



ÁRVORES DE DIÂMETRO MÍNIMO SUJEITAS A RESTRIÇÃO DE ORÇAMENTO

Amanda Ferreira de Azevedo

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Abilio Pereira de Lucena Filho

Rio de Janeiro
Janeiro de 2021

ÁRVORES DE DIÂMETRO MÍNIMO SUJEITAS A RESTRIÇÃO DE
ORÇAMENTO

Amanda Ferreira de Azevedo

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Orientador: Abilio Pereira de Lucena Filho

Aprovada por: Prof. Abilio Pereira de Lucena Filho

Prof. Nelson Maculan Filho

Prof. Yuri Abitbol de Menezes Frota

Prof. Vinícius Leal do Forte

RIO DE JANEIRO, RJ – BRASIL

JANEIRO DE 2021

Azevedo, Amanda Ferreira de

Árvores de Diâmetro Mínimo Sujeitas a Restrição de Orçamento/Amanda Ferreira de Azevedo. – Rio de Janeiro: UFRJ/COPPE, 2021.

XIV, 38 p.: il.; 29, 7cm.

Orientador: Abilio Pereira de Lucena Filho

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2021.

Referências Bibliográficas: p. 35 – 37.

1. Diâmetro Mínimo. 2. Árvores Geradoras. 3. Árvores de Steiner. I. Lucena Filho, Abilio Pereira de. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

*There's a million things I haven't
done, but just you wait -
Alexander Hamilton*

Agradecimentos

Gostaria de agradecer primeiramente aos meus pais, Silvana Ferreira e Antônio Lobo, aos quais reconheço pelos esforços em dar as melhores oportunidades possíveis para as suas filhas, permitindo que eu chegasse até aqui. Obrigada por me apoiarem em toda e qualquer decisão.

À minha irmã querida, Ana Paula, que sempre acreditou em mim e esteve pronta a me apoiar em tudo nessa vida. Obrigada por ter sido uma das maiores incentivadoras durante esta e todas as outras etapas de minha vida.

Ao meu companheiro, Renan de Almeida, por estar sempre presente mesmo em situações de distância física. Agradeço por ter sido tão carinhoso comigo, por sempre me apoiar e principalmente pelos abraços nos momentos difíceis.

Aos meus amigos, Gabriela Freitas, Ayres Nishio, Roberta Moraes e Ana Beatriz, agradeço pelo grande incentivo e ajuda nesta etapa. Em especial, agradeço à minha amiga Suellen de Mello, que mesmo distante fisicamente, esteve ao meu lado para me acalmar e me ajudar nas horas difíceis. Obrigada por acreditar mais em mim do que eu mesma.

Agradeço aos amigos que tive a imensa alegria de conhecer no PESC. Em especial, agradeço ao Victor Hugo, que acompanhou todo o processo de minha dissertação, compartilhando comigo sua experiência. Sou muito grata pela oportunidade que tive de trabalhar com você, amigo. À Marcella Braga, uma amiga que levarei para vida toda, muito obrigada por toda ajuda, por vibrar a cada conquista minha e por ter estado super presente durante toda esta jornada. Ao Wanderson Lomenha, pelo apoio e por toda ajuda compartilhada comigo. À Amanda Camacho, pela ajuda e apoio e à Jéssica Costa, por sempre torcer por mim. Ao grupo de representantes estudantis do PESC, do qual tenho muita alegria de participar. Além de colegas de trabalho, ganhei amizades valiosas para a vida toda. Por fim, agradeço ao pessoal do LabOtim pela troca de experiências.

Agradeço ao meu orientador, Abilio Pereira de Lucena Filho, pelos ensinamentos valiosos que ultrapassaram os limites da pesquisa. Obrigada pela imensa disponibilidade, pela paciência e por me ajudar a me tornar uma pesquisadora. O senhor representa o genuíno significado da palavra Orientador e sempre será uma grande referência para mim. Sou eternamente grata por tudo que o senhor fez e continua

fazendo por mim.

Agradeço aos professores da UFRRJ, que, mesmo distantes, me apoiaram nesta jornada. Em especial, agradeço ao professor Vinícius Leal, por ter sempre acreditado em mim e por ter aceito o convite para participar desta banca. Ao professor Renan Pinto, pela imensa disponibilidade em ajudar, pelo carinho e apoio e ao professor Luciano Félix, pela paciência em me fazer acreditar que eu poderia concluir este trabalho.

Aos professores Nelson Maculan Filho e Yuri Abitbol de Menezes Frota, agradeço por aceitarem participar de minha defesa, é uma honra tê-los na banca.

À toda equipe do PESC, os quais sempre me atenderam de forma eficiente e gentil, muito obrigada. Em especial, agradeço ao Daniel, ao Ricardo e ao Gutty pelo incrível trabalho que desempenham.

Agradeço ao CNPq, pelo apoio financeiro dado a mim, sem o qual este trabalho não poderia ter sido realizado.

Agradeço a todos que ajudaram a construir este trabalho.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

ÁRVORES DE DIÂMETRO MÍNIMO SUJEITAS A RESTRIÇÃO DE ORÇAMENTO

Amanda Ferreira de Azevedo

Janeiro/2021

Orientador: Abilio Pereira de Lucena Filho

Programa: Engenharia de Sistemas e Computação

Dado um grafo não direcionado $G = (V, E)$, investigamos três problemas que demandam árvores de G de diâmetro mínimo. A definição de diâmetro aqui utilizada é a usual, correspondendo, para uma árvore $T = (V_T, E_T)$ de G , ao número de arestas no caminho de T contendo o maior número delas. Para qualquer um dos três problemas, *custos* são associados às arestas de G e uma *restrição de orçamento* se aplica ao valor máximo permitido para a soma dos custos das arestas de uma árvore. No primeiro problema, o *Budget Minimum Diameter Spanning Tree Problem*, as árvores devem ser necessariamente geradoras. No segundo, o *Budget Minimum Diameter Steiner Tree Problem*, vértices obrigatórios, $S \subset V$, são identificados de antemão e devem, necessariamente, fazer parte da solução. Esta, por sua vez, pode ser geradora ou não. Finalmente, o terceiro problema, o *Budget Minimum Diameter Terminal Steiner Tree Problem*, difere do segundo ao impor uma relação biunívoca entre as folhas de uma árvore viável e os vértices de S . Introduzimos o que nos parecem ser as primeiras formulações matemáticas propostas para qualquer um dos três problemas. Três formulações por problema. Feito isto, utilizamos os algoritmos de enumeração implícita do software *Gurobi* para resolvê-las de forma exata. Ainda pouco investigados na literatura, qualquer um dos três problemas tem um grande potencial de aplicação prática, particularmente no desenho de redes de telecomunicações. Além disso, se mostraram intrinsecamente interessantes e desafiadores, tanto em termos de modelagem quanto de algoritmos de solução.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

BUDGET CONSTRAINED MINIMUM DIAMETER TREES

Amanda Ferreira de Azevedo

January/2021

Advisor: Abilio Pereira de Lucena Filho

Department: Systems Engineering and Computer Science

Given an undirected graph $G = (V, E)$, we investigate three problems seeking minimum diameter trees of G . The definition used for the diameter of a tree, $T = (V_T, E_T)$, is the usual one. Namely, the number of edges in the path of T that contains the largest number of them. For any of the three problems, *costs* are associated with the edges of G and the sum of the edge costs of a tree may not exceed a given budget value. For the first problem, the Budget Minimum Diameter Spanning Tree Problem, feasible trees must necessarily be spanning. For the second, the Budget Minimum Diameter Steiner Tree Problem, terminal vertices $S \subset V$ are identified beforehand and must be part of any feasible tree. These trees, in turn, may be spanning or not. Finally, the third problem, the Budget Minimum Diameter Terminal Steiner Tree Problem, differs from the previous one in that it imposes a one-to-one relation between leaves of a tree and vertices of S . We propose what apparently are the very first formulations for any of the three problems. Three formulations for every problem. Next, we rely on the implicit enumeration algorithms of the solver *Gurobi* to obtain proven optimal solutions for any of them. Barely investigated in the literature, any of the three problems have practical application potential, particularly in the design of telecommunication networks. Additionally, they proved to be intrinsically interesting and defying, both in modelling and algorithmic terms.

Sumário

Lista de Figuras	xi
Lista de Tabelas	xii
Lista de Abreviaturas	xiii
1 Introdução	1
2 Formulações Matemáticas	7
2.1 Formulação Parcial da PCVA	8
2.2 Formulações Multifluxos de Caminhos Restritos	9
2.3 Formulações para o BDSTP	10
2.3.1 Formulação baseada nas restrições MTZ	10
2.3.2 Formulação multifluxos	11
2.3.3 Uma reformulação STP para o BDSTP	12
2.4 Formulações para o BMDSTP	16
2.4.1 Formulação baseada nas restrições MTZ	17
2.4.2 Formulação multifluxos	17
2.4.3 Uma reformulação STP para o BMDSTP	18
2.5 Formulações para BDTSTP	18
3 Resultados Computacionais	20
3.1 Algoritmo Branch-and-Cut	20
3.2 Instâncias	22
3.3 Descrição dos resultados	22
3.4 Resultados computacionais para o BDSTP	23
3.4.1 Grafos Completos	23
3.4.2 Grafos Esparsos	26
3.5 Resultados Computacionais para o BMDSTP	27
3.6 Resultados Computacionais para o BDTSTP	30
4 Conclusões e Trabalhos Futuros	32

Lista de Figuras

1.1	Ilustração de uma árvore geradora de custo mínimo.	1
1.2	Ilustração de uma árvore geradora de diâmetro mínimo restrita a $B = 14$	3
1.3	Ilustração de uma árvore geradora de diâmetro mínimo.	3
1.4	Soluções monopolar e dipolar para o GMDST com $n=4$	4
2.1	Obtenção de $D=(V', A)$	8
2.2	Exemplo: Grafo G e seu grafo em camadas, para $L=3$	13
2.3	À direita, a arborescência de Steiner, em D_c , correspondente à arborescência geradora BDSTP de diâmetro mínimo para o dígrafo $D = (V', A)$, à esquerda. O grafo $G = (V, E)$ da Figura 2.2, sujeito a $B = 14$, é utilizado como referência.	14

Lista de Tabelas

3.1	Resultados computacionais para grafos completos: valor de B igual a 20% do somatório dos $n - 1$ arcos de maior custo.	24
3.2	Resultados computacionais para grafos completos: valor de B igual a 30% do somatório dos $n - 1$ arcos de maior custo.	24
3.3	Resultados computacionais para grafos completos: valor de B igual a 40% do somatório dos $n - 1$ arcos de maior custo.	25
3.4	Resultados computacionais para grafos completos: valor de B igual a 35% do somatório dos $n - 1$ arcos de maior custo.	26
3.5	Resultados computacionais para grafos completos: valor de B igual a 45% do somatório dos $n - 1$ arcos de maior custo.	26
3.6	Resultados computacionais para grafos completos: valor de B igual a 55% do somatório dos $n - 1$ arcos de maior custo.	27
3.7	Resultados computacionais para instâncias com B igual a 100% do valor da solução ótima do STPG.	28
3.8	Resultados computacionais para instâncias com B igual a 110% do valor da solução ótima do STPG.	28
3.9	Resultados computacionais para instâncias com B igual a 120% do valor da solução ótima do STPG.	29
3.10	Resultados computacionais para B igual a 90% do valor da solução ótima do MDTSTP.	30
3.11	Resultados computacionais para B igual a 95% do valor da solução ótima do MDTSTP.	30
3.12	Resultados computacionais para B igual a 100% do valor da solução ótima do MDTSTP.. . . .	31

Lista de Abreviaturas

AB	Algoritmo de Borůvka, p. 2
AJ	Algoritmo de Jarník, p. 2
AK	Algoritmo de Kruskal, p. 2
AP	Algoritmo de Prim, p. 2
BC	Branch-and-Cut, p. 6
BDSTP	Budget Minimum Diameter Spanning Tree Problem, p. 3
BDTSTP	Budget Minimum Diameter Terminal Steiner Tree Problem, p. 5
BMDSTP	Budget Minimum Diameter Steiner Tree Problem, p. 5
CP	Constrained Path, p. 9
DCMSTP	Diameter Constrained Minimum Spanning Tree Problem, p. 5
DSTP	Minimum Diameter Steiner Tree Problem, p. 4
ECVA	Equações de Centralidade de Vértice ou Aresta, p. 8
FST	Full Steiner Trees, p. 5
GDMSTP	Geometric Minimum Diameter Spanning Tree Problem, p. 4
GS	Grafo Suporte, p. 21
HMSTP	Hop-Constrained Minimum Spanning Tree Problem, p. 6
MCF	Multi-Commodity Flows, p. 6
MDTSTP	Minimum Diameter Terminal Steiner Tree Problem, p. 5
MSTP	Minimum Spanning Tree Problem, p. 1
MTZ	Miller-Tucker-Zemlin, p. 6

PAC	Propriedade da Aresta Central, p. 5
PCVA	Propriedade da Centralidade de um Vértice ou Aresta, p. 7
PS	Problema de Separação, p. 21
PVC	Propriedade do Vértice Central, p. 5
STPG	Steiner Tree Problem in Graphs, p. 4
TeST	Terminal Steiner Tree Problem, p. 5

Capítulo 1

Introdução

Seja $G = (V, E)$ um grafo simples, conexo e não orientado, onde V é o conjunto de vértices e E é o conjunto de arestas. Associe um custo c_e à cada aresta $e = \{i, j\} \in E$ e defina como $E(S) \in E$ as arestas de G com ambas as extremidades em $S \subseteq V$. Uma árvore $T = (V_T, E_T)$, com $V_T \subseteq V$ e $E_T \subseteq E(V_T) \subseteq E$, é um subgrafo conexo e acíclico de G . A árvore é dita geradora [1] quando contém todos os vértices do grafo, isto é, quando $V_T = V$. Dentre todas as árvores geradoras de G , uma de menor custo é chamada *mínima* e o *Minimum Spanning Tree Problem* (MSTP) consiste em encontrá-la. Uma ilustração de uma árvore geradora de custo mínimo é dada na Figura 1.1.

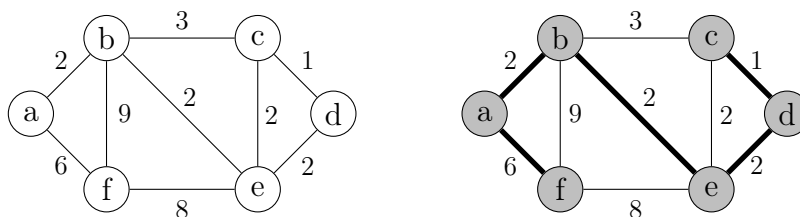


Figura 1.1: Ilustração de uma árvore geradora de custo mínimo.

