Universidade de Aveiro Departamento de Matemática
2007

Eulália Maria Mota Santos

Problema da Árvore de Suporte de Custo Mínimo com Restrições de Diâmetro

Universidade de Aveiro Departamento de Matemática
2007

Eulália Maria Mota Santos

Problema da Árvore de Suporte de Custo Mínimo com Restrições de Diâmetro

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Matemática, realizada sob a orientação científica da Doutora Maria Cristina Saraiva Requejo Agra, Professora Auxiliar do Departamento de Matemática da Universidade de Aveiro.

o júri

presidente

Professor Doutor Domingos Moreira Cardoso
professor Catedrático da Universidade de Aveiro

Professor Doutor Luís Eduardo Neves Gouveia
professor Associado com Agregação da Faculdade de Ciências da Universidade de Lisboa

Professora Doutora Maria Cristina Saraiva Requejo Agra
professora Auxiliar da Universidade de Aveiro (Orientadora)

agradecimentos

Desejo expressar os meus sinceros agradecimentos à Professora Doutora Maria Cristina Saraiva Requejo Agra pela oportunidade que me deu de realizar este trabalho sob a sua orientação e pela disponibilidade e apoio que me prestou ao longo da sua elaboração.
Gostaria também de agradecer aos meus pais, amigos e a todos os que têm contribuído para a minha formação.

palavras-chave Árvores com restrições de diâmetro, Árvores, Heurísticas.

resumo

Nesta tese desenvolvem-se alguns métodos heurísticos para o Problema da Árvore de Suporte de Custo Mínimo com Restrições de Diâmetro que é um problema de Optimização Combinatória. Este problema tem aplicação na área das telecomunicações e insere-se no âmbito de problemas do Desenho Topológico de Redes de Telecomunicações. O Problema do Desenho de Redes de Terminais consiste em encontrar a melhor maneira de ligar *n* terminais em diferentes localizações a um nodo central. A topologia óptima deste problema corresponde a uma árvore de suporte de custo mínimo. No Problema da Árvore de Suporte de Custo Mínimo com Restrições de Diâmetro pretende-se determinar uma árvore de suporte de custo mínimo cujo diâmetro não ultrapasse um determinado valor máximo (D). Esta imposição melhora o desempenho da rede.

Apresentam-se três heurísticas greedy que seleccionam iterativamente uma aresta a ser incluída na árvore e que se distinguem apenas na forma como são escolhidos os elementos iniciais (nodo/aresta). Descreve-se uma heurística de trocas locais (ou melhoramento) que efectua algumas trocas de arestas de acordo com uma regra estabelecida. Descrevem-se quatro heurísticas de aproximação que adaptam soluções de outro problema ao problema em questão. Na primeira destas heurísticas eliminam-se arestas da árvore de suporte de custo mínimo e, depois, constrói-se a árvore a partir da subárvore obtida. Na segunda proíbe-se a presença na solução de cada uma das arestas de um dado conjunto. Na terceira heurística exige-se que cada aresta de um dado conjunto esteja na solução e, na última exige-se que cada uma das arestas de um dado conjunto esteja na solução e que um conjunto de arestas não esteja na solução.

Apresentam-se resultados computacionais que mostram que as Heurísticas de Aproximação são as que obtêm melhores resultados.

keywords Diameter constrained trees, Spanning trees, Heuristics.

abstract

In this thesis we present some heuristics methods developed to the Diameter constrained Minimum Spanning Tree problem (DMST), which is a Combinatorial Optimization Problem. This is a telecommunication network design problem and the terminal layout problem consists of finding the best way to link *n* terminals, at different locations, to a central node. The optimal topology for these problems corresponds to a minimum spanning tree. In the DMST we want to obtain a minimum spanning tree which diameter does not surpass a maximum value (D). The diameter constraint improves the performance of the network.

We present three greedy heuristics that iteratively select an edge to be added to the tree and are distinguished in the form how initial elements (a node or an edge) are selected. We describe a local exchanges heuristic where improvements are accomplished with some edges exchanges according to an established rule. We also describe four approximation heuristics that adapt solutions from another problem to this problem. On the first heuristic we start it by eliminating edges from the minimal spanning tree and then we build the new tree from the obtained subtree. On the second heuristic, the presence of each edge of a certain set is forbidden in the solution. On the third heuristic, it is demanded that each edge of a certain set is present in the solution, and on the last heuristic it is demanded that each one of the edges of a certain set is present in the solution and that a set of edges is not in the solution.

Our computational experience shows that the best results are achieved with the approximation heuristics.

Conteúdo

Capítulo 1

Introdução

O Problema da Árvore de Suporte de Custo Mínimo com Restrições de Diâmetro é um Problema de Optimização Combinatória, que tem aplicação na área das telecomunicações inserindo-se no âmbito de Problemas de Desenho Topológico de Redes de Telecomunicações. O Problema do Desenho de Redes de Terminais consiste em encontrar a melhor maneira de ligar n terminais em diferentes localizações a um nodo central. A topologia óptima deste problema corresponde a uma árvore de suporte de custo mínimo.

Muitas soluções para o problema de Redes de Terminais possuem caminhos do nodo central para alguns terminais com um comprimento muito grande. Por esse motivo caminhos com um comprimento muito grande podem degradar a *performance* da rede uma vez que, a existência de uma grande quantidade de nodos, pode originar um tempo de transmissão bastante elevado. Um caminho com um comprimento muito grande pode afectar a fiabilidade da rede, pois a probabilidade de existir uma falha numa das ligações é directamente proporcional ao comprimento desse caminho [26]. Por esse motivo, uma vantagem das restrições de diâmetro é a de poderem ser usadas para evitar a degradação da qualidade do sinal nas redes.

O objectivo do Problema da Árvore de Suporte de Custo Mínimo com Restrições de Diâmetro (DMST) é o de encontrar uma árvore de suporte cujo diâmetro seja, no máximo, um valor D. Devido às propriedades do diâmetro [6] de uma árvore, temos de distinguir este problema em dois casos: o caso do diâmetro D ser par e o caso do diâmetro D ser ímpar.

Existem alguns problemas de Optimização Combinatória [16] para os quais se conhecem algoritmos eficientes que, em tempo polinomial, permitem obter a solução óptima. Um exemplo de tais problemas é o Problema da Árvore de Suporte de Custo Mínimo que pode ser resolvido eficientemente usando, por exemplo, o algoritmo de Prim [5, 21] ou o algoritmo de Kruskal [5]. O problema da Árvore de Suporte de Custo Mínimo com Restrições de Diâmetro

torna-se mais complicado, pois é um problema de árvore de suporte de custo mínimo com restrições adicionais (as restrições de diâmetro). Este problema pertence à chamada classe dos problemas *NP*- difíceis quando $D \geq 4$ (quando $D = 2$ ou 3 o problema é de fácil resolução) [8].

A dificuldade de resolução de problemas *NP*-difíceis tem levado ao seu estudo e ao desenvolvimento de técnicas de resolução que possibilitem a obtenção de bons limites para o valor da solução óptima, entre os quais as técnicas heurísticas.

Na literatura existem alguns trabalhos que apresentam heurísticas para o problema da Árvore de Suporte de Custo Mínimo com Restrições de Diâmetro. Nos trabalhos de Abdalla, Deo, Kumar e Terry [4], Abdalla, Deo e Fraceschini [3] e Deo e Abdalla ([1] e [2]) podemos encontrar duas heurísticas para o problema e nestes trabalhos os autores examinam implementações em paralelo dessas heurísticas. Uma das heurísticas apresentada é conhecida na literatura por Heurística OTTC e é baseada na forma modificada de greedy, seleccionando em cada passo uma aresta para adicionar à árvore que vai sendo construída. A segunda heurística conhecida por Heurística CIR refina iterativamente a árvore de suporte de custo mínimo até que a restrição do diâmetro seja satisfeita.

O objectivo desta tese é o de desenvolver e comparar heurísticas para o Problema da Árvore de Suporte de Custo Mínimo com Restrições de Diâmetro.

O termo *heurística* está associado à arte de inventar. As heurísticas englobam estratégias, procedimentos, métodos de aproximação tentativa/erro, sempre na procura da melhor forma de chegar a um determinado fim, a "*solução*". Os processos heurísticos exigem muitas vezes menos tempo que outro tipo de processos para determinar soluções, aproximam-se mais da forma como o ser humano raciocina, chega às resoluções dos problemas e garantem soluções eficientes. Quando se recorre a heurísticas a ideia subjacente é a de encontrar rapidamente soluções de boa qualidade.

As heurísticas deverão tirar partido da estrutura particular do problema a ser resolvido por forma a que, soluções satisfatórias sejam encontradas num espaço de tempo razoável para a situação prática a que se destina.

As heurísticas podem ser classificadas em duas grandes classes: as heurísticas específicas e as heurísticas gerais. As heurísticas específicas usam toda a informação que o problema lhes fornece e para cada novo problema é elaborada uma nova heurística. As heurísticas gerais podem ser usadas em todos os problemas, independentemente da sua natureza. As heurísticas apresentadas nesta tese serão baseadas em heurísticas conhecidas na literatura para outros problemas mas, que foram adaptadas para este problema. Podem, por isso ser consideradas heurísticas específicas, pois usam toda a informação que o problema lhes

fornece. São consideradas heurísticas construtivas greedy, heurísticas de trocas locais ou melhoramento e heurísticas de aproximação.

No Capítulo 2 descreve-se o Problema da Árvore de Suporte de Custo Mínimo com Restrições de Diâmetro e apresentam-se duas formulações para este problema: uma para o caso do diâmetro ser par e outra para o caso do diâmetro ser ímpar [12, 13].

No Capítulo 3 propõem-se algumas heurísticas para este problema. Começa-se por apresentar uma tentativa de heurística baseada no algoritmo de Kruskal, a qual não funcionou. De seguida apresentam-se três heurísticas construtivas *greedy*, que seleccionam iterativamente a melhor aresta a ser incluída na solução e apenas diferem na escolha dos elementos iniciais (nodo ou nodos). Em seguida apresenta-se uma heurística de trocas locais ou de melhoramento que efectua algumas trocas de arestas tentando melhorar uma solução já conhecida. Depois são apresentadas quatro heurísticas de aproximação que adaptam soluções de outro problema ao problema em questão. Na primeira heurística de aproximação primeiro eliminam-se as arestas de acordo com uma regra estabelecida e depois, constrói-se a árvore com diâmetro limitado a partir da subárvore obtida. Nas outras três heurísticas a diferença que existe entre elas reside no facto de uma delas obrigar cada uma das arestas de um dado conjunto a estarem na solução enquanto que numa outra se proibe a presença de cada uma das arestas de um dado conjunto na solução, a última heurística reúne os dois processos proibição e obrigação, ou seja, obriga-se cada uma das arestas de um dado conjunto a estarem na solução e ao mesmo tempo proíbe-se a presença de um dado conjunto de arestas na solução. Finalmente será apresentada uma breve descrição das duas heurísticas já consideradas na literatura para este problema [1, 2, 3, 4, 22] e como forma de sintetizar os resultados do exemplo trabalhado ao longo da tese apresenta-se um quadro resumo.

No Capítulo 4 são apresentados resultados computacionais relativos às heurísticas consideradas no Capítulo 3. Serão também apresentados resultados computacionais para uma das heurísticas já consideradas na literatura. Usam-se instâncias do problema que correspondem a grafos completos com um número de nodos entre 10 e 60 e um número de arestas entre 45 e 1770. O valor de D varia entre 4 e 9. Neste capítulo discutem-se ainda os resultados obtidos pelas diversas heurísticas, apresentando-se um estudo comparativo entre elas.

No Capítulo 5 são apresentadas algumas considerações finais.

Capítulo 2

O Problema da Árvore de Suporte de Custo Mínimo com Restrições de Diâmetro

Neste capítulo definimos o problema da Árvore de Suporte de Custo Mínimo com Restrições de Diâmetro que designamos por DMST.

O ***diâmetro*** de um grafo G é a maior distância entre os vértices do grafo [6].

E o problema DMST define-se da seguinte forma:

Considere-se um grafo não direccionado G=(V,E), onde V=$\{1, ..., n\}$ *denota o conjunto de nodos e E=*$\{\{i, j\}, i, j \in V\}$ *o conjunto de arestas. A cada aresta* $e \in E$ *está associado um custo* c_e*. O objectivo do Problema da Árvore de Suporte de Custo Mínimo com Restrições de Diâmetro é o de encontrar a árvore de suporte de menor custo sujeita a um limite D para o seu diâmetro (sendo D um número natural), ou seja, uma árvore de suporte de custo mínimo em que D é o número máximo de arestas em qualquer caminho da árvore.*

Considere-se um grafo para o qual se pretende obter a árvore de suporte de custo mínimo com diâmetro, no máximo, de $D = 3$. Usando, por exemplo, o algoritmo de Kruskal [5] ou o algoritmo de Prim [5, 22] obtém-se a árvore de suporte de custo mínimo da Figura 2.1.

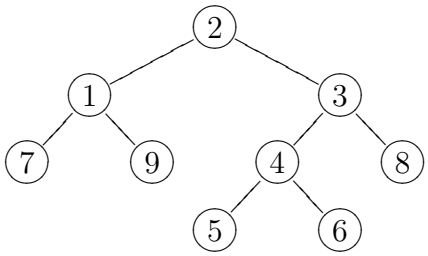


Figura 2.1: Árvore de suporte de custo mínimo, tem diâmetro 5.

Esta árvore tem diâmetro 5 pelo que é uma solução não admissível para o nosso problema, pois viola a restrição de diâmetro. Uma solução admissível seria, por exemplo, a árvore de suporte da Figura 2.2.

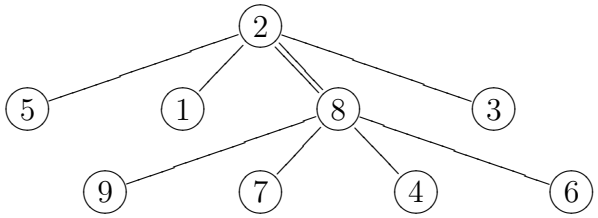


Figura 2.2: Árvore de suporte com diâmetro $D = 3$.

Os modelos de Gouveia e Magnanti [11] para o problema da Árvore de Suporte de Custo Mínimo com Restrições de Diâmetro foram desenvolvidos usando propriedades elementares do diâmetro de árvores. Todas as árvores têm um centro, que pode ser um vértice ou dois vértices adjacentes [6], temos portanto, de distinguir dois casos diferentes para o diâmetro, o caso em que o diâmetro D é par e o caso em que o diâmetro D é ímpar.

Apresentamos, de seguida, as duas propriedades elementares do diâmetro de árvores.

- ***Propriedade 1:*** Uma árvore tem diâmetro não superior a um valor D par se e só se algum nodo p da árvore satisfaz a propriedade de o caminho do nodo p para qualquer outro nodo da árvore conter, no máximo, $\frac{D}{2}$ arestas.
- ***Propriedade 2:*** Uma árvore tem diâmetro não superior a um valor D ímpar se e só se alguma aresta $\{p, q\}$ da árvore satisfaz a propriedade de o caminho ou do nodo p ou do nodo q para qualquer outro nodo da árvore conter, no máximo, $\frac{D-1}{2}$ arestas.

Na Figura 2.2 é apresentada uma solução admissível para o caso do diâmetro D ser ímpar ($D = 3$). Como se pode observar é verificada a Propriedade 2. Neste exemplo a aresta $\{2, 8\}$ é considerada a aresta central e verifica-se a existência de um único caminho de um dos seus nodos extremos, ou do nodo 2 ou do nodo 8, para cada um dos outros nodos do grafo que contém, no máximo, 1 ($\frac{D-1}{2} = \frac{3-1}{2}$) aresta.

Para o caso do diâmetro D ser par (por exemplo, $D = 4$) uma solução admissível pode ser a apresentada na Figura 2.3.

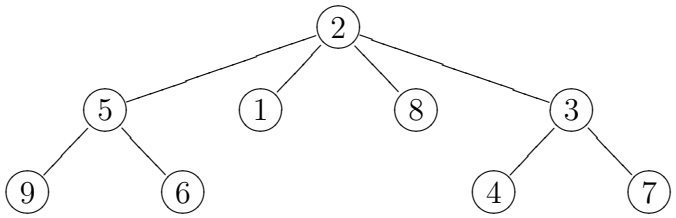


Figura 2.3: Árvore de suporte com diâmetro $D = 4$.

Neste exemplo é verificada a Propriedade 1, pelo que, do nodo 2 que é considerado o nodo central e é assegurada a existência de um único caminho para cada um dos outros nodos do grafo que contém, no máximo, 2 ($\frac{D}{2} = \frac{4}{2}$) arestas.

Nos modelos apresentados em Gouveia e Magnanti [11] são usadas as Propriedades 1 e 2 para descrever as formulações. Para facilitar a descrição das formulações, foi considerada uma rede aumentada, adicionando um novo nodo (nodo 0) e novas arestas $\{0, j\}$ para

todos os nodos $j \in V$, obtendo-se assim a chamada *"rede aumentada"*. Na nova rede aumentada considera-se $V_0 = V \cup \{0\}$ o conjunto de nodos e $E_0 = E \cup \{\{0, j\}, j \in V\}$ o conjunto de arestas. Considere-se ainda a versão orientada, em que cada aresta $\{i, j\} \in E$ é substituída por dois arcos, o arco (i, j) e o arco (j, i) cada um com o mesmo custo c_{ij} da aresta da rede original. Designe-se por $A = \{(i, j), i, j \in V\}$ o conjunto de arcos na rede original. Além disso, cada aresta $\{0, j\}$ é substituída apenas pelo arco $(0, j)$ pelo que se designa por A_0=$A \cup \{(0, j), j \in V\}$ o conjunto de arcos na rede aumentada.

Se S for uma solução admissível para o DMST na rede original, na rede aumentada as correspondentes soluções admissíveis são tais que passam a existir arestas que unem o nodo zero aos elementos centrais da seguinte forma:

- **se o diâmetro D é par:**

  o nodo 0 tem grau 1 e faz-se a sua ligação com o nodo central p, ou seja, adiciona-se a aresta $\{0, p\}$ de custo zero. A solução obtida é $S_0 = S \cup \{\{0, p\}\}$;

- **se o diâmetro D é ímpar:**

  o nodo 0 tem grau 2 e adicionam-se duas arestas, a aresta $\{0, p\}$ e a aresta $\{0, q\}$ ambas de custo zero que unem o nodo 0 aos extremos da aresta central $\{p, q\}$. A solução obtida é $S_0 = S \cup \{\{0, p\}, \{0, q\}\}$.

Nas Figuras 2.4 e 2.5 apresentam-se as soluções admissíveis, na rede aumentada, que correspondem às soluções na rede original apresentadas nas Figuras 2.2 e 2.3, respectivamente.

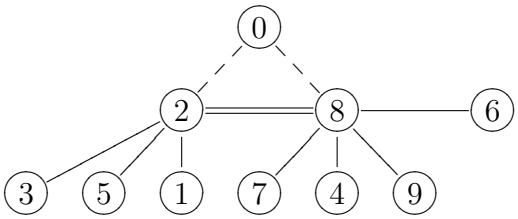


Figura 2.4: Solução na rede aumentada para o caso de D ser ímpar, $D = 3$.

Portanto, para obtermos uma solução admissível para o problema DMST na rede aumentada devemos proceder da seguinte forma.

Para o caso do diâmetro D ser ímpar teremos de seleccionar $n + 1$ arestas de menor custo do conjunto das $|E_0| = \frac{n.(n-1)}{2} + n$ arestas tais que:

- n arestas formam uma árvore de suporte e contém exactamente duas arestas $\{0, i\}$ e $\{0, j\}$ incidentes no nodo 0.
- o único caminho do nodo 0 para cada nodo contém, no máximo, $\frac{D-1}{2} + 1$ arestas.
- a $(n + 1)$-ésima aresta $e = \{i, j\}$ da rede original une os nodos i e j das duas arestas $\{0, i\}$ e $\{0, j\}$.

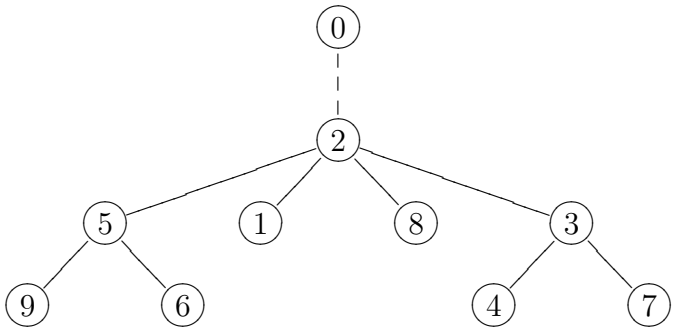


Figura 2.5: Solução na rede aumentada para o caso de D ser par, $D = 4$.

Para o caso do diâmetro D ser par teremos de seleccionar n arestas de custo mínimo do conjunto das $|E_0| = \frac{n.(n-1)}{2} + n$ arestas tais que:

- as n arestas formam uma árvore de suporte e contém exactamente uma aresta $\{0, j\}$ que incide no nodo 0.
- o único caminho do nodo 0 para cada nodo contém, no máximo, $\frac{D}{2} + 1$ arestas.

Apresenta-se de seguida uma formulação de fluxos não orientada para o caso de D ser par e para o caso de D ser ímpar.

O problema para o caso do diâmetro D ser par é formulado da seguinte forma:

$$min\ z = \sum_{e \in E_0} c_e\ x_e$$

s.a:

$$\sum_{e \in E_0} x_e = n$$

$$\sum_{j \in V} x_{\{0,j\}} = 1$$

$$\sum_{j \in V} y_{ij}^k - \sum_{j \in V_0} y_{ji}^k = \begin{cases} 1, & \text{se} \quad i = 0 \\ 0, & \text{se} \quad i \neq 0, k \\ -1, & \text{se} \quad i = k \end{cases} \quad \forall k \in V, \quad i \in V_0 \tag{2.1}$$

$$\sum_{(i,j) \in A_0} y_{ij}^k \leq (D/2) + 1, \quad k \in V$$

$$y_{ij}^k + y_{ji}^k \leq x_e, \quad \forall e = \{i,j\} \in E, \quad k \in V$$

$$y_{0j}^k \leq x_{0j}, \quad j, k \in V$$

$$y_{ij}^k \in \{0,1\}, \quad \forall (i,j) \in A_0, \quad k \in V$$

$$x_e \in \{0,1\}, \quad \forall e \in E_0$$

Este modelo usa vários conjuntos de variáveis: variáveis x_e que indicam se a aresta $\boldsymbol{e}$ está na árvore de suporte de custo mínimo e y_{ij}^k ($(i,j) \in A_0$; $k \in$V) variáveis de fluxo direccionado que especificam se o único caminho da origem com destino ao nodo k atravessa a aresta $\{i,j\}$ no sentido de i para j. As restrições $\sum_{e \in E_0} x_e = n$ são restrições de cardinalidade e garantem que escolhemos n arestas para formar a árvore de suporte não orientada, as restrições $\sum_{j \in V} x_{\{0,j\}} = 1$, garantem que apenas uma aresta é incidente no nodo origem 0

(nodo 0 tem grau 1). As restrições (2.1) são as restrições de conservação de fluxo, a primeira implica que o nodo 0 envia uma unidade de fluxo para o nodo k, a segunda implica que o fluxo que entra num nó é igual ao fluxo que sai desse nó e a terceira implica que o nodo k recebe uma unidade de fluxo. Este conjunto de restrições juntamente com as restrições $\sum_{e \in E_0} x_e = n$ garantem a existência de um caminho entre o nodo 0 e cada nodo k. As restrições $y_{ij}^k + y_{ji}^k \leq x_e$ são restrições de ligação e garantem que é possível enviar fluxo apenas através da aresta $\{i, j\}$ para cada nodo k, se a aresta está na solução da árvore. As restrições $\sum_{\{i,j\} \in A_0} y_(^k ij) \leq (D/2) + 1$ garantem que o caminho entre o nodo 0 e o nodo k não contém mais do que $\frac{D}{2} + 1$ arestas.

O problema para o caso do diâmetro D ser ímpar é formulado da seguinte forma:

$$min\ z = \sum_{e \in E_0} c_e\ x_e + \sum_{e \in E} c_e E_e$$

s.a:

$$\sum_{e \in E_0} x_e = n$$

$$\sum_{e \in E} E_e = 1$$

$$\sum_{i \in V} x_{\{0,i\}} = 2$$

$$x_{\{0,i\}} = \sum_{e \in E(i)} E_e, \quad \forall i \in V$$

$$\sum_{j \in V} y_{ij}^k - \sum_{j \in V_0} y_{ji}^k = \begin{cases} 1, & \text{se} \quad i = 0 \\ 0, & \text{se} \quad i \neq 0, k \\ -1, & \text{se} \quad i = k \end{cases} \quad \forall k \in V, \quad i \in V_0$$

$$\sum_{(i,j) \in A_0} y_{ij}^k \leq ((D-1)/2) + 1, \quad k \in V$$

$$y_{ij}^k + y_{ji}^k \leq x_e, \quad \forall e = \{i, j\} \in E, \quad k \in V$$

$$y_{0j}^k \leq x_{0j}, \quad j, k \in V$$

$$y_{0j}^k \leq x_{\{0,j\}} - E_{jk}, \quad \forall k, j \in V; \quad k \neq j$$

$$y_{ij}^k \in \{0,1\}, \quad \forall (i,j) \in A_0, \quad k \in V$$

$$x_e \in \{0,1\}, \quad \forall e \in E_0$$

$$E_e \in \{0,1\}, \quad \forall e \in E$$

Nesta formulação para além das variáveis consideradas na formulação anterior, consideramos as variáveis E_e para todo $e \in E$ que indicam se a aresta e é ou não a aresta central. A restrição $\sum_{e \in E} E_e = 1$ garante a existência de apenas uma aresta central entre as arestas da rede original. A restrição $\sum_{i \in V} x_{\{0,i\}} = 2$ garante que exactamente duas arestas são incidentes no nodo 0 e, portanto, garante a existência de dois nodos centrais, os nodos adjacentes ao nodo 0. As restrições $x_{\{0,i\}} = \sum_{e \in E(i)} E_e$ garantem que a aresta central é incidente nos nodos centrais. As restrições $\sum_{(i,j) \in A_0} y_{ij}^k \leq ((D-1)/2)+1$ garantem que o caminho entre o nodo 0 e qualquer nodo contém não mais do que $\frac{D-1}{2} + 1$ arestas. A restrição $y_{0j}^k \leq x_{\{0,j\}} - E_{jk}$ garante que se a aresta $\{j,k\}$ estiver na solução então o fluxo para o nodo k não será enviado através da aresta $\{0,j\}$ e será enviado directamente através da aresta $\{0,k\}$.

Observamos que estas formulações foram usadas para a obtenção dos valores óptimos ou limites inferiores que apresentamos no capítulo dos resultados computacionais.

Capítulo 3

Heurísticas para o Problema da Árvore de Suporte de Custo Mínimo com Restrições de Diâmetro

Uma heurística está relacionada com a arte de "inventar" formas de procurar soluções admissíveis tão boas quanto possível, mas sem garantir que seja obtida a solução óptima.

As técnicas heurísticas surgem da necessidade de obter soluções admissíveis para problemas com grande complexidade e para os quais não se conhecem algoritmos polinomiais para a obtenção da sua solução.

*Uma **heurística** é então, um método de resolução que pretende obter uma "boa" solução em tempo computacional razoável sem garantir optimalidade, nem em alguns casos admissíbilidade* [23].

