São José dos Campos - SP
LV SBPO
LV Simpósio Brasileiro de Pesquisa Operacional
São José dos Campos, SP - 6 a 9 de novembro de 2023

Formulações matemáticas para o problema da árvore t-spanner de custo mínimo*

Manoel Campêlo
ParGO, Departamento de Estatística e Matemática Aplicada, Universidade Federal do Ceará
Av. Humberto Monte, s/n, Campus do Pici, Bloco 910. CEP 60440-900. Fortaleza, Ceará, Brasil
mcampelo@ufc.br

Gabriel H. de Sousa
ParGO, Departamento de Computação, Universidade Federal do Ceará
Av. Humberto Monte, s/n, Campus do Pici, Bloco 910. CEP 60440-900. Fortaleza, Ceará, Brasil
gabrielsousa.9957@gmail.com

RESUMO

No Problema da Árvore t-Spanner de Custo Mínimo, cuja entrada é um grafo $G = (V, E)$ ponderado e um parâmetro $t \geq 1$, 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. Neste trabalho, estudamos e propomos aperfeiçoamentos para duas formulações matemáticas da literatura. Implementamos ambos os modelos e comparamos seus desempenhos computacionais com outros métodos exatos da literatura.

PALAVRAS CHAVE. Árvore t-Spanner de Custo Mínimo, Árvore Geradora, Formulação Matemática.

Tópicos: Programação Matemática, Otimização Combinatória, Teoria e Algoritmos em Grafos.

ABSTRACT

In the Minimum Weight t-Spanner Tree Problem, whose input is a weighted graph $G = (V, E)$ and a parameter $t \geq 1$, 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 T is at most t times the distance between i and j in G. In this work, we study and propose improvements to two literature mathematical formulations. We implemented both models and compared their computational performances with other exact methods in the literature.

KEYWORDS. Minimum Weight t-Spanner Tree. Spanning Tree. Mathematical Formulation.

Topics: Mathematical Programming, Combinatorial Optimization, Theory and Algorithms in Graphs.

*O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001, FUNCAP PS1-0186-00155.01.00/21, FUNCAP PNE 0112.00061.01.00-16, FUNCAP MLC-0191-00056.01.00/22 e CNPQ Proc. 309315/2019-0.

DOI: 10.59254/sbpo-2023-175148
http://live-proceedings.pantheonsite.io/p/175148

1. Introdução

Seja $G = (V, E)$ um grafo não direcionado, no qual V é seu conjunto de vértices e E seu conjunto de arestas ponderadas. Diz-se que $H = (\bar{V}, \bar{E})$ é um subgrafo de G se seus conjuntos de vértices e arestas são subconjuntos dos vértices e arestas de G. Caso $V = \bar{V}$, diz-se que H é um subgrafo gerador. Se conexo e acíclico, H é dito uma árvore geradora.

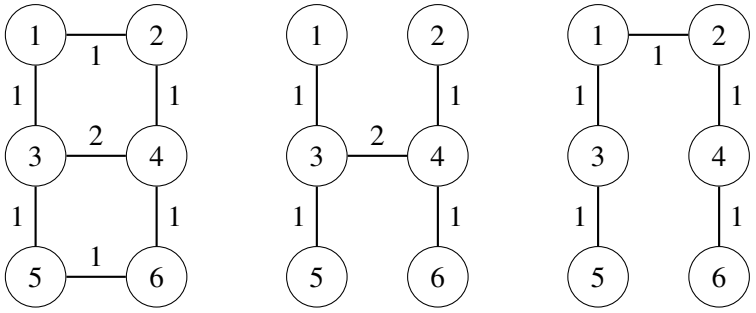
Problemas sobre subgrafos geradores com determinadas características aparecem em vários cenários, dentre eles: computação distribuída [Peleg e Ullman, 1989], redes de comunicação [Peleg e Reshef, 1999; Oliveira e Pardalos, 2005], localização e distribuição [Goldschmidt et al., 1996], robótica [Arikati et al., 1996; Marble e Bekris, 2011], bioinformática [Nguyen et al., 2008], entre outros. Aqui consideramos um problema de subgrafo gerador com um controle sobre o aumento das distâncias entre os vértices.