Uma visão histórica detalhada do MSTP é encontrada em GRAHAM e HELL [2], com diversas citações e traduções das primeiras contribuições sobre o tema. Outra contribuição mais recente, na mesma linha, é a de SCHRIJVER [3]. No que vamos descrever a seguir sobre o MTSP, tomamos por base, principalmente, essas duas referências.

O MSTP começou ser investigado na década de 1920, quando Otakar Borůvka [4] propôs um procedimento para desenhar uma rede elétrica de menor custo possível. A motivação para o trabalho se originou de uma aplicação prática, real, em uma região rural localizada no Sul da Morávia, na atual República Tcheca. Mais tarde, BORŮVKA [5] publicou uma descrição formal do procedimento sugerido em [4], o

Algoritmo de Borůvka (AB). A seguir, em 1930, Vojtěch Jarník sugeriu o que definiu ser uma *simplificação* do algoritmo de Borůvka, intitulada *Tree Growing Procedure* [6]. O Algoritmo de Jarník (AJ) [6] foi publicado sob o mesmo título e no mesmo jornal que AB. Mais ainda, se apresentou como um complemento à ideia original proposta em AB.

Em meados da década de 1950, KRUSKAL [7], fazendo menção apenas a AB, na sua versão informal descrita em [5], propôs diversos procedimentos de solução para o MSTP, sugerindo que o trabalho de Borůvka seria *desnecessariamente elaborado*. Em paralelo, PRIM [8], referenciando apenas [5, 7], redescobriu AJ. Desde então, o algoritmo descrito em [4, 5] e refinado em [7] é chamado de *Algoritmo de Kruskal* (AK), enquanto o algoritmo proposto em [6] e refinado em [8] é denominado *Algoritmo de Prim* (AP).

Em termos de complexidade computacional [9], o MSTP é um problema *fácil de resolver* e os algoritmos AK e AP demandam, no pior caso, respectivamente, $O(m \log n)$ e $O(n^2)$ operações algébricas elementares e de comparação para resolvê-lo [10], sendo $n = |V|$ e $m = |E|$.

Tanto o MSTP quanto suas inúmeras generalizações e casos particulares são amplamente investigadas na literatura. Isso se deve não só ao interesse intrínseco que despertam, mas também pelas suas diversas aplicações práticas. Em particular, GRAHAM e HELL [2] mencionam aplicações envolvendo o desenho de diversos tipos de redes. Por exemplo, redes de computadores, de telecomunicação, de transportes, de dutos e de conexões cabeadas.

Deve-se ressaltar que uma eventual imposição de restrições adicionais ao MSTP pode torná-lo um problema de difícil resolução. A título de exemplo, o *Problema da Árvore Geradora com Restrição de Grau* [11], que impõe uma restrição de grau máximo a cada vértice da árvore geradora, é um problema NP-Difícil. Outro exemplo é o *Problema da Árvore Geradora Mínima com Restrição de Número de Folhas* [12], que impõe uma restrição ao número máximo de folhas que uma árvore geradora pode conter.

Nesta dissertação investigamos problemas NP-Difíceis que têm por objetivo obter árvores de G de menor diâmetro possível, satisfazendo a uma restrição adicional. Antes de descrevê-los, no entanto, vamos complementar a notação a ser utilizada e discutir alguns problemas da literatura que são próximos a eles.

Um caminho em G é dito *simples* quando não visita um mesmo vértice mais de uma vez. Associe a cada aresta $e = \{i, j\} \in E$ um par de pesos não-negativos (l_e, c_e) , tal que l_e representa o tamanho da aresta e c_e , seu custo. Denota-se por d_{ij} ao comprimento do menor caminho simples ligando os vértices $i, j \in V$, ou seja, à *distância* entre eles no grafo G . Por fim, o *diâmetro* de G , d , é dado pela maior distância existente entre qualquer par de vértices de G .

Em complemento as definições acima, seja B um número positivo que impõe um teto de orçamento, ou seja, um limite superior para o quanto se pode gastar na escolha das arestas de uma árvore geradora $T = (V_T, E_T)$. Denominamos, então, por *Budget Minimum Diameter Spanning Tree Problem* (BDSTP) ao problema de encontrar uma árvore geradora T tal que a soma total de suas arestas não ultrapasse B e seu diâmetro seja o menor possível. O problema foi proposto por PLESNIK [13], em 1981, sob uma denominação imprecisa. Na mesma referência foi também identificado como NP-Difícil. Até onde sabemos, inexistem na literatura formulações e algoritmos exatos para o BDSTP. Uma ilustração de uma árvore geradora ótima para o problema é dada na Figura 1.2, para $l_e = 1, \forall e \in E$.

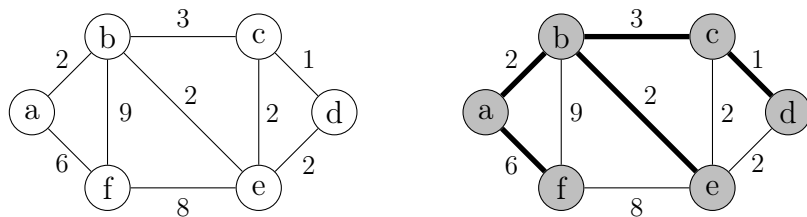


Figura 1.2: Ilustração de uma árvore geradora de diâmetro mínimo restrita a $B = 14$.

Além do interesse natural que desperta, o BDSTP se presta a modelar situações práticas de interesse. Podemos pensar, por exemplo, em problemas onde se deseja enviar mensagens através de uma rede de telecomunicação, recebidas, no destino, ao menor atraso possível. Tais redes, neste caso específico, devem ter uma topologia de árvore e não devem custar mais do que um valor B , previamente definido.

Quando os custos c_e das arestas de G são todos iguais a 1 e $B = |V| - 1$, o problema é equivalente ao *1-center problem* (veja [14] para mais detalhes), sendo então de fácil resolução. Neste caso, a restrição de orçamento se torna redundante e o problema poderia ser denominado, simplesmente, *Minimum Diameter Spanning Tree Problem* (MDSTP), como sugerido, posteriormente, por HASSIN e TAMIR [15]. Uma ilustração de uma árvore geradora ótima para o MDSTP é dada na Figura 1.3, assumindo $l_e = 1, \forall e \in E$.

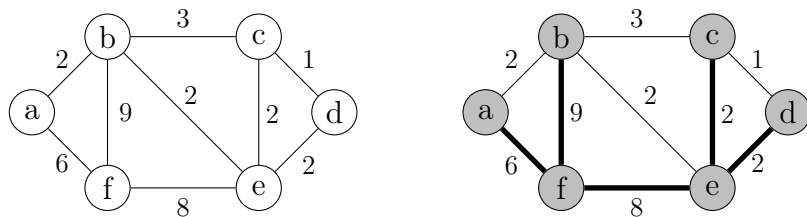


Figura 1.3: Ilustração de uma árvore geradora de diâmetro mínimo.

Mais adiante, HO *et al.* [16] investigam um caso particular do MDSTP, envolvendo grafos completos definidos no espaço Euclidiano. Neste caso, dado um conjunto de n pontos (vértices) distintos no espaço Euclidiano, deseja-se encontrar uma árvore $T = (V_T, E_T)$ que conecte esses pontos ao menor diâmetro possível. Os autores demonstraram que uma solução ótima para esse problema pode ser caracterizada como *monopolar* ou *dipolar*. Ela é monopolar se existir um vértice $s \in V$, chamado *monopolo*, de tal forma que todo vértice $i \in V \setminus \{s\}$ estará conectado diretamente a s em uma solução ótima. Será dipolar se existirem dois vértices $s_1, s_2 \in V$, chamados *dipolos*, tal que todo vértice $i \in V \setminus \{s_1, s_2\}$ estará conectado diretamente a um deles em uma solução ótima. Este resultado também se aplica quando o tamanho das arestas satisfaz a desigualdade triangular. O problema é então denominado *Geometric Minimum Diameter Spanning Tree Problem* (GDMSTP) e existem algoritmos polinomiais para resolvê-lo [16], de ordem $O(n \log n)$ para o caso monopolar e $O(n^3)$, para o dipolar. Na Figura 1.4, ilustramos graficamente soluções monopolar e dipolar para o GDMSTP, em exemplos onde $n=4$.

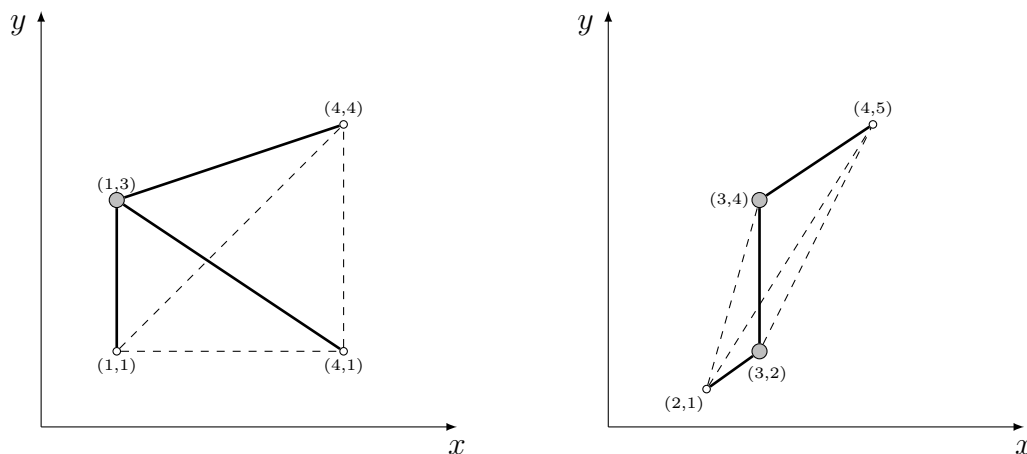


Figura 1.4: Soluções monopolar e dipolar para o GDMST com $n=4$.

Finalmente, HASSIN e TAMIR [15] estenderam o resultado anterior e demonstraram que o MDSTP é equivalente ao *1-center problem* para o caso geral, isto é, quando o tamanho das arestas não satisfaz, necessariamente, a desigualdade triangular. Para resolver este caso mais geral, propõem um algoritmo que demanda $O(mn + n^2 \log n)$ operações algébricas elementares e de comparação.

Outro problema próximo aos que aqui investigamos é o *Steiner Tree Problem in Graphs* (STPG) (vide [17, 18]). Este é um problema NP-Difícil [19] que possui inúmeras aplicações práticas e que foi investigado sob diversas variantes distintas. Quando o objetivo é relacionado ao diâmetro, define-se o *Minimum Diameter Steiner Tree Problem* (DSTP). Neste, dado um grafo $G = (V, E)$ e um subconjunto $S \subseteq V$, deseja-se encontrar uma árvore de diâmetro mínimo $T = (V_T, E_T)$ que con-

tenha todos os vértices de S . Vértices de S são denominados *vértices obrigatórios* ou *terminais* e os vértices de $V \setminus S \neq \emptyset$ são denominados *vértices opcionais*. Em particular, denominamos *Vértices de Steiner* àqueles em que $V_T \cap (V \setminus S)$. Para uma definição mais geral de diâmetro e para grafos completos, DING e QIU [20] sugeriram recentemente um algoritmo exato para resolver o problema em tempo polinomial.

Impondo-se um limite superior para o custo de uma árvore de Steiner, chegamos a outro problema investigado nesta dissertação, que denominamos *Budget Minimum Diameter Steiner Tree Problem* (BMDSTP). Este, aparentemente, não foi ainda investigado na literatura.

Finalmente, investigamos também um problema que se origina de uma versão restrita do BMDSTP. Nesta, as folhas de T , ou seja, vértices de grau 1, devem necessariamente pertencer a S . Quando árvores de Steiner possuem esta característica, são chamadas *Full Steiner Tree* (FSTs) [21] e encontrar uma FST de menor custo corresponde a um problema NP-Difícil denominado *Terminal Steiner Tree Problem* (TeST) [22]. Quando a FST deve ser de diâmetro mínimo, DING e QIU [20] denominam o problema a *Minimum Diameter Terminal Steiner Tree Problem* (MDTSTP). Para a definição mais geral de diâmetro e para grafos completos, os autores sugerem algoritmos exatos para resolver o problema em tempo polinomial.

Como fizemos anteriormente, impondo uma restrição de orçamento ao MDTSTP, definimos o terceiro problema investigado nesta dissertação, o *Budget Minimum Diameter Terminal Steiner Tree Problem* (BDTSTP). A este, podemos associar aplicações reais onde, por exemplo, receptores de uma rede de telecomunicação não podem atuar como transmissores.

Como contribuição desta dissertação, propomos formulações para o BDSTP, o BMDSTP e o BDTSTP. Nestas formulações, a definição de diâmetro é feita em termos de número de arestas. Em outras palavras, é assumido que $l_e = 1, \forall e \in E$.

Nossas formulações se baseiam nas seguintes propriedades de árvores: *Propriedade do Vértice Central* (PVC) e *Propriedade da Aresta Central* (PAC) (vide [23], para detalhes). Assumindo que o diâmetro d de uma árvore $T = (V_T, E_T)$, geradora ou não, é par, a primeira propriedade aponta para a existência de um *vértice central*, $i \in V_T$, tal que qualquer vértice $j \in V_T \setminus \{i\}$, em uma árvore geradora *viável*, está a não mais do que $d/2$ arestas de i . A segunda propriedade, que se aplica quando d é ímpar, aponta para a existência de uma *aresta central* $e = \{i, j\}$, tal que qualquer vértice $k \in V_T \setminus \{i, j\}$ está a não mais do que $(d + 1)/2$ arestas de i ou de j .

Outro problema relacionado aos que vamos aqui investigar, é o *Diameter Constrained Minimum Spanning Tree Problem* (DCMSTP). Este, consiste em encontrar uma árvore geradora de custo mínimo para G , contendo um diâmetro $d \leq D$, onde D é um inteiro positivo tal que $2 \leq D \leq |V| - 1$. O problema, em sua versão de

decisão, é NP-Completo para $D \geq 4$, como indicado em [24].

Deve-se ressaltar também que o DCMST pode ser reformulado como uma generalização do *Hop-Constrained Minimum Spanning Tree Problem* (HMSTP) [25]. Tal problema impõem um vértice *raiz*, $r \in V$, para a árvore geradora, $T = (V_T, E_T)$, que se quer construir. Da mesma forma, impõem que a distância entre r e $j \in V \setminus \{r\}$, não pode ultrapassar um número natural H de *hops* (arestas), definido de antemão. PVC e PAC naturalmente se aplicam à identificação de um vértice ou de uma aresta central para a *árvore Hop*, sendo, então, uma parte integrante de diversas formulações propostas para o DCMSTP.