O Problema da Árvore de Suporte de Custo Mínimo com Restrições de Diâmetro é um problema NP-difícil e portanto não se conhecem algoritmos que garantam a resolução de todas as suas instâncias em tempo polinomial. Neste capítulo descrevem-se heurísticas para o problema DMST. Descrevem-se primeiro algumas heurísticas *greedy*, depois, heurísticas de trocas locais ou melhoramento, em seguida heurísticas de aproximação e finalmente faz-se uma breve abordagem a duas heurísticas da literatura e que foram apresentadas por Abdalla et all [1, 2, 3, 4].

No final deste capítulo apresenta-se um quadro resumo com o valor obtido em cada heurística apresentada ao longo deste capítulo.

3.1 Heurísticas "Greedy" Construtivas

As heuristicas "greedy" são métodos que constroem uma solução admissível para o problema seleccionando iterativamente uma aresta candidata a ser incluída na solução [14].

Nesta secção descrevemos algumas heurísticas "greedy" que procuram determinar uma solução admissível para o Problema da Árvore de Suporte de Custo Mínimo com Restrições de Diâmetro.

Começamos por apresentar a primeira tentativa de construção de uma heurística tendo como base o algoritmo de Kruskal e explicamos porque não funcionou. Depois descrevemos três heurísticas baseadas no algoritmo de Prim que diferem na forma de escolha dos elementos iniciais (nodo se D for par ou aresta se D for ímpar).

A PRIMEIRA TENTATIVA

Para a construção de uma primeira tentativa de heurística tentamos fazer alterações ao algoritmo de Kruskal [5] que, contudo, não funciona para o problema DMST. Vamos apresentar a ideia de adaptação e explicar a razão pela qual não nos foi possível usar o algoritmo de Kruskal na construção de heurísticas para este problema.

O algoritmo de Kruskal [5] consiste na ordenação por ordem crescente dos custos das arestas e na escolha sucessiva da aresta de custo mínimo a ser inserida de forma a que a sua inserção na árvore não forme ciclos. Se a inserção de uma aresta formar um ciclo elimina-se da lista ordenada e escolhe-se a aresta seguinte da lista. Este procedimento é repetido até se obterem $n-1$ arestas na solução ou até não existirem mais arestas na lista ordenada.

Uma vez que o objectivo do nosso problema é o de determinar uma árvore abrangente cujo diâmetro não ultrapasse um determinado valor máximo que designamos por D, é necessária a inclusão de um teste que impeça a formação de caminhos com comprimento superior a D. Portanto, a heurística baseada no algoritmo de Kruskal para este problema terá também de

garantir que não existem caminhos com comprimento superior a D. Neste sentido teremos de ir seleccionando sucessivamente a aresta de custo mínimo da lista ordenada de arestas ainda não seleccionadas e inseri-la na árvore de tal forma que a sua inserção não forme ciclos nem caminhos com comprimento superior a D. Este procedimento é repetido até se obterem $n-1$ arestas na solução ou até não existirem mais arestas na lista ordenada. Quando usamos esta forma de construção de uma árvore podemos formar várias componentes conexas tornando complicado garantir a não existência de caminhos de comprimento superior a D.

Vejamos o que acontece com o seguinte exemplo. Considere-se um grafo completo com 7 nodos e cuja matriz de custos simétrica é dada por:

$$\begin{bmatrix} 0 & 30 & 14 & 20 & 10 & 15 & 25 \\ 30 & 0 & 10 & 11 & 20 & 25 & 3 \\ 14 & 10 & 0 & 5 & 22 & 10 & 7 \\ 20 & 11 & 5 & 0 & 30 & 20 & 10 \\ 10 & 20 & 22 & 30 & 0 & 30 & 15 \\ 15 & 25 & 10 & 20 & 30 & 0 & 10 \\ 25 & 3 & 7 & 10 & 15 & 10 & 0 \end{bmatrix}$$

Pretendemos obter uma árvore com diâmetro não superior a $D = 3$. Começamos por ordenar as arestas por ordem crescente de custo. Obtemos a seguinte lista ordenada $\{2,7\}$, $\{3,4\}$, $\{3,7\}$, $\{1,5\}$, $\{2,3\}$, $\{6,7\}$, $\{4,7\}$, $\{3,6\}$, $\{2,4\}$, $\{1,3\}$, $\{1,6\}$, $\{5,7\}$, $\{1,4\}$, $\{2,5\}$, $\{4,6\}$, $\{3,5\}$, $\{2,6\}$, $\{1,7\}$, $\{4,5\}$, $\{5,6\}$ e $\{1,2\}$. Seleccionando as arestas de menor custo elas são inseridas no grafo pela seguinte ordem $\{2,7\}$, $\{3,4\}$, $\{3,7\}$, $\{1,5\}$. A aresta $\{2,3\}$ não pode ser inserida porque forma ciclo. Inserimos a aresta $\{6,7\}$ e agora as restantes arestas não poderão ser inseridas pois, umas formam ciclo outras não verificam a restrição de diâmetro, por exemplo, ao inserir-se $\{5,7\}$ o diâmetro da árvore ficaria $D = 4$. Após a inserção destas arestas obtém-se o grafo da Figura 3.1. Ainda não foram inseridas 6 arestas na solução (e portanto ainda não temos uma árvore) e já não temos mais arestas na lista ordenada. Uma forma de resolver o problema seria a de saber como unir as diversas componentes conexas formadas.

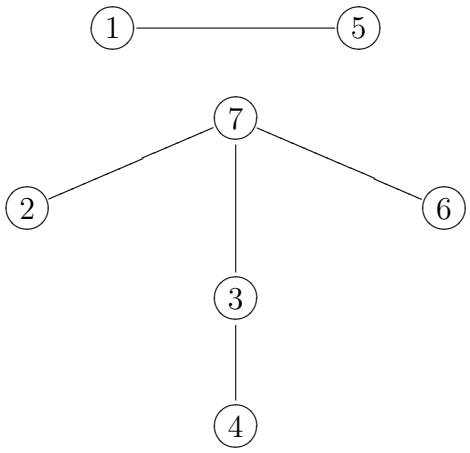


Figura 3.1: Grafo com várias componentes conexas.

Para este exemplo, facilmente se verifica que não é possível ligar as componentes de modo a obter o diâmetro $D = 3$ pretendido.

O que acontece é que ao usarmos esta forma de seleccionar as arestas a incluir na solução obtêm-se frequentemente várias componentes conexas. Para estes casos torna-se bastante complicado, ou até por vezes impossível, fazer a ligação das diversas componentes de forma a obter-se uma árvore que verifique as restrições de diâmetro. Por este motivo, esta primeira ideia fundamentada na forma de seleccionar arestas do algoritmo de Kruskal foi abandonada.

Podemos pois concluir que será necessário construir uma solução por forma a que não se formem componentes conexas. Portanto é mais conveniente uma estratégia de inserção de arestas como a que acontece no algoritmo de Prim.

No algoritmo de Prim [5, 22] começa-se por escolher um nó arbitrário a partir do qual se escolhe a aresta de custo mínimo de entre as arestas que lhe são incidentes. Depois, adiciona-se em cada iteração a aresta de custo mínimo, de entre as arestas incidentes num dos vértices da árvore que vai sendo construída desde que não forme ciclos. Este procedimento repete-se até que a árvore contenha $n - 1$ arestas.

Vamos de seguida descrever três heurísticas. Estas três heurísticas seleccionam as arestas a incluir na árvore como no algoritmo de Prim e distinguem-se apenas na forma como são escolhidos os elementos iniciais (nodo ou aresta). Numa o primeiro elemento é escolhido de modo arbitrário e nas outras duas é escolhido de acordo com uma regra estabelecida que descreveremos.

3.1.1 Heurística Greedy 1

Nesta primeira heurística começa-se por escolher um nodo arbitrário t como incluído na árvore e em cada iteração, tal como no algoritmo de Prim, uma aresta de custo mínimo é acrescentada à árvore que vai sendo construída de tal forma que a árvore não contenha caminhos com comprimento superior a D.

Seja S_1 o conjunto de nodos já incluídos na solução, S_0 o conjunto de nodos ainda isolados e seja S o conjunto das arestas incluídas na árvore. Esta heurística termina quando todos os nodos estiverem inseridos na árvore, ou seja, quando a árvore tiver $n-1$ arestas. A heurística pode ser descrita da seguinte forma.

HEURÍSTICA GREEDY 1

- **Passo 0** - *Inicialização*

  $S_1 = \{t\}$

  $S_0 = V \backslash \{t\}$

  $S = \{\}$

- **Passo 1** - *Selecção de Arestas*

  Seleccionar a aresta $\{k, \ell\}$ de custo mínimo ainda não seleccionada tal que $k \in S_1$ e $\ell \in S_0$.

- **Passo 2** - *Teste de Admissibilidade*

  Se $S \cup \{\{k, \ell\}\}$ forma uma árvore com diâmetro superior a D rejeita-se a aresta $\{k, \ell\}$ e volta-se ao Passo 1.

  Caso contrário, continuar no Passo 3.

- **Passo 3** - *Actualização da Solução e Teste de Paragem*

  $S_1 = S_1 \cup \{\ell\}$

  $S_0 = S_0 \setminus \{\ell\}$

  $S = S \cup \{\{k, \ell\}\}$

  Se $S_0 = \{\}$, STOP.

  Caso contrário, voltar ao Passo 1.

Notemos que no Passo 2 não é necessário um teste para impedir $S \cup \{\{k, \ell\}\}$ de formar um ciclo, pois ao escolhermos o nodo $k \in S_1$ e $\ell \in S_0$ a inclusão da aresta $\{k, \ell\}$ nunca origina ciclo.

Vamos considerar um outro exemplo para exemplificar as várias heurísticas apresentadas neste capítulo. Considere-se um grafo completo com apenas 5 nodos e matriz de custos simétrica dada por:

$$\begin{bmatrix} 0 & 10 & 14 & 20 & 10 \\ 10 & 0 & 10 & 11 & 20 \\ 14 & 10 & 0 & 5 & 22 \\ 20 & 11 & 5 & 0 & 30 \\ 10 & 20 & 22 & 30 & 0 \end{bmatrix}$$

Começamos por usar este exemplo para exemplificar a Heurística Greedy 1 para valores do diâmetro $D = 2$, $D = 3$ e $D = 4$.

Considerando $D = 2$ e iniciando a construção da árvore pelo nodo 4 obtém-se a árvore apresentada na Figura 3.2 (conhecida por grafo estrela) com nodo central 3.

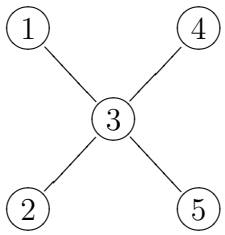


Figura 3.2: Grafo estrela com custo 51 obtido usando a Heurística Greedy 1.

A tabela que se apresenta a seguir mostra a ordem pela qual as arestas foram seleccionadas para obtenção da árvore da Figura 3.2.

Ordem	Aresta	Custo	Resultado do Teste
1	$\{3, 4\}$	5	Aresta Inserida
2	$\{2, 3\}$	10	Aresta Inserida
3	$\{1, 2\}$	10	Excluída, não verifica o diâmetro
4	$\{1, 3\}$	14	Aresta inserida
5	$\{1, 5\}$	10	Excluída, não verifica o diâmetro
6	$\{2, 5\}$	20	Excluída, não verifica o diâmetro
7	$\{3, 5\}$	22	Aresta Inserida

Tabela 3.1: Arestas seleccionadas pela Heurística Greedy 1 para o exemplo indicado, obtendo-se o grafo da Figura 3.2 quando se considera $D = 2$.

Quando se considera $D = 3$ e iniciando a construção da árvore pelo nodo 4 obtém-se a árvore da Figura 3.3 em que $\{2, 3\}$ é a aresta central.

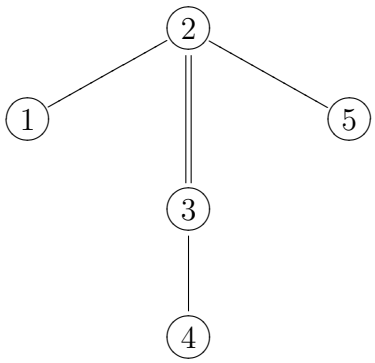


Figura 3.3: Grafo com diâmetro $D = 3$ e custo 45 obtido usando a Heurística Greedy 1.

A tabela que se apresenta a seguir mostra a ordem pela qual as arestas foram seleccionadas para obtenção da árvore da Figura 3.3.

Ordem	Aresta	Custo	Resultado do Teste
1	$\{3, 4\}$	5	Aresta Inserida
2	$\{2, 3\}$	10	Aresta Inserida
3	$\{1, 2\}$	10	Aresta Inserida
4	$\{1, 5\}$	10	Excluída, não verifica o diâmetro
5	$\{2, 5\}$	20	Aresta Inserida

Tabela 3.2: Arestas seleccionadas pela Heurística Greedy 1 para o exemplo indicado, obtendo-se o grafo da Figura 3.3 quando se considera $D = 3$.

Quando se considera $D = 4$ e iniciando a construção da árvore pelo nodo 4 obtém-se a árvore da Figura 3.4 em que o nodo 2 é o nodo central.

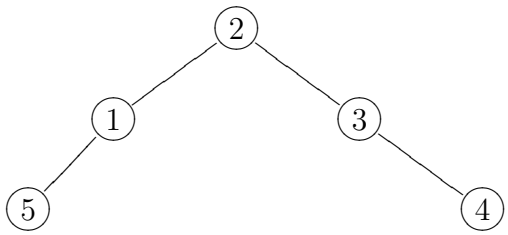


Figura 3.4: Grafo com diâmetro $D = 4$ e custo 35 obtido usando a Heurística Greedy 1.

A tabela que se apresenta a seguir mostra a ordem pela qual as arestas foram seleccionadas para obtenção da árvore da Figura 3.4.

Ordem	Aresta	Custo	Resultado do Teste
1	$\{3, 4\}$	5	Aresta Inserida
2	$\{2, 3\}$	10	Aresta Inserida
3	$\{1, 2\}$	10	Aresta Inserida
4	$\{1, 5\}$	10	Aresta Inserida

Tabela 3.3: Arestas seleccionadas pela Heurística Greedy 1 para o exemplo indicado, obtendo-se o grafo da Figura 3.4 quando se considera $D = 4$.

Note-se que o grafo da Figura 3.4 corresponde à árvore de suporte de custo mínimo o que faz com que o teste de admissibilidade relativamente ao diâmetro se verifique sempre.

Pode acontecer que não sejamos capazes de obter uma solução admíssível, por exemplo se o grafo não for completo.

Considere-se o seguinte grafo não completo:

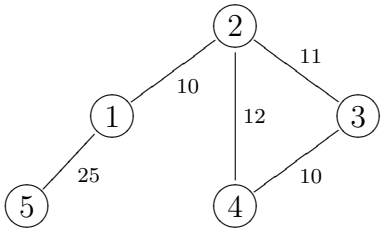
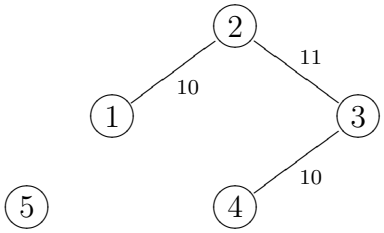


Figura 3.5: Grafo não completo.

Suponhamos que pretendemos obter, uma árvore de suporte de diâmetro 3. Iniciando a construção da árvore pelo nodo 1, começamos por inserir a aresta $\{1, 2\}$, depois a aresta $\{2, 3\}$ e em seguida a aresta $\{3, 4\}$ e obtemos o seguinte grafo:



Falta inserir o nodo 5. Ao adicionar-se a aresta $\{1, 5\}$ que é a única aresta incidente no nodo 5, é violada a restrição de diâmetro. Neste caso não é possível obter uma solução admíssível, usando esta heurística.

3.1.2 Heurística Greedy 2

Nesta heurística a selecção de arestas a incluir na árvore é efectuada como no algoritmo de Prim. Contudo não selecciona os elementos iniciais de forma aleatória, tal como acontecia na heurística anterior. Os autores Kershenbaum e Chou [19] notaram que a razão pela qual as diversas heurísticas seleccionam arestas diferentes, se deve a associarem diferentes níveis de importância aos diferentes nodos do grafo. Na primeira heurística todos os nodos tinham a mesma importância pelo que era seleccionado um nodo aleatoriamente. Nesta heurística associamos a cada nodo i um determinado peso e usamos essa atribuição de pesos aos nodos do grafo para efectuar a escolha dos elementos iniciais. Para além disso se o valor de D for ímpar, o elemento inicial é uma aresta. Portanto, usando os pesos que se atribuem aos nodos escolhe-se um nodo inicial se o valor de D for par ou dois nodos iniciais (e, portanto, uma aresta) se o valor de D for ímpar.

Considera-se o peso para cada nodo i como sendo a soma dos custos das arestas incidentes no nodo i. Para o caso de D ser par escolhe-se o nodo cujo peso é mínimo. Para o caso de D ser ímpar inicia-se o algoritmo seleccionando os dois nodos i e j cujo peso é mínimo. Sendo, neste caso, a aresta $\{i, j\}$ a primeira aresta a ser seleccionada.

Uma vez escolhidos os elementos para iniciar o algoritmo (nodo ou aresta) vão-se acrescentando as restantes arestas da mesma forma que no algoritmo de Prim [5] de modo a que não se formem caminhos com mais do que D arestas. Pára-se quando todos os nodos estiverem incluídos na árvore, ou seja, quando a árvore tiver $n - 1$ arestas. Assim propõe-se o seguinte algoritmo:

HEURÍSTICA GREEDY 2

- **Passo 0** - *Inicialização e Selecção dos Elementos Iniciais (nodo ou aresta)*

  Calcular o peso de cada nodo i (soma dos custos das arestas incidentes em i).

- ***Caso de D ímpar:***

  Seleccionar dois nodos k_1 e k_2 tal que o peso seja o menor possível e incluir a aresta $\{k_1, k_2\}$ na solução.

  $S_1 = \{k_1, k_2\}$

  $S_0 = V \backslash \{k_1, k_2\}$

  $S = \{\{k_1, k_2\}\}$

- ***Caso de D par:***

  Seleccionar o nodo k cujo peso seja o menor.

  $S_1 = \{k\}$

  $S_0 = V \backslash \{k\}$

  $S = \{\}$

- **Passo 1** - *Selecção de Arestas*

  Seleccionar a aresta de custo mínimo $\{k, \ell\}$ ainda não seleccionada tal que $k \in S_1$ e $\ell \in S_0$.

- **Passo 2** - *Teste de Admissibilidade*

  Se $S \cup \{\{k, \ell\}\}$ forma uma árvore com diâmetro superior a D rejeita-se a aresta $\{k, \ell\}$ e volta-se ao Passo 1.

  Caso contrário, continuar no Passo 3.

- **Passo 3** - *Actualização da Solução e Teste de Paragem*

  $S_1 = S_1 \cup \{\ell\}$

  $S_0 = S_0 \setminus \{\ell\}$

  $S = S \cup \{\{k, \ell\}\}$

  Se $S_0 = \{\}$, STOP.

  Caso contrário, voltar ao Passo 1.

Nesta heurística tal como na anterior também não é necessário a inclusão de um teste para impedir a formação de ciclos.

Apresenta-se de seguida o mesmo exemplo usado anteriormente para exemplificar a Heurística Greedy 2 considerando valores do diâmetro $D = 2$, $D = 3$ e $D = 4$. Começa-se por calcular o peso dos nodos e esta informação encontra-se na tabela seguinte.

Nodo	Arestas Incidentes	Custos das Arestas	Pesos
1	$\{1,2\}$, $\{1,3\}$, $\{1,4\}$ e $\{1,5\}$	10, 14, 20 e 10	54
2	$\{1,2\}$, $\{2,3\}$, $\{2,4\}$ e $\{2,5\}$	10, 10, 11 e 20	51
3	$\{1,3\}$, $\{2,3\}$, $\{3,4\}$ e $\{3,5\}$	14, 10, 5 e 22	51
4	$\{1,4\}$, $\{2,4\}$, $\{3,4\}$ e $\{4,5\}$	20, 11, 5 e 30	66
5	$\{1,5\}$, $\{2,5\}$, $\{3,5\}$ e $\{4,5\}$	10, 20, 22 e 30	82

Tabela 3.4: Atribuição dos pesos aos nodos do grafo.

Quando se considera $D = 2$ poder-se-ia escolher o nodo 2 ou 3 como nodo inicial. Selecciona-se o nodo 2 e inicia-se a construção da árvore a partir dele, obtendo-se o grafo da Figura 3.6:

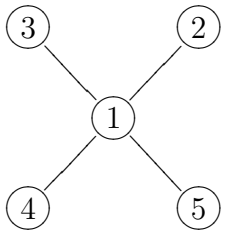


Figura 3.6: Grafo com diâmetro $D = 2$ e custo 54 obtido usando a Heurística Greedy 2.

O nodo 1 é o nodo central. Esta solução é diferente da solução obtida pela Heurística Greedy 1 em que o nodo central é o 3 e o custo é 51, menor que o obtido por esta heurística.

A tabela que se apresenta a seguir mostra a ordem pela qual as arestas foram seleccionadas para obtenção da árvore da Figura 3.6.

Ordem	Aresta	Custo	Resultado do Teste
1	$\{1,2\}$	10	Aresta inserida
2	$\{1,5\}$	10	Aresta inserida
3	$\{2,3\}$	10	Excluída, não verifica o diâmetro
4	$\{2,4\}$	11	Excluída, não verifica o diâmetro
5	$\{1,3\}$	14	Aresta inserida
6	$\{3,4\}$	5	Excluída, não verifica o diâmetro
7	$\{1,4\}$	20	Aresta Inserida

Tabela 3.5: Arestas seleccionadas pela Heurística Greedy 2 para o exemplo indicado, obtendo-se o grafo da Figura 3.6 quando se considera $D = 2$.

Quando se considera $D = 3$, começa-se por escolher os dois nodos cuja soma dos pesos é mínima, são eles o 2 e o 3 e inicia-se a construção da árvore com a aresta $\{2,3\}$, obtendo-se a árvore da Figura 3.7.

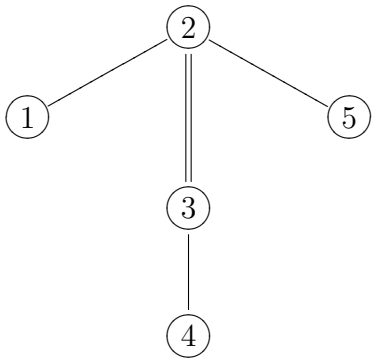


Figura 3.7: Grafo com diâmetro $D = 3$ e custo 45 obtido usando a Heurística Greedy 2.

A aresta $\{2,3\}$ é a aresta central e é obtida a mesma solução encontrada pela Heurística Greedy 1.

A tabela que se apresenta a seguir mostra a ordem pela qual as arestas foram seleccionadas para obtenção da árvore da Figura 3.7.

Ordem	Aresta	Custo	Resultado do Teste
1	$\{2,3\}$	10	Aresta Inserida
2	$\{3,4\}$	5	Aresta Inserida
3	$\{1,2\}$	10	Aresta Inserida
4	$\{1,5\}$	10	Excluída, não verifica o diâmetro
5	$\{2,5\}$	20	Aresta Inserida

Tabela 3.6: Arestas seleccionadas pela Heurística Greedy 2 para o exemplo indicado, obtendo-se o grafo da Figura 3.7 quando se considera $D = 3$.

Quando se considera $D = 4$, poder-se-ia escolher o nodo 2 ou 3. Selecciona-se o nodo 2 e inicia-se a construção da árvore pelo nodo 2 obtendo-se a árvore da Figura 3.8.

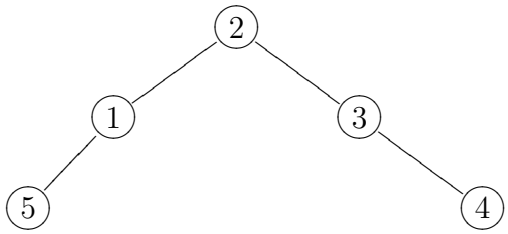


Figura 3.8: Grafo com diâmetro $D = 4$ e custo 35 obtido usando a Heurística Greedy 2.

O nodo 2 é o nodo central e obtém-se a mesma solução obtida pela Heurística Greedy 1 e que corresponde à árvore de suporte de custo mínimo.

A tabela que se apresenta a seguir mostra a ordem pela qual as arestas foram seleccionadas para obtenção da árvore da Figura 3.8.

Ordem	Aresta	Custo	Resultado do Teste
1	$\{1,2\}$	10	Aresta Inserida
2	$\{1,5\}$	10	Aresta Inserida
3	$\{2,3\}$	10	Aresta Inserida
4	$\{3,4\}$	5	Aresta Inserida

Tabela 3.7: Arestas seleccionadas pela Heurística Greedy 2 para o exemplo indicado, obtendo-se o grafo da Figura 3.8 quando se considera $D = 4$.

Notamos que tal como na heurística anterior pode acontecer que não sejamos capazes de obter uma solução admissível no caso do grafo não ser completo.

3.1.3 Heurística Greedy 3

A heurística que se segue também se baseia no algoritmo de Prim na forma de selecção das arestas a inserir na árvore. Mas, ao contrário do que acontecia na heurística anterior é usada uma sugestão de Martin [15], a qual efectua a escolha dos elementos iniciais através do nodo ou nodos que têm um maior peso.

Para efectuar esta escolha calcula-se o peso dos nodos da mesma forma que foi obtido na Heurística Greedy 2. Ao contrário do que acontecia na heurística anterior começamos por escolher o nodo mais pesado, se o valor de D for par, ou os dois nodos i e j mais pesados (e, portanto, a primeira aresta a ser seleccionada é a aresta $\{i,j\}$), se o valor de D for ímpar, para serem incluídos na árvore. Escolhe-se agora um nodo ou dois nodos mais pesados, consoante a paridade do diâmetro.

Uma vez escolhidos os elementos para iniciar o algoritmo vão-se acrescentando sucessivamente as arestas de custo mínimo ao grafo de modo a que não se formem ciclos ou caminhos com mais do que D arestas. Pára-se quando todos os nodos estiverem incluídos na árvore, ou seja, quando a árvore tiver $n-1$ arestas. Assim propõe-se o seguinte algoritmo:

HEURÍSTICA GREEDY 3

- **Passo 0** - *Inicialização e Selecção dos Elementos Iniciais (nodo ou aresta)*

  Calcular o peso de cada nodo i (soma dos custos das arestas incidentes em i).

  - ***Caso de D ímpar:***

    Seleccionar dois nodos k_1 e k_2 tal que o peso seja o maior possível e incluir a aresta $\{k_1, k_2\}$ na solução.

    $S_1 = \{k_1, k_2\}$

    $S_0 = V \setminus \{k_1, k_2\}$

    $S = \{\{k_1, k_2\}\}$

  - ***Caso de D par:***

    Seleccionar o nodo k de peso máximo.

    $S_1 = \{k\}$

    $S_0 = V \setminus \{k\}$

    $S = \{\}$

- **Passo 1** - *Selecção de Arestas*

  Seleccionar a aresta de custo mínimo $\{k, \ell\}$ ainda não seleccionada tal que $k \in S_1$ e $\ell \in S_0$.

- **Passo 2** - *Teste de Admissibilidade*

  Se $S \cup \{\{k, \ell\}\}$ forma uma árvore com diâmetro superior a D rejeita-se a aresta $\{k, \ell\}$ e volta-se ao Passo 1.

  Caso contrário, continuar no Passo 3.

