

# Formulação matemática para o problema da árvore $t$ -spanner de custo mínimo \*

Gabriel H. de Sousa<sup>1,3</sup>, Manoel Campêlo<sup>2,3</sup>

<sup>1</sup>Departamento de Computação, Universidade Federal do Ceará

<sup>2</sup>Departamento de Estatística e Matemática Aplicada, Universidade Federal do Ceará

<sup>3</sup>Grupo de Pesquisa ParGO - Paralelismo, Grafos e Otimização

gabriel@sousa.9957@gmail.com, mcampelo@ufc.br

**Abstract.** *The Minimum Weight  $t$ -Spanner Tree Problem, whose input is an edge-weighted simple graph  $G$  and a parameter  $t \geq 1$ , consists in determining a spanning tree  $T$  of  $G$  of minimum weight among those where the path between each pair of vertices  $i$  and  $j$  in  $T$  has weight at most  $t$  times the minimum weight of a path between  $i$  and  $j$  in  $G$ . We propose a mathematical formulation which can be derived from a projection of an existing formulation. We implemented both models and compared their computational performance.*

**Resumo.** *O Problema da Árvore  $t$ -Spanner de Custo Mínimo, cuja entrada é um grafo simples  $G$ , ponderado em arestas, e um parâmetro  $t \geq 1$ , consiste em determinar uma árvore geradora  $T$  de  $G$  de menor custo dentre aquelas onde a distância entre qualquer par de vértices  $i$  e  $j$  em  $T$  é no máximo  $t$  vezes o menor peso de um caminho entre  $i$  e  $j$  em  $G$ . Propomos uma formulação matemática que pode ser derivada pela projeção de uma formulação já existente. Implementamos ambos os modelos e comparamos seus desempenhos computacionais.*

## 1. Introdução

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

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

onde  $\text{dist}(i, j)$  é o peso de um  $i, j$ -caminho mínimo. Quando  $H$  é acíclico, portanto uma árvore geradora, dizemos que  $H$  é uma árvore  $t$ -spanner. No problema da árvore  $t$ -spanner de custo mínimo (AtSCM) queremos determinar, caso exista, uma árvore  $t$ -spanner em  $G$  de peso mínimo, ou seja, tal que a soma dos pesos de suas arestas seja mínima. Apesar da restrição (1) ser imposta para quaisquer dois vértices  $i$  e  $j$  do grafo  $G$ , é suficiente exigi-la apenas para os pares de vértices diretamente conectados em  $G$  [Cai e Coneil 1995].

Quando o grafo de entrada tem pesos das arestas unitários, temos um problema de decisão, visto que toda árvore geradora, em particular uma árvore  $t$ -spanner, terá peso

---

\*Financiado por Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) – Código 001, FUNCAP PS1-0186-00155.01.00/21 e PNE-0112.00061.01.00-16 e CNPQ Proc. 312417/2022-5.

$|V| - 1$ . Nesse caso, sabe-se que verificar se existe uma árvore  $t$ -spanner é um problema  $NP$ -completo para  $t \geq 4$  [Brandstadt et al. 2004]. Quando os pesos são arbitrários, já para  $t > 1$  fixo, este problema é  $NP$ -completo [Cai e Coneil 1995]. Casos polinomiais também são demonstrados [Cai e Coneil 1995].

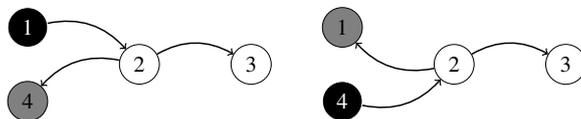
O AtSCM com pesos unitários tem sido bastante explorado, mas ainda há poucos estudos para o caso com pesos arbitrários, principalmente na área de programação matemática. De fato, a literatura do problema apresenta apenas um algoritmo exato e três formulações de programação inteira [Braga 2019, Campelo e Sousa 2021, Campelo e Sousa 2022]. Em se tratando de heurísticas, existe um algoritmo de colônia de abelhas [Singh e Sundar 2018]. Neste trabalho, propomos uma nova formulação, que elimina  $|E|^2$  variáveis da formulação (SR) da literatura [Braga 2019, Sousa 2021], às custas de um conjunto exponencial de restrições. Tais desigualdades, porém, podem ser separadas de forma eficiente. Avaliamos computacionalmente o desempenho da formulação modificada, comparando-a com a original.

