



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
DEPARTAMENTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO

GABRIEL HELLEN DE SOUSA

PROBLEMA DA ÁRVORE t -SPANNER DE CUSTO MÍNIMO

FORTALEZA

2021

GABRIEL HELLEN DE SOUSA

PROBLEMA DA ÁRVORE t -SPANNER DE CUSTO MÍNIMO

Dissertação apresentada ao Programa de Pós-Graduação em Ciências da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de Mestre em Ciência da Computação. Área de concentração: Teoria da Computação.

Orientador: Prof. Dr. Manoel Bezerra Campêlo Neto.

FORTALEZA

2021

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

S696p Sousa, Gabriel Hellen de.
Problema da árvore t-spanner de custo mínimo / Gabriel Hellen de Sousa. – 2021.
91 f. : il.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2021.

Orientação: Prof. Dr. Manoel Bezerra Campêlo Neto.

1. Árvore t-spanner de custo mínimo. 2. Formulação matemática. 3. Branch-and-bound. 4. Programação linear inteira. I. Título.

CDD 005

GABRIEL HELLEN DE SOUSA

PROBLEMA DA ÁRVORE t -SPANNER DE CUSTO MÍNIMO

Dissertação apresentada ao Programa de Pós-Graduação em Ciências da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de Mestre em Ciência da Computação. Área de concentração: Teoria da Computação.

Aprovada em: 29/04/2021

BANCA EXAMINADORA

Prof. Dr. Manoel Bezerra Campêlo Neto (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Rafael Castro de Andrade
Universidade Federal do Ceará (UFC)

Prof. Dr. Phablo Fernando Soares Moura
Universidade Federal de Minas Gerais (UFMG)

Aos meus pais, que sempre me incentivaram e investiram na minha educação. À comunidade científica, para que este trabalho possa ser usado como referência para auxiliar outros pesquisadores na mesma área de pesquisa.

AGRADECIMENTOS

Início agradecendo aos meus pais, que sempre investiram na minha educação, esperando um futuro promissor para o seu filho. Sem eles, não teria conquistado os meus objetivos.

Aos meus familiares por todo o apoio e incentivo.

Tenho um agradecimento especial ao Professor Manoel Campêlo, por mais uma vez me orientar e me guiar. Tive o primeiro contato com o Prof. Manoel na disciplina de Seminários de Tutoria II, no segundo semestre da graduação em Matemática Industrial. No terceiro semestre fiz a primeira disciplina lendária com ele: programação linear. Depois disso ele me convidou para uma bolsa de iniciação científica, quase que eu perco a bolsa, o e-mail dele foi pro SPAM do hotmail. Por sorte consegui entrar em contato com ele e virei seu bolsista de IC. Essa bolsa de IC foi essencial para manter minha vida acadêmica, onde, a partir daí, entrei na área de pesquisa de programação matemática, que gerou meu trabalho de conclusão de curso. Depois, mais uma vez, virei seu orientando no mestrado em computação e cheguei até este trabalho pela sua orientação excepcional. Muito obrigado Prof. Manoel, se eu cheguei até aqui hoje, foi porque o senhor me guiou.

Ao Professor Rafael Andrade por aceitar participar desta banca e estar presente em minha vida acadêmica deste o primeiro semestre da graduação na disciplina de Elementos de Matemática até hoje. Durante a minha graduação e mestrado, sempre que possível, esclarecia dúvidas em sua sala, muitas delas eram de disciplinas que ele não estava ministrando. Além disso, o Prof. Rafael disponibilizou as duas máquinas utilizadas nos experimentos computacionais deste trabalho. Obrigado Prof. Rafael.

Ao Professor Phablo Fernando Soares Moura por aceitar participar da banca da minha defesa de mestrado.

Aos professores do DEMA e DC que participaram e me ajudaram durante minha vida acadêmica.

Aos meus amigos dentro e fora da universidade por sempre estarem presentes e me ajudando. Não vou citar nomes porque a lista seria grande e tenho medo de esquecer de alguém.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

Obrigado a todos.

“Nossa maior fraqueza está em desistir. O caminho mais certo de vencer é tentar mais uma vez.”

(Thomas Edison)

RESUMO

No Problema da Árvore t -Spanner de Custo Mínimo, cuja entrada é uma tripla (G, w, t) , onde $G = (V, E)$ é um grafo simples, $w : E \rightarrow \mathbb{R}^+$ é uma função de ponderação das arestas e $t \geq 1$ é um parâmetro, denominado de fator de dilatação, o objetivo consiste em determinar uma árvore geradora T em G de menor custo, tal que a distância entre qualquer par de vértices i e j na árvore T é no máximo t vezes a distância entre i e j em G . A partir da definição formal do problema, apresentamos um contexto de aplicação, complexidade e trabalhos relacionados na literatura. Concentramo-nos em métodos de programação matemática para sua resolução. Propomos um algoritmo enumerativo baseado em um procedimento de *branch-and-bound*. Apresentamos e estudamos quatro formulações de programação linear inteira para o problema, uma exponencial e três compactas. Duas das formulações compactas são as únicas presentes na literatura do problema. Propomos desigualdades válidas com potencial para fortalecer a relaxação linear dos modelos. Implementamos e avaliamos computacionalmente as formulações, as desigualdades válidas e o algoritmo enumerativo usando a linguagem C++ e o resolvidor CPLEX. Reportamos os resultados computacionais obtidos por cada método e identificamos o que apresenta o melhor desempenho para cada grupo de instâncias.

Palavras-chave: árvore t -spanner de custo mínimo; formulação matemática; *branch-and-bound*; programação linear inteira.

ABSTRACT

In the Minimum Weight t -Spanner Tree Problem, whose input is a triplet (G, w, t) , where $G = (V, E)$ is a simple graph, $w : E \rightarrow \mathbb{R}^+$ is an edge weighting function and $t \geq 1$ is a parameter, called the dilatation factor, the objective consists in determining a spanning tree T in G of minimum weight, such that the distance between any pair of vertices i and j in tree T is at most t times the distance between i and j in G . Starting from the formal definition of the problem, we present an application context, the problem complexity and related works in the literature. We focus on mathematical programming methods for its resolution. We propose an enumerative algorithm based on a *branch-and-bound* procedure. We present and study an exponential and three compact integer linear programming formulations for the problem. Two of the compact formulations are the only ones present in the problem literature. We propose valid inequalities with the potential to strengthen the linear relaxation of the models. We implement and computationally evaluate the formulations, the valid inequalities, and the enumerative algorithm using C++ Language and CPLEX solver. We report on the computational results obtained by each method, and we identify the one that presents the best performance for each group of instances.

Keywords: minimum weight t -spanner tree; mathematical formulation; branch-and-bound; integer linear programming.

LISTA DE FIGURAS

Figura 1 – Grafo $G, t = 3$	16
Figura 2 – Árvore Geradora de Custo Mínimo (AGM*) de G	16
Figura 3 – Árvore t -spanner de Custo Mínimo (AtSCM*) de G	16
Figura 4 – Exemplo de um grafo G	21
Figura 5 – Exemplo de um digrafo simétrico D	23
Figura 6 – Ciclo direcionado	24
Figura 7 – Ciclo não direcionado	24
Figura 8 – Arborescência enraizada em 4	25
Figura 9 – grafo $G, t = 2, 5$	32
Figura 10 – Árvore 2,5-spanner	32
Figura 11 – Blocos de um grafo	33
Figura 12 – Árvore Bloco-Articulação	33
Figura 13 – Grafo $G, t = 3$	36
Figura 14 – Caminho 3-inviável entre 1 e 2	37
Figura 15 – Ramificação no caminho $P = \langle v_1, v_2, \dots, v_k, v_{k+1} \rangle$, com $e_i = \{v_i, v_{i+1}\}$	38
Figura 16 – Caminhos t -inviáveis para ramificação	38
Figura 17 – Ramificação no caminho 1-5	39
Figura 18 – Ramificação no caminho 3-5	39
Figura 19 – Arborescências sobrepostas	52
Figura 20 – Grafo ponderado $G, t = 3$	63
Figura 21 – Variáveis \bar{x}_e	63
Figura 22 – Variáveis $\bar{\lambda}_{ij}^1$, raiz 1	64
Figura 23 – Variáveis $\bar{\tau}_{ij}^1$, raiz 1	64
Figura 24 – Variáveis \bar{x}_e	65
Figura 25 – Variáveis $\bar{\lambda}_{ij}^1$, raiz 1	65
Figura 26 – Variáveis $\bar{\pi}_i^1$, raiz 1	65
Figura 27 – Grafo ponderado $G, t = 4$	66
Figura 28 – Variáveis x_e	69
Figura 29 – Crescimento do tempo da formulação CR com relação a $ V(G) $	74
Figura 30 – Crescimento do tempo da formulação CR com relação a p	75
Figura 31 – Crescimento do tempo da formulação CR com relação a t	76

LISTA DE TABELAS

Tabela 1 – Algoritmos aproximativos e limites de inaproximabilidade de tSCM	29
Tabela 2 – Complexidade de Problema da Árvore t -spanner (AtS)	29
Tabela 3 – Resultados computacionais resumidos para formulações básicas e algoritmo enumerativo	73
Tabela 4 – Resultados computacionais resumidos para variantes de SR	76
Tabela 5 – Resultados computacionais resumidos para variantes de CR	78
Tabela 6 – Resultados para FE-MTZ - Instâncias com pesos mais espaçados	86
Tabela 7 – Resultados para CR-SR-AP-AlgE - Instâncias com pesos menos espaçados .	87
Tabela 8 – Resultados para CR-SR-AP-AlgE - Instâncias com pesos mais espaçados . .	88
Tabela 9 – Resultados para CR-SR-AP-AlgE - Instâncias com distância Euclidiana . .	89
Tabela 10 – Resultados para CR-SR-AP-AlgE - Instâncias com pesos unitários	90
Tabela 11 – Resultados para variantes de (SR) para instâncias com peso unitário	91
Tabela 12 – Resultados para CR com adição de desigualdades válidas para instâncias com peso unitário	92

LISTA DE ABREVIATURAS E SIGLAS

AGM	Problema da Árvore Geradora de Custo Mínimo
AGM*	Árvore Geradora de Custo Mínimo
ASMFD	Problema da Árvore Spanner de Menor Fator de Dilatação
ASMFD*	Árvore Spanner de Menor Fator de Dilatação
AtS	Problema da Árvore t -spanner
AtS*	Árvore t -spanner
AtSCM	Problema da Árvore t -spanner de Custo Mínimo
AtSCM*	Árvore t -spanner de Custo Mínimo
tS	t -spanner
tSCM	Problema do t -spanner de Custo Mínimo
tSCM*	t -spanner de Custo Mínimo

LISTA DE SÍMBOLOS

G	Grafo
$V(G)$	Conjunto de vértices do grafo G
$E(G)$	Conjunto de arestas do grafo G
$A(G)$	Conjunto de arcos do grafo G
$\text{dist}_G(i, j)$	Distância entre i e j no grafo G
\mathbb{R}^n	Espaço vetorial real n -dimensional
\mathbb{Q}^n	Conjunto dos vetores n -dimensionais com coordenadas racionais
\mathbb{Z}^n	Conjunto dos vetores n -dimensionais com coordenadas inteiras
\mathbb{B}^n	Conjunto dos vetores n -dimensionais com coordenadas binárias $\{0,1\}$
n	Cardinalidade de $V(G)$
m	Cardinalidade de $E(G)$

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Estruturação da Dissertação	17
2	CONCEITOS PRELIMINARES	19
2.1	Programação Matemática	19
2.2	Teoria dos Grafos	21
3	t-SPANNERS	27
3.1	Problemas sobre t-spanners	27
3.2	Encontrando uma árvore t-spanner	30
3.2.1	<i>Restrições t-spanner Suficientes</i>	30
3.2.2	<i>Componente 2-conexa</i>	32
4	ALGORITMO ENUMERATIVO	36
4.1	Ramificação	37
4.1.1	<i>Resolução de subproblemas</i>	40
4.2	CrITÉRIOS DE Poda	42
4.2.1	<i>Poda por Otimalidade</i>	42
4.2.2	<i>Poda por Bound</i>	43
4.2.3	<i>Poda por Inviabilidade</i>	43
4.3	Procedimento Enumerativo	44
5	FORMULAÇÕES MATEMÁTICAS	45
5.1	Formulação Exponencial	45
5.1.1	<i>Formulação SEC's</i>	46
5.1.2	<i>Formulação MTZ</i>	48
5.2	Formulações Compactas	51
5.2.1	<i>Formulação SR</i>	54
5.2.2	<i>Formulação CR</i>	57
5.2.3	<i>Formulação Aresta-Potencial (AP)</i>	60
5.3	Desigualdades Válidas	62
5.3.1	<i>Desigualdade Potencial Mínimo</i>	62
5.3.1.1	<i>Desigualdade para formulação AP</i>	63
5.3.1.2	<i>Desigualdade para formulação CR</i>	64

5.3.1.3	<i>Desigualdade para formulação SR</i>	66
5.3.2	<i>Desigualdade Ciclo t-Inviável</i>	66
5.3.3	<i>Desigualdade t-Spanner Arborescente</i>	69
6	EXPERIMENTOS COMPUTACIONAIS	71
6.1	Instâncias Estudadas	71
6.2	Ambiente Computacional	72
6.3	Desempenho das Formulações Básicas	72
6.4	Desempenho das Formulações com Desigualdades Válidas	75
6.4.1	<i>Variantes da Formulação SR</i>	75
6.4.2	<i>Formulação CR com Desigualdades Válidas</i>	77
7	CONCLUSÃO	79
	REFERÊNCIAS	81
	APÊNDICE A – RESULTADOS DETALHADOS PARA FORMULAÇÕES BÁSICAS	85
	APÊNDICE B – RESULTADOS DETALHADOS PARA VARIANTES DE SR	91
	APÊNDICE C – RESULTADOS DETALHADOS PARA VARIANTES DE CR	92

1 INTRODUÇÃO

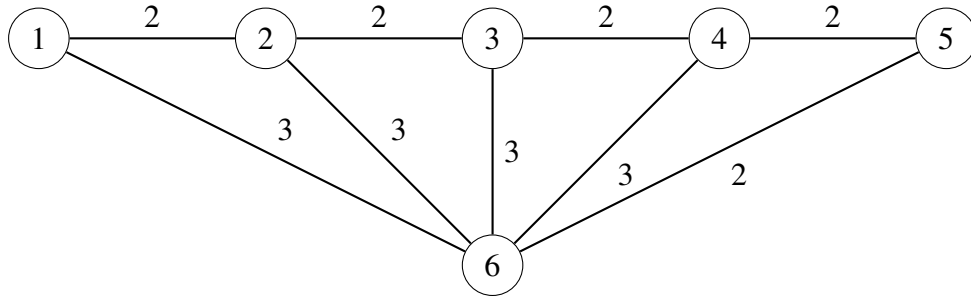
Seja $G = (V, E)$ um grafo simples, onde V é o conjunto de vértices e E , o conjunto de arestas. Um subgrafo $H = (V', E')$ de G é um outro grafo cujos vértices e arestas são subconjuntos dos vértices e arestas de G . O subgrafo H é dito gerador se $V' = V$, e, caso seja conexo e acíclico, H é denominado uma árvore geradora de G . Quando temos uma função de ponderação $w : E \rightarrow \mathbb{R}$, que associa a cada aresta $e \in E(G)$ um valor real, dizemos que G é um grafo ponderado. Nesse caso, o peso (ou custo) do grafo é a soma dos pesos de suas arestas.

Problemas sobre subgrafos geradores com pré-determinadas características aparecem em vários cenários, dentre eles: computação distribuída (AWERBUCH, 1985; PELEG; ULLMAN, 1989), redes de comunicação (PELEG; UPFAL, 1988; PELEG; RESHEF, 1999; OLIVEIRA; PARDALOS, 2005), localização e distribuição (GOLDSCHMIDT *et al.*, 1996), robótica (MARBLE; BEKRIS, 2011b; ARIKATI *et al.*, 1996; MARBLE; BEKRIS, 2011a) e bioinformática (NGUYEN *et al.*, 2008). Neste trabalho, focamos em um dos problemas sobre subgrafos geradores: o problema da árvore t -spanner de custo mínimo.

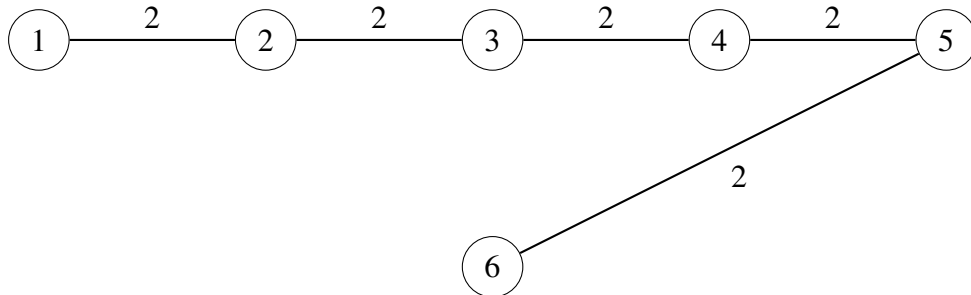
O Problema da Árvore t -spanner de Custo Mínimo (AtSCM) é a composição de dois problemas de otimização: o Problema da Árvore Geradora de Custo Mínimo (AGM) e o Problema do t -spanner de Custo Mínimo (tSCM). Temos como entrada uma tripla (G, w, t) , na qual $G = (V, E)$ é um grafo simples, $w : E \rightarrow \mathbb{R}^+$ é uma função de ponderação das arestas e $t \geq 1$ é um número real, denominado fator de dilatação. O problema consiste em determinar uma árvore geradora T em G de menor peso, tal que a distância entre qualquer par de vértices i e j na árvore T é no máximo t vezes a distância entre i e j em G , sendo a distância entre i e j dada pelo peso de um caminho de peso mínimo entre esses vértices no grafo considerado. As Figuras 1, 2 e 3 ilustram, respectivamente, um grafo G de entrada, a Árvore Geradora de Custo Mínimo (AGM*) de G e a Árvore t -spanner de Custo Mínimo (AtSCM*) para $t = 3$. O número próximo a cada aresta representa seu peso.

Quando $w_e = 1$, para toda aresta $e \in E(G)$, qualquer árvore geradora de G terá peso $|V(G)| - 1$, logo o problema de otimização se torna um problema de decisão, no qual desejamos determinar se G admite ou não uma árvore t -spanner. Nesse caso, denotaremos o problema como AtS.

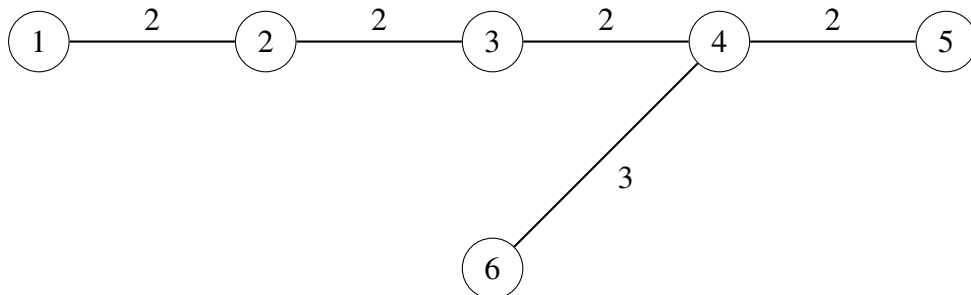
O problema da árvore t -spanner de custo mínimo já foi bastante explorado para o caso com pesos unitários (AtS), mas para o caso com pesos arbitrários ainda há poucos estudos, principalmente na área de programação matemática, visto que na literatura do problema só

Figura 1 – Grafo G , $t = 3$ 

Fonte: elaborado pelo autor.

Figura 2 – Árvore Geradora de Custo Mínimo (AGM*) de G 

Fonte: elaborado pelo autor.

Figura 3 – Árvore t -spanner de Custo Mínimo (AtSCM*) de G 

Fonte: elaborado pelo autor.

se encontra um algoritmo exato e duas formulações matemáticas, ambos propostos por Braga (2019).

Sabe-se que os dois problemas, AtS e AtSCM, são computacionalmente difíceis em geral. Grosseiramente falando, isso significa que não há garantia da existência de algoritmo para solução do problema cujo tempo de execução cresça polinomialmente com o tamanho do grafo de entrada. Está provado que AtS pode ser resolvido em tempo linear para $t = 2$, que é um problema NP -completo para $t \geq 4$ (CAI; CONEIL, 1995), mas para o caso $t = 3$ a complexidade ainda está em aberto. Como instâncias do AtS são um subconjunto das instâncias do AtSCM, então os casos NP -completos se traduzem em casos NP -difíceis para o AtSCM. Na verdade, para instâncias com pesos arbitrários, o AtSCM já é NP -difícil para $t > 1$ (CAI; CONEIL, 1995).

Neste trabalho, tratamos o AtSCM através de programação matemática, com o intuito

de resolver instâncias maiores que as já tratáveis, até o momento, pelos métodos existentes na literatura. Para um melhor entendimento do material aqui apresentado, revisamos inicialmente alguns conceitos de grafos que serão utilizados no decorrer do texto, além de conceitos relacionados à programação matemática. Apresentamos uma revisão bibliográfica do problema e estudamos, em particular, as formulações existentes na literatura (BRAGA, 2019). Propomos duas novas formulações e um novo algoritmo exato, que implementamos e avaliaremos computacionalmente. Resultados preliminares com relaxação linear das formulações inspiraram a determinação de desigualdades válidas, que serão apresentadas e implementadas, para avaliar a mudança de desempenho das formulações com suas adições aos modelos. Na seção a seguir apresentaremos, de modo mais detalhado, o que cada capítulo do texto irá tratar.

1.1 Estruturação da Dissertação

Esta dissertação está dividida em três partes principais: definição e propriedades do problema, métodos propostos e considerações finais.

No Capítulo 2, recapitulamos noções básicas de teoria dos grafos e programação matemática. Apresentamos os conceitos em grafos que serão utilizados neste trabalho, bem como sobre modelos de programação linear inteira, constituídos de variáveis, restrições e função objetivo.

No Capítulo 3, definimos formalmente os problemas de t -spanner em grafos e, especificamente, definimos o problema da árvore t -spanner de custo mínimo, alvo deste trabalho. Adicionalmente, apresentamos uma revisão bibliográfica, mostrando histórico, complexidade e trabalhos relacionados na literatura.

No Capítulo 4, propomos um algoritmo exato para o AtSCM, o primeiro para o problema que não utiliza resolução de formulações matemáticas como um de seus componentes. Lembramos que o algoritmo exato conhecido para o AtSCM foi proposto por Braga (2019), sendo um algoritmo do tipo *Branch-and-Price*. Nosso algoritmo aplica um procedimento de *Branch-and-Bound*, onde cada subproblema é solucionado por uma adaptação do clássico algoritmo de Kruskal (KRUSKAL, 1956), que determina, dado um grafo G ponderado, uma árvore geradora mínima de G .

No Capítulo 5, apresentamos as formulações lineares-inteiras estudadas neste trabalho, duas delas são as primeiras formulações para o AtSCM, que foram propostas por Braga (2019). Adicionalmente, propomos duas novas formulações para o problema, uma compacta e

outra com uma quantidade polinomial de variáveis, mas exponencial de restrições. Estudamos a relaxação linear das formulações (remoção das restrições de integralidade das variáveis) e, inspirados nos pontos fracionários, propomos desigualdades válidas, com o intuito de fortalecer os modelos.

No Capítulo 6, apresentamos os resultados computacionais obtidos pelos métodos de resolução, já existentes e propostos. Comparamos o desempenho da implementação, em C++, utilizando o solver CPLEX (IBM, 2021), das formulações matemáticas e do algoritmo enumerativo. Observamos o impacto no desempenho das formulações após a adição de desigualdades válidas. Analisamos os resultados, identificamos qual o melhor método para cada classe de instâncias e concluimos quais desigualdades válidas acarretaram melhoria na performance das formulações.

No Capítulo 7, apresentamos as conclusões do trabalho, reforçando nossas contribuições para o AtSCM, e indicamos possíveis direções para trabalhos futuros.

2 CONCEITOS PRELIMINARES

Neste capítulo, apresentaremos os principais conceitos da área de teoria dos grafos utilizados no decorrer do texto. Além disso, apresentaremos a definição de modelos de programação matemática, que são o alvo principal deste trabalho.

2.1 Programação Matemática

Genericamente, problemas de otimização consistem em determinar, dentre um conjunto de soluções possíveis, que atendem às condições do problema, uma que seja a melhor, segundo algum critério que ranqueia as soluções viáveis.

Quando nos deparamos com um problema desse tipo, intuitivamente procuramos formas de solucioná-lo. Às vezes, o fazemos de forma desorganizada e não conseguimos resultados. Uma abordagem para esses casos pode ser encontrada com programação matemática, onde modelamos e organizamos os dados do problema, por meio de variáveis que representam as características dele, de modo que ele fique mais simples de ser estudado. Modelos de programação matemática são compostos por três partes: variáveis, restrições e função objetivo.

As variáveis de um modelo matemático são elementos de decisão, podendo determinar uma condição ou quantificação de algum fator, este último podendo ser ou não fracionado. Restrições determinam limitações ou imposições para alguma característica do problema, a serem descritas por expressões matemáticas em termos das variáveis. Por fim, temos a função objetivo do modelo, que define o critério que queremos otimizar, dadas as variáveis e as restrições estabelecidas. O objetivo pode ser de dois tipos: minimização ou maximização.

A solução de um modelo matemático é a melhor atribuição de valores às suas variáveis, considerando a função objetivo, dentro dos limites definidos pelas restrições, denominada como solução ótima. A depender das características do modelo, são usados, para sua resolução, diferentes métodos de otimização. Em nosso estudo, o problema é tratado por meio de modelagens de programação linear-inteira (PLI), onde cada uma das funções que definem as restrições e a função objetivo são lineares, e algumas (ou todas) as variáveis devem receber valores pertencentes ao conjunto dos números inteiros, enquanto há possivelmente outras a quem se pode atribuir valores fracionários.

Uma formulação genérica para modelos PLI é:

$$\begin{aligned} \min \quad & c^t x + d^t y \\ \text{s.a:} \quad & Ax + Dy \leq b \\ & x \in \mathbb{Z}_+^{n_1} \\ & y \geq 0 \end{aligned}$$

Nessa descrição $c \in \mathbb{Q}^{n_1}$ e $d \in \mathbb{Q}^{n_2}$ são vetores de coeficientes, que definem a função objetivo $c^t x + d^t y$, a ser minimizada, como indica o termo *min*. $A_{m \times n_1}$ e $D_{m \times n_2}$ são matrizes de coeficientes racionais das m restrições do modelo, associadas às n_1 variáveis x e às n_2 variáveis y , respectivamente. Para transformarmos esse problema em maximização basta pré-multiplicarmos a função objetivo por -1 . $Ax + Dy \leq b$ representa o conjunto de restrições do modelo, e os vetores x e y compreendem as variáveis, respectivamente, inteiras e fracionárias. Por simplicidade, estamos supondo variáveis não negativas. O modelo obtido com a remoção das restrições de integralidade das variáveis x é chamado de relaxação linear.

Modelos matemáticos, apesar de representarem problemas de otimização de modo preciso, podem demandar grande esforço computacional para serem solucionados. Nesse contexto, podemos aperfeiçoar as formulações, geralmente modificando o conjunto de restrições, ou propor métodos alternativos para resolução do problema, por exemplo, com métodos heurísticos, que, apesar de não garantirem que a solução encontrada é ótima, tendem a ser mais rápidos que as formulações.

Uma das estratégias para fortalecer um modelo é a inclusão de desigualdades válidas. Sejam (P) o problema de PLI apresentado acima e (X, Y) o conjunto de todos os pontos viáveis de (P). Desse modo $(X, Y) = \{(x, y) \in \mathbb{Z}^{n_1} \times \mathbb{R}^{n_2} \mid Ax + Dy \leq b, x \geq 0, y \geq 0\}$. Além disso, seja (π_1, π_2, π^0) , com $\pi_1 \in \mathbb{Q}^{n_1}$, $\pi_2 \in \mathbb{Q}^{n_2}$ e $\pi^0 \in \mathbb{Q}$. Dizemos que (π_1, π_2, π^0) define uma desigualdade válida de (P) se:

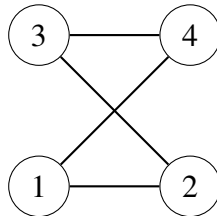
$$\pi_1^t x + \pi_2^t y \leq \pi^0 \quad \forall (x, y) \in (X, Y) \quad (2.1)$$

Sejam (X', Y') o conjunto de todos os viáveis da relaxação linear de (P) e $(x, y) \in (X', Y')$ tal que existe $x_i \notin \mathbb{Z}$, para algum $i \in \{1, \dots, n_1\}$. Dizemos que (π_1, π_2, π^0) corta o ponto (x, y) se $\pi_1^t x + \pi_2^t y > \pi^0$ (viola (2.1)). A inclusão de desigualdades válidas para (P) que cortem soluções da relaxação linear inviáveis para (P) podem contribuir para a resolução do modelo.

2.2 Teoria dos Grafos

Um grafo G simples é um par ordenado (V, E) , onde V é o conjunto de vértices e E , o conjunto de arestas, que representam uma relação entre pares de vértices. Dessa maneira, uma aresta $e \in E$ é dada por $\{i, j\} \subseteq V$, com $i \neq j$. A Figura 4 ilustra a representação gráfica de um grafo, onde $V = \{1, 2, 3, 4\}$ e $E = \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{1, 4\}\}$. A menos que especificado de outra forma, o conjunto de vértices e arestas de um grafo G será denotado por $V(G)$ e $E(G)$, respectivamente. Dizemos que G é ponderado se existe uma função de ponderação $w : E(G) \rightarrow \mathbb{R}$, que associa a cada aresta $e \in E(G)$ um valor real.

Figura 4 – Exemplo de um grafo G



Fonte: elaborado pelo autor.

Dizemos que grafo H é subgrafo de um outro grafo G , a ser denotado $H \subseteq G$, se $V(H) \subseteq V(G)$ e $E(H) \subseteq E(G)$; em particular, se $V(H) = V(G)$ então o denominamos de subgrafo gerador. H é dito subgrafo induzido quando, para quaisquer dois vértices i e j de H , se $\{i, j\} \in E(G)$ então $\{i, j\} \in E(H)$. Definimos um grafo $G = G_1 \cup G_2$ como a união de outros dois grafos G_1 e G_2 , quando $V(G) = V(G_1) \cup V(G_2)$ e $E(G) = E(G_1) \cup E(G_2)$. Defina $N_G(v)$ como o conjunto de vértices adjacentes ao vértice $v \in V(G)$, ou seja, $N_G(v) = \{i \mid \{v, i\} \in E(G)\}$, denominado de vizinhança de v . Quando o grafo estiver claro pelo contexto, o subíndice G pode ser descartado.

Um dos problemas fundamentais de otimização combinatória é a determinação de um caminho de menor comprimento em um grafo G , entre dois vértices dados. Um i, j -caminho em um grafo não-direcionado é uma sequência não vazia de vértices distintos $P = \langle v_0, v_1, \dots, v_{k-1}, v_k \rangle$ de G , com $v_0 = i$ e $v_k = j$, onde $\{v_l, v_{l+1}\} \in E(G)$, $l \in \{0, \dots, k-1\}$, que são chamadas arestas do caminho. O comprimento desse caminho é k , ou seja, a quantidade de arestas. Por exemplo, na Figura 4, $\langle 1, 2, 3, 4 \rangle$ é um $1, 4$ -caminho de comprimento 3. Sem prejuízo de entendimento, vamos também chamar de caminho o grafo definido pela sequência de vértices e arestas do caminho.

