

## 2

### O problema da árvore geradora de custo mínimo com restrição de diâmetro

O problema da Árvore Geradora de Custo Mínimo com Restrição de Diâmetro (AGMD) é definido sobre um grafo conexo  $G = (V, E)$ , onde  $V$  é o conjunto de vértices e  $E$  é o conjunto de arestas. Cada aresta  $[i, j] \in E$  possui um custo  $c_{ij} \geq 0$ . Qualquer árvore geradora de  $G$  possui um caminho único  $\mathcal{P}_{ij}$  que conecta cada par de nós distintos  $i, j \in V$ . Sendo  $d_{ij}$  a quantidade de arestas em  $\mathcal{P}_{ij}$ , o diâmetro de uma árvore é definido como  $\max\{d_{ij} : i, j \in V\}$ . AGMD consiste em encontrar uma árvore de custo mínimo cujo diâmetro seja menor ou igual a  $D$ , onde  $D$  é um inteiro positivo satisfazendo  $2 \leq D \leq |V| - 1$ .

AGMD é polinomial em quatro casos particulares. O primeiro ocorre quando  $D = |V| - 1$  e sua solução corresponde à AGM de  $G$ . O segundo caso ocorre quando  $D = 2$ . Neste caso, a solução é uma árvore em forma de estrela como ilustrado na Figura 2.1-(a). Para encontrar a estrela de menor custo basta comparar as  $O(|V|)$  estrelas de  $G$ . Este cálculo consome no pior caso  $O(|V|^2)$ . O terceiro caso ocorre quando  $D = 3$ , cuja solução corresponde a duas estrelas conectadas por uma ponte (chamada neste trabalho de aresta central), ver Figura 2.1-(b). Neste caso,  $O(|E|)$  árvores precisam ser comparadas. A cada

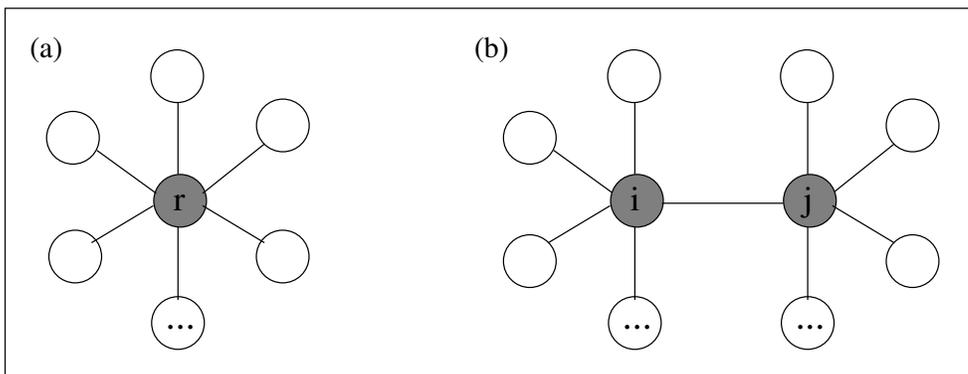


Figura 2.1: Casos polinomiais.

iteração uma aresta  $[i, j] \in E$  é tomada como aresta central da árvore. Em seguida, cada nó  $k$  fora da solução será conectado a uma das extremidades  $i$  ou  $j$  da aresta central. Escolhe-se a aresta  $[k, i]$  ou  $[k, j]$  que tenha o menor custo. A complexidade de pior caso deste procedimento é  $O(|E| \cdot |V|)$ . O quarto

caso ocorre quando todos os custos das arestas são iguais, não influenciando na construção da árvore. Neste caso, em grafos completos, qualquer estrela é ótima e a complexidade de calculá-la é linear. Em grafos esparsos, uma busca em largura pode ser aplicada para encontrar uma solução ótima [4].

## 2.1

### Problemas correlatos

AGMD é uma generalização do problema da Árvore Geradora de Custo Mínimo com Restrição de Salto (AGMS). Dado um grafo  $G = (V, E)$ , com custos  $c_{ij} \geq 0$  associados a cada aresta  $[i, j] \in E$  e estabelecido um nó raiz  $k \in V$ , AGMS consiste em encontrar uma árvore de custo mínimo onde os caminhos de  $k$  para todos os nós  $i \in V \setminus \{k\}$  tenham uma quantidade de arestas menor ou igual a um inteiro positivo  $H$ . Diversos modelos para AGMS são encontrados em [17, 25, 26, 27]. Desigualdades válidas e a descrição completa do politopo de AGMS quando  $H = 2$  ou  $H = 3$  são apresentadas em [18]. Uma formulação estendida para  $H = 4$  é apresentada em [19].

Outro problema relacionado com AGMD é o da Árvore de Steiner de Custo Mínimo com Restrição de Diâmetro (ASMD). Seja um grafo  $G = (V, E)$ , com custos  $c_{ij} \geq 0$  associados a cada aresta  $[i, j] \in E$ . Dado um subconjunto  $S$  de nós terminais ( $S \subseteq V$ ) e um inteiro positivo  $D$ , o ASMD consiste em encontrar uma árvore de custo mínimo que conecte os nós terminais, com diâmetro menor ou igual a  $D$ . O caso especial em que  $S = V$  corresponde ao AGMD. Um algoritmo pseudo-polinomial de complexidade  $O(|V|^{2D})$  para o ASMD é proposto em [44, 45]. Modelos são apresentados em [29, 30, 31].

## 2.2

### Revisão bibliográfica

As formulações presentes na literatura para AGMD utilizam implicitamente uma propriedade que caracteriza a viabilidade de árvores geradoras de custo mínimo com restrição de diâmetro [35]. Essa propriedade define que quando o diâmetro é par, deve existir um *vértice central*  $i \in V$  em toda árvore viável  $T$ , tal que o caminho de todos os vértices de  $T$  até  $i$  não contenha mais que  $D/2$  arestas (caso par). Quando o diâmetro é ímpar, deve existir uma *aresta central*  $e = [i, j] \in E$  em toda árvore viável  $T$ , tal que o caminho de todos os vértices de  $T$  até uma das extremidades de  $e$  não contenha mais que  $(D - 1)/2$  arestas (caso ímpar).

Outro aspecto compartilhado por algumas das formulações existentes é a inclusão de um nó artificial que centraliza a distribuição de fluxo para os nós de  $G$ . Este tipo de estratégia cria uma estrutura auxiliar de fluxo em redes que ajuda a reforçar os modelos.

As formulações pioneiras para AGMD foram propostas por Achuthan e Caccetta [2, 3] e utilizam idéias provenientes dos problemas do caixeiro viajante e de roteamento de veículos. A modelagem é feita tratando separadamente os casos em que o diâmetro requerido é par ou ímpar. No caso par é inserido um nó artificial. Em [2] foi desenvolvido um algoritmo de *branch-and-bound* que resolve instâncias esparsas de até 40 nós. Uma versão aprimorada deste algoritmo é apresentada em [3] e resolve através de um *branch-and-bound* instâncias esparsas de até 50 nós.

Gouveia e Magnanti [28, 30] apresentam diversas estratégias de modelagem para AGMD. São descritos modelos básicos multifluxos que não separam os casos par e ímpar. São apresentadas ainda formulações de fluxos em redes que tratam os casos par e ímpar separadamente. Observou-se que os limites inferiores para o caso ímpar são mais fracos que os produzidos para o caso par. Então, formulações mais elaboradas são apresentadas para o caso ímpar.

Requejo [58] apresenta diversas formulações alternativas para o caso ímpar. Na primeira abordagem são descritos modelos que utilizam fluxos em redes com a introdução de um nó artificial. A partir deste nó são enviadas duas unidades de fluxo para cada nó terminal. Na segunda estratégia, considera-se dois caminhos a partir do nó raiz, um mais curto e outro longo. A terceira abordagem trata o problema através da sobreposição de dois problemas com  $D$  par. A última proposta usa técnicas de programação disjuntiva para fornecer a envoltória convexa das soluções inteiras de um subproblema relacionado com a seleção da aresta central.

