	180	2602	2698	12,069	2651	14,269	26,338
4	240	3244	3319	15,145	3281	9,093	24,238
5	300	3749	3785	18,372	3758	5,146	23,518
6	360	4202	4228	19,583	4225	5,150	24,733
7	420	4629	4648	20,136	4643	4,017	24,153
8	480	5186	5186	22,532	5186	4,204	26,736
9	540	5424	5438	38,004	5424	3,988	41,992
10	600	5589	5589	27,925	5589	2,105	30,030

Tabela B.9.- Problemas com 600 vértices e 3000 arcos.

Prob No.	Vert. obr.	HEURÍSTICA DE WONG: Limites infer.	HEURÍSTICA DE WONG: Limites super.	HEURÍSTICA DE WONG: Tempo CPU	NOVA HEURIST.: Solução	NOVA HEURIST.: Tempo CPU	CPU Total
1	90	1722	1916	14,604	1831	16,859	31,463
2	180	2980	3124	20,310	3049	32,701	53,011
3	270	4013	4163	27,002	4090	19,348	46,350
4	360	4888	4944	29,710	4902	12,351	42,061
5	450	5883	5947	37,296	5909	21,383	58,679
6	540	6153	6176	39,901	6163	10,942	50,843
7	630	6883	6904	45,584	6888	6,419	52,003
8	720	7720	7737	51,941	7734	6,971	58,912
9	810	8012	8013	59,739	8012	6,877	66,616
10	900	8619	8619	114,087	8619	6,662	120,749

Tabela B.10.- Problemas com 900 vértices e 4500 arcos.

APÊNDICE C

ÍNDICES DE VARIAÇÃO DE INTERVALOS

A heurística de Wong, numa primeira etapa, gera um valor que representa um limite inferior para o valor da solução ótima do problema de Steiner. Posteriormente, permite obter uma solução aproximada cujo valor, que pode ser considerado um limite superior, junto com o limite inferior define um intervalo no qual se encontra o valor da solução ótima do problema de Steiner. Ao utilizar a nova heurística, nas duas versões, temos novas soluções aproximadas cujos valores podem ser considerados como limites superiores, cada um dos quais, junto com o limite inferior gerado pela heurística de Wong, nos dá novos intervalos para a solução ótima do problema de Steiner.

Nos quadros gerados a seguir, temos, para cada problema, os limites inferior e superior obtidos pela heurística de Wong, os limites superiores obtidos pelas duas versões da nova heurística, e as percentagens de diminuição ou de aumento dos novos intervalos em relação ao intervalo fornecido pela heurística de Wong.

HEURÍSTICA DE WONG			PRIMEIRA VERSÃO			SEGUNDA VERSÃO	
Prob. No.	lim. inf.	lim. sup.	lim. sup.	% dimi- nuição	% au- mento	lim. sup.	% dimi- nuição
1	211	228	214	82,35	------	214	82,35
2	155	160	157	60,00	------	157	60,00
3	246	251	254	------	60,00	249	40,00
4	166	166	166	0,00	0,00	166	0,00
5	179	198	190	42,11	------	184	73,68
6	141	143	143	0,00	0,00	143	0,00
7	228	231	236	------	166,67	231	0,00
8	115	115	115	0,00	0,00	115	0,00
9	226	235	235	0,00	0,00	235	0,00
10	206	234	206	100,00	------	206	100,00
11	265	276	269	63,64	------	269	63,64
12	282	282	282	0,00	0,00	282	0,00
13	273	274	274	0,00	0,00	274	0,00
14	268	287	282	26,32	------	274	68,42
15	324	324	324	0,00	0,00	324	0,00
16	284	298	298	0,00	0,00	298	0,00
17	314	328	332	------	28,57	320	57,14
18	331	331	341	------	******	331	0,00
19	240	245	245	0,00	0,00	245	0,00
20	274	282	282	0,00	0,00	279	37,50

Tabela C.1.- Problemas com 100 vértices e 500 arcos.

Nota: Nas informações referentes às percentagens, os valores substituidos por traços indicam inexistência dos mesmos, e os asteriscos representam valores infinitamente grandes.

HEURÍSTICA DE WONG			PRIMEIRA VERSÃO			SEGUNDA VERSÃO	
Prob. No.	lim. inf.	lim. sup.	lim. sup.	% dimi- nuição	% au- mento	lim. sup.	% dimi- nuição
21	353	354	354	0,00	0,00	354	0,00
22	346	352	350	33,33	------	348	66,67
23	337	348	339	81,82	------	339	81,82
24	536	586	549	74,00	------	549	74,00
25	430	433	433	0,00	0,00	433	0,00
26	374	385	384	9,09	------	384	9,09
27	457	467	490	------	230,00	463	40,00
28	414	414	414	0,00	0,00	414	0,00
29	512	524	516	66,67	------	516	66,67
30	453	455	462	------	350,00	455	0,00
31	484	484	484	0,00	0,00	484	0,00
32	538	575	564	29,73	------	571	10,81
33	583	599	595	25,00	------	595	25,00
34	477	488	487	9,09	------	487	9,09
35	461	462	461	100,00	------	461	100,00
36	437	457	441	80,00	------	441	80,00
37	551	562	562	0,00	0,00	555	63,64
38	495	512	513	------	5,88	512	0,00
39	563	573	566	70,00	------	566	70,00
40	519	533	527	42,86	------	527	42,86

Tabela C.1. (Continuação)

HEURÍSTICA DE WONG			PRIMEIRA VERSÃO			SEGUNDA VERSÃO	
Prob. No.	lim. inf.	lim. sup.	lim. sup.	% dimi- nuição	% au- mento	lim. sup.	% dimi- nuição
41	629	635	653	------	300,00	635	0,00
42	631	642	644	------	18,18	635	63,64
43	751	764	755	69,23	------	755	69,23
44	748	763	757	40,00	------	760	20,00
45	631	651	634	85,00	------	633	90,00
46	664	664	664	0,00	0,00	664	0,00
47	643	643	643	0,00	0,00	643	0,00
48	646	646	647	------	******	646	0,00
49	665	674	671	33,33	------	671	33,33
50	634	637	634	100,00	------	634	100,00
51	684	692	688	50,00	------	688	50,00
52	803	809	827	------	300,00	803	100,00
53	684	684	684	0,00	0,00	684	0,00
54	674	685	678	63,64	------	678	63,64
55	717	717	717	0,00	0,00	717	0,00
56	714	714	728	------	******	714	0,00
57	823	830	828	28,57	------	830	0,00
58	742	759	749	58,82	------	749	58,82
59	593	593	593	0,00	0,00	593	0,00
60	706	706	706	0,00	0,00	706	0,00

Tabela C.1. (Continuação)

HEURÍSTICA DE WONG			PRIMEIRA VERSÃO			SEGUNDA VERSÃO	
Prob. No.	lim. inf.	lim. sup.	lim. sup.	% dimi-nuição	% au-mento	lim. sup.	% dimi-nuição
61	695	696	695	100,00	------	695	100,00
62	752	752	752	0,00	0,00	752	0,00
63	818	836	818	100,00	------	818	100,00
64	721	729	727	25,00	------	726	37,50
65	890	898	902	------	50,00	893	62,50
66	712	712	712	0,00	0,00	712	0,00
67	712	712	712	0,00	0,00	712	0,00
68	808	810	808	100,00	------	808	100,00
69	780	794	794	0,00	0,00	794	0,00
70	694	694	694	0,00	0,00	694	0,00
71	909	909	909	0,00	0,00	909	0,00
72	856	856	856	0,00	0,00	856	0,00
73	755	755	755	0,00	0,00	755	0,00
74	746	749	749	0,00	0,00	749	0,00
75	917	917	917	0,00	0,00	917	0,00
76	904	912	905	87,50	------	904	100,00
77	901	905	902	75,00	------	903	50,00
78	822	826	827	------	25,00	826	0,00
79	873	873	873	0,00	0,00	873	0,00
80	977	978	978	0,00	0,00	978	0,00

Tabela C.1. (Continuação)