Dados um grafo não direcionado $G = (V, E)$, ponderado em arestas, e um real $t \geq 1$, um t-spanner de G é um subgrafo gerador H tal que, para todo par $\{i, j\}$, com $i, j \in V$, a distância (peso de um caminho mínimo) 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:

$$\text{dist}_H(i, j) \leq t * \text{dist}_G(i, j), \qquad \forall i, j \in V \tag{1}$$

no qual $\text{dist}(i, j)$ é o peso de um caminho mínimo entre i e j no grafo considerado. Se for uma árvore, além de subgrafo gerador, dizemos que H é uma árvore t-spanner. Ilustramos na Figura 1, da esquerda para a direita, um grafo ponderado G, uma árvore t-spanner e uma árvore geradora que não é um t-spanner, para $t = 4$.

Figura 1: Grafo G, $t = 4$. Número próximo a aresta é seu peso.



t-Spanners são estruturas que aparecem, por exemplo, em problemas de redes nos quais se objetiva simultaneamente rapidez de computação/comunicação e economia de recursos da rede, mantendo um certo compromisso entre esses critérios. Mais especificamente, considera-se que, numa rede de comunicação, deseja-se determinar um esquema de roteamento por caminhos par-a-par os mais curtos possíveis, sujeito à condição de que a demanda de memória local para armazenar as informações não seja excessiva (segundo um certo parâmetro). Nesse contexto, 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 [Peleg e Upfal, 1988].

Neste trabalho, tratamos o Problema da Árvore t-spanner de Custo Mínimo (AtSCM). Tomando como entrada um grafo não orientado G, ponderado, e um fator de dilatação $t \geq 1$, o problema consiste em determinar, dentre todas as árvores t-spanner, a de menor custo, ou provar que o grafo não admite uma árvore t-spanner.

DOI: 10.59254/sbpo-2023-175148
http://live-proceedings.pantheonsite.io/p/175148

Sabe-se que o AtSCM é computacionalmente difícil. Para $t > 1$ fixo, Cai e Coneil [1995] mostram que verificar se existe uma árvore t-spanner em um grafo ponderado é um problema NP-completo. Para o caso em que $t = 1$, uma árvore t-spanner, caso exista, é, em particular, uma árvore geradora mínima, e pode ser encontrada em $O(m \log \beta(m, n))$, onde $m = |E|$, $n = |V|$ e $\beta(m, n) = \min\{i | \log^{(i)} n \leq \frac{m}{n}\}$ [Cai e Coneil, 1995]. Quando os pesos são unitários, temos um problema de decisão, visto que toda árvore geradora, em particular uma árvore t-spanner, terá peso $|V| - 1$. Nesse caso, para $t \in \{1, 2\}$ existem algoritmos polinomiais que provam se G admite ou não uma árvore t-spanner [Cai, 1992; Cai e Coneil, 1995]. Para $t = 3$, sabe-se que o problema é polinomial para algumas classes de grafos [Cai, 1992; Madanlal et al., 1996; Fekete e Kremer, 2001; Venkatesan et al., 1997; Papoutsakis, 2018; Brandstadt et al., 2004], mas sua complexidade ainda está em aberto para grafos gerais. Para $t \geq 4$, o problema se torna NP-completo [Brandstadt et al., 2004; Dragan et al., 2011].

O AtSCM já tem sido abordado por métodos de otimização matemática. Considerando heurísticas, existe um algoritmo de colônia de abelhas proposto por Singh e Sundar [2018]. Com respeito a métodos exatos, há um algoritmo enumerativo e três formulações de programação linear inteira [Braga, 2019; Campelo e Sousa, 2021, 2022]. Neste trabalho, iremos propor aperfeiçoamentos para as duas formulações propostas por Braga [2019] (denominadas (SR) e (CR)). Iremos analisar seus desempenhos computacionais e comparar com o algoritmo enumerativo [Campelo e Sousa, 2021] e a outra formulação da literatura [Campelo e Sousa, 2022] (formulação (AP)). Na seção a seguir descrevemos formalmente as formulações.

2. Formulações Matemáticas

Nesta seção, serão propostos aperfeiçoamentos para as duas formulações da literatura, aqui denominadas formulação sem rótulos (SR) e formulação com rótulos (CR). Ambas foram propostas para o AtSCM por Braga [2019].