Quando o i, j -caminho possui o menor comprimento possível, dentre todos os i, j -

caminhos de G , ele é dito um i, j -caminho mínimo. Se G for um grafo ponderado, o peso do caminho é a soma dos pesos de suas arestas. Nesse caso, definimos o tamanho/custo de um i, j -caminho como seu peso, e assim definimos o caminho de peso mínimo. O caso de pesos unitários é equivalente àquele sem pesos. Vamos definir $\text{dist}_G(i, j)$ como a distância entre i e j , que representa o custo de um i, j -caminho de menor peso em G , em outras palavras, o custo de um caminho mínimo entre i e j .

Existem na literatura alguns algoritmos que, dados um grafo ponderado G e dois vértices i e j , calculam $\text{dist}_G(i, j)$ em uma quantidade polinomial de passos, em função da quantidade de vértices e arestas de G , caso a função de ponderação w atenda a determinadas condições, como por exemplo, $w : E(G) \rightarrow \mathbb{R}^+$ (Dijkstra (1959)) ou a hipótese de que toda aresta $e = \{u, v\} \in E(G)$ é tal que $w_e + \text{dist}_{G \setminus e}(u, v) \geq 0$, onde $G \setminus e$ é o grafo resultante da remoção da aresta e (Bellman (1958), Ford (1956), Floyd (1962), Warshall (1962)).

Um ciclo C em um grafo não direcionado G é uma sequência de vértices com pelo menos quatro vértices de G , todos distintos, a menos do último que será o primeiro, ou seja, $C = \langle v_0, v_1, \dots, v_{k-1}, v_k \rangle$, com $v_0 = v_k$, onde $\{v_l, v_{l+1}\} \in E(G)$, $l \in \{0, \dots, k-1\}$, que são chamadas arestas do ciclo. Podemos perceber C como um v_0, v_{k-1} -caminho com a adição da aresta $\{v_{k-1}, v_0\}$. O tamanho desse ciclo é k . Frequentemente, chamamos também de ciclo o grafo definido pela sequência de vértices e arestas do ciclo. Se $\{v_0, v_1, \dots, v_{k-1}\} = V(G)$, dizemos que C é um ciclo hamiltoniano. Caso exista um ciclo hamiltoniano no grafo, G é dito hamiltoniano, senão G é dito não hamiltoniano. Dizemos que G é acíclico se não possuir qualquer ciclo.

Um grafo conexo é tal que existe um caminho entre cada par de vértices. Um grafo $T = (V, E)$ é dito uma árvore se T é um grafo conexo e acíclico. Se T é um subgrafo gerador de um outro grafo G , dizemos que T é uma árvore geradora de G . Podemos caracterizar um determinado grafo G como uma árvore de várias maneiras, além de conectividade e aciclicidade. No Teorema 2.2.1 apresentamos as principais maneiras de identificar que um determinado grafo é uma árvore.

Teorema 2.2.1 (WEST, 2000) *As seguintes afirmações são equivalentes e caracterizam uma árvore G com n vértices ($n \geq 1$):*

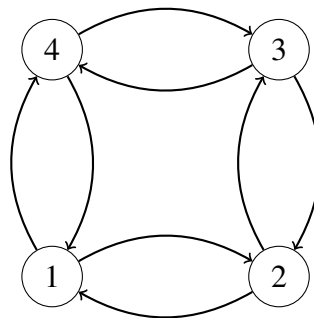
1. G é conexo e acíclico.
2. G é conexo e possui $n - 1$ arestas.
3. G possui $n - 1$ arestas e não contém ciclos.

4. Para $u, v \in V(G)$, G possui exatamente um u, v -caminho.

De modo semelhante ao problema do caminho mínimo, um outro problema de otimização surge: dado um grafo ponderado G , determinar uma árvore geradora de G que tem o menor custo, denotada por AGM^* . Na literatura, existem algoritmos polinomiais que determinam uma AGM^* em G (ver por exemplo Kruskal (1956) e Prim (1957)). Em particular, o algoritmo proposto por Kruskal (1956) será fundamental para o desenvolvimento do nosso algoritmo enumerativo, portanto, apresentaremos sua descrição no Capítulo 4.

Um grafo direcionado (digrafo) D é um par ordenado (V, A) , onde V é o conjunto de vértices e A , o conjunto de arcos, sendo cada arco um par ordenado de vértices. Diferentemente da definição de grafo (não direcionado), na qual uma aresta representa um par não ordenado de vértices, vamos denotar cada elemento de A como (i, j) , com $i, j \in V$. Dizemos que o arco (i, j) sai de i e chega (ou entra) em j . Um digrafo D é dito simétrico quando $(i, j) \in A$ implica $(j, i) \in A$. Podemos transformar um grafo não direcionado $G = (V, E)$ em um digrafo simétrico D simplesmente criando duas orientações para cada aresta $\{i, j\} \in E(G)$, ou seja, $D = (V(G), A)$, onde $A = \{(i, j), (j, i) \mid \{i, j\} \in E(G)\}$. A Figura 5 ilustra a transformação do grafo da Figura 4 em um grafo direcionado simétrico.

Figura 5 – Exemplo de um digrafo simétrico D

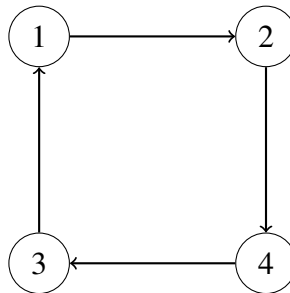


Fonte: elaborado pelo autor.

Da mesma maneira que em grafos não direcionados, abstraímos conceitos de vizinhança, caminho e ciclo para grafos direcionados. Dividimos vizinhança de um determinado vértice $v \in V(D)$ de duas maneiras: conjunto de vértices i , cujo arco (i, v) chega em v , denominado de vizinhança de entrada de v , e conjunto de vértices j , cujo arco (v, j) sai de v denominado vizinhança de saída v . Formalmente, denotamos a vizinhança de saída de v como $N_D^+(v) = \{j \mid (v, j) \in A(D)\}$ e vizinhança de entrada como $N_D^-(v) = \{i \mid (i, v) \in A(D)\}$. O subíndice D pode ser descartado quando o digrafo estiver claro pelo contexto.

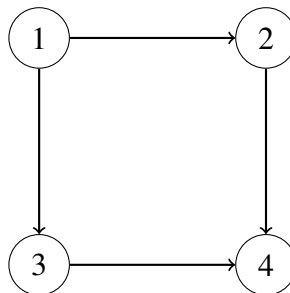
Um i, j -caminho P em um grafo direcionado D é uma sequência de vértices distintos $P = \langle v_0, v_1, \dots, v_{k-1}, v_k \rangle$, com $v_0 = i$ e $v_k = j$, onde $v_l \in V(D)$, $l \in \{0, \dots, k\}$ e, para todo par $\{v_l, v_{l+1}\}$, um dos arcos $(v_l, v_{l+1}), (v_{l+1}, v_l) \in A(D)$. Se para todo par $\{v_l, v_{l+1}\}$ o arco $(v_l, v_{l+1}) \in A(D)$ então dizemos que P é um caminho direcionado de i a j ; caso contrário, dizemos que P é não direcionado. Para ciclos temos duas possibilidades em digrafos: ciclo direcionado e ciclo não direcionado. Dada uma sequência $C = \langle v_0, v_1, \dots, v_{k-1}, v_k \rangle$ com pelo menos 3 vértices, distintos a menos de $v_0 = v_k$, C é dito ciclo direcionado se o arco $(v_l, v_{l+1}) \in A(G)$ para todo par $\{v_l, v_{l+1}\}$, ou o arco $(v_{l+1}, v_l) \in A(G)$ para todo par $\{v_l, v_{l+1}\}$; caso contrário C é dito não direcionado. As Figuras 6 e 7 ilustram, respectivamente, um ciclo direcionado e um ciclo não direcionado. Uma observação quanto à existência de ciclos em grafos direcionados é a possibilidade de eles envolverem apenas dois vértices i e j , caso $(i, j), (j, i) \in A(D)$, o que não ocorre em grafos não direcionados.

Figura 6 – Ciclo direcionado



Fonte: elaborado pelo autor.

Figura 7 – Ciclo não direcionado



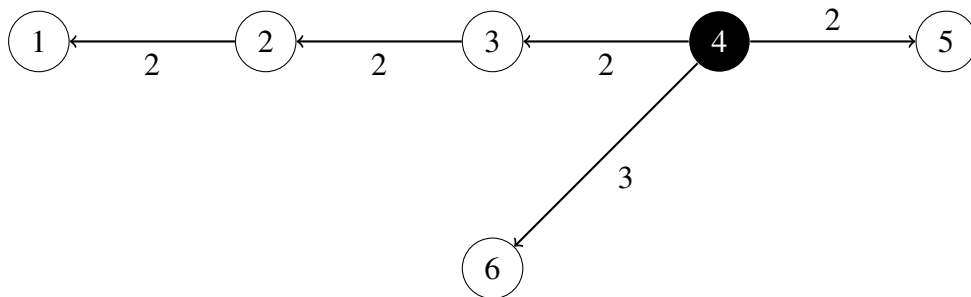
Fonte: elaborado pelo autor.

Um grafo orientado é um caso particular de um digrafo $D = (V, A)$, no qual $(i, j) \notin A$ ou $(j, i) \notin A$ para quaisquer dois vértices $i, j \in V$. Usualmente, esse termo refere-se ao digrafo obtido pela orientação de um grafo não direcionado $G = (V, E)$, processo que consiste em atribuir a cada aresta $\{i, j\} \in E(G)$ exatamente uma das orientações (i, j) ou (j, i) . A transformação reversa também é possível. A partir de um digrafo $D = (V, A)$, definimos o grafo subjacente

$G = (V, E)$, onde $E = \{\{i, j\} : (i, j) \text{ ou } (j, i) \in A\}$. Observe que há diferentes orientações de um grafo não orientado G , entretanto a transformação de cada um destes em um grafo não orientado corresponde ao próprio G .

Uma arborescência T' é uma orientação de uma árvore $T = (V, E)$, na qual um vértice $r \in V(T)$, chamado raiz, é tal que todas as arestas incidentes a r são orientadas saindo de r e, para todo vértice $j \in V(T) \setminus \{r\}$, exatamente uma aresta incidente a j é orientada chegando em j (as demais são orientadas saindo de j). Em outras palavras, uma arborescência T' (enraizada em r) é um grafo orientado onde $|N_{T'}^-(r)| = 0$ e $|N^-(j)| = 1$, para todo $j \in V(T') \setminus \{r\}$. Ilustramos na Figura 8 a arborescência relativa à árvore da Figura 3 enraizada no vértice 4.

Figura 8 – Arborescência enraizada em 4



Fonte: elaborado pelo autor.

Em nosso problema, consideramos grafos simples, ponderados e conexos, com pelo menos 3 vértices. A função w de ponderação será tal que $w : E(G) \rightarrow \mathbb{R}^+$, ou seja, todas as arestas do grafo terão peso não negativo. Definiremos alguns parâmetros que serão utilizados ao decorrer do texto:

- $w_e = w_{ij}$ = peso ou custo da aresta $e = \{i, j\} \in E(G)$.
- $w(X)$ = somatório dos pesos das arestas em X , de maneira simplificada, peso de X .

Podemos usar X para representar um grafo, caminho ou ciclo.

Observamos previamente que a determinação ou verificação de viabilidade de uma solução para problemas de spanner em um grafo G passam pelo conhecimento de $\text{dist}_G(i, j)$, para todo $i, j \in V(G)$ (ou pelo menos $\{i, j\} \in E(G)$). Como $w_e \geq 0$ em nosso caso, então existem algoritmos polinomiais que determinam um i, j -caminho mínimo, para quaisquer $i, j \in V(G)$. Em particular, utilizaremos em nosso trabalho o algoritmo de Floyd-Warshall ((FLOYD, 1962), (WARSHALL, 1962)), que determina o custo de todos os caminhos mínimos entre todos os pares de vértices, caso o grafo de entrada seja conexo e não possua ciclos negativos (um ciclo C é dito negativo se $w(C) < 0$).

Como entrada o algoritmo recebe uma matriz $D_{n \times n}$, $n = |V(G)|$, na qual $D_{ij} = w_{ij}$, se a aresta $\{i, j\}$ existe, ou $D_{ij} = +\infty$, caso contrário. A saída do algoritmo é uma nova matriz $D_{n \times n}$, onde cada posição D_{ij} corresponde a $\text{dist}_G(i, j)$, se $i \neq j$, e D_{ii} é o peso do menor ciclo contendo i ($D_{ii} = +\infty$ caso não exista tal ciclo). Um pseudocódigo do procedimento de Floyd-Warshall está apresentado no Algoritmo 1. Por simplicidade, vamos supor $V(G) = \{1, 2, \dots, n\}$.

Algoritmo 1: Floyd-Warshall($D_{n \times n}$)

```

1: for  $i \leftarrow 1$  to  $n$  do
2:   for  $j \leftarrow 1$  to  $n$  do
3:     for  $k \leftarrow 1$  to  $n$  do
4:       if  $D_{ij} > D_{ik} + D_{kj}$  then
5:          $D_{ij} \leftarrow D_{ik} + D_{kj}$ 
6:       end if
7:     end for
8:   end for
9: end for
10: return  $D_{n \times n}$ 

```

Para mais detalhes de conceitos de programação matemática, teoria dos grafos e algoritmos de caminho mínimo e árvore geradora mínima veja Bazaraa *et al.* (2009), Wolsey (1998), West (2000) e Cormen *et al.* (2009).

3 t -SPANNERS

Neste capítulo, definiremos formalmente t -spanner em grafos e, em específico, árvore t -spanner, assim como o problema associado – o problema da árvore t -spanner de custo mínimo. Apresentaremos uma revisão bibliográfica sobre o problema, expondo sua complexidade para o caso geral e em algumas classes específicas de grafos.

3.1 Problemas sobre t -spanners

Dados um grafo conexo G , ponderado em arestas, e um real $t \geq 1$, um t -spanner (tS) de G é um subgrafo gerador H tal que, para todo par $\{i, j\}$, com $i, j \in V(G)$, a distância entre i e j em H é no máximo t vezes a distância no grafo original. O parâmetro t é denominado de fator de dilatação. De maneira formal, um subgrafo gerador H é um t -spanner de G se:

$$\text{dist}_H(i, j) \leq t * \text{dist}_G(i, j), \quad \forall i, j \in V(G) \quad (3.1)$$

onde $\text{dist}(i, j)$ é o menor peso de um i, j -caminho no grafo considerado. Se além de subgrafo gerador, for acíclico, portanto uma árvore geradora, dizemos que H é uma árvore t -spanner. t -Spanners descrevem contextos onde se deseja “aproximar” o grafo original por subgrafos esparsos, mantendo controle sobre o aumento da distância vértice-a-vértice (PELEG; SCHAFFER, 1989). Dessa forma, eles encontram aplicações em áreas diversas, como redes de comunicação, sistemas distribuídos, robótica, projeto de redes, geometria computacional e bioinformática.

Problemas sobre t -spanners aparecem, em particular, quando se objetiva simultaneamente rapidez de computação/comunicação e economia de recursos em uma rede, mantendo um certo compromisso entre esses critérios, tendo em vista as disponibilidades da mesma. Esse é caso de situações em redes de comunicação nas quais se deseja determinar um esquema de roteamento que use caminhos par-a-par os mais curtos possíveis mas que demande memória local não excessiva (segundo um certo parâmetro) para armazenar as informações (PELEG; UPFAL, 1988). Cenários similares a esse são descritos por Peleg e Reshef (1999) e Oliveira e Pardalos (2005).

Nessas aplicações, o propósito é encontrar um roteamento “barato”, considerando todas as conexões usadas, porém mantendo controle sobre o custo de comunicação entre quaisquer dois pontos da rede, de modo a não ser muito maior que o custo na rede original. Podemos então admitir que uma certa constante $t \geq 1$ seja considerada para o controle de quanto maior poderia ser esse custo. Dessa forma, considerando que a rede original é representada por um

grafo G e que a constante t é o fator de dilatação, a solução corresponde a encontrar um t -spanner de G . Caso seja necessário/desejável que haja uma única possibilidade de comunicação entre quaisquer dois pontos da rede, ou seja, que o caminho selecionado entre qualquer par de pontos seja sempre o mesmo, desejamos encontrar uma árvore t -spanner. Ver Cai (1994), Cai e Coneil (1995) e referências lá citadas para maiores detalhes das aplicações.

Nesse contexto, dada como entrada uma tripla (G, w, t) , onde $G = (V, E)$ é um grafo simples, $w : E \rightarrow \mathbb{R}^+$ é uma função de ponderação das arestas e $t \geq 1$ é um número real (fator de dilatação), alguns problemas de otimização aparecem, a saber: Problema do t -spanner de Custo Mínimo (tSCM) e Problema da Árvore t -spanner de Custo Mínimo (AtSCM), este último alvo principal deste trabalho. No tSCM, queremos determinar, caso exista, um tS em G de peso mínimo, ou seja, tal que a soma dos pesos de suas arestas seja mínima, a ser denominado de t -spanner de Custo Mínimo (tSCM*). No AtSCM, queremos determinar, se existir, um tS de G que seja acíclico (em outras palavras, uma árvore t -spanner) e que tenha o menor peso possível, a qual denominamos de Árvore t -spanner de Custo Mínimo (AtSCM*). Quando o grafo de entrada é não-ponderado (de modo equivalente, $w_e = 1, \forall e \in E(G)$), o AtSCM se torna um problema de decisão, visto que toda árvore geradora, em particular uma árvore t -spanner, terá peso $|V| - 1$. Nesse caso, ele será referenciado simplesmente como Problema da Árvore t -spanner (AtS), e sua solução, como Árvore t -spanner (AtS*). Finalmente, note que outro problema de otimização, denominado Problema da Árvore Spanner de Menor Fator de Dilatação (ASMFD), pode ser definido quando apenas o par (G, w) é dado como entrada e desejamos determinar o menor real t tal que G admite uma árvore t -spanner.

Sabe-se que o tSCM e o ASMFD são *NP*-difíceis. Na verdade, determinar o tSCM* é *NP*-difícil já para $t \geq 2$ (CAI, 1992; PELEG; SCHAFFER, 1989). Algoritmos aproximativos e limites de inaproximabilidade também são conhecidos para o tSCM. Ver Tabela 1. Em relação ao ASMFD, Fekete e Kremer (2001) mostraram que o problema de determinar a Árvore Spanner de Menor Fator de Dilatação (ASMFD*) é *NP*-difícil para grafos planares. Além disso, é inaproximável com fator $2 - \epsilon$, $\epsilon > 0$, em grafos arbitrários (GALBIATI, 2001). Entretanto, o caso não ponderado do ASMFD é aproximável com fator $O(\log |V|)$ (EMEK; PELEG, 2008).

Os resultados de complexidade sobre o caso de árvores são mais abundantes. Primeiro, vale ressaltar que no caso de AtS pode-se restringir t aos inteiros (prova desse fato será apresentada na Subseção 3.2.1). Em geral, verificar se existe uma AtS* é *NP*-completo para $t \geq 4$. Quando $t = 1$ o problema se torna trivial, visto que G admite uma árvore t -spanner, se, e

Tabela 1 – Algoritmos aproximativos e limites de inaproximabilidade de tSCM

Trabalhos	Resultados
Pesos Unitários	
Kortsarz (2001)	INAPROX- $c \log n$ ($t = 2, c < 1$)
Kortsarz e Peleg (1994)	APROX- $O(\log m/n)$ ($t = 2$)
Dinitz <i>et al.</i> (2015)	INAPROX- $2^{\frac{\log^{1-\varepsilon} n}{t}}$ ($t \geq 3, \varepsilon > 0$)
Althofer <i>et al.</i> (1993)	APROX- $O(n^{\frac{2}{t-1}})$ ($t \geq 3$)
Pesos Arbitrários	
Chandra <i>et al.</i> (1992)	APROX- $O(n^{\frac{2+\varepsilon}{t-1}})$ ($t > 1, \varepsilon > 0$)
Chechik e Wulff-Nilsen (2016)	APROX- $O(n^{1/k}(1/\varepsilon)^{3+2/k})$ (t inteiro, $\varepsilon > 0$) ($k = (2t - 1)(1 + \varepsilon)$)
APROX (Algoritmo de Aproximação); INAPROX (Fator de Inaproximabilidade)	

Fonte: Adaptado de Braga (2019), pág. 19.

somente se, G for uma árvore (CAI; CONEIL, 1995). Para o caso $t = 2$, o problema de verificar se G possui uma árvore t -spanner também é polinomial (CAI, 1992; CAI; CONEIL, 1995). Para o caso $t = 3$, a complexidade ainda está em aberto para grafos arbitrários. A Tabela 2 (BRAGA, 2019) apresenta a complexidade do AtS para algumas classes específicas de grafos, na qual os casos em aberto estão identificados por ?.

Tabela 2 – Complexidade de AtS

Tipos de Grafos	$t=3$	t
cografo, split, complemento de bipartido	P (CAI, 1992)	P ($t \geq 1$) (NARAYANASWAMY; RAMAKRISHNA, 2015) NP-c ($t \geq 5$) (BRANDSTADT <i>et al.</i> , 2007) P ($t \geq 4$) (BRANDSTADT <i>et al.</i> , 1999) NP-c ($t \geq 4$) (BRANDSTADT <i>et al.</i> , 2004) NP-c ($t \geq 4$) (BRANDSTADT <i>et al.</i> , 2004) NP-c ($t \geq 4$) (BRANDSTADT <i>et al.</i> , 2004) NP-c ($t \geq 4$) (DRAGAN <i>et al.</i> , 2011)
intervalo, permutação, bipartido regular	P (MADANLAL <i>et al.</i> , 1996)	
planar	P (FEKETE; KREMER, 2001)	
exoplanar	P (VENKATESAN <i>et al.</i> , 1997)	
bipartido convexo	? (MADANLAL <i>et al.</i> , 1996)	
bipartido	? (MADANLAL <i>et al.</i> , 1996)	
grafo de cocomparabilidade	P (PAPOUTSAKIS, 2018)	
grau máximo $\leq b * \log n, b > 0$	P (BRANDSTADT <i>et al.</i> , 2004)	
fortemente cordal	P (BRANDSTADT <i>et al.</i> , 2004)	
1-split	P (BRANDSTADT <i>et al.</i> , 2004)	
cordal com diâmetro ≤ 2	P (BRANDSTADT <i>et al.</i> , 2004)	
cordal com diâmetro $\leq t + 1$ (para t par)		
cordal com diâmetro $\leq t + 2$ (para t ímpar)		
grafo com diâmetro ≤ 5	P (PAPOUTSAKIS, 2014)	
cordal	? (MADANLAL <i>et al.</i> , 1996)	
grafos que não possuem K_6 como menor		
P (Polinomial); NP-c (NP-completo)		

Fonte: Braga (2019), pág. 20.

Quando os pesos do grafo são arbitrários, falamos do AtSCM. Como instâncias do AtS são um subconjunto das instâncias do AtSCM, então os casos NP -completos pare aquele se traduzem em casos NP -difíceis para este. Mais ainda, Cai e Coneil (1995) mostram que, já para $t > 1$ fixo, verificar se existe uma árvore t -spanner em um grafo ponderado é um problema NP -completo. Para o caso em que $t = 1$, uma AtSCM*, caso exista, é, em particular, uma árvore

geradora mínima, e pode ser encontrada em $O(m \log \beta(m, n))$, onde $m = |E(G)|$, $n = |V(G)|$ e $\beta(m, n) = \min\{i \mid \log^{(i)} n \leq \frac{m}{n}\}$ (CAI; CONEIL, 1995).

Com respeito a abordagens para problemas de árvore t -spanner via programação matemática, apenas duas formulações de programação linear-inteira, ao nosso conhecimento, foram propostas na literatura para o AtSCM (BRAGA, 2019). Ambas as formulações são baseadas em determinar arborescências, enraizadas em diferentes vértices, que se sobrepõem, ou seja, as arestas utilizadas em cada arborescência são as mesmas, a diferir de orientações. Usar diferentes raízes é uma estratégia para calcular as distâncias na árvore. No Capítulo 5, apresentaremos formalmente as formulações da literatura e iremos propor duas novas formulações, além de estudá-las e derivar desigualdades válidas. Além disso, Braga (2019) apresenta um Algoritmo *Branch-and-Price* para o tSCM, que pode ser adaptado para o AtSCM. Em nosso trabalho, propomos, no Capítulo 4, um novo algoritmo enumerativo, que utiliza um procedimento de *Branch-and-Bound* e, diferente daquele já existente, não depende de formulações matemáticas para sua execução, mas de uma adaptação do clássico Algoritmo de Kruskal (KRUSKAL, 1956) para o cálculo de árvores geradoras mínimas.

3.2 Encontrando uma árvore t -spanner

Nesta seção, vamos apresentar algumas propriedades úteis para nortear a determinação de uma árvore t -spanner em um grafo G . Primeiro, recapitulamos dois resultados da literatura: (i) é suficiente impor a restrição (3.1) a pares de vértices que induzem arestas em G , (ii) no caso não ponderado, pode-se restringir o valor de t aos inteiros. Além disso, demonstraremos ser possível resolver o problema separadamente em cada bloco (componente 2-conexa) do grafo de entrada.

3.2.1 Restrições t -spanner Suficientes

Pela condição (3.1), devemos garantir que, em um t -spanner H , a distância entre quaisquer dois vértices i e j do grafo G é no máximo t vezes a distância entre i e j em G . Cai e Coneil (1995) mostram que não é necessário aplicar essa restrição a todos os pares de vértices. É suficiente impô-la apenas para os pares de vértices diretamente conectado em G . Além disso, é suficiente obrigar que a distância entre i e j em H seja no máximo o peso da aresta $\{i, j\}$. Uma apresentação formal dessas condições é resumida no Teorema 3.2.1.

Teorema 3.2.1 (CAI; CONEIL, 1995) *Seja H um subgrafo gerador de um grafo $G = (V, E, w)$. As seguintes afirmações são equivalentes:*

1. H é um t -spanner de G (respeita (3.1)).
2. Para toda aresta $\{i, j\} \in E(G)$, $\text{dist}_H(i, j) \leq t * \text{dist}_G(i, j)$.
3. Para toda aresta $\{i, j\} \in E(G) \setminus E(H)$, $\text{dist}_H(i, j) \leq t * \text{dist}_G(i, j)$.
4. Para toda aresta $\{i, j\} \in E(G)$, $\text{dist}_H(i, j) \leq t * w_{ij}$.
5. Para toda aresta $\{i, j\} \in E(G) \setminus E(H)$, $\text{dist}_H(i, j) \leq t * w_{ij}$.

Mais ainda, se H é árvore geradora, então as afirmações 2-5 são equivalentes a H ser árvore t -spanner de G .

Prova. As implicações $1 \rightarrow 2$, $2 \rightarrow 3$ e $4 \rightarrow 5$ são triviais, visto que se a desigualdade vale para um conjunto de pares de vértice, então ela vale para um subconjunto deste.

Para mostrar que $3 \rightarrow 4$, tome $\{i, j\} \in E(G)$ e $t \geq 1$. Se $\{i, j\} \notin E(H)$, temos que $\text{dist}_H(i, j) \leq t * \text{dist}_G(i, j)$ por hipótese e, portanto, $\text{dist}_H(i, j) \leq t * w_{ij}$, pois $\text{dist}_G(i, j) \leq w_{ij}$. Se $\{i, j\} \in E(H)$, temos trivialmente $\text{dist}_H(i, j) \leq w_{ij} \leq t * w_{ij}$, pois $w_{ij} \geq 0$.

Para $5 \rightarrow 1$, sejam i e j vértices distintos de G , e $t \geq 1$. Devemos mostrar que $\text{dist}_H(i, j) \leq t * \text{dist}_G(i, j)$. Seja P um i, j -caminho mínimo em G . Para toda aresta $\{u, v\} \in P$, se $\{u, v\} \in E(H)$, temos que $\text{dist}_H(u, v) \leq w_{uv} \leq t * w_{uv}$; caso contrário, $\text{dist}_H(u, v) \leq t * w_{uv}$, por hipótese. Portanto:

$$\text{dist}_H(i, j) \leq \sum_{\{u,v\} \in P} \text{dist}_H(u, v) \leq \sum_{\{u,v\} \in P} t * w_{uv}$$

Como P é um i, j -caminho mínimo em G , temos que $\text{dist}_G(i, j) = \sum_{\{u,v\} \in P} w_{uv}$. Então

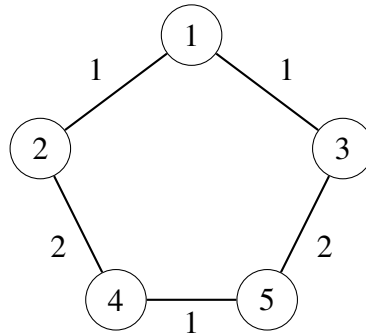
$$\text{dist}_H(i, j) \leq t * \text{dist}_G(i, j)$$

Finalmente, quando H é árvore geradora, as equivalências seguem analogamente para árvore t -spanner. ■

Pelo item 4 do Teorema 3.2.1, observe que só precisamos garantir que a distância, em um t -spanner H , entre quaisquer dois vértices i e j , adjacentes em G , seja no máximo t vezes o custo da aresta $\{i, j\}$ em G . Caso o problema em questão seja a versão com pesos unitários, ou seja, o AtS, então essa restrição passa a ser $\text{dist}_H(i, j) \leq t$. Mais ainda, como $\text{dist}_H(i, j)$ é um valor inteiro, já que a distância será dada pela quantidade de arestas no caminho entre i e j , podemos expressar a restrição como $\text{dist}_H(i, j) \leq \lfloor t \rfloor$. Isso implica que um t -spanner também é um $\lfloor t \rfloor$ -spanner. Em outras palavras, podemos admitir que t é inteiro quando se trata do AtS.

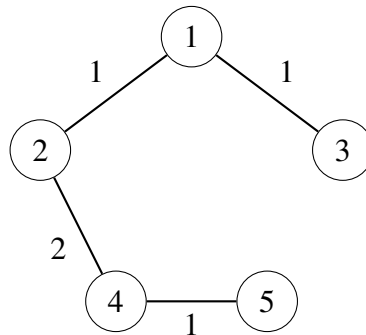
O mesmo não vale para o caso com pesos arbitrários. Apresentamos um contra-exemplo no grafo da Figura 9, onde uma árvore 2,5-spanner de G , mostrada na Figura 10, não é uma árvore 2-spanner, pois a distância entre os vértices 3 e 5 é 5, que é maior que $2 * w_{35} = 4$.

Figura 9 – grafo G , $t = 2,5$



Fonte: elaborado pelo autor.

Figura 10 – Árvore 2,5-spanner



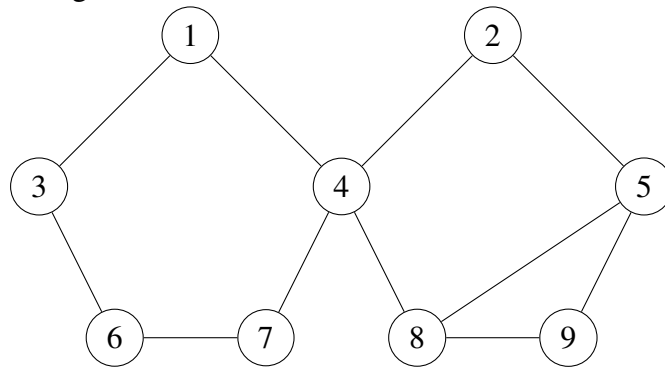
Fonte: elaborado pelo autor.