- **Passo 3** - *Actualização da Solução e Teste de Paragem*

  $S_1 = S_1 \cup \{\ell\}$

  $S_0 = S_0 \setminus \{\ell\}$

  $S = S \cup \{\{k, \ell\}\}$

  Se $S_0 = \{\}$, STOP.

  Caso contrário, voltar ao Passo 1.

Nesta heurística tal como nas anteriores não é necessário a inclusão de um teste para impedir a formação de ciclos.

Apresenta-se de seguida o mesmo exemplo usado nos algoritmos anteriores para exemplificar a Heurística Greedy 3, o peso dos nodos do grafo é o mesmo do considerado no algoritmo anterior (ver Tabela 3.4 da página 24).

Quando se considera $D = 2$ começa-se por seleccionar o nodo 5. Inicia-se a construção da árvore por esse nodo e obtém-se o grafo da Figura 3.9.

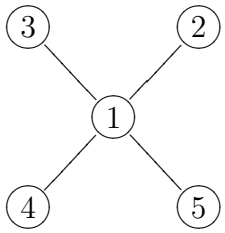


Figura 3.9: Grafo com diâmetro $D = 2$ e custo 54 obtido usando a Heurística Greedy 3.

O nodo 1 é o nodo central e obtém-se a mesma solução obtida usando a Heurística Greedy 2 e uma solução diferente da obtida usando a Heurística Greedy 1 e com um custo maior.

A tabela que se apresenta a seguir mostra a ordem pela qual as arestas foram seleccionadas para obtenção da árvore da Figura 3.9.

Ordem	Aresta	Custo	Resultado do Teste
1	$\{1,5\}$	10	Aresta inserida
2	$\{1,2\}$	10	Aresta Inserida
3	$\{2,3\}$	10	Excluída, não verifica o diâmetro
4	$\{2,4\}$	11	Excluída, não verifica o diâmetro
5	$\{1,3\}$	14	Aresta inserida
6	$\{3,4\}$	5	Excluída, não verifica o diâmetro
7	$\{1,4\}$	20	Aresta inserida

Tabela 3.8: Arestas seleccionadas pela Heurística Greedy 3 para o exemplo indicado, obtendo-se o grafo da Figura 3.9 quando se considera $D = 2$.

Quando se considera $D = 3$ começa-se por inserir os nodos 4 e 5 e inicia-se a construção da árvore com a aresta $\{4,5\}$ obtendo-se a árvore da Figura 3.10.

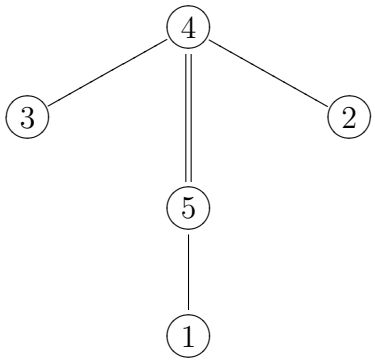


Figura 3.10: Grafo com diâmetro $D = 3$ e custo 56 obtido usando a Heurística Greedy 3.

A aresta $\{4,5\}$ é a aresta central. Esta solução difere das soluções obtidas anteriormente, obtendo-se uma solução com custo maior.

A tabela que se apresenta a seguir mostra a ordem pela qual as arestas foram seleccionadas para obtenção da árvore da Figura 3.10.

Ordem	Aresta	Custo	Resultado do Teste
1	$\{4,5\}$	30	Aresta Inserida
2	$\{3,4\}$	5	Aresta Inserida
3	$\{1,5\}$	10	Aresta Inserida
4	$\{1,2\}$	10	Excluída, não verifica o diâmetro
5	$\{2,3\}$	10	Excluída, não verifica o diâmetro
6	$\{2,4\}$	11	Aresta inserida

Tabela 3.9: Arestas seleccionadas pela Heurística Greedy 3 para o exemplo indicado, obtendo-se o grafo da Figura 3.10 quando se considera $D = 3$.

Quando se considera $D = 4$ começa-se por escolher o nodo 5 e inicia-se a construção da árvore obtendo-se a árvore da Figura 3.11.

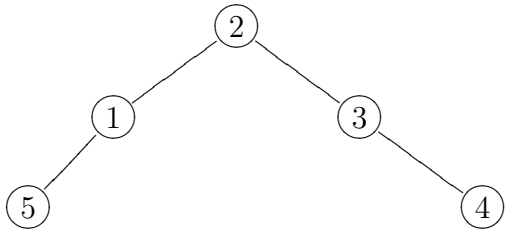


Figura 3.11: Grafo com diâmetro $D = 4$ e custo 35 obtido pela Heurística Greedy 3.

O nodo 2 é o nodo central e a solução obtida nesta heurística é igual às soluções obtidas usando as Heurísticas Greedy 1 e 2, esta solução corresponde à árvore de suporte de custo mínimo.

A tabela que se apresenta a seguir mostra a ordem pela qual as arestas foram seleccionadas para obtenção da árvore da Figura 3.11.

Ordem	Aresta	Custo	Resultado do Teste
2	$\{1,5\}$	10	Aresta Inserida
3	$\{1,2\}$	10	Aresta Inserida
4	$\{2,3\}$	10	Aresta Inserida
5	$\{3,4\}$	5	Aresta Inserida

Tabela 3.10: Arestas seleccionadas pela Heurística Greedy 3 para o exemplo indicado, obtendo-se o grafo da Figura 3.11 quando se considera $D = 4$.

Notamos que tal como nas duas heurísticas anteriores pode acontecer que não sejamos capazes de obter uma solução admissível no caso do grafo não ser completo.

Nesta secção começou-se por apresentar três Heurísticas Construtivas Greedy que diferem na escolha dos elementos iniciais (nodo ou nodos), ou seja diferem na forma de iniciar a construção da árvore. Na Heurística Greedy 1 inicia-se sempre a construção da árvore apenas com um nodo que é escolhido de forma aleatória. Nas Heurísticas Greedy 2 e 3 a construção da árvore inicia-se com um nodo se D for par ou inicia-se com uma aresta se D for ímpar. Usualmente produzem soluções diferentes para o problema DMST que dependem dos elementos que são escolhidos para iniciar a construção da árvore.

O algoritmo de Prim pode ser implementado com um número de operações proporcional a n^2 [9, 21]. Nas Heurísticas Construtivas Greedy para o problema DMST é importante que se observe que a inclusão de um novo teste que impede a árvore de conter caminhos com comprimento superior a D, não aumenta a complexidade do algoritmo original de Prim. Portanto, a complexidade das Heurísticas Construtivas Greedy apresentadas nesta tese é $O(n^2)$.

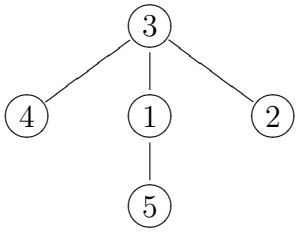
De seguida vamos abordar um outro tipo de heurísticas também muito conhecido que são as heurísticas de trocas locais ou de melhoramento que efectuam algumas trocas de arestas tentando melhorar uma solução admissível conhecida.

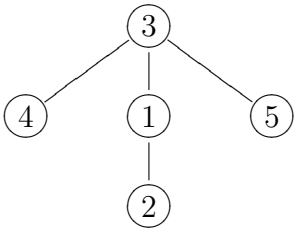
3.2 Heurística de Trocas Locais ou de Melhoramento

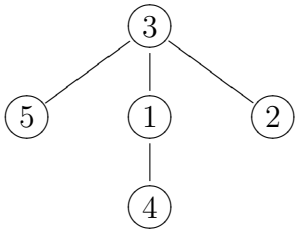
As Heurísticas de Melhoramento são métodos de pesquisa local. Consistem num processo iterativo que efectua pequenas alterações numa solução previamente obtida com o objectivo de melhorar essa solução. Usualmente as transformações que se efectuam são trocas de arestas.

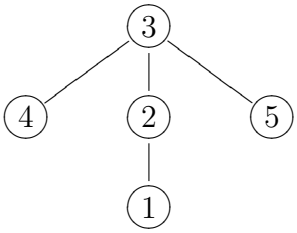
A Heurística de Trocas Locais ou de Melhoramento para o problema DMST que apresentamos nesta secção transforma uma solução admissível noutra solução admissível, portanto transforma uma árvore de suporte de custo mínimo com diâmetro não superior a D numa outra árvore de suporte de custo mínimo com diâmetro não superior a D. As trocas de arestas serão efectuadas de acordo com uma regra estabelecida, mas só se corresponderem a melhoramentos no valor da solução admissível conhecida.

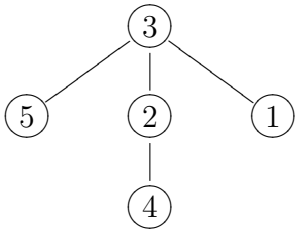
Quando numa solução admissível efectuamos uma troca de arestas obtemos uma solução a que chamamos de ***solução vizinha***. Portanto, a ***vizinhança de uma solução*** é o conjunto de todas as soluções admissíveis que se podem obter a partir dessa solução através da aplicação de uma transformação. A transformação definida pode ser uma troca de arestas e, dependendo do tipo de trocas que se efectuam numa solução, assim podemos definir várias vizinhanças. Por exemplo, se considerarmos a solução admissível que se encontra representada na Figura 3.2 da página 18 e definirmos uma troca de arestas que consiste em trocar uma aresta incidente no nodo central 3 por uma aresta não incidente no nodo central 3. As soluções vizinhas desta solução são as que se encontram representadas na Figura 3.12.

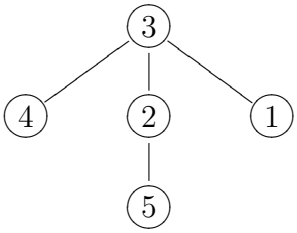

3
4
1
2
5


3
4
1
5
2


3
5
1
2
4


3
4
2
5
1


3
5
2
1
4


3
4
2
1
5

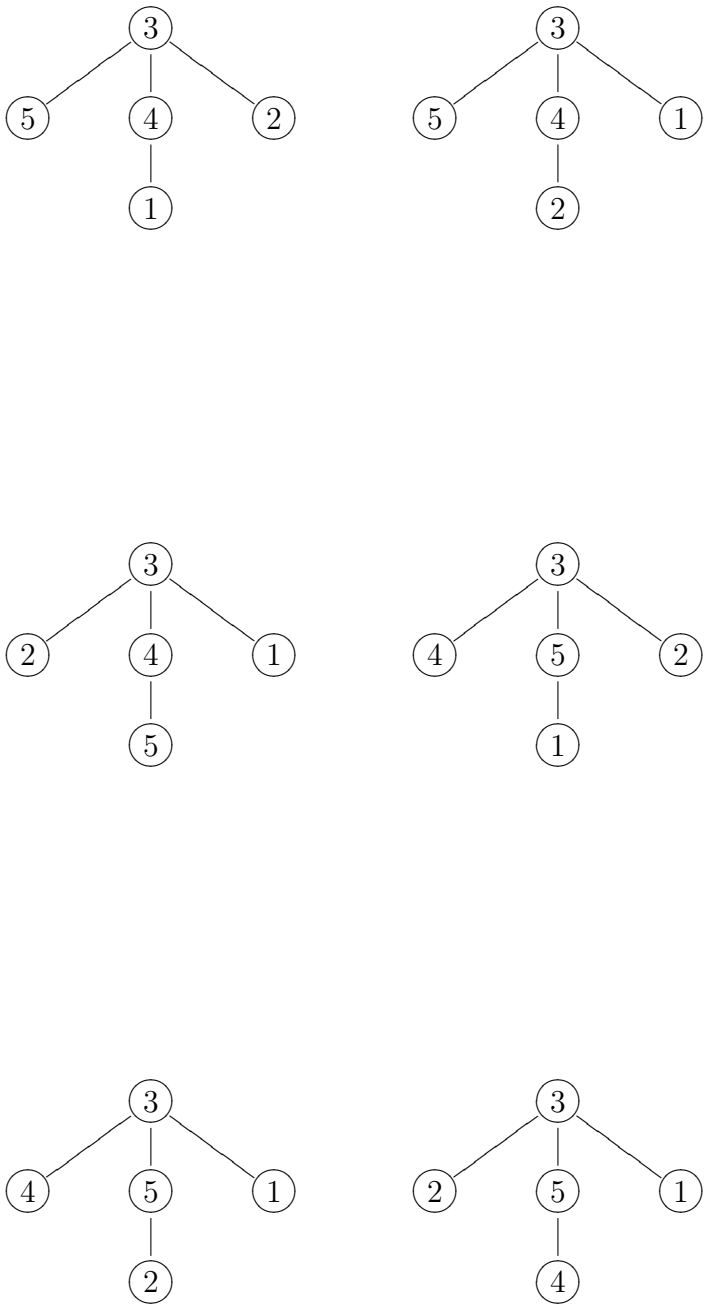


Figura 3.12: Soluções vizinhas da solução inicial da Figura 3.2 da página 18 considerando como transformação uma troca de uma aresta incidente no nodo 3 por uma aresta não incidente no nodo 3.

Uma vez definida a forma de construção da vizinhança e dada uma instância de um problema obtém-se uma solução inicial (usando, por exemplo, uma heurística construtiva). Depois obtém-se uma solução vizinha de custo inferior e sucessivamente obtém-se uma solução vizinha da última solução encontrada até não ser possível obter uma solução vizinha de custo inferior. A este método chama-se método de pesquisa local e evita a obtenção de todas as soluções vizinhas. A escolha de uma boa vizinhança é importante para o processo ser efectivo. A qualidade da solução óptima local depende da solução escolhida inicialmente e da estrutura da vizinhança. Uma vizinhança de grandes dimensões torna o processo de convergência para um óptimo local lento, enquanto que com uma vizinhança de pequenas dimensões corre-se o risco de cair demasiado rápido num "mau" óptimo local.

3.2.1 Heurística de Melhoramento

A Heurística de Melhoramento que consideramos efectua trocas de arestas e a solução vizinha é obtida fazendo uma ou várias trocas de arestas incidente no nodo central t por arestas não incidentes em t. Cada iteração desta heurística consiste portanto, em fazer uma troca entre um par de arestas de modo a transformar uma solução admissível noutra solução admissível. Usa-se uma solução óptima inicial com diâmetro $D = 2$ (grafo estrela) e identifica-se nessa solução o nodo central (t) e as trocas de arestas que é possível efectuar. Em cada iteração remove-se uma aresta incidente no nodo central t e introduz-se uma nova aresta não incidente no nodo central t. Repete-se este processo até se obter uma árvore com diâmetro não superior a D e de modo a que a transformação efectuada reduza o custo da árvore. Notamos que as trocas de arestas que se consideram em cada iteração são determinadas logo de início e portanto apenas se consideram soluções vizinhas da solução inicial.

A forma como estas trocas são efectuadas são semelhantes às que são efectuadas no algoritmo de Esau-Williams. O algoritmo de Esau-Williams [7] é um algoritmo greedy e é muito conhecido na literatura porque obtém soluções admissíveis e de boa qualidade para problemas relacionados com a formação de uma árvore de capacidade mínima (Capacitated Minimum Spanning Tree, CMST) [20]. Este algoritmo procura fazer em cada iteração uma transformação na árvore que consiste em remover uma aresta incidente no nodo central e introduzir uma aresta não incidente no nodo central de modo a manter-se a topologia em

forma de árvore. A troca de arestas a efectuar em cada iteração é aquela que produz uma maior redução no custo da solução. Este algoritmo termina quando não for possível fazer qualquer troca que permita uma redução no custo global da rede.

Para determinarmos as trocas de arestas a efectuar vamos, tal como no algoritmo de Esau-Williams, determinar a melhor troca a efectuar. Para isso, consideram-se dois nodos k e ℓ da solução óptima inicial e calculamos a poupança $p_{k\ell} = max\{c_{kt} - c_{k\ell}, c_{\ell t} - c_{k\ell}\}$ que se obtém se trocarmos a aresta $\{k,t\}$ ou $\{\ell,t\}$, aquela cujo custo é maior, pela aresta $\{k,\ell\}$. A transformação que se efectua corresponde a trocar uma aresta incidente no nodo central por outra não incidente no nodo central. Quando o valor de $p_{k\ell}$ é positivo significa que se consegue melhorar o custo da rede através da inserção da aresta $\{k,\ell\}$ e remoção da aresta que faz a ligação dos nodos k ou ℓ com o nodo central t. Este algoritmo termina quando a árvore tiver o diâmetro pretendido ou quando não for possível fazer qualquer troca que permita uma redução no custo global da rede.

HEURÍSTICA DE MELHORAMENTO

- **Passo 0** - *Inicialização*

  Obter uma solução óptima inicial com diâmetro igual a 2, seja S o conjunto de arestas dessa solução e t o nodo central.

  Obter os valores $p_{k\ell} = max\{c_{kt} - c_{k\ell}, c_{\ell t} - c_{k\ell}\}$ para todos os pares de nodos k e ℓ.

  Seja E a lista dos $p_{k\ell}$ positivos ordenados por ordem decrescente.

- **Passo 1** - *Selecção da Aresta a Inserir*

  Seleccionar a aresta $\{k,\ell\}$ que maximiza $p_{k\ell}$, ou seja, determinar a primeira aresta da lista E. A aresta $\{k,\ell\}$ é candidata a ser inserida.

- **Passo 2** - *Selecção da Aresta a Remover*

  Determinar $c_{it} = max\{c_{kt}, c_{\ell t}\}$ para identificar o nodo i^* que faz a ligação com o nó central. A aresta $\{i^*,t\}$ é candidata a ser removida.

- **Passo 3** - *Actualização da Solução e Teste de Paragem*

  $S = S \cup \{\{k, \ell\}\} \setminus \{\{i*, t\}\}$

  Se a árvore tiver diâmetro igual a D ou se a lista E estiver vazia, STOP.

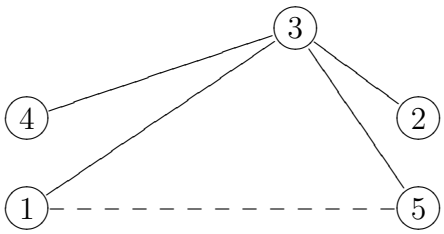
  Caso a árvore tenha diâmetro inferior a D voltar ao Passo 1.

Continuamos a usar o exemplo considerado nas heurísticas anteriores e usamos a solução óptima inicial que se encontra representada na Figura 3.2 da página 18 com um custo de 51 e na qual o nodo 3 é o nodo central ($t = 3$). Para obtermos as poupanças das soluções vizinhas desta solução determinamos os valores $p_{k\ell}$ que se apresentam na seguinte tabela.

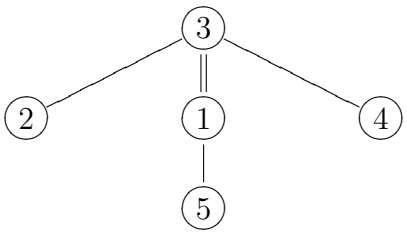
Arestas $\{k, \ell\}$	c_{kt}	$c_{\ell t}$	$c_{k\ell}$	$c_{kt} - c_{k\ell}$	$c_{\ell t} - c_{k\ell}$	$p_{k\ell}$
$\{1, 2\}$	14	10	10	4	0	4
$\{1, 4\}$	14	5	20	-6	-15	-6
$\{1, 5\}$	14	22	10	4	12	12
$\{2, 4\}$	10	5	11	-1	-6	-1
$\{2, 5\}$	10	22	20	-10	2	2
$\{4, 5\}$	5	22	30	-25	-8	-8

Tabela 3.11: Determinação dos $p_{k\ell}$ na Heurística de Melhoramento.

Existem apenas três valores de $p_{k\ell}$ positivos pelo que se podem efectuar no máximo três trocas que provoquem uma diminuição do valor da solução inicial. A lista E é constituída pelas arestas $\{1, 5\}$, $\{1, 2\}$ e $\{2, 5\}$ por esta ordem, isto é, já ordenadas por ordem decrescente. Pretendemos restringir o diâmetro a $D = 3$. A primeira candidata a ser inserida é a aresta $\{1, 5\}$ e as candidatas a serem removidas são a aresta $\{1, 3\}$ ou a aresta $\{3, 5\}$. Como $c_{13} = 14$ e $c_{35} = 22$ remove-se a aresta $\{3, 5\}$ pois tem maior custo e obtém-se uma poupança de 12.



$custo = 51$



$custo = 39$

Uma vez que já foi obtido o diâmetro pretendido o algoritmo termina e a solução obtida é a que se encontra representada no seguinte grafo.

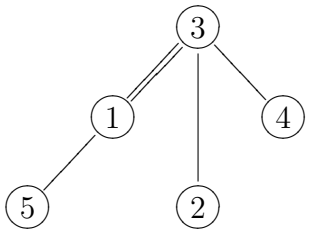
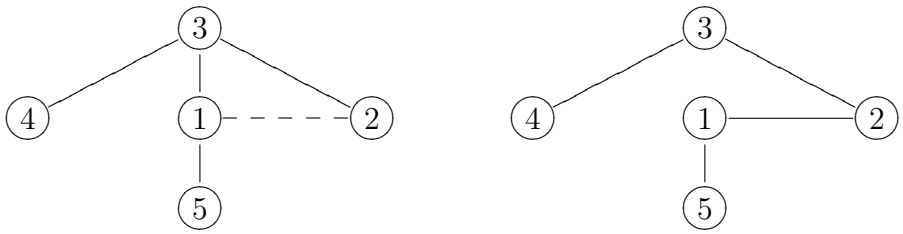


Figura 3.13: Grafo com diâmetro $D = 3$ e custo 39 obtido pela Heurística de Melhoramento.

É obtida uma solução com custo inferior aos custos das soluções obtidas usando as heurísticas anteriores e a aresta $\{1, 3\}$ é a aresta central.

Se pretendermos restringir o valor do diâmetro a $D = 4$, a primeira aresta a ser inserida e a primeira aresta a ser removida são as mesmas que para o caso $D = 3$. A segunda aresta a ser escolhida para inserir na árvore seria $\{1, 2\}$ e as candidatas a serem removidas são a aresta $\{1, 3\}$ ou a aresta $\{2, 3\}$. Como $c_{13} = 14$ e $c_{23} = 10$ remove-se a aresta $\{1, 3\}$ pois tem maior custo e obtém-se uma poupança de 4.



$custo = 39$ $custo = 35$

Desta forma obtém-se o seguinte grafo:

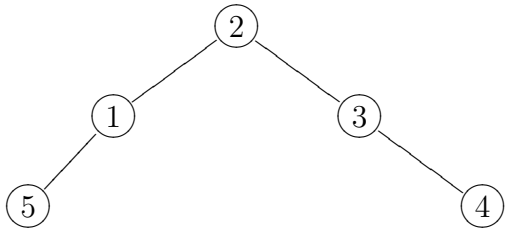


Figura 3.14: Grafo com diâmetro $D = 4$ e custo 35 obtido usando a Heurística de Melhoramento.

Este grafo é a solução correspondente à árvore de suporte de custo mínimo que também foi obtida para este exemplo por todas as heurísticas consideradas até agora.

Note-se que em cada iteração do algoritmo as transformações que levam às maiores reduções no custo da rede são usualmente aquelas que estão associadas às arestas de maior custo incidentes no nodo central. Este facto significa que, usualmente, as ligações envolvendo nodos afastados do nodo central são seleccionadas em primeiro lugar.

A obtenção da solução óptima inicial e a determinação dos p_{kl} requer um total de n^2 operações. A ordenação das arestas da lista E requer um tempo de execução $O(nlogn)$. O processo de trocar uma aresta incidente no nodo central por uma aresta não incidente no nodo central é executado em n^2 operações. Portanto, pode-se concluir que no pior dos casos, a complexidade da Heurística de Melhoramento é $O(n^2)$.

3.3 Heurísticas de Aproximação

Nesta secção apresentam-se heurísticas de aproximação que adaptam soluções de outro problema ao problema em questão.

Começamos por apresentar uma heurística que a partir de uma solução inicial que corresponde à árvore de suporte de custo mínimo, elimina arestas de acordo com uma regra estabelecida e depois a partir da subárvore obtida constrói a nova solução, a árvore de suporte de diâmetro restringido a um valor D.

Apresentamos depois três heurísticas de segunda ordem. Neste tipo de algoritmos repetem-se várias execuções do mesmo algoritmo de modo a que um determinado subconjunto de arestas esteja obrigatoriamente na solução e/ou outro subconjunto de arestas seja inibido de pertencer à solução. Cada execução do ciclo principal desse algoritmo, corresponde a várias execuções de um algoritmo designado por algoritmo de primeira ordem [17] no qual é efectuada a construção de uma solução.

Karnaugh [17] propõe dois tipos de procedimentos neste tipo de algoritmos. O procedimento de *inibição* que impede a inserção de arestas na solução e o procedimento de *junção* que consiste em forçar a inclusão de arestas na solução. Usualmente nos problemas referidos por Karnaugh com restrições de capacidade as modificações produzidas pelo procedimento de inibição permitem obter melhores soluções que as modificações impostas pelo procedimento de junção. As heurísticas baseadas nestes dois procedimentos são consideradas heurísticas de segunda ordem.

Vamos tentar adequar ao problema DMST os referidos procedimentos utilizados por Karnaugh. Para tal na primeira heurística vamos usar o procedimento de inibição, na segunda heurística o procedimento de junção e na terceira heurística os dois procedimentos em conjunto, ou seja junção e inibição de arestas.

3.3.1 Heurística Simples de Aproximação

Na heurística que agora descrevemos, vamos efectuar trocas na árvore de suporte de custo mínimo de modo a transformar a árvore de suporte de custo mínimo numa árvore de suporte com diâmetro restringido a um determinado valor D.

Gouveia [10] faz a descrição de uma heurística que transforma uma árvore de suporte de custo mínimo numa outra árvore de suporte com restrições de salto para obter limites superiores para o valor óptimo do problema da Árvore de Suporte de Custo Mínimo com Restrições de Salto. Com base nesta heurística vamos efectuar a sua adaptação ao problema DMST.

Obtida a árvore de suporte de custo mínimo pelo algoritmo de Kruskal ou Prim forma-se uma lista E constituída pelas arestas da árvore de suporte de custo mínimo e ordenada por ordem decrescente de custos. Depois, passamos à eliminação das arestas da àrvore de suporte de custo mínimo. Esta eliminação é feita percorrendo todas as arestas da lista E. Considera-se a primeira aresta da lista E. Se nessa aresta existir um vértice que tenha grau 1, então essa aresta é eliminada da lista E e da solução da árvore. Se essa aresta não tiver grau 1, então elimina-se apenas da lista E e passa-se à aresta seguinte. Quando o valor do diâmetro da subárvore S for igual ou inferior ao valor D pretendido paramos a eliminação de arestas e passamos à inclusão de arestas que unam os vértices que entretanto ficaram isolados. As arestas a incluir são seleccionadas de entre as possíveis ligações admissíveis. Considera-se uma ligação admissível como sendo aquela que não forma um caminho entre quaisquer dois nós com diâmetro superior a D. Para cada nodo isolado escolhe-se a aresta de custo mínimo que faz a ligação entre a subárvore S e esse nodo. Procede-se iterativamente de tal forma que todos os nodos isolados fiquem ligados e de modo que a árvore fique com diâmetro não superior a D. Se o valor do diâmetro da subárvore S for superior ao valor D e a lista E ficar vazia temos de voltar a formar a lista E com as arestas da solução da subárvore S e proceder da mesma forma até que o diâmetro da subárvore S seja inferior ou igual ao valor D pretendido e finalmente acrescentar as arestas à subárvore S.