As primeiras formulações do DCMSTP baseadas em PVC e PAC se devem a ACHUTAN e CACCETTA [24, 26] e se utilizam das desigualdades de *Miller-Tucker-Zemlin* (MTZ) para impor a conectividade das soluções. A seguir, GOUVEIA e MAGNANTI [27] investigaram diferentes variantes das formulações em [24, 26], baseadas em *Multi-Commodity Flows* (MCF). Estas, apesar de envolverem um grande número de variáveis, levam a relaxações lineares mais fortes que aquelas obtidas em [24, 26]. Em DOS SANTOS *et al.* [28], as formulações de [24, 26] são reforçadas através de um *lifiting* das desigualdades MTZ e de algumas desigualdades adicionais. Mais recentemente, GOUVEIA *et al.* [29] chegaram aos melhores resultados disponíveis na literatura, tanto para o HMSTP quanto para o DCMST, reformulando os dois problemas como STPs [17].

Nossas formulações para o BDSTP, BMDSTP e BDTSTP, descritas no Capítulo 2, se baseiam, em grande parte, nas formulações indicadas acima para o DCMSTP. São, como mencionamos anteriormente, as primeiras formulações matemáticas propostas para qualquer um destes três problemas. No Capítulo 3, apresentamos resultados computacionais para as formulações introduzidas no capítulo anterior. Tais resultados foram obtidos através de algoritmos do tipo *Branch-and-Cut*(BC), implementados na linguagem de programação *Julia*, através da interface oferecida pelo *JuMP* [30], que faz a intermediação entre a linguagem *Julia* e o software de programação inteira *Gurobi*. São, portanto, os primeiros algoritmos exatos propostos para os problemas. Por fim, a dissertação é concluída no Capítulo 4, com uma discussão dos principais resultados obtidos e sugestões de trabalhos futuros.

Capítulo 2

Formulações Matemáticas

Neste capítulo, introduzimos formulações matemáticas para o BDSTP, BMDSTP e o BDTSTP. Nossas formulações se aplicam à definição usual do diâmetro de um grafo não direcionado, em termos de número de arestas, e, tanto quanto sabemos, são as primeiras propostas para os três problemas.

Todos os problemas a serem aqui formulados têm por objetivo encontrar árvores restritas de um grafo, geradoras ou não, de menor diâmetro possível. Para qualquer um deles, a restrição imposta diz respeito a um *orçamento* para cobrir custos de arestas. Não é sabido de antemão se o diâmetro de uma árvore ótima é par ou ímpar e a contribuição mais importante de nossas formulações diz respeito exatamente ao tratamento que damos a esta questão. Mais especificamente, formulamos matematicamente a Propriedade da Centralidade de um Vértice ou Aresta (PCVA) de uma árvore de $G = (V, E)$. Feito isso, adaptamos para o BDSTP as principais formulações disponíveis para o DCMSTP. Neste processo, incorporamos a elas nossa formulação da PCVA, juntamente com a restrição de orçamento e procedemos aos ajustes necessários. Em seguida, adaptamos nossas formulações do BDSTP respectivamente para o BMDSTP e o BDTSTP.

Para formular o BDSTP tomamos, como ponto de partida, um grafo não direcionado $G = (V, E)$, ao qual associamos um dígrafo $D = (V', A)$. Ou seja, como é comum na literatura, direcionamos as arestas de G . Além disso, introduzimos um vértice artificial, r , que terá por função apontar para o vértice ou aresta central das arborescências que queremos obter. Temos então que $V' = V \cup \{r\}$ e $A = \{(i, j), (j, i) : e = \{i, j\} \in E\} \cup \{(r, i) : i \in V\}$. Para os custos dos arcos de D tomamos $c_{ij} = c_{ji} = c_e$, sempre que $e = \{i, j\} \in E$. Para os arcos que se originam de r , impomos $c_{ri} = 0, \forall i \in V$. Na Figura 2.1 ilustramos graficamente a obtenção de um dígrafo $D = (V', A)$.

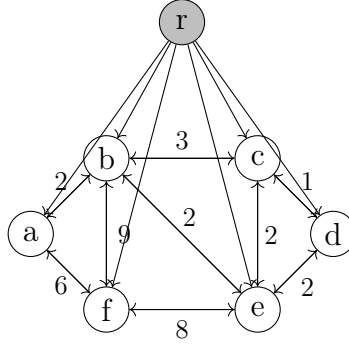


Figura 2.1: Obtenção de $D=(V', A)$

Obtido $D = (V', A)$, o que se deseja encontrar é uma arborescência $T = (V_T, A_T)$, onde a raiz é dada pelo vértice r . Para formular a paridade do diâmetro de T , uma variável binária, z , é utilizada, tomando o valor 1 quando o diâmetro é ímpar e 0, no caso par. Para contabilizar o número de arcos correspondentes ao diâmetro de T , utilizamos uma variável não-negativa, $s \in \mathbb{R}$. Mais especificamente, esta variável identifica o comprimento do maior caminho de T entre o vértice artificial e os demais vértices da arborescência. Um parâmetro importante em nossas formulações é um limite superior, L , para o comprimento desses caminhos. Este, pode ser estimado através de heurísticas ou simplesmente tomando $L = \left\lceil \frac{|V|}{2} \right\rceil$, que representa um limite superior para o comprimento de qualquer caminho em T , especificamente entre um vértice central, ou de uma das extremidades de uma aresta central, até uma folha qualquer de T . Além de L , um valor positivo, B , que impõem um limite superior para a soma dos custos dos arcos de uma arborescência, é também utilizado como um dado de entrada para o problema.

Como função objetivo, utilizamos como ponto de partida $2(s - 1)$, uma vez que busca-se minimizar a maior distância entre duas folhas de T . Além disso, adicionamos uma unidade a $2(s - 1)$, sempre que o diâmetro de uma árvore ótima é ímpar. Nossa função objetivo é, então, dada por $2(s - 1) + z$.

Vamos descrever agora um conjunto de restrições que denominamos Equações de Centralidade de Vértice ou Aresta (ECVA). Estas, formulam parcialmente a PCVA e combinadas com as demais desigualdades em várias das formulações aqui propostas, formulam a PCVA integralmente.

2.1 Formulação Parcial da PCVA

Associamos variáveis binárias x_{ij} à cada arco $(i, j) \in A$, indicando se o mesmo faz ou não parte da arborescência. Se o diâmetro da árvore é ímpar, variáveis binárias w_e ,

definidas para todo $e \in E$, são utilizadas para identificar uma *aresta central* para a árvore de G (correspondendo à arborescência de D). Neste caso, a variável binária, z , necessariamente assume o valor 1, indicando que se $w_e = 1$ se aplica a $e = \{i, j\}$, $x_{ri} = x_{rj} = 1$ também se aplica a (r, i) e (r, j) . Alternativamente, no caso em que o diâmetro da arborescência é par, z assume o valor 0, indicando que apenas um único arco da arborescência deverá apontar para fora de r . Uma formulação parcial para a PCVA é dada pela região poliédrica, \mathcal{R}_c , definida como

$$\sum_{j \in V} x_{rj} = z + 1 \quad (2.1)$$

$$\sum_{e \in E} w_e = z \quad (2.2)$$

$$w_e \leq x_{ri}, \quad e = \{i, j\} \in E, i < j \quad (2.3)$$

$$w_e \leq x_{rj}, \quad e = \{i, j\} \in E, i < j \quad (2.4)$$

$$0 \leq x_{ij} \leq 1, \quad \forall (i, j) \in A \quad (2.5)$$

$$w_e \geq 0, \quad \forall e = \{i, j\} \in E \quad (2.6)$$

$$0 \leq z \leq 1. \quad (2.7)$$

Como veremos nas formulações que se seguem, quando $z = 0$, a restrição (2.1) assegura que só poderá haver um único arco saindo de r , caracterizando o caso par. Alternativamente, quando $z = 1$, exatamente dois arcos deverão apontar para fora de r , um para cada uma das extremidades da aresta escolhida em (2.2). As restrições (2.3) e (2.4), por sua vez, impõem que, escolhida uma aresta $e = \{i, j\} \in E$, os arcos (r, i) e (r, j) devem necessariamente fazer parte da solução.

2.2 Formulações Multifluxos de Caminhos Restritos

Vamos agora descrever uma formulação MCF para um caminho restrito e denominá-la *Constrained Path* (CP), a exemplo do que é feito em GOUVEIA e MAGNANTI [27]. CP impõem as condições necessárias para o escoamento de fluxos na arborescência, além de restringir o comprimento dos caminhos associados à T . Na formulação de CP, modelamos o envio de uma unidade de mercadoria k , partindo do vértice artificial, r , para o vértice $k \in V$. Para tanto, utilizamos variáveis de fluxo $y_{ij}^k \geq 0$, associadas à cada arco $(i, j) \in A$ e à cada mercadoria k . Estas, indicam a quantidade de mercadoria k que atravessa o arco (i, j) , no seu trajeto da origem r ao seu destino $k \in V$. Uma formulação para CP está associada aos pontos inteiros pertencentes a uma região poliédrica, \mathcal{R}_p , definida como

$$\sum_{(i,j) \in A} y_{ij}^k - \sum_{(j,i) \in A} y_{ji}^k = \begin{cases} 1, & i = r \\ 0, & i \neq k, i \neq r \\ -1, & i = k, \end{cases} \quad i, k \in V \quad (2.8)$$

$$\sum_{(i,j) \in A} y_{ij}^k \leq L + 1, \quad k \in V \quad (2.9)$$

$$y_{ij}^k \geq 0, \quad \forall (i, j) \in A, k \in V. \quad (2.10)$$

Para os pontos inteiros de \mathcal{R}_p , as restrições (2.8) caracterizam os fluxos de transporte de mercadorias. No caso em que $i = r$, garantem a partida de cada mercadoria $k \in V$, de sua origem, r , para o seu destino k . Por outro lado, quando $i = k$, para $k \in V$, garantem a chegada da mercadoria k ao seu destino, k . Finalmente, quando $i \neq k, k \in V \setminus \{i\}$ e $i \neq r$, indicam se o vértice i é ou não utilizado no caminho de transporte da mercadoria k , de sua origem r ao seu destino k . Em complemento às restrições anteriores, as equações (2.9) impõem que o comprimento do caminho de transporte de uma mercadoria $k \in V$ não ultrapasse o valor $L + 1$, somando uma unidade associado aos arcos que saem de r . Por fim, as restrições (2.10) garantem um fluxo não-negativo de mercadorias nos arcos de D .

2.3 Formulações para o BDSTP

Nesta seção, introduzimos três formulações matemáticas para o BDSTP, reformulado em termos de $D = (V', A)$. O problema, como indicado anteriormente, consiste em encontrar uma arborescência geradora $T = (V_T, A_T)$, de menor diâmetro possível, restrita a uma soma de custos de arcos menor do que B .

2.3.1 Formulação baseada nas restrições MTZ

Nossa primeira formulação estende ao BDSTP a formulação proposta em DOS SANTOS *et al.* [28] para o DCMSTP. Para identificar a arborescência geradora, são utilizadas variáveis binárias x_{ij} , que indicam se o arco $(i, j) \in A$ faz parte ou não da arborescência. Além disso, a formulação se utiliza de um *lifting* das desigualdades Miller-Tucker and Zemlin (MTZ) [31], para impor a conectividade das soluções. Tais desigualdades são descritas através das variáveis $u_i \in \mathbb{R}_+, i \in V'$. Finalmente, a formulação é definida como:

$$\text{Min} \quad 2(s - 1) + z \quad (2.11)$$

$$\text{s.a} \quad \sum_{(i,j) \in A} x_{ij} = 1, \quad j \in V \quad (2.12)$$

$$u_i - u_j + (L + 1)x_{ij} + (L - 1)x_{ji} \leq L, \quad (i, j) \in A \quad (2.13)$$

$$\sum_{(i,j) \in A} c_{ij}x_{ij} + \sum_{e \in E} c_e w_e \leq B \quad (2.14)$$

$$s \geq u_i, \quad i \in V \quad (2.15)$$

$$(\mathbf{x}, \mathbf{w}, z) \in \mathcal{R}_c \quad (2.16)$$

$$0 \leq u_i \leq L + 1, \quad \forall i \in V' \quad (2.17)$$

$$s \geq 0. \quad (2.18)$$

As restrições (2.12) garantem a inclusão de todos os vértices de D na arborescência. As restrições (2.13) e (2.17) garantem que todo caminho, levando de r até um vértice de V , tem, no máximo, $L + 1$ arcos. Para assegurar que o custo total da arborescência não ultrapasse o valor B , é introduzida a restrição (2.14). Por sua vez, as equações (2.15) estabelecem um limite inferior para s , em termos dos comprimentos dos caminhos que levam de r à cada vértice $i \in V$. Note que ao minimizarmos nossa função objetivo, esse limite inferior é tornado tão pequeno quanto possível for. A propriedade PCVA é incorporada através de (2.16). Finalmente, a desigualdade (2.18) impõem que s seja não-negativo, uma vez que representa a quantidade de arcos de um caminho.

2.3.2 Formulação multifluxos

Outra formulação para o BDSTP, baseada em MCF, será apresentada a seguir. Muito se assemelha à formulação proposta em GOUVEIA e MAGNANTI [27] para o HMSTP, que utiliza como referência uma fonte única de fluxos. Como proposto em [27], r , o vértice artificial de D , irá aqui desempenhar o papel desta fonte. Associamos variáveis de fluxo, y_{ij}^k à cada $(i, j) \in A$, onde $y_{ij}^k = 1$ ocorre quando o caminho de transporte da mercadoria k , $k \in V$, entre r e k , passa pelo arco (i, j) . Mais uma vez, para identificar a arborescência geradora, utilizamos as variáveis binárias x_{ij} , definidas para cada $(i, j) \in A$. Uma formulação MCF para o BDSTP é dada por:

$$\text{Min} \quad 2(s - 1) + z \quad (2.19)$$

$$\text{s.a} \quad y_{ij}^k \leq x_{ij} \quad \forall (i, j) \in A, k \in V \quad (2.20)$$

$$\sum_{(i,j) \in A} c_{ij}x_{ij} + \sum_{e \in E} c_e w_e \leq B \quad (2.21)$$

$$(\mathbf{x}, \mathbf{w}, z) \in \mathcal{R}_c \quad (2.22)$$

$$\mathbf{y} \in \mathcal{R}_p \quad (2.23)$$

$$s \geq \sum_{(i,j) \in A} y_{ij}^k \quad k \in V \quad (2.24)$$

$$s \geq 0. \quad (2.25)$$

As restrições (2.20) interligam as variáveis de fluxo com as variáveis binárias \mathbf{x} , que definem a árvore. Ou seja, se houver fluxo não nulo passando pelo arco $(i, j) \in A$, este fará parte da arborescência geradora. Mais uma vez, incluímos, em (2.21) a restrição de orçamento. As restrições (2.22) são relativas a formulação parcial da PCVA, e as (2.23) correspondem a CP. Por fim, as equações (2.24), em conjunto com a função objetivo, estabelecem o valor de s , ao minimizar o caminho mais longo na arborescência.