3.2.2 Componente 2-conexa

Um subgrafo de G que é conexo e maximal, com respeito a propriedade de ser conexo, é dito uma *componente* (conexa). Dizemos que um vértice $v \in V(G)$ é um *vértice de corte* ou *articulação*, se $G - v$ tem mais componentes que G . Em particular, se G é conexo, v é vértice de corte se, e somente se, $G - v$ é desconexo. Um grafo conexo sem vértices de corte é chamado de grafo 2-conexo (*bloco*). Chamamos um subgrafo B de G de *bloco de G* , se B é um bloco e maximal com respeito a essa propriedade, ou seja, B é conexo e, para todo $v \in B$, temos que $B - v$ é conexo, mas para todo $v \in V(G) \setminus B$, $B \cup \{v\}$ é desconexo ou tem vértice de corte. No grafo da Figura 11 temos dois blocos B_1 e B_2 , onde $V(B_1) = \{1, 3, 4, 6, 7\}$ e $V(B_2) = \{2, 4, 5, 8, 9\}$. Ele possui apenas a articulação 4.

A decomposição em blocos de um grafo consiste em dividi-lo em seus blocos,

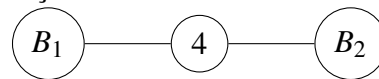
Figura 11 – Blocos de um grafo



Fonte: elaborado pelo autor.

gerando uma árvore, chamada árvore de bloco-articulação. Essa árvore tem um vértice para cada bloco e um vértice para cada articulação, bem como uma aresta entre um bloco e uma articulação se esta pertence àquele. A Figura 12 ilustra a árvore bloco-articulação correspondente ao grafo da Figura 11. Observe que uma folha de uma árvore bloco-articulação corresponde a um bloco com uma única articulação, por isso chamado bloco-folha. Por outro lado, toda articulação define um vértice interno da árvore, pois está na interseção de pelo menos dois blocos. Na verdade, os vizinhos da articulação na árvore são exatamente os blocos a que ela pertence.

Figura 12 – Árvore Bloco-Articulação



Fonte: elaborado pelo autor.

Lema 3.2.1 *Sejam G um grafo conexo, k um vértice de corte de G e C o conjunto de vértices de uma componente conexa de $G \setminus \{k\}$. Se $G' = G[C \cup \{k\}]$ ou $G' = G \setminus C$, então $\text{dist}_{G'}(u, v) = \text{dist}_G(u, v)$ para todos $u, v \in V(H)$.*

Prova. Como G' é subgrafo de G , então $\text{dist}_{G'}(u, v) \geq \text{dist}_G(u, v)$. Por outro lado, como k é vértice de corte em G , temos nas duas opções para G' que, para quaisquer $u, v \in V(G')$, todo u, v -caminho em G é um u, v -caminho em G' . Logo, $\text{dist}_{G'}(u, v) \leq \text{dist}_G(u, v)$. ■

Com o auxílio do lema acima, provamos a seguir que os problemas tSCM e AtSCM podem ser decompostos em subproblemas restritos a cada bloco do grafo.

Proposição 3.2.2 *Se G é um grafo conexo e B_1, B_2, \dots, B_p são seus blocos, então:*

$$tSCM^*(G) = \bigcup_{\ell=1}^p tSCM^*(B_\ell), \quad (3.2)$$

$$AtSCM^*(G) = \bigcup_{\ell=1}^p AtSCM^*(B_\ell), \quad (3.3)$$

onde $tSCM^*(G')$ e $AtSCM^*(G')$ representam uma solução ótima de $tSCM$ e $AtSCM$ sobre o grafo G' .

Prova. Se G é um bloco, o resultado vale trivialmente. Do contrário, sejam G_1 um bloco-folha de G e k o vértice de corte de G em G_1 . Seja $G_2 = G \setminus (V(G_1) \setminus \{k\})$. Note que G_1 e G_2 são subgrafos de G que satisfazem as condições do Lema 3.2.1. Vamos mostrar que $tSCM^*(G) = tSCM^*(G_1) \cup tSCM^*(G_2)$ e $AtSCM^*(G) = AtSCM^*(G_1) \cup AtSCM^*(G_2)$. Então, aplicando recursivamente essa estratégia a G_2 , chegamos ao resultado desejado.

Sejam $H^* = AtSCM^*(G)$ e H_i o subgrafo de H^* induzido pelos vértices de G_i , para $i \in \{1, 2\}$. Note que k é vértice de corte de H^* e que H_1 e H_2 satisfazem as condições do Lema 3.2.1. Então, para todos $u, v \in V(G_i)$, temos que $\text{dist}_{H_i}(u, v) = \text{dist}_{H^*}(u, v) \leq t * \text{dist}_G(u, v) = t * \text{dist}_{G_i}(u, v)$, onde a desigualdade decorre de H^* ser t -spanner de G . Logo, H_i é t -spanner para G_i e, por conseguinte,

$$w(AtSCM^*(G_1)) + w(AtSCM^*(G_2)) \leq w(H_1) + w(H_2) = w(AtSCM^*(G)). \quad (3.4)$$

Considere agora $H_i^* = AtSCM^*(G_i)$, para $i \in \{1, 2\}$, e defina $H = H_1^* \cup H_2^*$. Temos que k é vértice de corte para H e H_1^* e H_2^* satisfazem as condições do Lema 3.2.1. Vamos mostrar que H é um t -spanner de G . Sejam dois vértices u e v quaisquer de G . Se u e v estão no mesmo subgrafo G_i (ou equivalentemente H_i^*), pelo Lema 3.2.1, $\text{dist}_H(u, v) = \text{dist}_{H_i^*}(u, v) \leq t * \text{dist}_{G_i}(u, v) = t * \text{dist}_G(u, v)$. Caso contrário, podemos admitir s.p.g. que $u \in G_1 \setminus \{k\}$ e $v \in G_2 \setminus \{k\}$. Observe que todo caminho entre u e v , em H ou em G , possui k como nó interno, em particular, todo caminho mínimo. Assim, $\text{dist}_H(u, v) = \text{dist}_H(u, k) + \text{dist}_H(k, v) \leq t * \text{dist}_G(u, k) + t * \text{dist}_G(k, v) = t * \text{dist}_G(u, v)$, onde a desigualdade deve-se ao fato de que $u, k \in V(G_1)$ e $v, k \in V(G_2)$. Portanto, H é um t -spanner em G . Desse modo, concluímos que

$$w(AtSCM^*(G)) \leq w(H) = w(AtSCM^*(G_1)) + w(AtSCM^*(G_2)). \quad (3.5)$$

De (3.4) e (3.5), concluímos que igualdades acontecem nas duas expressões. Logo, H_1 e H_2 são ótimos para $AtSCM(G_1)$ e $AtSCM(G_2)$, respectivamente, assim como H é ótimo para

$\text{AtSCM}(G)$. Isso prova (3.2). Para obter (3.3), a demonstração é similar, bastando notar adicionalmente que: (i) se H^* é árvore, então $H_1 = H^*[V(G_1)]$ e $H_2 = H^*[V(G_2)]$ também o são; (ii) se H_1^* e H_2^* são árvores, então $H = H_1^* \cup H_2^*$ também é árvore.



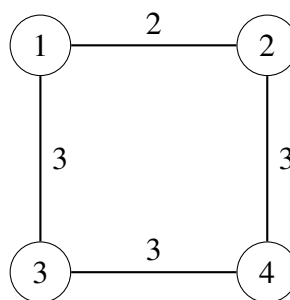
4 ALGORITMO ENUMERATIVO

Neste capítulo, iremos apresentar um algoritmo para determinar uma árvore t -spanner de custo mínimo de G ou mostrar que tal árvore não existe. O algoritmo é baseado em um procedimento enumerativo de ramificação e poda (*branch-and-bound*), no qual cada subproblema é solucionado como um problema de árvore geradora mínima, por meio de uma adaptação do clássico algoritmo de Kruskal (KRUSKAL, 1956). Denotaremos por AtSCM^* e AGM^* uma solução ótima do AtSCM e do AGM, respectivamente.

Seja (G, w, t) uma instância do AtSCM. Note que o peso de uma AGM^* é um limitante inferior para uma AtSCM^* , já que qualquer AtSCM^* é uma solução viável para o AGM. Se uma AGM^* de G for um t -spanner, então a própria será uma AtSCM^* . Do contrário, significa que alguma restrição (3.1) está violada, ou seja, existe algum i, j -caminho nessa AGM^* cujo peso é maior que $t * \text{dist}_G(i, j)$.

Seguindo essa constatação, a ideia chave do nosso algoritmo é impedir, dada uma árvore geradora T que não é t -spanner, um i, j -caminho de T que viola a restrição (3.1) relativa ao par (i, j) , o qual denominaremos de caminho t -inviável. Formalmente, um caminho P é dito t -inviável se $\text{dist}_P(i, j) > t * \text{dist}_G(i, j)$. Por exemplo, considerando a instância do AtSCM apresentada na Figura 13, na Figura 14 o caminho $P = \langle 1, 3, 4, 2 \rangle$ é 3-inviável, pois $\text{dist}_P(1, 2) = 9 > 6 = 3 * \text{dist}_G(1, 2)$.

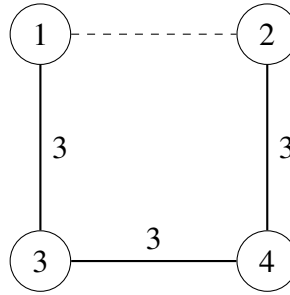
Figura 13 – Grafo G , $t = 3$



Fonte: elaborado pelo autor.

O algoritmo começa encontrando uma AGM^* . Se ela for um t -spanner, paramos o procedimento, $\text{AtSCM}^* = \text{AGM}^*$. Caso contrário, selecionamos um caminho t -inviável de AGM^* e ramificamos o problema em novos subproblemas, de modo que esse caminho não seja selecionado para solução. Em cada subproblema proibimos uma aresta do caminho t -inviável. Veremos com detalhes na seção a seguir como o processo de ramificação ocorre.

Figura 14 – Caminho 3-inviável entre 1 e 2



Fonte: elaborado pelo autor.

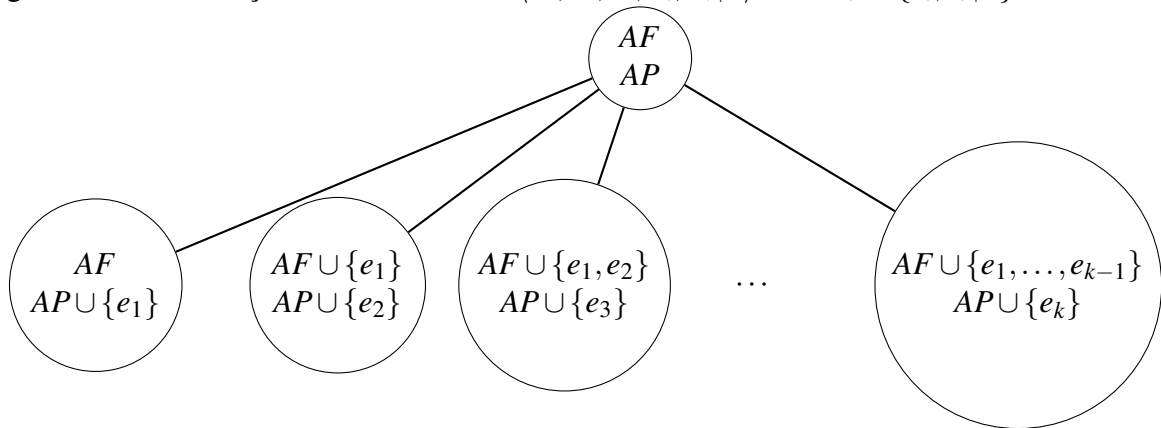
4.1 Ramificação

Seja $P = \langle v_1, v_2, \dots, v_k, v_{k+1} \rangle$, $k \geq 1$, um caminho t -inviável. Então, as arestas $e_i = \{v_i, v_{i+1}\}$, para $i = 1, \dots, k$, não podem ser selecionadas simultaneamente para compor uma árvore t -spanner, ou seja, no máximo $k - 1$ dessas arestas pertencem a uma árvore t -spanner. Vamos dividir o conjunto viável em k subconjuntos disjuntos, de modo que no i -ésimo deles a aresta e_i é proibida e as arestas e_1, \dots, e_{i-1} são obrigatórias. A melhor solução, caso exista, dos subproblemas definidos por esses subconjuntos é solução do problema original. Essa ramificação pode ser aplicada recursivamente, gerando uma árvore. Vale ressaltar que esta estratégia de ramificação já é conhecida na literatura e foi utilizada, por exemplo, para o problema de roteamento de veículos capacitado assimétrico e o problema da árvore geradora mínima com restrição de grau (veja BATSYN *et al.* (2013), ANDRADE e FREITAS (2013)).

Formalmente, seja $S = (G, AF, AP)$ um nó dessa árvore, onde $AF \subseteq E(G)$ e $AP \subseteq E(G)$ são os subconjuntos de arestas fixas e proibidas nesse nó, respectivamente. Denotamos por $AGM(S)$ o problema de determinar uma árvore geradora mínima de $G \setminus AP$ contendo AF (ver Subseção 4.1.1). Caso exista, seja T uma solução desse problema. Se satisfizer (3.1), T é árvore t -spanner ótima para o AtSCM restrito a S . Suponha, então, $P = \langle v_1, v_2, \dots, v_k, v_{k+1} \rangle$ um caminho em T que é t -inviável e seja $e_i = \{v_i, v_{i+1}\}$, para $i = 1, 2, \dots, k$. Nesse caso, S será ramificado em k subproblemas, $S_i = (G, AF_i, AP_i)$, para $i \in \{1, 2, \dots, k\}$, onde $AF_1 = AF$, $AP_1 = AP \cup \{e_1\}$ e $AF_i = AF \cup \{e_1, \dots, e_{i-1}\}$, $AP_i = AP \cup \{e_i\}$, para $2 \leq i \leq k$. Ver Figura 15.

Note que qualquer árvore geradora de que P faz parte (em particular T) é inviável para cada subproblema $AGM(S_i)$, pois proíbe a aresta $e_i \in E(P)$. Além disso, toda árvore geradora de $G \setminus AP$ contendo AF é viável para um desses subproblemas, a menos daquelas que contêm P . De fato, considere uma árvore T^* que é viável para $AGM(S)$ e não contém todas as arestas de P . Seja e_c a primeira aresta de P que não está em T^* e, portanto, $\{e_1, \dots, e_{c-1}\} \subset E(T^*)$.

Figura 15 – Ramificação no caminho $P = \langle v_1, v_2, \dots, v_k, v_{k+1} \rangle$, com $e_i = \{v_i, v_{i+1}\}$

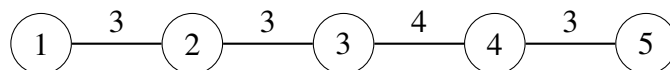


Fonte: elaborado pelo autor.

Como T^* é solução viável de $AGM(S)$, temos que $AF \subset E(T^*)$ e $AP \cap E(T^*) = \emptyset$. Então, T^* é viável para S_c , pois todas as arestas de $AF_c = AF \cup \{e_1, \dots, e_{c-1}\}$ estão em $E(T^*)$, enquanto toda aresta de $AP_c = AP \cup \{e_c\}$ não está. Em outras palavras, a solução do problema original está em um desses subproblemas, sendo estes uma versão mais restrita do problema, com um conjunto de arestas fixas e outras proibidas. Eventualmente, iremos encontrar, caso exista, uma solução que não contém caminhos t -inviáveis, ou seja, uma árvore t -spanner.

Dada uma solução de um subproblema, podemos ter vários caminhos t -inviáveis, todos candidatos a gerar uma ramificação. Vamos selecionar, dentre eles, aquele que tem o menor custo. Considere, por exemplo, um grafo G , com fator de dilatação $t = 3$, no qual $\text{dist}_G(1, 5) = 4$ e $\text{dist}_G(3, 5) = 2$. Suponha que a árvore solução de um determinado subproblema contém o caminho apresentado na Figura 16. Note que são 3-inviáveis tanto o caminho entre 1 e 5 quanto o caminho entre 3 e 5. Nas Figuras 17 e 18 apresentamos a ramificação desta solução caso um desses dois caminhos seja selecionado. Note que a ramificação relativa ao caminho $\langle 3, 4, 5 \rangle$ está proibindo, além dele, o caminho $\langle 1, 2, 3, 4, 5 \rangle$, pois as arestas daquele são um subconjunto das arestas deste. Além disso, a quantidade de subproblemas gerada é menor, pelo fato do número de arestas do caminho $\langle 3, 4, 5 \rangle$ ser estritamente menor. Dessa forma vamos selecionar um dos caminhos t -inviáveis que tenha o menor peso para gerar a ramificação, pois evitamos que ela seja feita em um caminho t -inviável, no qual há um subcaminho também t -inviável.

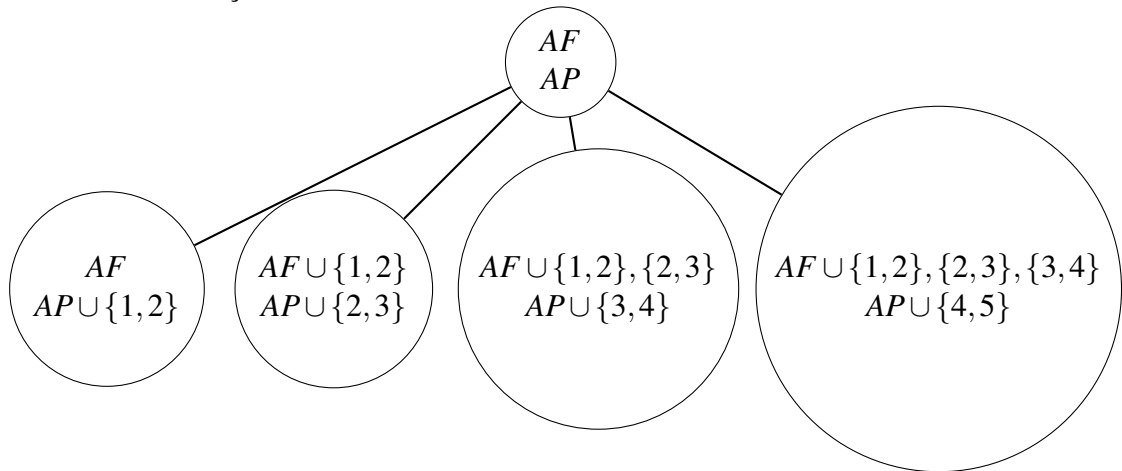
Figura 16 – Caminhos t -inviáveis para ramificação



Fonte: elaborado pelo autor.

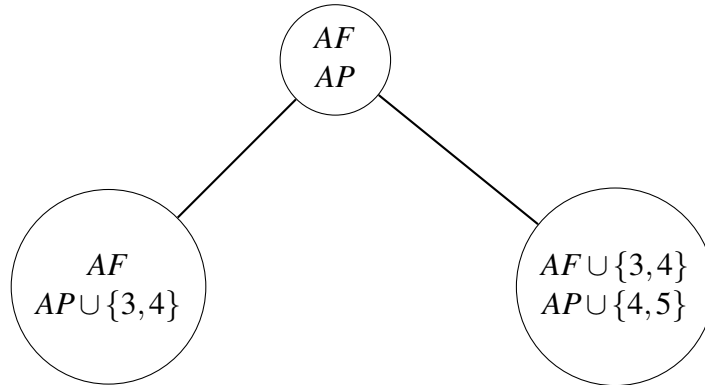
Um pseudocódigo do procedimento de ramificação encontra-se no Algoritmo 2. A

Figura 17 – Ramificação no caminho 1-5



Fonte: elaborado pelo autor.

Figura 18 – Ramificação no caminho 3-5



Fonte: elaborado pelo autor.

árvore de *branch-and-bound* é armazenada numa pilha, *B&BTree*, de modo que a enumeração segue um processo tipo *backtracking*. Cada nó da árvore guarda, além de $S = (G, AF, AP)$, o valor de um limite inferior para o subproblema restrito a S , a ser dado pelo ótimo de $AGM(S)$, ou seja, pelo peso mínimo de uma árvore geradora de $G \setminus AP$ contendo AF . Na criação do nó, esse limite inferior é inicializado com aquele do nó pai.

Dada uma árvore geradora T^* , solução de $AGM(S)$, para selecionar o caminho t -inviável para o algoritmo de ramificação (passo 1), necessitamos conhecer o custo de cada caminho na árvore, bem como as arestas do mesmo. Para tal, utilizaremos uma estratégia de busca em profundidade na árvore para determiná-los, conforme descrito no Algoritmo 3, o qual recebe como entrada a árvore e a matriz de distâncias do grafo original. Realizamos $|V|$ buscas em profundidade, cada uma partindo de cada vértice $i \in V(G)$ do grafo. Ao final de cada busca determinamos os caminhos, bem como seus custos, de i a todos os outros vértices na árvore. Portanto, ao fim das $|V|$ buscas, basta selecionar dentre os caminhos t -inviáveis (basta verificar

Algoritmo 2: Ramificação ($B\&BTree, T^*$)

```

1:  $\{e_1, e_2, \dots, e_k\} \leftarrow \text{MenorCaminhoViolado}(T^*, \text{dist}_G)$ 
2:  $(G, AF, AP, w) \leftarrow B\&BTree[\text{Tamanho}(B\&BTree)]$ 
3: Desempilhar  $B\&BTree$ 
4:  $AF^* \leftarrow AF$ 
5: for  $i \leftarrow 1$  to  $k$  do
6:    $AP^* \leftarrow AP \cup \{e_i\}$ 
7:   for  $j \leftarrow 1$  to  $i - 1$  do
8:      $AF^* \leftarrow AF^* \cup \{e_j\}$ 
9:   end for
10:  for  $e' \in E(G) \setminus AF^*$  do
11:    if  $e' \cup AF^*$  induz ciclo then
12:       $AP^* \leftarrow AP^* \cup e'$ 
13:    end if
14:  end for
15:  Empilhar  $(G, AF^*, AP^*, w(T^*))$  em  $B\&BTree$ 
16: end for
17: Retorne

```

se o custo do caminho viola (3.1)) o de menor custo. Como observação, utilizamos este método para cálculo dos caminhos na árvore pois é mais “barato” computacionalmente que o algoritmo de Floyd-Warshall. Vamos definir $N_{T^*}(i).prox$, no Algoritmo 3, como um vértice $v \in V$ na vizinhança de i em T^* que não foi encontrado, até então, pela busca corrente.

4.1.1 Resolução de subproblemas

Para resolver cada subproblema $AGM(S)$, vamos adaptar o Algoritmo de Kruskal (KRUSKAL, 1956), utilizado para determinar uma árvore geradora mínima de um grafo G . Observe que $AGM(G, \emptyset, \emptyset)$ corresponde ao AGM em G . De maneira geral, o algoritmo de Kruskal recebe como entrada um grafo conexo ponderado $G = (V, E, w)$ e uma ordenação $E^* = (e_1^*, e_2^*, \dots, e_{|E(G)|}^*)$ do seu conjunto de arestas, onde $w(e_i^*) \leq w(e_{i+1}^*)$, para $i \in \{1, \dots, |E(G)| - 1\}$. Iniciamos com um subgrafo gerador $T = (V(G), \emptyset)$, sem arestas. A cada iteração i de 1 a $|E(G)|$, caso a adição da aresta e_i^* não gere ciclo em T , então a adicionamos em $E(T)$; em outras palavras, se $(V(G), E(T) \cup \{e_i^*\})$ é acíclico, então $E(T) \leftarrow E(T) \cup e_i^*$. Paramos quando $|E(T)| = |V| - 1$. Um pseudocódigo do procedimento está apresentado no Algoritmo 4.

Vamos adaptar o Algoritmo 4 para solucionar um dado subproblema $AGM(S)$, com $S = (G, AF, AP)$. Para o conjunto AP , vamos apenas supor que as arestas não “existem” no grafo de entrada do algoritmo, ou seja, vamos ignorar cada aresta de AP no passo 4 do algoritmo. Já para as aresta de AF , vamos inicializar o método com elas pré-selecionadas, mudando assim o

Algoritmo 3: MenorCaminhoViolado (T^* , dist_G)

```

1:  $P_{|V| \times |V|} \leftarrow \emptyset, \bar{W}_{|V| \times |V|} \leftarrow 0$ 
2: for  $\ell \leftarrow 1$  to  $|V|$  do
3:    $k \leftarrow 1, \bar{W} \leftarrow 0, \bar{P} \leftarrow \emptyset, i \leftarrow \ell, i' \leftarrow 0$ 
4:   while  $k < |V|$  do
5:     if  $N_{T^*}(i).prox = \emptyset$  then
6:        $\bar{W} \leftarrow \bar{W} - w_{ii'}$ 
7:        $\bar{P} \leftarrow \bar{P} \setminus \{i, i'\}$ 
8:        $i \leftarrow i'$ 
9:     else
10:       $j \leftarrow N_{T^*}(i).prox$ 
11:       $\bar{W} \leftarrow \bar{W} + w_{ij}$ 
12:       $\bar{P} \leftarrow \bar{P} \cup \{i, j\}$ 
13:       $i' \leftarrow i$ 
14:       $i \leftarrow j$ 
15:       $P(\ell, i) \leftarrow \bar{P}$ 
16:       $W(\ell, i) \leftarrow \bar{W}$ 
17:       $k \leftarrow k + 1$ 
18:     end if
19:   end while
20: end for
21:  $(u, v) \leftarrow \arg \min \{W(u, v) : W(u, v) > t * \text{dist}_G(u, v)\}$ 
22: Retorne  $P(u, v)$ 

```

Algoritmo 4: Kruskal (G, E^*)

```

1:  $T \leftarrow (V(G), \emptyset)$ 
2:  $i \leftarrow 1$ 
3: while  $|E(T)| < |V(G)| - 1$  do
4:   if  $(V(G), E(T) \cup e_i^*)$  é acíclico then
5:      $E(T) \leftarrow E(T) \cup e_i^*$ 
6:   end if
7:    $i \leftarrow i + 1$ 
8: end while
9: Retorne  $T$ 

```

passo 1 para $T \leftarrow (V(G), AF)$, e ignorando-as, similarmente como feito com AP , no passo 4. Com essa modificação temos o pseudocódigo deste novo procedimento no Algoritmo 5.

Algoritmo 5: Kruskal Modificado ($S=(G, AF, AP), E^*$)

```

1:  $T \leftarrow (V(G), AF)$ 
2:  $i \leftarrow 1$ 
3: while  $|E(T)| < |V(G)| - 1$  e  $i \leq |E^*|$  do
4:   if  $(V(G), E(T) \cup e_i^*)$  é acíclico,  $e_i^* \notin AF$  e  $e_i^* \notin AP$  then
5:      $E(T) \leftarrow E(T) \cup e_i^*$ 
6:   end if
7:    $i \leftarrow i + 1$ 
8: end while
9: Retorne  $T$ 

```

O peso $w(T)$ da árvore geradora T retornada pelo Algoritmo 5 é o limite inferior do nó corrente e deveria ser armazenado. Entretanto, como iremos ramificar imediatamente esse nó, isso não será necessário.

4.2 Critérios de Poda

Ao término da resolução de um subproblema, abstraímos da solução algumas informações. O menor de todos os custos de solução (relaxada) dos subproblemas resolvidos, até então, gera um limitante inferior para o custo da AtSCM*. Similarmente, o menor custo encontrado, dentre as soluções que representem árvores t -spanner, gera um limitante superior para a solução do problema. Com base nessas informações podemos ignorar subproblemas, em que a melhor solução que ele possa gerar para o AtSCM, seja maior que uma solução viável já encontrada. A seguir vamos apresentar os critérios de poda de um determinado subproblema $S = (G, AF, AP)$ do algoritmo enumerativo. O pseudocódigo do procedimento de poda está no Algoritmo 6.

4.2.1 Poda por Otimalidade

Se a solução do subproblema $AGM(S)$ corrente for uma árvore t -spanner, então esta solução é ótima para o AtSCM restrito a S e viável o AtSCM em G . Nesse caso, podemos esse subproblema por otimalidade. Além disso, caso o custo da solução seja o melhor já encontrado (o custo da árvore encontrada seja menor que o menor já identificado até então), atualizamos o limitante superior para a solução ótima, com o valor desta solução viável. Ver passos 1–8 do

Algoritmo 6: Poda ($B\&BTree, T^*, AP, UB, T_{OPT}$)

```

1: if  $T^*$  é uma árvore  $t$ -spanner then
2:   if  $w(T^*) < UB$  then
3:      $UB \leftarrow w(T^*)$ 
4:      $T_{OPT} \leftarrow T^*$ 
5:   end if
6:   Desempilhar  $B\&BTree$ 
7:   Retorne 1
8: end if
9: if  $w(T^*) \geq UB$  then
10:  Desempilhar  $B\&BTree$ 
11:  Retorne 1
12: end if
13: for all  $i, j \in V(G)$  do
14:  if  $\text{dist}_{(V, E \setminus AP)}(i, j) > t * \text{dist}_G(i, j)$  then
15:    Desempilhar  $B\&BTree$ 
16:    Retorne 1
17:  end if
18: end for
19: Retorne 0

```

Algoritmo 6. Em particular, o teste realizado no passo 1 pode ser feito com o Algoritmo 3. Após o passo 18 desse algoritmo, se $W(u, v) \leq t * \text{dist}_G(u, v)$, para todo $\{u, v\} \in E(G)$, temos que T^* é uma árvore t -spanner.

4.2.2 Poda por Bound

Caso o custo da solução de um subproblema $AGM(S)$ seja maior ou igual que o limitante superior da solução ótima, podemos este subproblema por *bound*, pois a melhor solução que esse subproblema poderia gerar é, no melhor caso (em termos de função objetivo), uma solução de custo igual ao atual limitante superior. Ver passos 9–12 do Algoritmo 6.

4.2.3 Poda por Inviabilidade

Caso existam dois vértices i e j nos quais $\text{dist}_{(V, E \setminus AP)}(i, j) > t * \text{dist}_G(i, j)$, então podemos esse subproblema por inviabilidade, visto que o caminho mínimo entre i e j no grafo do subproblema já viola a restrição (3.1). Essa poda corresponde aos passos 13–18 do Algoritmo 6.

4.3 Procedimento Enumerativo

Agora iremos combinar os procedimentos vistos acima para definir nosso algoritmo enumerativo, cujo pseudocódigo é apresentado no Algoritmo 7. Como antecipado, vamos armazenar nossa estrutura em uma pilha, que denominaremos $B\&BTree$, na qual o último subproblema empilhado será o primeiro a ser solucionado. Cada elemento i da pilha $B\&BTree$ irá armazenar as arestas fixas (AF) e proibidas (AF) naquele nó, além de um limitante inferior para a solução desse subproblema (w). LB e UB irão armazenar, respectivamente, os limitantes inferior e superior para a AtSCM*. Note que $LB = \min_{1 \leq i \leq \text{Tamanho}(B\&BTree)} w(i)$. O critério de otimalidade será busca completa, de modo que o espaço de busca será totalmente explorado, a menos de subproblemas podados, ou a detecção, em alguma etapa do algoritmo, de um limitante inferior e um limitante superior de mesmo valor, o que garante otimalidade.