## 2. Formulações matemáticas

A formulação (SR) toma como base a formulação de Martin [Martin 1991] para o problema da árvore geradora mínima e adiciona outras variáveis e restrições para expressar e limitar a distância entre os vértices. Considerando a orientação simétrica de  $G$ , onde cada aresta  $\{i, j\}$  dá origem aos arcos  $i \rightarrow j$  e  $j \rightarrow i$ , a formulação constrói  $|V|$  arborescências, cada uma delas enraizada em um nó diferente de  $G$  mas descrevendo todas uma orientação de uma mesma árvore geradora de  $G$ , que será uma árvore  $t$ -spanner. Uma arborescência é uma orientação de uma árvore de modo que todo vértice tem grau de entrada 1, a menos de um nó, chamado raiz, que tem grau de entrada zero. Dizemos que duas arborescências se sobrepõem se são orientações de uma mesma árvore (enraizadas em nós diferentes).

Para a representação dessas arborescências, o modelo de Martin usa as seguintes variáveis binárias:  $x_e$  para indicar se a aresta  $e \in E$  está na solução;  $\lambda_{ij}^v$  para indicar se a aresta  $\{i, j\} \in E$  está na solução com orientação  $i \rightarrow j$  na arborescência enraizada em  $v \in V$ . Além disso, (SR) define variáveis binárias  $y_{ij}^{uv}$  que indicam explicitamente a presença de uma aresta  $\{i, j\} \in E$  no caminho entre dois vértices  $u, v$  com  $\{u, v\} \in E$ . As variáveis  $y$  podem ser expressas em função das variáveis  $\lambda$  como segue. Sejam  $T^u$  e  $T^v$  duas arborescências sobrepostas definidas pelas variáveis  $\lambda$ , enraizadas, respectivamente, nos vértices  $u$  e  $v$ , distintos. 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 sobrepostas da figura abaixo enraizadas em 1 (à esquerda) e 4 (à direita). Desse modo, podemos obter a relação entre as variáveis  $\lambda$  e  $y$  como:

$$y_{ij}^{uv} = \max\{\lambda_{ij}^u - \lambda_{ij}^v, \lambda_{ji}^u - \lambda_{ji}^v\}, \quad \forall \{u, v\}, \{i, j\} \in E \quad (2)$$



Sendo  $w_{ij}$  o peso de  $\{i, j\} \in E$ , a formulação (SR) pode ser expressa como:

$$(SR) \min \sum_{\{i,j\} \in E} w_{ij} x_{ij} \quad (3)$$

$$\text{s.a: } \sum_{\{i,j\} \in E} x_{ij} = |V| - 1 \quad (4)$$

$$\sum_{i \in N^-(j)} \lambda_{ij}^v = 1, \quad \forall v \in V, \forall j \in V \setminus \{v\} \quad (5)$$

$$\sum_{i \in N^-(v)} \lambda_{iv}^v = 0, \quad \forall v \in V \quad (6)$$

$$x_{ij} = \lambda_{ij}^v + \lambda_{ji}^v, \quad \forall v \in V, \forall \{i, j\} \in E \quad (7)$$

$$y_{ij}^{uv} \geq \lambda_{ij}^u - \lambda_{ij}^v, y_{ij}^{uv} \geq \lambda_{ji}^u - \lambda_{ji}^v, \quad \forall \{u, v\} \in E, \forall \{i, j\} \in E \quad (8)$$

$$\sum_{\{i,j\} \in E} w_{ij} y_{ij}^{uv} \leq t * \text{dist}_G(u, v), \quad \forall \{u, v\} \in E \quad (9)$$

$$x \in \mathbb{B}^{|E|}, \lambda \in \mathbb{B}^{2|E| \times |V|}, y \in \mathbb{B}^{|E| \times |E|} \quad (10)$$

As restrições (4)-(7) garantem que a solução irá definir as  $|V|$  arborescências sobrepostas. As restrições (8) determinam, para cada par de vértices  $\{u, v\} \in E$ , as arestas que estão no caminho entre  $u$  e  $v$ . Finalmente, a restrição (9) limita o tamanho dos caminhos de modo a satisfazer (1).