2.3.3 Uma reformulação STP para o BDSTP

Nossa próxima formulação para o BDSTP se baseia em uma proposta de GOUVEIA *et al.* [32], que explora uma equivalência entre o HMSTP e o STP. Isto ocorre ao se adaptar o grafo original, G , a um *grafo em camadas*. Neste novo grafo, é possível utilizar qualquer formulação do STP para resolver o HMSTP. No nosso caso, vamos utilizar uma formulação MCF. Esta, por sua vez, tem origem no modelo *Revised Hop-NF*, sugerido em [32] para o HMSTP. *Revised Hop-NF*, vale aqui ressaltar, define uma versão refinada do modelo *Hop-cut*, também proposto pelos mesmos autores.

Considere um grafo em camadas $D_c = (V_c, A_c)$, obtido a partir do grafo D definido anteriormente. Para representar a raiz do HMSTP, utilizamos o vértice artificial, r , incluindo assim uma primeira camada, a camada 0, ao grafo D_c . As camadas, neste caso, possuem o papel de definir a profundidade dos caminhos da arborescência. Para defini-las, utilizamos a constante L , estabelecida anteriormente. Por conta da camada associada ao vértice artificial, a profundidade máxima permitida a um caminho será $L + 1$.

Todas as camadas, à exceção da camada 0, definida unicamente por r , são formadas por cópias dos vértices de V . Mais precisamente, D_c contém $L + 1$ cópias desses vértices. Com isso, um nó $i_h \in V_c$ representa uma cópia do vértice $i \in V$, localizada na camada $1 \leq h \leq L + 1$ de D_c . Formalmente, definimos $D_c = (V_c, A_c)$ como

- $V_c = \{i_h : i \in V, 1 \leq h \leq L + 1\} \cup \{r\}$,
- $A_0 = \{(r, j_1) : j \in V\}$,
- $A_1 = \{(i_h, j_{h+1}) : (i, j) \in A, 1 \leq h \leq L\}$,

- $A_2 = \{(i_h, i_{L+1}) : i \in V, 1 \leq h \leq L\}$,
- $A_c = A_0 \cup A_1 \cup A_2$.

Como exemplo, a Figura 2.2 descreve um grafo $G = (V, E)$ e o grafo em camadas, $D_c = (V_c, A_c)$, que associamos a ele.

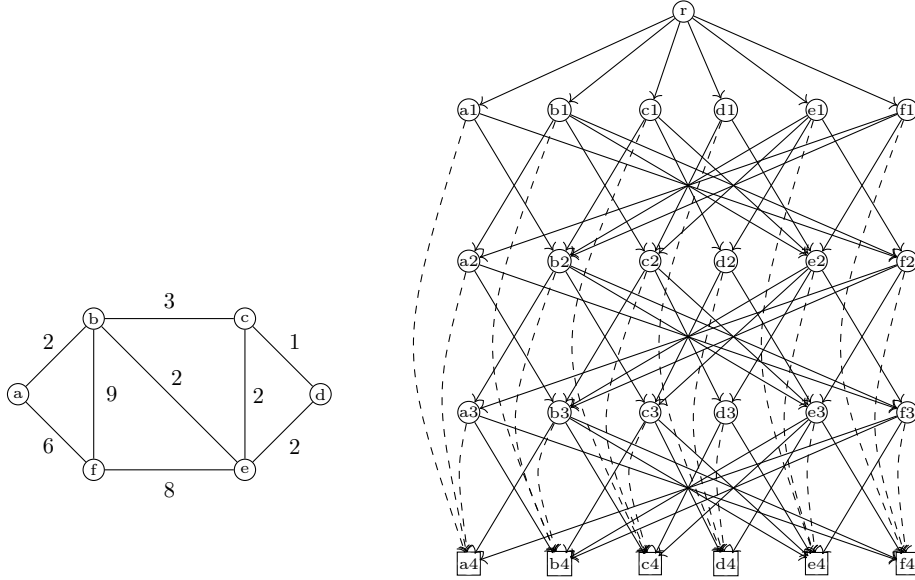


Figura 2.2: Exemplo: Grafo G e seu grafo em camadas, para $L=3$.

Em nossa reformulação STP para o BDSTP, os vértices terminais são definidos por $S = \{i_{L+1} : i \in V\}$ e correspondem aos quadrados da Figura 2.2. Em relação aos arcos de D_c , associamos custos nulos à cada arco $(r, j_1) \in A_0$. Por sua vez, custos c_{ij} são associados aos arcos (i_h, j_{h+1}) definidos em A_1 . Finalmente, atribuímos custos iguais a zero à cada arco $(i_h, i_{L+1}) \in A_2$, representados na Figura 2.2 pelos arcos tracejados. Ilustramos na Figura 2.3 uma solução ótima para o BDSTP, expressa em termos dos elementos de $D = (V', A)$ e $D_c = (V_c, A_c)$, associados ao grafo G da Figura 2.2.

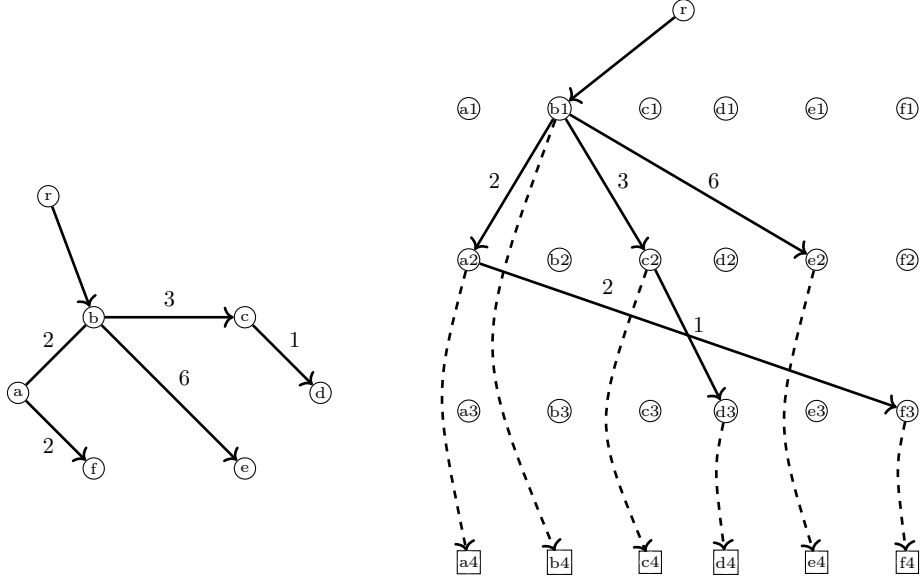


Figura 2.3: À direita, a arborescência de Steiner, em D_c , correspondente à arborescência geradora BDSTP de diâmetro mínimo para o dígrafo $D = (V', A)$, à esquerda. O grafo $G = (V, E)$ da Figura 2.2, sujeito a $B = 14$, é utilizado como referência.

Note, na Figura 2.3, que na arborescência de Steiner para $D_c = (V_c, A_c)$, qualquer caminho que leva de r a uma cópia de $i \in V$ tem sempre um comprimento menor ou igual a $L + 1$, como desejado.

Como mencionado anteriormente, nossa reformulação de Steiner para o BDSTP consiste em uma formulação MCF para o STP, definido sobre $D_c = (V_c, A_c)$, a exemplo da reformulação MCF do *Hop-cut*, proposta por GOUVEIA *et al.* [32] para o HMSTP. Naquela referência, após algumas operações algébricas, foi possível omitir o uso de variáveis \mathbf{x} associadas aos arcos de A_2 . Fazemos o mesmo aqui. Utilizamos também, a exemplo daquela referência, variáveis binárias x_{rj}^1 associadas à cada arco $(r, j_1) \in A_0$. Para as camadas subsequentes de D_c , utilizamos variáveis x_{ij}^h para cada arco $(i_{h-1}, j_h) \in A_c$ com $2 \leq h \leq L + 1$. Estas, têm o intuito de definir se a cópia do arco $(i, j) \in A$, situada entre as camadas $h - 1$ e h , está ou não na solução. Note que, ao modelarmos tal condição, os arcos de D_c incidentes aos vértices obrigatórios, ou seja, aqueles em S , são implicitamente incorporados à solução. Definimos também variáveis não-negativas de fluxo y_{rj}^{1k} , para cada arco $(r, j_1) \in A_0$ e para cada mercadoria $k \in V$, enviada de r ao vértice k . Da mesma forma, utilizamos também variáveis y_{ij}^{hk} , definidas para cada arco $(i_{h-1}, j_h) \in A_c$ com $(i, j) \in A$ e $2 \leq h \leq L + 1$, para complementar, eventualmente, o caminho de transporte da mercadoria k , $k \in V$, de sua origem r a seu destino k , através da cópia (i_{h-1}, j_h) de $(i, j) \in A$. Como antes, as variáveis $w_e \in E$ são aqui utilizadas no intuito de identificar uma aresta central, se o diâmetro da arborescência BDSTP for ímpar. Uma reformulação do problema, em termos do grafo em camadas D_c , é

então descrita como

$$\text{Min } 2(s-1) + z \quad (2.26)$$

$$\text{s.a } \sum_{j \in V} x_{rj}^1 = z + 1 \quad (2.27)$$

$$\sum_{j \in V} y_{rj}^{1k} = 1, \quad k \in V \quad (2.28)$$

$$\sum_{h=2}^{L+1} \sum_{i:(i,j) \in A, i \neq r} x_{ij}^h + x_{rj}^1 = 1, \quad j \in V \quad (2.29)$$

$$y_{ri}^{1k} - \sum_{j:(i,j) \in A} y_{ij}^{2k} = 0, \quad k \in V, i \in V; i \neq k \quad (2.30)$$

$$\sum_{j:(j,i) \in A, j \neq k} y_{ji}^{hk} - \sum_{j:(i,j) \in A} y_{ij}^{(h+1)k} = 0, \quad k \in V, i \in V, h = 2, \dots, L \quad (2.31)$$

$$\sum_{h=2}^{L+1} \sum_{j:(j,k) \in A} y_{jk}^{hk} + y_{rk}^{1k} = 1, \quad k \in V \quad (2.32)$$

$$y_{rj}^{1k} \leq x_{rj}^1, \quad k \in V, j \in V \quad (2.33)$$

$$y_{ij}^{hk} \leq x_{ij}^h, \quad k \in V, (i,j) \in A, h = 2, \dots, L+1 \quad (2.34)$$

$$\sum_{e \in E} w_e = z \quad (2.35)$$

$$w_e \leq x_{ri}^1, \quad e = \{i, j\} \in E, i < j \quad (2.36)$$

$$w_e \leq x_{rj}^1, \quad e = \{i, j\} \in E, i < j \quad (2.37)$$

$$\sum_{h=2}^{L+1} \sum_{(i,j) \in A} c_{ij} x_{ij}^h + \sum_{e \in E} c_e w_e \leq B \quad (2.38)$$

$$s \geq \sum_{h=2}^{L+1} \sum_{(i,j) \in A} y_{ij}^{hk} + \sum_{j \in V} y_{rj}^{1k}, \quad k \in V \quad (2.39)$$

$$x_{rj}^1 \in \{0, 1\}, \quad (r, j) \in A \quad (2.40)$$

$$x_{ij}^h \in \{0, 1\}, \quad (i, j) \in A, i \neq r, h = 2, \dots, L+1 \quad (2.41)$$

$$w_e \in \{0, 1\}, \quad \forall e = \{i, j\} \in E, i < j \quad (2.42)$$

$$z \in \{0, 1\} \quad (2.43)$$

$$y_{rj}^{1k} \geq 0, \quad (r, j) \in A, k \in V \quad (2.44)$$

$$y_{ij}^{hk} \geq 0, \quad (i, j) \in A, i \neq r, k \in V, h = 2, \dots, L + 1 \quad (2.45)$$

$$s \geq 0. \quad (2.46)$$

A equação (2.27) possui o papel de identificar a paridade, par ou ímpar, da árvore geradora de G a ser obtida. Isso é feito em função de z e, neste caso, modelado na primeira camada do grafo D_c . Assim, para garantir que todo o fluxo que se origina de r passe pelos dois vértices conectados a ele, no caso ímpar (ou pelo vértice único, no caso par), introduzimos as restrições (2.28). As restrições (2.29), por sua vez, garantem que cada vértice de V será visitado uma única vez, em alguma camada de D_c . As equações (2.30), (2.31) e (2.32) estabelecem os fluxos viáveis de mercadorias em D_c . As restrições (2.30) garantem a conexão dos vértices da primeira camada com os da segunda. Já as equações (2.31) garantem a conservação de fluxos para os vértices da rede, da segunda camada em diante. Finalmente, as restrições (2.32) garantem que as mercadorias chegarão a seus respectivos destinos. As equações (2.33) e (2.34) fazem a conexão entre as variáveis de fluxos e as variáveis que definem a arborescência de Steiner. A restrição (2.35), por sua vez, identifica uma aresta central, no caso em que o diâmetro é ímpar. As restrições (2.36) e (2.37) impõem que, existindo uma aresta central, arcos da primeira camada de D_c apontando de r para as duas extremidades da aresta, façam parte da solução. A equação (2.38) calcula o custo total dos arcos da arborescência de Steiner e impõe à eles um limite de orçamento, B . As restrições (2.39) impõem um limite inferior ao valor de s , sendo este definido pelo número de arcos no maior caminho que leva do vértice artificial ao destino final de uma mercadoria. As restrições (2.40)-(2.43) impõem a integralidade das variáveis. Por fim, a obrigatoriedade de fluxos não-negativos é imposta por (2.44) e (2.45).

2.4 Formulações para o BMDSTP

Nesta seção, introduzimos três formulações matemáticas para o BMDSTP, todas elas baseadas no dígrafo $D = (V', A)$. Como definido anteriormente, o problema consiste em encontrar, sob uma restrição de orçamento para os arcos, uma arborescência de Steiner de diâmetro mínimo para D . Denotamos por $S_T \subseteq V$ os vértice obrigatórios de D e por $T = (V_T, A_T)$ uma arborescência de Steiner para o mesmo. Note, neste caso, que T deve necessariamente ter uma soma de custos de arcos que não ultrapasse B .

2.4.1 Formulação baseada nas restrições MTZ

Nossa primeira formulação para o BMDSTP é uma adaptação da formulação MTZ que introduzimos para o BDSTP. Para defini-la, primeiramente, devemos substituir as restrições (2.12) por

$$\sum_{(i,j) \in A} x_{ij} = 1, \quad j \in S_T. \quad (2.47)$$

Note, neste caso, que não precisamos incluir todos os vértices de D na solução, apenas os obrigatórios.