A estratégia proposta por Gouveia, Magnanti e Requejo em [29, 31] para inserção de um nó artificial para o caso ímpar considera a geração de duas árvores orientadas partindo do nó artificial  $r$ , como na Figura 2.2. Cada árvore possui uma única conexão do nó artificial com uma das extremidades da aresta central, ver Figuras 2.2-(a) e 2.2-(b).

Santos, Lucena e Ribeiro [60] também apresentam uma estratégia de inserção de um nó artificial para o caso ímpar. O nó artificial  $r$  é conectado através de duas arestas com as extremidades do nó central, como na Figura 2.3.

Gruber e Raidl [32] descrevem uma modelagem para AGMD em que todas as variáveis são inteiras. Neste trabalho não existe introdução de nó artificial e os casos par e ímpar são tratados separadamente. A idéia desta modelagem é construir a árvore através de uma relação de precedência entre os nós.

As maiores instâncias em grafos completos são resolvidas em [32, 60] e contêm 25 nós. As maiores instâncias em grafos esparsos contêm 60 nós e 600 arestas e são resolvidas nesta tese e em [31].

Abdalla [1] apresenta quatro heurísticas para AGMD. A heurística *One*

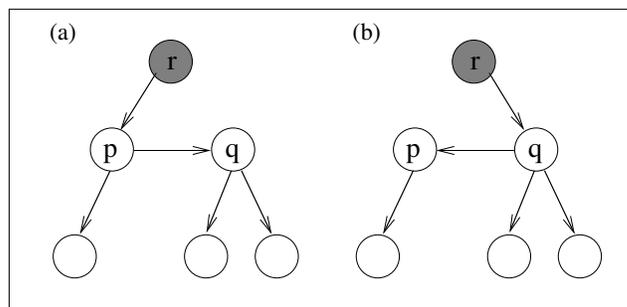


Figura 2.2: Inserção de um nó artificial no caso ímpar por Gouveia, Magnanti e Requejo [29].

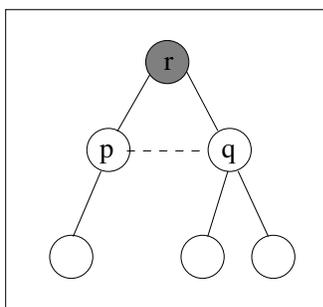


Figura 2.3: Inserção de um nó artificial no caso ímpar por Santos, Lucena e Ribeiro [60].

*Time Tree* (OTT), três algoritmos de refinamento iterativo da AGM e um algoritmo para o caso em que o diâmetro requerido é menor ou igual a 4. Versões paralelas para estes algoritmos são apresentadas em [20]. Experimentos comparativos com instâncias em grafos completos com até 1000 nós são realizados. Dentre os algoritmos testados, a OTT é descrita como sendo a heurística mais eficiente e eficaz para resolver instâncias de grafos completos. Para instâncias em grafos esparsos, o autor relata que a segunda heurística de refinamento iterativo da AGM descrita em [1] é a melhor.

Raidl e Julstrom [56] desenvolveram um algoritmo denominado *Randomized Greedy Heuristic* (RGH) e um algoritmo genético. Os testes comparativos da RGH com a OTT indicam que a RGH produz resultados substancialmente melhores. Versões aprimoradas do algoritmo genético são apresentadas em [42, 43]. Instâncias em grafos completos com até 1000 nós são testadas em [56]. Nos experimentos realizados com os algoritmos aprimorados, instâncias com no máximo 500 nós foram testadas em [42, 43].

Gruber e Raidl [33] apresentaram uma heurística do tipo VNS para AGMD. Este algoritmo produz melhores resultados do que as versões aprimoradas do algoritmo genético.

Na Seção 2.3, são apresentadas as formulações que influenciaram o desenvolvimento de extensões neste trabalho. Embora sejam uma amostra

dos modelos existentes para AGMD, constituem uma base fundamental e ilustram diferentes abordagens. Na Seção 2.4, são relatadas as heurísticas OTT e RGH. Essas heurísticas são detalhadas porque duas de suas variantes são propostas nesta tese. Em seguida, nas Seções 2.4.3 e 2.4.4 são apresentadas respectivamente heurísticas de refinamento iterativo da AGM e heurísticas de melhoria baseadas em metaheurísticas.

## 2.3

### Modelos para AGMD

Para as formulações apresentadas na Seção 2.3.1 e os modelos direcionados da Seção 2.3.2, considera-se o grafo  $G' = (V, A)$  obtido de  $G = (V, E)$  tomando-se cada aresta  $[i, j] \in E$ , com  $i < j$ , e transformando-a em dois arcos  $(i, j)$  e  $(j, i) \in A$ , com custos  $c_{ij} = c_{ji}$ . Para os casos em que um nó artificial  $r$  é introduzido em  $G'$ , o grafo  $G'' = (V', A')$  é obtido tomando-se  $V' = V \cup \{r\}$  e  $A' = A \cup \{(r, 1), \dots, (r, |V|)\}$ . Então,  $|A'| = 2 \cdot |E| + |V|$  e  $|V'| = |V| + 1$ . Os custos dos arcos conectados ao nó artificial são iguais a zero.

Nos modelos onde uma raiz é fixada, considera-se que a quantidade máxima de arestas nos caminhos a partir da raiz são iguais a  $L = D/2$  quando  $D$  é par e  $L = (D - 1)/2$  quando  $D$  é ímpar.

#### 2.3.1

##### Modelos de comodidade única

Modelos de comodidade única são aqueles em que uma única comodidade (ou unidade de fluxo) é enviada para construir uma árvore. A seguir são apresentados os principais modelos presentes na literatura que possui esta característica.

##### Modelos propostos por Achuthan e Caccetta

Os modelos desenvolvidos por Achuthan e Caccetta [2, 3] separam os casos par e ímpar. Esses modelos foram aprimorados neste tese a partir da inserção de inequações válidas. Uma adaptação que inclui um nó artificial para o caso ímpar é apresentada no Capítulo 3.

Nestes modelos, as variáveis binárias  $x_{ij}$  associadas a cada arco  $(i, j) \in A'$  no caso par (resp.  $(i, j) \in A$  no caso ímpar) identificam a árvore geradora. Variáveis não negativas  $u_i$  são associadas a cada vértice  $i \in V'$  no caso par (resp.  $i \in V$  no caso ímpar) e representam a profundidade de  $i$  em relação à raiz. A profundidade da raiz é 0. Para o caso ímpar, existem ainda as variáveis binárias  $z_{ij}$  associadas a cada aresta  $[i, j] \in E$ , com  $i < j$ . Se  $z_{ij} = 1$ , então  $[i, j]$  é a aresta central da árvore geradora. Caso contrário,  $z_{ij} = 0$ .

Em uma solução viável para o caso par, as arestas  $[i, j] \in E$  tais que  $x_{ij} = 1$  ou  $x_{ji} = 1$  obtidas como solução do modelo (2-1) a (2-6) definem uma árvore geradora de  $G$  com diâmetro menor ou igual a  $D$ :

$$\min \sum_{(i,j) \in A} c_{ij} \cdot x_{ij} \quad (2-1)$$

sujeito a:

$$\sum_{j \in V} x_{rj} = 1, \quad (2-2)$$

$$\sum_{(i,j) \in A'} x_{ij} = 1 \quad \forall j \in V, \quad (2-3)$$

$$u_i - u_j + (L + 1) \cdot x_{ij} \leq L \quad \forall (i, j) \in A', \quad (2-4)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A', \quad (2-5)$$

$$0 \leq u_i \leq L + 1 \quad \forall i \in V'. \quad (2-6)$$

As equações (2-2) garantem que o nó artificial  $r$  é conectado a um e somente um vértice de  $V$ . As restrições (2-3) asseguram que existe um e somente um arco incidente em cada vértice de  $V$ . As inequações (2-4) e (2-6) estabelecem que os caminhos do vértice artificial  $r$  para cada vértice  $i \in V$  não contêm mais de  $L$  arcos. As restrições (2-5) estabelecem a integralidade das variáveis  $x$ .