HEURÍSTICA SIMPLES DE APROXIMAÇÃO

- **Passo 0** - *Inicialização*
  - Obter a árvore de suporte de custo mínimo pelo algoritmo de Kruskal ou Prim, seja S essa solução.
  - Seja E a lista das arestas de S ordenada por ordem decrescente de custos.
  - Seja D_{as} o diâmetro da árvore de suporte de custo mínimo.
  - Se $D_{as} \leq D$, STOP.
  - Se $D_{as} > D$, continuar no Passo 1.
- **Passo 1** - *Eliminação de Arestas*

  Percorrer ordenadamente todas as arestas da lista E; seja $\{k, \ell\}$ a primeira aresta da lista.
  - $E = E \backslash \{k, \ell\}$
  - Se um dos vértices da aresta $\{k, \ell\}$ tem grau 1 eliminar a aresta da solução: $S = S \backslash \{k, \ell\}$
  - Se a subárvore S obtida tiver diâmetro $d = D$, continuar no Passo 2.
  - Se $d > D$ e a lista E estiver vazia, continuar no Passo 3.
- **Passo 2** - *Acrescentar Novas Arestas*

  Acrescentar à subárvore S novas arestas:
  - Ordenar as arestas que não estão na subárvore S por ordem crescente de custos;
  - Inserir as arestas na subárvore S de tal forma que não se forme uma árvore com diâmetro superior a D.
- **Passo 3** - *Actualização da Solução e Teste de Paragem*

  Actualizar a lista E que vai ser formada pelas arestas da solução da subárvore S que tem diâmetro $d > D$ e voltar ao Passo 1.

Usamos novamente o exemplo considerado para as heurísticas anteriores. A árvore de suporte de custo mínimo tem diâmetro $D = 4$, e um custo de 35 e apresenta-se na Figura 3.15.

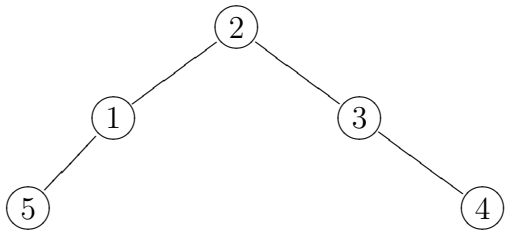


Figura 3.15: Grafo com diâmetro $D = 4$ que corresponde à árvore de suporte de custo mínimo com custo 35.

Vamos supor que pretendemos uma árvore com um diâmetro $D = 2$. Como a árvore de suporte tem diâmetro 4 teremos de proceder à eliminação das arestas de maior custo e tal que um dos nodos tenha grau 1. Apresenta-se na Tabela 3.12 a ordem de eliminação das arestas da árvore de suporte de custo mínimo, ou seja, as arestas que constituem a lista E.

Ordem	Aresta	Custo	Resultados do Teste
1	$\{1, 2\}$	10	Aresta não eliminada pois $grau(1) = grau(2) = 2$
2	$\{1, 5\}$	10	Aresta eliminada, subárvore fica com $d = 3 > D$
3	$\{2, 3\}$	10	Aresta não eliminada pois $grau(2) = grau(3) = 2$
5	$\{3, 4\}$	5	Aresta eliminada, subárvore fica com $d = 2 = D$

Tabela 3.12: Ordenação das arestas da árvore de suporte de custo mínimo por custo e teste de eliminação das arestas seleccionadas pela Heurística Simples de Aproximação para o exemplo da Figura 3.15.

Após a eliminação das arestas $\{1,5\}$ e $\{3,4\}$ obtém-se o grafo da Figura 3.16 cuja subárvore tem diâmetro $d = 2$ e com os nodos 4 e 5 isolados.

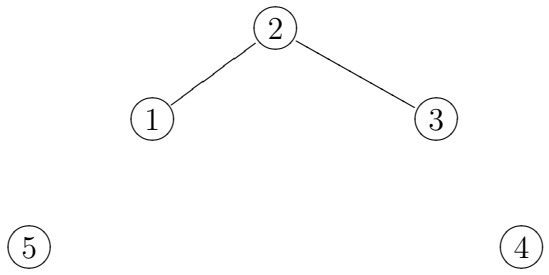


Figura 3.16: Subárvore com diâmetro $d = 2$ obtida por eliminação de arestas da árvore de suporte de custo mínimo da Figura 3.15 da página 44.

Procede-se de seguida à adição das arestas para a construção da nova árvore abrangente. Vão-se adicionando sucessivamente as arestas de custo mínimo, que verifiquem o diâmetro $D = 2$ pretendido, para unir os nodos 4 e 5 à subárvore conforme nos mostra a Tabela 3.13, onde estão ordenadas as arestas que não estão na árvore.

Ordem	Aresta	Custo	Resultados do Teste
1	$\{3,4\}$	5	Aresta não inserida, não verifica o diâmetro
2	$\{1,5\}$	10	Aresta não inserida, não verifica o diâmetro
3	$\{2,4\}$	11	Aresta inserida
4	$\{2,5\}$	20	Aresta inserida
5	$\{3,5\}$	22	Aresta não testada

Tabela 3.13: Arestas a inserir na subárvore da Figura 3.16.

Note-se que ao ser inserida a aresta $\{2,5\}$ a árvore abrangente fica determinada pelo que já não testamos a aresta $\{3,5\}$. Após a inserção das arestas na subárvore da Figura 3.16, obtém-se a árvore da Figura 3.17.

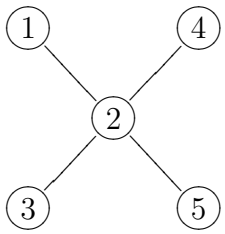


Figura 3.17: Grafo com diâmetro $D = 2$ e custo 51 obtido pela Heurística Simples de Aproximação.

O nodo central é o nodo 2 e obtém-se uma solução diferente das obtidas pelas heurísticas anteriores mas com o mesmo custo da solução obtida pela Heurística Greedy 1.

Para um diâmetro $D = 3$ a ordem pela qual se consideram as arestas é a mesma. Quando se elimina a aresta $\{1, 5\}$ obtém-se uma subárvore com diâmetro $d = 3 = D$, ao contrário do exemplo para $D = 2$ já não é necessário eliminar a aresta $\{3, 4\}$, pois já se obteve um valor para o diâmetro $d = 3 = D$ e obtém-se a subárvore da Figura 3.18 com o nodo 5 isolado.

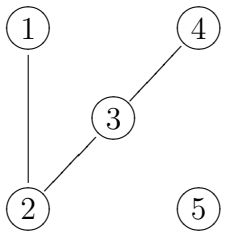


Figura 3.18: Subárvore com diâmetro $D = 3$ obtida por eliminação das arestas da árvore de suporte de custo mínimo da Figura 3.15 da página 44.

De seguida forma-se a lista das arestas que não estão na árvore, ordenada por ordem crescente de custo. A aresta $\{1, 5\}$ é a primeira candidata a ser inserida na árvore mas, uma vez que não verifica o diâmetro não vai ser inserida, seguidamente insere-se a aresta $\{2, 5\}$ e a árvore fica determinada. Portanto, já não testamos a aresta $\{3, 5\}$. Obtém-se a árvore da Figura 3.19.

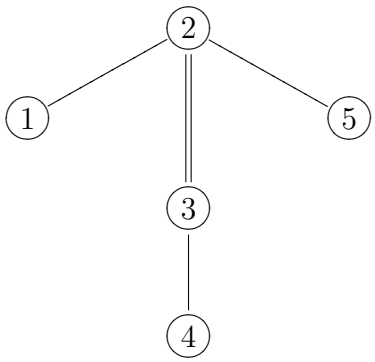


Figura 3.19: Grafo com diâmetro $D = 3$ e custo 45 obtido pela Heurística Simples de Aproximação.

A aresta central é $\{2, 3\}$ e obtém-se a mesma solução obtida anteriormente pelas Heurísticas Greedy 1 e 2.

Para um valor de $D = 4$ a solução correspondente é a solução obtida na árvore de suporte de custo mínimo que é igual à solução obtida para um valor de $D = 4$ pelas heurísticas anteriores (Figura 3.15 da página 44).

A Heurística Simples de Aproximação numa primeira fase elimina as arestas da árvore de suporte até se obter uma subárvore de acordo com regras estabelecidas e numa segunda fase constrói a nova árvore a partir da subárvore obtida de modo a obter-se uma árvore com o diâmetro pretendido.

A obtenção da árvore de suporte de custo mínimo obtida pelo algoritmo de Prim é implementada em tempo $O(n^2)$. Após a construção da árvore de suporte de custo mínimo procede-se à eliminação de arestas, o que requer um tempo de execução $O(nlogn)$, pois antes de se proceder à eliminação de arestas estas terão de ser ordenadas por ordem decrescente. Depois da eliminação obtemos uma subárvore e necessitamos de lhe adicionar arestas para encontrar a solução final. No pior dos casos serão inseridas $n - 1$ arestas, então o número de operações é de $O(n^2)$. Desta forma, a Heurística Simples de Aproximação pode ser implementada, no pior dos casos, em $O(n^2)$.

3.3.2 Heurística de Inibição

Nesta heurística consideramos a proibição (*inibição de presença*) de algumas arestas na solução. Esta proibição de arestas poderá trazer melhoramentos para a solução.

Considere-se uma solução admissível inicial, que será considerada a solução de referência. Seja S_d o conjunto de arestas da solução de referência, ou seja, o conjunto de arestas da solução admissível inicial e seja z_d o valor da solução de referência, isto é, o valor da solução admissível inicial.

Obtém-se a árvore de suporte de custo mínimo e nela determina-se o nodo t com maior grau, se houver vários escolhe-se o nodo cuja soma dos pesos das arestas incidentes seja o menor possível.

Do conjunto de arestas S_d forma-se uma sequência de arestas não ligadas ao nodo t. A sequência das arestas na solução de referência S_d não ligadas ao nodo t será designada por A_d, seja k o número de arestas na sequência A_d e cada aresta dessa sequência será designada por a_i, $i = 1, \ldots, k$. Para cada uma das arestas a_i da sequência A_d determina-se uma solução admissível que não possua a aresta a_i, $i = 1, \ldots, k$ (para obtenção destas soluções podemos usar qualquer heurística anterior, como por exemplo a Heurística Greedy 2). O conjunto das arestas dessa solução é designado por S_{a_i} e o seu valor designado por z_{a_i}, $i = 1, \ldots, k$. De entre todas as soluções obtidas escolhe-se a solução de menor custo. Designe-se por S_c o seu conjunto de arestas e por z_c o seu valor. Caso z_c seja menor do que o valor da solução de referência alteramos a solução de referência que passa a ser a solução S_c, se tiver valor maior a solução de referência continua a ser S_d.

HEURÍSTICA DE INIBIÇÃO

- **Passo 0** - *Inicialização*
  - Determinar o vértice t de maior grau na árvore de suporte de custo mínimo (caso existam vários escolher o vértice cuja soma dos pesos das arestas incidentes seja o menor possível).
  - Determinar uma solução admissível inicial.
  - Obter S_d, conjunto de arestas desta solução.
  - Determinar a sequência $A_d = [a_1, a_2, \ldots, a_k]$, de arestas não ligadas ao nodo t, sendo k o seu número.

- **Passo 1** - *Obtenção das soluções com inibição de uma aresta*

  Para cada aresta a_i da sequência A_d determinar uma solução admissível inibindo a presença da aresta a_i na solução. Designar por z_{a_i} o valor dessa solução (esta solução admissível é obtida usando a Heurística Greedy 2).

- **Passo 2** - *Escolha da solução de referência*

  Escolher $z_c = min(z_{a_1}, \ldots, z_{a_k})$ e seja S_c o correspondente conjunto de arestas.

  Caso $z_c < z_d$ a nova solução de referência é S_c.

  Caso contrário, a solução e o valor final obtido serão S_d e z_d respectivamente.

Consideramos o mesmo exemplo usado nas heurísticas anteriores para exemplificar também esta heurística.

Consideramos $D = 3$. Começamos por obter o nodo t que corresponde ao nodo de maior grau da árvore de suporte de custo mínimo cujo grafo se encontra representado na Figura 3.15 da página 44. Como há três nodos, o 1, o 2 e o 3 nestas condições, escolhe-se $t = 3$ porque é o nodo cuja soma das arestas incidentes é menor (20, 20 e 15 são os pesos dos nodos 1, 2 e 3, respectivamente). Consideramos uma solução admissível inicial com $D = 3$ (solução de referência) que pode ser observada na Figura 3.13 da página 39 e que foi obtida usando a Heurística de Melhoramento. O conjunto das arestas de referência é $S_d = \{\{1,3\},\{1,5\},\{2,3\},\{3,4\}\}$ e o valor da solução de referência é $z_d = 39$. A sequência de arestas não ligadas ao nodo 3 é $A_d = \{\{1,5\}\}$ e a solução admissível obtida sem a aresta $\{1,5\}$ está representada no grafo da Figura 3.20.

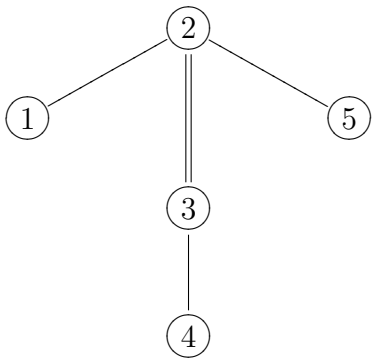


Figura 3.20: Grafo com diâmetro $D = 3$ e custo $z_{a_1} = 45$ obtido durante o processo de resolução da Heurística de Inibição proibindo a aresta $\{1, 5\}$ de estar na solução.

Como $z_c = 45 > z_d = 39$, não há alteração da solução de referência logo, $z_d = 39$ e a solução obtida pela Heurística de Inibição encontra-se representada através do grafo da Figura 3.13 da página 39.

Nesta Heurística obtém-se a mesma solução que a obtida pela Heurística de Melhoramento e com custo inferior à solução obtida pelas restantes heurísticas.

Consideremos $D = 4$. Tal como já vimos para o caso do diâmetro $D = 3$ temos $t = 3$ e consideremos a solução admissível inicial representada através do grafo da Figura 3.14 da página 40 com diâmetro $D = 4$ obtida usando a Heurística de Melhoramento. Seja $S_d = \{\{1, 2\}, \{1, 5\}, \{2, 3\}, \{3, 4\}\}$ o conjunto de arestas de referência e $z_d = 39$ o valor de referência. A sequência de arestas não ligadas ao nodo 3 é $A_d = \{\{1, 2\}, \{1, 5\}\}$. Obtemos duas soluções admissíveis, uma inibindo a aresta $a_1 = \{1, 2\}$ e a outra inibindo a aresta $a_2 = \{1, 5\}$ com custos respectivamente de $z_{a_1} = 39$ e $z_{a_2} = 45$, que se encontram representadas nos grafos da Figura 3.21.

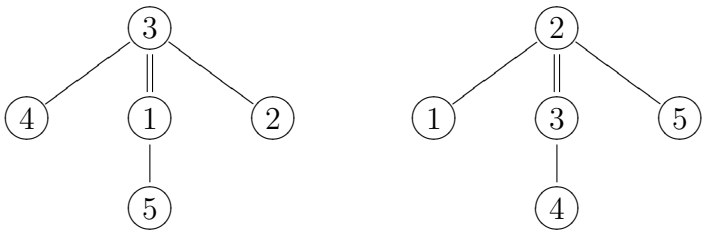


Figura 3.21: Soluções admissíveis com custos $z_{a_1} = 39$ e $z_{a_2} = 45$, respectivamente, e diâmetro $D = 3$ obtidas durante o processo de resolução da Heurística de Inibição proibindo a aresta $a_1 = \{1, 2\}$ e a aresta $a_2 = \{1, 5\}$ respectivamente de estarem na solução.

Temos $z_c = 39 \leq z_d = 39$, logo não há alteração da solução de referência. Desta forma é obtida uma solução igual à obtida pela Heurística de Melhoramento e com custo inferior ao custo obtido pelas restantes heurísticas.

3.3.3 Heurística de Junção

Nesta heurística consideramos transformações na árvore inicial obrigando a presença (*junção*) de arestas na solução. Esta junção de arestas à solução poderá trazer melhoramentos para a solução.

Tal como na heurística anterior também se obtém uma solução admissível inicial que será designada por solução de referência S_d. Obtém-se o nodo t, sendo a sua escolha feita da mesma forma que na heurística anterior. Forma-se uma sequência de arestas, que designamos por E_d, e essa sequência é formada pelas arestas incidentes no nodo t na solução. Seja k o número de elementos de E_d. As arestas desta sequência são designadas por a_i, $i = 1, \ldots, k$. Para cada uma das arestas a_i de E_d determina-se uma solução admissível na qual se obriga a aresta a_i a estar na solução. Esta solução é obtida usando a Heurística Greedy 2, apenas com a única diferença de a escolha da aresta inicial ser a aresta a_i que se obriga a estar na solução. Desta forma obtém-se para cada solução o valor z_{a_i}, $i = 1, \ldots, k$. Escolhe-se para solução o menor valor dos z_{a_i}, $(i = 1, \ldots, k)$ seja ele z_c. Se esta solução for menor que z_d (valor da solução admissível inicial) escolhe-se z_d, caso contrário escolhe-se z_c.

HEURÍSTICA DE JUNÇÃO

- **Passo 0** - *Inicialização*
  - Determinar o vértice t de maior grau da árvore de suporte de custo mínimo (caso existam vários escolher o vértice cuja soma dos pesos das arestas incidentes seja o menor possível).
  - Determinar uma solução admissível inicial.
  - Obter S_d, conjunto de arestas desta solução.
  - Determinar a sequência $E_d = [a_1, a_2, \ldots, a_k]$, de arestas ligadas ao nodo t, sendo k o seu número.
- **Passo 1** - *Obtenção das soluções com junção de uma aresta*

  Para cada aresta a_i da sequência E_d determinar uma solução admissível exigindo que a aresta a_i esteja na solução. Designemos por z_{a_i} o valor dessa solução.
- **Passo 2** - *Escolha da solução de referência*

  Escolher $z_c = min(z_{a_1}, \ldots, z_{a_k})$ e seja S_c o correspondente conjunto de arestas.

  Caso $z_c < z_d$ a nova solução de referência é S_c.

  Caso contrário, a solução e o valor final obtido serão S_d e z_d respectivamente.

Apresenta-se de seguida o mesmo exemplo usado nas heurísticas anteriores para exemplificar a Heurística de Junção.

Consideremos $D = 3$. Começa-se por obter o nodo t, sendo este $t = 3$ (a árvore de suporte de custo mínimo encontra-se na Figura 3.14 da página 40). Escolhendo uma solução admissível inicial (por exemplo a representada na Figura 3.10 da página 30), esta solução é constituída pelas seguintes arestas $S_d = \{\{1,5\},\{2,4\},\{3,4\},\{4,5\}\}$ e com $z_d = 56$. A sequência de arestas ligadas ao nodo 3 desta solução é $E_d = [\{3,4\}]$. Exige-se agora que a aresta $\{3,4\}$ esteja presente na solução e obtém-se o grafo da Figura 3.22.

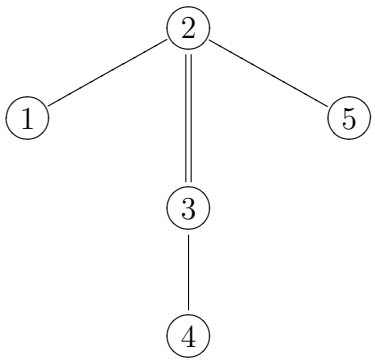


Figura 3.22: Solução admissível com custo $z_{a_1} = 45$ e diâmetro $D = 3$ obtida durante o processo de resolução da Heurística de Inibição, obrigando a aresta $\{3, 4\}$ a estar na solução.

Como $z_c = 45 < z_d = 56$ escolhe-se $z_c = 45$. A solução obtida por esta heurística é a representada na Figura 3.22.

É obtida a mesma solução que a obtida pelas Heurísticas Greedy 1, 2 e Heurística Simples de Aproximação e com custo superior ao obtido pelas Heurísticas de Melhoramento e de Inibição.

Para $D = 4$ obtém-se a solução correspondente à árvore de suporte de custo mínimo representada na Figura 3.15 da página 44.

3.3.4 Heurística de Inibição e Junção

Nesta heurística consideram-se transformações na árvore inicial por proibição (*inibição*) e obrigação (*junção*) de presença de arestas na solução. Esta inibição/junção poderá trazer melhoramentos para a solução.

Os autores Kershenbaum, Boorstyn e Openheim [18] através de resultados computacionais realizados para exemplos de dimensão pequena verificaram que em geral, algumas das arestas da solução óptima, mas não pertencentes a uma solução admissível, são arestas da árvore de suporte de custo mínimo. Deste modo arestas contidas na árvore de suporte de custo mínimo e não numa solução admissível são arestas que deverão ser analisadas quando se pretendem melhorar soluções admissíveis. Seguindo esta sugestão apresentamos a seguinte heurística para o DMST.

Considere-se S_a a solução correspondente à árvore de suporte de custo mínimo e S_d uma

solução admissível inicial que será a solução de referência. Vamos agora considerar um novo conjunto designado por E_{ad}=$S_d \setminus S_a$ (conjunto de arestas de inibição), isto é, o conjunto das arestas que estão na solução S_d e não estão na solução da árvore de suporte de custo mínimo e considere-se um outro conjunto O_{ad}=$S_a \setminus S_d$ (conjunto de arestas de junção), isto é, as arestas que estão na árvore de suporte de custo mínimo e não estão na solução S_d. A escolha dos conjuntos E_{ad} e O_{ad} tem portanto a ver com o facto de as arestas da árvore de suporte de custo mínimo serem consideradas boas candidatas a estarem na solução e, por esse motivo exigirmos que as arestas que estão na árvore S_a e não estão na solução S_d sejam boas candidatas a serem obrigadas a estarem presentes na solução, enquanto que as arestas que não estão na árvore S_a mas estão na solução S_d são boas candidatas a serem proíbidas de estar na solução.

Quando o número de nodos de um grafo aumenta os conjuntos E_{ad} e O_{ad} podem conter muitas arestas (no máximo podem conter $n-1$ arestas). Será então necessário tornar estes conjuntos mais pequenos, para tal, designa-se por E_G o conjunto das arestas do grafo completo ordenadas por ordem crescente de custo. Seja S_{c_1} o conjunto formado pelas $\frac{n(n-1)}{2} - 2n$ primeiras arestas do conjunto E_G e S_{c_2} o conjunto formado pelas $2n$ últimas arestas do conjunto E_G e obtém-se novos conjuntos $E_{adc} = E_{ad} \setminus S_{c_1}$ (conjunto de arestas de inibição) e $O_{adc} = O_{ad} \setminus S_{c_2}$ (conjunto de arestas de junção). Este procedimento vai ser efectuado para grafos com mais do que 5 nodos.

Uma vez obtidos os conjuntos de arestas de inibição e de junção, vamos obter k soluções, sendo k o número de arestas do conjunto O_{adc}. Cada solução é obtida por junção de uma aresta do conjunto O_{adc} e por eliminação de todas as arestas do conjunto E_{adc}. Para obtenção destas k soluções inicia-se a construção da árvore com a aresta $\{k, \ell\}$ (elemento inicial) e procede-se à adição das arestas tal como na Heurística Greedy 2 com a excepção de proibir as arestas do conjunto E_{adc} de pertencerem à solução. Obtém-se assim, para cada a_i, $i = 1, \ldots, k$ um valor z_{a_i}, $i = 1, \ldots, k$ e de entre todos os valores z_{a_i}, $i = 1, \ldots, k$ escolhe-se a solução de menor custo, seja ela z_{adc}. Se esta solução for maior que z_d (valor da solução de referência) escolhe-se z_d, caso contrário escolhe-se z_{adc}.

HEURÍSTICA DE INIBIÇÃO E JUNÇÃO

- **Passo 0** - *Inicialização*

  - Obter a solução correspondente à árvore de suporte de custo mínimo, seja S_a essa solução.
  - Obter uma solução de referência, seja S_d o conjunto das suas arestas e o seu valor z_d.
  - Obter E_{ad}=$S_d \setminus S_a$ e O_{ad}=$S_a \setminus S_d$.

  Se o grafo tiver mais do que 5 nodos:

  - Obter E_{adc}=$E_{ad} \setminus S_{c_1}$ e O_{adc}=$O_{ad} \setminus S_{c_2}$, onde k é o números de arestas do conjunto O_{adc}.

- **Passo 1** - *Obtenção das soluções com junção e inibição de arestas*

  Determinar uma solução admissível exigindo que cada aresta a_i do conjunto O_{adc} esteja na solução e que nenhuma aresta do conjunto E_{adc} esteja na solução.

- **Passo 2** - *Escolha da solução de referência*

  Escolher z_{adc}=min(z_{a_1}, ..., z_{a_k}) e seja S_{adc} o correspondente conjunto de arestas.

  Caso $z_{adc} < z_d$ a nova solução de referência é S_{adc}.

  Caso contrário, a solução e o valor final obtido serão S_d e z_d respectivamente.

Apresenta-se de seguida o mesmo exemplo usado nas heurísticas anteriores para exemplificar a Heurística de Inibição e Junção.

Consideremos $D = 3$. Obtemos a solução da árvore de suporte de custo mínimo que se encontra na Figura 3.14 da página 40 sendo $S_a = \{\{1,2\},\{1,5\},\{2,3\},\{3,4\}\}$. Considere-se como solução de referência inicial a solução obtida pela Heurística de Melhoramento que se encontra representada na Figura 3.13 da página 39 sendo $S_d = \{\{1,3\},\{1,5\},\{2,3\},\{3,4\}\}$ e $z_d = 39$. Os conjuntos $E_{ad} = \{\{1,3\}\}$ e $O_{ad} = \{\{1,2\}\}$. A aresta $\{1,3\}$ está proibída de estar na solução e a aresta {1,2} é obrigada a estar na solução. Assim, obtém-se o grafo da Figura 3.23.

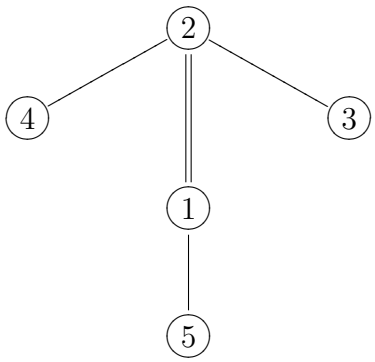


Figura 3.23: Grafo com diâmetro $D = 3$ e custo $z_{a_1} = 41$ obtido durante o processo de resolução da Heurística de Inibição e Junção por proibição da aresta $\{1, 3\}$ e obrigação da presença da aresta $\{1, 2\}$ na solução.

Como $z_{ad} > z_d$ escolhe-se $z_d = 39$. Esta solução encontra-se representada na Figura 3.13 da página 39 e é igual à da Heurística de Melhoramento e à Heurística de Inibição e com custo inferior às restantes heurísticas.

Para $D = 4$ os conjuntos E_{ad} e O_{ad} ficam vazios, portanto obtém-se neste caso a solução de referência que corresponde à árvore de suporte de custo mínimo.

As Heurísticas de Inibição e Junção diferem no facto de umas obrigarem e outras proibirem cada uma das arestas de um dado conjunto a estarem presentes na solução. A Heurística de Inibição/Junção usa as duas estratégias em conjunto.

Em ambas as Heurísticas de Inibição e de Junção é necessário a obtenção de uma solução admissível inicial. Usando para tal uma das Heurísticas Construtivas apresentadas, significa que este processo requer um total de $O(n^2)$ operações. A procura de arestas não ligadas ao nodo central requer, no pior dos casos, n comparações. Após este processo a formação de k árvores requer $O(kn^2)$ operações. No pior dos casos serão obtidas n árvores. Logo, a complexidade da Heurística de Inibição e da Heurística de Junção será, no pior dos casos, $O(n^3)$.