Outra imposição a considerar é o fato dos vértices opcionais, aqueles em $V \setminus S_T$, serem necessariamente *vértices de passagem*, quando utilizados. Ou seja, devemos impedi-los de serem folhas em T . Uma forma de atender a tal condição é

$$\sum_{(i,j) \in A} x_{ij} \leq \sum_{(j,i) \in A} x_{ji}, \quad j \in V \setminus S_T \quad (2.48)$$

que restringe a quantidade de arcos saindo de um vértice de $V \setminus S_T$ a ser sempre maior ou igual à quantidade de arcos entrando no mesmo.

Por fim, para impor conectividade entre o vértice artificial e os vértices obrigatórios de D , adicionamos à formulação desigualdades do tipo *cutsets*. Denotamos por $[V \setminus C, C]$ o corte de D tal que $r \in V \setminus C$ e $C \cap S_T \neq \emptyset$. Assim, garantimos que, se houver pelo menos um vértice obrigatório em C , então qualquer solução viável para o problema precisa conter pelo menos um arco no corte. Desta forma, adicionamos à formulação as seguintes restrições:

$$\sum_{i \in V \setminus C, j \in C} x_{ij} \geq 1 \quad r \in V \setminus C, C \cap S_T \neq \emptyset. \quad (2.49)$$

Note que (2.49) envolve uma quantidade exponencial de restrições. No próximo capítulo, indicamos como lidamos com elas.

2.4.2 Formulação multifluxos

Apresentamos a seguir uma formulação MCF para o BMDSTP. Esta, como veremos, consiste em uma adaptação direta de nossa formulação MCF para o BDSTP. Neste processo, consideramos apenas o envio de mercadorias para os vértices de S_T .

Outro ponto importante a considerar diz respeito à eventual utilização de *vértices opcionais*, pertencentes a $V \setminus S_T$, na arborescência de Steiner. A função objetivo do BMDSTP, vale notar, não restringe diretamente o uso de arcos com extremidades em $V \setminus S_T$. Isso é feito apenas de forma indireta, através da restrição de orçamento. Quando esta é *folgada*, arcos desnecessários podem eventualmente serem escolhidos, sem afetar o diâmetro mínimo. Para evitar tal situação, utilizamos então as

desigualdades

$$x_{ij} \leq \sum_{k \in S_T} y_{ij}^k, \quad (i, j) \in A. \quad (2.50)$$

Como resultado, garantimos que se existir um fluxo nulo passando por ele, o arco $(i, j) \in A$ não fará parte da arborescência. Adicionalmente, devemos assegurar que os vértices de $V \setminus S_T$ sejam vértices de passagem. Tal restrição, como indicamos anteriormente, é imposta através das desigualdades

$$\sum_{(i,j) \in A} x_{ij} \leq \sum_{(j,i) \in A} x_{ji}, \quad j \in V \setminus S_T. \quad (2.51)$$

2.4.3 Uma reformulação STP para o BMDSTP

Nossa próxima formulação será definida a partir de ajustes na formulação (2.27)-(2.46), proposta para o BDSTP. Semelhante ao que fizemos acima, é fundamental restringir o envio de mercadorias apenas para vértices obrigatórios k , $k \in S_T$. Adicionalmente, substituímos a equação (2.29) por

$$\sum_{h=2}^{L+1} \sum_{i:(i,j) \in A, i \neq r} x_{ij}^h + x_{rj}^1 = 1, \quad j \in S_T. \quad (2.52)$$

Note que, em (2.52), restringimos j aos vértices de S_T , de acordo com a definição do problema.

Devemos, também, restringir o uso indevido de vértices opcionais na solução, além de garantir que, quando utilizados, sejam vértices de passagem. Assim, adicionamos à formulação as seguintes restrições:

$$x_{ij}^h \leq \sum_{k \in S_T} y_{ij}^{hk}, \quad (i, j) \in A, h \in 2, \dots, L+1 \quad (2.53)$$

$$\sum_{i:(i,j) \in A} x_{ij}^h \leq \sum_{i:(j,i) \in A} x_{ji}^h, \quad j \in V \setminus S_T, h = 2, \dots, L+1. \quad (2.54)$$

2.5 Formulações para BDTSTP

Nesta seção, apresentamos formulações para o BDTSTP. Isto é feito através de ajustes em nossas formulações BMDSTP. Neste caso, exigimos que os vértices obrigatórios sejam necessariamente folhas da arborescência de Steiner. O objetivo, então, é encontrar uma FST de menor diâmetro possível, onde o custo total dos arcos não ultrapasse nosso orçamento, B .

Partindo de nossas formulações para o BMDSTP, eliminamos todas as variáveis \mathbf{x} associadas a arcos apontando pra fora de j , $j \in S_T$. Nos restringimos assim às

variáveis

$$0 \leq x_{ij} \leq 1 \quad \forall (i, j) \in A \setminus \{[(i, j) : i \in S_T, j \in V] \cup [(r, j) : j \in S_T]\}. \quad (2.55)$$

Como resultado desta redefinição de \mathbf{x} , os vértices de S_T serão, necessariamente, folhas de T . Complementando nossas formulações para o BDTSTP, removemos as variáveis que conectam o vértice artificial, r , diretamente aos vértices obrigatórios, impedindo que estes possam ser apontados por r na solução.

Capítulo 3

Resultados Computacionais

Neste capítulo, apresentamos os resultados computacionais associados à resolução das formulações propostas no capítulo anterior. O computador que utilizamos nos experimentos é equipado com um processador Intel(R) Core(TM) i7-8565U, com 1.80GHz de velocidade de processamento e memória RAM de 20GB DDR4. O sistema operacional empregado foi o *Linux*, na distribuição *elementary OS*, versão 5.1.7 Hera. Para implementação de algoritmos, utilizamos a linguagem de programação *Julia*, versão 1.4.1. Para todo o procedimento de resolução de problemas de Programação Inteira, utilizamos o *solver Gurobi*, versão 9.0.3, por meio da interface oferecida pelo pacote computacional *JuMP*. Utilizamos também pacotes computacionais como o *LightGraphs* [33], para manuseio de grafos e o *GraphPlot*, para a visualização dos resultados obtidos. O valor de L que utilizamos para todas as formulações foi de $\left\lceil \frac{|V|}{2} \right\rceil$, para qualquer instância. O tempo máximo de CPU permitido para resolver qualquer uma delas foi limitado a 1200 segundos. Além disso, os procedimentos de paralelismo do *Gurobi* foram desabilitados e utilizamos uma única *thread*.

Para simplificar a apresentação dos resultados, utilizamos uma nomenclatura padrão para identificar cada uma de nossas formulações. Exemplificando, BDSTP-MTZ corresponde à formulação do BDTSP baseada nas desigualdades MTZ. Para as demais formulações que se utilizam destas desigualdades, utilizamos BMDSTP-MTZ e BDTSTP-MTZ, respectivamente. Da mesma forma, utilizamos BDSTP-MCF, BMDSTP-MCF e BDTSTP-MCF para identificar nossas formulações multi-fluxos para os três problemas e BDSTP-NF, BMDSTP-NF e BDTSTP-NF, para as reformulações do problema como um STPG definido sobre um grafo em camadas.

3.1 Algoritmo Branch-and-Cut

Deixamos a cargo do *Gurobi* todo o gerenciamento da árvore de enumeração implícita por ele implementada. Ou seja, dentre outros, que heurística utilizar, a escolha

das variáveis de *branching* e a frequência da separação dos cortes que o mesmo disponibiliza para reforçar as formulações. Quando nossas formulações envolvem um número exponencial de desigualdades, como é o caso das formulações MTZ para o BMDSTP e BDTSTP, utilizamos o Algoritmo de Dinic [34] para resolver o Problema de Separação (PS) associado às desigualdades de *cutset*, (2.49). Tal problema, como iremos indicar, corresponde à resolução de uma série de problemas de fluxo máximo, definidos sobre um *grafo suporte* (GS), $\bar{D} = (\bar{V}', \bar{A})$, que se origina de $D = (V', A)$.

Para a definição de \bar{D} , assumamos que $\bar{\mathbf{x}}$ corresponde aos valores de \mathbf{x} em uma relaxação linear da formulação, BMDSTP-MTZ ou BDTSTP-MTZ, em um nó qualquer da árvore de enumeração implícita. No nó 0 desta árvore, as desigualdades (2.49) são inicialmente descartadas da formulação. Em nós subsequentes, aquelas já introduzidas anteriormente e que sejam *ativas*, são mantidas. GS, por sua vez, é construído associando-se a \bar{D} um arco $(i, j) \in \bar{A}$, para cada \bar{x}_{ij} não nulo. Além disso, tomamos $\bar{x}_{ij} > 0$ como sendo a capacidade do arco $(i, j) \in \bar{A}$, em \bar{D} . PS corresponde então a encontrar um corte mínimo em \bar{D} (vide o *max flow - min cut theorem*, [35]), separando $r \in \bar{V}'$ de cada vértice em S_T . Se o valor do corte é inferior a 1, uma *cutset* violada é identificada e agregada à relaxação linear, reforçando-a. A relaxação linear reforçada é então resolvida, em um procedimento iterativo de *geração de planos de corte*, até que inexistam desigualdades de *cutset* violadas na solução. Um pseudo-código desse algoritmo é apresentado a seguir.

Algorithm 3.1.1

```

1: procedure PS( $\bar{D}$ )
2:    $s \leftarrow r$ 
3:   for  $t \in S_T$  do
4:      $V \setminus C, C, \text{cap} = \text{mincut}(\bar{D}, s, t)$  ▷ cap = capacidade do corte
5:     if  $\text{cap} < 1$  then
6:        $\text{add\_cut}(\sum_{i \in V \setminus C, j \in C} x_{ij} \geq 1)$  ▷ Adiciona à desigualdade de cutset à formulação
7:     end if
8:   end for
9: end procedure

```

Uma forma de acelerar o procedimento discutido acima é adicionar *cortes reversos* ao mesmo. Para tanto, após sua aplicação, revertamos as orientações de todos os arcos do grafo suporte \bar{D} . Ou seja, um arco (i, j) , de capacidade \bar{x}_{ij} passa a ser o arco (j, i) , com capacidade \bar{x}_{ji} . Feito isso, identificamos os cortes mínimos separando a raiz, r , de cada vértice de S_T . Cortes com valor menor do 1 identificam, mais uma vez, desigualdades (2.49) violadas por $\bar{\mathbf{x}}$.

3.2 Instâncias

Em nossos testes computacionais, utilizamos três grupos distintos de instâncias associadas à cada problema aqui investigado. Como nenhum deles foi resolvido anteriormente, definimos tais instâncias.

Para o BDSTP, escolhemos grafos normalmente associados ao DCMST, como aqueles encontrados em [28]. Consideramos 19 grafos distintos, sendo 9 deles esparsos. Os grafos completos contém entre 10 e 25 vértices e foram gerados a partir de uma distribuição uniforme de pontos (vértices) em um quadrado de lado 100. A distância euclidiana de pontos é utilizada como o custo de um eventual arco/aresta entre eles. Os grafos esparsos, por sua vez, contém entre 20 a 60 vértices e são gerados como descrito em [36].

Para o BMDSTP, escolhemos um grupo de instâncias comumente associadas ao STPG, disponíveis na *OR-Library* [37]. Mais especificamente, selecionamos o grupo B de instâncias, correspondendo a 18 grafos esparsos de 50 a 100 vértices, com custos aleatórios de arestas.

Para o BDTSTP, diferentemente dos problemas anteriores, decidimos gerar instâncias específicas para o mesmo¹. Para tal, utilizamos a biblioteca *LightGraphs* do *Julia* para gerar grafos esparsos conexos com custos definidos aleatoriamente no intervalo de 1 a 50. Geramos, por sua vez, 9 instâncias associadas a grafos esparsos contendo entre 50 a 100 nós. Isso é feito porque várias das instâncias esparsas originalmente propostas para o STPG não contém FSTs.

3.3 Descrição dos resultados

Além do valor L , indicado anteriormente, um valor para B também corresponde a um dado da entrada para os problemas. Este, por sua vez, define um limite para os gastos por meio da escolha de arcos/arestas e, tem assim, um impacto direto na topologia de uma arborescência ótima. Desta forma, nossa escolha de valores de B levou em conta argumentos associados a cada tipo de árvore envolvida. Selecionamos os mesmos com o objetivo de caracterizar o comportamento de cada formulação em situações relevantes. Para cada instância, consideramos assim três valores distintos de B . Descrevemos, a seguir, como chegamos à esses valores.

Para o BDSTP, calculamos, inicialmente, a soma das $n - 1$ arestas de maior custo no grafo original, G . Escolhemos, então, três percentuais daquele valor para associar a B . Respectivamente, 20%, 30% e 40% do mesmo, quando os grafos são completos. Por sua vez, quando os grafos são esparsos, utilizamos 35%, 45% e 55% daquele valor.

¹Veja <https://gitlab.com/afazevedo29/instances>

Com relação ao BMDSTP, consideramos, para cada grafo, percentuais do valor correspondente à solução ótima do STPG para o mesmo. Desta forma, tomamos para B , inicialmente, 100% daquele valor. Isso é feito tendo em mente que inexistem arborescências de Steiner com custo menor. Para os dois valores adicionais de B , tomamos, respectivamente, 110% e 120% daquele valor.

Por fim, para o BDTSTP, eliminamos a restrição de orçamento e resolvemos o MDTSTP resultante. Feito isso, escolhemos os valores de B como sendo, respectivamente, 90%, 95% e 100% do valor obtido.

Os resultados de nossos experimentos computacionais são apresentados no formato de tabelas. Para cada problema investigado, três tabelas estão disponíveis, cada um delas correspondendo a um valor distinto de B .

Apresentamos os resultados computacionais obtidos nas Tabelas 3.1 a 3.12. Cada tabela é organizada contendo, em suas quatro primeiras colunas, dados das instâncias consideradas. Respectivamente, uma denominação para a instância, **Instância**, na primeira coluna, seguida de $|\mathbf{V}|$, $|\mathbf{A}|$ e **OPT**, o valor de sua solução ótima, quando obtida por algum algoritmo. A estas colunas se seguem 3 blocos de 4 colunas, um bloco para cada formulação considerada. Cada algoritmo, por sua vez, é identificado através da formulação matemática a ele associada. As colunas de cada bloco indicam, respectivamente, o número de nós, **Nós**, na árvore de enumeração, o *gap* percentual, **Gap(%)**, entre o valor de **OPT** e o valor da relaxação linear inicial, o tempo de CPU, **T(s)**, em segundos, gasto pelo algoritmo e o valor **Z** do melhor limite dual obtido para a instância. Quando um algoritmo não consegue oferecer um certificado de otimalidade, * aparece na posição do valor **Z**. Adicionalmente, **INF** aparece na posição de **OPT** quando inexistente uma solução viável para a instância.

3.4 Resultados computacionais para o BDSTP

Nesta seção, apresentamos resultados computacionais para os três algoritmos distintos associados às nossas formulações do BDSTP. Dividimos essa apresentação em dois blocos: um para grafos completos e outro para grafos esparsos.