Para o caso ímpar, o modelo é definido pelas expressões (2-7) a (2-13). A aresta central e as arestas  $[i, j] \in E$  tais que  $x_{ij} = 1$  ou  $x_{ji} = 1$  definem uma árvore geradora de  $G$  com diâmetro menor ou igual a  $D$ :

$$\min \sum_{(i,j) \in A} c_{ij} \cdot x_{ij} + \sum_{[i,j] \in E} c_{ij} \cdot z_{ij} \quad (2-7)$$

sujeito a:

$$\sum_{[i,j] \in E} z_{ij} = 1, \quad (2-8)$$

$$\sum_{(i,j) \in A} x_{ij} + \sum_{[i,j] \in E: i < j} z_{ij} + \sum_{[j,i] \in E: j < i} z_{ji} = 1 \quad \forall j \in V, \quad (2-9)$$

$$u_i - u_j + (L + 1) \cdot x_{ij} \leq L \quad \forall (i, j) \in A, \quad (2-10)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A, \quad (2-11)$$

$$z_{ij} \in \{0, 1\} \quad \forall [i, j] \in E, \quad (2-12)$$

$$0 \leq u_i \leq L \quad \forall i \in V. \quad (2-13)$$

Na função objetivo é adicionado um termo para contabilizar o custo da aresta central (2-7). As equações (2-8) garantem que existe uma e somente uma aresta central. As restrições (2-9) asseguram que para todo vértice  $i \in V$

ou existe um arco incidente a  $i$  ou então  $i$  deve ser uma das extremidades da aresta central. As inequações (2-10) e (2-13) estabelecem que o caminho da extremidade mais próxima da aresta central para cada vértice  $i \in V$  contém no máximo  $L$  arestas. As restrições (2-11) e (2-12) estabelecem respectivamente a integralidade das variáveis  $x$  e  $z$ .

Nesses modelos, as versões para o caso par e ímpar contêm  $O(|E| + |V|)$  restrições. Para o caso par, o modelo contém  $O(|E| + |V|)$  variáveis inteiras e  $O(|V|)$  variáveis contínuas. Para o caso ímpar, existem  $O(|E|)$  variáveis inteiras e  $O(|V|)$  variáveis contínuas.

### Modelos propostos por Gruber e Raidl

Os modelos propostos por Gruber e Raidl [32] também separam os casos onde o diâmetro é par ou ímpar. Nos experimentos realizados em [32], uma comparação foi realizada com os modelos de Achuthan e Caccetta reforçados, propostos nesta tese.

As variáveis  $p_{ij}$ ,  $(i, j) \in A$ , indicam que o predecessor do nó  $j$  na árvore  $T$  é o nó  $i$ . Implicitamente, os arcos de  $T$  são obtidos a partir dessas variáveis. Se  $p_{ij} = 1$ , então o arco  $(i, j)$  está na solução. As variáveis  $u_{il}$  indicam que o nó  $i$  encontra-se na profundidade  $l$ . Essas variáveis controlam o diâmetro da árvore. Para o caso ímpar, existem ainda as variáveis binárias  $r_{ij}$  associadas a cada aresta  $[i, j] \in E$ . Se  $r_{ij} = 1$ , então a aresta  $[i, j]$  é a aresta central da árvore geradora. Caso contrário,  $r_{ij} = 0$ .

O modelo para o caso par é definido pelas expressões (2-14) a (2-20). As equações (2-15) asseguram que cada nó  $i$  possui uma única profundidade. As restrições (2-16) garantem que existe uma única raiz e esta encontra-se na profundidade 0. As equações (2-17) estabelecem que cada nó, exceto a raiz, possui um único predecessor. As desigualdades (2-18) determinam que o nó  $i$  só pode ser predecessor do nó  $j$  se o nó  $i$  encontra-se em um nível de profundidade imediatamente anterior ao do nó  $j$ . As restrições (2-19) e (2-20) estabelecem respectivamente a integralidade das variáveis  $u$  e  $p$ .

$$\min \sum_{(i,j) \in A} c_{ij} \cdot p_{ij} \tag{2-14}$$

sujeito a:

$$\sum_{l=0, \dots, L} u_{il} = 1 \quad \forall i \in V, \tag{2-15}$$

$$\sum_{i \in V} u_{i0} = 1, \tag{2-16}$$

$$\sum_{i: (i,j) \in A} p_{ij} = 1 - u_{j0} \quad \forall j \in V, \tag{2-17}$$

$$p_{ij} \leq 1 - u_{jl} + u_{i(l-1)} \quad \forall (i, j) \in A, \quad l = 1, \dots, L, \tag{2-18}$$

$$u_{ij} \in \{0, 1\} \quad \forall i \in V, \quad l = 0, \dots, L, \quad (2-19)$$

$$p_{ij} \in \{0, 1\} \quad \forall (i, j) \in A. \quad (2-20)$$

O modelo de (2-21) a (2-28) define a formulação para o caso ímpar. As variáveis  $r_{ij}$  definem a aresta central. A função objetivo (2-21) é adaptada para incluir o custo da aresta central:

$$\min \sum_{(i,j) \in A} c_{ij} \cdot p_{ij} + \sum_{[i,j] \in E} c_{ij} \cdot r_{ij} \quad (2-21)$$

sujeito a:

$$\sum_{i \in V} u_{i0} = 2, \quad (2-22)$$

$$\sum_{j: [i,j] \in E} r_{ij} = u_{i0} \quad \forall i \in V, \quad (2-23)$$

$$\sum_{l=0, \dots, L} u_{il} = 1 \quad \forall i \in V, \quad (2-24)$$

$$\sum_{i: (i,j) \in A} p_{ij} = 1 - u_{j0} \quad \forall j \in V, \quad (2-25)$$

$$p_{ij} \leq 1 - u_{jl} + u_{i(l-1)} \quad \forall (i, j) \in A, \quad l = 1, \dots, L, \quad (2-26)$$

$$u_{il} \in \{0, 1\} \quad \forall i \in V, \quad l = 0, \dots, L, \quad (2-27)$$

$$p_{ij} \in \{0, 1\} \quad \forall (i, j) \in A. \quad (2-28)$$

As equações (2-22) garantem que existem dois nós que constituem as extremidades da aresta central. Para esses nós é atribuída profundidade igual a 0. As restrições (2-23) asseguram que existirá uma aresta conectando os nós que estão na profundidade 0. As equações (2-24) asseguram que cada nó  $i$  possui uma única profundidade. As equações (2-25) estabelecem que cada nó, exceto as extremidades da aresta central, possui um único predecessor. As desigualdades (2-26) determinam que o nó  $i$  só pode ser predecessor do nó  $j$  se o nó  $i$  está localizado na profundidade imediatamente anterior à do nó  $j$ . Finalmente, (2-27) e (2-28) são respectivamente as restrições de integralidade das variáveis  $u$  e  $p$ .

As versões para os casos par e ímpar propostos por Gruber e Raidl contém  $O(|E| \cdot D)$  restrições e  $O(|E| + |V| \cdot D)$  variáveis inteiras.

Para ambos os casos par e ímpar, os limites generalizados (2-29) e (2-30) abaixo são válidos. As desigualdades (2-29) garantem que os nós na profundidade 0 (zero) não podem ter predecessor. As inequações (2-30) asseguram que um nó na profundidade  $L$  não pode ser predecessor de nenhum nó.

$$u_{j0} \leq 1 - p_{ij} \quad (i, j) \in A. \quad (2-29)$$

$$u_{iL} \leq 1 - p_{ij} \quad (i, j) \in A. \quad (2-30)$$

Os modelos descritos acima foram testados em um Pentium IV com relógio de 2.8 GHz e com 2 Gbytes de memória RAM [32]. Utilizou-se a versão 8.1 do resolvidor de programação linear inteira CPLEX. Foram resolvidas instâncias em grafos completos com 25 nós. As maiores instâncias esparsas resolvidas contêm respectivamente 40 nós e 100 arestas e 30 nós e 200 arestas.