HEURÍSTICA DE WONG			PRIMEIRA VERSÃO			SEGUNDA VERSÃO	
Prob. No.	lim. inf.	lim. sup.	lim. sup.	% dimi- nuição	% au- mento	lim. sup.	% dimi- nuição
81	821	831	830	10,00	--------	830	10,00
82	920	920	920	0,00	0,00	920	0,00
83	911	911	911	0,00	0,00	911	0,00
84	960	960	960	0,00	0,00	960	0,00
85	903	903	903	0,00	0,00	903	0,00
86	925	925	925	0,00	0,00	925	0,00
87	878	878	878	0,00	0,00	878	0,00
88	933	933	933	0,00	0,00	933	0,00
89	901	904	921	--------	566,67	901	100,00
90	881	881	881	0,00	0,00	881	0,00
91	1009	1009	1009	0,00	0,00	1009	0,00
92	1103	1103	1103	0,00	0,00	1103	0,00
93	896	896	896	0,00	0,00	896	0,00
94	1065	1065	1065	0,00	0,00	1065	0,00
95	975	975	982	--------	******	975	0,00
96	1068	1068	1068	0,00	0,00	1068	0,00
97	1110	1110	1114	--------	******	1110	0,00
98	807	807	807	0,00	0,00	807	0,00
99	986	986	986	0,00	0,00	986	0,00
100	895	895	895	0,00	0,00	895	0,00

Tabela C.1. (Continuação)

HEURÍSTICA DE WONG			PRIMEIRA VERSÃO			SEGUNDA VERSÃO	
Prob. No.	lim. inf.	lim. sup.	lim. sup.	% dimi-nuição	% au-mento	lim. sup.	% dimi-nuição
1	369	435	403	48,48	------	399	54,55
2	718	739	746	------	33,33	725	66,67
3	885	887	901	------	700,00	887	0,00
4	1084	1090	1103	------	216,67	1088	33,33
5	1329	1371	1348	54,76	------	1348	54,76
6	1331	1337	1337	0,00	0,00	1337	0,00
7	1700	1709	1703	66,67	------	1703	66,67
8	1728	1728	1734	------	******	1728	0,00
9	1856	1856	1856	0,00	0,00	1856	0,00
10	1834	1834	1834	0,00	0,00	1834	0,00

Tabela C.2.- Problemas com 200 vértices e 1000 arcos.

HEURÍSTICA DE WONG			PRIMEIRA VERSÃO			SEGUNDA VERSÃO	
Prob. No.	lim. inf.	lim. sup.	lim. sup.	% dimi-nuição	% au-mento	lim. sup.	% dimi-nuição
1	510	530	524	30,00	------	523	35,00
2	946	1001	986	27,27	------	989	21,82
3	1147	1195	1161	70,83	------	1161	70,83
4	1496	1541	1520	46,67	------	1519	48,89
5	1896	1945	1914	63,27	------	1914	63,27
6	2100	2118	2111	38,99	------	2103	83,33
7	2462	2467	2467	0,00	0,00	2467	0,00
8	2464	2476	2465	91,67	------	2468	66,67
9	2744	2758	2748	71,43	------	2747	78,57
10	2866	2866	2866	0,00	0,00	2866	0,00

Tabela C.3.- Problemas com 300 vértices e 1500 arcos.

HEURÍSTICA DE WONG			PRIMEIRA VERSÃO			SEGUNDA VERSÃO	
Prob. No.	lim. inf.	lim. sup.	lim. sup.	% diminuição	% aumento	lim. sup.	% diminuição
1	1225	1390	1313	46,67	------	1324	40,00
2	1955	2038	2012	31,33	------	2006	38,55
3	2602	2698	2680	18,75	------	2651	48,96
4	3244	3319	3287	42,67	------	3281	50,67
5	3749	3785	3758	75,00	------	3758	75,00
6	4202	4228	4250	------	84,62	4225	11,54
7	4629	4648	4647	5,26	------	4643	26,32
8	5186	5186	5187	------	******	5186	0,00
9	5424	5438	5424	100,00	------	5424	100,00
10	5589	5589	5589	0,00	0,00	5589	0,00

Tabela C.4.- Problemas com 600 vértices e 3000 arcos.

HEURÍSTICA DE WONG			PRIMEIRA VERSÃO			SEGUNDA VERSÃO	
Prob. No.	lim. inf.	lim. sup.	lim. sup.	% diminuição	% aumento	lim. sup.	% diminuição
1	1722	1916	1864	26,80	------	1831	43,81
2	2980	3124	3085	27,08	------	3049	52,08
3	4013	4163	4108	36,67	------	4090	48,67
4	4888	4944	4908	64,29	------	4902	75,00
5	5883	5947	5914	51,56	------	5909	59,38
6	6153	6176	6166	43,88	------	6163	56,52
7	6883	6904	6900	19,05	------	6888	76,19
8	7720	7737	7741	------	23,53	7734	17,65
9	8012	8013	8012	100,00	------	8012	100,00
10	8619	8619	8633	------	******	8619	0,00

Tabela C.4.- Problemas com 900 vértices e 4500 arcos.

APÊNDICE D

EXEMPLO

D.1 - DESCRIÇÃO DO PROBLEMA

Consideremos o grafo G da figura (D.1), onde os valores junto a cada arco representam o peso associado a eles. Pede-se resolver o problema de Steiner, no grafo direcionado G, considerando os seguintes vértices obrigatórios: 1, 4, 5, 7, 9, e 11.

D.2 - MÉTODO DUAL ASCENDENTE

Utilizando o método Dual Ascendente, devido a Richard T. Wong, podemos determinar um limite inferior para o valor da solução ótima do problema definido anteriormente.

Nos quadros que serão mostrados a seguir, em cada uma das iterações, no correspondente ao grafo G', a coluna identificada como **margem** representa o valor S(i*,j*) associado ao arco (i*,j*) escolhido para ser incluido no grafo auxiliar G'.