3.4.1 Grafos Completos

Cada tabela está relacionada a um valor distinto de B . Neste caso específico, 20%, 30% e 40% do valor do somatório dos $n - 1$ arcos de maior custo, para cada grafo considerado.

Instância	V	OPT	BDSTP-MTZ				BDSTP-MCF				BDSTP-NF			
			Nós	Gap(%)	T(s)	Z	Nós	Gap(%)	T(s)	Z	Nós	Gap(%)	T(s)	Z
10_d4	10	INF	-	-	-	-	-	-	-	-	-	-	-	-
10_d5	10	INF	-	-	-	-	-	-	-	-	-	-	-	-
10_d6	10	INF	-	-	-	-	-	-	-	-	-	-	-	-
10_d7	10	INF	-	-	-	-	-	-	-	-	-	-	-	-
10_d8	10	INF	-	-	-	-	-	-	-	-	-	-	-	-
10_d10	10	INF	-	-	-	-	-	-	-	-	-	-	-	-
15_d4	15	7	59803	100	30,643	7	688	48,48	10,284	7	38	51,98	94,872	7
15_d8	15	4	4088	100	2,001	4	22	29,07	2,153	4	1	30,67	6,036	4
20_d4	20	6	655467	100	444,789	6	13.098	37,55	380,221	6	1	27,52	>1200	*5
20_d5	20	6	491783	100	427,220	6	1.508	34,16	85,315	6	1	36,95	370,052	6
20_d6	20	5	114564	100	78,229	5	595	28,05	42,991	5	1	29,91	53,716	5
20_d7	20	6	1,94E+05	100	188,035	6	582	32,76	38,096	6	1	35,18	210,451	6
20_d8	20	6	1,50E+06	100	>1200	*4	15.798	39,44	352,538	6	1	29,69	>1200	*5
20_d10	20	5	2036038	100	>1200	*4	14.013	31,13	457,084	5	3	32,93	423,182	5
25_d4	25	4	2,17E+06	100	>1200	*4	121	10,60	52,895	4	1	12,17	>1200	*4
25_d5	25	4	1,69E+05	100	184,844	4	849	16,42	118,826	4	1	17,06	111,442	4
25_d6	25	4	84.329	100	59,465	4	571	17,396	169,534	4	1	18,113	85,052	4
25_d8	25	5	1855616	100	>1200	*4	10202	18,542	>1200	*4,3	1	13,042	>1200	*4
25_d9	25	4	34400	100	37,483	4	1961	20,119	425,856	4	1	20,320	89,613	4

Tabela 3.1: Resultados computacionais para grafos completos: valor de B igual a 20% do somatório dos $n - 1$ arcos de maior custo.

Instância	V	OPT	BDSTP-MTZ				BDSTP-MCF				BDSTP-NF			
			Nós	Gap(%)	T(s)	Z	Nós	Gap(%)	T(s)	Z	Nós	Gap(%)	T(s)	Z
10_d4	10	4	3149	100	0,984	4	189	33,458	1,078	4	1	36,272	0,283	4
10_d5	10	4	1521	100	0,307	4	39	31,897	0,447	4	1	34,038	0,381	4
10_d6	10	4	1549	100	0,352	4	66	32,978	0,562	4	1	36,025	0,384	4
10_d7	10	5	1641	100	0,503	5	333	39,603	1,474	5	1	45,205	0,917	5
10_d8	10	4	1800	100	0,426	4	91	30,43	0,544	4	1	33,29	0,373	4
10_d10	10	5	896	100	0,308	5	159	40,32	0,730	5	1	44,21	0,702	5
15_d4	15	4	10202	100	5,763	4	2.315	36,18	14,596	4	1	36,53	3,264	4
15_d8	15	3	10533	100	5,550	3	422	31,02	4,853	3	1	31,24	3,3	3
20_d4	20	3	10240	100	8,095	3	166	16,10	10,489	3	1	16,10	15,919	3
20_d5	20	4	10781	100	10,519	4	4.704	34,55	78,887	4	1	34,55	22,010	4
20_d6	20	4	10202	100	7,738	4	17.503	35,87	219,465	4	1	35,87	21,015	4
20_d7	20	4	1,02E+04	100	6,048	4	9.753	35,85	113,994	4	1	35,85	25,793	4
20_d8	20	4	1,02E+04	100	9,168	4	14.600	36,98	137,555	4	1	36,98	24,033	4
20_d10	20	3	10305	100	7,782	3	425	16,65	14,847	3	1	16,65	21,688	3
25_d4	25	4	2,55E+04	100	54,677	4	33.009	26,76	>1200	*3,3	1	38,23	465,770	4
25_d5	25	3	1,02E+04	100	17,111	3	284	15,35	36,039	3	1	15,35	61,757	3
25_d6	25	3	10.221	100	13,393	3	695	20,760	107,232	3	1	20,760	381,080	3
25_d8	25	4	14193	100	35,021	4	54743	36,032	845,056	4	1	36,032	491,783	4
25_d9	25	4	14732	100	39,197	4	68659	31,844	>1200	*3,6	1	37,603	269,441	4

Tabela 3.2: Resultados computacionais para grafos completos: valor de B igual a 30% do somatório dos $n - 1$ arcos de maior custo.

Instância	V	OPT	BDSTP-MTZ				BDSTP-MCF				BDSTP-NF			
			Nós	Gap(%)	T(s)	Z	Nós	Gap(%)	T(s)	Z	Nós	Gap(%)	T(s)	Z
10_d4	10	3	931	100	0,446	3	158	27,523	0,570	3	1	28,340	0,270	3
10_d5	10	3	2694	100	0,580	3	116	30,150	0,548	3	1	30,375	0,206	3
10_d6	10	3	1104	100	0,421	3	144	31,531	0,509	3	1	31,706	0,234	3
10_d7	10	3	521	100	0,214	3	230	19,644	0,722	3	1	22,517	0,426	3
10_d8	10	3	1521	100	0,356	3	267	30,52	1,027	3	1	30,66	0,373	3
10_d10	10	3	563	100	0,395	3	89	21,88	0,507	3	1	23,36	0,353	3
15_d4	15	3	10215	100	5,980	3	388	29,67	3,499	3	1	29,67	2,673	3
15_d8	15	2	10430	100	3,052	2	1	5,16	0,811	2	1	5,16	1,382	2
20_d4	20	2	10234	100	5,057	2	1	0,76	1,387	2	1	0,76	12,601	2
20_d5	20	2	10265	100	8,325	2	1	0,11	1,346	2	1	0,11	12,008	2
20_d6	20	3	10257	100	8,975	3	924	31,05	26,484	3	1	31,05	19,020	3
20_d7	20	2	1,03E+04	100	6,848	2	1	0,29	1,305	2	1	0,29	13,405	2
20_d8	20	3	1,03E+04	100	8,714	3	726	32,62	13,307	3	1	32,62	15,340	3
20_d10	20	2	10216	100	7,069	2	1	0,24	1,572	2	1	0,24	11,780	2
25_d4	25	2	1,02E+04	100	11,620	2	1	0,78	0,986	2	1	0,78	62,085	2
25_d5	25	3	1,03E+04	100	9,157	3	1.133	29,67	45,123	3	1	29,67	62,794	3
25_d6	25	2	10.247	100	9,792	2	1	1,464	0,828	2	1	1,464	53,899	2
25_d8	25	3	10281	100	9,668	3	1533	31,036	100,075	3	1	31,036	47,415	3
25_d9	25	2	10241	100	10,448	2	1	0,208	1,602	2	1	0,208	64,091	2

Tabela 3.3: Resultados computacionais para grafos completos: valor de B igual a 40% do somatório dos $n - 1$ arcos de maior custo.

Como seria de se esperar, uma instância do BDSTP se torna mais fácil de resolver quando o valor de B é maior e isso se aplica a todos os algoritmos que testamos para o problema.

Quando comparada às outras formulações, o algoritmo associado à formulação BDSTP-MTZ tende a explorar uma maior quantidade de nós na árvore de enumeração implícita. Além disso, para todas as instâncias testadas, o valor de sua relaxação linear se manteve sempre no mínimo possível, ou seja, 0. Em termos de desempenho, obteve apenas 9 dos 13 certificados de otimalidade possíveis na Tabela 3.1. Por outro lado, para as instâncias das Tabelas 3.2 e 3.3, foi o algoritmo que atingiu o melhor resultado.

Em relação à formulação BDSTP-MCF, observamos que seu algoritmo correspondente alcançou o maior número de certificados de otimalidade, quando comparado aos demais. Mais especificamente, alcançou 48 dos 51 possíveis certificados em comparação com 47 dos outros modelos. Para valores maiores de B , *gaps* baixos foram obtidos para algumas instâncias da Tabela 3.3. Para as instâncias das Tabelas 3.2 e 3.1, podemos observar *gaps* de dualidade e tempos de CPU mais elevados, tomando por base o desempenho no outro caso. Entretanto, para as instâncias da Tabela 3.1, o algoritmo obteve, na maioria dos casos, o melhor desempenho.

Por fim, notamos que o algoritmo referente à formulação BDSTP-NF foi o que menos explorou nós na árvore de enumeração implícita. Mais precisamente, na maior parte dos casos, garantiu certificados de otimalidade explorando apenas um

único nó. No entanto, para valores menores de B , como visto na Tabela 3.1, obteve apenas 9 certificados de otimalidade para as 13 instâncias consideradas. Em termos de tempo de CPU, seu desempenho foi inferior às demais formulações, para a maioria das instâncias testadas. Com relação aos valores de **Gap**, observamos um comportamento similar ao do algoritmo correspondente à formulação BDSTP-MCF. Note que o número reduzido de nós na árvore de enumeração implícita se deve, neste caso, também ao *Gurobi*. Este conseguiu fechar, em alguns casos, *gaps* de dualidade expressivos.

3.4.2 Grafos Esparsos

A seguir, apresentaremos três tabelas ilustrando os resultados dos experimentos computacionais relativos a grafos esparsos. Como mencionado anteriormente, neste caso tomamos para B respectivamente 35%, 45% e 55% do valor da soma dos $n - 1$ arcos de maior custo no grafo.

Instância	V	A	OPT	BDSTP-MTZ				BDSTP-MCF				BDSTP-NF			
				Nós	Gap(%)	T(s)	Z	Nós	Gap(%)	T(s)	Z	Nós	Gap(%)	T(s)	Z
2050_d4	20	50	INF	-	-	-	-	-	-	-	-	-	-	-	-
2050_d5	20	50	6	48.482	100	22,276	6	139	25,68	2,482	6	1	31,51	4,881	6
2050_d6	20	50	6	13.820	100	9,277	6	716	32,82	7,556	6	1	35,98	6,499	6
2050_d7	20	50	7	24.048	100	16,279	7	176	37,22	2,085	7	1	40,25	8,965	7
2050_d8	20	50	INF	-	-	-	-	-	-	-	-	-	-	-	-
2050_d9	20	50	INF	-	-	-	-	-	-	-	-	-	-	-	-
40100_d4	20	50	6	2,02E+06	100	>1200	*4	12.033	32,31	245,696	6	1	32,47	1.066,9	6
40100_d5	40	100	5	2,11E+06	100	>1200	*4	256	17,97	15,588	5	1	18,22	223,404	5
40100_d6	40	100	5	1938935	100	836,085	5	7.748	24,46	274,277	5	17	24,46	800,559	5

Tabela 3.4: Resultados computacionais para grafos completos: valor de B igual a 35% do somatório dos $n - 1$ arcos de maior custo.

Instância	V	A	OPT	BDSTP-MTZ				BDSTP-MCF				BDSTP-NF			
				Nós	Gap(%)	T(s)	Z	Nós	Gap(%)	T(s)	Z	Nós	Gap(%)	T(s)	Z
2050_d4	20	50	5	11.150	100	5,019	5	289	27,08	5,172	5	1	27,96	7,819	5
2050_d5	20	50	4	1.382	100	0,473	4	83	17,99	2,291	4	1	18,37	2,214	4
2050_d6	20	50	4	1.346	100	0,465	4	87	19,43	2,688	4	1	20,50	2,491	4
2050_d7	20	50	4	521	100	0,440	4	70	19,81	2,470	4	1	21,35	2,159	4
2050_d8	20	50	4	998	100	0,439	4	18	12,20	1,765	4	1	14,69	2,577	4
2050_d9	20	50	4	998	100	0,434	4	18	12,20	1,780	4	1	14,69	2,592	4
40100_d4	20	50	4	2.610	100	1,847	4	58	21,30	3,816	4	1	21,30	114,767	4
40100_d5	40	100	4	977	100	1,156	4	56	23,75	3,902	4	1	23,75	102,565	4
40100_d6	40	100	4	2.359	100	1,988	4	159	22,11	3,683	4	1	22,11	109,414	4

Tabela 3.5: Resultados computacionais para grafos completos: valor de B igual a 45% do somatório dos $n - 1$ arcos de maior custo.

Instância	V	A	OPT	BDSTP-MTZ				BDSTP-MCF				BDSTP-NF			
				Nós	Gap(%)	T(s)	Z	Nós	Gap(%)	T(s)	Z	Nós	Gap(%)	T(s)	Z
2050_d4	20	50	4	1.575	100	0,523	4	115	27,92	1,893	4	1	27,97	2,534	4
2050_d5	20	50	4	2.156	100	0,578	4	401	29,10	2,646	4	1	29,10	2,198	4
2050_d6	20	50	4	1.396	100	0,469	4	146	30,09	1,351	4	1	30,36	2,874	4
2050_d7	20	50	4	1.089	100	0,633	4	155	33,93	1,058	4	1	34,32	2,481	4
2050_d8	20	50	4	2.815	100	0,557	4	99	26,29	1,101	4	1	26,29	1,981	4
2050_d9	20	50	4	2.815	100	0,575	4	99	26,29	1,116	4	1	26,29	2,283	4
40100_d4	20	50	4	951	100	1,070	4	390	30,96	9,878	4	1	30,96	98,785	4
40100_d5	40	100	4	585	100	0,872	4	486	33,36	9,469	4	1	33,36	91,911	4
40100_d6	40	100	4	528	100	0,707	4	434	31,35	8,052	4	1	31,35	105,286	4

Tabela 3.6: Resultados computacionais para grafos completos: valor de B igual a 55% do somatório dos $n - 1$ arcos de maior custo.

De forma semelhante ao que observamos pra grafos completos, diâmetros maiores são obtidos para valores relativamente mais baixos de B . Além disso, os algoritmos mostram, como seria de se esperar, maior dificuldade na obtenção de certificados de otimalidade considerando esses valores a B .

Notamos que o algoritmo associado à formulação BDSTP-MTZ continua a ser o que mais explora nós na árvore de enumeração implícita, além de manter suas relaxações lineares iniciais no menor valor possível, ou seja, 0. Em termos de CPU, seu desempenho se mostrou superior aos demais algoritmos quando considerados valores maiores de B . Por outro lado, para as instâncias da Tabela 3.4, apresentou maior dificuldade em obter certificados de otimalidade.