### 2.3.2

#### Modelos multifluxo

Os modelos multifluxo são aqueles em que fluxos distintos são enviados para cada nó do grafo objetivando construir uma árvore.

A modelagem multifluxo básica apresentada em [28] não separa os casos par e ímpar. Para esta formulação é proposta uma relaxação lagrangeana descrita no Capítulo 3.

O modelo multifluxo básico é definido de (2-31) a (2-39). Neste modelo,  $x_{ij} = 1$  indica que a aresta  $[i, j] \in E$  pertence à árvore geradora, sendo  $i < j$ . Contém, ainda, variáveis orientadas de fluxo  $y_{ij}^{pq} \in [0, 1]$  que especificam se o caminho único do nó  $p$  para o nó  $q$  passa pelo arco  $(i, j)$ , onde  $(i, j) \in A$  e  $p, q \in V$ . Para estas variáveis é estabelecido que  $i \neq q$  e  $j \neq p$ . Isto ocorre porque se o nó  $p$  é escolhido como origem do fluxo, então não pode ser um nó de transbordo para este mesmo fluxo. Se o nó  $q$  é o destino do fluxo, então não pode ser nó de transbordo para este mesmo fluxo.

$$\min \sum_{[i,j] \in E} c_{ij} \cdot x_{ij} \quad (2-31)$$

sujeito a:

$$\sum_{[i,j] \in E} x_{ij} = |V| - 1, \quad (2-32)$$

$$\sum_{i:(i,j) \in A} y_{ji}^{jq} = 1 \quad \forall j, q \in V, \quad (2-33)$$

$$\sum_{i:(i,j) \in A} y_{ij}^{pq} - \sum_{i:(i,j) \in A} y_{ji}^{pq} = 0 \quad \forall p, q, j \in V; j \neq p, q, \quad (2-34)$$

$$\sum_{i:(i,j) \in A} y_{ij}^{pj} = 1 \quad \forall j, p \in V, \quad (2-35)$$

$$y_{ij}^{pq} + y_{ji}^{p'q'} \leq x_{ij} \quad \forall [i, j] \in E; p, q, q' \in V, \quad (2-36)$$

$$\sum_{(i,j) \in A} y_{ij}^{pq} \leq D \quad \forall p, q \in V, \quad (2-37)$$

$$y_{ij}^{pq} \in [0, 1] \quad \forall (i, j) \in A; p, q \in V, \quad (2-38)$$

$$x_{ij} \in \{0, 1\} \quad \forall [i, j] \in E. \quad (2-39)$$

As equações (2-32) garantem que a árvore terá  $|V| - 1$  arestas. As

restrições (2-33) a (2-35) são de conservação de fluxo. As equações (2-33) indicam que todo nó  $j \in V$  envia uma unidade de fluxo para cada nó  $q \in V$ . As restrições (2-35) especificam que todo nó  $j \in V$  recebe uma unidade de fluxo proveniente de cada nó  $p \in V$ . As equações (2-34) garantem que a quantidade de fluxo que chega e que sai de um nó é a mesma. As desigualdades (2-36) asseguram que se a aresta  $[i, j] \in E$  não está na árvore, então nenhum caminho passa por  $[i, j]$ . As restrições (2-37) especificam que a quantidade de arestas que compõem o caminho entre os nós  $p$  e  $q$  não contém mais que  $D$  arcos. (2-38) e (2-39) são respectivamente as restrições de integralidade das variáveis  $y$  e  $x$ .

No modelo multifluxo básico se existe um caminho de  $p \in V$  até  $q \in V$  como ilustrado na Figura 2.4-(a), então necessariamente existe um caminho de  $q$  até  $p$  como mostrado na Figura 2.4-(b). Estes caminhos de  $p$  até  $q$  e de  $q$  até  $p$  devem utilizar obrigatoriamente as mesmas arestas, pois, em caso contrário, existiria um ciclo. Além disto, o modelo multifluxo básico apresentado acima define uma árvore. Portanto, a equação (2-40) é válida para o mesmo. Este fato permite reduzir à metade a quantidade de variáveis  $y$ :

$$y_{ij}^{pq} = y_{ji}^{qp} \quad \forall (i, j) \in A, p, q \in V. \quad (2-40)$$

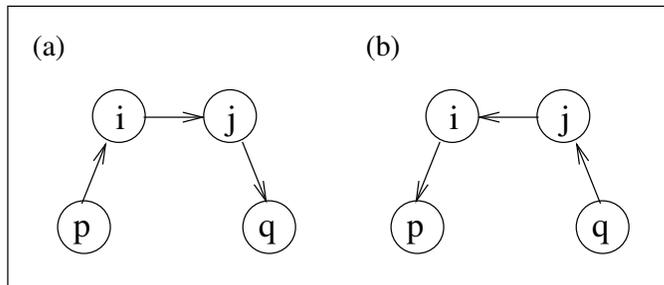


Figura 2.4: Restrição válida para o modelo básico.

Ao invés das restrições (2-36), restrições mais fracas  $y_{ij}^{pq} + y_{ji}^{qp} \leq x_{ij}, \forall [i, j] \in E, p, q \in V$  são utilizadas em [28]. Nesta tese, utiliza-se as restrições originais sugeridas em [51]. Sem as restrições de diâmetro (2-37), o modelo (2-31)-(2-39) resulta em uma AGM de  $G$ . Está provado em [51] que a relaxação linear do modelo (2-31) a (2-39) fornece naturalmente uma AGM. Tal como publicado em [28], a relaxação linear resultante não fornece, necessariamente, uma AGM de  $G$ .

O modelo multifluxo básico contém  $O(|E| \cdot |V|^3)$  restrições,  $O(|E| \cdot |V|^2)$  variáveis contínuas e  $O(|E|)$  variáveis inteiras.

Os testes realizados em [28] foram executados em um Pentium com relógio de 450Mhz e utilizou-se a versão 5.0 do resolvedor CPLEX. Relata-se que foi consumido mais de uma semana de tempo de computação para resolver

instâncias pequenas e esparsas de 20 nós e 50 arestas, devido ao grande número de variáveis e restrições do modelo.

Os primeiros modelos direcionados propostos em [28] separam os casos par e ímpar, mas nenhum nó artificial é utilizado. Nessas formulações, uma árvore geradora direcionada é construída a partir de um nó central  $i \in V$  no caso par e das extremidades da aresta central  $[i, j] \in E$  no caso ímpar. Nesta tese, avaliou-se a inserção de um nó artificial nestes modelos. Entretanto, os resultados obtidos não foram promissores neste caso.

As versões para o caso par e ímpar para o modelo multifluxo direcionado são apresentadas respectivamente de (2-41) a (2-50) e de (2-51) a (2-60). Nestes modelos, as variáveis  $x_{ij} = 1$  especificam que o arco  $(i, j) \in A$  está na solução. Caso contrário,  $x_{ij} = 0$ . São também utilizadas variáveis de fluxo direcionado  $y_{ijk}$  que especificam se o caminho único do nó central até o nó  $k$  utiliza o arco  $(i, j)$ . Além disto, existem ainda variáveis  $r_j, \forall j \in V$ , no caso par (resp.  $r'_{ij}, \forall [i, j] \in E$ , no caso ímpar). No caso par, se  $r_j = 1$ , então o nó  $j$  é o nó central da árvore. No caso ímpar, se  $r'_{ij} = 1$ , então a aresta  $[i, j]$  é a aresta central. O modelo multifluxo direcionado para o caso par é definido pelas expressões (2-41) a (2-50):

$$\min \sum_{(i,j) \in A} c_{ij} \cdot x_{ij} \quad (2-41)$$

sujeito a:

$$\sum_{j \in V} r_j = 1, \quad (2-42)$$

$$\sum_{i \in V} x_{ij} = 1 - r_j \quad \forall j \in V, \quad (2-43)$$

$$\sum_{i \in V} y_{ijk} - \sum_{i \in V} y_{jik} = -r_j \quad \forall j, k \in V, j \neq k, \quad (2-44)$$