Algoritmo 7: Algoritmo Enumerativo (G, t)

- 1: Pilha $B\&BTree \leftarrow \emptyset$
 - 2: $E^* \leftarrow$ Ordenação de $E(G)$ por pesos crescentes
 - 3: $T \leftarrow$ Kruskal Modificado ($G, \emptyset, \emptyset, E^*$)
 - 4: **if** T é t -spanner **then**
 - 5: Retorne T
 - 6: **end if**
 - 7: Empilhar ($G, \emptyset, \emptyset, w(T)$) em $B\&BTree$
 - 8: $UB \leftarrow |V(G)| * \max_{(i,j) \in E(G)} \{w_{ij}\}$
 - 9: $T^* \leftarrow \emptyset$
 - 10: $LB \leftarrow 0$
 - 11: **while** $\text{Tamanho}(B\&BTree) > 0$ e $LB < UB$ **do**
 - 12: Ramificação ($B\&BTree, T$)
 - 13: $j \leftarrow \text{Tamanho}(B\&BTree)$
 - 14: $LB \leftarrow \min_{1 \leq i \leq j} w(B\&BTree[i])$
 - 15: $T \leftarrow$ Kruskal Modificado ($G, AF(B\&BTree[j]), AP(B\&BTree[j]), E^*$)
 - 16: **while** Poda ($B\&BTree, T, AP(B\&BTree[j]), UB, T^*) = 1$ e $j > 0$ **do**
 - 17: $LB \leftarrow \min_{1 \leq i \leq j} w(B\&BTree[i])$
 - 18: $j \leftarrow j - 1$
 - 19: $T \leftarrow$ Kruskal Modificado ($G, AF(B\&BTree[j]), AP(B\&BTree[j]), E^*$)
 - 20: **end while**
 - 21: **end while**
 - 22: Retorne T^*
-

5 FORMULAÇÕES MATEMÁTICAS

Neste capítulo, apresentaremos formulações matemáticas para o AtSCM, duas delas foram as primeiras formulações apresentadas para o problema em Braga (2019). Adicionalmente, propomos duas novas formulações, uma compacta e outra com uma quantidade polinomial de variáveis, mas exponencial de restrições. Adicionalmente propomos desigualdades válidas para as formulações, inspiradas em soluções fracionárias (soluções das formulações com a remoção das restrições de integralidade).

5.1 Formulação Exponencial

Seja P_{ij} um caminho t -inviável entre os vértices i e j no grafo G . Observe que todas as árvores geradoras que têm como subconjunto de arestas aquelas desse caminho são inviáveis para o AtSCM, pois o custo de P_{ij} é maior que t vezes a distância entre i e j no grafo original. Dessa forma, toda árvore t -spanner é uma árvore geradora que não possui caminho t -inviável. Árvores geradoras que contêm caminho t -inviável serão denominadas de árvores inviáveis.

Assim, podemos deduzir que, dentre as $|P|$ arestas de um caminho t -inviável P , podemos selecionar no máximo $|P| - 1$ para a construção de uma árvore t -spanner. Essa condição nos fornece uma formulação matemática para o problema, cujas variáveis representam a presença ou não de uma aresta do grafo na solução e as restrições se separam em dois tipos: no primeiro conjunto a solução deverá fornecer uma árvore geradora, e no segundo a solução não pode conter caminhos t -inviáveis. De forma geral, nossa formulação, que denominaremos como formulação exponencial (FE), pois podemos ter uma quantidade exponencial de caminhos t -inviáveis, pode ser expressa como:

$$\min \sum_{e \in E(G)} w_e x_e \quad (5.1)$$

$$\text{s.a: } X = [x_e]_{e \in E(G)} \text{ é árvore geradora} \quad (5.2)$$

$$\sum_{e \in E(P)} x_e \leq |P| - 1, \quad \forall P \in CI \quad (5.3)$$

$$x_e \in \mathbb{B} \quad \forall e \in E(G) \quad (5.4)$$

onde CI representa o conjunto de todos os caminhos t -inviáveis de G . A variável x_e será 1, se a aresta $e \in E(G)$ estiver na solução, ou 0, caso contrário. Por (5.2), representamos um conjunto de restrições que irão impor que a atribuição de valores às variáveis x_e descrevam

uma árvore geradora de $G = (V, E)$. Para cada caminho t -inviável $P \in CI$, temos uma restrição correspondente em (5.3), garantindo que selecionamos no máximo $|P| - 1$ das arestas deste caminho para compor a árvore solução.

Vale observar que o algoritmo enumerativo proposto no Capítulo 4 pode ser visto como um *branch-and-bound* que toma como base a formulação acima. Em lugar de usar a relaxação linear e forçar a integralidade através da ramificação, ele considera a relaxação obtida com a remoção das restrições (5.3) para o *bounding* enquanto assegura o cumprimento delas com o *branching*. Nesse caso, a relaxação consiste em um problema de árvore geradora mínima, com algumas arestas fixadas e outras proibidas devido à ramificação, e pode ser resolvido sem uma formulação matemática.

Para empregar um *branch-and-bound* baseado em relaxação linear, porém, precisamos usar em (5.3) restrições que descrevam um árvore geradora. Existem na literatura do AGM várias formulações matemáticas (veja algumas em Magnanti e Wolsey (1995)). Dessa maneira, podemos entender (FE) como uma classe de formulações. Para construir *uma* (FE), utilizaremos uma formulação qualquer para o AGM e adicionaremos o conjunto de restrições (5.3) em sua descrição. Como exemplo, apresentamos duas formulações para o AGM, baseadas em formulações para o problema do caixeiro viajante (PCV): o modelo de eliminação de sub-rotas proposto por Dantzig *et al.* (1954) e o modelo proposto por Miller *et al.* (1960). No decorrer do texto, denominaremos essas formulações, respectivamente, como SEC's (*Subtour elimination constraints*) e MTZ (Miller-Tucker-Zemli).

Para as formulações, dado o grafo de entrada G , defina os seguintes parâmetros:

- $V(G)$: conjunto de vértices de G .
- $E(G)$: conjunto de arestas de G .
- $A(G) = \{(i, j), (j, i) \mid \{i, j\} \in E(G)\}$: conjunto de arcos da orientação simétrica de G , onde cada aresta $\{i, j\}$ dá lugar aos arcos $i \leftarrow j$ e $i \rightarrow j$.
- CI : conjunto de todos os caminhos t -inviáveis de G .

5.1.1 Formulação SEC's

A ideia da formulação SEC's é garantir duas das propriedades do Teorema 2.2.1: o subgrafo escolhido contém $|V(G)| - 1$ arestas e é acíclico. Para tal defina as variáveis:

$$- x_e = \begin{cases} 1, & \text{se a aresta } e \in E(G) \text{ está na solução;} \\ 0, & \text{caso contrário.} \end{cases}$$

Para garantir que a solução da formulação contenha exatamente $|V(G)| - 1$ arestas, usamos a seguinte restrição, onde o somatório das variáveis x é exatamente esta quantidade:

$$\sum_{e \in E(G)} x_e = |V(G)| - 1$$

Para assegurar que a solução é acíclica, definimos o seguinte conjunto de restrições:

$$\sum_{e = \{i,j\} \in E(G), i,j \in S} x_e \leq |S| - 1, \forall S \subset V(G)$$

Note que a restrição acima impõe que, para todo subconjunto S de vértices do grafo G , selecionamos no máximo $|S| - 1$ arestas de G incidentes em pares de vértices de S . Vamos mostrar que ela garante que a solução é acíclica. Suponha que não, então seja $C = \langle v_1, v_2, \dots, v_s, v_1 \rangle$ um ciclo da solução. Como as restrições são impostas para todos os subconjuntos de vértices, temos em particular aquela relativa a $S = \{v_1, v_2, \dots, v_s\}$. Como

$$\sum_{e = \{i,j\} \in E(G), i,j \in S} x_e = \sum_{e \in E(C)} x_e + \sum_{e = \{i,j\} \in E(G) \setminus E(C), i,j \in S} x_e,$$

tal restrição escreve-se como

$$\sum_{e \in E(C)} x_e + \sum_{e = \{i,j\} \in E(G) \setminus E(C), i,j \in S} x_e \leq |S| - 1.$$

Dado que $x_e \geq 0$, $e \in E(G)$, obtemos:

$$\sum_{e \in E(C)} x_e \leq |S| - 1.$$

Por outro lado, sendo C parte solução, todas as arestas de $E(C)$ estão presentes e, portanto, $\sum_{e \in E(C)} x_e = |S| > |S| - 1$: uma contradição. Logo, a solução é acíclica.

Como desejamos encontrar, dentre as árvores t -spanner, a de menor custo, temos a seguinte função objetivo, cujo sentido será de minimização:

$$\sum_{e \in E(G)} c_e x_e.$$

Dessa maneira, obtemos a seguinte (FE), adicionando à formulação SEC's as restrições (5.3):

$$\min \sum_{e \in E(G)} c_e x_e \quad (5.5)$$

$$\text{s.a: } \sum_{e \in E(G)} x_e = |V(G)| - 1 \quad (5.6)$$

$$\sum_{e=\{i,j\} \in E(G), i,j \in S} x_e \leq |S| - 1, \quad \forall S \subset V(G) \quad (5.7)$$

$$\sum_{e \in E(P)} x_e \leq |P| - 1, \quad \forall C \in CI \quad (5.8)$$

$$x_e \in \mathbb{B}, \quad e \in E(G) \quad (5.9)$$

Vale registrar que, na formulação SEC's para o AGM (5.5–5.7,5.9), o conjunto viável da relaxação linear já é a envoltória convexa dos pontos inteiros viáveis (EDMONDS, 1991). Ao adicionarmos as restrições (5.8), porém, esse conjunto possivelmente terá vértices fracionários.

5.1.2 Formulação MTZ

Para a formulação MTZ, vamos considerar o digrafo simétrico obtido de G , com os mesmos vértices e conjunto de arcos $A(G) = \{(i, j), (j, i) | \{i, j\} \in E(G)\}$. Da mesma forma que na formulação acima, vamos garantir que a solução possua $|V(G)| - 1$ arestas e seja acíclica.

Vamos definir as variáveis como:

$$\begin{aligned} - x_{ij} &= \begin{cases} 1, & \text{se } (i, j) \in A(G) \text{ está na solução;} \\ 0, & \text{caso contrário.} \end{cases} \\ - \pi_i &= \text{potencial do vértice } i \in V(G). \end{aligned}$$

Note que há duas variáveis, x_{ij} e x_{ji} , para cada aresta $e = \{i, j\}$ de G . Assim, a soma $x_{ij} + x_{ji}$ corresponde à variável x_e em (FE) .

Iremos construir uma arborescência de G , que representará a árvore geradora da solução. Para garantir que a árvore terá exatamente $|V(G)| - 1$ arestas, temos a seguinte restrição de cardinalidade:

$$\sum_{(i,j) \in A(G)} x_{ij} = |V(G)| - 1$$

Como a solução também será uma arborescência, adicionamos as seguintes restrições, que impõem a presença de no máximo um arco entrando em cada vértice:

$$\sum_{j \in N(i)} x_{ji} \leq 1, \quad \forall i \in V(G) \quad (5.10)$$

Para garantir que a solução será acíclica, vamos utilizar as variáveis π , que representarão o potencial de cada vértice. Cada potencial π_i irá representar a quantidade de arcos que estão no caminho da raiz r até o vértice i da arborescência. Tomando o potencial da raiz como 0, vamos determinar o potencial de cada outro vértice como sendo 1 unidade a mais que potencial do vértice antecedente a ele na arborescência. Em outras palavras, se o arco $(i, j) \in A(G)$ está na solução então $\pi_j = \pi_i + 1$. Para modelar esta implicação temos, o seguinte grupo de restrições propostas por Miller *et al.* (1960):

$$-(M+1)(1-x_{ij})+1 \leq \pi_j - \pi_i \leq 1 + (M-1)(1-x_{ij}), \quad \forall (i, j) \in A(G) \quad (5.11)$$

De fato, se o arco (i, j) está na solução. então a variável $x_{ij} = 1$, de modo que:

$$1 \leq \pi_j - \pi_i \leq 1 \quad \Leftrightarrow \quad \pi_j = \pi_i + 1$$

Caso contrário, ou seja, $x_{ij} = 0$, obtemos:

$$-M \leq \pi_j - \pi_i \leq M.$$

Nesse caso, a restrição deve ser uma redundância, quer dizer, ser satisfeita e não interferir no valor dos potenciais. Para garantir redundância, basta atribuir um valor constante M de modo que a diferença dos potenciais sempre estará no intervalo $[-M, M]$. Pela definição dos potenciais, temos $0 \leq \pi_i \leq |V(G)| - 1, \forall i \in V(G)$, pois o incremento de cada potencial em relação ao do vértice precedente é 1, levando no pior caso um potencial π a ter $|V(G)| - 1$ incrementos (quantidade de arcos da arborescência). Dessa forma, o menor valor que $\pi_i - \pi_j$ pode atingir é $-(|V(G)| - 1)$. Por outro lado, essa diferença é no máximo $|V(G)| - 1$. Dito isso, podemos assumir $M = |V(G)| - 1$.

Mostramos que as restrições (5.10)-(5.11) proíbem ciclos. Suponha, ao contrário, que a solução contenha ciclo C . Caso fosse não direcionado, C violaria $\sum_{j \in N(i)} x_{ji} \leq 1$ para algum $i \in V(C)$. Assuma que $C = \langle v_1, v_2, \dots, v_k, v_1 \rangle$ é um ciclo direcionado e seja $\pi_1 = s$, para algum natural s . Como as arestas do ciclo representam uma sequência de arcos direcionados no mesmo sentido, temos $\pi_k = s + k - 1$ por (5.11). Mas como o arco (v_k, v_1) também está na solução, essas restrições determinam $\pi_1 = \pi_k + 1 = k + s$, uma contradição. Logo, a solução é acíclica.

Desrochers e Laporte (1991) fortaleceram as desigualdades (5.11), fazendo *lifting* na variável x_{ij} . Observando que $x_{ij} + x_{ji} \in \{0, 1\}$, obtiveram

$$(M-1)x_{ji} - (M+1)(1-x_{ij}) + 1 \leq \pi_j - \pi_i \leq 1 + (M-1)(1-x_{ij}) - (M+1)x_{ji}, \forall (i, j) \in A(G)$$

Quando $x_{ji} = 0$, essas desigualdades são equivalentes a (5.11). Quando $x_{ji} = 1$ e, consequentemente, $x_{ij} = 0$, elas se tornam

$$-1 \leq \pi_j - \pi_i \leq -1 \quad \Leftrightarrow \quad \pi_i = \pi_j + 1,$$

ou seja, a desigualdade (5.11) relativa a (j, i) . Note, assim, que esse par de desigualdades, além de mais forte, torna-se o mesmo para (i, j) e (j, i) . Portanto, passamos a ter apenas duas desigualdades para cada aresta $\{i, j\} \in E(G)$, que podem ser reescritas como

$$(M-1)x_{ji} + (M+1)x_{ij} - M \leq \pi_j - \pi_i \leq M - (M-1)x_{ij} - (M+1)x_{ji}, \quad \forall \{i, j\} \in E(G) \quad (5.12)$$

Alternativamente, poderíamos escrever uma desigualdade para cada arco $(i, j) \in A(G)$.

Para expressar uma formulação (FE) em termos de variáveis associadas a $A(G)$ e não a $E(G)$, precisamos adaptar a restrição (5.3) e a função objetivo (5.1). Note que a aresta $e = \{i, j\} \in E(G)$ está na solução se, e somente se, $x_{ij} + x_{ji} = 1$. Portanto, (5.3) se expressa em termos dessas variáveis como:

$$\sum_{\{i,j\} \in E(P)} (x_{ij} + x_{ji}) \leq |P| - 1, \quad \forall P \in CI.$$

Similarmente, a função objetivo, cujo sentido é de minimização, reescreve-se como:

$$\sum_{(i,j) \in A(G)} w_{ij} x_{ij}$$

Dessa maneira, obtemos a seguinte (FE), adicionando à formulação MTZ as restrições (5.3):

$$\min \sum_{(i,j) \in A(G)} w_{ij} x_{ij} \quad (5.13)$$

$$\text{s.a: } \sum_{(i,j) \in A(G)} x_{ij} = |V(G)| - 1 \quad (5.14)$$

$$\sum_{j \in N(i)} x_{ji} \leq 1, \quad \forall i \in V(G) \quad (5.15)$$

$$\pi_j - \pi_i \leq M - (M-1)x_{ij} - (M+1)x_{ji}, \quad \forall (i, j) \in A(G) \quad (5.16)$$

$$0 \leq \pi_i \leq |V(G)| - 1, \quad \forall i \in V(G) \quad (5.17)$$

$$\sum_{\{i,j\} \in E(P)} (x_{ij} + x_{ji}) \leq |P| - 1, \quad \forall P \in CI \quad (5.18)$$

$$x_{ij} \in \mathbb{B}, \quad \forall (i, j) \in A(G) \quad (5.19)$$

Cabem alguns comentários sobre a formulação cima:

1. Diferentemente da formulação SEC's para o AGM, na formulação MTZ (5.13 – 5.17) a envoltória convexa dos pontos inteiros viáveis não é mais dada pelo poliedro obtido com a remoção das restrições de integralidade das variáveis x . Em outras palavras, ela fornece uma relaxação linear mais fraca.
2. Devido à restrição de cardinalidade (5.14), todas as desigualdades (5.15) serão satisfeitas na igualdade, a menos de uma, relacionada ao vértice que será raiz. Na literatura, usualmente, a raiz r é fixada a priori. Nesse caso, podemos colocar igualdade em (5.15), para todo $i \neq r$, e substituir a desigualdade relativa a r por $\sum_{j \in N^-(r)} x_{jr} = 0$, além de fixar $\pi_r = 0$. Em (5.16), podemos tomar $M = |V(G)| - 1$, se $i = r$, ou $M = |V(G)| - 2$, se $i \neq r$ (pois r não pode estar no caminho entre i e j).
3. Quando deixamos o modelo enraizar a árvore, podemos tomar $M = \lfloor |V(G)|/2 \rfloor$, porque é sempre possível escolher uma raiz tal que a distância máxima entre ela e algum outro vértice seja esse valor. De fato, seja r uma raiz que minimiza essa distância máxima. Suponha, por absurdo, que existe vértice $v \neq r$ tal que o (r, v) -caminho direcionado tenha comprimento maior que $\lfloor |V(G)|/2 \rfloor$. Seja r' o primeiro vértice após r nesse caminho. Mudando a raiz da árvore para r' , argumentamos que a maior distância decresce de uma unidade, levando ao absurdo. Na subárvore T' enraizada em r' há pelo menos $\lfloor |V(G)|/2 \rfloor + 1$ vértices. Descontando esses vértices, concluímos que há no máximo outros $\lfloor |V(G)|/2 \rfloor - 1$ vértices. Portanto, a distância de r' para qualquer desses outros vértices é no máximo $\lfloor |V(G)|/2 \rfloor$. Por outro lado, a distância de r' para qualquer vértice em T' diminui de uma unidade (em relação àquela a partir de r), incluindo a distância até v (mas que ainda é pelo menos $\lfloor |V(G)|/2 \rfloor$). Logo, tomando r' como raiz, a distância máxima continua aquela relativa à v , porém uma unidade menor.

5.2 Formulações Compactas

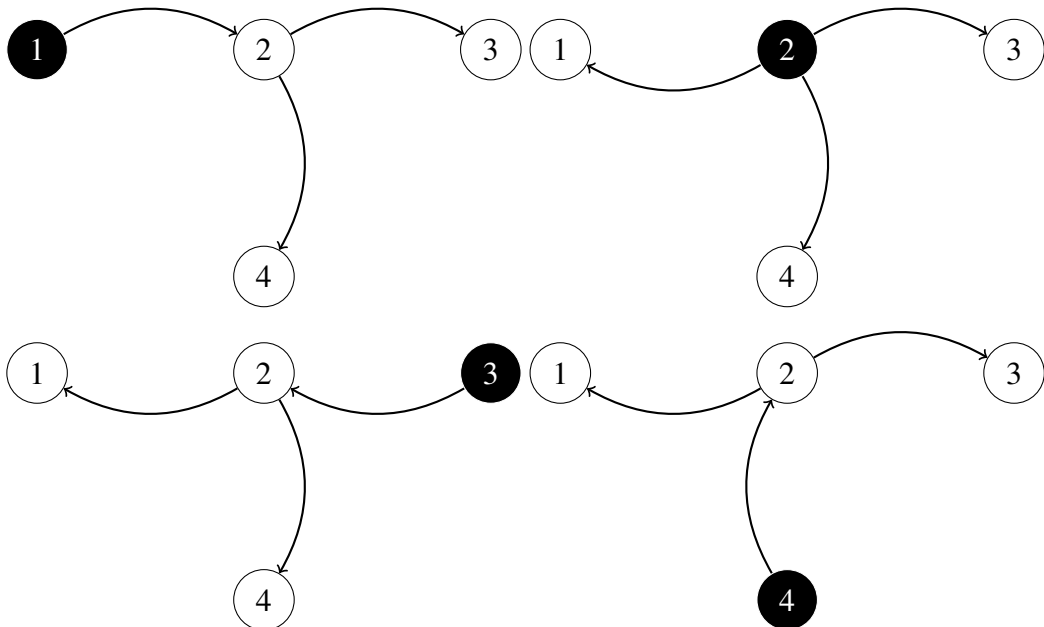
Nesta seção, apresentaremos três formulações de tamanho polinomial para o problema. As duas primeiras foram propostas para o AtSCM por Braga (2019), aqui denominadas de sem rótulos (SR) e com rótulos (CR). A terceira formulação está sendo apresentada em nosso trabalho. Vamos utilizar os seguintes parâmetros, referentes ao grafo de entrada G :

- $V(G)$: conjunto de vértices de G .
- $E(G)$: conjunto de arestas de G .

- $A(G) = \{(i, j), (j, i) | \{i, j\} \in E(G)\}$: conjunto de arcos da orientação simétrica de G , onde cada aresta $\{i, j\}$ dá origem aos arcos $i \leftarrow j$ e $j \rightarrow i$.

Considerando a orientação simétrica de G , cada formulação constrói $|V(G)|$ arborescências que se sobrepõem em G , cada uma enraizada em um nó diferente do grafo. O termo sobrepor significa que existe uma árvore geradora de G de quem cada arborescência é uma orientação diferente, ou seja, as arestas de G associadas aos arcos de cada arborescência serão as mesmas. Assim, todas essas arborescências representarão uma mesma árvore geradora de G . Na Figura 19 ilustramos 4 arborescências diferentes, cada uma enraizada em um nó do grafo (destacado em preto), porém todas elas associadas a mesma árvore geradora.

Figura 19 – Arborescências sobrepostas



Fonte: elaborado pelo autor.

Para a representação dessas arborescências, definimos as seguintes variáveis:

- $x_e = \begin{cases} 1, & \text{se } e \in E(G) \text{ está na solução;} \\ 0, & \text{caso contrário.} \end{cases}$
- $\lambda_{ij}^v = \begin{cases} 1, & \text{se } (i, j) \in A(G) \text{ está na arborescência enraizada em } v \in V(G); \\ 0, & \text{caso contrário.} \end{cases}$

De modo semelhante às duas formulações apresentadas na seção anterior, vamos construir uma solução com $|V(G)| - 1$ arestas e acíclica. Para garantir esta quantidade de arestas

temos:

$$\sum_{e \in E(G)} x_e = |V(G)| - 1$$

Para assegurar que a solução represente as $|V(G)|$ arborescências e, além disso, as arestas que elas utilizam sejam as mesmas, temos:

$$\sum_{i \in N^-(v)} \lambda_{iv}^v = 0, \quad \forall v \in V(G)$$

$$\sum_{i \in N^-(j)} \lambda_{ij}^v = 1, \quad \forall v \in V(G), \forall j \in V(G) \setminus \{v\}$$

$$x_e = \lambda_{ij}^v + \lambda_{ji}^v, \quad \forall v \in V(G), \forall e = \{i, j\} \in E(G)$$

Com essas equações, garantimos que, para a arborescência enraizada no vértice $v \in V(G)$, não há vértice na vizinhança de entrada de v , e, para cada vértice $j \in V(G) \setminus \{v\}$, há exatamente um vértice na vizinhança de entrada de j . A terceira restrição impõe que uma aresta é utilizada para compor a solução se, e somente se, toda arborescência utiliza uma das orientações dela, garantindo que as arborescências terão como base as mesmas arestas.

Podemos mostrar que as arestas escolhidas por esse sistema de igualdades induzem um subgrafo acíclico. Suponha, ao contrário, que a solução contenha um ciclo C . Seja $G_r = (V(G), A_r)$ o subdigrafo induzido pelos arcos $A_r = \{(i, j) \in A(G) : \lambda_{ij}^r = 1\}$, para todo $r \in V(G)$. Uma orientação do ciclo C aparece em cada G_r . Caso C seja não direcionado em G_r , para algum $r \in V(G)$, então existe um vértice $v \in V(C)$ que possui pelo menos dois vértices em sua vizinhança de entrada, ou seja, $|N_{G_r}^-(v)| \geq 2$; mas $\sum_{i \in N^-(v)} \lambda_{iv}^r \leq 1$, uma contradição. Admita então que C é um ciclo direcionado em todo G_r , em particular para $r \in V(C)$. Então $|N_{G_r}^-(r)| \geq 1$, uma contradição pois $\sum_{i \in N^-(r)} \lambda_{ir}^r = 0$.

Este sistema de igualdades lineares foi proposto por Martin (1991) para modelar o AGM. Ele mostrou que, tal qual o sistema linear da formulação SEC's para o AGM, esse sistema tem a propriedade de que a envoltória convexa dos pontos viáveis é a mesma quando removemos as restrições que impõem integralidade das variáveis x e λ . Ver também Conforti *et al.* (2010) e Conforti *et al.* (2013).

Braga (2019) usou essa formulação de Martin (1991) como base para obter duas formulações para o AtSCM. De modo a garantir que a árvore geradora seja também t -spanner, ou seja, que satisfaça a condição (3.1), usou variáveis adicionais para expressar as distâncias entre vértices. Nas próximas três subseções, apresentaremos essas duas formulações para o AtSCM e

mais uma terceira. Cada formulação irá introduzir variáveis e restrições diferentes no sistema apresentado acima, objetivando capturar o custo do caminho entre quaisquer dois vértices i e j na solução e limitá-lo a $t * \text{dist}_G(i, j)$.

Em todas elas, como desejamos encontrar, dentre as árvores t -spanner, a de menor custo, temos a seguinte função objetivo, cujo sentido será de minimização:

$$\sum_{e \in E(G)} w_e x_e$$

5.2.1 Formulação SR

Nessa formulação, introduzida por Braga (2019), vamos utilizar o seguinte conjunto de variáveis para identificar explicitamente a presença de uma aresta $\{u, v\} \in E(G)$ no caminho entre dois vértices i e j , na solução do sistema:

$$- y_{ij}^{uv} = \begin{cases} 1, & \text{se } \{i, j\} \in E(G) \text{ está no caminho entre } u \text{ e } v; \\ 0, & \text{caso contrário.} \end{cases}$$

As variáveis y podem ser expressas em função das variáveis λ como segue. Sejam T^u e T^v duas arborescências definidas pelas variáveis λ , enraizadas, respectivamente, nos vértices u e v , distintos. Ambas são uma orientação de uma mesma árvore geradora T . Em T^u (resp. T^v), temos exatamente um caminho (direcionado) de u a v (resp. de v a u). Toda aresta $\{i, j\}$ de T que não está no caminho entre u e v possui a mesma orientação tanto em T^u quanto em T^v , ou seja $\lambda_{ij}^u = \lambda_{ij}^v$ e $\lambda_{ji}^u = \lambda_{ji}^v$. Já uma aresta $\{i, j\}$ do caminho entre u e v terá em T^u orientação oposta àquela em T^v ; em outras palavras, $\lambda_{ij}^u = \lambda_{ji}^v$ e $\lambda_{ji}^u = \lambda_{ij}^v$. Podemos observar esta propriedade nas arborescências da Figura 19. Desse modo, podemos obter a relação entre as variáveis λ e y na Proposição 5.2.1.