A partir da orientação simétrica de $G = (V, E)$, na qual cada aresta $\{i, j\}$ dá origem aos arcos $i \to j$ e $j \to i$, $|V|$ arborescências que se sobrepõem em G são construídas por cada formulação, cada uma enraizada em um nó diferente do grafo. Lembramos que uma arborescência é uma árvore orientada, na qual todo vértice tem grau de entrada igual a 1, exceto um deles, chamado raiz, que possui grau de entrada zero. Aqui, o termo sobrepor significa que há uma mesma árvore geradora de G de quem cada arborescência é uma orientação diferente, ou seja, as arestas de G associadas aos arcos de cada arborescência são as mesmas. Assim, todas essas arborescências representam uma mesma árvore geradora do grafo. Na Figura 2, ilustramos 4 arborescências diferentes, cada uma enraizada em um nó do grafo (destacado em preto), porém todas elas associadas à mesma árvore geradora.

Para a representação dessas arborescências, sejam as seguintes variáveis:

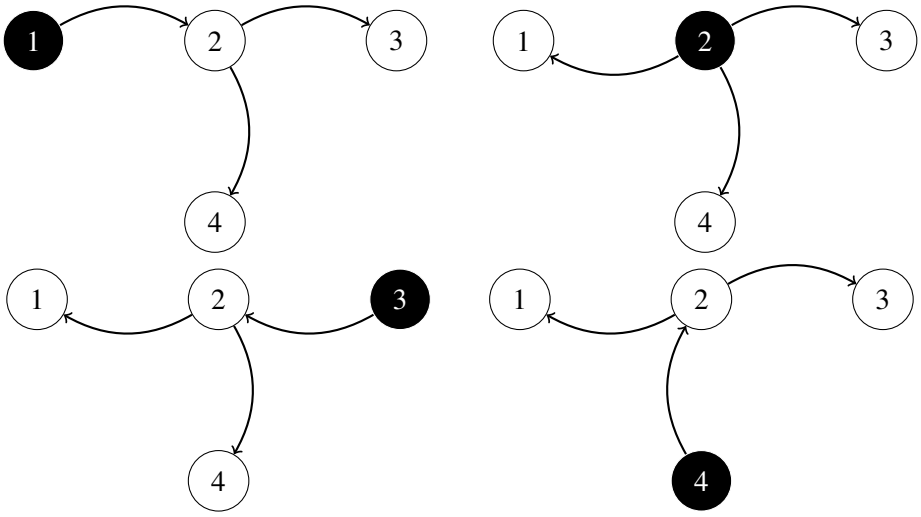
- $x_e = \begin{cases} 1, & \text{se } e \in E \text{ está na solução;} \\ 0, & \text{caso contrário.} \end{cases}$

- $\lambda_{ij}^v = \begin{cases} 1, & \text{se } \{i, j\} \in E \text{ está na arborescência enraizada em } v \in V, \text{ com direção } i \to j; \\ 0, & \text{caso contrário.} \end{cases}$

Para garantir que a solução seja uma árvore geradora, o subgrafo escolhido deve conter $|V| - 1$ arestas e ser acíclico. Essas propriedades são garantidas por:

DOI: 10.59254/sbpo-2023-175148
http://live-proceedings.pantheonsite.io/p/175148

Figura 2: Arborescências sobrepostas



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

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

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

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

Esse sistema de igualdades lineares foi proposto por Martin [1991] para resolver o problema da árvore geradora mínima. Braga [2019] e Campelo e Sousa [2022] usaram essa formulação de Martin [1991] como base para construir três formulações para o AtSCM. De modo a garantir que a árvore geradora seja também um t-spanner, ou seja, que a condição (1) seja satisfeita, usaram variáveis adicionais para expressar as distâncias entre os vértices. Nas próximas duas subseções, apresentaremos as duas formulações estudadas nesse trabalho. Cada formulação introduzirá novas variáveis e restrições diferentes do sistema apresentado acima, objetivando capturar o custo do caminho entre quaisquer dois vértices i e j na solução e restringi-lo a $t * \text{dist}_G(i, j)$. Na verdade, para garantir (1), é suficiente impor este limitante quando $\{i, j\} \in E$ [Cai e Coneil, 1995].