$$\sum_{i \in V} y_{ijj} = 1 - r_j \quad \forall j \in V, \quad (2-45)$$

$$y_{ijk} \leq x_{ij} \quad \forall (i, j) \in A, k \in V, \quad (2-46)$$

$$\sum_{(i,j) \in A} y_{ijk} \leq L \quad \forall k \in V, \quad (2-47)$$

$$y_{ijk} \in [0, 1] \quad \forall (i, j) \in A, k \in V, \quad (2-48)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A, \quad (2-49)$$

$$r_j \in \{0, 1\} \quad \forall j \in V. \quad (2-50)$$

A restrição (2-42) estabelece que a árvore contém apenas um nó central. As equações (2-43) asseguram que existe um arco incidindo em cada nó do grafo, exceto na raiz. As restrições (2-44), (2-45) e (2-46) são de conservação de fluxo. Nas inequações (2-44), se  $r_j = 0$  então o nó  $j$  pode ser utilizado como

transbordo para a comodidade  $k$  ( $k \neq j$ ). As inequações (2-45) garantem que todos os nós recebem uma unidade de fluxo e o nó raiz nenhuma. Em (2-46) é estabelecido que se o arco  $(i, j)$  não encontra-se na solução, então nenhum fluxo passa por  $(i, j)$ . As restrições (2-47) limitam o diâmetro da árvore geradora. As restrições (2-48), (2-49) e (2-50) correspondem respectivamente à integralidade das variáveis  $y$ ,  $x$  e  $r$ .

A versão do modelo para o caso ímpar é definida pelas expressões de (2-51) a (2-60):

$$\min \sum_{(i,j) \in A} c_{ij} \cdot x_{ij} + \sum_{[i,j] \in E} c_{ij} \cdot r'_{ij} \quad (2-51)$$

sujeito a:

$$\sum_{[i,j] \in E} r'_{ij} = 1, \quad (2-52)$$

$$\sum_{i \in V} x_{ik} = 1 - \sum_{[i,j] \in E(k)} r'_{ij} \quad \forall k \in V, \quad (2-53)$$

$$\sum_{i \in V} y_{ijk} - \sum_{i \in V} y_{jik} \geq - \sum_{[i,j] \in E(j)/E(k)} r'_{ij} \quad \forall j, k \in V \text{ e } j \neq k, \quad (2-54)$$

$$\sum_{i \in V} y_{ijj} = 1 - \sum_{[i,j] \in E(j)} r'_{ij} \quad \forall j \in V, \quad (2-55)$$

$$y_{ijk} \leq x_{ij} \quad \forall (i, j) \in A; k \in V, \quad (2-56)$$

$$\sum_{(i,j) \in A} y_{ijk} \leq L \quad \forall k \in V, \quad (2-57)$$

$$y_{ijk} \in [0, 1] \quad \forall (i, j) \in A, k \in V, \quad (2-58)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A, \quad (2-59)$$

$$r'_{ij} \in \{0, 1\} \quad \forall [i, j] \in E. \quad (2-60)$$

A função objetivo (2-51) foi adaptada para contabilizar o custo da aresta central. A equação (2-52) garante que a árvore contém apenas uma aresta central. A igualdade (2-53) estabelece que existe um arco incidindo em cada nó do grafo, exceto os nós que são as extremidades da raiz. As desigualdades (2-54) representam a conservação de fluxo. Nessas restrições, os nós nas extremidades da raiz apenas enviam fluxo. As equações (2-55) garantem que todos os nós recebem uma unidade de fluxo, exceto as extremidades da raiz. As desigualdades (2-56) estabelecem que se o arco  $(i, j)$  não encontra-se na solução, então nenhum fluxo passa por  $(i, j)$ . As restrições (2-57) limitam o diâmetro da árvore geradora. As restrições (2-58), (2-59) e (2-60) correspondem respectivamente à integralidade das variáveis  $y$ ,  $x$  e  $r$ .

Ambos os modelos, para os casos par e ímpar, contêm  $O(|E| \cdot |V| + |V|^2)$  restrições e  $O(|E| \cdot |V|)$  variáveis contínuas. Para os casos par e ímpar existem, respectivamente,  $O(|E| + |V|)$  e  $O(|E|)$  variáveis inteiras.

Essas formulações produzem limites inferiores melhores do que a formulação multifluxo básica descrita nesta seção. Gouveia e Magnanti [28] desenvolveram também modelos baseados nas formulações para AGMS [17, 25, 26, 27]. Entretanto, são particularmente interessantes em casos onde o diâmetro requerido é pequeno (4 ou 5, por exemplo). Vale ressaltar que diâmetros pequenos são difíceis para a resolução através de métodos exatos.

## 2.4 Heurísticas e metaheurísticas presentes na literatura

As heurísticas para AGMD descritas na literatura utilizam duas maneiras para controlar o diâmetro da árvore. A primeira considera que a árvore é orientada a partir de um nó central (caso par) ou a partir das extremidades da aresta central (caso ímpar). Neste caso, soluções viáveis com essa estrutura obedecem à propriedade de viabilidade de árvores definida por [35] e descrita no início deste Capítulo. O diâmetro é controlado através de uma contagem de arestas no caminho do nó central no caso par (resp. extremidades da aresta central no caso ímpar) até os outros nós da árvore. A estratégia usada é a identificação da profundidade de cada nó. Deste modo, cada nó que entra na árvore terá seu respectivo nível de profundidade igual à profundidade de seu pai mais um. A atualização da profundidade de um novo nó que entra na árvore consome no pior caso tempo  $O(1)$ . Entretanto, fixar o centro da árvore pode impedir a inserção de arestas na solução que de fato não violam o diâmetro. Este fato pode ser observado através da Figura 2.5.

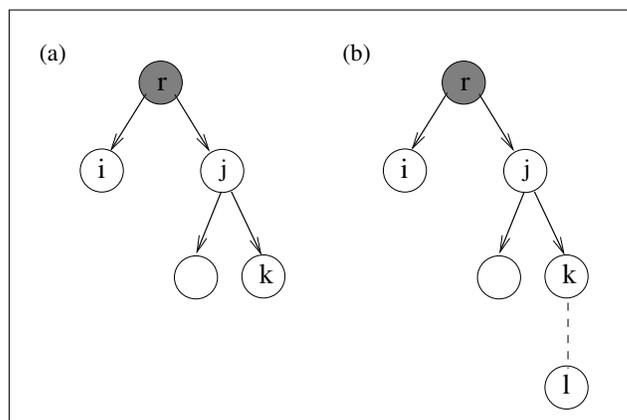


Figura 2.5: Exemplo de falsa violação de diâmetro.

Procura-se uma árvore com diâmetro menor ou igual a 4. A Figura 2.5-(a) é o estado atual da construção. Supondo-se que o nó  $l$  é inserido na solução nesta iteração, seja  $[k, l]$  a aresta de menor custo dentre todas aquelas fora da solução que possuem  $l$  como extremidade. A aresta  $[k, l]$  não pode ser incluída na solução porque, a partir do nó central  $w$ , a profundidade de  $l$  é maior que

$D/2$ . Entretanto, inserir a aresta  $[k, l]$  de fato não violaria o diâmetro, desde que o nó central fosse considerado como sendo o nó  $j$ , como pode ser observado na Figura 2.5-(b).

A segunda maneira de controlar o diâmetro é manter as distâncias entre todos os pares de nós (chamada nesta tese de estrutura não-orientada). Para cada novo nó que entra na árvore, sua distância em relação aos nós que já se encontravam na solução é calculada. Esta operação tem complexidade  $O(|V|)$ . Neste tipo de estrutura, como o nó central no caso par (resp. a aresta central no caso ímpar) não é fixado, então não ocorre o problema de bloquear a inserção de arestas que de fato não violam o diâmetro. Entretanto, a inclusão de um novo nó na solução é realizado com complexidade  $O(|V|)$ , enquanto que na estrutura orientada isto é feito em  $O(1)$ .

As heurísticas RGH e OTT utilizam respectivamente as estruturas orientada e não-orientadas para controlar o diâmetro. Existe garantia de se encontrar uma solução viável com ambas heurísticas somente em grafos completos.