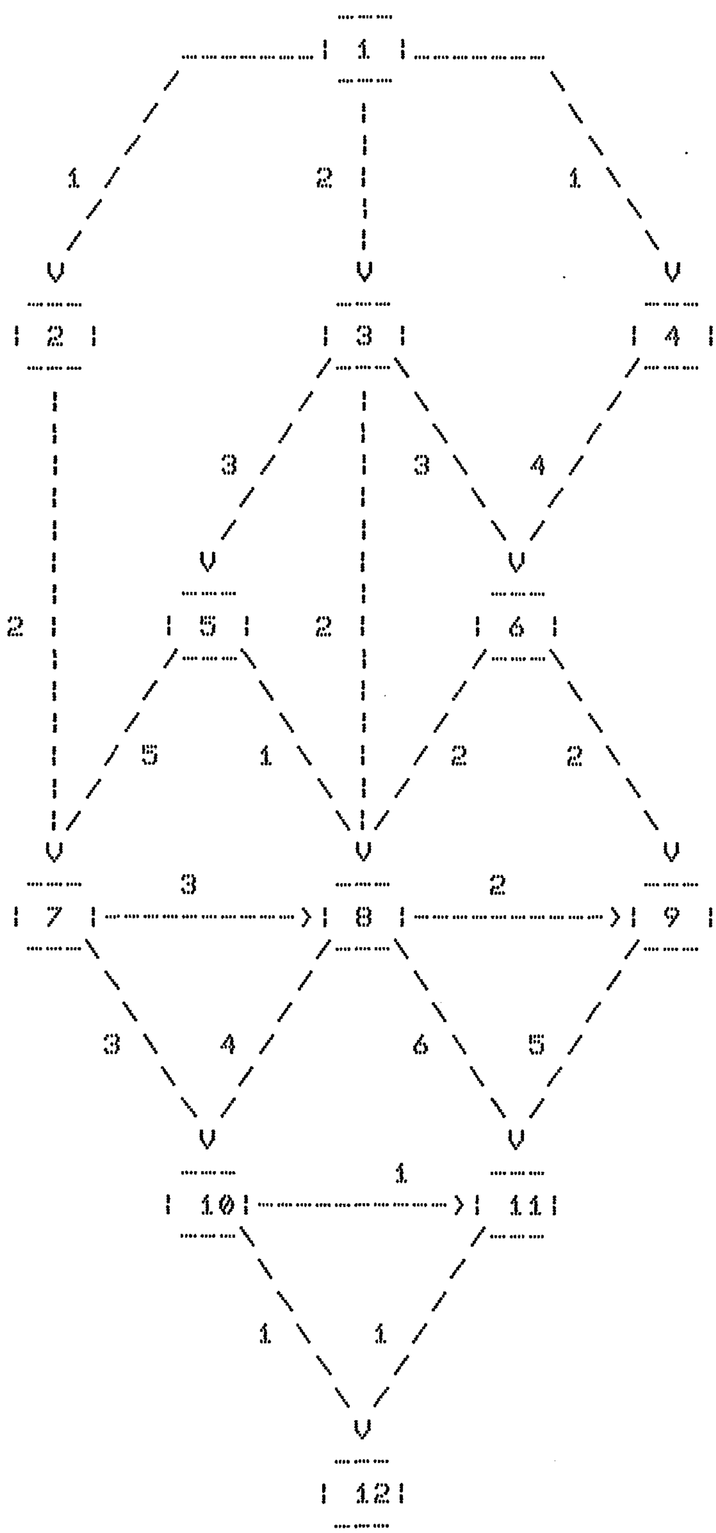


Fig. D.1

INICIALIZAÇÃO

Passo 0: Consideremos, inicialmente, no grafo original, o valor S(i,j) = C(i,j) associado a cada arco. Consideremos também o grafo auxiliar G' com todos os vértices do grafo original mas sem nenhum arco. Assim, temos a seguinte composição de arcos nos grafos G e G':

G'

arco	margem

G

arco	S(i,j)	arco	S(i,j)
1- 2	1	6- 9	2
1- 3	2	7- 8	3
1- 4	1	7-10	3
2- 7	2	8- 9	2
3- 5	3	8-10	4
3- 6	3	8-11	6
3- 8	2	9-11	5
4- 6	4	10-11	1
5- 7	5	10-12	1
5- 8	1	11-12	1
6- 8	2		

Inicialmente, LI = 0.

ITERAÇÃO 1.

Passo 1: Escolhemos uma componente fortemente conexa que contenha pelo menos um vértice obrigatório e que não seja alcançada por nenhum outro vértice obrigatório, por exemplo, a componente formada só pelo vértice 4.

Passo 2: O vértice obrigatório é k = 4. O corte do vértice 4 está formado pelo único arco (1,4). Então S(i*,j*) = S(1,4) = 1. Fazer S(1,4) = 0 e LI = 0 + 1 = 1.

Passo 3: Os grafos G' e G atualizados ficam:

G'

arco	margem
1- 4	1

G

arco	S(i,j)	arco	S(i,j)
1- 2	1	6- 9	2
1- 3	2	7- 8	3
1- 4	0	7-10	3
2- 7	2	8- 9	2
3- 5	3	8-10	4
3- 6	3	8-11	6
3- 8	2	9-11	5
4- 6	4	10-11	1
5- 7	5	10-12	1
5- 8	1	11-12	1
6- 8	2		

ITERAÇÃO 2.

Passo 1: Escolhemos uma componente fortemente conexa que contenha pelo menos um vértice obrigatório e que não seja alcançada por nenhum outro vértice obrigatório, por exemplo, a componente formada só pelo vértice 5.

Passo 2: O vértice obrigatório é k = 5. O corte do vértice 5 está formado pelo único arco (3,5). Então S(i*,j*) = S(3,5) = 3. Fazer S(3,5) = 0 e LI = 1 + 3 = 4.

Passo 3: Os grafos G' e G atualizados ficam:

G'

arco	margem
1- 4	1
3- 5	3

G

arco	S(i,j)	arco	S(i,j)
1- 2	1	6- 9	2
1- 3	2	7- 8	3
1- 4	0	7-10	3
2- 7	2	8- 9	2
3- 5	0	8-10	4
3- 6	3	8-11	6
3- 8	2	9-11	5
4- 6	4	10-11	1
5- 7	5	10-12	1
5- 8	1	11-12	1
6- 8	2		

ITERAÇÃO 3.

Passo 1: Escolhemos uma componente fortemente conexa que contenha pelo menos um vértice obrigatório e que não seja alcançada por nenhum outro vértice obrigatório, por exemplo, a componente formada só pelo vértice 5.

Passo 2: O vértice obrigatório é k = 5. O corte do vértice 5 está formado pelo único arco (1,3). Então S(i*,j*) = S(1,3) = 2. Fazer S(1,3) = 0 e LI = 4 + 2 = 6.

Passo 3: Os grafos G' e G atualizados ficam:

G'

arco	margem
1- 4	1
3- 5	3
1- 3	2

G

arco	S(i,j)	arco	S(i,j)
1- 2	1	6- 9	2
1- 3	0	7- 8	3
1- 4	0	7-10	3
2- 7	2	8- 9	2
3- 5	0	8-10	4
3- 6	3	8-11	6
3- 8	2	9-11	5
4- 6	4	10-11	1
5- 7	5	10-12	1
5- 8	1	11-12	1
6- 8	2		

ITERAÇÃO 4.

Passo 1: Escolhemos uma componente fortemente conexa que contenha pelo menos um vértice obrigatório e que não seja alcançada por nenhum outro vértice obrigatório, por exemplo, a componente formada só pelo vértice 7.

Passo 2: O vértice obrigatório é k = 7. O corte do vértice 7 está formado pelos arcos (2,7) e (5,7). Então S(i*,j*) = min { S(2,7), S(5,7) } = S(2,7) = 2. Fazer S(2,7) = 0 e S(5,7) = 3 e LI = 6 + 2 = 8.

Passo 3: Os grafos G' e G atualizados ficam:

G'

arco	margem
1- 4	1
3- 5	3
1- 3	2
2- 7	2

G

arco	S(i,j)	arco	S(i,j)
1- 2	1	6- 9	2
1- 3	0	7- 8	3
1- 4	0	7-10	3
2- 7	0	8- 9	2
3- 5	0	8-10	4
3- 6	3	8-11	6
3- 8	2	9-11	5
4- 6	4	10-11	1
5- 7	3	10-12	1
5- 8	1	11-12	1
6- 8	2		

ITERAÇÃO 5.

Passo 1: Escolhemos uma componente fortemente conexa que contenha pelo menos um vértice obrigatório e que não seja alcançada por nenhum outro vértice obrigatório, por exemplo, a componente formada só pelo vértice 7.

Passo 2: O vértice obrigatório é k = 7. O corte do vértice 7 está formado pelos arcos (1,2) e (5,7). Então S(i*,j*) = min (S(1,2), S(5,7)) = S(1,2) = 1. Fazer S(1,2) = 0 e S(5,7) = 2 e LI = 8 + 1 = 9.

Passo 3: Os grafos G' e G atualizados ficam:

G'

arco	margem
1- 4	1
3- 5	3
1- 3	2
2- 7	2
1- 2	1

G

arco	S(i,j)	arco	S(i,j)
1- 2	0	6- 9	2
1- 3	0	7- 8	3
1- 4	0	7-10	3
2- 7	0	8- 9	2
3- 5	0	8-10	4
3- 6	3	8-11	6
3- 8	2	9-11	5
4- 6	4	10-11	1
5- 7	2	10-12	1
5- 8	1	11-12	1
6- 8	2		

ITERAÇÃO 6.

Passo 1: Escolhemos uma componente fortemente conexa que contenha pelo menos um vértice obrigatório e que não seja alcançada por nenhum outro vértice obrigatório, por exemplo, a componente formada só pelo vértice 9.

Passo 2: O vértice obrigatório é k = 9. O corte do vértice 9 está formado pelos arcos (6,9) e (8,9). Então S(i*,j*) = min { S(6,9), S(8,9) } = S(8,9) = 2. Fazer S(8,9) = 0 e S(6,9) = 0 e LI = 9 + 2 = 11.