Em todas elas temos a seguinte função objetivo, cujo sentido será de minimização:

$$\sum_{e \in E} w_e x_e$$

2.1. Formulação (SR)

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

DOI: 10.59254/sbpo-2023-175148
http://live-proceedings.pantheonsite.io/p/175148

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

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 2. Desse modo, podemos obter a relação entre as variáveis λ e y na Proposição 2.1.

Proposição 2.1 *Para todo $u, v \in V$, $u \neq v$, e $\{i,j\} \in E$,*

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

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

Pela Proposição 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

$$y_{ij}^{uv} \leq \lambda_{ij}^u + \lambda_{ij}^v \tag{2}$$
$$y_{ij}^{uv} \leq \lambda_{ji}^u + \lambda_{ji}^v \tag{3}$$

Desse modo, observa-se que $\sum_{\{i,j\} \in E} w_{ij} y_{ij}^{uv}$ é a expressão do custo do caminho entre u e v na solução. Assim, objetivando modelar a condição (1), devemos limitar essa expressão a $t * \text{dist}_G(u,v)$, obtendo:

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

DOI: 10.59254/sbpo-2023-175148
http://live-proceedings.pantheonsite.io/p/175148

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

$$\text{(SR)} \min \sum_{\{i,j\}\in E} w_{ij}x_{ij} \tag{4}$$

$$\text{s.a:} \sum_{\{i,j\}\in E} x_{ij} = |V| - 1 \tag{5}$$

$$\sum_{i\in N^-(j)} \lambda_{ij}^v = 1, \qquad \forall v \in V, \forall j \in V\backslash\{v\} \tag{6}$$

$$\sum_{i\in N^-(v)} \lambda_{iv}^v = 0, \qquad \forall v \in V \tag{7}$$

$$x_{ij} = \lambda_{ij}^v + \lambda_{ji}^v, \qquad \forall v \in V, \forall\{i,j\} \in E \tag{8}$$

$$y_{ij}^{uv} \geq \lambda_{ij}^u - \lambda_{ij}^v, \qquad \forall\{u,v\} \in E, \forall\{i,j\} \in E \tag{9}$$

$$y_{ij}^{uv} \geq \lambda_{ji}^u - \lambda_{ji}^v, \qquad \forall\{u,v\} \in E, \forall\{i,j\} \in E \tag{10}$$

$$\sum_{\{i,j\}\in E} w_{ij}y_{ij}^{uv} \leq t * \text{dist}_G(u,v), \qquad \forall\{u,v\} \in E \tag{11}$$

$$x_{ij} \in \mathbb{B}, \qquad \forall\{i,j\} \in E \tag{12}$$

$$\lambda_{ij}^v, \lambda_{ji}^v \in \mathbb{B}, \qquad \forall v \in V, \forall\{i,j\} \in E \tag{13}$$

$$y_{ij}^{uv} \in \mathbb{B}, \qquad \forall\{i,j\} \in E, \forall\{u,v\} \in E \tag{14}$$

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

1. As variáveis y_{ij}^{uv} estão definidas apenas para $\{i,j\},\{u,v\} \in E$, entretanto, a priori, deveriam ser definidas para pares $\{u,v\} \notin E$. Cai e Coneil [1995] mostram que é suficiente impor a condição (1) apenas para pares de vértices que formam aresta no grafo, desse modo, a condição para os demais pares também é garantida. Portanto definimos as variáveis em consonância com esse fato, implicando uma quantidade menor de variáveis y ($|E|^2$ em lugar de $|E||V|^2$).

2. Devido à função objetivo, não é necessário adicionar à formulação as restrições (2) e (3). Como queremos minimizar o custo da solução, só nos interessa ativar uma variável y_{ij}^{uv} quando obrigatório, de modo que as desigualdades (9) e (10) 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 (9)-(10), além de satisfazer (11) pois $y' \leq y$ e $w \geq 0$. Além disso, ela satisfaz as desigualdades excluídas (2)–(3), devido à Proposição 2.1. Desse modo, diferentemente de Braga [2019], eliminamos (2) e (3) da formulação (SR).