Na Heurística de Inibição/Junção começamos por obter a solução correspondente à árvore de suporte de custo mínimo que requer $O(n^2)$ operações e temos também de obter uma solução de referência para a qual também são necessárias $O(n^2)$ operações. Para a obtenção dos conjuntos O_{adc} e E_{adc} são necessárias n comparações para a formação de cada um dos conjuntos. Após a obtenção destes conjuntos formam-se k árvores, o que requer $O(kn^2)$

operações. No pior dos casos serão obtidas n árvores. Logo, a complexidade da Heurística de Inibição/Junção, no pior dos casos é de $O(n^3)$.

3.4 Heurísticas conhecidas na literatura

Na literatura existem alguns trabalhos que apresentam heurísticas para o problema da Árvore de Suporte de Custo Mínimo com Restrições de Diâmetro. Os trabalhos de Abdalla, Deo, Kumar e Terry [4], Abdalla, Deo e Fraceschini [3], Deo e Abdalla([1] e [2]) apresentam duas heurísticas para o problema e examinam implementações em paralelo dessas heurísticas. Uma das heurísticas apresentadas baseia-se numa modificação do algoritmo de Prim, é pois uma heurística construtiva. A outra heurística é uma heurística de trocas locais ou melhoramento e efectua trocas na árvore de suporte de custo mínimo de modo a obter-se uma árvore de suporte com diâmetro restringido a um determinado valor D.

Apresentamos nesta secção uma descrição sumária destas duas heurísticas apresentando-se para cada uma delas o mesmo exemplo utilizado nas heurísticas apresentadas ao longo deste capítulo.

HEURÍSTICA OTTC

A primeira heurística apresentada por estes autores é baseada na forma construtiva de greedy. *One-Time Tree Construction* (OTTC) foi o nome atribuído a esta heurística pelos seus autores e baseia-se numa modificação do algoritmo de Prim. Começa-se por escolher um nodo para iniciar a construção da árvore e em cada iteração vai-se introduzindo a aresta de custo mínimo, por forma a verificar-se a restrição imposta pelo diâmetro (o comprimento de qualquer caminho não pode ser superior a D). Como o algoritmo é sensível ao nodo que inicialmente é escolhido, este procedimento é repetido n vezes, tantas quanto o número de nodos do grafo. Em cada repetição é iniciada a construção da árvore por um nodo diferente, obtendo-se assim n árvores de suporte, uma por cada nodo inicial seleccionado. No final, de entre as n árvores de suporte obtidas, escolhe-se para solução final a árvore de menor custo. Este é o processo implementado para grafos com poucos nodos. Quando um grafo é constituído por muitos nodos o tempo requerido com este processo é significativamente grande o que levou os autores a diminuirem o número de árvores construídas e, portanto, a

repetirem o procedimento apenas um número constante q de vezes ($q < n$), independentemente do valor de n. Muitas das vezes os autores consideram $q = 5$ e seleccionam os q nodos cuja soma dos pesos das arestas incidentes seja a menor possível . De entre as q árvores de suporte construídas no final, escolhe-se para solução a árvore de suporte de menor custo.

Para grafos não completos este algoritmo pode nem sempre encontrar uma solução.

A Heurística Greedy 1 apresentada nesta tese difere apenas da Heurística OTTC no número de árvores construídas, uma vez que na Heurística Greedy 1 apenas se constrói uma árvore, cuja construção se inicia num nodo arbitrário. Não são construídas q árvores.

Usando o exemplo considerado para as heurísticas anteriores vamos exemplificar o funcionamento desta heurística.

Consideremos $D = 2$. Escolhendo para iniciar a árvore o nodo 1, 2 ou 5 obtém-se o primeiro grafo da Figura 3.24 e iniciando a construção da árvore pelo nodo 3 ou 4 obtém-se o segundo grafo da mesma figura.

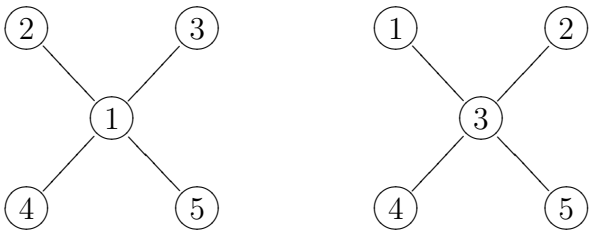


Figura 3.24: Soluções admissíveis com custos 54 e 51, respectivamente, e diâmetro $D = 2$ obtidas pela Heurística OTTC.

Deste modo escolhe-se para solução final o grafo que corresponde à árvore com custo 51, ou seja, a solução de menor custo. A solução obtida por esta heurística é a mesma que a obtida pela Heurística Greedy 1 e inferior à obtida nas Heurísticas Greedy 2 e 3.

Consideremos $D = 3$. Escolhendo para iniciar a árvore o nodo 1, 2 ou 5 obtém-se o primeiro grafo da Figura 3.25 e iniciando a construção da árvore pelo nodo 3 ou 4 obtém-se o segundo grafo da mesma figura.

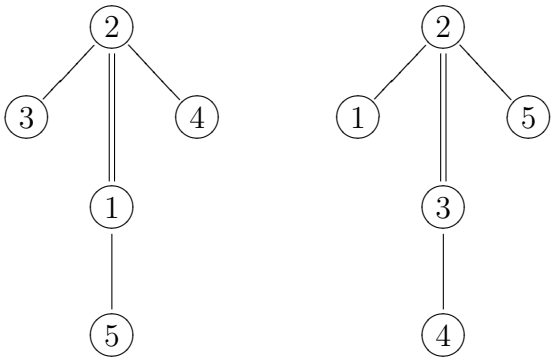


Figura 3.25: Soluções admissíveis com custos 41 e 45, respectivamente, e diâmetro $D = 3$ obtidas pela Heurística OTTC.

Escolhe-se para solução final o grafo correspondente ao custo 41, ou seja, a solução de menor custo. A solução obtida por esta heurística tem um custo superior à obtida pela Heurística de Melhoramento, de Inibição, de Inibição/Junção e inferior às restantes heurísticas.

Para $D = 4$ obtém-se a árvore de suporte de custo mínimo representada no grafo da Figura 3.15 da página 44.

HEURÍSTICA CIR

A segunda heurística, cujo nome que lhe foi atribuído é *Composite Iterative Refinement* (CIR) começa por obter a árvore de suporte de custo mínimo. Depois, iterativamente refina a árvore de suporte de custo mínimo até que a restrição do diâmetro seja satisfeita. Cada iteração deste refinamento heurístico é formada por duas partes. Uma primeira parte consiste na selecção da aresta a remover da árvore de forma a que esta remoção possa contribuir para a redução do diâmetro. Uma segunda parte consiste na escolha de uma aresta para substituir a aresta removida.

Considera-se que uma aresta no centro de um longo caminho é uma boa candidata a ser removida desde que divida o caminho em dois subconjuntos com o mesmo comprimento. Para a identificação dessa aresta é necessário determinar o(s) nodo(s) u da árvore de suporte de custo mínimo T tais que $ecc_T(u)=\lceil \frac{d}{2} \rceil$, onde d é o diâmetro da árvore de suporte de custo mínimo e $ecc_T(u)$ denota a excentricidade do nodo u com respeito à árvore T, ou seja, é a distância máxima de u a algum outro nodo de T. Se existir dois nodos x e y tais que a excentricidade é igual a $\lceil \frac{d}{2} \rceil$, então o diâmetro da árvore de suporte de custo mínimo é ímpar e a aresta $\{x, y\}$ é a aresta central, portanto a candidata a remover da árvore. Ao remover-se a aresta $\{x, y\}$ da árvore divide-se a árvore em duas subárvores (*subtree1* e *subtree2*) que têm, cada uma, um nodo v com excentricidade igual ao diâmetro d da árvore T. No caso do diâmetro da árvore de suporte de custo mínimo ser par temos apenas um nodo, digamos x com excentricidade igual a $\lceil \frac{d}{2} \rceil$ e neste caso todas as arestas incidentes no nodo x são candidatas a ser removidas. Como há mais do que uma aresta candidata a ser removida, forma-se uma lista ordenada do custo mais alto para o mais baixo e escolhe-se a aresta de maior custo, ou seja, a primeira da lista. No caso da aresta candidata a ser removida não conter um nodo v com excentricidade igual ao diâmetro da árvore remove-se da lista e considera-se a aresta seguinte. Caso a lista fique vazia identifica-se o nodo u com $ecc_T(u)=\lceil \frac{d}{2} \rceil + b$, onde b é inicializado a zero e aumenta uma unidade em cada iteração.

Quando se remove uma aresta da árvore T, digamos $\{x, y\}$, particcionamos a árvore em duas subárvores e temos de seleccionar uma aresta que não pertença à árvore T para fazer a ligação das duas subárvores de maneira que reduza o comprimento do caminho mais longo sem aumentar o diâmetro. Suponhamos que ficamos com as subárvores: $subtree1$ e $subtree2$, onde $x \in subtree1$ e $y \in subtree2$. A aresta que vai subtituir a aresta removida é a aresta $\{a, b\}$ de peso mínimo tal que: (i) a nova aresta não aumenta o diâmetro da árvore e (ii) reduz em uma unidade o comprimento do caminho mais longo da árvore. Ou seja, a aresta $\{a, b\}$ que vai substituir a aresta removida $\{x, y\}$ é tal que:

(i) $ecc_{subtree1}(a) \leq ecc_{subtree1}(x)$ **e** $ecc_{subtree2}(b) \leq ecc_{subtree2}(y)$

(ii) $ecc_{subtree1}(a) < ecc_{subtree1}(x)$ **ou** $ecc_{subtree2}(b) < ecc_{subtree2}(y)$

Considerando mais uma vez o exemplo usado para todas as heurísticas apresentadas ao longo desta tese, vamos exemplicar esta heurística.

Usando o algoritmo de Prim obtém-se a árvore de suporte de custo mínimo representada na Figura 3.15 da página 44, que tem diâmetro $d = 4$. A excentricidade de cada um dos nodos dessa árvore é:

$$ecc_T(1) = 3$$
$$ecc_T(2) = 2$$
$$ecc_T(3) = 3$$
$$ecc_T(4) = 4$$
$$ecc_T(5) = 4$$

Começamos por identificar os nodos u da árvore tais que $ecc_T(u) = \lceil \frac{d}{2} \rceil = 2$. Temos portanto, $u = 2$ e $\{1, 2\}$ e $\{2, 3\}$ são as arestas candidatas a serem removidas da árvore, pois o diâmetro da árvore é par ($d = 4$). Como $ecc_T(1) = 3 \neq 4 = d$ e $ecc_T(3) = 3 \neq 4 = d$ estes nodos não têm excentricidade igual ao diâmetro da árvore pelo que, as arestas $\{1, 2\}$ e $\{2, 3\}$ deixam de ser candidatas a ser removidas. Pelo que agora teremos de encontrar os nodos u tal que $ecc_T(u) = 2 + 1 = 3$. Assim, os nodos com $ecc_T(u) = 3$ são os nodos 1 e 3 e deste modo a lista das arestas a remover é formada pelas arestas $\{1, 5\}$ e $\{3, 4\}$. Desta vez, os nodos 4 e 5 têm excentricidade igual ao diâmetro da árvore, pelo que a lista de arestas é formada por: $\{1, 5\}$ e $\{3, 4\}$. Começa-se por remover a aresta $\{1, 5\}$ pois tem maior custo e ao removermos esta aresta ficamos com a árvore T dividida em duas subárvores, a subárvore 1 (*subtree1*) constituída apenas pelo nodo 5 e a subárvore 2 (*subtree2*) constituída pelas arestas $\{1, 2\}$, $\{2, 3\}$ e $\{3, 4\}$ como se apresenta na Figura 3.26.

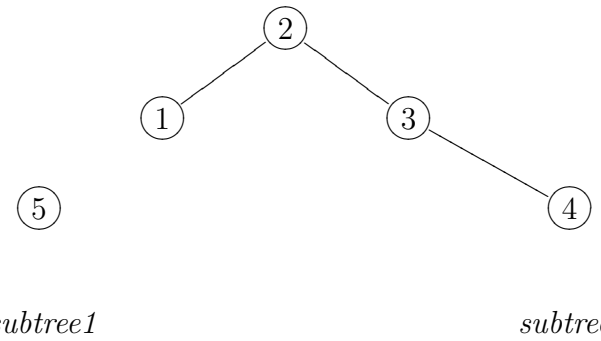


Figura 3.26: Subárvores obtidas após a eliminação da aresta $\{1, 5\}$.

Para obtenção de uma árvore com diâmetro $D = 3$ a aresta a ser inserida não pode aumentar o diâmetro da árvore ($d = 4$) e tem que reduzir em uma unidade o comprimento do caminho mais longo da árvore. Assim, as duas arestas nestas condições são a $\{2, 5\}$

ou $\{3,5\}$. Insere-se a aresta $\{2,5\}$ pois é a aresta de menor custo. Desta forma obtém-se o diâmetro pretendido e o grafo correspondente com custo 45 encontra-se representado na Figura 3.27.

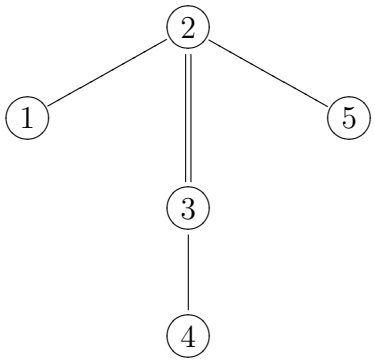


Figura 3.27: Grafo com diâmetro $D = 3$ e custo 45 obtido pela Heurística CIR.

Esta solução obtida é a mesma solução que a determinada pelas Heurísticas Greedy 1 e 2, pela Heurística Simples de Aproximação e pela Heurística de Junção e tem um valor superior ao valor obtido pelas Heurísticas de Melhoramento, de Inibição, de Inibição/Junção e pela OTTC. A única heurística que apresenta um valor superior a esta é a Heurística Greedy 3.

Para obtenção de uma árvore com diâmetro $D = 2$ tal como anteriormente as arestas candidatas a ser removidas são $\{1,5\}$ e $\{3,4\}$. Removendo a aresta $\{1,5\}$ obtém-se o grafo representado na Figura 3.27 com diâmetro $D = 3$. Para reduzir o diâmetro em uma unidade remove-se da árvore a segunda aresta da lista, sendo ela $\{3,4\}$ e obtém-se duas subárvores, a subárvore 1 (*subtree1*) que é constituída pelas arestas $\{1,5\}$, $\{1,2\}$ e $\{2,3\}$ e a subárvore 2 (*subtree2*) que é constituída apenas pelo nodo 4 e encontram-se representadas na Figura 3.28.

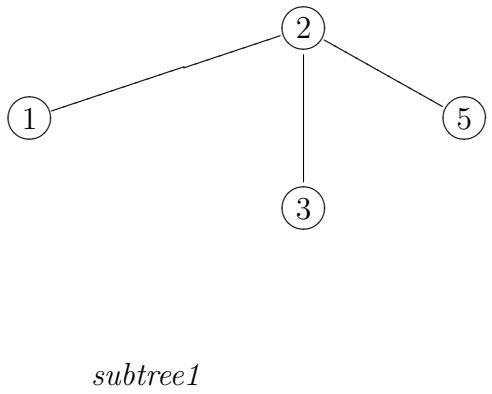


Figura 3.28: Subárvores obtidas após a eliminação da aresta $\{3, 4\}$.

A aresta a ser inserida não pode aumentar o diâmetro da árvore da Figura 3.27 e tem que reduzir em uma unidade o comprimento do caminho mais longo dessa árvore. Assim a única aresta nestas condições é a aresta $\{2, 4\}$. Desta forma obtém-se o diâmetro pretendido. Apresenta-se no grafo da Figura 3.29 a solução respectiva.

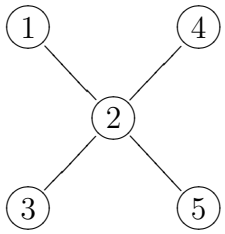


Figura 3.29: Grafo com diâmetro $D = 2$ e custo 51 obtido pela Heurística CIR.

Na Heurística OTTC a obtenção de cada árvore (iniciando a sua construção por um nodo) é proporcional a n^2. Como nesta Heurística se obtêm q árvores requer um tempo de execução $O(qn^2)$. No máximo podemos construir n árvores, cada uma delas com início num nodo diferente, pelo que no pior dos casos a complexidade da Heurística OTTC é $O(n^3)$ [1].

A Heurística OTTC é uma heurística baseada numa modificação do algoritmo de Prim e a escolha do nodo para iniciar a construção da árvore tem um grande impacto na obtenção de boas soluções. Surgiu assim a Heurística chamada *randomized centre-based tree construction* (RTC)[22] que consiste em iniciar a construção da árvore por um nodo arbitrário, considerando-o como sendo o nodo central (se o diâmetro D for par) ou tomar dois nodos iniciais, ou seja, uma aresta (se o diâmetro D for ímpar), obtendo-se com esta modificação

heurística limites superiores mais baixos.

Na Heurística CIR a obtenção da árvore de suporte de custo mínimo usando o algoritmo de Prim e o cálulo das excentricidades de todos os nodos desta requer um total de $O(n^2)$ operações. No pior caso encontrar a lista de arestas candidatas a serem removidas da árvore requer n comparações e a sua ordenação requer um tempo de execução $O(nlogn)$. O processo de remoção e substituição de arestas é executado em $O(n^2)$ operações. Contudo, no pior caso, a complexidade da Heurística CIR é de $O(n^3)$ [1].

3.5 Quadro síntese

Ao longo deste capítulo apresentámos diversas heurísticas: Heurísticas Greedy, Heurísticas de Trocas Locais ou Melhoramento e Heurísticas de Aproximação e fizemos ainda referência a duas Heurísticas conhecidas na literatura; a Heurística OTTC e a Heurística CIR. Para exemplificar estas heurísticas usámos uma instância de 5 nodos. No quadro seguinte apresentamos um resumo indicando o valor obtido por cada heurística considerada para a instância usada neste capítulo para diâmetros $D = 2$, $D = 3$ e $D = 4$ e indicamos também a complexidade computacional de cada uma. Foram considerados exemplos para $D = 2$ e $D = 3$ por uma questão de simplicidade.

	Complexidade Computacional	$D = 2$	$D = 3$	$D = 4$
óptimo		51	39	35
HG1	$O(n^2)$	**51**	45	**35**
HG2	$O(n^2)$	54	45	**35**
HG3	$O(n^2)$	54	56	**35**
HM	$O(n^2)$	-	**39**	**35**
HA	$O(n^2)$	-	45	**35**
HI	$O(n^3)$	-	**39**	39
HJ	$O(n^3)$	-	45	35
HIJ	$O(n^3)$	-	**39**	**35**
OTTC	$O(n^3)$	**51**	41	**35**
CIR	$O(n^3)$	**51**	45	**35**

Tabela 3.14: Resultados das heurísticas para o exemplo de 5 nodos apresentado ao longo deste capítulo, cuja árvore de suporte de custo mínimo tem diâmetro $D = 4$ e custo 35.

Notamos que na coluna correspondente ao diâmetro $D = 2$ algumas heurísticas não têm o valor da solução, pois usam uma solução com $D = 2$ como solução inicial é o caso da Heurística de Melhoramento cuja solução inicial têm diâmetro $D = 2$ e das Heurísticas de Aproximação em que o nodo t corresponde ao nodo central da solução $D = 2$.

Capítulo 4

Resultados Computacionais

Neste capítulo apresentam-se alguns resultados computacionais das heurísticas apresentadas no capítulo anterior para instâncias do Problema da Árvore de Suporte de Custo Mínimo com Restrições de Diâmetro para grafos completos com custos Euclideanos e com um número de nodos a variar entre 10 e 60 (10, 20, 30, 40 e 60) e com o valor do diâmetro D a variar entre 4 e 9.

Para obtermos as instâncias com custos Euclideanos, foram geradas aleatoriamente numa grelha de dimensão 100×100 as coordenadas de n pontos, que correspondem aos nodos do grafo original. Posteriormente, o valor dos custos c_{ij} de cada aresta $\{i, j\}$ foi obtido tomando a parte inteira da distância Euclideana entre os pontos i e j gerados na rede.

As heurísticas foram implementadas em Maple versão 10. Os resultados computacionais foram obtidos através de um Pentium Mobile 1500 GHz (CPU) e 496 MB de RAM.

Nas tabelas que se apresentam neste capítulo encontram-se os resultados obtidos no computador com o tempo de execução em segundos e o respectivo valor da solução admissível obtida. Para avaliar cada heurística comparamos o valor obtido (limite superior) com o valor óptimo ou limite inferior (lower bound) obtido usando as formulações descritas em [11] e [12] e que foram também apresentadas no Capítulo 2. Na primeira linha das tabelas encontra-se o diâmetro D (a variar entre 4 e 9), na segunda linha encontra-se o valor óptimo ou limite inferior (os valores correspondentes a limites inferiores estão assinalados nas tabelas com o símbolo "*") obtido usando as formulações descritas em [11] e [12] e apresentadas no Capítulo 2. Notemos que quando o símbolo utilizado for "$\circ$", significa que os valores não são os da relaxação linear, porque a resolução do dual não foi efectuada até o final por falta de memória do computador. Para ter noção da qualidade do limite superior obtido determinamos para cada heurística o valor do gap correspondente, que é dado por $gap = \frac{H-z}{z} \times 100$ (onde "z" é o valor óptimo ou limite inferior e "H" é o valor do limite

superior obtido pelas heurísticas apresentadas).

Para cada uma das instâncias consideradas é referido, na legenda da respectiva tabela, o valor óptimo da árvore de suporte de custo mínimo obtido pelo algoritmo de Prim e o respectivo tempo de execução.

As Heurísticas Greedy 1, 2 e 3 são designadas por HG1, HG2 e HG3, respectivamente. A Heurística de Melhoramento será designada por HM e a Heurística Simples de Aproximão será designada por HA. As Heurísticas de Inibição, Junção e Inibição/Junção serão designadas por HI, HJ e HIJ, respectivamente. A solução de referência é obtida pela Heurística Greedy 3. Esta escolha da solução de referência deve-se ao facto dos valores obtidos pela Heurística Greedy 3 não serem muito bons.

Uma vez que a Heurística OTTC é considerada pelos autores [2] um excelente algoritmo para obter soluções para o problema DMST serão também obtidos resultados computacionais para essa heurística para comparação com as heurísticas aqui descritas. Esta será designada nas tabelas por OTTC.

De seguida apresentamos quatro grupos de tabelas seguidas do gráfico com os respectivos gaps e alguns comentários. O primeiro grupo refere-se às instâncias de 20 nodos, o segundo às instâncias de 30 nodos, o terceiro às instâncias de 40 nodos e o quarto às instâncias de 60 nodos. Finalmente apresentamos um gráfico relativo aos gaps de todas as instâncias.

	D	**D=4**	**D=5**	**D=6**	**D=7**
	óptimo	235	227	222	219
HG1	t	0,3	0,2	0,3	0,0
	v	311	239	235	**219**
	gap	32,3	5,3	5,9	0,0
HG2	t	0,3	0,3	0,3	0,0
	v	284	263	258	224
	gap	20,9	15,9	16,2	2,3
HG3	t	0,2	0,3	0,2	0,0
	v	293	305	258	250
	gap	24,7	34,4	16,2	14,2
HM	t	0,7	0,9	0,8	0,9
	v	281	**227**	224	224
	gap	19,6	0,0	0,9	2,3
HA	t	0,6	0,4	0,4	0,0
	v	311	274	258	**219**
	gap	32,3	20,7	16,2	0,0
HI	t	1,4	1,2	1,1	0,9
	v	**255**	**227**	**222**	**219**
	gap	8,5	0,0	0,0	0,0
HJ	t	1,0	0,9	0,8	0,8
	v	293	239	258	**219**
	gap	24,7	5,3	16,2	0,0
HIJ	t	1,0	1,0	0,4	0,4
	v	284	239	235	**219**
	gap	20,9	5,3	5,9	0,0
OTTC	t	1,0	0,7	0,8	0,6
	v	284	239	258	**219**
	gap	20,9	5,3	16,2	0,0

Tabela 4.1: Exemplo ***tc10-1***: Resultados das heurísticas para o exemplo de 10 nodos, cuja árvore de suporte de custo mínimo tem diâmetro $D = 7$, custo 219 e tempo de execução 0 segundos.

	D	**D=4**	**D=5**	**D=6**	**D=7**	**D=8**	**D=9**
	óptimo	369	347	322	316	308	302
HG1	t	2,3	1,6	1,9	1,5	1,0	1,1
	v	596	444	472	400	320	348
	gap	61,5	28,0	46,2	26,6	3,9	15,2
HG2	t	2,0	2,4	1,7	1,5	1,5	1,1
	v	481	510	454	413	398	348
	gap	30,4	47,0	41,0	30,7	29,2	15,2
HG3	t	2,9	1,8	1,2	1,0	1,0	1,4
	v	685	474	**348**	**342**	328	387
	gap	85,6	36,6	8,1	8,2	6,5	28,1
HM	t	3,7	4,0	4,1	4,8	5,7	6,7
	v	564	482	460	381	322	**310**
	gap	52,8	38,9	42,9	20,6	4,5	2,6
HA	t	3,1	2,8	3,1	2,3	3,0	3,1
	v	481	431	449	345	395	387
	gap	30,4	24,2	39,4	9,2	28,2	28,1
HI	t	33,0	44,1	33,7	17,4	16,3	18,1
	v	**473**	**383**	**348**	**342**	**310**	314
	gap	28,2	10,4	8,1	8,2	0,6	4,0
HJ	t	5,8	4,4	2,5	8,2	5,5	5,0
	v	559	474	**348**	**342**	328	348
	gap	51,5	36,6	8,1	8,2	6,5	15,2
HIJ	t	35,8	31,0	11,5	12,8	5,4	9,2
	v	**473**	431	**348**	**342**	320	348
	gap	28,2	24,2	8,1	8,2	3,9	15,2
OTTC	t	11,5	9,7	9,1	8,5	6,0	4,5
	v	481	431	454	413	398	348
	gap	30,4	24,2	41,0	30,7	29,2	15,2

Tabela 4.2: Exemplo ***tc20-1***: Resultados das heurísticas para o exemplo de 20 nodos, cuja árvore de suporte de custo mínimo tem diâmetro $D = 13$, custo 292 e tempo de execução 0,4 segundos.

	D	**D=4**	**D=5**	**D=6**	**D=7**	**D=8**	**D=9**
	óptimo	406	379	353	345	334	328
HG1	t	2,8	3,2	2,6	3,0	1,4	1,1
	v	707	860	705	769	454	362
	gap	74,1	126,9	99,7	122,9	35,9	10,4
HG2	t	2,7	2,4	2,2	2,4	2,2	2,2
	v	642	593	580	591	574	562
	gap	58,1	56,5	64,3	71,3	71,9	71,3
HG3	t	3,7	3,3	2,6	1,8	1,5	1,0
	v	1037	860	713	500	454	362
	gap	155,4	126,9	102,0	44,9	35,9	10,4
HM	t	3,9	3,9	4,4	3,8	3,9	6,1
	v	699	656	550	**455**	428	**353**
	gap	72,2	73,1	55,8	31,9	28,1	7,6
HA	t	4,1	3,4	3,1	3,0	2,5	2,6
	v	642	499	470	456	**387**	390
	gap	58,1	31,7	33,1	31,9	15,9	18,9
HI	t	50,0	41,6	40,3	38,4	39,2	36,4
	v	642	**443**	564	500	410	362
	gap	58,1	16,9	59,8	44,9	22,8	10,4
HJ	t	6,0	5,2	4,1	4,3	4,2	3,6
	v	**542**	486	**434**	500	454	362
	gap	33,5	28,2	22,9	44,9	35,9	10,4
HIJ	t	47,0	40,3	33,5	27,5	19,5	14,6
	v	570	499	580	500	454	362
	gap	40,4	31,7	64,3	44,9	35,9	10,4
OTTC	t	14,4	12,6	10,7	10,5	9,2	7,5
	v	642	593	564	500	454	362
	gap	58,1	56,5	59,8	44,9	35,9	10,4

Tabela 4.3: Exemplo ***tc20-2***: Resultados das heurísticas para o exemplo de 21 nodos, cuja árvore de suporte de custo mínimo tem diâmetro $D = 14$, custo 314 e tempo de execução 0,3 segundos.