Passo 3: Os grafos G' e G atualizados ficam:

G'

arco	margem
1- 4	1
3- 5	3
1- 3	2
2- 7	2
1- 2	1
8- 9	2

G

arco	S(i,j)	arco	S(i,j)
1- 2	0	6- 9	0
1- 3	0	7- 8	3
1- 4	0	7-10	3
2- 7	0	8- 9	0
3- 5	0	8-10	4
3- 6	3	8-11	6
3- 8	2	9-11	5
4- 6	4	10-11	1
5- 7	2	10-12	1
5- 8	1	11-12	1
6- 8	2		

ITERAÇÃO 7.

Passo 1: Escolhemos uma componente fortemente conexa que contenha pelo menos um vértice obrigatório e que não seja alcançada por nenhum outro vértice obrigatório, por exemplo, a componente formada só pelo vértice 9.

Passo 2: O vértice obrigatório é k = 9. O corte do vértice 9 está formado pelos arcos (3,8), (5,8), (6,8), (6,9), e (7,8). Então S(i*,j*) = min (S(3,8), S(5,8), S(6,8), S(6,9), S(7,8)) = S(6,9) = 0. Os valores S(i,j), correspondentes aos arcos do corte, não se alteram. Fazer LI = 11 + 0 = 11.

Passo 3: Os grafos G' e G atualizados ficam:

G'

arco	margem
1- 4	1
3- 5	3
1- 3	2
2- 7	2
1- 2	1
8- 9	2
6- 9	0

G

arco	S(i,j)	arco	S(i,j)
1- 2	0	6- 9	0
1- 3	0	7- 8	3
1- 4	0	7-10	3
2- 7	0	8- 9	0
3- 5	0	8-10	4
3- 6	3	8-11	6
3- 8	2	9-11	5
4- 6	4	10-11	1
5- 7	2	10-12	1
5- 8	1	11-12	1
6- 8	2		

ITERAÇÃO 8.

Passo 1: Escolhemos uma componente fortemente conexa que contenha pelo menos um vértice obrigatório e que não seja alcançada por nenhum outro vértice obrigatório, por exemplo, a componente formada só pelo vértice 9.

Passo 2: O vértice obrigatório é k = 9. O corte do vértice 9 está formado pelos arcos (3,8), (3,6), (4,6), (5,8), e (7,8). Então S(i*,j*) = min (S(3,8), S(3,6), S(4,6), S(5,8), S(7,8)) = S(5,8) = 1. Fazer S(3,8) = 1, S(3,6) = 2, S(4,6) = 3, S(5,8) = 0 e S(7,8) = 2 e LI = 11 + 1 = 12.

Passo 3: Os grafos G' e G atualizados ficam:

G'

arco	margem
1- 4	1
3- 5	3
1- 3	2
2- 7	2
1- 2	1
8- 9	2
6- 9	0
5- 8	1

G

arco	S(i,j)	arco	S(i,j)
1- 2	0	6- 9	0
1- 3	0	7- 8	2
1- 4	0	7-10	3
2- 7	0	8- 9	0
3- 5	0	8-10	4
3- 6	2	8-11	6
3- 8	1	9-11	5
4- 6	3	10-11	1
5- 7	2	10-12	1
5- 8	0	11-12	1
6- 8	2		

ITERAÇÃO 9.

Passo 1: Escolhemos uma componente fortemente conexa que contenha pelo menos um vértice obrigatório e que não seja alcançada por nenhum outro vértice obrigatório, por exemplo, a componente formada só pelo vértice 11.

Passo 2: O vértice obrigatório é k = 11. O corte do vértice 11 está formado pelos arcos (8,11), (9,11) e (10,11). Então S(i*,j*) = min { S(8,11), S(9,11), S(10,11) } = S(10,11) = 1. Fazer S(8,11) = 5, S(9,11) = 4 e S(10,11) = 0 e LI = 12 + 1 = 13.

Passo 3: Os grafos G' e G atualizados ficam:

G'

arco	margem
1- 4	1
3- 5	3
1- 3	2
2- 7	2
1- 2	1
8- 9	2
6- 9	0
5- 8	1
10-11	1

G

arco	S(i,j)	arco	S(i,j)
1- 2	0	6- 9	0
1- 3	0	7- 8	2
1- 4	0	7-10	3
2- 7	0	8- 9	0
3- 5	0	8-10	4
3- 6	2	8-11	5
3- 8	1	9-11	4
4- 6	3	10-11	0
5- 7	2	10-12	1
5- 8	0	11-12	1
6- 8	2		

ITERAÇÃO 10.

Passo 1: Escolhemos uma componente fortemente conexa que contenha pelo menos um vértice obrigatório e que não seja alcançada por nenhum outro vértice obrigatório, por exemplo, a componente formada só pelo vértice 11.

Passo 2: O vértice obrigatório é k = 11. O corte do vértice 11 está formado pelos arcos (7,10), (8,10), (8,11) e (9,11). Então S(i*,j*) = min { S(7,10), S(8,10), S(8,11), S(9,11) } = S(7,10) = 3. Fazer S(7,10) = 0, S(8,10) = 1, S(8,11) = 2 e S(9,11) = 1 e LI = 13 + 3 = 16.

Passo 3: Os grafos G' e G atualizados ficam:

G'

arco	margem
1- 4	1
3- 5	3
1- 3	2
2- 7	2
1- 2	1
8- 9	2
6- 9	0
5- 8	1
10-11	1
7-10	3

G

arco	S(i,j)	arco	S(i,j)
1- 2	0	6- 9	0
1- 3	0	7- 8	2
1- 4	0	7-10	0
2- 7	0	8- 9	0
3- 5	0	8-10	1
3- 6	2	8-11	2
3- 8	1	9-11	1
4- 6	3	10-11	0
5- 7	2	10-12	1
5- 8	0	11-12	1
6- 8	2		

ITERAÇÃO 11.

Passo 1: Escolhemos uma componente fortemente conexa que contenha pelo menos um vértice obrigatório e que não seja alcançada por nenhum outro vértice obrigatório. Como não