Proposição 5.2.1 Para todos $\{u, v\}, \{i, j\} \in E(G)$,

$$y_{ij}^{uv} = \max\{\lambda_{ij}^u - \lambda_{ij}^v, \lambda_{ji}^u - \lambda_{ji}^v\} = \min\{\lambda_{ij}^u + \lambda_{ij}^v, \lambda_{ji}^u + \lambda_{ji}^v\}.$$

Prova. Sejam $L = \max\{\lambda_{ij}^u - \lambda_{ij}^v, \lambda_{ji}^u - \lambda_{ji}^v\}$ e $U = \min\{\lambda_{ij}^u + \lambda_{ij}^v, \lambda_{ji}^u + \lambda_{ji}^v\}$. Se $\{i, j\}$ não está na solução, então $\lambda_{ij}^u = \lambda_{ij}^v = \lambda_{ji}^u = \lambda_{ji}^v = 0$; logo $L = \max\{0, 0\} = 0$ e $U = \min\{0, 0\}$, ou seja, $L = U = y_{ij}^{uv}$. Caso contrário, temos $x_{ij} = 1$. Se $\{i, j\}$ está no caminho entre u e v , então $\lambda_{ij}^u = \lambda_{ji}^v = 1$ e $\lambda_{ji}^u = \lambda_{ij}^v = 0$, ou $\lambda_{ij}^u = \lambda_{ij}^v = 1$ e $\lambda_{ji}^u = \lambda_{ji}^v = 0$; nos dois casos, $L = \max\{-1, 1\} = 1$ e $U = \min\{1, 1\} = 1$, implicando em $L = U = y_{ij}^{uv}$. Se $\{i, j\}$ não está no caminho entre u e v , então

$\lambda_{ij}^u = \lambda_{ij}^v = 1$ e $\lambda_{ji}^u = \lambda_{ji}^v = 0$, ou $\lambda_{ji}^u = \lambda_{ji}^v = 1$ e $\lambda_{ij}^u = \lambda_{ij}^v = 0$; nos dos casos, $L = \max\{0, 0\} = 0$ e $U = \min\{2, 0\} = 0$, levando a $L = U = y_{ij}^{uv}$. ■

Pela Proposição 5.2.1, concluímos que a valoração de y_{ij}^{uv} é precisamente determinada pelas desigualdades $\max\{\lambda_{ij}^u - \lambda_{ij}^v, \lambda_{ji}^u - \lambda_{ji}^v\} \leq y_{ij}^{uv} \leq \min\{\lambda_{ij}^u + \lambda_{ij}^v, \lambda_{ji}^u + \lambda_{ji}^v\}$, ou melhor, pelas desigualdades lineares

$$y_{ij}^{uv} \geq \lambda_{ij}^u - \lambda_{ij}^v$$

$$y_{ij}^{uv} \geq \lambda_{ji}^u - \lambda_{ji}^v$$

e

$$y_{ij}^{uv} \leq \lambda_{ij}^u + \lambda_{ij}^v \quad (5.20)$$

$$y_{ij}^{uv} \leq \lambda_{ji}^u + \lambda_{ji}^v \quad (5.21)$$

Para modelar a condição (3.1), seja $\sum_{\{i,j\} \in E(G)} w_{ij} y_{ij}^{uv}$ a expressão que calcula explicitamente o custo do caminho entre u e v na solução. Vamos limitá-la a $t * \text{dist}_G(u, v)$, obtendo:

$$\sum_{\{i,j\} \in E(G)} w_{ij} y_{ij}^{uv} \leq t * \text{dist}_G(u, v), \quad \forall \{u, v\} \in E(G)$$

De maneira geral a formulação (SR) pode ser expressa como:

$$(SR) \min \sum_{\{i,j\} \in E(G)} w_{ij} x_{ij} \quad (5.22)$$

$$\text{s.a:} \quad \sum_{\{i,j\} \in E(G)} x_{ij} = |V(G)| - 1 \quad (5.23)$$

$$\sum_{i \in N^-(j)} \lambda_{ij}^v = 1, \quad \forall v \in V(G), \forall j \in V(G) \setminus \{v\} \quad (5.24)$$

$$\sum_{i \in N^-(v)} \lambda_{iv}^v = 0, \quad \forall v \in V(G) \quad (5.25)$$

$$x_{ij} = \lambda_{ij}^v + \lambda_{ji}^v, \quad \forall v \in V(G), \forall \{i, j\} \in E(G) \quad (5.26)$$

$$y_{ij}^{uv} \geq \lambda_{ij}^u - \lambda_{ij}^v, \quad \forall \{u, v\} \in E(G), \forall \{i, j\} \in E(G) \quad (5.27)$$

$$y_{ij}^{uv} \geq \lambda_{ji}^u - \lambda_{ji}^v, \quad \forall \{u, v\} \in E(G), \forall \{i, j\} \in E(G) \quad (5.28)$$

$$\sum_{\{i,j\} \in E(G)} w_{ij} y_{ij}^{uv} \leq t * \text{dist}_G(u, v), \quad \forall \{u, v\} \in E(G) \quad (5.29)$$

$$x_{ij} \in \mathbb{B}, \quad \forall \{i, j\} \in E(G) \quad (5.30)$$

$$\lambda_{ij}^v, \lambda_{ji}^v \in \mathbb{B}, \quad \forall v \in V(G), \forall \{i, j\} \in E(G) \quad (5.31)$$

$$y_{ij}^{uv} \in \mathbb{B}, \quad \forall \{i, j\} \in E(G), \forall \{u, v\} \in E(G) \quad (5.32)$$

Algumas observações podem ser feitas com relação à formulação (SR):

1. As variáveis y_{ij}^{uv} estão definidas apenas para $\{i, j\}, \{u, v\} \in E(G)$, entretanto, a priori, também deveríamos defini-las para pares $\{u, v\} \notin E(G)$. O Teorema 3.2.1 estabelece que é suficiente garantir a distância na solução apenas para pares que formam aresta no grafo, portanto definimos as variáveis em consonância com esse fato, implicando uma quantidade menor de $|E|^2$ variáveis, ao contrário de $|E||V|^2$ variáveis.
2. Devido à função objetivo, não é necessário adicionar à formulação as restrições (5.20) e (5.21). Como queremos minimizar o custo da solução, só nos interessa ativar uma variável y_{ij}^{uv} quando obrigatório, de modo que as desigualdades (5.27) e (5.28) são suficientes para garantir o valor adequado de y_{ij}^{uv} no ótimo. Mais precisamente, para toda solução (x, λ, y) viável para a relaxação linear da formulação SR, existe solução viável (x, λ, y') , de mesmo valor, onde $(y')_{ij}^{uv} = \max\{\lambda_{ij}^u - \lambda_{ij}^v, \lambda_{ji}^u - \lambda_{ji}^v\}$. De fato, essa solução trivialmente satisfaz (5.27)-(5.28), além de satisfazer (5.29) pois $y' \leq y$ e $w \geq 0$. Além disso, ela satisfaz as desigualdades excluídas (5.20)–(5.21), devido à Proposição 5.2.1.
3. Pelo Teorema 3.2.1, para cada $\{u, v\} \in E$, é possível substituir a restrição (5.29) por $\sum_{\{i,j\} \in E(G)} w_{ij} y_{ij}^{uv} \leq t * w_{uv}$, que é a restrição usada no trabalho de Braga (2019). Isso dispensaria o cálculo de $\text{dist}_G(u, v)$. Entretanto, essa desigualdade é claramente dominada por aquela do modelo, visto que $\text{dist}_G(u, v) \leq w_{uv}$. Usando essa restrição mais forte, obtemos uma relaxação linear “mais apertada”.
4. Por (5.26), temos que $\lambda_{ij}^u + \lambda_{ji}^u = \lambda_{ij}^v + \lambda_{ji}^v = x_{ij}$. Então, $\lambda_{ji}^u - \lambda_{ji}^v = -(\lambda_{ij}^u - \lambda_{ij}^v)$, de modo que podemos reescrever (5.27)-(5.28) como

$$-y_{ij}^{uv} \leq \lambda_{ij}^u - \lambda_{ij}^v \leq y_{ij}^{uv}. \quad (5.33)$$

Além disso, devido a $\lambda_{ij}^u + \lambda_{ij}^v = (\lambda_{ij}^u - \lambda_{ij}^v) + x_{ij}$ e $\lambda_{ji}^u + \lambda_{ji}^v = -(\lambda_{ij}^u - \lambda_{ij}^v) + x_{ij}$, (5.20) e (5.21) equivalem a

$$y_{ij}^{uv} - x_{ij} \leq \lambda_{ij}^u - \lambda_{ji}^v \leq x_{ij} - y_{ij}^{uv}. \quad (5.34)$$

Essas novas expressões fornecem visões alternativas das anteriores.

5. Quando uma aresta $\{u, v\}$ é usada (isto é, $x_{uv} = 1$), apenas essa aresta aparece no caminho entre u e v . Isso nos leva às seguintes (des)igualdades válidas:

$$y_{uv}^{uv} = x_{uv} \quad (5.35)$$

$$\sum_{j \in N(i) \setminus \{v\}} y_{ij}^{uv} + x_{uv} = 1, \quad \forall \{u, v\} \in E, i \in \{u, v\} \quad (5.36)$$

$$y_{ij}^{uv} + x_{uv} \leq 1, \quad \forall \{i, j\}, \{u, v\} \in E(G), \{i, j\} \cap \{u, v\} = \emptyset \quad (5.37)$$

As igualdades (5.35) servem para reduzir a quantidade de variáveis y inicialmente definidas. Mais importante, (5.36)–(5.37) são úteis para fixar variáveis durante um processo de ramificação. Para reduzir a quantidade das desigualdades (5.37) poderíamos agregá-las na restrição mais fraca $\sum_{\{i,j\} \neq \{u,v\}} y_{ij}^{uv} + Mx_{uv} \leq M$, usando um big-M para a maior quantidade de arestas entre u e v numa solução. Por exemplo, podemos tomar $M = |V(G)| - 1$.

6. Caso a aresta $\{i, j\}$ apareça no caminho entre u e v (isto é, $y_{ij}^{uv} = 1$) e a aresta $e = \{u, i\}$ seja usada (isto é, $x_{ui} = 1$), então e deve figurar no caminho entre u e v . O mesmo vale para qualquer e tal que $|e \cap \{u, v\}| = |e \cap \{i, j\}| = 1$. Assim, obtemos as seguintes desigualdades válidas:

$$y_e^{uv} \geq y_{ij}^{uv} + x_e - 1 \quad \forall \{i, j\}, \{u, v\}, e \in E(G), |e \cap \{u, v\}| = |e \cap \{i, j\}| = 1 \quad (5.38)$$

Na verdade, essas desigualdades são inócuas quando $\{i, j\} = \{u, v\}$ e devem ser consideradas apenas em caso contrário.

7. A integralidade das variáveis x pode ser relaxada, visto que ela é decorrência daquela das variáveis λ . Entretanto, a ramificação numa variável x_{ij} tem grande influência na fixação de outras variáveis. Por um lado, $x_{ij} = 0$ leva a $\lambda_{ij}^v = \lambda_{ji}^v = 0$, para todo $v \in V(G)$, e, portanto, $y_{ij}^{uv} = 0$ para todo $\{u, v\} \in E(G)$. Nesse caso, eliminamos $|E(G)|$ variáveis y e $2|V(G)|$ variáveis λ associadas a $\{i, j\}$. Por outro lado, considere agora $x_{uv} = 1$. Devido a (5.36)–(5.37) e (5.33), temos $y_{ij}^{uv} = 0$ e, por conseguinte, $\lambda_{ij}^u = \lambda_{ij}^v$ e $\lambda_{ji}^u = \lambda_{ji}^v$, para todo $\{i, j\} \in E(G) \setminus \{\{u, v\}\}$. Similarmente, devido a (5.35) e (5.34), obtemos $y_{uv}^{uv} = 1$ e, com isso, $\lambda_{uv}^u = \lambda_{vu}^v$ e $\lambda_{vu}^u = \lambda_{uv}^v$. Novamente, eliminamos $|E(G)|$ variáveis y e associadas a $\{u, v\}$ e $2|V(G)|$ variáveis λ com superíndice u ou v .

5.2.2 Formulação CR

Em lugar de marcar explicitamente as arestas do caminho entre cada par de vértices, a segunda formulação apresentada por Braga (2019), a ser denominada (CR), rotula cada vértice com os pesos dos caminhos mínimos de cada raiz até ele. O cálculo desses rótulos pode ser feito com uma adaptação das variáveis de potencial π da formulação MTZ. Mais precisamente definimos as seguintes variáveis:

- π_i^j = potencial do vértice i na arborescência enraizada em j , com $i, j \in V(G)$, ou seja, o peso do i, j caminho nessa arborescência.

Seja r a raiz de uma arborescência da solução. Se construirmos os potenciais

da mesma maneira que a formulação MTZ temos $\pi_j^r = \pi_i^r + 1$, quando $\lambda_{ij}^r = 1$, $\forall r \in V(G)$, $\forall (i, j) \in A(G)$, ou seja, o valor do potencial π_j^r corresponde a quantidade de arestas no caminho entre r e j . Como estamos interessados em captar o custo do caminho, trocaremos a condição acima para: $\pi_j^r = \pi_i^r + w_{ij}$, se $\lambda_{ij}^r = 1$, $\forall r \in V(G)$, $(i, j) \in A(G)$. Desse modo, o valor de cada potencial $\pi_i^j = \pi_j^i$ representará o peso do caminho entre os vértices i e j na solução.

Como não há caminho de um vértice $r \in V(G)$ a ele mesmo, então $\pi_r^r = 0$. Para garantir a construção dos demais potenciais, tome as seguintes desigualdades, similares a (5.12):

$$(M_{ij}^r - w_{ij})\lambda_{ji}^r + (M_{ij}^r + w_{ij})\lambda_{ij}^r - M_{ij}^r \leq \pi_j^r - \pi_i^r \leq M_{ji}^r - (M_{ji}^r - w_{ij})\lambda_{ij}^r - (M_{ji}^r + w_{ij})\lambda_{ji}^r, \\ \forall r \in V(G), \{i, j\} \in E(G),$$

nas quais M_{ij}^r e M_{ji}^r são constantes. Como anteriormente, podemos alternativamente escrever

$$\pi_i^r - \pi_j^r \leq M_{ij}^r - (M_{ij}^r + w_{ij})\lambda_{ij}^r - (M_{ij}^r - w_{ij})\lambda_{ji}^r, \quad \forall r \in V(G), (i, j) \in A(G), \quad (5.39)$$

onde usamos $A(G)$ em lugar de $E(G)$.

Para verificar a validade dessas desigualdades, tome $r \in V(G)$ e $\{i, j\} \in E(G)$. Se $\lambda_{ij}^r = 1$ (e assim $\lambda_{ji}^r = 0$), concluímos pelas primeiras expressões que $\pi_j^r = \pi_i^r + w_{ij}$. Similarmente, se $\lambda_{ji}^r = 1$ (e assim $\lambda_{ij}^r = 0$), temos que $\pi_i^r = \pi_j^r - w_{ij}$. Finalmente, quando $\lambda_{ij}^r = \lambda_{ji}^r = 0$, as desigualdades tornam-se $-M_{ij}^r \leq \pi_j^r - \pi_i^r \leq M_{ji}^r$. Portanto, para que sejam válidas, M_{ij}^r deve ser um limite superior para $\pi_i^r - \pi_j^r$, ou seja, para a diferença entre as distâncias de r a i e de r a j , para todos $\{i, j\} \in E$ e $r \in V(G)$.

Sejam $i, j, r \in V(G)$. Detalhamos a seguir o valor adotado para M_{ij}^r . Seja w_M o peso de uma árvore geradora máxima de G . Defina

$$U_{ij} = \min\{t * \text{dist}_G(i, j), w_M\}.$$

Lembrando que π_i^j é o tamanho do caminho de j até i na arborescência enraizada em j , concluímos que $\pi_i^j \leq t * \text{dist}_G(i, j)$, pela condição (3.1), e que $\pi_i^j \leq w_M$, pois π_i^j é peso de um caminho em G . Logo, $\pi_i^j \leq U_{ij}$. Por outro lado, trivialmente $\pi_i^j \geq \text{dist}_G(i, j)$. Dessa forma, obtemos

$$\pi_i^r - \pi_j^r \leq U_{ir} - \text{dist}_G(r, j).$$

Seja ℓ_{ij}^r o tamanho do caminho entre i e j na árvore enraizada em r . Note que $\pi_i^r - \pi_j^r \leq |\pi_i^r - \pi_j^r| \leq \ell_{ij}^r \leq U_{ij}$. Pelo exposto, podemos tomar

$$M_{ij}^r = \min\{U_{ij}, U_{ir} - \text{dist}_G(r, j)\}.$$

Para garantir a condição (3.1) basta limitar superiormente cada potencial π_i^j , $i, j \in V(G)$, a $t * \text{dist}_G(i, j)$, ou melhor a $U_{ij} = \min\{t * \text{dist}_G(i, j), w_M\}$. Por outro lado, π_i^j é limitado inferiormente a $\text{dist}_G(i, j)$. Além disso, como a árvore (grafo não orientado) subjacente a cada arborescência é a mesma, em particular, as enraizadas em i e j , então $\pi_i^j = \pi_j^i$, pois o caminho entre i e j é único (Teorema 2.2.1). Dessa forma temos as seguintes desigualdades:

$$\text{dist}_G(i, j) \leq \pi_i^j = \pi_j^i \leq \min\{t * \text{dist}_G(i, j), w_M\}, \quad \forall i, j \in V(G)$$

De maneira geral a formulação (CR) pode ser expressa como:

$$\text{(CR) min } \sum_{\{i, j\} \in E(G)} w_{ij} x_{ij} \quad (5.40)$$

$$\text{s.a: } \sum_{\{i, j\} \in E(G)} x_{ij} = |V(G)| - 1 \quad (5.41)$$

$$\sum_{i \in N^-(j)} \lambda_{ij}^v = 1, \quad \forall v \in V(G), \forall j \in V(G) \setminus \{v\} \quad (5.42)$$

$$\sum_{i \in N^-(v)} \lambda_{iv}^v = 0, \quad \forall v \in V(G) \quad (5.43)$$

$$x_{ij} = \lambda_{ij}^v + \lambda_{ji}^v, \quad \forall v \in V(G), \forall \{i, j\} \in E(G) \quad (5.44)$$

$$\pi_i^r - \pi_j^r + (M_{ij}^r + w_{ij}) \lambda_{ij}^r + (M_{ij}^r - w_{ij}) \lambda_{ji}^r \leq M_{ij}^r, \quad \forall r \in V(G), (i, j) \in A(G) \quad (5.45)$$

$$\pi_r^r = 0, \quad \forall r \in V(G) \quad (5.46)$$

$$\text{dist}_G(i, j) \leq \pi_i^j = \pi_j^i \leq \min\{t * \text{dist}_G(i, j), w_M\}, \quad \forall i, j \in V(G) \quad (5.47)$$

$$x_{ij} \in \mathbb{B}n \quad \forall \{i, j\} \in E(G) \quad (5.48)$$

$$\lambda_{ij}^v, \lambda_{ji}^v \in \mathbb{B}, \quad \forall v \in V(G), \forall \{i, j\} \in E(G) \quad (5.49)$$

$$\pi_i^j \in \mathbb{R}^+, \quad \forall i, j \in V(G) \quad (5.50)$$

Algumas observações podem ser feitas com relação à formulação (CR):

1. Em (5.47), pelo Teorema 3.2.1, é suficiente adicionar apenas as desigualdades relativas a pares $\{i, j\}$ que representam arestas em $E(G)$ para corretude da formulação. Entretanto, optamos mantê-las para todo par pois elas não aumentam o tamanho do modelo, já que podem ser tratadas como *bounds* das variáveis π .
2. Também com o Teorema 3.2.1, vimos que basta limitar o peso de cada i, j -caminho na solução a t vezes o peso da aresta $\{i, j\}$, tornando $t * w_{ij}$ o lado direito de (5.47), que é a limitação imposta na formulação (CR) apresentada no trabalho de Braga (2019). Aqui, vamos impor o limite superior de (5.47), pois, em termos de relaxação linear, torna a formulação mais “apertada”.

3. Também em (5.47), as igualdades $\pi_i^j = \pi_j^i$, não são necessárias. Devido às arborescências enraizadas em i e j representarem a mesma árvore geradora, a igualdade $\pi_i^j = \pi_j^i$ é garantida pela integralidade das variáveis λ , entretanto optamos por utilizá-las pois, assim, podemos reduzir a quantidade de variáveis da formulação. Além disso, Braga (2019) utiliza estas igualdades apenas para pares $\{i, j\} \in E$, o que permite uma redução bem menor de variáveis da formulação.

5.2.3 Formulação Aresta-Potencial (AP)

A segunda formulação proposta neste trabalho é denominada de Aresta-Potencial (AP). Ela se baseia numa alteração dos potenciais da formulação (CR), de modo que estes serão aplicados aos arcos da orientação simétrica em vez dos vértices. Para isso defina:

- $\tau_{ij}^r =$ potencial do arco $(i, j) \in A(G)$ na árvore enraizada em $r \in V(G)$.

Sejam r a raiz de uma arborescência da solução e $(i, j) \in A(G)$ um arco da orientação simétrica de G . Vamos construir os potenciais de modo que:

$$- \tau_{ij}^r = \begin{cases} \sum_{v \in N^-(i)} \tau_{vi}^r + w_{ij}, & \text{se } \lambda_{ij}^r = 1; \\ 0, & \text{caso contrário.} \end{cases}$$

Desse modo, exatamente um dos potenciais τ_{vi}^r , com $v \in N^-(i)$, irá receber o valor do caminho entre os vértices r e i na solução, enquanto os demais receberão valor 0. Ou seja, a expressão $\sum_{v \in N^-(i)} \tau_{vi}^r$ corresponde ao peso do caminho entre r e i na arborescência enraizada em r .

Para a construção das variáveis τ , usamos as seguintes desigualdades:

$$\tau_{ij}^r \geq \sum_{v \in N^-(i)} \tau_{vi}^r + w_{ij} - M_{ij}^r(1 - \lambda_{ij}^r), \quad \forall r \in V(G) \forall (i, j) \in A(G) \quad (5.51)$$

$$\tau_{ij}^r \leq \sum_{v \in N^-(i)} \tau_{vi}^r + w_{ij}, \quad \forall r \in V(G) \forall (i, j) \in A(G) \quad (5.52)$$

$$\tau_{ij}^r \leq M_{ij}^r \lambda_{ij}^r, \quad \forall r \in V(G) \forall (i, j) \in A(G) \quad (5.53)$$

onde M_{ij}^r é uma constante suficientemente grande, conforme detalhado a seguir.

Se $\lambda_{ij}^r = 1$ então, de (5.51)–(5.52), obtemos $\tau_{ij}^r = \sum_{v \in N^-(i)} \tau_{vi}^r + w_{ij}$, e, de (5.53), temos $\tau_{ij}^r \leq M_{ij}^r$. Suponha agora $\lambda_{ij}^r = 0$. De (5.53), obtemos $\tau_{ij}^r \leq 0$ e, portanto, $\tau_{ij}^r = 0$ e (5.52) fica redundante. Já por (5.51), obtemos $\tau_{ij}^r \geq \sum_{v \in N^-(i)} \tau_{vi}^r + w_{ij} - M_{ij}^r$. Resta, então, atribuir um valor para cada M_{ij}^r de modo que as restrições (5.51) e (5.53) fiquem redundantes quando, respectivamente, $\lambda_{ij}^r = 0$ e $\lambda_{ij}^r = 1$. Nos dois casos, precisamos ter $M_{ij}^r \geq \sum_{v \in N^-(i)} \tau_{vi}^r + w_{ij}$.

Como visto anteriormente, U_{ri} é um limite superior para o tamanho do caminho entre r e i em uma árvore t -spanner, ou seja, $\sum_{v \in N^-(i)} \tau_{vi}^r \leq U_{ri}$. Logo, podemos tomar $M_{ij}^r = U_{ri} + w_{ij}$.

Para garantir a condição (3.1), basta limitar a distância entre os vértices i e j a $t * \text{dist}_G(i, j)$. Além disso, essa distância não pode ultrapassar w_M . Assim, vamos limitá-la a U_{ij} . Mais ainda, como a distância entre i e j é a mesma nas duas arborescências (uma enraizada em i e a outra em j) então $\sum_{v \in N^-(j)} \tau_{vj}^i = \sum_{v \in N^-(i)} \tau_{vi}^j$. Dessa forma temos as seguintes desigualdades:

$$\sum_{v \in N^-(j)} \tau_{vj}^i = \sum_{v \in N^-(i)} \tau_{vi}^j \leq \min\{t * \text{dist}_G(i, j), w_M\}, \quad \forall i, j \in V(G)$$

De maneira geral, a formulação (AP) pode ser expressa como:

$$(AP) \min \sum_{\{i,j\} \in E(G)} w_{ij} x_{ij} \quad (5.54)$$

$$\text{s.a:} \quad \sum_{\{i,j\} \in E(G)} x_{ij} = |V(G)| - 1 \quad (5.55)$$

$$\sum_{i \in N^-(j)} \lambda_{ij}^v = 1, \quad \forall v \in V(G), \forall j \in V(G) \setminus \{v\} \quad (5.56)$$

$$\sum_{i \in N^-(v)} \lambda_{iv}^v = 0, \quad \forall v \in V(G) \quad (5.57)$$

$$x_{ij} = \lambda_{ij}^v + \lambda_{ji}^v, \quad \forall v \in V(G), \forall \{i, j\} \in E(G) \quad (5.58)$$

$$\tau_{ij}^r \geq \sum_{v \in N^-(i)} \tau_{vi}^r + w_{ij} - M_{ij}^r (1 - \lambda_{ij}^r), \quad \forall r \in V(G), \forall (i, j) \in A(G) \quad (5.59)$$

$$\sum_{v \in N^-(j)} \tau_{vj}^i = \sum_{v \in N^-(i)} \tau_{vi}^j \leq \min\{t * \text{dist}_G(i, j), w_M\}, \quad \forall i, j \in V(G) \quad (5.60)$$

$$x_{ij} \in \mathbb{B}, \quad \forall \{i, j\} \in E(G) \quad (5.61)$$

$$\lambda_{ij}^r, \lambda_{ji}^r \in \mathbb{B}, \quad \forall r \in V(G), \forall \{i, j\} \in E(G) \quad (5.62)$$

$$\tau_{ij}^r \in \mathbb{R}^+, \quad \forall r \in V(G), \forall (i, j) \in A(G) \quad (5.63)$$

Algumas observações podem ser feitas com relação à formulação (AP):

1. Devido à função objetivo, não é necessário adicionar à formulação as restrições (5.52) e (5.53). Como queremos minimizar o custo da solução, só nos interessa acumular um potencial τ_{ij}^r quando obrigatório, de modo que as desigualdades (5.51) são suficientes para garantir o valor adequado de τ_{ij}^r no ótimo. Mais precisamente, para toda solução (x, λ, τ) viável para a formulação (AP), existe solução viável (x, λ, τ') , de mesmo valor, onde $(\tau')_{ij}^r = \min\{\sum_{v \in N^-(i)} (\tau')_{vi}^r + w_{ij}, M_{ij}^r \lambda_{ij}^r\}$. De fato, esta solução trivialmente satisfaz (5.52)-(5.53). Como $\tau_{ij}^r \geq \min\{\sum_{v \in N^-(i)} (\tau')_{vi}^r + w_{ij}, M_{ij}^r \lambda_{ij}^r\} = (\tau')_{ij}^r$, então $\sum_{v \in N^-(j)} (\tau')_{vj}^r \leq$

$\sum_{v \in N^-(j)} \tau_{vj}^r \leq \min\{t * \text{dist}_G(i, j), w_M\}$, ou seja, (5.60) também é satisfeita. Portanto, (x, λ, τ') é viável para (AP) e com mesmo valor de função objetivo de (x, λ, τ) .

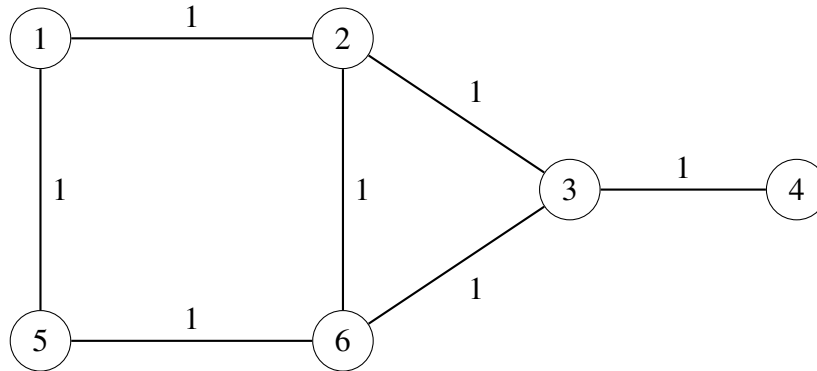
2. Pelo Teorema 3.2.1, vimos que basta limitar o peso de cada i, j -caminho na solução a t vezes o peso da aresta $\{i, j\}$, tornando $t * w_{ij}$ o lado direito de (5.60). Aqui, vamos impor o limite superior de (5.60), pois, em termos de relaxação linear, torna a formulação mais “apertada”. Além disso, seria suficiente que a restrição fosse adicionada apenas para pares $\{i, j\} \in E(G)$, mas optamos por adicionar todas elas, embora aumente o tamanho do modelo, pois poderiam potencialmente fortalecer a relaxação linear da formulação.
3. A estratégia da formulação (AP) de registrar os potenciais nas arestas em lugar de associá-los aos vértices, como na formulação (CR), possibilita um fortalecimento da relaxação linear, às custas do uso de um maior número de variáveis. De fato, a partir das definições das variáveis dessas formulações, chegamos à relação $\pi_j^r = \sum_{v \in N^-(j)} \tau_{vj}^r$, para todo $(r, j) \in A(G)$. Então, se trocarmos o lado esquerdo de (5.59) de τ_{ij}^r para $\sum_{v \in N^-(j)} \tau_{vj}^r$ obtemos uma desigualdade mais fraca, porém ainda válida. Usando a relação entre π e τ , a desigualdade enfraquecida torna-se $\pi_j^r \geq \pi_i^r + w_{ij} - M_{ij}^r(1 - \lambda_{ij}^r)$, ou seja, a desigualdade MTZ (5.11). De maneira similar, a mesma transformação de variáveis feita sobre (5.60) resulta em (5.47). Desse modo, chegamos à formulação (CR) (com as desigualdades MTZ originais).

5.3 Desigualdades Válidas

Nesta seção, apresentaremos algumas desigualdades válidas para as formulações. Adicionalmente, apresentaremos soluções fracionárias que inspiraram cada desigualdade que elimina essa solução do conjunto viável relaxado.

5.3.1 Desigualdade Potencial Mínimo

A primeira desigualdade será denominada potencial mínimo. Antes de derivá-la, apresentamos uma solução ótima fracionária (da relaxação linear) que ela corta, relativa à instância do AtSCM ilustrada na Figura 20. Nas Figuras 21–23 e nas figuras 24–26 exibimos, respectivamente, uma solução para a relaxação linear das formulações (AP) e (CR). Como observação, não adicionamos o limite inferior das variáveis π (5.47) na formulação (CR).

Figura 20 – Grafo ponderado G , $t = 3$ 

Fonte: elaborado pelo autor.

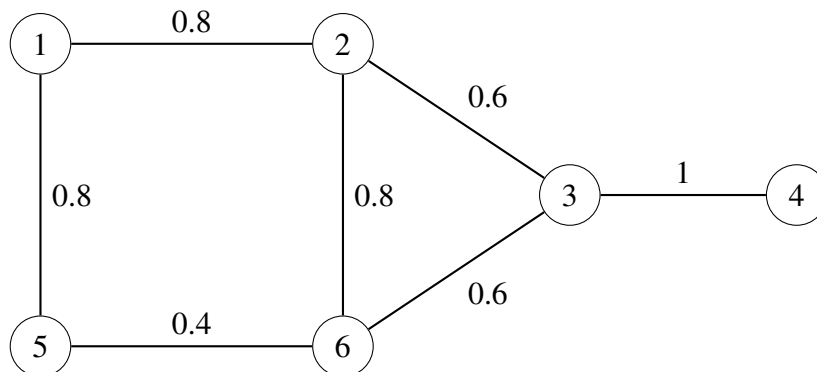
5.3.1.1 Desigualdade para formulação AP

Sejam $(i, j) \in A(G)$ e $r \in V(G)$. Considere uma solução $(\bar{x}, \bar{\lambda}, \bar{\tau})$ da formulação (AP). Primeiro suponha que $i = r$. Se o arco (r, j) está presente na arborescência enraizada em r , ou seja, $\bar{\lambda}_{rj}^r = 1$, então $\bar{\tau}_{rj}^r = w_{rj}$. Caso contrário, $\bar{\lambda}_{rj}^r = \bar{\tau}_{rj}^r = 0$. Em ambos os casos, temos a igualdade $\bar{\tau}_{rj}^r = w_{rj} \bar{\lambda}_{rj}^r$. Suponha agora $i \neq r$. Nesse caso, $\bar{\tau}_{ij}^r = 0$, se $\bar{\lambda}_{ij}^r = 0$, e $\bar{\tau}_{ij}^r = w_{ij} + \sum_{v \in N^-(i)} \bar{\tau}_{vi}^r$, se $\bar{\lambda}_{ij}^r = 1$. Assim, $\bar{\tau}_{ij}^r \geq (w_{ij} + \sum_{v \in N^-(i)} \bar{\tau}_{vi}^r) \bar{\lambda}_{ij}^r$. Como $\sum_{v \in N^-(i)} \bar{\tau}_{vi}^r \geq \text{dist}_G(i, r)$, obtemos $\bar{\tau}_{ij}^r \geq (w_{ij} + \text{dist}_G(i, r)) \bar{\lambda}_{ij}^r$. De maneira geral, as seguintes desigualdades são válidas para AP:

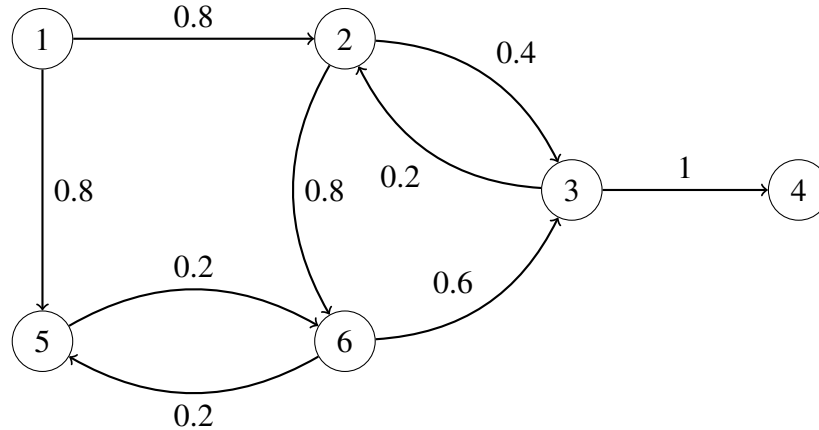
$$\bar{\tau}_{rj}^r = w_{rj} \bar{\lambda}_{rj}^r \quad \forall (r, j) \in A(G), r \in V(G) \quad (5.64)$$

$$\bar{\tau}_{ij}^r \geq (w_{ij} + \text{dist}_G(i, r)) \bar{\lambda}_{ij}^r \quad \forall (i, j) \in A(G), r \in V(G), i \neq j \quad (5.65)$$

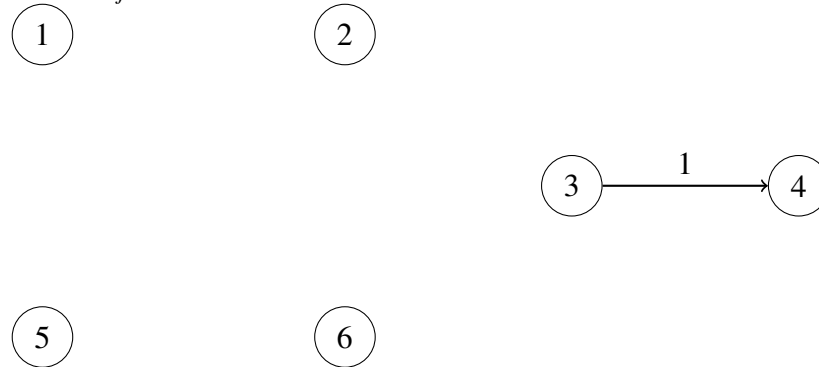
Observe que as desigualdades acima são interessantes para a formulação (AP), visto que a solução fracionária ilustrada nas Figuras 21, 22 e 23 é eliminada com a adição de (5.64)-(5.65) ao modelo. Esse fato pode ser observado, por exemplo, considerando a variável τ_{34}^1 . De (5.65), obtemos a violação $1 = \bar{\tau}_{34}^1 \geq (w_{34} + \text{dist}_G(1, 3)) \bar{\lambda}_{34}^1 = 3$.