	D	**D=4**	**D=5**	**D=6**	**D=7**	**D=8**	**D=9**
	óptimo	393	355	328	308	295	284
HG1	t	3,0	1,7	1,6	1,3	1,7	0,9
	v	717	520	468	377	442	346
	gap	82,4	46,5	42,7	22,4	49,8	21,8
HG2	t	1,9	1,7	1,7	1,7	1,7	1,1
	v	541	520	464	446	442	346
	gap	37,7	46,5	41,5	44,8	49,8	21,8
HG3	t	4,7	2,5	3,0	1,7	1,7	1,4
	v	957	658	631	506	442	446
	gap	143,5	85,4	92,4	64,3	49,8	57,0
HM	t	5,3	5,5	5,5	5,6	6,5	7,0
	v	624	536	497	461	378	341
	gap	58,8	51,0	51,5	49,7	28,1	20,1
HA	t	3,3	3,2	3,1	3,0	2,9	2,9
	v	573	529	**405**	384	**329**	**300**
	gap	45,8	49,0	23,5	24,7	11,5	5,6
HI	t	35,4	30,8	20,4	20,1	24,9	15,1
	v	**512**	520	424	446	350	**300**
	gap	30,3	46,5	29,3	44,8	18,6	5,6
HJ	t	7,1	5,9	10,0	13,8	5,7	4,7
	v	628	642	424	**372**	356	325
	gap	59,8	80,8	29,3	20,8	20,7	14,4
HIJ	t	40,4	28,7	23,1	14,4	9,8	7,7
	v	540	**483**	422	385	356	**300**
	gap	37,4	36,1	28,7	25,0	20,7	5,6
OTTC	t	9,8	8,3	7,6	7,9	4,7	3,5
	v	541	520	464	446	356	**300**
	gap	37,7	46,5	41,5	44,8	20,7	5,6

Tabela 4.4: Exemplo ***tc20-3***: Resultados das heurísticas para o exemplo de 20 nodos, cuja árvore de suporte de custo mínimo tem diâmetro $D = 11$, custo 278 e tempo de execução 0,3 segundos.

	D	**D=4**	**D=5**	**D=6**	**D=7**	**D=8**	**D=9**
	óptimo	450	411	390	377	369	361
HG1	t	4,6	1,9	1,5	1,5	1,2	1,1
	v	644	559	**477**	482	**396**	**383**
	gap	43,1	36,0	22,3	27,9	7,3	6,1
HG2	t	3,2	1,6	1,5	1,5	1,4	1,1
	v	643	518	**477**	488	474	411
	gap	42,9	26,0	22,3	29,4	28,5	13,9
HG3	t	3,5	2,4	2,8	1,6	2,3	1,2
	v	922	689	848	549	678	458
	gap	104,9	67,6	117,4	45,6	83,7	26,9
HM	t	4,0	4,2	4,6	5,0	5,6	6,8
	v	1066	994	789	673	623	456
	gap	136,9	141,8	102,3	78,5	68,8	26,3
HA	t	3,1	3,4	3,3	2,8	2,2	2,7
	v	**551**	557	523	485	486	426
	gap	22,4	35,5	34,1	28,6	31,7	18,0
HI	t	43,0	21,1	27,7	18,9	22,8	16,2
	v	625	518	**477**	**405**	**396**	388
	gap	38,9	26,0	22,3	7,4	7,3	7,5
HJ	t	5,6	15,8	4,8	6,2	3,8	4,5
	v	597	**514**	500	482	412	**383**
	gap	32,7	25,1	28,2	27,9	11,7	6,1
HIJ	t	39,7	30,7	24,5	22,2	13,7	7,7
	v	643	536	**477**	452	**396**	**383**
	gap	42,9	30,4	22,3	19,9	7,3	6,1
OTTC	t	11,5	10,0	8,2	7,5	6,2	5,5
	v	643	559	**477**	482	**396**	**383**
	gap	42,9	35,3	22,3	27,9	7,3	6,1

Tabela 4.5: Exemplo ***tc20-4***: Resultados das heurísticas para o exemplo de 20 nodos, cuja árvore de suporte de custo mínimo tem diâmetro $D = 10$, custo 332 e tempo de execução 0,3 segundos.

	D	**D=4**	**D=5**	**D=6**	**D=7**	**D=8**	**D=9**
	óptimo	397	371	337	325	312	307
HG1	t	2,5	2,1	5,4	1,3	1,2	0,8
	v	791	542	431	391	422	325
	gap	99,2	46,1	27,9	20,3	35,3	5,9
HG2	t	2,9	2,4	1,4	1,3	1,3	1,0
	v	824	746	488	436	422	349
	gap	107,6	101,1	44,8	34,2	35,3	13,7
HG3	t	2,6	2,4	2,4	2,0	1,8	1,5
	v	791	658	604	569	479	386
	gap	99,2	77,4	79,2	75,1	53,5	25,7
HM	t	4,6	4,7	4,8	6,0	6,1	8,3
	v	710	615	572	388	368	318
	gap	78,8	65,8	69,7	19,4	17,9	3,6
HA	t	4,2	3,1	2,9	3,0	3,3	2,9
	v	655	**492**	480	455	444	379
	gap	65,0	32,6	42,4	40,0	42,3	23,5
HI	t	63,0	28,1	24,1	21,4	20,2	15,9
	v	791	515	**395**	**343**	329	**313**
	gap	99,2	38,8	17,2	5,5	5,4	2,0
HJ	t	5,4	4,8	4,3	3,6	3,3	2,8
	v	626	556	508	409	440	333
	gap	57,7	49,9	50,7	25,8	41,0	8,5
HIJ	t	37,2	25,0	17,8	13,3	9,7	7,6
	v	**614**	498	472	379	**325**	**313**
	gap	54,7	34,2	40,1	16,6	4,2	2,0
OTTC	t	12,4	9,6	7,1	6,2	5,3	4,1
	v	**614**	542	472	434	422	**313**
	gap	54,7	46,1	40,1	33,5	35,3	2,0

Tabela 4.6: Exemplo ***tc20-5***: Resultados das heurísticas para o exemplo de 20 nodos, cuja árvore de suporte de custo mínimo tem diâmetro $D = 13$, custo 300 e tempo de execução 0,3 segundos.

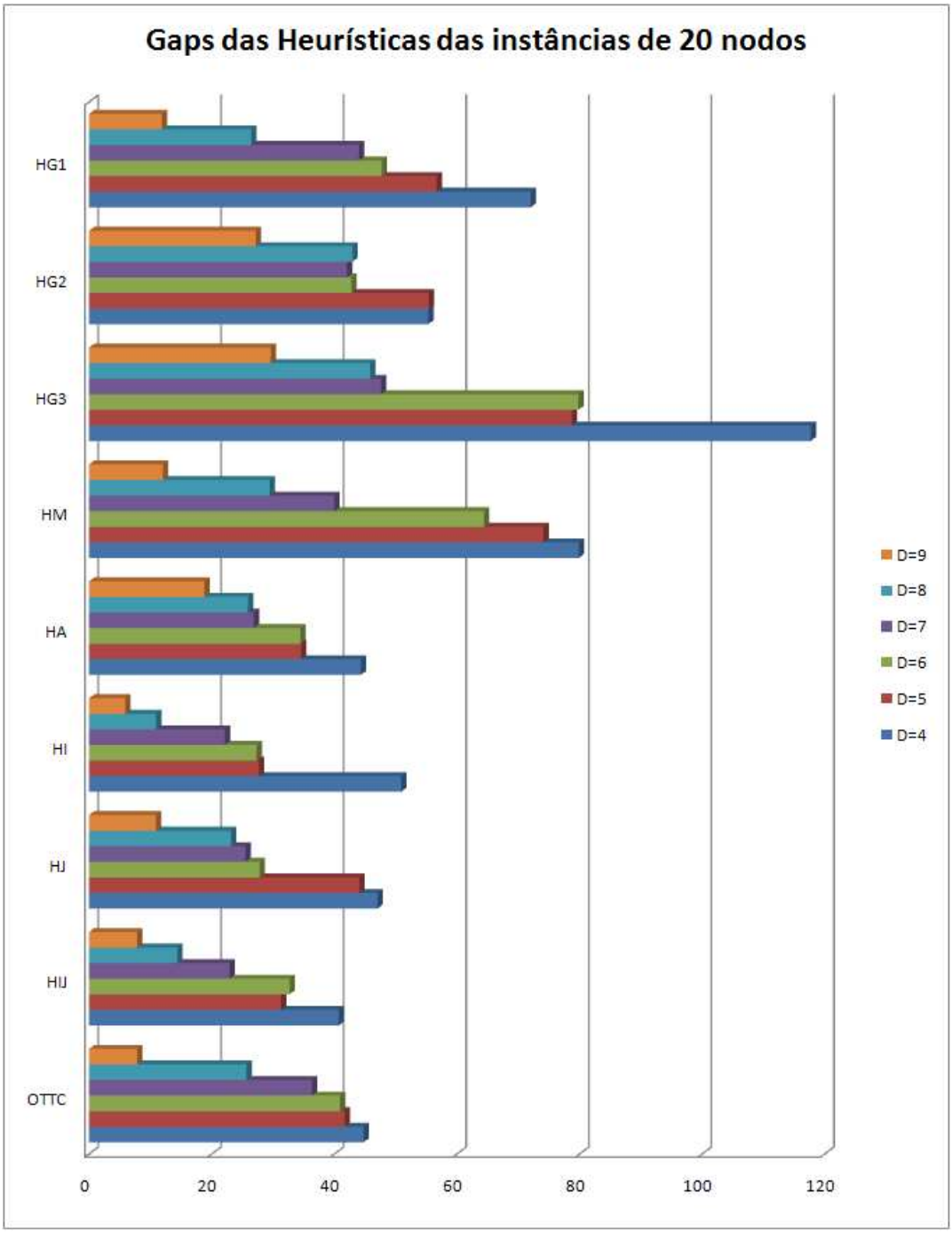


Figura 4.1: Gráfico relativo aos gaps das instâncias de 20 nodos.

Por observação do gráfico da Figura 4.1, relativo aos gaps das heurísticas das instâncias de 20 nodos podemos concluir que, a heurística que apresenta gaps mais elevados é a Heurística Greedy 3, para todos os diâmetros. Para um diâmetro 4, a heurística que apresenta gaps mais baixos é a Heurística de Inibição/Junção e para os restantes diâmetros a heurística que apresenta gaps mais baixos é a Heurística de Inibição. Existe sempre uma das três Heurísticas de Aproximação HI, HJ e HIJ a obter gaps mais baixos do que as restantes heurísticas.

Em relação às três Heurísticas Construtivas Greedy, a Heurística Greedy 1 é a que apresenta os gaps mais baixos para diâmetros 8 e 9, para os restantes diâmetros os gaps obtidos pela Heurística Greedy 2 são os mais baixos. Então, a Heurística Greedy 3 é a heurística construtiva que apresenta gaps mais elevados. Em geral a Heurística OTTC obtém gaps inferiores a qualquer das Heurísticas Construtivas Greedy. Notemos que existem alguns casos particulares para o qual, este facto não se verifica, como é o caso do exemplo tc20-1 onde a Heurística HG1 obtém para diâmetros 7 e 8 gaps inferiores aos obtidos pela Heurística OTTC e, nesse mesmo exemplo, a Heurística HG3 obtém para diâmetros 6, 7 e 8 gaps inferiores à Heurística OTTC e a Heurística Greedy 2 obtém para diametros 4, 6, 7, 8 e 9 gaps iguais aos obtidos pela Heurística OTTC. No exemplo tc20-4 também se pode observar que, para um diâmetro 5 o gap obtido pela Heurística HG2 é inferior ao obtido pela Heurística OTTC.

Nas Heurísticas de Inibição, de Junção e de Inibição/Junção é usada uma solução de referência obtida através da Heurística Greedy 3, uma vez que, nesta se obtêm valores elevados. Podemos observar graficamente que, existe uma grande redução nos gaps destas três heurísticas comparativamente com a Heurística Greedy 3. Por exemplo, para um diâmetro $D = 4$ obtém-se uma redução de 66,78 % (117,72 - 50,94), 70,68 % (117,72 - 47,04) e 77 % (117,72 - 40,72) respectivamente para as Heurísticas HI, HJ e HIJ.

A Heurística Simples de Aproximação obtém valores dos gaps inferiores à Heurística OTTC para diâmetros 4, 5, 6 e 7. Os gaps obtidos pela Heurística de Melhoramento em geral são superiores aos obtidos pela Heurística OTTC, mas existem certos casos particulares onde tal facto não se verifica, como é o caso da instância tc20-2 onde a Heurística HM para diâmetros 7 e 9 é a que obtém os gaps mais baixos comparando com as restantes heurísticas.

	D	**D=4**	**D=5**	**D=6**	**D=7**	**D=8**	**D=9**
	óptimo	587	534	482	463	437	423
HG1	t	11,4	8,4	5,9	4,3	4,8	6,3
	v	973	797	672	569	545	607
	gap	65,8	49,3	39,4	22,9	24,7	43,5
HG2	t	8,3	10,4	6,4	4,9	4,2	3,9
	v	845	862	672	577	522	502
	gap	44,0	61,4	39,4	24,6	19,5	18,7
HG3	t	12,2	18,7	7,3	8,3	4,4	6,8
	v	982	1230	687	779	522	661
	gap	67,3	130,3	42,5	68,3	19,5	56,3
HM	t	19,2	20,1	20,4	22,2	25,0	29,9
	v	969	929	889	778	600	497
	gap	65,1	74,0	84,4	68,3	37,3	17,5
HA	t	14,3	12,9	11,3	10,1	10,2	10,0
	v	834	731	641	622	572	457
	gap	42,1	36,9	33,0	34,3	30,9	8,0
HI	t	319,3	234,8	176,1	154,9	98,7	104,6
	v	776	723	**605**	**565**	496	**452**
	gap	32,2	35,4	25,5	22,0	13,5	6,9
HJ	t	21,6	28,8	14,9	56,9	40,4	22,8
	v	829	807	672	585	**468**	495
	gap	41,2	51,1	39,4	26,3	7,1	17,0
HIJ	t	320,8	272,5	138,0	126,6	75,2	59,6
	v	**771**	**689**	620	**565**	496	484
	gap	31,3	29,0	28,6	22,0	13,5	14,4
OTTC	t	55,4	45,9	42,6	35,5	30,5	30,4
	v	893	723	704	585	613	495
	gap	52,1	35,4	46,1	26,3	40,3	17,0

Tabela 4.7: Exemplo ***tc30-1***: Resultados das heurísticas para o exemplo de 30 nodos, cuja árvore de suporte de custo mínimo tem diâmetro $D = 15$, custo 396 e tempo de execução 0,8 segundos.

	D	**D=4**	**D=5**	**D=6**	**D=7**	**D=8**	**D=9**
	óptimo	539	499	445	424	404	395
HG1	t	13,9	17,6	9,8	13,5	7,2	4,1
	v	1262	1140	900	959	633	441
	gap	134,1	128,5	102,2	126,2	56,7	11,6
HG2	t	14,4	13,1	9,9	10,9	5,4	4,5
	v	1173	958	900	857	539	485
	gap	117,6	92,0	102,2	102,1	33,4	22,8
HG3	t	13,9	13,8	9,5	7,5	8,0	7,1
	v	1262	1256	834	704	628	648
	gap	134,1	151,7	87,4	66,0	57,9	64,1
HM	t	18,8	19,4	19,7	21,0	32,3	32,6
	v	1055	932	852	780	715	684
	gap	95,7	86,8	91,5	84,0	77,0	73,2
HA	t	19,4	18,3	13,0	12,1	14,7	14,2
	v	1138	1102	788	765	931	811
	gap	111,1	120,8	77,1	80,4	130,4	105,3
HI	t	326,2	343,4	289,3	313,7	147,8	158,2
	v	859	933	**581**	704	**467**	**417**
	gap	59,4	87,0	30,6	66,0	15,6	5,6
HJ	t	28,6	24,5	20,9	17,5	33,5	15,8
	v	925	**671**	833	704	539	647
	gap	71,6	34,5	87,2	66,0	33,4	63,8
HIJ	t	320,8	272,5	254,0	112,0	75,2	95,0
	v	**771**	689	615	**535**	496	441
	gap	43,0	38,1	38,2	26,2	22,8	4,1
OTTC	t	73,1	60,1	51,9	42,8	28,0	19,0
	v	928	818	750	597	539	455
	gap	72,2	63,9	68,5	40,8	33,4	15,2

Tabela 4.8: Exemplo ***tc30-2***: Resultados das heurísticas para o exemplo de 30 nodos, cuja árvore de suporte de custo mínimo tem diâmetro $D = 18$, custo 374 e tempo de execução 0,8 segundos.

	D	**D=4**	**D=5**	**D=6**	**D=7**	**D=8**	**D=9**
	óptimo	572	515	468	450	430	417
HG1	t	14,0	12,7	10,2	6,3	5,0	4,0
	v	940	933	761	594	541	**482**
	gap	64,3	81,2	62,6	32,0	25,8	15,6
HG2	t	12,4	8,0	8,7	7,5	6,2	4,2
	v	928	735	745	652	584	**482**
	gap	62,2	42,7	59,2	44,9	35,8	15,6
HG3	t	15,5	16,3	11,4	10,8	6,6	6,6
	v	1129	1123	848	826	578	660
	gap	97,4	118,1	81,2	83,6	34,4	58,3
HM	t	23,8	24,9	25,3	26,9	27,2	31,2
	v	988	950	874	770	707	565
	gap	72,7	84,5	86,8	71,1	64,4	35,5
HA	t	15,6	14,9	13,1	10,8	11,0	9,3
	v	926	819	729	620	580	501
	gap	61,9	59,0	55,8	37,8	34,9	20,1
HI	t	204,9	188,7	218,1	200,2	157,4	113,7
	v	**651**	**645**	633	**565**	**490**	**482**
	gap	13,8	25,2	35,3	25,6	14,0	15,6
HJ	t	27,1	26,0	23,6	17,9	28,7	21,0
	v	826	698	761	574	506	**482**
	gap	44,4	35,5	62,6	27,6	17,7	15,6
HIJ	t	324,6	259,5	135,8	150,2	85,1	60,3
	v	741	715	**620**	594	541	**482**
	gap	29,5	38,8	32,5	32,0	25,8	15,6
OTTC	t	58,5	43,3	45,2	39,6	32,1	21,4
	v	881	735	745	652	584	**482**
	gap	54,0	42,7	59,2	44,9	35,8	15,6

Tabela 4.9: Exemplo ***tc30-3***: Resultados das heurísticas para o exemplo de 30 nodos, cuja árvore de suporte de custo mínimo tem diâmetro $D = 18$, custo 374 e tempo de execução 0,8 segundos.

	D	**D=4**	**D=5**	**D=6**	**D=7**	**D=8**	**D=9**
	óptimo	604	544	501	477	454	442
HG1	t	15,7	7,3	7,7	10,2	6,5	5,6
	v	1304	780	741	810	604	599
	gap	115,9	43,4	47,9	69,8	33,0	35,5
HG2	t	9,0	9,2	9,0	8,2	5,7	8,0
	v	964	877	883	849	647	820
	gap	59,6	61,2	76,2	78,0	42,5	85,5
HG3	t	20,3	15,0	13,3	13,2	8,8	9,0
	v	1615	1273	1123	1042	851	786
	gap	167,4	134,0	124,2	118,4	87,4	77,8
HM	t	24,6	24,6	26,8	30,9	32,7	37,0
	v	1132	1079	871	618	570	510
	gap	87,4	98,3	73,9	29,6	25,6	15,4
HA	t	16,0	14,0	11,5	11,9	9,6	9,2
	v	921	770	**662**	664	**535**	494
	gap	52,5	41,5	32,1	39,2	17,8	11,8
HI	t	217,1	261,7	178,7	169,2	135,1	122,4
	v	**830**	857	675	640	545	513
	gap	37,4	57,5	34,7	34,2	20,0	16,1
HJ	t	30,5	23,2	23,9	20,3	28,5	15
	v	913	**738**	883	668	547	**453**
	gap	51,2	35,7	76,2	40,0	20,5	2,5
HIJ	t	328,0	265,4	246,0	137,6	128,0	85,9
	v	**830**	780	699	**609**	547	494
	gap	37,4	43,4	39,5	27,7	20,5	11,8
OTTC	t	50,1	44	45,1	31,4	25,7	28,2
	v	964	780	717	658	547	597
	gap	59,6	43,4	43,1	37,9	20,5	35,1

Tabela 4.10: Exemplo ***tc30-4***: Resultados das heurísticas para o exemplo de 30 nodos, cuja árvore de suporte de custo mínimo tem diâmetro $D = 14$, custo 414 e tempo de execução 1,3 segundos.

	D	**D=4**	**D=5**	**D=6**	**D=7**	**D=8**	**D=9**
	óptimo	574	533	477*	457	435	427
HG1	t	3,3	13,0	12,1	14,1	5,7	4,9
	v	1176	1161	885	1011	590	558
	gap	104,9	117,8	85,5	121,2	35,6	30,7
HG2	t	11,7	6,1	11,1	5,9	5,3	5,7
	v	929	**682**	873	648	590	624
	gap	61,8	28,0	83,0	41,8	35,6	46,1
HG3	t	16,1	10,7	13,2	9,1	6,9	4,5
	v	1355	963	1122	737	682	583
	gap	136,1	80,7	135,2	61,3	56,8	36,5
HM	t	25,2	25,5	27,4	27,5	30,1	31,6
	v	1044	1001	885	851	663	608
	gap	81,9	87,8	85,5	86,2	52,4	42,4
HA	t	17,6	14,0	12,0	11,0	10,4	8,8
	v	961	785	693	638	558	527
	gap	67,4	47,3	45,3	39,6	28,3	23,4
HI	t	338,1	177,8	337,0	164,3	137,1	151,8
	v	929	**682**	771	**579**	550	489
	gap	61,8	28,0	61,6	26,7	26,4	14,5
HJ	t	26,8	22,9	21,8	21,3	81,4	16,9
	v	871	722	671	737	**496**	**481**
	gap	51,7	35,5	40,7	61,3	14,0	12,6
HIJ	t	318,3	276,3	235,9	209,9	224,0	99,7
	v	**775**	726	**592**	642	592	**481**
	gap	35,0	36,2	24,1	40,5	36,1	12,6
OTTC	t	73,6	62,2	60,1	55,5	41,9	29,3
	v	929	897	873	828	590	**481**
	gap	61,8	68,3	83,0	81,2	35,6	12,6

Tabela 4.11: Exemplo ***tc30-5***: Resultados das heurísticas para o exemplo de 30 nodos, cuja árvore de suporte de custo mínimo tem diâmetro $D = 16$, custo 402 e tempo de execução 0,8 segundos.

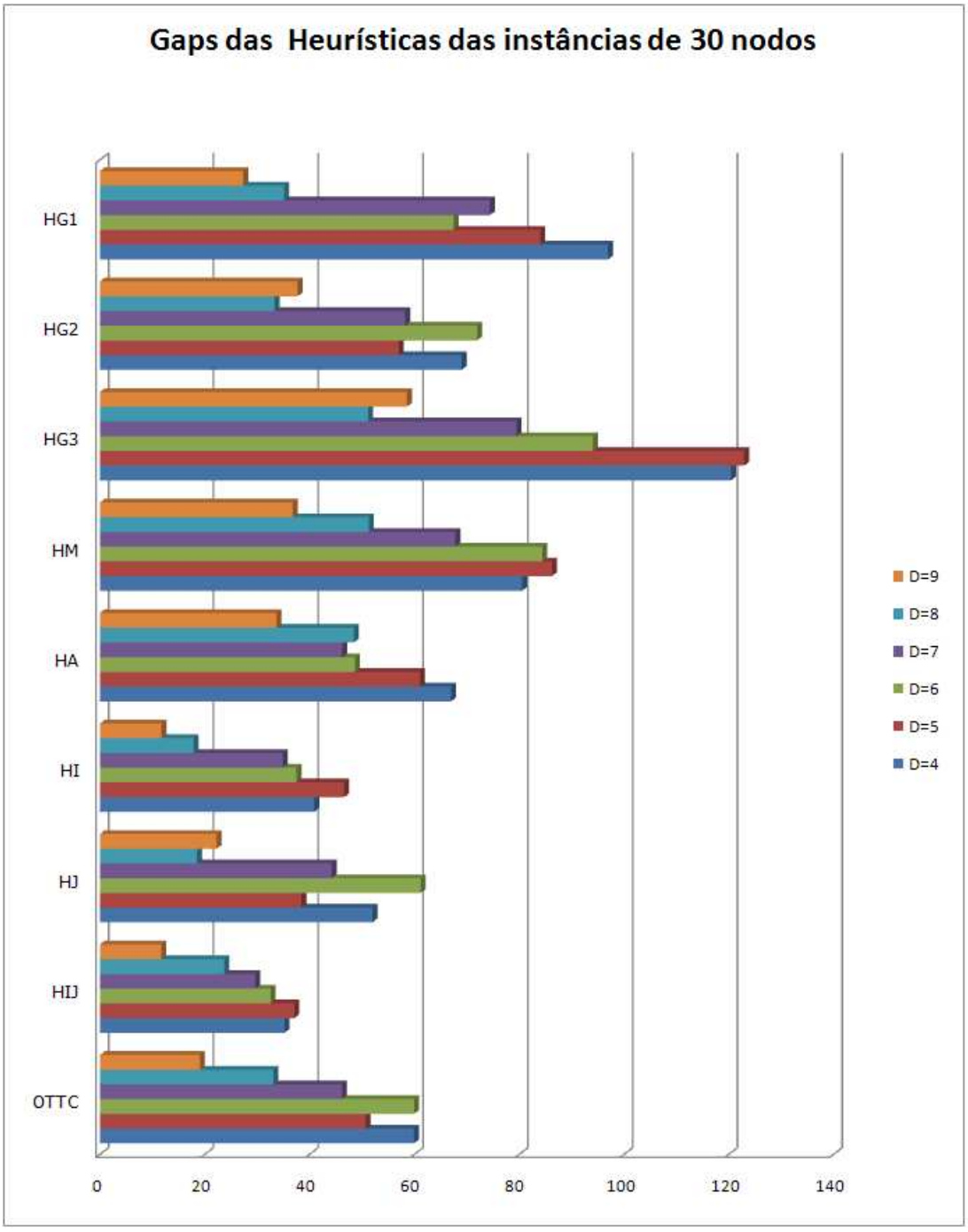


Figura 4.2: Gráfico relativo aos gaps das instâncias de 30 nodos.