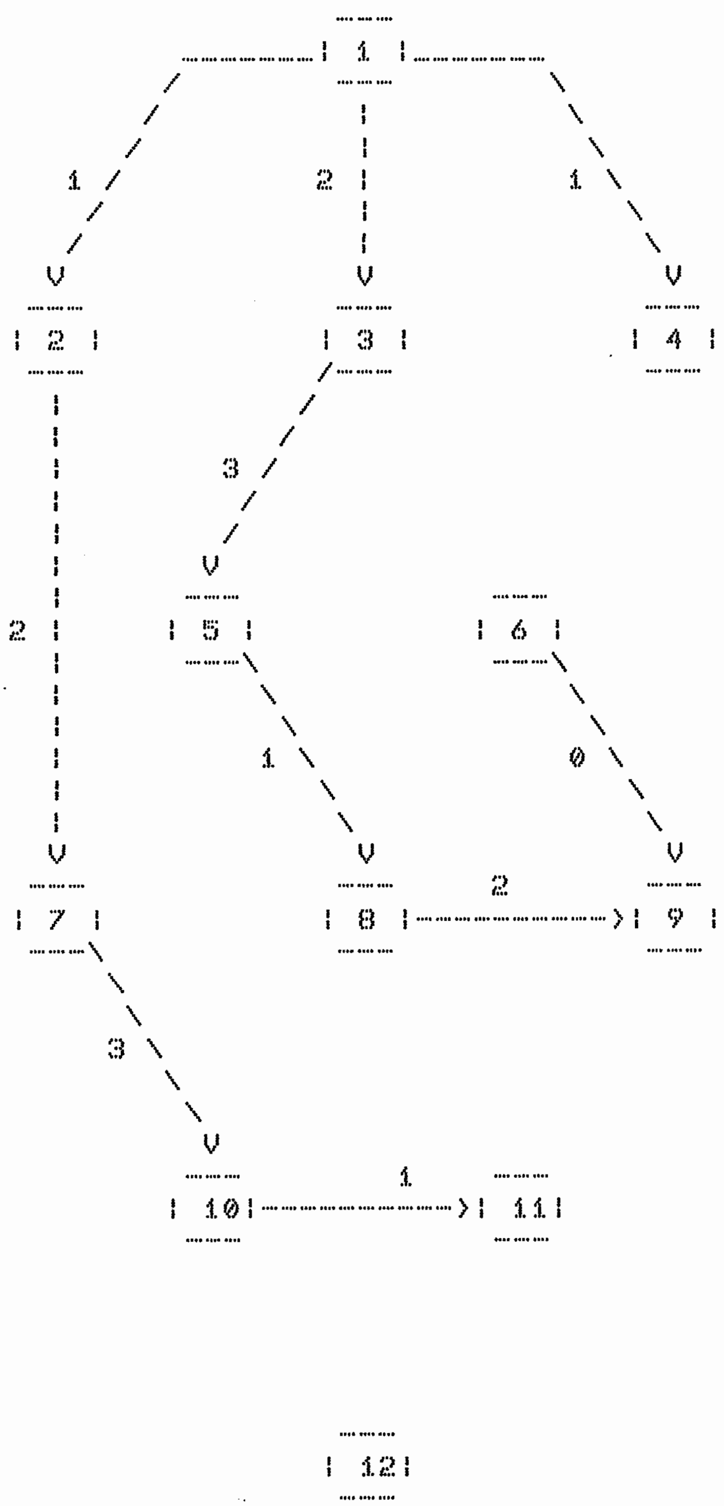


Fig. D.2

existe nenhuma componente fortemente conexa com tais características, FIM. O valor que limita inferiormente o valor da solução ótima corresponde à soma das margens de cada um dos arcos do grafo auxiliar G' e representado pelo valor de LI e ,neste caso, é igual a 16.
O grafo G', construido durante a utilização do método Dual Ascendente, pode ser observado na figura (D.2).

D.3 - HEURÍSTICA DE WONG

Para determinar o limite superior do intervalo, podemos utilizar a heurística de Wong, a qual, baseada no método Dual Ascendente, permite encontrar uma solução viável.

(1) Utilizar o método Dual Ascendente. (Consideremos o resultado obtido no Apêndice C.2)

(2) Considerando o grafo auxiliar G', obtido no Apêndice C, determinar o conjunto Q de todos os v'ertices alcançados pela raiz (vértice 1):
Q = (1, 2, 3, 4, 5, 7, 8, 9, 10, 11).
Os vértices 6 e 12 não são alcançados pela raiz.
(3) Construir uma arborescência geradora de peso mínimo com os vértices de Q, considerando todos os arcos do grafo original G entre vértices do conjunto Q, utilizando o próprio método Dual Ascendente.

Assim, o grafo G2 a ser considerado é o representado na figura (D.3) e, agora, todos os seus vértices são considerados obrigatórios.

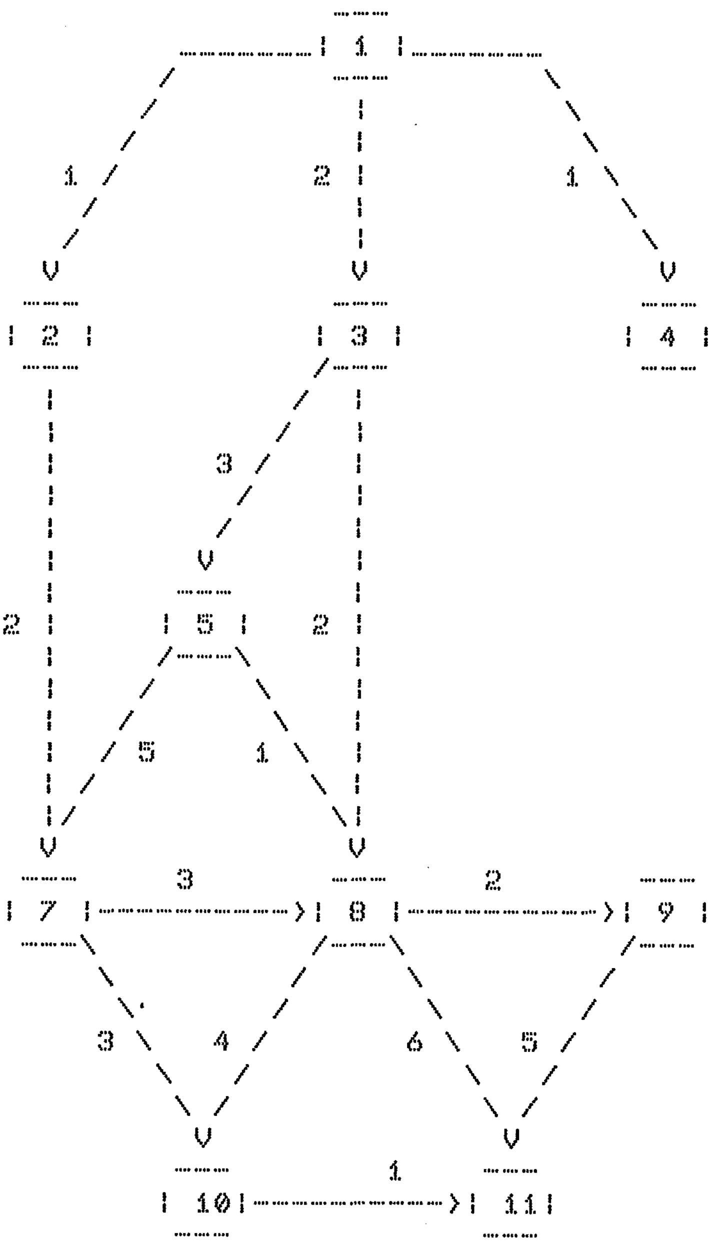


Fig. D.3.- Grafo G2.

INICIALIZAÇÃO

Passo 0: Consideremos, inicialmente, no grafo G2, o valor S(i,j) = C(i,j) associado a cada arco. Consideremos também o grafo auxiliar G″ com todos os vértices do grafo G2 mas sem nenhum arco. Assim, temos a seguinte composição de arcos nos grafos G2 e G″:

G″

arco	margem

G2

arco	S(i,j)	arco	S(i,j)
1- 2	1	7-10	3
1- 3	2	8- 9	2
1- 4	1	8-10	4
2- 7	2	8-11	6
3- 5	3	9-11	5
3- 8	2	10-11	1
5- 7	5		
5- 8	1		
7- 8	3		

Inicialmente, LI = 0.

ITERAÇÃO 1.

Passo 1: Escolhemos uma componente fortemente conexa que contenha pelo menos um vértice obrigatório e que não seja alcançada por nenhum outro vértice obrigatório, por exemplo, a componente formada só pelo vértice 2.

Passo 2: O vértice obrigatório é k = 2. O corte do vértice 2 está formado pelo único arco (1,2). Então S(i*,j*) = S(1,2) = 1. Fazer S(1,2) = 0 e LI = 0 + 1 = 1.

Passo 3: Os grafos G″ e G2 atualizados ficam:

G″

arco	margem
1- 2	1

G2

arco	S(i,j)	arco	S(i,j)
1- 2	0	7-10	3
1- 3	2	8- 9	2
1- 4	1	8-10	4
2- 7	2	8-11	6
3- 5	3	9-11	5
3- 8	2	10-11	1
5- 7	5		
5- 8	1		
7- 8	3		

ITERAÇÃO 2.

Passo 1: Escolhemos uma componente fortemente conexa que contenha pelo menos um vértice obrigatório e que não seja alcançada por nenhum outro vértice obrigatório, por exemplo, a componente formada só pelo vértice 3.

Passo 2: O vértice obrigatório é k = 3. O corte do vértice 3 está formado pelo único arco (1,3). Então S(i*,j*) = S(1,3) = 2. Fazer S(1,3) = 0 e LI = 1 + 2 = 3.

Passo 3: Os grafos G″ e G2 atualizados ficam:

G″

arco	margem
1- 2	1
1- 3	2

G2

arco	S(i,j)	arco	S(i,j)
1- 2	0	7-10	3
1- 3	0	8- 9	2
1- 4	1	8-10	4
2- 7	2	8-11	6
3- 5	3	9-11	5
3- 8	2	10-11	1
5- 7	5		
5- 8	1		
7- 8	3		

ITERAÇÃO 3.