Figura 21 – Variáveis \bar{x}_e 

Fonte: elaborado pelo autor.

Figura 22 – Variáveis $\bar{\lambda}_{ij}^1$, raiz 1

Fonte: elaborado pelo autor.

Figura 23 – Variáveis $\bar{\tau}_{ij}^1$, raiz 1

Fonte: elaborado pelo autor.

5.3.1.2 Desigualdade para formulação CR

As variáveis π e τ , utilizadas para o cálculo das distâncias nas formulações (CR) e (AP), respectivamente, estão relacionadas pela seguinte expressão:

$$\pi_j^r = \sum_{i \in N^-(j)} \tau_{ij}^r \quad \forall j, r \in V(G).$$

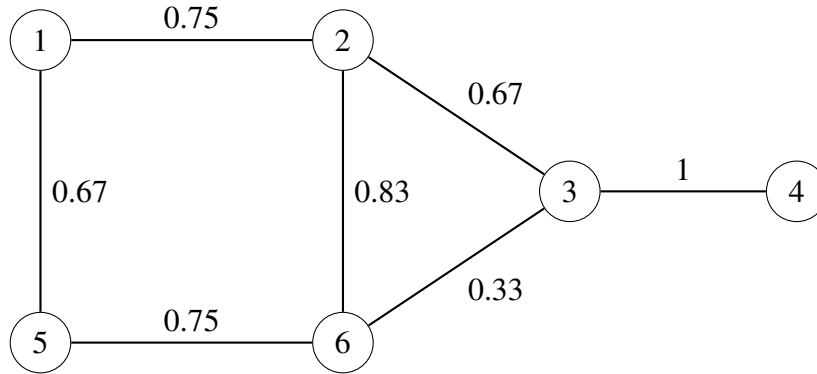
Essa relação nos leva a obter desigualdades de potencial mínimo também para a formulação (CR). Assumindo $\text{dist}_G(i, i) = 0$, para $i \in V(G)$, ao somarmos as (in)equações (5.64) e (5.65) relativas a $i \in N^-(j)$ e $r \in V(G)$ obtemos: $\sum_{i \in N^-(j)} \tau_{ij}^r \geq \sum_{i \in N^-(j)} (w_{ij} + \text{dist}_G(i, r)) \lambda_{ij}^r$. Então, substituindo a expressão de π_j^r destacada acima, obtemos a seguinte desigualdade válida para a formulação (CR):

$$\pi_j^r \geq \sum_{i \in N^-(j)} (w_{ij} + \text{dist}_G(i, r)) \lambda_{ij}^r \quad \forall j, r \in V(G) \quad (5.66)$$

O mesmo fato que ocorre na formulação (AP) é observado na formulação (CR) quando adicionamos a desigualdade (5.66). A solução fracionária de (CR) ilustrada nas Figuras

24, 25 e 26 é eliminada do conjunto viável relaxado. Por exemplo, observamos a violação de (5.66) em $2 = \bar{\pi}_4^1 \geq (w_{34} + \text{dist}_G(1, 3))\bar{\lambda}_{34}^1 = 3$.

Figura 24 – Variáveis \bar{x}_e



Fonte: elaborado pelo autor.

Figura 25 – Variáveis $\bar{\lambda}_{ij}^1$, raiz 1

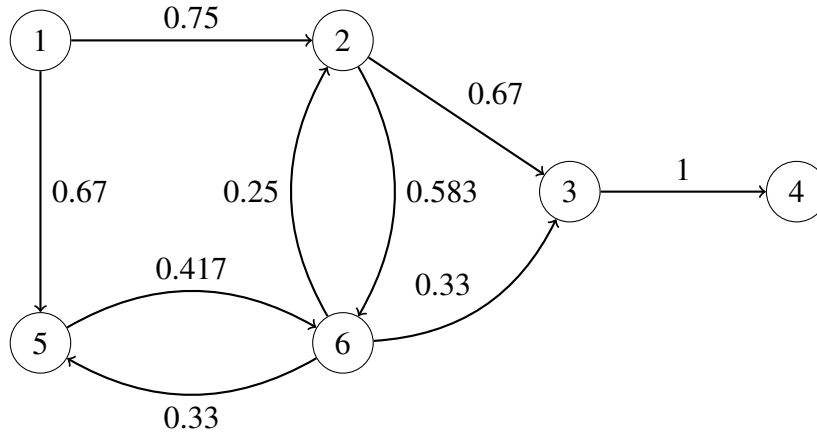
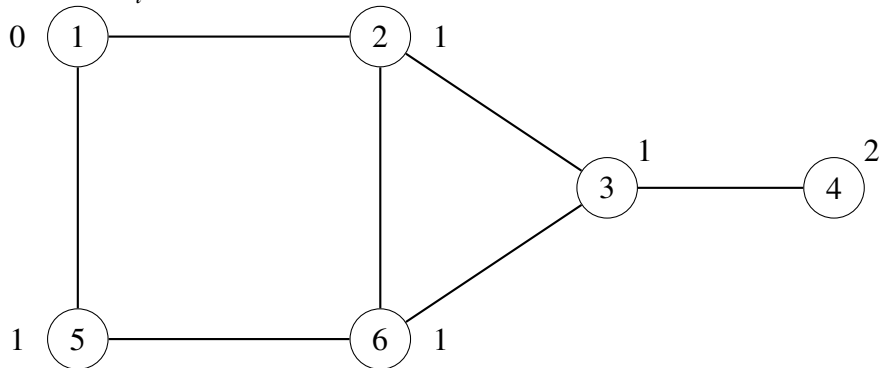


Figura 26 – Variáveis $\bar{\pi}_i^1$, raiz 1



Fonte: elaborado pelo autor.

5.3.1.3 Desigualdade para formulação SR

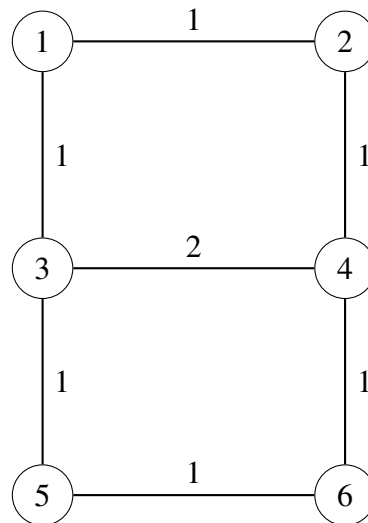
As desigualdades de potencial mínimo também podem ser adaptadas para a formulação (SR). Nessa formulação, o peso do caminho entre dois vértices r e j é dado por $\sum_{(a,b) \in E(G)} w_{ab} y_{ab}^{jr}$. Dado que esse valor corresponde ao potencial π_j^r da formulação (CR), a versão das desigualdades (5.66) para a formulação (SR) fica:

$$\sum_{\{a,b\} \in E(G)} w_{ab} y_{ab}^{jr} \geq \sum_{i \in N^-(j)} (w_{ij} + \text{dist}_G(i, r)) \lambda_{ij}^r \quad \forall \{j, r\} \in E(G) \quad (5.67)$$

5.3.2 Desigualdade Ciclo t -Inviável

A desigualdade válida que iremos apresentar agora denominamos de ciclo t -inviável. Como o próprio nome sugere, ela se baseia em um ciclo do grafo de entrada. Mais ainda, tal ciclo incorporará um conjunto de caminhos t -inviáveis, que serão proibidos. Como exemplo, considere a instância do AtSCM dada pelo grafo da Figura 27. Destacamos o ciclo $C = \langle 1, 2, 4, 6, 5, 3, 1 \rangle$ desse grafo. Evidentemente, pelo menos uma aresta de C não pode pertencer à solução. Todavia, observe que a remoção de qualquer uma das arestas $\{\{1, 2\}, \{2, 4\}, \{4, 6\}, \{5, 6\}, \{3, 5\}, \{1, 3\}\}$ do ciclo gera um caminho de tamanho 5. Sem perda de generalidade, seja $\{1, 2\}$ a aresta removida. O peso do caminho resultante P_{12} é 5, mas a distância máxima permitida entre 1 e 2 numa solução é 4, ou seja, P_{12} é um caminho 4-inviável. Então pelo menos mais uma aresta de C não pode pertencer à solução. Essa ideia nos conduz à definição formal de ciclo t -inviável.

Figura 27 – Grafo ponderado G , $t = 4$



Fonte: elaborado pelo autor.

Definição 5.3.1 Um ciclo C em G é dito ciclo t -inviável se $w(C) > (t + 1) \max_{e \in E(C)} w_e$. Em particular, no caso não ponderado, C é t -inviável se $|C| > t + 1$.

Proposição 5.3.1 O caminho obtido de um ciclo t -inviável C com a remoção de qualquer aresta de C é t -inviável.

Prova. Sejam C um ciclo t -inviável e $e^* = \arg \max_{e \in E(C)} w_e$. Seja P_{ij} o caminho obtido de C com a remoção da aresta $\{i, j\} \in E(C)$. Observe que $w(C) = w(P_{ij}) + w_{ij}$. Portanto, $w(P_{ij}) + w_{ij} > tw_{e^*} + w_{e^*}$, ou seja, $w(P_{ij}) > tw_{e^*} + (w_{e^*} - w_{ij})$. Como $w_{e^*} \geq w_{ij} \geq \text{dist}_G(i, j)$ e $(w_{e^*} - w_{ij}) \geq 0$, temos $w(P_{ij}) > t \text{dist}_G(i, j)$. Logo, P_{ij} é um caminho t -inviável. ■

Como resultado direto desta propriedade, a seguinte desigualdade é válida para as formulações apresentadas, para todo ciclo C t -inviável:

$$\sum_{e \in E(C)} x_e \leq |C| - 2 \quad (5.68)$$

De modo similar, seja $N(C)$ o conjunto de arestas incidentes em C que não pertencem a C ; em outras palavras, $N(C) = \{\{i, j\} \in E(G) \setminus E(C) : i \in V(C) \text{ ou } j \in V(C)\}$. As desigualdades a seguir também são válidas e proíbem que $|C| - 1$ ou mais arestas do ciclo sejam selecionadas para a solução:

$$\sum_{e \in N(C)} x_e \geq 1, \quad \text{se } |C| = |V(G)| \quad (5.69)$$

$$\sum_{e \in N(C)} x_e \geq 2, \quad \text{se } |C| < |V(G)| \quad (5.70)$$

Apesar de apresentarem expressões diferentes, na verdade (5.68) garante (5.69)–(5.70) e, além disso, é mais “apertada” quando se trata da relaxação linear das variáveis x . Essa conclusão segue da seguinte proposição:

Proposição 5.3.2 $\sum_{e \in E(C)} x_e \leq |C| - 2$ implica $\sum_{e \in N(C)} x_e \geq 1$, se $|C| = |V|$, ou $\sum_{e \in N(C)} x_e \geq 2$, caso contrário.

Prova. Seja \bar{x} uma solução fracionária satisfazendo as restrições de eliminação de sub-rotas (5.7), apresentadas na formulação exponencial. Seja C ciclo em G . Por hipótese, $\sum_{e \in E(C)} \bar{x}_e \leq |C| - 2$. Suponha primeiro que C é não hamiltoniano. Seja $S = V(G) \setminus V(C)$. Então $\sum_{e \in E(S)} \bar{x}_e \leq |S| - 1$. Além disso, $|V(G)| - 1 = \sum_{e \in E(C)} \bar{x}_e + \sum_{e \in N(C)} \bar{x}_e + \sum_{e \in E(S)} \bar{x}_e \leq |C| - 2 + \sum_{e \in N(C)} \bar{x}_e + |S| - 1 = |V(G)| - 3 + \sum_{e \in N(C)} \bar{x}_e$, portanto, $\sum_{e \in N(C)} \bar{x}_e \geq 2$. Se C é hamiltoniano então $|V(G)| - 1 =$

$\sum_{e \in E(C)} \bar{x}_e + \sum_{e \in N(C)} \bar{x}_e \leq |C| - 2 + \sum_{e \in N(C)} \bar{x}_e = |V(G)| - 2 + \sum_{e \in N(C)} \bar{x}_e$, portanto, $\sum_{e \in N(C)} \bar{x}_e \geq 1$. ■

Vale observar que as desigualdades (5.68) são válidas para todas as formulações apresentadas neste trabalho. Mais ainda, elas dominam as desigualdades (5.3) para todos os caminhos t -inviáveis obtidos com a remoção de uma aresta do ciclo. De fato, se P é o caminho obtido de um ciclo t -inviável C com a remoção de uma aresta qualquer, então $\sum_{e \in E(P)} x_e \leq \sum_{e \in E(C)} x_e \leq |C| - 2 = |P| - 1$. Adicionalmente, mostramos a seguir que as desigualdades (5.68) podem substituir as restrições t -spanner no caso do problema de decisão, o AtS. Nesse caso, como $w_e = 1$, para toda aresta $e \in E(G)$, um ciclo C é t -inviável se $|C| > t + 1$. Vamos mostrar que qualquer árvore geradora que satisfaça as restrições de ciclo t -inviável é t -spanner.

Proposição 5.3.3 *Seja T uma árvore geradora de G e $x \in \mathbb{B}^{|E|}$ tal que $x_e = 1$ se, e somente se, $e \in E(T)$. Se, para todo ciclo C t -inviável,*

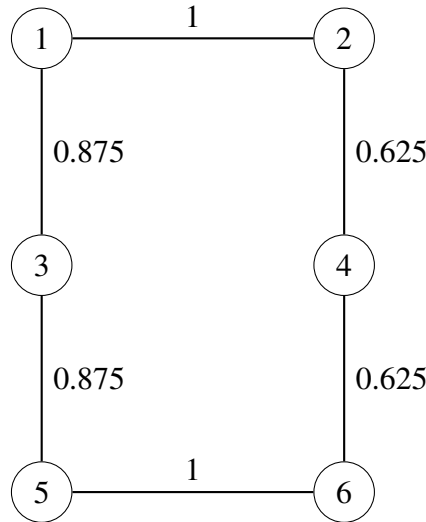
$$\sum_{e \in E(C)} x_e \leq |C| - 2,$$

então T é árvore t -spanner.

Prova. Suponha, pela contrapositiva, que T não é t -spanner. Deste modo, pelo Teorema 3.2.1, existe um par $\{i, j\} \in E(G)$ tal que $\text{dist}_T(i, j) > t * \text{dist}_G(i, j) = t$. Seja P o caminho (t -inviável) entre i e j na árvore T . Então, $w(P) > t$. Observe que a adição da aresta $\{i, j\}$ a P gera um ciclo C . Como $w(C) = w(P) + 1$, temos $w(C) > t + 1$, ou seja, C é um ciclo t -inviável. Além disso, (5.68) para C está violada, pois $\sum_{e \in E(C)} x_e = |P| = |C| - 1$. ■

Ilustramos na Figura 28 um exemplo onde as desigualdades de ciclo t -inviável cortam a solução da relaxação linear da formulação (CR). Consideramos o caso sem pesos e $t = 4$. O ciclo apresentando na figura é 4-inviável. Associado a cada aresta, é mostrado na figura o valor das variáveis x para uma solução fracionária da formulação (CR). Observe que a adição das desigualdades (5.68) à formulação é interessante, pois a solução fracionária passa a ser inviável com a adição da restrição $x_{12} + x_{24} + x_{46} + x_{56} + x_{35} + x_{13} \leq 4$. O caminho (t -inviável) obtido com a remoção da aresta $\{2, 4\}$ levaria também a uma restrição violada, $x_{12} + x_{46} + x_{56} + x_{35} + x_{13} \leq 4$, porém mais fraca que a anterior.

Apesar da adição das desigualdades (5.68) ser interessante, determinar um ciclo t -inviável pode ser computacionalmente difícil. Vamos mostrar que decidir se existe um ciclo t -inviável é NP -completo mesmo para grafos com peso unitário.

Figura 28 – Variáveis x_e 

Fonte: elaborado pelo autor.

Proposição 5.3.4 *Seja G um grafo não ponderado e $t \in \mathbb{Z}^+$. Decidir se G contém ciclo t -inviável é NP-completo.*

Prova. Claramente o problema está em NP, pois, dado um conjunto de arestas, podemos verificar em tempo polinomial se ele induz um ciclo C e se $|C| > t + 1$. Considere agora o problema do ciclo hamiltoniano em G , que é NP-Completo. Afirmamos que ele equivale a decidir se G contém ciclo t -inviável, para $t = |V(G)| - 2$. De fato, um ciclo $(|V(G)| - 2)$ -inviável em G contém $|V(G)|$ vértices. Logo, o problema em foco é NP-Completo. ■

5.3.3 Desigualdade t -Spanner Arborescente

Considere uma formulação para o AtSCM baseada na formulação de Martin (1991), que usa as variáveis λ_{ij}^r para definir as $|V(G)|$ arborescências. A partir de (5.29) e da Proposição 5.2.1, podemos expressar a restrição t -spanner (3.1) como:

$$\sum_{\{i,j\} \in E(G)} w_{ij} \max\{\lambda_{ij}^u - \lambda_{ij}^v, \lambda_{ji}^u - \lambda_{ji}^v\} \leq t * \text{dist}_G(u, v).$$

Seja, então, $E' \subseteq E(G)$. Trivialmente, temos

$$\sum_{\{i,j\} \in E'} (\lambda_{ij}^u - \lambda_{ij}^v) + \sum_{\{i,j\} \in E(G) \setminus E'} (\lambda_{ji}^u - \lambda_{ji}^v) \leq \sum_{\{i,j\} \in E(G)} \max\{\lambda_{ij}^u - \lambda_{ij}^v, \lambda_{ji}^u - \lambda_{ji}^v\}.$$

Sendo assim, chegamos às seguintes desigualdades válidas para as três formulações compactas:

$$\sum_{\{i,j\} \in E'} w_{ij} (\lambda_{ij}^u - \lambda_{ij}^v) + \sum_{\{i,j\} \in E(G) \setminus E'} w_{ij} (\lambda_{ji}^u - \lambda_{ji}^v) \leq t * \text{dist}_G(u, v), \forall \{u, v\} \in E(G), \forall E' \subseteq E(G) \quad (5.71)$$

Visto que $\lambda_{ij}^u + \lambda_{ji}^u = \lambda_{ij}^v + \lambda_{ji}^v$, ou melhor $\lambda_{ij}^u - \lambda_{ij}^v = -(\lambda_{ji}^u - \lambda_{ji}^v)$, podemos reescrever (5.71) como

$$\sum_{\{i,j\} \in E'} w_{ij}(\lambda_{ij}^u - \lambda_{ij}^v) + \sum_{\{i,j\} \in E(G) \setminus E'} w_{ij}(\lambda_{ij}^v - \lambda_{ij}^u) \leq t * \text{dist}_G(u, v), \forall \{u, v\} \in E(G), \forall E' \subseteq E(G) \quad (5.72)$$

Essas desigualdades são em número exponencial, mas podem ser separadas em tempo polinomial. Dado $\bar{\lambda}$ e fixada uma aresta $\{u, v\}$, o subconjunto $E' \subseteq E(G)$ que maximiza o lado esquerdo da expressão (5.72) é $E' = \{\{i, j\} \in E : \lambda_{ij}^u - \lambda_{ij}^v \geq 0\}$, levando ao valor máximo $\sum_{\{i,j\} \in E(G)} w_{ij} |\lambda_{ij}^u - \lambda_{ij}^v|$. Assim, caso $\sum_{\{i,j\} \in E(G)} w_{ij} |\lambda_{ij}^u - \lambda_{ij}^v| > t * \text{dist}_G(u, v)$, a desigualdade referente a $\{u, v\}$ e E' está violada; caso contrário, todas as desigualdades relativas a $\{u, v\}$ estão satisfeitas.

Vale observar que as desigualdades (5.71) são redundantes na formulação (SR). De fato, somando (5.27) para $\{i, j\} \in E'$ e (5.28) para $\{i, j\} \in E(G) \setminus E'$, obtemos

$$\sum_{\{i,j\} \in E'} y_{ij}^{uv} + \sum_{\{i,j\} \in E(G) \setminus E'} y_{ij}^{uv} \geq \sum_{\{i,j\} \in E'} (\lambda_{ij}^u - \lambda_{ij}^v) + \sum_{\{i,j\} \in E(G) \setminus E'} (\lambda_{ji}^u - \lambda_{ji}^v).$$

Então, a restrição (5.29), ou seja,

$$\sum_{\{i,j\} \in E(G)} w_{ij} y_{ij}^{uv} \leq t * \text{dist}_G(u, v)$$

domina (5.71).

Por outro lado, a inclusão das desigualdades (5.71) em (SR) dispensa o uso das variáveis y (e restrições em que essas variáveis aparecem). Dessa forma, obtemos uma formulação alternativa, como bem menos variáveis, porém com um número exponencial de restrições, mas que podem ser separadas de forma eficiente.

6 EXPERIMENTOS COMPUTACIONAIS

Neste capítulo, iremos apresentar os resultados computacionais obtidos pelas formulações matemáticas e pelo algoritmo enumerativo. Para os modelos matemáticos apresentaremos o tempo necessário para resolução, bem como a quantidade de instâncias resolvidas e valor de solução, para as instâncias viáveis. Além disso, o desempenho das formulações será comparado entre a implementação com o grupo de restrições de sua descrição e a implementação com a adição das desigualdades propostas na seção 5.3. Para o algoritmo enumerativo será apresentada apenas um tabela de resultados, visto que seu desempenho não se mostrou promissor. Vale mencionar que não conseguimos acesso às instâncias utilizadas por Braga (2019), portanto, iremos gerar as instâncias estudadas de maneira similar às dele.

6.1 Instâncias Estudadas

Para os testes computacionais, foram utilizados grafos gerados aleatoriamente. Para gerar cada grafo, usamos dois parâmetros de entrada: a quantidade de vértices $n \in \mathbb{N}^*$; e uma probabilidade $p \in [0, 1]$ relativa à ocorrência de aresta. Quanto à ponderação das arestas, separamos em três casos: pesos unitários (AtS), pesos aleatórios e distância euclidiana. A seguir temos com precisão os valores de cada parâmetro:

- Quantidade de vértices n ($|V(G)|$) variando em $\{15, 30, 45, 60\}$.
- Probabilidade p de ocorrência de cada aresta variando em $\{0.2, 0.5, 0.8, 1\}$.
- Fator de dilatação t variando em $\{1.1, 2, 3, 4\}$.
- Função de ponderação das arestas dividida em três casos:
 - Pesos Unitários ($w_e = 1$, para $e \in E(G)$).
 - Distância euclidiana, na qual cada vértice inicialmente recebe uma coordenada aleatória (x, y) em um quadrado de lado 100 e, para cada aresta $\{i, j\} \in E(G)$, seu peso será a distância euclidiana entre i e j no quadrado.
 - Pesos arbitrários separados em dois sub-casos:
 - * Pesos mais espaçados, variando em $\{1, 2, 4, 8, 16\}$.
 - * Pesos menos espaçados, variando em $\{1, 2, 3, 4\}$.

Os parâmetros definidos acima e suas valorações são os mesmos usados por Braga (2019), sendo que incluímos o peso 4 no caso mais espaçado.

Sejam os três parâmetros n , p e t fixos. Iniciamos o grafo da instância como uma

árvore com n vértices, para garantir conectividade do grafo. A seleção das arestas que irão compor a árvore é aleatória, cada uma com mesma probabilidade de escolha. Após a criação da árvore, para todo par $\{i, j\} \in V(G)$, se $\{i, j\}$ não é aresta da árvore inicial, geramos um número aleatório $p' \in [0, 1]$. Se $p' \leq p$ adicionamos a aresta $\{i, j\}$ ao grafo. Note que, quanto maior esse fator p , maior será a quantidade de arestas do grafo, o que se reflete diretamente na densidade do mesmo. Nesse ponto, já geramos uma instância para a versão de decisão do problema (AtS). Caso a instância seja ponderada e com distância euclidiana, o peso de toda aresta $\{i, j\}$ será a distância euclidiana entre as coordenadas de i e j . Caso seja de pesos arbitrários e fixado um dos conjuntos (mais ou menos espaçados), para toda aresta $\{i, j\} \in E(G)$, selecionamos aleatoriamente um peso dentre os elementos do conjunto, cada um com mesma probabilidade de escolha.

Embora a estrutura do grafo seja essencialmente definida pelo par (n, p) , vale ressaltar que um grafo diferente é gerado para cada combinação dos demais parâmetros. Por exemplo, o grafo com $n = 15$, $p = 0.5$, $t = 1.1$ e pesos unitários é diferente do grafo com $n = 15$, $p = 0.5$, $t = 2$ e essa mesma ponderação, que, por sua vez, também é diferente do grafo com $n = 15$, $p = 0.5$, $t = 1.1$ e pesos euclidianos.

Uma observação quanto as instâncias estudadas é que todas são definidas a partir de grafos 2-conexos. Descartamos instâncias constituídas por grafos com vértice de corte por conta da Proposição 3.2.2.

6.2 Ambiente Computacional

Os modelos foram resolvidos com uso do solver CPLEX (IBM, Versão 12.10). A implementação dos mesmos e do algoritmo enumerativo descritos ao longo do texto foi toda feita na linguagem de programação C++. Os experimentos serão realizados em duas máquinas com processador Intel ® Core™ i7-7700 CPU com 3.60 GHz, 32 GB RAM e sistema operacional Ubuntu. O tempo limite estabelecido para resolução de cada instância por cada método avaliado foi de 1200 segundos (20 minutos).

6.3 Desempenho das Formulações Básicas

Nesta seção, apresentamos os resultados obtidos pelo algoritmo enumerativo e pelas formulações matemáticas ((FE), (SR), (CR) e (AP)) sem adição de desigualdades válidas. As

formulações (SR), (CR) e (AP) foram implementadas conforme as expressões (5.22–5.32), (5.40–5.50) e (5.54–5.63), respectivamente, à exceção que a integralidade de x foi descartada. Quando à formulação (FE), apresentamos somente os resultados para a versão MTZ (5.13 – 5.19), pois esta se apresentou mais rápida que a versão SECs na maioria dos testes preliminares. Lembramos, porém, que o algoritmo enumerativo pode ser visto como um *branch-and-bound* aplicado à formulação (FE), onde o *bound* é dado pela relaxação obtida descartando as restrições de t -spanner e o *branch* é feito sobre essas restrições.

Registramos para cada método, em cada instância, o tempo demandado para resolução e seu valor da solução, caso a mesma admita uma árvore t -spanner. Informações detalhadas, instância por instância, podem ser vistas nas Tabelas 6, 7, 8, 9 e 10 nos apêndices. Na Tabela 3, exibimos o tempo médio e quantidade de instâncias solucionadas pelos métodos apresentados, considerando as 64 instâncias para cada um dos quatro tipos de ponderações das arestas. Na coluna ‘Método’ indicamos o método utilizado, e nas colunas ‘Tempo’ e ‘Resolvidas’, apresentamos, respectivamente, o tempo médio gasto (em segundos) e a quantidade de instâncias resolvidas no tempo limite.

Tabela 3 – Resultados computacionais resumidos para formulações básicas e algoritmo enumerativo

Método	Peso Mais Esp.		Peso Menos Esp.		Peso Unitário		Dist. Euclidiana	
	Resolvidas	Tempo	Resolvidas	Tempo	Resolvidas	Tempo	Resolvidas	Tempo
Form. CR	59	114.19	57	164.14	49	338.69	59	106.34
Form. SR	44	408.00	41	476.53	33	595.22	48	320.34
Form. AP	39	497.39	36	536.94	36	545.36	47	359.42
Alg. Enum.	45	358.17	42	428.61	42	429.80	42	416.14
FE-MTZ	9	1031.83	x	x	x	x	x	x

Fonte: Elaborado pelo autor.

A partir dos dados apresentados na Tabela 3, observamos que a formulação (CR) foi a que apresentou o melhor desempenho com relação às outras. Além de ser a que solucionou mais instâncias para cada ponderação, foi a que apresentou o menor tempo médio de execução. Vale observar que os tempos médios para as diferentes ponderações foram similares, com uma elevação para as instâncias com pesos unitários.

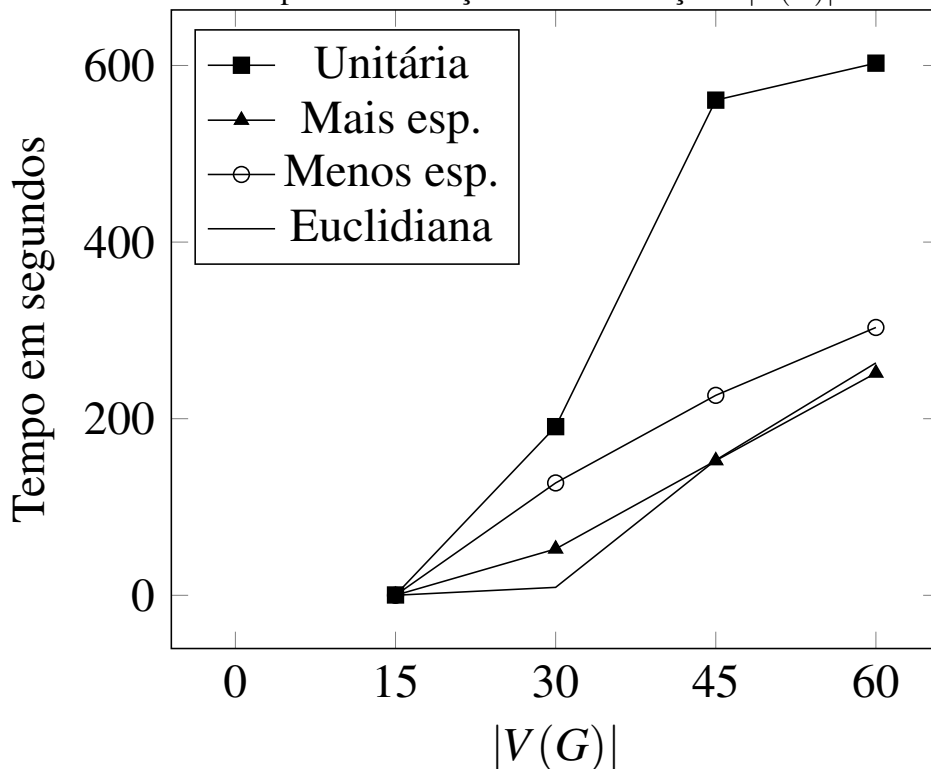
Nota-se ainda que as formulações (SR) e (AP) e o algoritmo enumerativo apresentaram desempenhos similares. Por outro lado, a formulação FE-MTZ mostrou-se bastante inferior às demais, tendo excedido o limite de tempo em quase todas as instâncias. Por esse motivo,

executamos essa formulação integralmente apenas com o grupo de instâncias com pesos mais espaçados. Nos demais grupos, o marcador ‘x’ indica que não temos os valores precisos, embora testes parciais apontaram comportamento similar.