Diferentemente do caso anterior, o algoritmo associado à formulação BDSTP-MCF obteve tempos de CPU relativamente baixos em todos os casos considerados. Apesar de apresentar valores de *gaps* relativamente altos e do grande número de nós explorados na árvore de enumeração implícita, é o algoritmo de melhor desempenho na obtenção de certificados de otimalidade.

Por sua vez, o algoritmo associado à formulação BDSTP-NF, continuou resolvendo quase todas as instâncias em apenas um único nó da árvore de enumeração. Seus tempos de de CPU, no entanto, são relativamente altos. Vale aqui ressaltar, mais uma vez, o desempenho dos procedimentos de pré-processamento e fortalecimento de formulações do *Gurobi*. Estes conseguiram fechar em apenas um único nó da árvore de enumeração, *gaps* de dualidade relativamente altos.

3.5 Resultados Computacionais para o BMDSTP

Nesta seção, apresentamos resultados computacionais para os algoritmos associados à nossas formulações para o BMDSTP. Consideramos, mais uma vez, três valores distintos de B . Estes são definidos como percentuais do valor da solução ótima

correspondente ao STPG, para cada grafo considerado. Mais precisamente, tomamos para B 100%, 110% e 120% daquele valor.

Instancia	V	A	OPT	BMDSTP-MTZ				BMDSTP-MCF				BMDSTP-NF			
				Nós	Gap(%)	T(s)	Z	Nós	Gap(%)	T(s)	Z	Nós	Gap(%)	T(s)	Z
steinb1	50	63	9	385	100	0,439	9	15	39,87	1,858	9	13	39,87	115,232	9
steinb2	50	63	11	392	100	0,544	11	79	44,91	2,148	11	1	50,82	>1200	5,41
steinb3	50	63	14	580	100	0,654	14	89	46,34	3,695	14	1	57,07	>1200	*6,01
steinb4	50	100	11	235	100	0,609	11	66	54,33	3,298	11	1	33,45	>1200	*7,32
steinb5	50	100	10	788	100	1,07	10	71	44,93	3,51	10	1	50,20	>1200	*4,98
steinb6	50	100	15	1493	100	2,42	15	112	49,89	12,89	15	1	63,33	>1200	*5,5
steinb7	75	94	11	1541	100	2,44	11	58	45,50	3,78	11	1	33,18	>1200	*7,35
steinb8	75	94	14	571	100	1,22	14	82	55,43	7,08	14	0	-	>1200	-
steinb9	75	94	17	1503	100	4,48	17	137	49,19	33,75	17	0	-	>1200	-
steinb10	75	150	11	537	100	1,30	11	59	48,38	6,05	11	0	-	>1200	-
steinb11	75	150	13	596	100	1,64	13	105	55,52	27,23	13	0	-	>1200	-
steinb12	75	150	17	2561	100	10,06	17	284	54,30	94,32	17	0	-	>1200	-
steinb13	100	125	11	1936	100	8,36	11	54	34,57	11,97	11	0	-	>1200	-
steinb14	100	125	14	14258	100	74,14	14	91	41,05	29,25	14	0	-	>1200	-
steinb15	100	125	18	5143	100	31,26	18	214	48,67	111,59	18	0	-	>1200	-
steinb16	100	200	15	9683	100	51,34	15	143	55,60	35,70	15	0	-	>1200	-
steinb17	100	200	13	3.535	100	19,098	13	98	37,99	70,250	13	0	-	>1200	-
steinb18	100	200	24	2721	100	16,964	24	177	58,32	512,138	24	0	-	>1200	-

Tabela 3.7: Resultados computacionais para instâncias com B igual a 100% do valor da solução ótima do STPG.

Instancia	V	A	OPT	BMDSTP-MTZ				BMDSTP-MCF				BMDSTP-NF			
				Nós	Gap(%)	T(s)	Z	Nós	Gap(%)	T(s)	Z	Nós	Gap(%)	T(s)	Z
steinb1	50	63	8	1231	100	1,039	8	9	32,35	2,393	8	6	32,353	442,444	8
steinb2	50	63	9	2964	100	6,312	9	96	34,20	2,508	9	19	34,244	799,614	9
steinb3	50	63	9	5625	100	7,217	9	19	25,55	7,457	9	1	25,625	650,421	9
steinb4	50	100	8	2524	100	8,482	8	130	41,63	4,820	8	1	41,80	>1200	*4,65
steinb5	50	100	7	17614	100	40,77	7	105	26,07	4,93	7	1	26,21	>1200	*5,16
steinb6	50	100	10	138175	100	304	10	609	39,83	120,67	10	1	40,31	>1200	*5,96
steinb7	75	94	8	3364	100	9,06	8	25	26,76	4,68	8	0	-	>1200	-
steinb8	75	94	9	8974	100	29,18	9	75	33,07	10,84	9	0	-	>1200	-
steinb9	75	94	11	58780	100	150	11	25	26,22	36,23	11	0	-	>1200	-
steinb10	75	150	8	9510	100	53,16	8	221	33,29	23,93	8	0	-	>1200	-
steinb11	75	150	8	35649	100	148,82	8	147	33,54	72,39	8	0	-	>1200	-
steinb12	75	150	9	1	100	>1200	*0	1	30,21	>1200	*6,28	0	-	>1200	-
steinb13	100	125	9	76727	100	285,93	9	28	22,26	16,07	9	0	-	>1200	-
steinb14	100	125	11	1	100	>1200	*0	52	28,48	71,15	11	0	-	>1200	-
steinb15	100	125	11	1	100	>1200	*0	60	20,54	163,51	11	0	-	>1200	-
steinb16	100	200	8	1	100	>1200	*0	67	23,39	55,64	8	0	-	>1200	-
steinb17	100	200	10	1	100	>1200	*0	361	32,93	869,556	10	0	-	>1200	-
steinb18	100	200	11	1	100	>1200	*0	1	33,69	>1200	*7,29	0	-	>1200	-

Tabela 3.8: Resultados computacionais para instâncias com B igual a 110% do valor da solução ótima do STPG.

Instancia	V	A	OPT	BMDSTP-MTZ				BMDSTP-MCF				BMDSTP-NF			
				Nós	Gap(%)	T(s)	Z	Nós	Gap(%)	T(s)	Z	Nós	Gap(%)	T(s)	Z
steinb1	50	63	8	936	100	1,651	8	14	32,35	1,573	8	7	32,353	316,912	8
steinb2	50	63	9	10251	100	21,109	9	35	34,20	2,574	9	1	34,24	1200	*5,91
steinb3	50	63	9	2646	100	6,742	9	11	25,55	9,282	9	1	25,625	138,53	9
steinb4	50	100	7	19940	100	50,056	7	976	34,87	18,376	7	1	43,13	1200	*4,54
steinb5	50	100	7	42070	100	121	7	204	28,11	11,16	7	1	28,35	1200	*5,01
steinb6	50	100	8	1	100	1200	*0	895	30,99	338,39	8	1	44,86	1200	*5,51
steinb7	75	94	8	6138	100	14,75	8	31	26,76	6,55	8	0	-	1200	-
steinb8	75	94	8	10514	100	32,72	8	29	24,70	9,72	8	0	-	1200	-
steinb9	75	94	11	9718	100	38	10	33	26,22	42,28	11	0	-	1200	-
steinb10	75	150	7	110736	100	560,78	7	50	25,69	18,17	7	0	-	1200	-
steinb11	75	150	7	1	100	1200	*0	95	26,25	51,16	7	0	-	1200	-
steinb12	75	150	9	1	100	1200	*0	0	35,22	1200	*5,83	0	-	1200	-
steinb13	100	125	9	1	100	1200	*0	24	22,26	11,23	9	0	-	1200	-
steinb14	100	125	10	1	100	1200	*0	198	21,40	90,25	10	0	-	1200	-
steinb15	100	125	11	1	100	1200	*0	90	20,54	398,95	11	0	-	1200	-
steinb16	100	200	8	1	100	1200	*0	337	26,01	244,91	8	0	-	1200	-
steinb17	100	200	10	1	100	1200	*0	0	36,40	1200	*6,36	0	-	1200	-
steinb18	100	200	11	1	100	1200	*0	0	37,27	1200	*6,9	0	-	1200	-

Tabela 3.9: Resultados computacionais para instâncias com B igual a 120% do valor da solução ótima do STPG.

Com base nos resultados apresentados acima, notamos uma maior dificuldade na resolução das instâncias, quando comparado ao BDSTP. Ao contrário do que ocorreu para aquele problema, aqui, a medida que B aumenta, nota-se uma maior dificuldade na obtenção de certificados de otimalidade. Tal comportamento não é intuitivo e não encontramos explicações para ele.

De modo geral, o algoritmo associado à formulação BMDSTP-MTZ conseguiu obter poucos certificados de otimalidade. Além disso, seu valores de *gaps* ficaram, novamente, no maior patamar possível, além de explorar uma maior quantidade de nós na árvore de enumeração implícita. No entanto, para valores menores de B , apresentou melhores tempos de CPU, como é o caso das instâncias associadas a Tabela 3.7. Para instâncias das Tabelas 3.8 e 3.9, no entanto, observamos uma dificuldade maior para obter certificados de otimalidade em tempos de CPU aceitáveis.

Por sua vez, o algoritmo referente à formulação BMDSTP-MCF, obteve, em geral, um melhor desempenho em termos de CPU. Embora seu valores de **Gap** sejam elevados, foi o algoritmo que conseguiu o maior número de certificados de otimalidade, quando comparado aos outros. Em especial, para as instâncias da Tabela 3.9, onde o algoritmo conseguiu obter 15 dos 18 certificados de otimalidade possíveis.

Finalmente, quando observamos os resultados associados à formulação BMDSTP-NF, notamos um desempenho relativamente fraco para seu algoritmo correspondente. De modo geral, são poucas as instâncias resolvidas à otimalidade. Pior

ainda, para a maioria das instâncias propostas, não foi possível obter nem mesmo sua relaxação linear no tempo limite estipulado. Além disso, ao contrário do que prevaleceu para o BDSTP, o algoritmo explora um maior número de nós na árvore de enumeração implícita. Por fim, seus valores de **Gap** se mostraram elevados, para todas as instâncias onde foi possível obtê-lo.

3.6 Resultados Computacionais para o BDTSTP

Nesta seção, apresentamos os resultados obtidos para os algoritmos associados as nossas formulações apresentadas para o BDTSTP. Neste caso, resolvemos inicialmente o MDTSTP e consideramos três valores distintos para B . Como mencionado anteriormente, estes correspondem a respectivamente 90%, 95% e 100% do valor da solução ótima para o MDTSTP.

Instancia	V	A	OPT	BDTSTP-MTZ				BDTSTP-MCF				BDTSTP-NF			
				Nós	Gap(%)	T(s)	Z	Nós	Gap(%)	T(s)	Z	Nós	Gap(%)	T(s)	Z
fst_50_100_10	50	100	9	2847	77,78	6,016	9	110	41,21	2,710	9	1	41,22	>1200	*5,29
fst_50_100_11	50	100	8	35229	75	49,518	8	160	30,58	4,472	8	1	30,63	>1200	*5,55
fst_50_100_12	50	100	8	18305	75	36	8	65	28,79	2,923	8	1	28,88	>1200	*5,69
fst_75_200_10	50	200	7	320661	71,43	>1200	*4	157	31,55	25,466	7	0	-	>1200	-
fst_75_200_11	50	200	7	240834	71,43	>1200	*2,40	250	30,59	32,38	7	0	-	>1200	-
fst_75_200_12	50	200	7	134472	71,43	>1200	*2,46	167	30,09	23,17	7	0	-	>1200	-
fst_100_300_10	80	300	8	148587	75	>1200	*6	881	36,84	92	8	0	-	>1200	-
fst_100_300_11	80	300	7	131950	71,43	>1200	*3,35	328	29,98	116,39	7	0	-	>1200	-
fst_100_300_12	80	300	7	121241	71,43	>1200	*4	293	29,75	149,73	7	0	-	>1200	-

Tabela 3.10: Resultados computacionais para B igual a 90% do valor da solução ótima do MDTSTP.

Instancia	V	A	OPT	BDTSTP-MTZ				BDTSTP-MCF				BDTSTP-NF			
				Nós	Gap(%)	T(s)	Z	Nós	Gap(%)	T(s)	Z	Nós	Gap(%)	T(s)	Z
fst_50_100_10	50	100	7	3166	71,43	18,005	7	102	25,12	4,120	7	1	25,14	>1200	*5,24
fst_50_100_11	50	100	7	72534	71,43	126,057	7	39	21,57	2,633	7	25	21,57	964	7
fst_50_100_12	50	100	7	50049	71,43	71	7	149	18,62	7,472	7	1	18,71	>1200	*5,69
fst_75_200_10	50	200	7	207099	71,43	>1200	*2,38	424	32,33	40,894	7	0	-	>1200	-
fst_75_200_11	50	200	7	131003	71,43	>1200	*2,10	279	31,66	37,73	7	0	-	>1200	-
fst_75_200_12	50	200	7	177810	71,43	>1200	*2,12	732	31,16	76,33	7	0	-	>1200	-
fst_100_300_10	80	300	8	172481	75	>1200	*3,68	1540	38,82	534,88	8	0	-	>1200	-
fst_100_300_11	80	300	7	166477	71,43	>1200	*2,51	498	30,50	184,09	7	0	-	>1200	-
fst_100_300_12	80	300	7	142832	71,43	>1200	*2	338	30,05	179,51	7	0	-	>1200	-

Tabela 3.11: Resultados computacionais para B igual a 95% do valor da solução ótima do MDTSTP.

Instancia	V	A	OPT	BDTSTP-MTZ				BDTSTP-MCF				BDTSTP-NF			
				Nós	Gap(%)	T(s)	Z	Nós	Gap(%)	T(s)	Z	Nós	Gap(%)	T(s)	Z
fst_50_100_10	50	100	6	24028	66,67	36,155	6	34	12,75	3,681	6	11	12,75	399,372	6
fst_50_100_11	50	100	6	88395	66,67	114,154	6	24	8,52	4,055	6	15	8,52	221	6
fst_50_100_12	50	100	6	68792	66,67	168	6	1	5,05	0,859	6	1	5,05	33	6
fst_75_200_10	50	200	6	169272	66,67	>1200	*2	411	21,62	35,718	6	0	-	>1200	-
fst_75_200_11	50	200	6	215627	66,67	>1200	*2	71	21,03	18,68	6	0	-	>1200	-
fst_75_200_12	50	200	6	172700	66,67	>1200	*2,04	627	20,79	118,61	6	0	-	>1200	-
fst_100_300_10	80	300	6	116940	66,67	>1200	*2,38	323	19,39	89,67	6	0	-	>1200	-
fst_100_300_11	80	300	6	130233	66,67	>1200	*2	136	19,14	79,79	6	0	-	>1200	-
fst_100_300_12	80	300	6	99603	66,67	>1200	*2,05	101	18,39	63,92	6	0	-	>1200	-

Tabela 3.12: Resultados computacionais para B igual a 100% do valor da solução ótima do MDTSTP..