Passo 1: Escolhemos uma componente fortemente conexa que contenha pelo menos um vértice obrigatório e que não seja alcançada por nenhum outro vértice obrigatório, por exemplo, a componente formada só pelo vértice 4.

Passo 2: O vértice obrigatório é k = 4. O corte do vértice 4 está formado pelo único arco (1,4). Então S(i*.j*) = S(1,4) = 1. Fazer S(1,4) = 0 e LI = 3 + 1 = 4.

Passo 3: Os grafos G″ e G2 atualizados ficam:

G″

arco	margem
1- 2	1
1- 3	2
1- 4	1

G2

arco	S(i,j)	arco	S(i,j)
1- 2	0	7-10	3
1- 3	0	8- 9	2
1- 4	0	8-10	4
2- 7	2	8-11	6
3- 5	3	9-11	5
3- 8	2	10-11	1
5- 7	5		
5- 8	1		
7- 8	3		

ITERAÇÃO 4.

Passo 1: Escolhemos uma componente fortemente conexa que contenha pelo menos um vértice obrigatório e que não seja alcançada por nenhum outro vértice obrigatório, por exemplo, a componente formada só pelo vértice 5.

Passo 2: O vértice obrigatório é k = 5. O corte do vértice 5 está formado pelo único arco (3,5). Então S(i*,j*) = S(3,5) = 3. Fazer S(3,5) = 0 e LI = 4 + 3 = 7.

Passo 3: Os grafos G″ e G2 atualizados ficam:

G″

arco	margem
1- 2	1
1- 3	2
1- 4	1
3- 5	3

G2

arco	S(i,j)	arco	S(i,j)
1- 2	0	7-10	3
1- 3	0	8- 9	2
1- 4	0	8-10	4
2- 7	2	8-11	6
3- 5	0	9-11	5
3- 8	2	10-11	1
5- 7	5		
5- 8	1		
7- 8	3		

ITERAÇÃO 5.

Passo 1: Escolhemos uma componente fortemente conexa que contenha pelo menos um vértice obrigatório e que não seja alcançada por nenhum outro vértice obrigatório, por exemplo, a componente formada só pelo vértice 7.

Passo 2: O vértice obrigatório é k = 7. O corte do vértice 7 está formado pelos arcos (2,7) e (5,7). Então S(i*,j*) = min { S(2,7), S(5,7) } = S(2,7) = 2. Fazer S(2,7) = 0, S(5,7) = 3 e LI = 7 + 2 = 9.

Passo 3: Os grafos G″ e G2 atualizados ficam:

G″

arco	margem
1- 2	1
1- 3	2
1- 4	1
3- 5	3
2- 7	2

G2

arco	S(i,j)	arco	S(i,j)
1- 2	0	7-10	3
1- 3	0	8- 9	2
1- 4	0	8-10	4
2- 7	0	8-11	6
3- 5	0	9-11	5
3- 8	2	10-11	1
5- 7	3		
5- 8	1		
7- 8	3		

ITERAÇÃO 6.

Passo 1: Escolhemos uma componente fortemente conexa que contenha pelo menos um vértice obrigatório e que não seja alcançada por nenhum outro vértice obrigatório, por exemplo, a componente formada só pelo vértice 8.

Passo 2: O vértice obrigatório é k = 8. O corte do vértice 8 está formado pelos arcos (3,8), (5,8) e (7,8). Então S(i*,j*) = min { S(3,8), S(5,8), S(7,8) } = S(5,8) = 1. Fazer S(3,8) = 1, S(5,8) = 0, S(7,8) = 2 e LI = 9 + 1 = 10.

Passo 3: Os grafos G″ e G2 atualizados ficam:

G″

arco	margem
1- 2	1
1- 3	2
1- 4	1
3- 5	3
2- 7	2
5- 8	1

G2

arco	S(i,j)	arco	S(i,j)
1- 2	0	7-10	3
1- 3	0	8- 9	2
1- 4	0	8-10	4
2- 7	0	8-11	6
3- 5	0	9-11	5
3- 8	1	10-11	1
5- 7	3		
5- 8	0		
7- 8	2		

ITERAÇÃO 7.

Passo 1: Escolhemos uma componente fortemente conexa que contenha pelo menos um vértice obrigatório e que não seja alcançada por nenhum outro vértice obrigatório, por exemplo, a componente formada só pelo vértice 9.

Passo 2: O vértice obrigatório é k = 9. O corte do vértice 9 está formado só pelo arco (8,9). Então S(i*,j*) = S(8,9) = 2. Fazer S(8,9) = 0 e LI = 10 + 2 = 12.

Passo 3: Os grafos G″ e G2 atualizados ficam:

G″

arco	margem
1- 2	1
1- 3	2
1- 4	1
3- 5	3
2- 7	2
5- 8	1
8- 9	2

G2

arco	S(i,j)	arco	S(i,j)
1- 2	0	7-10	3
1- 3	0	8- 9	0
1- 4	0	8-10	4
2- 7	0	8-11	6
3- 5	0	9-11	5
3- 8	1	10-11	1
5- 7	3		
5- 8	0		
7- 8	2		

ITERAÇÃO 8.

Passo 1: Escolhemos uma componente fortemente conexa que contenha pelo menos um vértice obrigatório e que não seja alcançada por nenhum outro vértice obrigatório, por exemplo, a componente formada só pelo vértice 10.

Passo 2: O vértice obrigatório é k = 10. O corte do vértice 10 está formado pelos arcos (7,10) e (8,10). Então S(i*,j*) = min { S(7,10), S(8,10) } = S(7,10) = 3. Fazer S(7,10) = 0, S(8,10) = 1 e LI = 12 + 3 = 15.

Passo 3: Os grafos G″ e G2 atualizados ficam:

G″

arco	margem
1- 2	1
1- 3	2
1- 4	1
3- 5	3
2- 7	2
5- 8	1
8- 9	2
7-10	3

G2

arco	S(i,j)	arco	S(i,j)
1- 2	0	7-10	0
1- 3	0	8- 9	0
1- 4	0	8-10	1
2- 7	0	8-11	6
3- 5	0	9-11	5
3- 8	1	10-11	1
5- 7	3		
5- 8	0		
7- 8	2		

ITERAÇÃO 9.

Passo 1: Escolhemos uma componente fortemente conexa que contenha pelo menos um vértice obrigatório e que não seja alcançada por nenhum outro vértice obrigatório, por exemplo, a componente formada só pelo vértice 11.

Passo 2: O vértice obrigatório é k = 11. O corte do vértice 11 está formado pelos arcos (8,11), (9,11) e (10,11). Então S(i*,j*) = min { S(8,11), S(9,11), S(10,11) } = S(10,11) = 1. Fazer S(8,11) = 5, S(9,11) = 5, S(10,11) = 0 e LI = 15 + 1 = 16.

Passo 3: Os grafos G″ e G2 atualizados ficam:

G″

arco	margem
1- 2	1
1- 3	2
1- 4	1
3- 5	3
2- 7	2
5- 8	1
8- 9	2
7-10	3
10-11	1

G2

arco	S(i,j)	arco	S(i,j)
1- 2	0	7-10	0
1- 3	0	8- 9	0
1- 4	0	8-10	1
2- 7	0	8-11	5
3- 5	0	9-11	4
3- 8	1	10-11	0
5- 7	3		
5- 8	0		
7- 8	2		

ITERAÇÃO 10.

Passo 1: Escolhemos uma componente fortemente conexa que contenha pelo menos um vértice obrigatório e que não seja alcançada por nenhum outro vértice obrigatório. Como não existe, FIM. O valor de LI corresponde ao valor da solução aproximada e pode ser utilizado como limite superior. O grafo G″, gerado durante esta segunda utilização do método Dual Ascendente, contém a arborescência associada ao peso LI. Neste caso particular, o grafo G″ é tal arborescência, e pode ser observado na figura (D.4).