Nos gráficos das Figuras 29, 30 e 31 ilustramos a variação do tempo médio demandado pela formulação (CR) para as instâncias estudadas em cada ponderação como função dos parâmetros $|V(G)|$, t e p , respectivamente. Observamos que, à medida que cada parâmetro cresce, o tempo médio também cresce. Esse comportamento é similar para os outros métodos. No caso da formulação (CR), observamos que seu desempenho foi bem pior no grupo de instâncias com pesos unitários. Nas demais ponderações, o desempenho se mostrou similar em função dos três parâmetros.

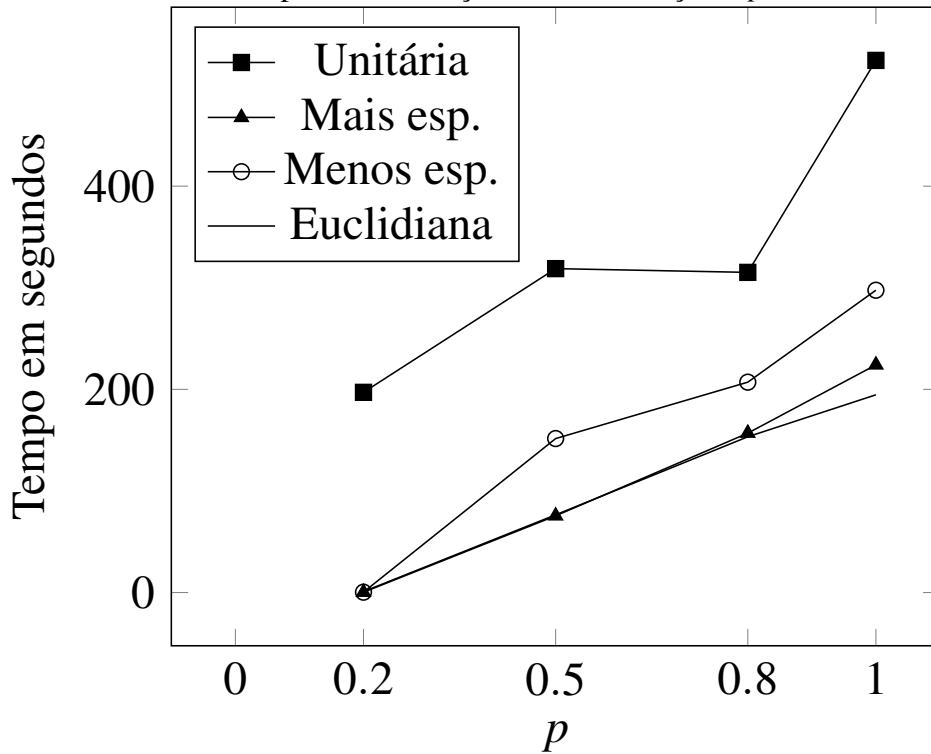
Particularmente com respeito ao fator de dilatação, destacamos que, para $t = 1.1$ e $t = 2$, os métodos demandaram pouco tempo computacional. Já para $t = 3$ e $t = 4$, para grafos a partir de 30 vértices, o tempo de resolução sofreu uma grande elevação, em muitos casos atingindo o tempo limite de 20 minutos (veja as tabelas instância por instância nos apêndices). Na Figura 31, observamos uma grande elevação do tempo médio de $t = 2$ para $t = 3$ nas instâncias com pesos unitários e de $t = 3$ para $t = 4$ nas demais.

Figura 29 – Crescimento do tempo da formulação CR com relação a $|V(G)|$



Fonte: elaborado pelo autor.

Figura 30 – Crescimento do tempo da formulação CR com relação a p



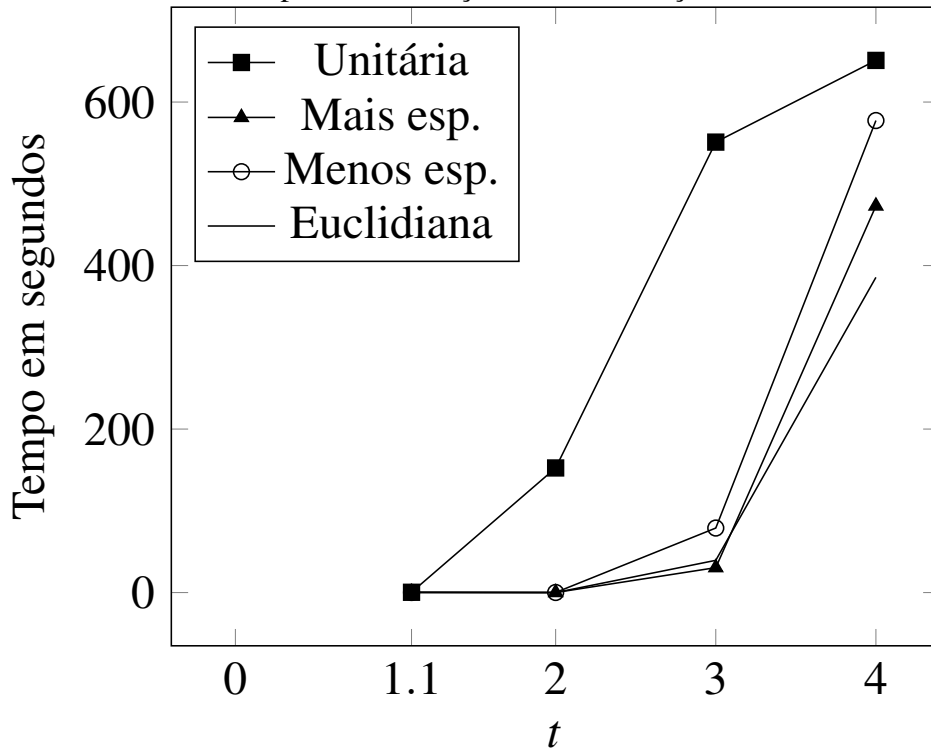
Fonte: elaborado pelo autor.

6.4 Desempenho das Formulações com Desigualdades Válidas

Nesta seção, adicionaremos desigualdades válidas às formulações (SR) e (CR), na tentativa de obter uma melhoria de desempenho. Vamos considerar apenas as instâncias com $t = 3$ e $t = 4$, a partir de 30 vértices, pois foram as que se apresentaram mais difíceis. Pelo mesmo motivo, utilizaremos as instâncias com pesos unitários para este experimento.

6.4.1 Variantes da Formulação SR

Vimos nos experimentos da Seção 6.3 que o desempenho da formulação (SR) foi pior quando comparado à formulação (CR). Um dos principais motivos é o maior número de variáveis de (SR). Na tentativa de melhorar sua performance, testamos duas variações no modelo. Na primeira, adicionamos as desigualdades (5.35)-(5.37) à formulação e priorizamos a ramificação nas variáveis x , agora consideradas inteiras. Como visto na observação 7 da página 57, a ramificação em x leva à fixação de muitas variáveis. Alternativamente, na segunda variação, removemos as variáveis y , bem como as desigualdades associadas, e adicionamos as

Figura 31 – Crescimento do tempo da formulação CR com relação a t 

Fonte: elaborado pelo autor.

desigualdades (5.72). Relembramos que as desigualdades (5.72) são em número exponencial, o que torna impraticável a adição de todas ao modelo inicialmente. Para tal, utilizamos a função *Lazy Constraint Callback* do CPLEX, que resolve a formulação sem (5.72) e, dada uma solução inteira relaxada, se existe uma desigualdade violada, adiciona a primeira detectada e reotimiza a formulação. Esse processo segue até que todas as desigualdades estejam satisfeitas. Testes também foram realizados adicionando, a cada iteração, todas as desigualdades violadas detectadas, porém resultando em desempenho inferior.

Tabela 4 – Resultados computacionais resumidos para variantes de SR

Método	Resolvidas	Tempo
Form. SR	5	979.50
Form. SR+ESP	5	955.83
Form. SR sem Y	10	768.46

Fonte: Elaborado pelo autor.

Na Tabela 4 exibimos o tempo médio (em segundos) e quantidade de instâncias solucionadas pela formulação e pelas duas variantes, considerando as 24 instâncias com $t \in \{3, 4\}$, pesos unitários e pelo menos 30 vértices. Identificamos por ‘SR+ESP’ a formulação com adição

das desigualdades específicas (5.35)-(5.37) e priorizando a ramificação nas variáveis x e por ‘SR sem Y’ a formulação com remoção das variáveis y e restrições em que elas aparecem e com adição das desigualdades (5.72).

Observe que o tempo médio de ‘SR+ESP’ é ligeiramente menor que o da formulação (SR). Por outro lado, a redução foi bem mais considerável para a variante ‘SR sem Y’. Além disso, observamos que o tempo de ‘SR sem Y’ foi menor que o da formulação (CR) para estas instâncias. Entretanto, na próxima seção apresentaremos uma melhora no tempo médio de (CR) com adição de desigualdades válidas que a torna mais rápida, em média, que ‘SR sem Y’. Para consultar a solução e tempo de cada instância veja a tabela 4 nos apêndices.

6.4.2 *Formulação CR com Desigualdades Válidas*

Observamos nos primeiros experimentos da Seção 6.3 que a formulação (CR) apresentou o melhor desempenho quando comparada aos demais métodos. Decidimos investir nessa formulação, fortalecendo-a com as desigualdades válidas propostas na Seção 5.3. O intuito é possivelmente melhorar o desempenho. Testamos três variantes, cada uma definida pela adição de um tipo de desigualdade: t -spanner arborescente, potencial mínimo e ciclo t -inviável.

Para a t -spanner arborescente, adicionamos as desigualdades (5.72) como cortes para a formulação, por meio da função *cut callback* do CPLEX. Para as de potencial mínimo, adicionamos todas as desigualdades (5.66) diretamente à formulação inicial. Com respeito às desigualdades (5.68), vimos na Proposição 5.3.4 que encontrar ciclos t -inviáveis é computacionalmente difícil para grafos com pesos unitários, logo determinar todas essas desigualdades é impraticável. Como estratégia para determinar algumas delas, calculamos uma árvore geradora T do grafo. Se T já é um t -spanner, não resolvemos a formulação pois encontramos sua solução. Caso contrário, vimos na demonstração da Proposição 5.3.3 como detectar um ciclo t -inviável. Portanto, adicionamos todas as desigualdades de ciclo t -inviável, relativas a T , à formulação (CR).

Na Tabela 5, exibimos o tempo médio (em segundos) e quantidade de instâncias solucionadas pela formulação (CR) e pelas duas variantes, considerando as 24 instâncias com $t \in \{3,4\}$, pesos unitários e pelo menos 30 vértices. Identificamos por ‘CR+tSA’, ‘CR+CtI’ e ‘CR+PM’ a formulação com adição, respectivamente, das desigualdades (5.72), (5.68) e (5.66).

Observe que a adição de (5.72) piorou o desempenho da formulação (CR). Além de aumentar o tempo médio, a formulação resolveu bem menos instâncias. Por outro lado, a adição

Tabela 5 – Resultados computacionais resumidos para variantes de CR

Método	Resolvidas	Tempo
Form. CR	11	801.08
Form. CR+tSA	6	891.42
Form. CR+CtI	10	737.87
Form. CR+PM	11	750.58

Fonte: Elaborado pelo autor.

de (5.68) ou (5.66) reduziu o tempo médio gasto pelo modelo, fazendo com que sua performance volte a ser a melhor, ultrapassando a variante ‘SR sem Y’ testada na subseção anterior.

7 CONCLUSÃO

Neste trabalho, estudamos o problema da árvore t -spanner de custo mínimo. Apresentamos a definição do problema bem como sua complexidade, além de elencar algumas propriedades que se mostraram interessantes para o desenvolvimento dos métodos de resolução. Usamos uma abordagem de programação matemática, pouco explorada nesse tema. Apresentamos e comparamos por meio dos testes computacionais modelos de programação linear inteira e um algoritmo enumerativo. Dentre as formulações apresentadas, duas delas foram propostas por Braga (2019), e, até então, eram as únicas presentes na literatura do problema, as restantes (formulação (AP) e (FE)), aqui apresentadas, foram introduzidas neste trabalho.

Propusemos um algoritmo enumerativo para o problema, o primeiro que não utiliza resolução de formulações matemáticas como um de seus componentes. Para as instâncias com peso euclidiano, ele apresentou o maior tempo médio. Por outro lado, nas demais classes, foi o segundo método mais rápido, em média, perdendo apenas para a formulação (CR). Em particular, no caso de pesos unitários, o algoritmo consegue resolver otimamente, no tempo de 20 minutos, instâncias viáveis que não foram possível com (CR).

Apresentamos e avaliamos quatro formulações matemáticas neste trabalho. Como contribuição efetiva, acrescentamos duas novas formulações para a literatura do problema, assim como estudamos e fortalecemos a relaxação linear das formulações de Braga (2019), tornando as restrições mais apertadas. Particularmente, reduzimos os valores dos *big-M*'s usados. Além disso, propusemos algumas desigualdades válidas, tanto para as nossas formulações, quanto para as da literatura. Os experimentos computacionais apresentaram o método baseado na formulação (CR) como aquele com melhor desempenho computacional dentre os outros, nos testes realizados. Conseguimos melhorar a performance da formulação (SR) com a eliminação das variáveis y e restrições associadas, adicionando as desigualdades (5.72). Com a adição das desigualdades (5.68) e (5.66), também melhoramos a performance da formulação (CR), que continuou a ser o método mais rápido.

Introduzimos neste trabalho o conceito de ciclo t -inviável. Provamos que determinar esse tipo de ciclo em um grafo é um problema computacionalmente difícil, mesmo para o caso com pesos unitários. Além disso, para esse caso, provamos que uma árvore geradora que satisfaz as desigualdades (5.68) é também uma árvore t -spanner. Ficou em aberto uma prova ou contra-exemplo de que esse resultado também vale para o caso com pesos. Na prática, vimos que a adição de (algumas das) desigualdades de ciclo t -inviável à formulação (CR) melhorou

sua performance para o caso unitário. Como trabalho futuro, pretendemos estudar o problema de separação associado a essas desigualdades e encontrar um modo mais eficiente de gerar uma maior quantidade delas. Na linha do que fizemos, uma alternativa seria verificar várias árvores geradoras a priori, e não apenas uma, como foi implementado para os experimentos computacionais. Outra possibilidade, seria gerar essas árvores ao longo do *branch-and-bound*, usando a informação da solução corrente.

Mostramos aqui que é possível, pela Proposição 3.2.2, admitir ser 2-conexo o grafo de entrada da instância. Caso contrário, podemos fazer a decomposição em blocos do grafo e resolver o problema separadamente em cada bloco. Como trabalho futuro, podemos verificar se realizar a decomposição em blocos para grafos que possuem vértice de corte apresenta vantagem em tempo computacional resolvendo o problema em cada bloco, ou se é mais vantajoso resolver logo o problema para o grafo todo. Há diferentes algoritmos para determinar a árvore bloco-articulação de um grafo, em geral, eles se baseiam em uma busca em profundidade. O algoritmo clássico foi proposto em Hopcroft e Tarjan (1973) e é uma alternativa para realizar esse experimento.

Uma outra direção de pesquisa é a utilização de métodos alternativos para resolução das formulações como Decomposição de Benders, Dantzig-Wolfe, Relaxação Lagrangeana, entre outras. Além disso, pode-se aperfeiçoar ainda mais as formulações, por meio de novas desigualdades válidas e um estudo poliédrico de seus conjuntos viáveis. Também é relevante o aprimoramento dos passos de ramificação e poda do algoritmo enumerativo, visto que foi um método competitivo com as formulações, apesar de não ser atualmente o melhor.

REFERÊNCIAS

- ALTHOFER, I.; DAS, G.; DOBKIN, D.; JOSEPH, D.; SOARES, J. On sparse spanners of weighted graphs. **Discrete Comput. Geom.**, [s.l.], v. 9, n. 1, p. 81 – 100, 1993.
- ANDRADE, R. C.; FREITAS, A. T. Disjunctive combinatorial branch in a subgradient tree algorithm for the dcmst problem with vns-lagrangian bounds. **Electronic Notes in Discrete Mathematics, Elsevier**, [s.l.], v. 41, p. 5 – 12, 2013.
- ARIKATI, S. R.; CHEN, D. Z.; CHEW, L. P.; DAS, G.; SMID, M. H. M.; ZAROLIAGIS, C. D. Planar spanners and approximate shortest path queries among obstacles in the plane. **Proceedings of the Fourth Annual European Symposium on Algorithms**, Springer, Berlin, Heidelberg, p. 514 – 528, 1996.
- AWERBUCH, B. Complexity of network synchronization. **J. ACM**, [s.l.], v. 32, p. 802 – 823, 1985.
- BATSYN, M.; GOLDENGORIN, B.; KOCHETUROV, A.; PARDALOS, P. M. Tolerance-based vs. cost-based branching for the asymmetric capacitated vehicle routing problem. **Models, Algorithms, and Technologies for Network Analysis**, [s.l.], p. 1 – 10, 2013.
- BAZARAA, M.; JARVIS, A.; SHERALI, H. **Linear Programming and Network Flows**. 4. ed. [s.l.]: Wiley, 2009. ISBN 978-0-470-46272-0.
- BELLMAN, R. On a routing problem. **Quarterly of Applied Mathematics**, [s.l.], v. 16, n. 1, p. 87 – 90, 1958.
- BRAGA, H. V. V. **Algoritmos exatos para problemas de spanner em grafos**. 2019. Tese (Doutorado em Ciência da Computação) - Universidade de São Paulo, São Paulo. 2019.
- BRANDSTADT, A.; CHEPOI, V.; DRAGAN, F. Distance approximating trees for chordal and dually chordal graphs. **J. Algorithms**, [s.l.], v. 30, n. 1, p. 166 – 184, 1999.
- BRANDSTADT, A.; DRAGAN, F. F.; LE, H.-O.; LE, V. B. Tree spanners on chordal graphs: complexity and algorithms. **Theor. Comput. Sci.**, v. 310, n. 1, p. 329 – 354, 2004.
- BRANDSTADT, A.; DRAGAN, F. F.; LE, H.-O.; LE, V. B.; UEHARA, R. Tree spanners for bipartite graphs and probe interval graphs. **Algorithmica**, v. 47, n. 1, p. 27 – 51, 2007.
- CAI, L. **Tree spanners: spanning trees that approximate distances**. 1992. Tese (Doutorado em Computer Science) - University of Toronto, Toronto. 1992.
- CAI, L. Np-completeness of minimum spanner problems. **Discrete Applied Mathematics**, [s.l.], v. 48, n. 2, p. 187 – 194, 1994.
- CAI, L.; CONEIL, D. G. Tree spanners. **Journal on Discrete Mathematics**, [s.l.], v. 8, n. 3, p. 359 – 387, 1995.
- CHANDRA, B.; DAS, G.; NARISIMHAN, G.; SOARES, J. New sparseness results on graph spanners. **Proceedings of the Eighth Annual Symposium on Computational Geometry**, New York, NY, USA, p. 192 – 201, 1992.

- CHECHIK, S.; WULFF-NILSEN, C. Near-optimal light spanners. **Proceedings of the Twenty-seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '16**, Philadelphia, PA, USA, p. 883 – 892, 2016.
- CONFORTI, M.; CORNUÉJOLS, G.; ZAMBELLI, G. Extended formulations in combinatorial optimization. **4OR - Q. J. Oper. Res.**, [s.l.], v. 8, p. 1 – 48, 2010.
- CONFORTI, M.; CORNUÉJOLS, G.; ZAMBELLI, G. Extended formulations in combinatorial optimization. **Annals OR**, [s.l.], v. 204, n. 1, p. 97 – 143, 2013.
- CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. **Introduction to Algorithms, Third Edition**. 3rd. ed. [s.l.]: The MIT Press, 2009. ISBN 0262033844, 9780262033848.
- DANTZIG, G. B.; FULKERSON, D. R.; JOHNSON, S. M. Solution of a large scale traveling salesman problem. **Operations Research**, JSTOR, v. 2, n. 4, p. 393 – 410, 1954.
- DESROCHERS, M.; LAPORTE, G. Improvements and extensions to the miller-tucker-zemlin subtour elimination constraints. **Operations Research Letters**, [s.l.], v. 10, p. 27 – 36, 1991.
- DIJKSTRA, E. W. A note on two problems in connexion with graphs. **Numerische Mathematik**, [s.l.], v. 1, n. 1, p. 269 – 271, 1959.
- DINITZ, M.; KORTSARZ, G.; RAZ, R. Label cover instances with large girth and the hardness of approximating basic k-spanner. **ACM Trans. Algorithms**, [s.l.], v. 12, n. 2, p. 1 – 16, 2015.
- DRAGAN, F. F.; FOMIN, F. V.; GOLOVACH, P. A. Spanners in sparse graphs. **J. Comput. Syst. Sci.**, [s.l.], v. 77, n. 6, p. 1108 – 1119, 2011.
- EDMONDS, J. Matroids and the greedy algorithm. **Mathematical Programming**, [s.l.], v. 1, p. 127 – 136, 1991.
- EMEK, Y.; PELEG, D. Approximating minimum max-stretch spanning trees on unweighted graphs. **SIAM Journal on Computing**, [s.l.], v. 38, n. 5, p. 1761 – 1781, 2008.
- FEKETE, S. P.; KREMER, J. Tree spanners in planar graphs. **Discrete Applied Mathematics**, [s.l.], v. 108, n. 1, p. 85 – 103, 2001.
- FLOYD, R. W. Algorithm 97: Shortest path. **Commun. ACM**, ACM, New York, NY, USA, v. 5, n. 6, jun. 1962. ISSN 0001-0782.
- FORD, L. R. Network flow theory. RAND Corporation, Santa Monica, CA, 1956.
- GALBIATI, G. On min-max cycle bases. **Algorithms and Computation, volume 2223 of Lecture Notes in Computer Science**, Springer-Verlag, p. 116 – 123, 2001.
- GOLDSCHMIDT, O.; TAKVORIAN, A.; YU, G. On finding a biconnected spanning planar subgraph with applications to the facilities layout problem. **European Journal of Operational Research**, [s.l.], v. 94, n. 1, p. 97 – 105, 1996.
- HOPCROFT, J.; TARJAN, R. Algorithm 447: efficient algorithms for graph manipulation. **Communications of the ACM**, [s.l.], v. 16, p. 372 – 378, 1973.
- IBM. **IBM CPLEX Optimizer**. 2021. Disponível em: <https://www.ibm.com/analytics/cplex-optimizer>.

- KORTSARZ, G. On the hardness of approximating spanners. **Algorithmica**, [s.l.], v. 30, n. 3, p. 432 – 450, 2001.
- KORTSARZ, G.; PELEG, D. Generating sparse 2-spanners. **Journal of Algorithms**, [s.l.], v. 17, n. 2, p. 222 – 236, 1994.
- KRUSKAL, J. On the shortest spanning subtree of a graph and the traveling salesman problem. **Proceedings of the American Mathematical Society**, American Mathematical Society, p. 48 – 51, 1956.
- MADANLAL, M.; VENKATESAN, G.; RANGAN, C. Tree 3-spanners on interval, permutation and regular bipartite graphs. **Information Processing Letters**, [s.l.], v. 59, n. 2, p. 97 – 102, 1996.
- MAGNANTI, T. L.; WOLSEY, L. A. Optimal trees. **Handbooks in Operations Research and Management Science**, Elsevier, v. 7, p. 503 – 615, 1995.
- MARBLE, J. D.; BEKRIS, K. E. Asymptotically near-optimal is good enough for motion planning. **Proc. of the 15th International Symposium on Robotics Research (ISRR-11)**, [s.l.], 2011.
- MARBLE, J. D.; BEKRIS, K. E. Computing spanners of asymptotically optimal probabilistic roadmaps. **IEEE/RSJ International Conference on Intelligent Robots and Systems**, San Francisco, CA, USA, p. 4292 – 4298, 2011.
- MARTIN, R. K. Using separation algorithms to generate mixed integer model reformulations. **Operations Research Letters**, [s.l.], v. 10, p. 119 – 128, 1991.
- MILLER, C. E.; TUCKER, A. W.; ZEMLIN, R. A. Integer programming formulation of traveling salesman problems. **J. ACM**, [s.l.], v. 7, n. 4, p. 326 – 329, 1960.
- NARAYANASWAMY, N.; RAMAKRISHNA, G. Tree t-spanners in outerplanar graphs via supply demand partition. **Discrete Applied Mathematics**, [s.l.], v. 195, p. 104 – 109, 2015.
- NGUYEN, C.; SHEN, J.; HOU, M.; SHENG, L.; MILLER, W.; ZHANG, L. Approximating the spanning star forest problem and its application to genomic sequence alignment. **SIAM Journal on Computing**, [s.l.], v. 38, n. 3, p. 946 – 962, 2008.
- OLIVEIRA, C. A. S.; PARDALOS, P. M. A survey of combinatorial optimization problems in multicast routing. **Comput. Oper. Res.**, [s.l.], v. 32, n. 8, p. 1953 – 1981, 2005.
- PAPOUTSAKIS, I. Tree 3-spanners of diameter at most 5. **CoRR**, [abs/1402.3573](https://arxiv.org/abs/1402.3573), [s.l.], 2014.
- PAPOUTSAKIS, I. Tree spanners of bounded degree graphs. **Discrete Appl. Math.**, [s.l.], v. 236, p. 395 – 407, 2018.
- PELEG, D.; RESHEF, E. A variant of the arrow distributed directory with low average complexity. **Automata, Languages and Programming**, ICAL '99, p. 615 – 624, 1999.
- PELEG, D.; SCHAFFER, A. A. Graph spanners. **Journal of Graph Theory**, [s.l.], p. 99 – 116, 1989.
- PELEG, D.; ULLMAN, J. D. An optimal synchronizer for the hypercube. **SIAM J. Comput.**, [s.l.], v. 18, n. 4, p. 740 – 747, 1989.

PELEG, D.; UPFAL, E. A tradeoff between space and efficiency for routing tables. **Proceedings of the twentieth annual ACM symposium on Theory of computing**, New York, NY, USA, p. 43 – 52, 1988.

PRIM, R. C. Shortest connection networks and some generalizations. **Bell System Technical Journal**, Nokia Bell Labs, v. 36, n. 6, p. 1389 – 1401, 1957.

VENKATESAN, G.; ROTICS, U.; MADANLAL, M.; MAKOWSKY, J.; RANGAN, C. P. Restrictions of minimum spanner problems. **Information and Computation**, [s.l.], v. 136, n. 2, p. 143 – 164, 1997.

WARSHALL, S. A theorem on boolean matrices. **J. ACM**, ACM, New York, NY, USA, v. 9, n. 1, p. 11 – 12, jan. 1962. ISSN 0004-5411.

WEST, D. B. **Introduction to Graph Theory**. 2. ed. [s.l.]: Prentice Hall, 2000. ISBN 0130144002.

WOLSEY, L. **Integer Programming**. [s.l.]: John Wiley & Sons, 1998. ISBN 978-0-471-28366-9.

APÊNDICE A – RESULTADOS DETALHADOS PARA FORMULAÇÕES BÁSICAS

Neste apêndice detalhamos os resultados computacionais resumidos e analisados na Seção 6.3.

Experimentos preliminares indicaram que a formulação FE demanda bem mais tempo computacional quando comparada às outras formulações aqui apresentadas. Dito isso, optamos por apresentar os resultados obtidos com essa formulação apenas para um grupo de instâncias, no caso o de pesos mais espaçados. Seu desempenho é similar para as outras ponderações. Além disso, exibimos na Tabela 6 apenas os resultados com a versão FE-MTZ, que se mostrou melhor que a versão FE-SECs.

As colunas ‘Instância’, ‘ $|V(G)|$ ’, ‘ p ’, ‘ t ’, ‘OPT’, ‘Status’, ‘LB’, ‘Tempo’, apresentam, respectivamente, o índice da instância, a quantidade de vértices do grafo, a probabilidade de existência de cada aresta, o fator de dilatação, a solução da instância, solução obtida pela formulação dentro do tempo limite de 1200s, melhor limite inferior, caso o tempo de execução ultrapasse o limite, e tempo de execução em segundos. Na coluna ‘Status’, identificamos por ‘inv’ as instâncias que a formulação solucionou e detectou sua inviabilidade, enquanto o marcador ‘?’ indica que ela não solucionou a instância no tempo limite. Codificação similar aparece na coluna ‘OPT’, considerando todos os métodos avaliados. Para instâncias não solucionadas, apresentamos na coluna ‘LB’ o melhor limitante inferior obtido pela formulação.

Observe que FE-MTZ é incapaz de solucionar algumas instâncias, dentro do limite de tempo, já para grafos com 15 vértices, assim como todas as instâncias a partir de 30 vértices, à exceção da número 20. Com esse experimento observamos a impossibilidade de resolução de instâncias maiores, dentro de um limite cabível de tempo pela formulação FE.