Em geral para instâncias de 30 nodos é a Heurística Greedy 3 a que apresenta gaps mais elevados e a Heurística de Inibição/Junção a que apresenta gaps mais baixos (ver Figura 4.2) . Para um diâmetro 8 a heurística que apresenta gaps mais baixos é a Heurística de Inibição e para os restantes diâmetros é a Heurística de Inibição/Junção. Tal como no gráfico da Figura 4.1, relativo aos gaps das heurísticas de instâncias de 20 nodos este gráfico também mostra que as três Heurísticas de Aproximação HI, HJ e HIJ obtém gaps inferiores às restantes heurísticas, mesmo em relação à Heurística OTTC. Ao contrário do que acontecia no gráfico da Figura 4.1 que era a Heurística de Inibição que obtinha uma melhor qualidade para o limite superior da maioria dos diâmetros, as instâncias de 30 nodos é a Heurística de Inibição/Junção a obter gaps mais baixos.

Comparando as três Heurísticas Construtivas Greedy podemos observar que para diâmetros 6 e 9 é a Heurística Greedy 1 que tem gaps mais baixos, para os restantes diâmetros a melhor Heurística Construtiva Greedy é a HG2. Tal como acontecia nas instâncias de 20 nodos também aqui se verifica que a Heurística HG3 é a que apresenta gaps mais elevados. Em geral também se verifica que a Heurística OTTC obtém gaps inferiores a qualquer Heurística Construtiva Greedy, mas por exemplo, na instância tc30-1 a Heurística HG1 para diâmetros 6, 7 e 8, a Heurística HG2 para diâmetros 4, 6, 7 e 8 e a Heurística HG3 para o diâmetro 8 obtêm gaps inferiores à Heurística OTTC.

Nas Heurísticas de Aproximação houve uma redução significativa do valor da solução de referência (obtida através da Heurística HG3), por exemplo, em relação a um diâmetro $D = 5$ houve uma redução nos gaps de 76,34 % (122,96 % - 46,62 %), 84,5 % (122,96 % - 38,46 %), 85,86 % (122,96 % - 37,1 %) respectivamente para as Heurísticas HI, HJ e HIJ.

A Heurística Simples de Aproximação obtém valores dos gaps inferiores à Heurística OTTC para diâmetros 6 e 7. Na instância tc30-4 a Heurística HA obtém os melhores valores, ou seja, os gaps mais baixos para um diâmetro 6 e 8 em relação às restantes heurísticas.

Os gaps obtidos pela Heurística de Melhoramento em geral, são superiores aos obtidos pela Heurística OTTC, mas por exemplo, na instância tc30-1 para um diâmetro 8 é obtido um gap (37,3 %) que é inferior ao obtido pela Heurística OTTC (40,3%).

	D	**D=4**	**D=5**	**D=6**	**D=7**	**D=8**	**D=9**
	óptimo	739	663	590*	569	534	522
HG1	t	29,7	60,7	68,0	31,8	51,0	63,3
	v	1091	1783	1682	1107	1414	1582
	gap	47,6	168,9	185,1	94,6	164,8	203,1
HG2	t	30,5	37,4	29,2	36,8	27,4	35,1
	v	1156	1229	1001	1193	965	1108
	gap	56,4	85,4	69,1	109,7	80,7	112,3
HG3	t	16,1	10,7	13,2	9,1	6,9	4,5
	v	1355	963	1122	737	682	583
	gap	83,4	45,2	90,2	29,5	27,7	11,7
HM	t	69,2	69,1	74,9	77,5	84,4	87,2
	v	1444	1394	1256	1058	806	748
	gap	95,4	110,3	112,9	85,9	50,9	43,3
HA	t	56,0	58,7	31,5	55,4	14,7	29,7
	v	1659	1850	1065	1404	696	1041
	gap	124,5	179,0	80,5	146,7	30,3	99,4
HI	t	1446,4	1075,1	915,5	597,2	820,5	520,1
	v	**1063**	989	**727**	**717**	656	**570**
	gap	43,8	49,2	23,2	26,0	22,8	9,2
HJ	t	87,0	101,5	54,3	74,0	45,3	188,5
	v	1169	1253	882	757	**648**	740
	gap	58,2	89,0	49,5	33,0	21,3	41,8
HIJ	t	1818,2	1517,1	988,0	1120,1	542,1	758,1
	v	1091	**942**	899	753	709	649
	gap	47,6	42,1	52,4	32,3	32,8	24,3
OTTC	t	209,6	178,5	142,2	77,4	82,4	65,3
	v	1197	1048	890	744	758	690
	gap	62,0	58,1	50,8	30,8	41,9	32,2

Tabela 4.12: Exemplo ***tc40-1***: Resultados das heurísticas para o exemplo de 40 nodos, cuja árvore de suporte de custo mínimo tem diâmetro $D = 19$, custo 470 e tempo de execução 1,5 segundos.

	D	**D=4**	**D=5**	**D=6**	**D=7**	**D=8**	**D=9**
	óptimo	664	612	555	527	506*	495
HG1	t	52,2	53,9	36,2	43,3	25,4	21,9
	v	1686	1405	1148	1181	876	799
	gap	153,9	129,6	106,8	124,1	73,1	61,4
HG2	t	52,3	26,1	46,5	25,5	37,6	12,5
	v	1534	1001	1389	940	1117	633
	gap	131,0	63,6	150,3	78,4	120,8	27,9
HG3	t	47,1	58,9	40,3	43,0	24,6	24,2
	v	1295	1819	1107	1325	875	946
	gap	95,0	197,2	99,5	151,4	72,9	91,1
HM	t	65,2	66,2	69,4	72,3	77,0	77,7
	v	1425	1317	1217	1171	1042	1002
	gap	114,6	115,2	119,3	122,2	105,9	102,4
HA	t	51,1	52,4	36,0	36,4	29,2	28,1
	v	1332	1351	902	894	843	765
	gap	100,6	120,8	62,5	69,6	66,6	54,5
HI	t	2128,7	1250,3	1785,1	1123,7	1497,6	572,2
	v	1182	**915**	**879**	**705**	876	**568**
	gap	78,0	49,5	58,4	33,8	73,1	14,7
HJ	t	92,6	317,5	64,8	197,9	56,0	116,3
	v	**1068**	1613	887	1096	**672**	796
	gap	60,8	163,6	59,8	108,0	32,8	60,8
HIJ	t	1871,6	1609,6	1326,1	1084,1	640,5	515,6
	v	1088	1031	1004	822	740	620
	gap	63,9	68,5	80,9	56,0	46,2	25,3
OTTC	t	206,4	187,2	192,0	194,7	162,6	121,4
	v	1096	999	1114	1096	876	592
	gap	65,1	63,2	100,7	108,0	73,1	19,6

Tabela 4.13: Exemplo ***tc40-2***: Resultados das heurísticas para o exemplo de 40 nodos, cuja árvore de suporte de custo mínimo tem diâmetro $D = 22$, custo 450 e tempo de execução 2,5 segundos.

	D	**D=4**	**D=5**	**D=6**	**D=7**	**D=8**	**D=9**
	óptimo	713*	636	568*	545	524*	513*
HG1	t	55,7	37,5	37,8	37	41,1	36,9
	v	1462	1126	1149	1051	1176	1035
	gap	105,0	77,0	102,3	92.8	124,4	101,8
HG2	t	27,1	28,6	22,6	22,8	16,0	14,7
	v	**947**	1037	884	818	745	671
	gap	32,8	63,1	55,6	50,1	42,2	30,8
HG3	t	51,9	60,6	28,7	42,4	39,1	31,9
	v	1648	1806	993	1160	1053	911
	gap	131,1	184,0	74,8	112,8	101,0	77,6
HM	t	88,1	88,3	91,7	96,1	104,2	105,9
	v	1373	1320	1135	1056	957	894
	gap	92,6	107,5	99,8	93,8	82,6	74,3
HA	t	58,3	51,8	46,6	42,6	38,4	34,5
	v	1259	1138	1024	959	867	794
	gap	76,6	78,9	80,3	76,0	65,5	54,8
HI	t	1767,4	1236,2	1228,7	679,8	810,7	657,4
	v	1130	953	870	**689**	789	**611**
	gap	58,5	49,8	53,2	26,4	50,6	19,1
HJ	t	101,0	105,2	71,0	76,0	264,9	63,7
	v	1199	1138	1055	922	**722**	791
	gap	68,2	78,9	85,7	69,2	37,8	54,2
HIJ	t	1667,3	1620,7	857,9	771,2	849,6	685,8
	v	985	**884**	**716**	712	725	757
	gap	38,1	39,0	26,1	30,6	38,4	47,6
OTTC	t	183,0	184,9	151,4	160,2	147,4	125,9
	v	1066	990	924	843	796	791
	gap	49,5	57,7	62,7	54,7	51,9	54,2

Tabela 4.14: Exemplo ***tc40-3***: Resultados das heurísticas para o exemplo de 40 nodos, cuja árvore de suporte de custo mínimo tem diâmetro $D = 23$, custo 456 e tempo de execução 1,5 segundos.

	D	**D=4**	**D=5**	**D=6**	**D=7**	**D=8**	**D=9**
	óptimo	723	650	586*	571	536*	525*
HG1	t	61,5	49,0	48,0	37,2	42,4	22,1
	v	1763	1518	1370	1195	1303	855
	gap	143,8	133,5	133,8	109,3	143,1	62,9
HG2	t	43,8	33,2	43,7	28,3	42,0	28,3
	v	1516	1292	1440	1029	1324	993
	gap	109,7	98,8	145,7	80,2	147,0	89,1
HG3	t	48,1	42,1	42,6	25,3	25,6	19,8
	v	1567	1667	1215	1127	915	878
	gap	116,7	156,5	107,3	97,4	70,7	67,2
HM	t	84,9	86,7	91,8	105,2	107,5	117,8
	v	1539	1432	1246	947	884	744
	gap	112,9	120,3	112,6	65,8	64,9	41,7
HA	t	41,9	41,9	48,1	41,8	25,7	22,9
	v	1226	1176	1344	1101	**816**	752
	gap	69,6	80,9	129,4	92,8	52,2	43,2
HI	t	1314,4	897,7	1776,6	1343,9	1094,6	867,8
	v	1366	**944**	990	**783**	858	833
	gap	88,9	45,2	68,9	37,1	60,1	58,7
HJ	t	113,4	89,7	106,9	65,9	50,1	48,9
	v	1510	1402	1731	1195	827	825
	gap	108,9	115,7	195,4	109,3	54,3	57,1
HIJ	t	1515,2	1498,7	1321,8	960,1	985,6	703,4
	v	**1126**	976	**958**	847	899	861
	gap	55,7	50,2	63,5	48,3	67,7	64,0
OTTC	t	228,1	219,7	211,2	200,5	205,0	139,5
	v	**1126**	1050	960	961	1072	**697**
	gap	55,7	61,5	63,8	68,3	100,0	32,8

Tabela 4.15: Exemplo ***tc40-4***: Resultados das heurísticas para o exemplo de 40 nodos, árvore de suporte de custo mínimo tem diâmetro $D = 22$, custo 472 e tempo de execução 1,5 segundos.

	D	**D=4**	**D=5**	**D=6**	**D=7**	**D=8**	**D=9**
	óptimo	705	649	577*	567	535	526
HG1	t	61,7	60,0	57,3	39,1	45,7	31,5
	v	1807	1747	1641	1167	1205	1035
	gap	156,3	169,2	184,4	105,8	125,2	96,8
HG2	t	41,7	33,9	38,0	37,4	29,9	29,9
	v	1236	1125	1170	1167	1067	1035
	gap	75,3	73,3	102,8	105,8	99,4	96,8
HG3	t	67,6	62,4	42,5	45,7	23,4	26,1
	v	1700	1453	1385	1299	894	983
	gap	141,1	123,9	140,0	129,1	67,1	86,9
HM	t	63,1	63,8	67,5	70,1	83,1	92,2
	v	1414	1371	1215	1147	805	701
	gap	100,6	111,2	110,6	102,3	50,5	33,3
HA	t	60,3	44,1	43,3	34,1	30,5	26,7
	v	**1062**	1180	1143	875	889	748
	gap	50,6	81,8	98,1	54,3	66,2	42,2
HI	t	1770,7	470,5	731,3	646,8	462,3	586,5
	v	1287	**715**	**779**	**728**	**715**	**609**
	gap	82,6	10,2	35,0	28,4	33,6	15,8
HJ	t	99,9	100,7	73,1	63,9	61,3	52,1
	v	1065	1079	828	798	832	707
	gap	51,1	66,3	43,5	40,7	55,5	34,4
HIJ	t	1688,3	792,7	478,9	383,2	250,2	261,9
	v	1236	1070	1164	1167	1099	983
	gap	75,3	64,9	101,7	105,8	105,4	86,9
OTTC	t	224,8	198,9	175,4	221,5	190,6	138,0
	v	1236	995	1072	1177	1067	751
	gap	75,3	53,3	85,8	107,6	99,4	42,8

Tabela 4.16: Exemplo ***tc40-5***: Resultados das heurísticas para o exemplo de 40 nodos, árvore de suporte de custo mínimo tem diâmetro $D = 22$, custo 480 e tempo de execução 1,8 segundos.

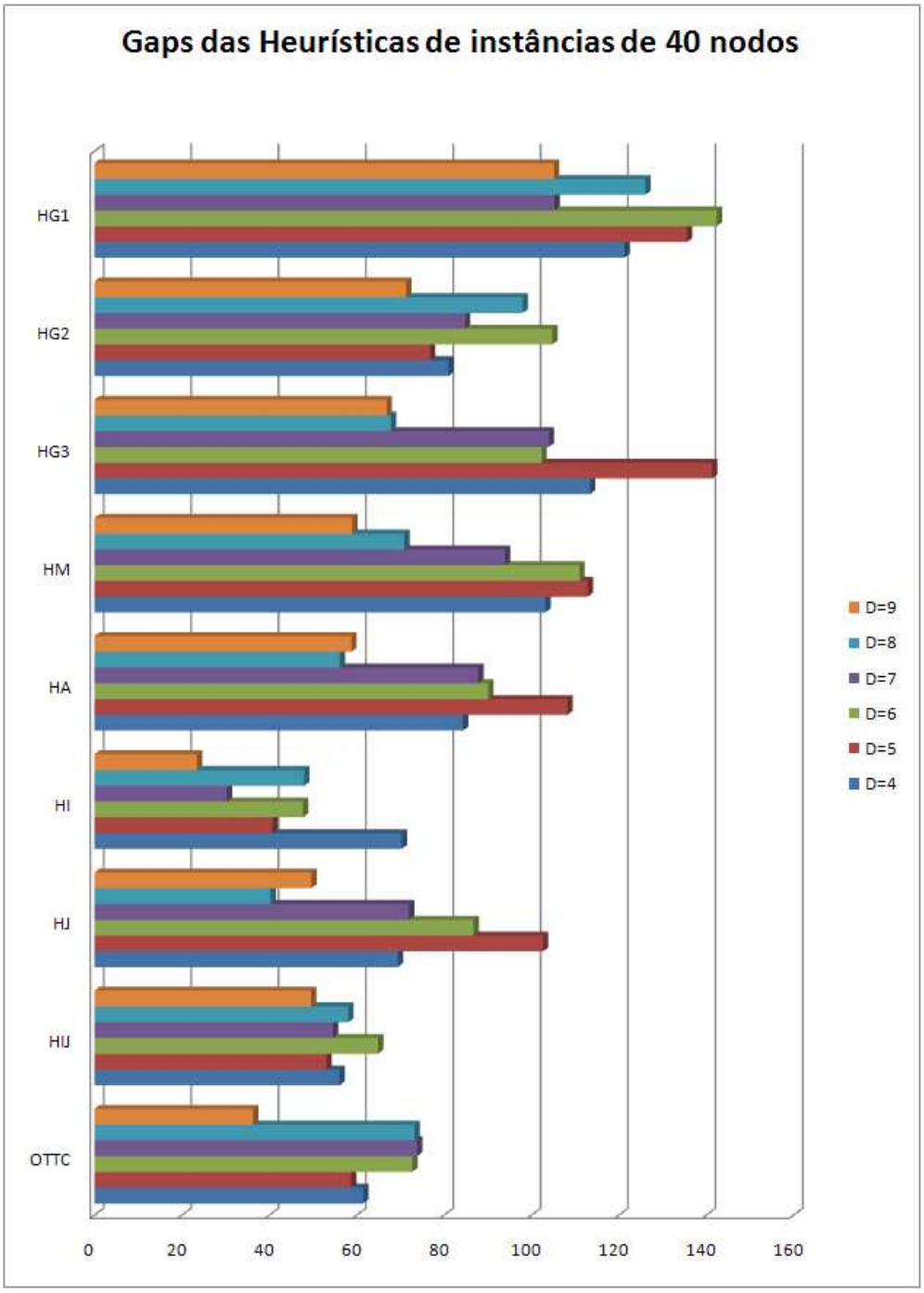


Figura 4.3: Gráfico relativo aos gaps das instâncias de 40 nodos.

Por observação do gráfico da Figura 4.3 podemos concluir que, em geral para instâncias de 40 nodos é a Heurística de Inibição a que apresenta gaps mais baixos e a Heurística Greedy 1 a que apresenta gaps mais elevados. Para diâmetros 5, 6, 7 e 9 é a Heurística de Inibição que apresenta gaps mais baixos, para o diâmetro 4 é a Heurística de Inibição/Junção a que apresenta gaps mais baixos e para o diâmetro 8 é a heurística de Junção que apresenta gaps mais baixos.

Em relação às três Heurísticas de Aproximação HI, HJ e HIJ continuam, tal como, para as instâncias de 20 e 30 nodos a obterem os valores mais baixos para os gaps. Há sempre uma destas Heurísticas a obter gaps mais baixos do que a Heurística OTTC.

Ao efectuarmos a comparação dos gaps das três Heurísticas Construtivas Greedy, observamos que para diâmetros 4, 5 e 7 é a Heurística Greedy 2 que obtém gaps inferiores e para diâmetros 6, 8 e 9 desta vez, é a Heurística Greedy 3 que se realça. Para as instâncias de 40 nodos ao contrário do que acontecia com as intâncias de 20 e 30 é a Heurística Greedy 1 aquela que obtém os gaps mais altos. Em geral também se observa que a Heurística OTTC em média obtém valores dos gaps muito mais baixos do que qualquer Heurística Construtiva Greedy. Podemos observar que, em particular existem alguns casos, como é o caso da instância tc40-3 onde para um diâmetro 4 é a Heurística HG2 que obtém o melhor valor (947), ou seja, o gap mais baixo (32,8 %). Na instância tc40-5 para um diâmetro 8 é a Heurística HG3 a obter um gap inferior à Heurística OTTC e na instância tc40-1 para um diâmetro 4 é a Heurística Greedy 1 a obter um gap melhor que a Heurística OTTC.

As reduções obtidas nos gaps usando as três Heurísticas de Aproximação HI, HJ e HIJ comparativamente com a solução de referência (HG3) utilizada são respectivamente para um diâmetro $D = 5$; 100,58 % $(141,36\% - 40,78\%)$, 38,66 % $(141,36\% - 102,7\%)$ e 88,22 % $(141,36\% - 53,14\%)$.

A Heurística Simples de Aproximação e a Heurística de Melhoramento obtém gaps inferiores à Heurística OTTC para um diâmetro 8. Com a Heurística Simples de Aproximação consegue-se obter a melhor solução para a instância tc40-4 e para um diâmetro 8 e também na instância tc40-5 para um diâmetro 4.

	D	**D=4**	**D=5**	**D=6**	**D=7**	**D=8**	**D=9**
	óptimo	1062*	956*	846*	809*	766°	747°
HG1	t	375,2	405,5	158,6	240,2	304,0	154,0
	v	2518	3089	1647	1827	2251	1372
	gap	137,1	223,1	94,7	125,8	193,9	83,7
HG2	t	144,9	137,3	132,5	96,6	98,1	92,2
	v	1788	1708	1605	1284	1247	1160
	gap	68,4	78,7	89,7	58,7	62,8	55,3
HG3	t	358,3	355,5	326,2	340,2	146,1	289,9
	v	3103	3126	2389	2513	1483	2137
	gap	192,2	227,0	182,4	210,6	93,6	186,1
HM	t	422,6	428,6	432,9	463,7	469,6	496,3
	v	2394	2332	2218	1832	1788	1506
	gap	125,4	143,9	162,2	126,5	133,4	101,6
HA	t	408,5	1772,6	348,8	304,0	278,3	237,4
	v	2612	3479	2266	1932	1802	1530
	gap	146,0	263,9	167,8	138,8	135,2	104,8
HI	t	14482,7	13423,3	8155,0	6569,1	6600,9	4996,7
	v	2053	1727	1304	1296	1188	1021
	gap	93,3	80,6	54,1	60,2	55,1	36,7
HJ	t	552	677,4	528,6	421,4	266,9	220,6
	v	1968	1994	1567	1372	1198	**1017**
	gap	85,3	108,6	85,2	69,6	56,4	36,1
HIJ	t	16624,4	14504,0	11255,8	10552,8	8970,3	7721,5
	v	**1524**	**1530**	**1236**	**1261**	**1127**	1097
	gap	43,5	60,0	46,1	55,9	47,1	46,9
OTTC	t	1250,7	976,8	1016,5	848,0	579,8	635,1
	v	2053	1825	1705	1402	1250	1122
	gap	93,3	90,9	101,5	73,3	63,2	50,2

Tabela 4.17: Exemplo ***tc60-1***: Resultados das heurísticas para o exemplo de 60 nodos, árvore de suporte de custo mínimo tem diâmetro $D = 29$, custo 662 e tempo de execução 4,4 segundos.

	D	**D=4**	**D=5**	**D=6**	**D=7**	**D=8**	**D=9**
	óptimo	1049*	943*	832*	789*	752*	784°
HG1	t	412,5	248,3	308,6	304,6	225,0	309,0
	v	2995	2291	2633	2105	1785	2062
	gap	185,5	142,9	216,5	166,8	137,4	180,9
HG2	t	233,2	119,8	295,3	138,5	134,0	118,0
	v	2140	1788	2120	1710	1447	1482
	gap	104,0	89,6	154,8	116,7	92,4	101,9
HG3	t	379,5	384,4	384,4	355,2	370,2	287,1
	v	3676	3537	3735	2869	3253	2243
	gap	250,4	275,1	348,9	263,6	332,6	205,6
HM	t	467,4	468,6	471,8	479,9	490,2	507,0
	v	2482	2418	2356	2238	2072	1824
	gap	136,6	156,4	183,2	183,7	175,5	148,5
HA	t	372,1	331,7	305,8	336,8	221,2	200,6
	v	2578	2248	1977	2301	1563	1546
	gap	145,8	138,4	137,6	191,6	107,8	110,6
HI	t	14778,4	13171,4	14272,7	13997,9	8514,2	7815,7
	v	2263	**1784**	2049	1741	1443	1319
	gap	115,7	89,2	146,3	120,7	91,9	79,7
HJ	t	607,8	650,6	528,6	549,7	512,5	382,5
	v	1996	1893	**1332**	1651	1390	1251
	gap	90,3	100,7	60,1	109,3	84,8	70,4
HIJ	t	21465,5	19780,3	15842,8	13981,2	12397,2	10876,7
	v	**1960**	2125	1532	**1528**	**1019**	**1018**
	gap	86,8	125,3	84,1	93,7	35,5	38,7
OTTC	t	1269,9	1334,1	1206,3	1134,9	1069,2	943,6
	v	2263	2125	1960	1810	1447	1255
	gap	115,7	125,3	135,6	129,4	92,4	71,0

Tabela 4.18: Exemplo ***tc60-2***: Resultados das heurísticas para o exemplo de 60 nodos, árvore de suporte de custo mínimo tem diâmetro $D = 34$, custo 638 e tempo de execução 4,8 segundos.

	D	**D=4**	**D=5**	**D=6**	**D=7**	**D=8**	**D=9**
	óptimo	1082*	993*	867*	832*	791*	772*
HG1	t	171,9	380,2	243,9	326,6	270,3	213,4
	v	**1914**	3448	2077	2264	2209	1828
	gap	76,9	247,2	139,6	172,1	179,3	136,8
HG2	t	279,7	132,4	203,6	119,2	106,8	58,1
	v	2336	1784	2099	1520	1270	**1013**
	gap	115,9	79,7	142,1	82,7	60,6	31,2
HG3	t	361,9	381,9	353,7	339,1	197,2	247,7
	v	2796	3424	2016	2644	1850	2065
	gap	158,4	244,8	132,5	217,8	133,9	167,5
HM	t	319,2	324,2	331,4	335,0	368,1	397,1
	v	2481	2419	2295	2239	1931	1655
	gap	129,3	143,6	164,7	169,1	144,1	114,4
HA	t	420,6	434,0	331,9	276,1	257,8	151,0
	v	2702	2603	2380	2126	1991	1332
	gap	149,7	162,1	174,5	155,5	151,7	72,5
HI	t	16591,8	8607,7	8970,3	7785,1	9177,0	5821,3
	v	2141	**1704**	1701	**1305**	1285	1107
	gap	97,9	71,6	96,2	56,9	62,5	43,4
HJ	t	696,7	634,2	539,1	1652,2	273,7	359,8
	v	1986	1909	1907	2264	1338	1341
	gap	83,5	92,2	120,0	172,1	69,2	73,7
HIJ	t	21064,5	18610,1	14832,5	11558,3	9527,1	6777,1
	v	**1914**	1868	**1484**	1379	**1243**	**1044**
	gap	76,9	88,1	71,2	65,7	57,1	35,2
OTTC	t	1280	1225,4	1061,4	954,4	1006,4	649,5
	v	**1914**	2040	1843	1577	1460	1178
	gap	76,9	105,4	112,6	89,5	84,6	52,6

Tabela 4.19: Exemplo ***tc60-3***: Resultados das heurísticas para o exemplo de 60 nodos, árvore de suporte de custo mínimo tem diâmetro $D = 28$, custo 682 e tempo de execução 4,2 segundos.

	D	**D=4**	**D=5**	**D=6**	**D=7**	**D=8**	**D=9**
	óptimo	1053*	953*	838*	805*	762*	733°
HG1	t	502,5	327,8	208,4	235,4	304,7	255,1
	v	3336	2327	1850	1846	2039	1878
	gap	216,8	144,2	120,8	129,3	167,6	156,2
HG2	t	224,8	220,4	144,8	163,7	139,7	94,8
	v	2106	2005	1578	1607	1459	1232
	gap	100,0	110,4	88,3	99,6	91,5	68,1
HG3	t	399,7	400,4	421,7	314,1	292,1	253,2
	v	3954	3490	2935	2544	2181	1809
	gap	275,5	266,2	250,2	216,0	186,2	146,8
HM	t	524,1	527,1	533,2	543,8	570,5	585,0
	v	2356	2294	2186	2032	1847	1678
	gap	123,7	140,7	160,9	152,4	142,4	128,9
HA	t	273,9	269,2	253,3	123,4	182,2	179,0
	v	2121	2005	1887	2163	1478	1450
	gap	101,4	110,4	125,2	168,7	94,0	97,8
HI	t	13584,2	10099,6	8464,5	11181,2	12134,1	8086,4
	v	**1693**	**1784**	1405	1476	1361	1233
	gap	60,8	87,2	67,7	83,4	78,6	68,2
HJ	t	628,6	652,1	559,3	458,3	400,5	361,3
	v	1808	1868	**1356**	1449	1298	1223
	gap	71,7	96,0	61,8	80,0	70,3	66,8
HIJ	t	18704,1	16862,4	14032,4	12529,8	10642,0	6374,8
	v	1856	1791	1465	**1366**	**1150**	1055
	gap	76,3	87,9	74,8	69,7	50,9	43,9
OTTC	t	1119,9	1268,6	984,6	964,3	876,8	707,6
	v	1753	1962	1578	1519	1465	**1054**
	gap	66,5	105,9	88,3	88,7	92,3	43,8

Tabela 4.20: Exemplo ***tc60-4***: Resultados das heurísticas para o exemplo de 60 nodos, árvore de suporte de custo mínimo tem diâmetro $D = 25$, custo 636 e tempo de execução 5,0 segundos.