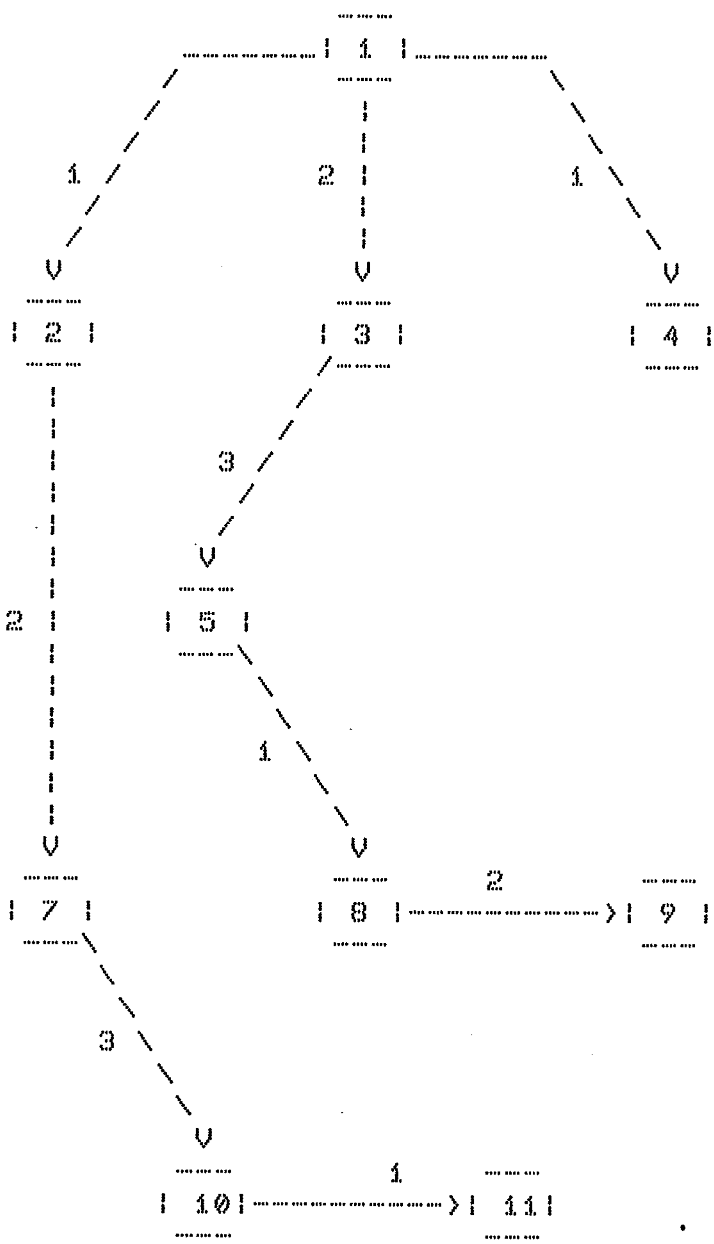


Fig. D.4

D.4 - NOVA HEURÍSTICA

Passo 1: arborescência geradora inicial.

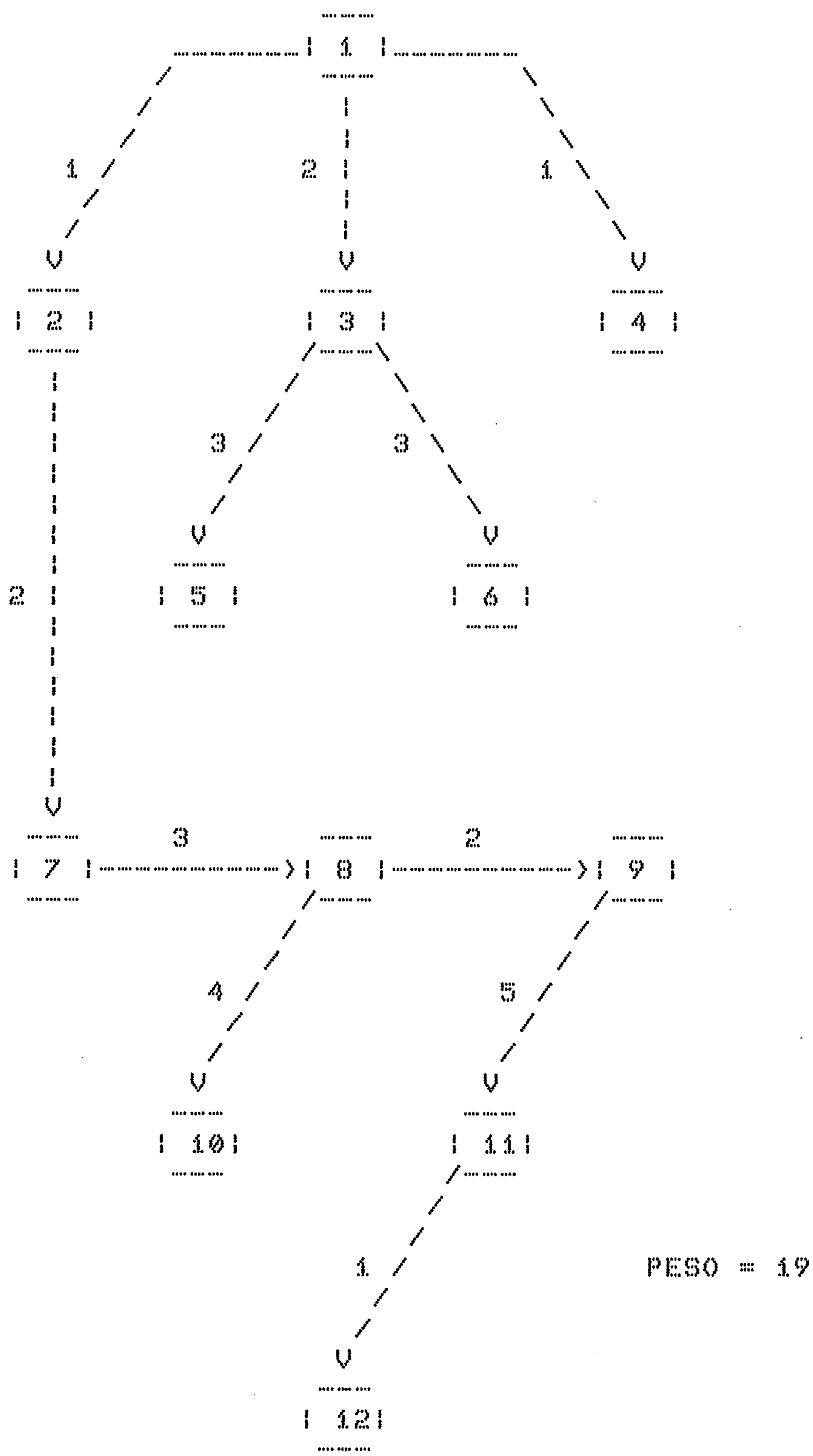


Fig. D.5 - Arborescência inicial.

Esta arborescência inicial tem um peso associado à subarborescência de Steiner igual a 19. (Peso total = 27.)

vértice	CS	PP	arcos	fluxo
1	0	1	(1, 2)	3
2	1	1	(2, 7)	3
3	2	1	(7, 8)	2
4	1	1	(8, 9)	2
5	5	1	(9,11)	1
6	5	3	(11,12)	0
7	3	1	(8,10)	0
8	6	7	(1, 3)	1
9	8	7	(3, 5)	1
10	10	8	(3, 6)	0
11	13	9	(1, 4)	1
12	14	11		

PESO = 19

Passo 2: Procurar um arco ou lista de arco. INICIO = 1.

Passo 2.0: Escolher um arco iniciado no vértice 1 para ser testado, que não esteja na arborescência. Não existe.

Passo 2.0: Escolher um arco iniciado no vértice 2 para ser testado, que não esteja na arborescência. Não existe.

Passo 2.0: Escolher um arco iniciado no vértice 3 para ser testado, que não esteja na arborescência.

Arco (3,8): O arco a ser substituido, (7,8), tem fluxo positivo. Pela expressão (III.1) temos

ALTER = C(3,8) - CS(8) + CS(7) + CC = 2 - 6 + 3 + 0 = -1.

O arco (3,8) permite melhorar o peso da arborescência.

Passo 3: Atualizar a arborescência geradora.