### 2.4.1 One Time Tree

A heurística OTT (*One Time Tree*) procura construir uma solução viável a partir do algoritmo de Prim, apresentado por exemplo em [15]. O algoritmo de Prim obtém uma árvore geradora de custo mínimo através de uma estratégia gulosa. A cada passo, é acrescentada à árvore a aresta de custo mínimo que não provoca a formação de um ciclo. Além dessa verificação, para se construir uma árvore para AGMD, é testado se a aresta não viola a restrição de diâmetro.

O pseudo-código para a heurística OTT é apresentado no Algoritmo 1. Neste algoritmo,  $T^v$  contém os nós que já entraram na solução,  $T^a$  é o conjunto de arestas pertencentes à árvore e  $Q$  contém os nós que ainda estão fora da solução. Para cada nó  $j \in Q$ , chama-se de *chave* $[j]$  o índice do nó  $i \in T^v$  cuja inclusão da aresta  $[i, j]$  na árvore não viola a restrição de diâmetro e o custo  $c_{ij}$  é mínimo dentre as arestas com extremidade em  $j$ . Se *chave* $[j]$  tiver valor igual a -1, isto indica que não se pode incluir o nó  $j$  na solução na iteração considerada. A matriz *distância* contém a distância entre todos os pares de nós de  $T^a$ . Um vetor auxiliar, chamado de *max\_distância* é utilizado para guardar a quantidade de arestas no maior caminho de  $i \in T^v$  para algum outro nó pertencente à solução. Se o valor *max\_distância* $[i]$  é igual ao diâmetro, então nenhum nó pode ser conectado a  $i$  porque violaria a restrição de diâmetro. A variável auxiliar  $c_{min}$  guarda o valor do menor custo dentre todas as arestas candidatas a entrarem na solução.

As estruturas *distância* e *max\_distância* são inicializadas respectivamente nas linhas 1 e 2. Na linha 3, um nó  $i \in V$  é escolhido aleatoriamente. O

```

Algoritmo: OTT()
Entrada:  $G = (V, E)$ 
Saída:  $T^a$ 

1   $distância[i][j] \leftarrow 0 \quad \forall i, j \in V;$ 
2   $max\_distância[i] \leftarrow 0 \quad \forall i \in V;$ 
3  Selecionar aleatoriamente um nó  $i \in V;$ 
4   $T^v \leftarrow \{i\};$ 
5   $Q \leftarrow V - \{i\};$ 
6   $T^a \leftarrow \emptyset;$ 
7  enquanto ( $Q \neq \emptyset$ ) faça
8       $c_{min} \leftarrow \infty;$ 
9      para todo ( $j \in Q$ ) faça
10          $chave[j] \leftarrow -1;$ 
11          $melhor \leftarrow \infty;$ 
12         para todo ( $i \in T^v$ ) faça
13             se ( $c_{ij} < melhor$ ) e ( $max\_distância[i] + 1 \leq D$ ) então
14                  $chave[j] \leftarrow i;$ 
15                  $melhor \leftarrow c_{ij};$ 
16         fim
17     fim
18      $i \leftarrow chave[j];$ 
19     se ( $c_{ij} < c_{min}$ ) e ( $i \neq -1$ ) então
20          $c_{min} \leftarrow c_{ij};$ 
21          $u \leftarrow i;$ 
22          $v \leftarrow j;$ 
23     fim
24 fim
25  $T^v \leftarrow T^v \cup \{v\};$ 
26  $Q \leftarrow Q - \{v\};$ 
27  $T^a \leftarrow T^a \cup \{[u, v]\};$ 
28 Atualizar  $distância$  e  $max\_distância$  de  $v$  em relação a todos
    os nós na solução;
29 fim
30 retorna  $T^a;$ 

```

**Algoritmo 1:** Pseudo-código da heurística OTT.

conjunto  $T^v$  recebe o primeiro nó da árvore na linha 4. Na linha 5,  $Q$  recebe os nós que encontram-se fora da árvore. O conjunto de arestas da árvore  $T^a$  é inicializado na linha 6. O laço das linhas de 7 a 29 é repetido até que todos os nós sejam incluídos na solução. Vale ressaltar que alguns cuidados precisam ser tomados em instâncias de grafos esparsos. Isto porque não existe garantia que uma solução viável seja encontrada. Pode-se, por exemplo, incluir as arestas que não existem com custos muito elevados. Ao término do algoritmo, se uma dessas arestas existir na solução, significa que não foi encontrada uma solução viável.

A variável  $c_{min}$  é inicializada na linha 8. No laço das linhas de 9 a 17, são coletados os valores  $chave[j]$  para cada nó  $j \in Q$  e escolhe-se uma aresta para entrar na solução. Nas linhas 10 e 11,  $chave[j]$  e  $melhor$  são inicializadas. O laço das linhas de 12 a 17 é repetido para cada nó que já está na solução. No teste realizado na linha 13 é averiguado se o custo da aresta  $[i, j]$  é menor do que o custo da aresta de custo mínimo encontrada até o estado corrente e se a inclusão de  $[i, j]$  mantém a viabilidade da solução. Satisfeitas estas condições,  $chave[j]$  e  $melhor$  são atualizadas respectivamente nas linhas 14 e 15.

Escolhe-se, nas linhas 18 a 23, uma aresta para entrar na árvore. Inicialmente, na linha 18,  $i$  recebe o valor armazenado em  $chave[j]$ . Se foi encontrada anteriormente uma aresta  $[i, j]$  de custo mínimo dentre aquelas do conjunto  $\{[i, j] : j \in Q, i \in T^v\}$  e a inclusão de  $[i, j]$  mantém a viabilidade da solução, então  $i$  é diferente de -1. Satisfeitas estas condições, o custo e as extremidades da aresta  $[i, j]$  são guardados respectivamente nas linhas 20 a 22.

O nó  $v$  é incluído em  $T^v$  na linha 25 e excluído de  $Q$  na linha 26. O conjunto de arestas da árvore  $T^a$  é atualizado na linha 27. As estruturas *distância* e *max.distância* relativas ao nó  $v$  que entrará na árvore são atualizadas na linha 28. O procedimento retorna  $T^a$  na linha 30.

A complexidade de pior caso deste algoritmo é  $O(|V|^3)$ . Esta heurística é sensível ao nó pelo qual se inicia a construção. Testar cada nó como ponto de partida da construção aumenta a complexidade do algoritmo para  $O(|V|^4)$ .

### 2.4.2 Randomized Greedy Heuristic

Na heurística RGH (*Randomized Greedy Heuristic*), escolhe-se um nó aleatoriamente a cada iteração. Este nó é inserido na solução através de uma aresta (com a outra extremidade em um nó pertencente à solução) de custo mínimo que não provoque uma violação da restrição de diâmetro.

O pseudo-código da heurística RGH está descrito no Algoritmo 2. Neste algoritmo,  $T^v$  contém os nós que já entraram na solução,  $T^a$  é o conjunto de arestas pertencentes à árvore e  $Q$  contém os nós que ainda estão fora da solução. O vetor  $p$  guarda a profundidade de cada nó de  $T^v$  em relação ao nó central no caso par ou a extremidade mais próxima da aresta central no caso ímpar. A variável  $melhor$  guarda o menor custo dentre aqueles das arestas que podem entrar na solução.