3. Além de mostrar que é suficiente garantir a distância na solução apenas para pares que formam aresta no grafo, de modo a garantir (1), Cai e Coneil [1995] mostram que basta restringir a distância entre dois vértices por t vezes o peso da aresta que os conecta, substituindo a restrição (11) por $\sum_{\{i,j\}\in E} w_{ij}y_{ij}^{uv} \leq t * w_{uv}$, que é a restrição usada no trabalho de Braga [2019]. Dessa forma, o cálculo de $\text{dist}_G(u,v)$ é dispensado. 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".

DOI: 10.59254/sbpo-2023-175148
http://live-proceedings.pantheonsite.io/p/175148

2.2. Formulação CR

Em lugar de marcar explicitamente quais as arestas do caminho entre cada par de vértices, a segunda formulação, denominada (CR), rotula cada vértice com os pesos dos caminhos mínimos de cada raiz até ele. O cálculo desses rótulos pode ser feito com base em uma adaptação das variáveis de potencial π das restrições clássicas de Miller, Tuker e Zemlim (MTZ)[Miller et al., 1960], fortalecidas por Desrochers e Laporte [1991]. Mais precisamente constrói-se as seguintes variáveis:

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

Seja r a raiz de uma arborescência da solução. Construindo os potenciais da mesma maneira que as restrições MTZ clássicas, temos $\pi_j^r = \pi_i^r + 1$, quando $\lambda_{ij}^r = 1, \forall r \in V, \forall \{i, j\} \in E$, ou seja, o valor do potencial π_j^r corresponde ao número de arestas no caminho entre r e j. Como estamos interessados em captar o custo do caminho, a condição acima é alterada para: $\pi_j^r = \pi_i^r + w_{ij}$, se $\lambda_{ij}^r = 1, \forall r \in V, \{i, j\} \in E$. 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 $r \in V$ a ele mesmo, então $\pi_r^r = 0$. Para garantir a construção dos demais potenciais, tome as seguintes desigualdades:

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

nas quais M_{ij}^r e M_{ji}^r são constantes. Podemos alternativamente escrever

$$\pi_i^r - \pi_j^r \leq M_{ij}^r - (M_{ij}^r + w_{ij})\lambda_{ij}^r - (M_{ij}^r - w_{ij})\lambda_{ji}^r, \quad \forall r \in V, (i, j), (j, i) : \{i, j\} \in E, \quad (15)$$

Para verificar a validade dessas desigualdades, tome $r \in V$ e $\{i, j\} \in E$. 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 $\pi_j^r = \pi_i^r - w_{ij}$. 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 $r \in V$.

Sejam $i, j, r \in V$. A seguir, detalhamos o valor adotado para M_{ij}^r. Seja w_M o peso de uma árvore geradora de peso máximo em 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 (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

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

Similarmente, sendo ℓ_{ij}^r o tamanho do caminho entre i e j na árvore enraizada em r, temos que

$$\pi_i^r - \pi_j^r \leq |\pi_i^r - \pi_j^r| \leq \ell_{ij}^r \leq U_{ij}.$$

DOI: 10.59254/sbpo-2023-175148
http://live-proceedings.pantheonsite.io/p/175148

Pelo exposto, podemos tomar

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

Para garantir a condição (1), basta limitar superiormente cada potencial π_i^j, $i,j \in V$, 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. Dessa forma temos as seguintes desigualdades:

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

Resumindo, a formulação (CR) pode ser expressa como:

(CR)

$$\min \sum_{\{i,j\}\in E} w_{ij}x_{ij} \tag{16}$$

$$\text{s.a:} \sum_{\{i,j\}\in E} x_{ij} = |V| - 1 \tag{17}$$

$$\sum_{i\in N^-(j)} \lambda_{ij}^v = 1, \qquad \forall v \in V, \forall j \in V\backslash\{v\} \tag{18}$$

$$\sum_{i\in N^-(v)} \lambda_{iv}^v = 0, \qquad \forall v \in V \tag{19}$$

$$x_{ij} = \lambda_{ij}^v + \lambda_{ji}^v, \qquad \forall v \in V, \forall \{i,j\} \in E \tag{20}$$

$$\pi_i^r - \pi_j^r + (M_{ij}^r + w_{ij})\lambda_{ij}^r + (M_{ij}^r - w_{ij})\lambda_{ji}^r \leq M_{ij}^r, \quad \forall r \in V, (i,j),(j,i) : \{i,j\} \in E \tag{21}$$