Entra o arco (3,8) e sai o arco (7,8). A nova arborescência fica assim:

vértice	CS	PP	arcos	fluxo
1	0	1	(1, 2)	1
2	1	1	(2, 7)	1
3	2	1	(3, 8)	2
4	1	1	(8, 9)	2
5	5	3	(9,11)	1
6	5	3	(11,12)	0
7	3	1	(8,10)	0
8	4	3	(1, 3)	3
9	6	3	(3, 5)	1
10	8	8	(3, 6)	0
11	11	9	(1, 4)	1
12	12	11		

PESO = 18

Passo 2: Procurar um arco ou lista de arco. INICIO = 3.

Passo 2.0: Escolher um arco iniciado no vértice 3 para ser testado, que não esteja na arborescência. Não existe.

Passo 2.0: Escolher um arco iniciado no vértice 4 para ser testado, que não esteja na arborescência.

Arco (4,6):

ALTER = C(4,6) - CS(6) + CS(3) + CC = 4 - 5 + 2 + 0 = 1.

O arco (4,6) n"ao é bom e ALTER - CC = 1 > 0.

Não tem outro arco iniciado no vértice 4.

Passo 2.0: Escolher um arco iniciado no vértice 5 para ser testado, que não esteja na arborescência.

Arco (5,7):

ALTER = C(5,7) - CS(7) + CS(1) + CC = 5 - 3 + 0 + 0 = 2.

O arco (5,7) não é bom e ALTER - CC = 2 > 0.

Arco (5,8):

ALTER = C(5,8) - CS(8) + CS(3) + CC = 1 - 4 + 2 + 0 = -1.

Incluir o arco (5,8) em substituição do arco (3,8).

Passo 3: Atualizar a arborescência geradora.

Entra o arco (5,8) e sai o arco (3,8). A nova arborescência fica assim:

vértice	CS	PP
1	0	1
2	1	1
3	2	1
4	1	1
5	5	1
6	5	3
7	3	1
8	6	5
9	8	5
10	10	8
11	13	9
12	14	11

arcos	fluxo
(1, 2)	1
(2, 7)	1
(5, 8)	2
(8, 9)	2
(9,11)	1
(11,12)	0
(8,10)	0
(1, 3)	3
(3, 5)	3
(3, 6)	0
(1, 4)	1

PESO = 17

Passo 2: Procurar um arco ou lista de arco. INICIO = 5.

Passo 2.0: Escolher um arco iniciado no vértice 5 para ser testado, que não esteja na arborescência.

Arco (5,7):

$$ALTER = C(5,7) - CS(7) + CS(1) + CC = 5 - 3 + 0 + 0 = 2.$$

O arco (5,7) n"ao é bom e ALTER - CC = 2 > 0.

Não tem outro arco iniciado no vértice 5.

Passo 2.0: Escolher um arco iniciado no vértice 6 para ser testado, que não esteja na arborescência.

Arco (6,8):

$$ALTER = C(6,8) - CS(8) + CS(5) + CC = 2 - 6 + 5 + 3 = 4.$$

O arco (6,8) não é bom e ALTER - CC = 4 - 3 = 1 > 0.

Arco (6,9):

$$ALTER = C(6,9) - CS(9) + CS(5) + CC = 2 - 8 + 5 + 3 = 2.$$

Mas, como ALTER - CC = 2 - 3 = -1 < 0, e o fluxo do segmento do vértice 6 é zero, podemos iniciar a formação de uma lista Tipo I.

Passo 2.3.1: O arco (6,9) é incluido na lista:

$$LISTA = \{ (6,9) \} \text{ e}$$

$$V = C(6,9) - CS(9) + CS(5) = 2 - 8 + 5 = -1$$

$$x = 6.$$

Passo 2.3.2:

$$T = V + CS(6) - CS(3) = -1 + 5 - 2 = 2 > 0.$$

Passo 2.3.3: Procurar um arco (x,y), que não forme ciclo, tal que o fluxo do segmento do vértice y seja positivo.

Arco (6,8): existe o arco (6,9) na lista tal que o vértice 8 pertence ao segmento do vértice 9 e, portanto,

$$Q = C(6,8) + CS(9) - CS(8) - C(6,9) = 2 + 8 - 6 - 2 = 2.$$

Passo 2.3.4: Q > 0.

Passo 2.3.3: Não tem mais arco (x,y) para x = 6.

Passo 2.3.5: w = 3. Como w = PP(6), não é possível continuar com a formação desta lista.

Passo 2.0: Escolher um arco iniciado no vértice 7 para ser testado, que não esteja na arborescência.

Arco (7,8):

ALTER = C(7,8) - CS(8) + CS(5) + CC = 3 - 6 + 5 + 0 = 2.

O arco (7,8) não é bom e ALTER - CC = 2 > 0.

Arco (7,10) (o fluxo do segmento do vértice 10 é zero):

ALTER = C(7,10) - CS(10) + CS(8) + CC = 3 - 10 + 6 + 0 = -1.

Incluir o arco (7,10) em substituição do arco (8,10).

Passo 3: Atualizar a arborescência geradora.

Entra o arco (7,10) e sai o arco (8,10). A nova arborescência fica assim:

vértice	CS	PP
1	0	1
2	1	1
3	2	1
4	1	1
5	5	1
6	5	3
7	3	1
8	6	5
9	8	5
10	6	7
11	13	9
12	14	11

arcos	fluxo
(1, 2)	1
(2, 7)	1
(5, 8)	2
(8, 9)	2
(9,11)	1
(11,12)	0
(7,10)	0
(1, 3)	3
(3, 5)	3
(3, 6)	0
(1, 4)	1

PESO = 17

Passo 2: Procurar um arco ou lista de arco. INICIO = 7.

Passo 2.0: Escolher um arco iniciado no vértice 7 para ser testado, que não esteja na arborescência.

Arco (7,8):

ALTER = C(7,8) - CS(8) + CS(5) + CC = 3 - 6 + 5 + 0 = 2.

O arco (7,8) n"ao é bom e ALTER - CC = 2 > 0.

Não tem outro arco iniciado no vértice 7.

Passo 2.0: Escolher um arco iniciado no vértice 8 para ser testado, que não esteja na arborescência.

Arco (8,10):

ALTER = C(8,10) - CS(10) + CS(7) + CC = 4 - 6 + 3 + 0 = 1.

O arco (8,10) não é bom e ALTER - CC = 1 > 0.

Arco (8,11):

ALTER = C(8,11) - CS(11) + CS(9) + CC = 6 - 13 + 8 + 0 = 1.

O arco (8,11) não é bom e ALTER - CC = 1 > 0.

Passo 2.0: Escolher um arco iniciado no vértice 9 para ser testado, que não esteja na arborescência. Não existe.

Passo 2.0: Escolher um arco iniciado no vértice 10 para ser testado, que não esteja na arborescência. Não existe.

Arco (10,11):

ALTER = C(10,11) - CS(11) + CS(9) + CC = 1 - 13 + 8 + 3 = -1.

Incluir o arco (10,11) em substituição do arco (9,11).

Passo 3: Atualizar a arborescência geradora.

Entra o arco (10,11) e sai o arco (9,11). A nova arborescência fica assim:

vértice	CS	PP
1	0	1
2	1	1
3	2	1
4	1	1
5	5	1
6	5	3
7	3	1
8	6	5
9	8	5
10	6	7
11	7	7
12	8	11

arcos	fluxo
(1, 2)	1
(2, 7)	2
(5, 8)	1
(8, 9)	1
(10,11)	1
(11,12)	0
(7,10)	1
(1, 3)	2
(3, 5)	2
(3, 6)	0
(1, 4)	1

PESO = 16

Passo 2.0: Todos os testes feitos a partir de agora, na tentativa de melhorar o peso da arborescência, não permitirão determinar novos arcos ou listas Tipos I ou III para serem incluidos.

Passo 2.5: Tentar determinar uma lista Tipo II. Não existe nenhum vértice optativo que seja inicio de algum fragmento de fluxo positivo. FIM.

###########

A solução aproximada para o problema de Steiner, que neste caso coincide com a solução ótima, pode ser observada na Figura (D.4).