Na linha 1, o vetor  $p$  é inicializado. Se o diâmetro requerido é par (linha 2), então um nó é escolhido aleatoriamente para ser o centro da árvore na linha 3. A profundidade deste nó é atualizada na linha 4. O nó  $v$  é incluído em  $T^v$  na linha 5. Os nós fora da solução são incluídos em  $Q$  na linha 6. Caso o diâmetro requerido seja ímpar (linha 7), então uma aresta é escolhida aleatoriamente

```

Algoritmo: RGH()
Entrada:  $G = (V, E)$ 
Saída:  $T^a$ 

1  $p[v] \leftarrow \infty \quad \forall v \in V;$ 
2 se ( $D$  é par) então
3   Selecionar aleatoriamente um nó  $v \in V;$ 
4    $p[v] \leftarrow 0;$ 
5    $T^v \leftarrow \{v\};$ 
6    $Q \leftarrow V - \{v\};$ 
7 senão
8   Selecionar aleatoriamente uma aresta  $[u, v] \in E;$ 
9    $p[u] \leftarrow 0;$ 
10   $p[v] \leftarrow 0;$ 
11   $T^v \leftarrow \{u, v\};$ 
12   $Q \leftarrow V - \{u, v\};$ 
13   $T^a \leftarrow \{[u, v]\};$ 
14 fim
15 enquanto ( $Q \neq \emptyset$ ) faça
16   Selecionar aleatoriamente um nó  $v \in Q;$ 
17    $melhor \leftarrow \infty;$ 
18   para todo ( $i \in T^v$ ) faça
19     se ( $c_{iv} < melhor$ ) e ( $p[i] + 1 \leq D/2$ ) então
20        $u \leftarrow i;$ 
21        $melhor \leftarrow c_{uv};$ 
22     fim
23   fim
24    $T^v \leftarrow T^v \cup \{v\};$ 
25    $Q \leftarrow Q - \{v\};$ 
26    $T^a \leftarrow T^a \cup \{[u, v]\};$ 
27    $p[v] \leftarrow p[u] + 1;$ 
28 fim
29 retorna  $T^a;$ 

```

**Algoritmo 2:** Pseudo-código da heurística RGH.

para ser o centro da árvore na linha 8. A profundidade das extremidades da aresta central são atualizadas respectivamente nas linha 9 e 10. Os nós  $u$  e  $v$  são incluídos em  $T^v$  na linha 11.  $Q$  recebe os nós fora da solução na linha 12. A solução recebe a primeira aresta  $[u, v]$  na linha 13.

O laço das linhas de 15 a 28 é repetido até que não existam mais nós a serem incluídos na solução. O mesmo cuidado com instâncias de grafos esparsos, citado anteriormente para a heurística OTT, deve ser tomado também para a heurística RGH.

Um nó é escolhido aleatoriamente na linha 16 para entrar na solução. Na linha 17, a variável  $melhor$  é inicializada. No laço das linhas de 18 a 23, procura-se um nó  $i$  pertencente à solução para conectar  $v$ , de modo que a aresta

resultante  $[i, v]$  tenha custo mínimo e mantenha a viabilidade da solução. No teste realizado na linha 19, é averiguado se o custo da aresta  $[i, v]$  é menor do que o custo da aresta de custo mínimo encontrada até o estado corrente e se a restrição de diâmetro é respeitada. Satisfeitas essas condições, guarda-se o nó  $i$  e o custo da aresta  $[i, v]$ , respectivamente nas linhas 20 e 21. O nó  $v$  é inserido em  $T^v$  e excluído de  $Q$ , respectivamente nas linhas 24 e 25. O conjunto de arestas da árvore  $T^a$  e a profundidade do nó  $v$  são atualizados respectivamente nas linhas 26 e 27. O procedimento retorna  $T^a$  na linha 29. Este algoritmo possui complexidade de pior caso  $O(|V|^2)$ .

### 2.4.3

#### Heurísticas de refinamento iterativo da AGM

Heurísticas de refinamento para AGMD são apresentadas em [1, 20]. Dado um grafo  $G = (V, E)$  e uma árvore geradora de custo mínimo de  $G$ , a idéia central dessas heurísticas é tentar refinar iterativamente a AGM até que a restrição de diâmetro seja respeitada. O refinamento é realizado seguindo dois princípios diferentes. No primeiro, os custos das arestas no centro de caminhos longos são penalizados e a AGM é calculada novamente. O segundo princípio consiste em quebrar a árvore através da remoção de arestas e em seguida conectar as sub-árvores através da inserção de arestas com extremidades nas mesmas. Diferentes critérios de penalização, remoção e inserção de arestas foram examinados em [1, 20], gerando várias variantes de heurísticas.

A primeira heurística de refinamento proposta em [1] penaliza os custos de algumas arestas em caminhos da árvore maiores que o diâmetro requerido. Em seguida, a árvore geradora de custo mínimo é calculada novamente. Duas outras heurísticas utilizam o segundo princípio citado acima. Ambas utilizam o mesmo critério de remoção de arestas, mas a forma de conexão das sub-árvores é diferente. Na operação de quebra de árvore, arestas próximas ao centro da árvore são incluídas em uma lista de candidatos a saírem da solução. A escolha de um dos candidatos é realizada considerando-se a aresta com o maior custo associado. Uma aresta é removida a cada iteração gerando duas sub-árvores. Uma estratégia para conectar as sub-árvores geradas propostas em [1] consiste em procurar os nós centrais de cada uma dessas sub-árvores e conectá-los. Uma outra estratégia conecta as sub-árvores considerando que a aresta que entrará na solução é a que possui o menor custo e que reduz o tamanho do caminho quebrado em pelo menos uma aresta.

A calibração destas heurísticas é difícil porque existem muitos parâmetros e aspectos a serem avaliados. Por exemplo: escolher as arestas a saírem da solução e definir a quantidade de arestas a terem seus custos penalizados (aspecto da primeira heurística descrita acima). Existem ainda outros problemas

como ciclagem e dificuldade de se obter uma solução viável ao final desses algoritmos.

#### 2.4.4

##### Heurísticas baseadas em metaheurísticas

O algoritmo genético (AG1) proposto em [56] representa as soluções como listas de arestas que definem uma árvore geradora. A aptidão de um indivíduo corresponde ao custo total da árvore. A população inicial é gerada através da heurística RGH apresentada na Seção 2.4.2. Recombinação e mutação são os operadores utilizados. O operador de recombinação gera um indivíduo a partir de arestas escolhidas de dois conjuntos formados com dois indivíduos dados. O primeiro conjunto possui arestas comuns aos dois indivíduos, incidentes a vértices com profundidade menor que  $D/2$ . O segundo conjunto possui arestas presentes em apenas um dos dois indivíduos, também incidentes a vértices com profundidade menor que  $D/2$ . Quatro tipos de mutações são utilizadas. No primeiro tipo, uma aresta é removida aleatoriamente de um indivíduo. Um novo indivíduo é gerado utilizando recombinação como descrito acima. No segundo tipo de mutação, um nó adjacente ao nó central da árvore (resp. aresta adjacente à aresta central) é escolhido aleatoriamente e torna-se o centro da árvore no caso par (resp. no caso ímpar). No terceiro tipo, uma aresta também é removida aleatoriamente de um indivíduo, como no primeiro tipo de mutação, mas as sub-árvores resultantes são conectadas através da aresta de menor custo com extremidades nessas sub-árvores, mantendo a viabilidade da solução. Finalmente, no quarto tipo de mutação, um nó predecessor de um nó folha é considerado como nó central da árvore. Neste algoritmo genético, inicialmente, é gerada uma população de 400 indivíduos utilizando-se a heurística RGH. Indivíduos descendentes da população inicial são gerados utilizando-se recombinação. Em seguida, uma das estratégias de mutação apresentadas acima é aplicada aos indivíduos obtidos a partir da recombinação. O critério de parada estabelecido termina sua execução quando a melhor solução conhecida não é aprimorada após a geração de 100.000 novos indivíduos.

Um segundo algoritmo genético (AG2) é proposto em [42]. Os indivíduos são representados por permutações de vértices. A posição de cada vértice no cromossomo indica a ordem em que a heurística deve incluí-lo na solução. A heurística RGH é utilizada para calcular a árvore. A avaliação da aptidão de um indivíduo é realizada através do custo total da árvore. A população inicial é composta de permutações geradas aleatoriamente. Recombinação e mutação são os operadores utilizados. O operador de recombinação é similar ao proposto em [56]. Na mutação utilizada, dois vértices são trocados aleatoriamente de posição no cromossomo. Neste algoritmo, uma população inicial de 400 in-

divíduos é gerada. O operador de recombinação é utilizado para gerar um novo indivíduo e, em seguida, a mutação é aplicada. O critério de parada utilizado é o mesmo de AG1.