Apresentamos em seguida os resultados detalhados para as formulações CR, SR e AP, bem como para o algoritmo enumerativo. As Tabelas 7, 8, 9 e 10 referem-se, respectivamente, às instâncias com pesos mais espaçados, pesos menos espaçados, distâncias euclidianas e pesos unitários. As colunas ‘Instância’, ‘ $|V(G)|$ ’, ‘ p ’, ‘ t ’, ‘OPT’, ‘Status X’, ‘Tempo X’, apresentam, respectivamente, o índice da instância, a quantidade de vértices do grafo, a probabilidade de existência de cada aresta, o fator de dilatação, a solução da instância, a solução obtida pela formulação X (ou algoritmo enumerativo) dentro do tempo limite de 1200s e tempo de execução do método X em segundos. Na coluna ‘Status’, identificamos por ‘inv’ as instâncias que o método solucionou e detectou sua inviabilidade, enquanto o marcador ‘?’ indica que ele não solucionou

Tabela 6 – Resultados para FE-MTZ - Instâncias com pesos mais espaçados

Instância	$ V(G) $	p	t	OPT	Status	LB	Tempo
1	15	0.2	1.1	inv	inv	-	5
2	15	0.2	2	inv	inv	-	1
3	15	0.2	3	45	45	-	0
4	15	0.2	4	inv	inv	-	7
5	15	0.5	1.1	inv	?	34	1204
6	15	0.5	2	inv	?	30	1203
7	15	0.5	3	25	25	-	0
8	15	0.5	4	19	19	-	1
9	15	0.8	1.1	inv	?	16	1201
10	15	0.8	2	inv	?	19	1206
11	15	0.8	3	inv	?	20	1204
12	15	0.8	4	16	16	-	0
13	15	1	1.1	inv	?	15	1200
14	15	1	2	inv	?	16	1200
15	15	1	3	inv	?	17	1200
16	15	1	4	14	14	-	0
17	30	0.2	1.1	inv	?	70	1200
18	30	0.2	2	inv	?	44	1200
19	30	0.2	3	inv	?	48	1201
20	30	0.2	4	64	64	-	4
21	30	0.5	1.1	inv	?	34	1200
22	30	0.5	2	inv	?	33	1200
23	30	0.5	3	inv	?	30	1200
24	30	0.5	4	inv	?	30	1200
25	30	0.8	1.1	inv	?	30	1200
26	30	0.8	2	inv	?	29	1200
27	30	0.8	3	inv	?	29	1200
28	30	0.8	4	31	?	31	1200
29	30	1	1.1	inv	?	29	1200
30	30	1	2	inv	?	29	1200
31	30	1	3	inv	?	29	1200
32	30	1	4	29	?	29	1200
33	45	0.2	1.1	inv	?	29	1200
34	45	0.2	2	inv	?	29	1200
35	45	0.2	3	inv	?	29	1200
36	45	0.2	4	inv	?	29	1200
37	45	0.5	1.1	inv	?	44	1200
38	45	0.5	2	inv	?	46	1200
39	45	0.5	3	inv	?	44	1200
40	45	0.5	4	inv	?	44	1200
41	45	0.8	1.1	inv	?	44	1200
42	45	0.8	2	inv	?	44	1200
43	45	0.8	3	inv	?	44	1200
44	45	0.8	4	?	?	44	1200
45	45	1	1.1	inv	?	44	1200
46	45	1	2	inv	?	44	1200
47	45	1	3	inv	?	44	1200
48	45	1	4	?	?	44	1200
49	60	0.2	1.1	inv	?	68	1200
50	60	0.2	2	inv	?	70	1200
51	60	0.2	3	inv	?	72	1200
52	60	0.2	4	inv	?	63	1200
53	60	0.5	1.1	inv	?	60	1200
54	60	0.5	2	inv	?	59	1200
55	60	0.5	3	inv	?	60	1200
56	60	0.5	4	?	?	59	1200
57	60	0.8	1.1	inv	?	59	1200
58	60	0.8	2	inv	?	59	1200
59	60	0.8	3	inv	?	59	1200
60	60	0.8	4	?	?	59	1200
61	60	1	1.1	inv	?	59	1200
62	60	1	2	inv	?	59	1200
63	60	1	3	inv	?	59	1200
64	60	1	4	?	?	59	1200
Média							1031.83

Fonte: Elaborado pelo autor.

a instância no tempo limite. Codificação similar aparece na coluna ‘OPT’, considerando todos os métodos avaliados. Os melhores resultados estão destacados em negrito.

Tabela 7 – Resultados para CR-SR-AP-AlgE - Instâncias com pesos menos espaçados

Instância	$V(G)$	p	t	OPT	Tempo CR	Status CR	Tempo SR	Status SR	Tempo AP	Status AP	Tempo AlgE	Status AlgE
1	15	0,2	1,1	inv	0	inv	0	inv	0	inv	0	inv
2	15	0,2	2	inv	0	inv	0	inv	0	inv	0	inv
3	15	0,2	3	inv	0	inv	1	inv	1	inv	0	inv
4	15	0,2	4	30	0	30	0	30	0	30	0	30
5	15	0,5	1,1	inv	0	inv	0	inv	0	inv	0	inv
6	15	0,5	2	inv	0	inv	0	inv	0	inv	0	inv
7	15	0,5	3	21	0	21	0	21	0	21	0	21
8	15	0,5	4	16	0	16	0	16	0	16	0	16
9	15	0,8	1,1	inv	0	inv	0	inv	0	inv	0	inv
10	15	0,8	2	inv	0	inv	1	inv	0	inv	0	inv
11	15	0,8	3	inv	0	inv	0	inv	1	inv	1	inv
12	15	0,8	4	14	0	14	1	14	3	14	5	14
13	15	1	1,1	inv	0	inv	0	inv	0	inv	0	inv
14	15	1	2	inv	0	inv	1	inv	1	inv	0	inv
15	15	1	3	inv	0	inv	34	inv	21	inv	17	inv
16	15	1	4	15	0	15	2	15	3	15	35	15
17	30	0,2	1,1	inv	0	inv	1	inv	0	inv	0	inv
18	30	0,2	2	inv	0	inv	1	inv	0	inv	0	inv
19	30	0,2	3	inv	1	inv	1	inv	6	inv	0	inv
20	30	0,2	4	inv	0	inv	1020	inv	1200	?	1200	?
21	30	0,5	1,1	inv	0	inv	1	inv	3	inv	0	inv
22	30	0,5	2	inv	0	inv	1	inv	47	inv	0	inv
23	30	0,5	3	inv	0	inv	1062	inv	1200	?	38	inv
24	30	0,5	4	inv	1	inv	1200	?	1200	?	1200	?
25	30	0,8	1,1	inv	0	inv	2	inv	6	inv	0	inv
26	30	0,8	2	inv	0	inv	432	inv	217	inv	0	inv
27	30	0,8	3	inv	1	inv	1201	?	1200	?	1200	?
28	30	0,8	4	29	880	29	1200	?	1200	?	1200	?
29	30	1	1,1	inv	0	inv	2	inv	1	inv	0	inv
30	30	1	2	inv	0	inv	11	inv	407	inv	14	inv
31	30	1	3	inv	1	inv	1201	?	1200	?	1200	?
32	30	1	4	29	1152	29	1200	?	1200	?	1200	?
33	45	0,2	1,1	inv	0	inv	2	inv	0	inv	0	inv
34	45	0,2	2	inv	0	inv	6	inv	1	inv	0	inv
35	45	0,2	3	inv	0	inv	8	inv	1200	?	1	inv
36	45	0,2	4	inv	1	inv	1201	?	1200	?	776	inv
37	45	0,5	1,1	inv	0	inv	5	inv	1	inv	0	inv
38	45	0,5	2	inv	0	inv	15	inv	5	inv	0	inv
39	45	0,5	3	inv	2	inv	1202	?	1200	?	1200	?
40	45	0,5	4	?	1200	?	1202	?	1200	?	1200	?
41	45	0,8	1,1	inv	1	inv	8	inv	2	inv	0	inv
42	45	0,8	2	inv	0	inv	1203	?	1200	?	110	inv
43	45	0,8	3	inv	5	inv	1200	?	1200	?	1200	?
44	45	0,8	4	?	1200	?	1201	?	1201	?	1200	?
45	45	1	1,1	inv	0	inv	7	inv	2	inv	0	inv
46	45	1	2	inv	0	inv	1201	?	1200	?	1200	?
47	45	1	3	inv	7	inv	1200	?	1201	?	1200	?
48	45	1	4	*	1200	*	1201	?	1200	?	1200	?
49	60	0,2	1,1	inv	0	inv	9	inv	1	inv	0	inv
50	60	0,2	2	inv	0	inv	30	inv	4	inv	0	inv
51	60	0,2	3	inv	1	inv	44	inv	1200	?	2	inv
52	60	0,2	4	inv	3	inv	1204	?	1200	?	1200	?
53	60	0,5	1,1	inv	1	inv	15	inv	3	inv	0	inv
54	60	0,5	2	inv	0	inv	89	inv	1200	?	32	inv
55	60	0,5	3	inv	21	inv	1200	?	1201	?	1200	?
56	60	0,5	4	?	1200	?	1201	?	1203	?	1200	?
57	60	0,8	1,1	inv	0	inv	28	inv	4	inv	0	inv
58	60	0,8	2	inv	1	inv	1201	?	1201	?	1200	?
59	60	0,8	3	inv	24	inv	1200	?	1202	?	1200	?
60	60	0,8	4	?	1201	?	1200	?	1201	?	1200	?
61	60	1	1,1	inv	0	inv	36	inv	7	inv	0	inv
62	60	1	2	inv	1	inv	1202	?	1201	?	1200	?
63	60	1	3	?	1200	?	1201	?	1204	?	1200	?
64	60	1	4	?	1200	?	1200	?	1202	?	1200	?
Média					164,140625		476,53125		536,9375		428,609375	

*: Instância viável, Melhor limite inferior 44, com GAP de 0.266667

Fonte: Elaborado pelo autor.

Tabela 8 – Resultados para CR-SR-AP-AlgE - Instâncias com pesos mais espaçados

Instância	$ V(G) $	p	t	OPT	Tempo CR	Status CR	Tempo SR	Status SR	Tempo AP	Status AP	Tempo AlgE	Status AlgE
1	15	0,2	1,1	inv	0	inv	0	inv	0	inv	0	inv
2	15	0,2	2	inv	0	inv	0	inv	0	inv	0	inv
3	15	0,2	3	45	0	45	0	45	0	45	0	45
4	15	0,2	4	inv	0	inv	0	inv	0	inv	0	inv
5	15	0,5	1,1	inv	0	inv	0	inv	0	inv	0	inv
6	15	0,5	2	inv	0	inv	0	inv	0	inv	0	inv
7	15	0,5	3	25	0	25	0	25	0	25	0	25
8	15	0,5	4	19	0	19	0	19	0	19	0	19
9	15	0,8	1,1	inv	0	inv	0	inv	0	inv	0	inv
10	15	0,8	2	inv	0	inv	1	inv	0	inv	0	inv
11	15	0,8	3	inv	1	inv	0	inv	0	inv	0	inv
12	15	0,8	4	16	0	16	0	16	1	16	0	16
13	15	1	1,1	inv	0	inv	0	inv	0	inv	0	inv
14	15	1	2	inv	0	inv	0	inv	0	inv	0	inv
15	15	1	3	inv	0	inv	10	inv	6	inv	3	inv
16	15	1	4	14	0	14	2	14	3	14	0	14
17	30	0,2	1,1	inv	0	inv	1	inv	0	inv	0	inv
18	30	0,2	2	inv	0	inv	1	inv	0	inv	0	inv
19	30	0,2	3	inv	0	inv	1	inv	1	inv	0	inv
20	30	0,2	4	64	0	64	2	64	1	64	0	64
21	30	0,5	1,1	inv	0	inv	0	inv	0	inv	0	inv
22	30	0,5	2	inv	0	inv	2	inv	1	inv	0	inv
23	30	0,5	3	inv	0	inv	462	inv	945	inv	1	inv
24	30	0,5	4	inv	2	inv	1200	?	1200	?	1200	?
25	30	0,8	1,1	inv	0	inv	1	inv	6	inv	0	inv
26	30	0,8	2	inv	0	inv	4	inv	193	inv	0	inv
27	30	0,8	3	inv	1	inv	1200	?	1200	?	1200	?
28	30	0,8	4	31	82	31	350	31	1200	?	1200	?
29	30	1	1,1	inv	0	inv	1	inv	6	inv	0	inv
30	30	1	2	inv	0	inv	6	inv	25	inv	0	inv
31	30	1	3	inv	1	inv	1201	?	1200	?	1200	?
32	30	1	4	29	754	29	916	29	1200	?	1200	?
33	45	0,2	1,1	inv	0	inv	2	inv	0	inv	0	inv
34	45	0,2	2	inv	0	inv	9	inv	1	inv	0	inv
35	45	0,2	3	inv	0	inv	5	inv	26	inv	0	inv
36	45	0,2	4	inv	3	inv	1202	?	1200	?	5	inv
37	45	0,5	1,1	inv	0	inv	4	inv	0	inv	1	inv
38	45	0,5	2	inv	0	inv	11	inv	593	inv	0	inv
39	45	0,5	3	inv	1	inv	1202	?	1200	?	1200	?
40	45	0,5	4	inv	2	inv	1202	?	1200	?	1200	?
41	45	0,8	1,1	inv	0	inv	8	inv	1	inv	0	inv
42	45	0,8	2	inv	0	inv	33	inv	1201	?	11	inv
43	45	0,8	3	inv	3	inv	1203	?	1200	?	1200	?
44	45	0,8	4	?	1200	?	1203	?	1201	?	1200	?
45	45	1	1,1	inv	0	inv	10	inv	1	inv	0	inv
46	45	1	2	inv	0	inv	1204	?	1201	?	82	inv
47	45	1	3	inv	29	inv	1205	?	1200	?	1200	?
48	45	1	4	?	1201	?	1205	?	1201	?	1200	?
49	60	0,2	1,1	inv	0	inv	10	inv	0	inv	0	inv
50	60	0,2	2	inv	1	inv	50	inv	3	inv	0	inv
51	60	0,2	3	inv	0	inv	26	inv	1200	?	3	inv
52	60	0,2	4	inv	1	inv	1205	?	1200	?	1200	?
53	60	0,5	1,1	inv	1	inv	15	inv	2	inv	0	inv
54	60	0,5	2	inv	0	inv	58	inv	1201	?	0	inv
55	60	0,5	3	inv	2	inv	1200	?	1200	?	1200	?
56	60	0,5	4	?	1200	?	1200	?	1200	?	1200	?
57	60	0,8	1,1	inv	1	inv	26	inv	4	inv	0	inv
58	60	0,8	2	inv	0	inv	53	inv	1201	?	17	inv
59	60	0,8	3	inv	21	inv	1200	?	1200	?	1200	?
60	60	0,8	4	?	1200	?	1200	?	1200	?	1200	?
61	60	1	1,1	inv	1	inv	1200	?	7	inv	0	inv
62	60	1	2	inv	0	inv	1200	?	1201	?	1200	?
63	60	1	3	inv	399	inv	1200	?	1200	?	1200	?
64	60	1	4	?	1200	?	1200	?	1200	?	1200	?
Média					114,1875		408		497,390625		358,171875	

Fonte: Elaborado pelo autor.

Tabela 9 – Resultados para CR-SR-AP-AlgE - Instâncias com distância Euclidiana

Instância	$V(G)$	p	t	OPT	Tempo CR	Status CR	Tempo SR	Status SR	Tempo AP	Status AP	Tempo AlgE	Status AlgE
1	15	0,2	1,1	inv	0	inv	0	inv	0	inv	0	inv
2	15	0,2	2	inv	0	inv	0	inv	0	inv	0	inv
3	15	0,2	3	inv	0	inv	0	inv	0	inv	0	inv
4	15	0,2	4	495,309	0	495,309	0	495,309	1	495,309	0	495,309
5	15	0,5	1,1	inv	0	inv	0	inv	0	inv	0	inv
6	15	0,5	2	inv	0	inv	0	inv	0	inv	0	inv
7	15	0,5	3	inv	0	inv	1	inv	0	inv	0	inv
8	15	0,5	4	359,487	0	359,487	1	359,487	1	359,487	11	359,487
9	15	0,8	1,1	inv	0	inv	0	inv	0	inv	0	inv
10	15	0,8	2	inv	0	inv	0	inv	0	inv	0	inv
11	15	0,8	3	inv	1	inv	1	inv	3	inv	2	inv
12	15	0,8	4	294,783	0	294,783	0	294,783	0	294,783	0	294,783
13	15	1	1,1	inv	0	inv	0	inv	0	inv	0	inv
14	15	1	2	inv	0	inv	0	inv	0	inv	0	inv
15	15	1	3	244,523	0	244,523	1	244,523	3	244,523	146	244,523
16	15	1	4	241,977	0	241,977	1	241,977	0	241,977	0	241,977
17	30	0,2	1,1	inv	0	inv	1	inv	0	inv	0	inv
18	30	0,2	2	inv	0	inv	1	inv	0	inv	0	inv
19	30	0,2	3	inv	0	inv	1	inv	1	inv	1	inv
20	30	0,2	4	946,256	1	946,256	5	946,256	6	946,256	67	946,256
21	30	0,5	1,1	inv	0	inv	1	inv	0	inv	0	inv
22	30	0,5	2	inv	0	inv	1	inv	0	inv	0	inv
23	30	0,5	3	inv	0	inv	4	inv	4	inv	1200	?
24	30	0,5	4	inv	4	inv	1200	?	1200	?	1200	?
25	30	0,8	1,1	inv	0	inv	1	inv	0	inv	0	inv
26	30	0,8	2	inv	0	inv	3	inv	1	inv	0	inv
27	30	0,8	3	inv	2	inv	54	inv	121	inv	1200	?
28	30	0,8	4	inv	27	inv	1200	?	1200	?	1200	?
29	30	1	1,1	inv	0	inv	1	inv	0	inv	0	inv
30	30	1	2	inv	0	inv	4	inv	2	inv	0	inv
31	30	1	3	inv	1	inv	846	inv	1200	?	1200	?
32	30	1	4	inv	109	inv	1201	?	1200	?	1200	?
33	45	0,2	1,1	inv	0	inv	2	inv	0	inv	0	inv
34	45	0,2	2	inv	0	inv	3	inv	1	inv	0	inv
35	45	0,2	3	inv	1	inv	10	inv	4	inv	1	inv
36	45	0,2	4	1183,59	2	1183,59	1202	*	1200	**	1200	?
37	45	0,5	1,1	inv	0	inv	2	inv	0	inv	0	inv
38	45	0,5	2	inv	0	inv	9	inv	2	inv	2	inv
39	45	0,5	3	inv	1	inv	29	inv	1129	inv	1200	?
40	45	0,5	4	inv	15	inv	1202	?	1201	?	1200	?
41	45	0,8	1,1	inv	1	inv	5	inv	0	inv	0	inv
42	45	0,8	2	inv	0	inv	14	inv	14	inv	3	inv
43	45	0,8	3	inv	4	inv	67	inv	1200	?	1200	?
44	45	0,8	4	?	1200	?	1200	?	1200	?	1200	?
45	45	1	1,1	inv	1	inv	6	inv	1	inv	0	inv
46	45	1	2	inv	0	inv	24	inv	24	inv	0	inv
47	45	1	3	inv	25	inv	1200	?	1202	?	1200	?
48	45	1	4	?	1200	?	1200	?	1202	?	1200	?
49	60	0,2	1,1	inv	1	inv	6	inv	0	inv	0	inv
50	60	0,2	2	inv	0	inv	20	inv	2	inv	0	inv
51	60	0,2	3	inv	0	inv	63	inv	1049	inv	1200	?
52	60	0,2	4	inv	10	inv	1200	?	1200	?	1200	?
53	60	0,5	1,1	inv	1	inv	9	inv	1	inv	0	inv
54	60	0,5	2	inv	0	inv	28	inv	22	inv	0	inv
55	60	0,5	3	inv	3	inv	1200	?	1201	?	1200	?
56	60	0,5	4	?	1200	?	1200	?	1200	?	1200	?
57	60	0,8	1,1	inv	1	inv	14	inv	2	inv	0	inv
58	60	0,8	2	inv	0	inv	41	inv	70	inv	0	inv
59	60	0,8	3	inv	17	inv	1200	?	1201	?	1200	?
60	60	0,8	4	?	1200	?	1200	?	1201	?	1200	?
61	60	1	1,1	inv	1	inv	17	inv	2	inv	0	inv
62	60	1	2	inv	0	inv	1200	?	126	inv	1200	?
63	60	1	3	inv	576	inv	1200	?	1201	?	1200	?
64	60	1	4	?	1200	?	1200	?	1202	?	1200	?
Média					106,34375		320,34375		359,421875		416,140625	

*GAP=0.0307771 LB=1147.17

**GAP=0.0285734 LB=1149.77

Fonte: Elaborado pelo autor.

Tabela 10 – Resultados para CR-SR-AP-AlgE - Instâncias com pesos unitários

Instância	$V(G)$	p	t	OPT	Tempo CR	Status CR	Tempo SR	Status SR	Tempo AP	Status AP	Tempo AlgE	Status AlgE
1	15	0,2	1,1	inv	0	inv	0	inv	1	inv	0	inv
2	15	0,2	2	inv	0	inv	0	inv	0	inv	0	inv
3	15	0,2	3	inv	1	inv	0	inv	0	inv	0	inv
4	15	0,2	4	inv	0	inv	1	inv	0	inv	0	inv
5	15	0,5	1,1	inv	0	inv	0	inv	0	inv	0	inv
6	15	0,5	2	inv	0	inv	2	inv	8	inv	1	inv
7	15	0,5	3	inv	0	inv	52	inv	876	inv	1105	inv
8	15	0,5	4	14	0	14	1	14	0	14	0	14
9	15	0,8	1,1	inv	0	inv	0	inv	0	inv	0	inv
10	15	0,8	2	14	0	14	2	14	31	14	1200	?
11	15	0,8	3	14	3	14	1	14	0	14	0	14
12	15	0,8	4	14	0	14	0	14	0	14	0	14
13	15	1	1,1	inv	0	inv	0	inv	0	inv	0	inv
14	15	1	2	14	1	14	3	14	0	14	0	14
15	15	1	3	14	1	14	0	14	0	14	0	14
16	15	1	4	14	0	14	0	14	0	14	0	14
17	30	0,2	1,1	inv	0	inv	1	inv	3	inv	0	inv
18	30	0,2	2	inv	0	inv	1	inv	44	inv	0	inv
19	30	0,2	3	inv	0	inv	1200	?	1200	?	1200	?
20	30	0,2	4	inv	742	inv	1200	?	1200	?	1200	?
21	30	0,5	1,1	inv	0	inv	1	inv	0	inv	0	inv
22	30	0,5	2	inv	1	inv	1201	?	1200	?	1200	?
23	30	0,5	3	inv	872	inv	1200	?	1200	?	1200	?
24	30	0,5	4	29	58	29	48	29	1	29	0	29
25	30	0,8	1,1	inv	0	inv	2	inv	3	inv	0	inv
26	30	0,8	2	inv	0	inv	1200	?	1200	?	1200	?
27	30	0,8	3	29	127	29	430	29	1201	?	1200	?
28	30	0,8	4	29	84	29	197	29	92	29	0	29
29	30	1	1,1	inv	0	inv	2	inv	2	inv	0	inv
30	30	1	2	29	8	29	1201	?	1	29	0	29
31	30	1	3	29	601	29	12	29	2	29	0	29
32	30	1	4	29	563	29	11	29	2	29	0	29
33	45	0,2	1,1	inv	0	inv	4	inv	1	inv	0	inv
34	45	0,2	2	inv	0	inv	1200	?	1200	?	1	inv
35	45	0,2	3	inv	1	inv	1200	?	1200	?	1200	?
36	45	0,2	4	?	1200	?	1201	?	1200	?	1200	?
37	45	0,5	1,1	inv	0	inv	5	inv	10	inv	0	inv
38	45	0,5	2	inv	1	inv	1202	?	1200	?	1200	?
39	45	0,5	3	?	1200	?	1200	?	1201	?	1200	?
40	45	0,5	4	44	566	44	1201	?	1200	?	1200	?
41	45	0,8	1,1	inv	1	inv	11	inv	6	inv	0	inv
42	45	0,8	2	inv	1	inv	1200	?	1201	?	1200	?
43	45	0,8	3	?	1200	?	1203	?	1200	?	1200	?
44	45	0,8	4	44	1200	?	1201	?	1200	?	0	44
45	45	1	1,1	inv	1	inv	20	inv	20	inv	0	inv
46	45	1	2	44	1200	?	1201	?	8	44	0	44
47	45	1	3	44	1200	?	1200	?	9	44	0	44
48	45	1	4	44	1201	?	1200	?	8	44	0	44
49	60	0,2	1,1	inv	1	inv	12	inv	2	inv	0	inv
50	60	0,2	2	inv	0	inv	1200	?	1200	?	1200	?
51	60	0,2	3	inv	8	inv	1200	?	1200	?	1200	?
52	60	0,2	4	?	1200	?	1200	?	1201	?	1200	?
53	60	0,5	1,1	inv	1	inv	20	inv	22	inv	0	inv
54	60	0,5	2	inv	2	inv	1203	?	1202	?	1200	?
55	60	0,5	3	?	1200	?	1201	?	1203	?	1200	?
56	60	0,5	4	?	1200	?	1200	?	1201	?	1200	?
57	60	0,8	1,1	inv	1	inv	36	inv	1200	?	0	inv
58	60	0,8	2	inv	24	inv	1200	?	1203	?	1200	?
59	60	0,8	3	?	1201	?	1201	?	1203	?	1200	?
60	60	0,8	4	59	1200	?	1200	?	1203	?	0	59
61	60	1	1,1	inv	1	inv	1200	?	126	inv	0	inv
62	60	1	2	59	1201	?	1201	?	1202	?	0	59
63	60	1	3	59	1201	?	1201	?	1201	?	0	59
64	60	1	4	59	1201	?	1201	?	1203	?	0	59
média					338,6875		595,21875		545,359375		429,796875	

Fonte: Elaborado pelo autor.

APÊNDICE B – RESULTADOS DETALHADOS PARA VARIANTES DE SR

Neste apêndice, detalhamos os resultados com as duas variantes da formulação (SR) apresentadas na Subseção 6.4.1.

Na Tabela 11, as colunas ‘Instância’, ‘ $|V(G)|$ ’, ‘ p ’, ‘ t ’, ‘OPT’, ‘Status SR’, ‘Tempo SR’, apresentam, respectivamente, o índice da instância, a quantidade de vértices do grafo, a probabilidade de existência de cada aresta, o fator de dilatação, a solução da instância, a solução obtida pela formulação (SR) dentro do tempo limite de 1200s e seu tempo de execução. Na coluna ‘Status’, identificamos por ‘inv’ as instâncias que o método solucionou e detectou sua inviabilidade, enquanto o marcador ‘?’ indica que ele não solucionou a instância no tempo limite. Codificação similar aparece na coluna ‘OPT’, considerando todos os métodos avaliados. Caso o nome da formulação esteja identificado por ‘+ESP’ ou ‘Sem Y’, significa, respectivamente, que as desigualdades (5.35)-(5.37) foram adicionadas à formulação e priorizada a ramificação nas variáveis x , ou feita realizada a remoção das variáveis y do modelo, bem como as restrições em que elas aparecem, e a adição das desigualdades válidas t -spanner arborescentes. Os melhores resultados estão destacados em negrito.

Tabela 11 – Resultados para variantes de (SR) para instâncias com peso unitário

Instância	$ V(G) $	p	t	OPT	Tempo SR	Status SR	Tempo SR+ESP	Status SR+ESP	Tempo SR Sem Y	Status SR Sem Y
19	30	0,2	3	inv	1200	?	55	inv	6	inv
20	30	0,2	4	inv	1200	?	1201	?	26	inv
23	30	0,5	3	inv	1200	?	1200	?	23	inv
24	30	0,5	4	29	48	29	14	29	26	29
27	30	0,8	3	29	430	29	1203	?	61	29
28	30	0,8	4	29	197	29	18	29	64	29
31	30	1	3	29	12	29	14	29	1200	?
32	30	1	4	29	11	29	24	29	1200	?
35	45	0,2	3	inv	1200	?	1201	?	495	inv
36	45	0,2	4	?	1200	?	1201	?	1200	?
39	45	0,5	3	?	1200	?	1200	?	1200	?
40	45	0,5	4	44	1201	?	1201	?	1200	?
43	45	0,8	3	?	1203	?	1200	?	1201	?
44	45	0,8	4	44	1201	?	1201	?	774	44
47	45	1	3	44	1200	?	1202	?	1200	?
48	45	1	4	44	1200	?	1200	?	1200	?
51	60	0,2	3	inv	1200	?	1201	?	3	inv
52	60	0,2	4	?	1200	?	1200	?	1200	?
55	60	0,5	3	?	1201	?	1201	?	1200	?
56	60	0,5	4	?	1200	?	1200	?	1200	?
59	60	0,8	3	?	1201	?	1201	?	1200	?
60	60	0,8	4	59	1200	?	1200	?	1200	?
63	60	1	3	59	1201	?	1201	?	164	59
64	60	1	4	59	1201	?	1201	?	1200	?
Média					979,5		955,83		768,46	

Fonte: Elaborado pelo autor.

APÊNDICE C – RESULTADOS DETALHADOS PARA VARIANTES DE CR

Neste apêndice, detalhamos os resultados com as três variantes da formulação CR apresentadas na Subseção 6.4.2, cada uma delas obtida pela inclusão de um tipo de desigualdade válida: t -spanner arborescente, ciclo t -inviável ou potencial mínimo.

Na Tabela 12, as colunas ‘Instância’, ‘ $|V(G)|$ ’, ‘ p ’, ‘ t ’, ‘OPT’, ‘Status CR’, ‘Tempo CR’, apresentam, respectivamente, o índice da instância, a quantidade de vértices do grafo, a probabilidade de existência de cada aresta, o fator de dilatação, a solução da instância, a solução obtida pela formulação CR dentro do tempo limite de 1200s e seu tempo de execução. Na coluna ‘Status’, identificamos por ‘inv’ as instâncias que o método solucionou e detectou sua inviabilidade, enquanto o marcador ‘?’ indica que ele não solucionou a instância no tempo limite. Codificação similar aparece na coluna ‘OPT’, considerando todos os métodos avaliados. Quando o nome da formulação (CR) está modificado por ‘+tSA’, ‘+CtI’ ou ‘+PM’, as desigualdades t -spanner arborescente, ciclo t -inviável ou potencial mínimo foram adicionadas ao modelo, respectivamente. Os melhores resultados estão destacados em negrito.

Tabela 12 – Resultados para CR com adição de desigualdades válidas para instâncias com peso unitário

Instância	$ V(G) $	p	t	OPT	Tempo CR	Status CR	Tempo CR+tSA	Status CR+tSA	Tempo CR+CtI	Status CR+CtI	Tempo CR+PM	Status CR+PM
19	30	0,2	3	inv	0	inv	0	inv	3	inv	3	inv
20	30	0,2	4	inv	742	inv	1200	?	258	inv	295	inv
23	30	0,5	3	inv	872	inv	1200	?	1200	?	904	inv
24	30	0,5	4	29	58	29	1200	?	192	29	194	29
27	30	0,8	3	29	127	29	30	29	246	29	277	29
28	30	0,8	4	29	84	29	13	29	13	29	15	29
31	30	1	3	29	601	29	1200	?	33	29	33	29
32	30	1	4	29	563	29	832	29	25	29	25	29
35	45	0,2	3	inv	1	inv	0	inv	2	inv	1	inv
36	45	0,2	4	?	1200	?	1200	?	1200	?	1200	?
39	45	0,5	3	?	1200	?	1201	?	1201	?	1200	?
40	45	0,5	4	44	566	44	1200	?	1200	?	1200	?
43	45	0,8	3	?	1200	?	1200	?	1200	?	1200	?
44	45	0,8	4	44	1200	?	1200	?	1201	?	1200	?
47	45	1	3	44	1200	?	1200	?	1200	?	1201	?
48	45	1	4	44	1201	?	1201	?	1200	?	1200	?
51	60	0,2	3	inv	8	inv	2	inv	5	inv	537	inv
52	60	0,2	4	?	1200	?	1200	?	1201	?	1200	?
55	60	0,5	3	?	1200	?	1201	?	1212	?	1209	?
56	60	0,5	4	?	1200	?	1209	?	1212	?	1210	?
59	60	0,8	3	?	1201	?	1200	?	1200	?	1201	?
60	60	0,8	4	59	1200	?	1201	?	1201	?	1200	?
63	60	1	3	59	1201	?	1200	?	1201	?	1206	?
64	60	1	4	59	1201	?	104	59	103	59	103	59
Média					801,08		891,42		737,87		750,58	

Fonte: Elaborado pelo autor.