Para o BDTSTP, de modo geral, os algoritmos enfrentaram maior dificuldade em obter certificados de otimalidade. Da mesma forma, como observado anteriormente para o BMDSTP, a medida que B aumenta, maior a dificuldade dos algoritmos.

Em particular, o algoritmo associado à formulação BDTSTP-MTZ apresentou bastante dificuldade em obter certificados de otimalidade. Para todos os casos de B considerados, conseguiu alcançar esse objetivo para três instâncias apenas. No entanto, apesar de apresentar valores de **Gap** altos, estes foram significativamente melhores do que ocorreu para os problemas anteriores. Continuou sendo, entretanto, o algoritmo que mais explora nós na árvore de enumeração implícita, principalmente para as instâncias das Tabelas 3.11 e 3.12.

Por sua vez, o algoritmo associado à formulação BDTSTP-MCF conseguiu obter certificados de otimalidade para todas as instâncias testadas, independente de seus valores de B . Em especial, os tempos de CPU para valores menores de B , como é o caso das instâncias da Tabela 3.10, foram relativamente baixos. No entanto, os valores de **Gap** para a formulação se mantiveram relativamente altos, principalmente para as instâncias das Tabelas 3.10 e 3.11.

Por fim, o algoritmo associado à formulação BDTSTP-NF apresentou não só uma grande dificuldade em obter certificados de otimalidade como, pior ainda, não conseguiu nem mesmo resolver a relaxação linear inicial para diversas instâncias. A título de exemplo, para o valor de B correspondente à Tabela 3.10, não conseguiu obter nenhum certificado. Observamos ainda, em relação aos outros problemas considerados, que o algoritmo explorou mais nós na árvore de enumeração implícita. Da mesma forma, os valores referentes ao **Gap** se mantiveram relativamente altos e muito próximos àquele atingidos pela formulação BDTSTP-MCF.

Capítulo 4

Conclusões e Trabalhos Futuros

Nesta dissertação introduzimos as primeiras formulações propostas para o BDSTP, o BMDSTP e o BDTSTP. Estes problemas, embora ainda pouco investigados na literatura, são desafiadores e têm um grande potencial de aplicações práticas. Particularmente no desenho de redes de telecomunicação. Não se impõe de antemão a paridade das árvores ótimas desses problemas, sendo este o principal desafio para formulá-los. Tratamos esta questão de forma indireta, através de uma formulação para a Propriedade da Centralidade de um Vértice ou Aresta (PCVA) de uma árvore. Em seguida, incluímos a restrição de orçamento e incorporamos a PCVA a formulações conhecidas de problemas envolvendo a obtenção de árvores sob alguma restrição de diâmetro. Fazendo os ajustes necessários, chegamos as nossas três formulações matemáticas para cada um dos três problemas. Feito isso, utilizamos o solver *Gurobi* para resolver de forma exata cada uma dessas formulações. Neste processo, conduzimos um estudo computacional comparando o desempenho das formulações/algoritmos para diversos valores de B .

Através dos experimentos computacionais, ressaltamos o impacto da restrição de orçamento na resolução exata das formulações propostas. Para nossa surpresa, contrariando o senso comum, restrições de orçamento *mais folgadas* não levaram, necessariamente, a soluções *mais fáceis*. Enumeramos, a seguir, alguns pontos a destacar em nossos experimentos computacionais.

1. Para o BDSTP, tanto para grafos completos quanto para grafos esparsos, observamos uma dificuldade maior dos algoritmos em obter certificados de otimalidade quando os valores de B diminuem. Isto, de certa forma, está de acordo com o senso comum, já que, diante de um orçamento infinito, o problema se torna polinomial. Por outro lado, para o BMDSTP e o BDTSTP, observamos o efeito contrário e não encontramos explicações plausíveis para esse comportamento.
2. Em geral, os algoritmos associados às formulações que envolvem restrições

MTZ exploraram uma maior quantidade de nós na árvore de enumeração implícita. Este é um comportamento esperado, já que tais formulações são reconhecidamente *fracas*. Além disso, observamos que, na maioria das vezes, suas relaxações lineares se mantiveram no menor valor possível, ou seja, 0. Apesar dessa limitação, o algoritmo para o BDSTP apresentou um desempenho relativamente bom, em termos de tempo de CPU. Alternativamente, para os dois problemas associados a árvores de Steiner, os algoritmos tiveram um desempenho relativamente fracos.

3. De modo geral, os algoritmos associados a formulações MCF foram os que apresentaram o melhor desempenho. Em particular para os problemas envolvendo árvores de Steiner. As formulações MCF, vale aqui notar, apresentam valores de **Gap** relativamente bons e demandam um esforço computacional aceitável para obtê-los.
4. Os algoritmos associados a formulações baseadas em grafos em camadas, apresentaram, de modo geral, os piores tempos de CPU. Em alguns casos, não foi possível resolver nem mesmo a relaxação linear de suas formulações. De qualquer forma, um ponto a destacar em relação a tais formulações é o fato de serem as que mais se beneficiaram dos módulos de pré-processamento e reforço de formulações do *Gurobi*. Isto pode ser constatado pelo fato do *solver* conseguir fechar, em apenas um único nó na árvore de enumeração, *gaps* de dualidade expressivos.

Devido ao curto espaço de tempo disponível, não foi possível investigar estratégias para melhorar a performance dos algoritmos e/ou reforçar as formulações. Isto será efetuado posteriormente.

Como proposta de trabalhos futuros, pretendemos investigar, dentre outras:

- **Heurísticas e Meta-heurísticas:** Não encontramos nenhuma abordagem desse tipo para os problemas aqui investigados. Vale ressaltar que nossas formulações e algoritmos exatos têm muito a se beneficiar de soluções heurísticas de boa qualidade. Em particular, para *apertar* o valor de L , um dado de entrada para nossas formulações.
- **Testes de Pré-Processamento:** Procedimentos para eliminar arestas/arcs desnecessários podem levar a um salto de qualidade em nossas formulações/algoritmos.
- **Métodos de Decomposição:** Para o BDSTP, nos interessa em investigar procedimentos de *Relaxação Lagrangeana* e *Decomposição de Benders*. Note que o BDSTP se presta naturalmente à aplicação de métodos de decomposição.

Isto porque pode ser visto como uma interseção de um problema polinomial com um problema pseudo-polinomial. O problema da árvore geradora de diâmetro mínimo, sob uma definição mais geral de diâmetro, sendo o primeiro problema e o problema da mochila 0-1, sendo o segundo.

- **Reforço das desigualdades de orçamento:** As desigualdades de orçamento correspondem a desigualdades da mochila 0-1. Reforçar nossas formulações através de *Lifted Minimum Covers* dessas desigualdades seria algo extremamente desejável.
- **Atualização do valor de L :** Algumas de nossas formulações podem ser facilmente reforçadas ao longo de algoritmos de enumeração implícita, sempre que uma nova solução viável nos leve a uma redução no valor de L .

Finalmente, expandindo o tema central que aqui investigamos, pretendemos também trabalhar com os seguintes problemas:

- **Budget Constrained Minimum Radius Trees:** Uma adaptação natural dos problemas que aqui estudamos, buscando encontrar árvores de menor raio possível restritas a um orçamento B .
- **Minimum Diameter p -Forest Problem:** Problema introduzido em [15], que é uma extensão natural dos nossos problemas, onde busca-se encontrar p florestas de diâmetro mínimo. Os autores demonstram a equivalência deste problema com o *p -Center Problem* [14].
- **Minimum Diameter Dominating Tree:** Problema ainda não investigado na literatura, onde desejamos encontrar uma *árvore dominante* de G com menor diâmetro possível.

Referências Bibliográficas

- [1] OBRUČA, A. K. “Spanning tree manipulation and the travelling salesman problem”, *Computer Journal*, v. 10, n. 4, pp. 374–377, feb 1968.
- [2] GRAHAM, R. L., HELL, P. “On the History of the Minimum Spanning Tree Problem”, *Annals of the History of Computing*, v. 7, n. 1, pp. 43–57, 1985.
- [3] SCHRIJVER, A. “On the History of Combinatorial Optimization (Till 1960)”. In: *Handbooks in Operations Research and Management Science*, v. 12, pp. 1–68, Elsevier, jan 2005.
- [4] BORŮVKA, O. “Příspěvek k otázce ekonomické stavby elektrovodných sítí (Contribution to the solution of a problem of economical construction of electrical networks)”, *Elektrotechnický obzor 15*, pp. 153–154, 1926.
- [5] BORŮVKA, O. “jistém problému minimálním (About a certain minimum problem)”, *Práce Moravské přírodovědecké společnosti*, v. sv. III, spis 3, pp. 37–58, 1926.
- [6] JARNÍK, V. “jistém problému minimálním (About a certain minimum problem)”, *Práce Moravské Přírodovědecké Společnosti*, v. 6, pp. 57–63, 1930.
- [7] KRUSKAL, J. B. “On the Shortest Spanning Subtree of a Graph and the Travelling Salesman Problem”, *Proceedings of the American Mathematical Society*, v. 7, n. 1, pp. 48, feb 1956.
- [8] PRIM, R. C. “Shortest Connection Networks And Some Generalizations”, *Bell System Technical Journal*, v. 36, n. 6, pp. 1389–1401, 1957.
- [9] SZWARCFITER, J. *Teoria Computacional De Grafos*. ELSEVIER (MEDICINA), 2018.
- [10] CHERITON, D., TARJAN, R. E. “Finding Minimum Spanning Trees”, *SIAM Journal on Computing*, v. 5, n. 4, pp. 724–742, dec 1976.
- [11] CAMPÊLO, M., ANDRADE, R., DIAS, F., et al. “Problema de Árvore Geradora Mínima com Restrição de Grau Mínimo e Centrais Fixos”. 09 2013.

- [12] EDELSON, W., GARGANO, M. “Leaf constrained minimal spanning trees solved by a genetic algorithm”, *Congressus Numerantium*, 01 2002.
- [13] PLESNIK, J. “The complexity of designing a network with minimum diameter”, *Networks*, v. 11, n. 1, pp. 77–85, 1981.
- [14] HAKIMI, S. L., SCHMEICHEL, E. F., PIERCE, J. G. “On p-Centers in Networks”, *Transportation Science*, v. 12, n. 1, pp. 1–15, feb 1978.
- [15] HASSIN, R., TAMIR, A. “On the minimum diameter spanning tree problem”, *Information Processing Letters*, v. 53, n. 2, pp. 109–111, jan 1995.
- [16] HO, J. M., LEE, D. T., CHANG, C. H., et al. “Minimum diameter spanning trees and related problems”, *SIAM Journal on Computing*, v. 20, n. 5, pp. 987–997, 1991.
- [17] MACULAN, N. “The Steiner Problem in Graphs”, *Annals of Discrete Mathematics*, v. 31, pp. 185 – 212, 1987.
- [18] GOEMANS, M. X., MYUNG, Y.-S. “A catalog of steiner tree formulations”, *Networks*, v. 23, n. 1, pp. 19–28, jan 1993.
- [19] KARP, R. M. “Reducibility among Combinatorial Problems”. In: *Complexity of Computer Computations*, Springer US, pp. 85–103, 1972.
- [20] DING, W., QIU, K. “Algorithms for the minimum diameter terminal Steiner tree problem”, *Journal of Combinatorial Optimization*, v. 28, n. 4, pp. 837–853, jan 2014.
- [21] LU, C. L., TANG, C. Y., LEE, R. C. T. “The full Steiner tree problem”, *Theoretical Computer Science*, v. 306, n. 1-3, pp. 55–67, sep 2003.
- [22] LIN, G., XUE, G. “On the terminal Steiner tree problem”, *Information Processing Letters*, v. 84, n. 2, pp. 103–107, oct 2002.
- [23] HANDLER, G. Y. “Minimax Location of a Facility in an Undirected tree graph”, *Transportation Science*, v. 7, n. 3, pp. 287–293, aug 1973.
- [24] ACHUTAN, N. R., CACCETTA, L. “Minimum Weight Spanning Trees with Bounded Diameter”, *Australasian Journal of Combinatorics*, v. 5, pp. 261–276, 1990.
- [25] DAHL, G., GOUVEIA, L., REQUEJO, C. “On Formulations and Methods for the Hop-Constrained Minimum Spanning Tree Problem”. pp. 493–515, 01 2006.

- [26] ACHUTAN, N. R., CACCETTA, L. “Computational methods for the diameter restricted minimum weight spanning tree”, *Australasian Journal of Combinatorics*, pp. 51–61, 1994.
- [27] GOUVEIA, L., MAGNANTI, T. L. “Network Flow Models for Designing Diameter-Constrained Minimum-Spanning and Steiner Trees”, *Networks*, v. 41, n. 3, pp. 159–173, may 2003.
- [28] DOS SANTOS, A. C., LUCENA, A., RIBEIRO, C. C. “Solving diameter constrained minimum spanning tree problems in dense graphs”, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 3059, pp. 458–467, 2004. ISSN: 16113349.
- [29] GOUVEIA, L., SIMONETTI, L., UCHOA, E. “Modeling hop-constrained and diameter-constrained minimum spanning tree problems as Steiner tree problems over layered graphs”, *Mathematical Programming*, v. 128, n. 1-2, pp. 123–148, jun 2011.
- [30] DUNNING, I., HUCHETTE, J., LUBIN, M. “JuMP: A Modeling Language for Mathematical Optimization”, *SIAM Review*, v. 59, n. 2, pp. 295–320, 2017.
- [31] MILLER, C. E., ZEMLIN, R. A., TUCKER, A. W. “Integer Programming Formulation of Traveling Salesman Problems”, *Journal of the ACM (JACM)*, v. 7, n. 4, pp. 326–329, oct 1960.
- [32] GOUVEIA, L., SIMONETTI, L., UCHOA, E. *Modelling hop-constrained and diameter-constrained minimum spanning tree problems as Steiner tree problems over layered graphs*. Relatório técnico, 2008.
- [33] SETH BROMBERGER, J. F., OTHER CONTRIBUTORS. “Julia-Graphs/LightGraphs.jl: an optimized graphs package for the Julia programming language”. 2017.
- [34] DINIC, E. A. “Algorithm for solution of a problem of maximum flow in networks with power estimation”. 1970.
- [35] WOLSEY, L. A. *Integer programming*. Wiley, 1998.
- [36] GOUVEIA, L., MAGNANTI, T. “Modelling and solving the diameter-constrained minimum spanning tree problem”. In: *Fifth INFORMS Telecommunications Conference*, 2000.

- [37] BEASLEY, J. "OR-Library: Distributing Test Problems by Electronic Mail."
J Oper Res Soc 41, 1990.