Uma variante (AG3) da heurística AG2 é apresentada em [43]. Nesta variante, os indivíduos são representados diferentemente de AG2. Um exemplo do tipo de representação utilizado é dado a seguir. Considera-se que os vértices de um grafo de cinco nós (com índices de 0 a 4) são representados respectivamente pelos valores (0,55, 0,43, 0,60, 0,11, 0,70) escolhidos aleatoriamente no intervalo [0,1]. Isto significa que o nó de índice 0 passa a ser identificado por 0,55, o nó de índice 1 é identificado por 0,43 e assim por diante. Então, ordenando-se aqueles valores, obtêm-se (0,11, 0,43, 0,55, 0,60, 0,70). O indivíduo (permutação) correspondente é dado por (3, 1, 0, 2, 4). Uma vantagem deste tipo de representação é que em operações do tipo *crossover*, os indivíduos gerados são sempre válidos, como apresentado em [43]. Depois de obter uma permutação deste tipo, o procedimento AG3 é similar ao AG2. Entretanto, o critério de parada é diferente e estabelece que o algoritmo termina depois de  $1000 \cdot |V|$  novos indivíduos (permutações) terem sido gerados.

Uma heurística do tipo VNS que utiliza um esquema VND como estratégia de busca local é apresentada em [33]. As soluções são árvores orientadas a partir de um nó central no caso par (resp. a partir das extremidades da aresta central no caso ímpar). A heurística RGH é utilizada para gerar uma solução inicial. Quatro tipos de vizinhanças são utilizadas. Na primeira vizinhança, um movimento consiste em remover uma aresta da árvore e reconectar as duas sub-árvores resultantes. Um movimento na segunda vizinhança consiste em trocar na árvore um nó  $v$  por um de seus filhos  $u$  na árvore. O nó  $v$  torna-se uma folha, pois todos os seus filhos são transferidos para o nó  $u$ . No terceiro tipo de movimento, um nó é transferido para outro nível de profundidade. No quarto tipo de movimento, o nó central no caso par é trocado com outro nó da árvore e o antigo nó central é conectado como um nó folha no maior nível de profundidade da árvore. No caso ímpar, é trocada uma das extremidades da aresta central. A primeira, a segunda, a quarta e a terceira vizinhanças são exploradas nesta ordem em um esquema do tipo VND. Uma solução inicial é gerada e a busca local do tipo VND é aplicada a mesma. Quando não existem mais melhorias a serem realizadas usando as quatro vizinhanças, uma das vizinhanças é escolhida aleatoriamente. Movimentos aleatórios são aplicados à melhor solução conhecida, utilizando a vizinhança escolhida. A solução resultante é submetida ao VND. Este processo é repetido até que o critério de parada seja atingido. O procedimento pára se 1000 aplicações consecutivas de movimentos aleatórios forem realizadas sem melhorar a melhor solução conhecida ou quando um limite de tempo estabelecido é atingido.

Um resumo dos resultados obtidos a partir das heurísticas AG1, AG2, AG3 e VNS descritas acima são apresentados na Tabela 2.1. Os resultados de AG1 foram extraídos de [56]. Os testes foram realizados em um Pentium III com relógio de 800MHz. Nos testes realizados com AG2 e AG3, respectivamente em [42] e [43], não foi informado o equipamento utilizado. Os testes realizados com o VNS em [33] foram realizados em um Pentium 4 com relógio de 2.8 GHz. A memória RAM não foi informada. As heurísticas AG1, AG2, AG3 foram executadas 50 vezes para cada instância e o VNS foi executado 30 vezes para cada instância.

As instâncias utilizadas são provenientes da seção da OR-Library relativa às instâncias euclidianas do problema de Steiner. A biblioteca contém instâncias em grafos completos com diferentes quantidades de vértices. Para cada quantidade de vértices existem 15 instâncias, sendo utilizadas nos experimentos de AG1, AG2, AG3 e VNS as cinco primeiras instâncias com determinada quantidade de vértices. Na Tabela 2.1, para cada instância são indicados nas três primeiras colunas a quantidade  $|V|$  de vértices, a quantidade  $|E|$  de arestas e o valor  $D$  do diâmetro, além do identificador da instância. São mostrados o valor da melhor solução encontrada e a média dos valores das soluções obtidas por cada algoritmo. O símbolo “-” indica que a instância não foi utilizada nos experimentos de um algoritmo. Os valores das melhores soluções conhecidas para uma determinada instância encontram-se destacados em negrito.

Uma comparação direta entre as heurísticas não é possível porque as condições dos experimentos realizados foram diferentes. Entretanto, observa-se que VNS obteve as melhores soluções e as melhores médias dos valores das soluções para as instâncias de tamanho 100, 250, 500 e 1000.

V	E	D	identificador	AG1		AG2		AG3		VNS	
				melhor	média	melhor	média	melhor	média	melhor	média
50	1225	5	1	<b>7,60</b>	7,93	<b>7,60</b>	7,78	7,61	7,85	–	–
			2	<b>7,68</b>	7,87	7,75	7,84	7,75	7,87	–	–
			3	<b>7,24</b>	7,51	7,25	7,41	7,25	7,49	–	–
			4	<b>6,59</b>	6,75	6,62	6,65	6,62	6,71	–	–
			5	<b>7,32</b>	7,49	7,39	7,47	7,39	7,49	–	–
70	2415	7	1	–	–	<b>7,23</b>	7,32	<b>7,23</b>	7,36	–	–
			2	–	–	<b>7,12</b>	7,21	<b>7,12</b>	7,25	–	–
			3	–	–	<b>6,99</b>	7,13	7,01	7,15	–	–
			4	–	–	7,53	7,62	<b>7,52</b>	7,64	–	–
			5	–	–	<b>7,27</b>	7,32	<b>7,27</b>	7,34	–	–
100	4950	10	1	8,00	8,30	7,77	7,89	7,83	7,92	<b>7,76</b>	7,82
			2	8,10	8,41	7,88	8,02	<b>7,85</b>	8,04	<b>7,85</b>	7,89
			3	8,22	8,61	7,93	8,11	7,98	8,14	<b>7,90</b>	7,96
			4	8,27	8,57	8,00	8,13	8,00	8,12	<b>7,98</b>	8,05
			5	8,48	8,72	<b>8,16</b>	8,30	8,20	8,31	8,17	8,20
250	31125	15	1	12,93	13,36	12,41	12,52	12,45	12,58	<b>12,30</b>	12,43
			2	12,86	13,25	12,20	12,37	12,22	12,39	<b>12,02</b>	12,17
			3	12,69	13,06	12,12	12,23	12,18	12,32	<b>12,04</b>	12,11
			4	13,22	13,65	12,64	12,77	12,63	12,80	<b>12,51</b>	12,62
			5	13,02	13,40	12,37	12,56	12,38	12,62	<b>12,28</b>	12,42
500	124750	20	1	18,33	18,77	17,10	17,35	17,16	17,43	<b>16,97</b>	17,13
			2	18,17	18,60	16,97	17,20	17,10	17,29	<b>16,88</b>	17,05
			3	18,33	18,76	17,13	17,28	17,16	17,37	<b>16,98</b>	17,15
			4	18,32	18,74	17,09	17,29	17,27	17,43	<b>16,99</b>	17,17
			5	17,80	18,40	16,79	17,03	16,87	17,09	<b>16,57</b>	16,79
1000	499500	25	1	26,13	26,72	–	–	–	–	<b>25,18</b>	25,57
			2	26,14	26,58	–	–	–	–	<b>25,02</b>	25,34
			3	25,47	26,21	–	–	–	–	<b>24,82</b>	25,09
			4	26,13	26,65	–	–	–	–	<b>25,29</b>	25,57
			5	25,91	26,29	–	–	–	–	<b>25,03</b>	25,25

Tabela 2.1: Resultados apresentados na literatura das heurísticas baseadas em metaheurísticas.