	D	**D=4**	**D=5**	**D=6**	**D=7**	**D=8**	**D=9**
	óptimo	1258*	1123*	805*	935*	744°	935*
HG1	t	418,9	454,1	233,9	206,2	233,2	209,3
	v	3826	4077	2200	2230	2186	2097
	gap	204,1	263,0	173,3	200,1	193,8	124,3
HG2	t	217,6	207,0	200,5	198,4	219,2	196,9
	v	2357	2217	2119	2107	2194	2101
	gap	87,4	97,4	163,2	183,6	194,9	124,7
HG3	t	467,0	480,0	407,2	374,5	369,8	351,3
	v	4246	3802	3406	3064	2912	2757
	gap	237,5	238,6	323,1	312,4	291,4	194,9
HM	t	455,1	458,1	462,8	488,6	491,4	507,3
	v	2854	2788	2660	2325	2273	2171
	gap	126,9	148,3	230,4	212,9	205,5	132,2
HA	t	345,7	334,4	334,0	330,1	326,7	162,3
	v	2975	2737	2693	2695	2645	1562
	gap	136,5	143,7	234,5	262,7	255,5	67,1
HI	t	11432,0	10738,0	11877,3	7378,3	10826,6	10602,7
	v	2275	**1765**	2069	1530	1614	1305
	gap	80,8	57,2	157,0	105,9	116,9	39,6
HJ	t	714,3	810,7	7254,3	486,1	5840,3	442,1
	v	2386	2911	**1641**	**1383**	**1203**	**1264**
	gap	89,7	159,2	103,9	86,1	61,7	35,2
HIJ	t	19904,4	19049,1	15642,4	12345,0	11929,0	10642,4
	v	2357	2298	1785	1685	1769	1744
	gap	87,4	104,6	121,7	126,8	137,8	86,5
OTTC	t	1163,5	1181,6	1205,0	1259,4	1116,7	1137,5
	v	**2256**	2368	2070	2022	2027	1963
	gap	79,3	110,9	157,1	172,1	172,4	109,9

Tabela 4.21: Exemplo ***tc60-5***: Resultados das heurísticas para o exemplo de 60 nodos, árvore de suporte de custo mínimo tem diâmetro $D = 33$, custo 740 e tempo de execução 4,3 segundos.

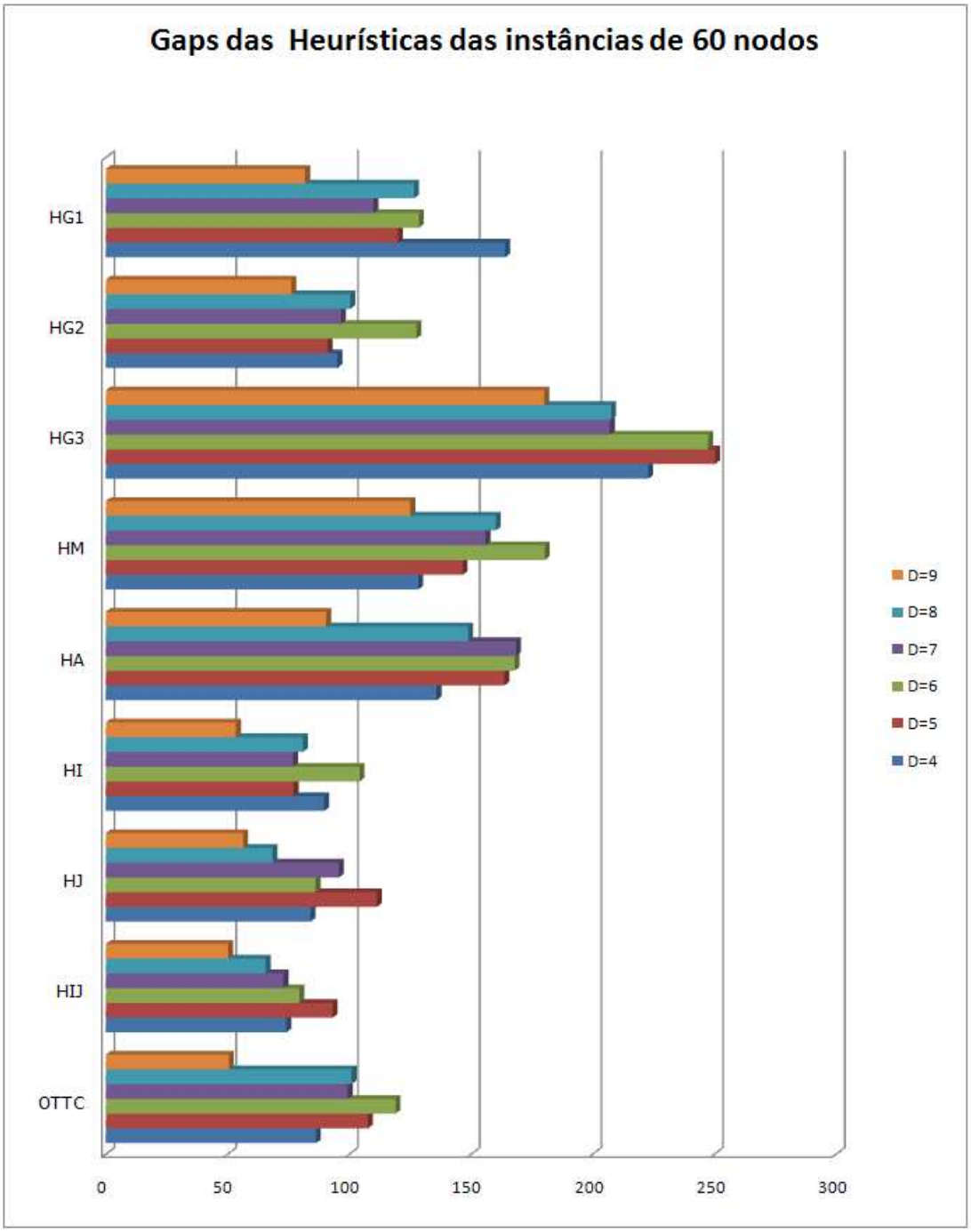


Figura 4.4: Gráfico relativo aos gaps das instâncias de 60 nodos.

Por observação do gráfico da Figura 4.4 podemos concluir que, em geral para instâncias de 60 nodos é a Heurística de Inibição e Junção a que apresenta gaps mais baixos e a Heurística Greedy 3 a que apresenta gaps mais elevados. Para diâmetros 4, 6, 7, 8 e 9 é a Heurística de Inibição/Junção que apresenta gaps mais baixos, para um diâmetro 5 é a Heurística de Inibição a que apresenta gaps mais baixos.

Em relação às três Heurísticas de Aproximação HI, HJ e HIJ continuam, tal como, para as instâncias de 20, 30 e 40 nodos a obterem os valores mais baixos para os gaps. Há sempre uma destas Heurísticas a obter gaps mais baixos do que a Heurística OTTC.

Nas instâncias de 60 nodos a Heurística Simples de Aproximação e a Heurística de Melhoramento apresentam gaps mais ou menos idênticos.

Ao efectuarmos a comparação dos gaps das três Heurísticas Construtivas Greedy, observamos que a Heurística Greedy 2 é a que obtém gaps mais baixos. A Heurística Greedy que apresenta os gaps mais elevados é a Heurística Greedy 3. Para diâmetros 5, 7 e 8 a Heurística HG2 obtém gaps inferiores aos obtidos pela Heurística OTTC. Podemos também observar que existem apenas dois casos, como é o caso da instância tc60-1 onde para um diâmetro 6 é a Heurística HG1 a obter um gap inferior ao da Heurística OTTC e o caso da instância tc60-3 onde para um diâmetro 4 o valor obtido pela Heurística HG2 é o mesmo que o obtido pela Heurística OTTC. Em relação à Heurística Greedy 3 nas tabelas não se observam valores inferiores aos obtidos pela Heurística OTTC.

As reduções obtidas nos gaps usando as três Heurísticas de Aproximação HI, HJ e HIJ comparativamente com a solução de referência (HG3) utilizada são respectivamente para um diâmetro $D = 7$; 130,18 % (207,14 % - 76,96%), 111,36 % (207,14 % - 95,78 %) e 134,1 % (207,14 % - 73,04 %).

Os gaps obtidos pelas Heurísticas de Melhoramento e Simples de Aproximação em geral são superiores aos obtidos pela Heurística OTTC, apenas na instância tc60-5 para um diâmetro 9 a Heurística Simples de Aproximação consegue obter um gap 67,1 % inferior ao obtido pela OTTC.

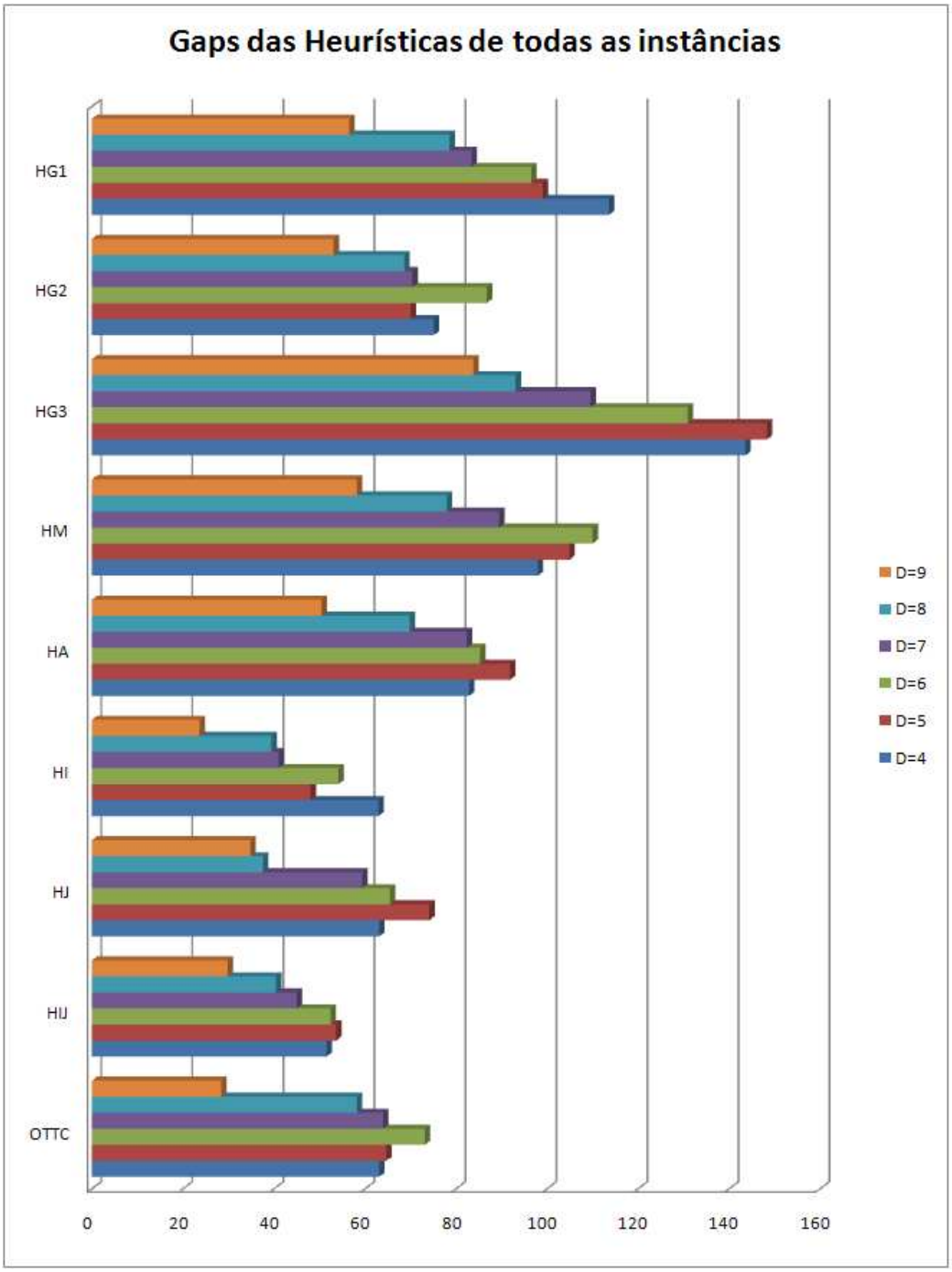


Figura 4.5: Gráfico relativo aos gaps de todas as instâncias.

Por observação do gráfico da Figura 4.5 relativo aos gaps das Heurísticas de todas as instâncias podemos observar que, em geral são as Heurísticas HI, HJ e HIJ que obtêm os gaps mais baixos e que a Heurística HG3 é a que obtém os gaps mais elevados.

A solução da Heurística Greedy 1 depende do nodo que é escolhido para iniciar a construção da árvore, ou seja, a escolha do nodo inicial influencia o valor da solução final. O tempo de execução em geral vai diminuindo à medida que o número de nodos do grafo aumenta. Para um diâmetro igual ao diâmetro da árvore de suporte de custo mínimo obtém-se as mesmas soluções que as obtidas pelo algoritmo de Prim.

A solução da Heurística Greedy 2 também vai depender do nodo (aresta) escolhido(a) para iniciar a construção da árvore, pois admitindo que o diâmetro pretendido é ímpar a escolha do elemento inicial é feita pelos dois elementos com peso mínimo (menor soma dos custos das arestas incidentes), ou seja é escolhida uma aresta para iniciar a construção da árvore e a escolha dessa aresta pode não ser a que tem custo mínimo, este facto pode levar à construção de uma árvore de suporte que não corresponde à árvore de suporte de custo mínimo.

Na Heurística Greedy 3 obtém-se soluções com custos bastante elevados comparativamente com as duas heurísticas Greedy anteriores, este facto deve-se à escolha inicial do nodo ou da aresta, pois esta escolha é feita pelo nodo com maior peso no caso do diâmetro ser par e pelos dois nodos cujo peso é maior, ou seja, por uma aresta, no caso do diâmetro ser ímpar.

As Heurísticas OTTC e Greedy 1 diferem apenas no número de árvores construídas. A Heurística Greedy 1 constrói apenas uma árvore e a construção dessa árvore inicia-se num nodo arbitrário. A Heurística OTTC constrói q árvores e escolhe-se para solução final a árvore de menor custo obtida. O processo utilizado na Heurística OTTC por um lado é vantajoso, porque se escolhe de entre um conjunto de árvores construídas a árvore de menor custo obtida. Contudo, com este procedimento, o tempo de execução aumenta e nem sempre a Heurística OTTC obtém soluções de custo inferior às obtidas pela Heurística Greedy 1.

Quanto às Heurísticas construtivas é muito importante a escolha do nodo inicial, pois a escolha vai influenciar a solução final. Notemos que nas Heurísticas Construtivas greedy para um diâmetro $D = 2$, nada nos garante que o nodo inicialmente escolhido seja o nodo

central e pode acontecer que nenhuma das soluções obtidas corresponda à solução óptima (na Heurística OTTC no exemplo tc20-1, os valores obtidos nas 20 árvores são: 680, 741, 712, 766, 705, 923, 840, 825, 856 e 1180 e o valor óptimo é 633). Isto acontece porque estamos a construir grafos estrelas e não obrigamos os nodos inicialmente escolhidos a serem o nodo central.

Ao contrário das heurísticas anteriores o tempo de execução da Heurística de Melhoramento aumenta à medida que aumenta o diâmetro da árvore, pois quanto maior for o diâmetro da árvore mais arestas vão ser examinadas para possíveis trocas, uma vez que o algoritmo pára assim que for obtido o diâmetro pretendido.

Quando pretendemos determinar uma árvore cujo valor do diâmetro seja igual ao da árvore de suporte de custo mínimo, obtém-se valores superiores aos valores da árvore de suporte de custo mínimo, isto porque o número de trocas admissíveis não consegue transformar a árvore inicial numa árvore com o mesmo diâmetro da árvore de suporte de custo mínimo.

O número de iterações, que corresponde ao número de trocas possíveis, aumenta à medida que aumenta o tamanho da rede, ou seja, para uma rede com muitos nodos existe um maior número de trocas possíveis. Note-se que nem todas essas trocas possíveis são consideradas trocas admissíveis, uma vez que, podem vir a aumentar o custo da rede.

A Heurística Simples de Aproximação não tem um tempo de execução muito elevado. Numa primeira fase determina a árvore de suporte de custo mínimo. Depois efectua a eliminação das arestas da árvore de suporte de custo mínimo de modo a obter-se uma subárvore e, finalmente, efectua a construção da nova árvore a partir da subárvore obtida.

As Heurísticas de Inibição, Junção e Inibição/Junção são baseadas no método Pilot ([24] e [25]), este método pode ser visto como um sistema de repetição para cada escolha possível.

Para as Heurísticas de Inibição, Junção e Inibição/Junção utiliza-se como solução admissível inicial a solução obtida pela Heurística Greedy 3. A escolha desta heurística para obtenção das soluções iniciais deve-se ao facto de com ela se terem obtido soluções com valores elevados. Observa-se nitidamente que em geral as soluções obtidas por estas heurísticas reduzem consideravelmente o custo das soluções obtidas pela Heurística Greedy 3.

Para a Heurística de Inibição no "Passo 0" foi também experimentada uma outra forma de escolher o nodo t. Em vez de escolhermos o nodo de maior grau da árvore de suporte de custo mínimo, escolhemos o nodo central da solução admissível inicial no caso do diâmetro ser par e a aresta central da solução admissível inicial no caso do diâmetro ser ímpar. Foram

efectuados alguns testes e verificou-se que os valores obtidos seriam ou os mesmos ou mais elevados do que a outra forma descrita de escolha do nodo t.

A escolha do nodo t nas Heurísticas de Aproximação é muito importante, pois dela depende a obtenção de boas ou más soluções.

Em relação à Heurística de Inibição e Junção muitas das vezes o conjunto E_{adc} torna-se grande e ao eliminarem-se todas as arestas deste conjunto poderá acontecer que não seja possível formar uma árvore. Quando isto acontece vão-se eliminando sucessivamente as arestas deste conjunto até se formar uma árvore.

A escolha dos conjuntos E_{adc} e O_{adc} é muito importante, pois dela vai depender a obtenção de boas ou más soluções.

Do ponto de vista global são as heurísticas baseadas em procedimentos de Junção e Inibição que apresentam os melhores resultados, obtendo-se com estas heurísticas valores inferiores aos obtidos com a Heurística OTTC. Como se pode observar através das tabelas apresentadas em anexo a Heurística OTTC só apresenta melhores resultados comparativamente com as Heurísticas HI, HJ e HIJ nos exemplos tc60-5 para um diâmetro 4, para o exemplo tc60-4 para um diâmetro 9 e para o exemplo tc40-4 para um diâmetro 9.

Em alguns exemplos as Heurísticas Construtivas Greedy 1, 2 e 3 apresentam melhores valores que os obtidos pela Heurística OTTC como é o caso do exemplo tc20-1 onde para diâmetros 6, 7 e 8 respectivamente a Heurística Greedy 3 apresenta valores inferiores aos obtidos pela Heurística OTTC para os mesmos diâmetros.

No exemplo tc20-4 obtém-se para a Heurística Greedy 1 valores tão bons quanto os obtidos pela Heurística OTTC para diâmetros superiores a 4.

A Heurística de Melhoramento apresenta os melhores resultados para o exemplo tc20-2 e para diâmetros 7 e 9, embora neste exemplo os valores encontrados para diâmetros 6 e 8 são inferiores aos obtidos pela Heurística OTTC, sendo de notar que no exemplo tc20-1 também se obtém um bom resultado com esta Heurística para um diâmetro 9.

A Heurística Simples de Aproximação obtém um dos melhores resultados no exemplo tc20-2 para um diâmetro 8. Em relação ao exemplo tc20-3 para diâmetros 6 e 8 obtém-se os melhores resultados, para $D = 7$ obtém-se um resultado inferior à OTTC e para $D = 9$ o resultado obtido é o mesmo que o obtido pela Heurística de Inibição, pela Heurística de Inibição/Junção e pela Heurística OTTC.

Em relação às três Heurísticas que se baseiam na proibição de um dado conjunto de arestas de pertencer à solução, a Heurística de Inibição e a Heurística de Inibição/Junção são as que mais se realçam na obtenção dos melhores resultados.

Capítulo 5

Considerações finais

Ao longo desta tese apresentaram-se heurísticas que produzem valores aproximados para o problema DMST. Começou-se por apresentar Heurísticas Construtivas, depois Heurísticas de Melhoramento e depois Heurísticas de Aproximação e no final fizemos uma breve abordagem a duas heurísticas conhecidas na literatura, a Heurística OTTC e a Heurística CIR.

As Heurísticas Construtivas Greedy apresentadas nesta tese diferem na escolha dos elementos iniciais (nodo ou nodos), ou seja, diferem na forma de iniciar a construção da árvore. Na Heurística Greedy 1 inicia-se sempre a construção da árvore apenas com um nodo que é escolhido de forma aleatória. Nas Heurísticas Greedy 2 e 3 a construção da árvore inicia-se com um nodo se D for par ou inicia-se com uma aresta se D for ímpar. A escolha dos elementos iniciais é muito importante para a obtenção de boas soluções.

A Heurística de Melhoramento é uma heurística de trocas locais e esta tenta melhorar uma solução já conhecida.

As Heurísticas de Aproximação adaptam soluções de outro problema ao problema em questão. A Heurística Simples de Aproximação numa primeira fase elimina as arestas da árvore de suporte até se obter uma subárvore de acordo com regras estabelecidas e numa segunda fase constrói a nova árvore a partir da subárvore obtida de modo a obter-se uma árvore com o diâmetro pretendido.

As Heurísticas de Inibição e Junção diferem no facto de a primeira obrigar e a segunda proibir cada uma das arestas de um dado conjunto de estarem presentes na solução. A Heurística de Inibição/Junção usa as duas estratégias em conjunto.

A escolha do nodo t (nodo pelo qual se obtém a sequência das arestas na solução de referência não ligadas a t) nas Heurísticas de Aproximação que usam as estratégias de Inibição e Junção é também muito importante para a obtenção de soluções de boa qualidade.

Os resultados computacionais mostraram que as Heurísticas de Inibição, Junção

e Inibição/Junção são as que apresentam os melhores valores (gaps mais baixos) mesmo quando comparadas com a Heurística OTTC. Os valores obtidos através destas Heurísticas são melhores que os obtidos pela Heurística Greedy 3 que foi usada como solução de referência nestas Heurísticas.

Com a obtenção dos resultados computacionais e ao observarmos os valores encontrados, juntamente com algumas características observadas nas heurísticas concluímos que podemos fazer melhoramentos em algumas das heurísticas apresentadas com o objectivo de melhorar os valores obtidos pelas heurísticas.

Bibliografia

[1] A. Abdalla, N. Deo (2000), *Computing a diameter-constrained minimum spanning tree in parallel*, Lecture Notes in Computer Sciense 1767, 17-31.

[2] A. Abdalla, N. Deo (2002), *Random-Tree diameter and diameter-constrained MST*, Intern. J. Computer Math., vol. 79(6), 651-663.

[3] A. Abdalla, N. Deo e R. Fraceschini(1999), *Parallel heuristics for diameter-constrained minimum spanning tree problems*, Congress Num 136, 97-118.

[4] A. Abdalla, N. Deo, N. Kumar e T. Terry (1997), *Parallel computation of a diameter-constrained MST and related problems*, Congressus Numerantium 126, 131-155.

[5] R.K.Ahuja, T.L.Magnanti e J.B. Orlin (1993), *Network Flows, Theory, Algorithms and Applications*, Prentice-Hall, London.

[6] R. Balakrishnau e K. Ranganathan (2000), *A textbook of graph theory*, Springer-Verlag, NewYork.

[7] L.R. Esau e K.C. Williams (1966), *A method for approximating the optimal network*, IBM Syst. J., Vol. 5, 142.

[8] M. Garey e D. Johnson (1979), *Computers and intractability: A guide to the theory of NP-completeness*, Freeman, San Francisco.

[9] B. Gavish and K. Altinkemer(1986), *Parallel savings heuristics for the topological design of local access tree networks*, Proceedings of the IEEE INFOCOM'86 Conference, IEEE Communications Society, 130-139.

[10] L. Gouveia (1996), *Multicommodity flow models for spanning trees with hop constraints*, European Journal of Operational Research 95(3), 178-190.

[11] L. Gouveia e T.L. Magnanti (2003), *Network Flow Models for Designing Diameter-Constrained Minimum-Spanning and Steiner Trees*, Networks 41(3), 159-173.

[12] L. Gouveia e T.L. Magnanti e C. Requejo(2004), *A 2-path approach for odd-diameter-constrained minimum spanning and Steiner trees*, Networks 44 (4), 254-265.

[13] L. Gouveia e T.L. Magnanti e C. Requejo(2006), *An intersecting tree model for odd-diameter-constrained minimum spanning and Steiner trees*, Ann Oper Res, 146, 19-39.

[14] T. Magnanti e L. Wolsey (1995), *Optimal Trees in Network Models*, Handbooks in Operations Research and Management Science, Vol. 7, Elsevier Science Publishers, North-Holland 503-615

[15] J. Martin (1967), *Design of Real-Time Computer Systems*, Englewood Cliffs, New Jersey, Prentice Hall, 534-535.

[16] G.L. Nemhauser e L. A. Wolsey (1988), *Integer and Combinatorial Optimization*, Wiley.

[17] M. Karnaugh (1976), *A New Class of Algorithms for Multipoint Network Optimization*, Vol. 24, IEEE Transactions on communications 500-505.

[18] A. Kershenbaum, R. Boorstyn e R. GOppenheim(1980), *Second-Order Greedy Algorithms for Centralized Teleprocessing Network Design*, IEEE Trans. Communication, 28, 10, 1835-1838.

[19] A. Kershenbaum e W. Chou (1974), *A unified algorithm for designing multidrop teleprocessing networks*, IEEE Trans. Commun., vol. COM-22, 1762-1772.

[20] R. Patterson, H. Pirkul e E. Rolland (1999), *A Memory Adaptive Reasoning Technique for Solving the Capacitated Minimum Spanning Tree Problem*, Journal of Heuristics, 5, 159-180.

[21] R. Prim (1957), *Shortest connection networks and some generalization*, Bell Systems Technical Journal, 36, 1389-1401.

[22] G. R. Raidl e B. A. Julstrom (2003), *Greedy heuristics and an evolutionary algorithm for the bounded-diameter minimum spanning tree problem*, In G. Lamont

et al. editors, Proceedings of the 2003 ACM Symposium on Applied Computing, pages 747-752. New york, ACM Press.

[23] C. R. Reeves (ed.) (1995), *Modern Heurístic Techniques for Combinatorial Problems*, McGraw Hill, London.

[24] S. Voss e C. Duin (1999), *The Pilot Method: A Strategy for Heuristic Repetition with Application to the Steiner Problem in Graphs*, Networks 34, 181-191.

[25] S. Voss, A. Fink e C. Duin (2005), *Looking Ahed with the Pilot Method*, Annals of Operations Research, 136, 285-302.

[26] K. Woolston e S. Albin (1988), *The design of Centralized networks with reliability and availability constraints*, Comput Oper Res 15, 207-217.