$$\pi_r^r = 0, \qquad \forall r \in V \tag{22}$$

$$\text{dist}_G(i,j) \leq \pi_i^j = \pi_j^i \leq \min\{t * \text{dist}_G(i,j), w_M\}, \qquad \forall i,j \in V \tag{23}$$

$$x_{ij} \in \mathbb{B} \qquad \forall \{i,j\} \in E \tag{24}$$

$$\lambda_{ij}^v, \lambda_{ji}^v \in \mathbb{B}, \qquad \forall v \in V, \forall \{i,j\} \in E \tag{25}$$

$$\pi_i^j \in \mathbb{R}^+, \qquad \forall i,j \in V \tag{26}$$

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

1. Em (23) adicionar apenas as desigualdades relativas a pares $\{i,j\}$ que representam arestas em E é suficiente para corretude da formulação. Entretanto, optamos por mantê-las para todo par, pois não aumentam o tamanho do modelo, já que podem ser tratadas como *bounds* das variáveis π.

2. Além disso, basta limitar o peso de cada caminho entre i e j na solução a t vezes o peso da aresta $\{i,j\}$, tornando $t * w_{ij}$ o lado direito de (23), que é a limitante imposta na formulação (CR) apresentada no trabalho de [Braga, 2019]. Nesse trabalho, estamos impondo o limite superior de (23), pois, em termos de relaxação linear, a formulação se torna mais "apertada".

3. Ainda em (23), as igualdades $\pi_i^j = \pi_j^i$, podem ser descartadas. 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 sua utilização é interessante pois, assim,

DOI: 10.59254/sbpo-2023-175148
http://live-proceedings.pantheonsite.io/p/175148

podemos reduzir a quantidade de variáveis da formulação. Além disso, estas igualdades são utilizadas por Braga [2019] apenas para pares $\{i, j\} \in E$, permitindo uma redução bem menor de variáveis da formulação.

4. Finalmente, os limitantes M, propostos por Braga [2019], que propôs o limite $M^r_{ij} = t * \text{dist}_G(i, j)$ foram melhorados. Em nosso trabalho utilizamos $M^r_{ij} = \min\{U_{ij}, U_{ir} - \text{dist}_G(r, j)\}$, no qual $U_{ij} = \min\{t * \text{dist}_G(i, j), w_M\}$, que é pelo menos tão forte quanto o de Braga [2019].

3. Experimentos Computacionais

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, pesos aleatórios e distância euclidiana. A seguir temos com precisão os valores de cada parâmetro:

- Quantidade de vértices n ($|V|$) 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 divida em três casos:
  - Pesos Unitários ($w_e = 1$, para $e \in E$).
  - Distância euclidiana, na qual cada vértice inicialmente recebe uma coordenada aleatória (x, y) em um quadrado de lado 100 e, para cada aresta $\{i, j\} \in E$, 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 $\{1, 2, 3, 4\}$.

Sejam os três parâmetros n, p e t fixos. Iniciamos o grafo da instância como uma árvore com n vértices, para garantir conexidade 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 $\{i, j\} \in V$, 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 com pesos unitários. 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 $\{i, j\} \in E$, selecionamos aleatoriamente um peso dentre os elementos do conjunto, cada um com mesma probabilidade de escolha.

Uma observação quanto às instâncias estudadas é que todas são definidas a partir de grafos 2-conexos. Instâncias constituídas por grafos com vértice de corte podem ser solucionadas resolvendo o problema separadamente em cada uma de suas componentes 2-conexas maximais (blocos). Veja em [Sousa, 2021] a demonstração dessa propriedade.

Além da formulação (SR) e (CR), avaliamos o desempenho da formulação (AP)[Campelo e Sousa, 2022] e do algoritmo enumerativo[Campelo e Sousa, 2021], nesse mesmo conjunto de

DOI: 10.59254/sbpo-2023-175148
http://live-proceedings.pantheonsite.io/p/175148

instâncias. Os modelos foram resolvidos com uso do solver CPLEX (IBM, Versão 12.10). A implementação dos mesmos e do algoritmo enumerativo foi toda feita na linguagem de programação C++. Os experimentos foram realizados em uma máquina 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 modelo foi de 1200 segundos (20 minutos).

Tabela 1: Resultados computacionais resumidos para formulações originais e aprimoradas (instâncias com 15 e 30 vértices)

	Peso Mais Esp.		Peso Menos Esp.		Peso Unitário		Dist. Euclidiana	
Método	Resolvidas	Tempo	Resolvidas	Tempo	Resolvidas	Tempo	Resolvidas	Tempo
Form. CR	32	26.28	32	63.62	32	95.68	32	4.53
Form. CR ori	26	230.34	23	337.62	20	458.34	29	119.75
Form. SR	29	167.53	27	268.00	26	249.03	29	141.53
Form. SR ori	27	190.68	26	283.65	26	239.59	29	134.78

Tabela 2: Resultados computacionais resumidos para formulações e algoritmo enumerativo

	Peso Mais Esp.		Peso Menos Esp.		Peso Unitário		Dist. Euclidiana	
Método	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

Registramos para cada método e cada instância, o tempo demandado para determinar uma árvore t-spanner (e seu valor da solução) ou para provar sua inexistência. Objetivando comparar o efeito do aprimoramento das formulações (CR) e (SR), realizamos testes nas instâncias com $n \in \{15, 30\}$ vértices e todos os parâmetros p, t e ponderação. Na Tabela 1, exibimos o tempo médio e quantidade de instâncias solucionadas pelas duas versões de cada formulação, na qual 'Form. CR ori' (resp. 'Form. SR ori') representa a formulação (CR) (resp. (SR)) proposta por Braga [2019], e 'Form. CR' (resp. 'Form. SR') representa a formulação com os aperfeiçoamentos propostos neste trabalho. 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. Desse modo, observamos um ganho de performance para as duas formulações, especialmente para a (CR), além da redução do tempo médio de resolução, a formulação (CR) aperfeiçoada solucionou todas as 32 instâncias nas 4 ponderações dentro do tempo limite, o que não se mostrou possível por sua versão original. Já a formulação (SR) solucionou pelo menos a mesma quantidade de instâncias que sua versão original, perdendo em pouco em tempo computacional nas instâncias com peso unitário e distância euclidiana.

Visando comparar as formulações aperfeiçoadas com outros métodos da literatura, apresentamos na Tabela 2, o tempo médio e quantidade de instâncias solucionadas pelas formulações

DOI: 10.59254/sbpo-2023-175148
http://live-proceedings.pantheonsite.io/p/175148

aperfeiçoadas, juntamente com a formulação (AP) de Campelo e Sousa [2022] e do algoritmo enumerativo de Campelo e Sousa [2021], considerando as 64 instâncias para cada um dos quatro tipos de ponderações das arestas. 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. Quanto às formulações (SR) e (AP) e ao algoritmo enumerativo, observamos que apresentaram desempenhos similares. Vale observar que os tempos médios para as diferentes ponderações foram similares, com uma elevação para as instâncias com pesos unitários, o que nos mostra que essa classe apresenta-se como a mais desafiadora.

4. Conclusão

Neste trabalho, estudamos o problema da árvore t-spanner de custo mínimo, usando uma abordagem de programação matemática. Estudamos e aprimoramos duas formulações da literatura. Apresentamos e comparamos por meio de testes computacionais seus desempenhos, juntamente com a outra formulação e algoritmo enumerativo da literatura. Observamos que dentre os métodos estudados, a formulação (CR) foi a que apresentou melhor performance, comparada com os outros métodos. Além disso, observamos que a classe de instâncias com pesos unitários foi a que se apresentou mais desafiadora. Como trabalho futuro, podemos aperfeiçoar as formulações por meio de desigualdades válidas e um estudo poliédrico de seus conjuntos viáveis. Além disso, podemos estudar o uso de métodos de decomposição para problemas de programação inteira, de modo a solucionar instâncias maiores.

Referências

Arikati, S. R., Chen, D. Z., Chew, L. P., Das, G., Smid, M. H. M., e Zaroliagis, C. D. (1996). Planar spanners and approximate shortest path queries among obstacles in the plane. *Proccedings of the Fourth Annual European Symposium on Algorithms*, p. 514 – 528.

Braga, H. V. V. (2019). **Algoritmos exatos para problemas de spanner em grafos.** 2019. Tese (Doutorado em Ciência da Computação) - Universidade de São Paulo, São Paulo.

Brandstadt, A., Dragan, F. F., Le, H.-O., e Le, V. B. (2004). Tree spanners on chordal graphs: complexity and algorithms. *Theor. Comput. Sci.*, 310(1):329 – 354.

Cai, L. e Coneil, D. G. (1995). Tree spanners. *Journal on Discrete Mathematics*, 8(3):359 – 387.

Cai, L. (1992). **Tree spanners: spanning trees that approximate distances.** 1992. Tese (Doutorado em Computer Science) - University of Toronto, Toronto.

Campelo, M. e Sousa, G. H. (2021). An enumerative algorithm for the minimum weight t-spanner tree problem. *LIII Simpósio Brasileiro de Pesquisa Operacional*, p. 1 – 12.

Campelo, M. e Sousa, G. H. (2022). Nova formulação e desigualdades válidas para o problema da Árvore t-spanner de custo mínimo. *LIV Simpósio Brasileiro de Pesquisa Operacional*, p. 1 – 12.

Desrochers, M. e Laporte, G. (1991). Improvements and extensions to the miller-tucker-zemlin subtour elimination constraints. *Operations Research Letters*, 10:27 – 36.

Dragan, F. F., Fomin, F. V., e Golovach, P. A. (2011). Spanners in sparse graphs. *J. Comput. Syst. Sci.*, 77(6):1108 – 1119.

DOI: 10.59254/sbpo-2023-175148
http://live-proceedings.pantheonsite.io/p/175148

Fekete, S. P. e Kremer, J. (2001). Tree spanners in planar graphs. *Discrete Applied Mathematics*, 108(1):85 – 103.

Goldschmidt, O., Takvorian, A., e Yu, G. (1996). On finding a biconnected spanning planar subgraph with applications to the facilities layout problem. *European Journal of Operational Research*, 94(1):97 – 105.

Madanlal, M., Venkatesan, G., e Rangan, C. (1996). Tree 3-spanners on interval, permutation and regular bipartite graphs. *Information Processing Letters*, 59(2):97 – 102.

Marble, J. D. e Bekris, K. E. (2011). Asymptotically near-optimal is good enough for motion planning. *Proc. of the 15th International Symposium on Robotics Research (ISRR-11)*.

Martin, R. K. (1991). Using separation algorithms to generate mixed integer model reformulations. *Operations Research Letters*, 10:119 – 128.

Miller, C. E., Tucker, A. W., e Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *J. ACM*, 7(4):326 – 329.

Nguyen, C., Shen, J., Hou, M., Sheng, L., Miller, W., e Zhang, L. (2008). Approximating the spanning star forest problem and its application to genomic sequence alignment. *SIAM Journal on Computing*, 38(3):946 – 962.

Oliveira, C. A. S. e Pardalos, P. M. (2005). A survey of combinatorial optimization problems in multicast routing. *Comput. Oper. Res.*, 32(8):1953 – 1981.

Papoutsakis, I. (2018). Tree spanners of bounded degree graphs. *Discrete Appl. Math.*, 236:395 – 407.

Peleg, D. e Reshef, E. (1999). A variant of the arrow distributed directory with low average complexity. *Automata, Languages and Programming*, p. 615 – 624.

Peleg, D. e Ullman, J. D. (1989). An optimal synchronizer for the hypercube. *SIAM J. Comput.*, 18 (4):740 – 747.

Peleg, D. e Upfal, E. (1988). A tradeoff between space and efficiency for routing tables. *Proceedings of the twentieth annual ACM symposium on Theory of computing*, p. 43 – 52.

Singh, K. e Sundar, S. (2018). Artifical bee colony algorithm using problem-specific neighborhood strategies for the tree t-spanner problem. *Applied Soft Computing*, 62:110 – 118.

Sousa, G. H. (2021). O problema da árvore t-spanner de custo mínimo. Master's thesis, Universidade Federal do Ceará.

Venkatesan, G., Rotics, U., Madanlal, M., Makowsky, J., e Rangan, C. P. (1997). Restrictions of minimum spanner problems. *Information and Computation*, 136(2):143 – 164.

DOI: 10.59254/sbpo-2023-175148
http://live-proceedings.pantheonsite.io/p/175148