A partir de (9) e (2), podemos expressar a restrição  $t$ -spanner (1) para  $\{u, v\}$  como:

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

Veja que  $\lambda_{ij}^u + \lambda_{ji}^u = \lambda_{ij}^v + \lambda_{ji}^v$ , ou melhor  $\lambda_{ij}^u - \lambda_{ij}^v = -(\lambda_{ji}^u - \lambda_{ji}^v)$ . Então, a expressão acima pode ser reescrita como:

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

Essa desigualdade não linear pode ser substituída pelas seguintes desigualdades lineares:

$$\sum_{\{i,j\} \in E'} w_{ij} (\lambda_{ij}^u - \lambda_{ij}^v) + \sum_{\{i,j\} \in E \setminus E'} w_{ij} (\lambda_{ij}^v - \lambda_{ij}^u) \leq t * \text{dist}_G(u, v), \quad \forall E' \subseteq E \quad (11)$$

De fato, o subconjunto  $E' \subseteq E$  que maximiza o lado esquerdo da expressão (11) é  $E' = \{\{i, j\} \in E : \lambda_{ij}^u - \lambda_{ij}^v \geq 0\}$ , levando ao valor máximo  $\sum_{\{i,j\} \in E} w_{ij} |\lambda_{ij}^u - \lambda_{ij}^v|$ . Isso mostra também que tais desigualdades podem ser separadas em tempo polinomial. Dado  $\bar{\lambda}$ , caso  $\sum_{\{i,j\} \in E} w_{ij} |\bar{\lambda}_{ij}^u - \bar{\lambda}_{ij}^v| > t * \text{dist}_G(u, v)$ , 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 (11) são redundantes na formulação (SR), pois são dominadas por (9) e (8). De fato, somando as restrições à esquerda em (8) para  $\{i, j\} \in E'$  e à direita para  $\{i, j\} \in E \setminus E'$ , obtemos

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

Por outro lado, a inclusão das desigualdades (11) em (SR) dispensa o uso das variáveis  $y$  (e restrições em que essas variáveis aparecem). Dessa forma, obtemos uma nova formulação, como bem menos variáveis, porém com um número exponencial de restrições, mas que podem ser separadas de forma eficiente. Assim, podemos resolvê-la aplicando o método de planos-de-corte.

### 3. Experimentos Computacionais

Avaliamos o desempenho das duas formulações em instâncias geradas aleatoriamente. Cada grafo é gerado a partir de quatro parâmetros:  $n \in \{30, 45, 60\}$  (quantidade de vértices),  $p \in \{0.2, 0.5, 0.8, 1\}$  (probabilidade de existência de cada aresta),  $t \in \{3, 4\}$  e ponderação (pesos unitários e pesos arbitrários), totalizando 48 instâncias. Para garantir conectividade, iniciamos o grafo da instância como uma árvore com  $n$  vértices. 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, gera-se um número aleatório  $p' \in [0, 1]$ . Se  $p' \leq p$  adicionamos a aresta  $\{i, j\}$  ao grafo. As instâncias com esses parâmetros se apresentaram mais difíceis [Sousa 2021]; por esse motivo, iremos avaliar e comparar os desempenhos do modelo (SR) e da formulação proposta nesse conjunto de instâncias. Os modelos foram resolvidos com uso do solver CPLEX (IBM, Versão 12.10). Sua implementação 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).

Formulação	Pesos Unitários		Pesos Arbitrários	
	Instâncias Resolvidas	Tempo Médio	Instâncias Resolvidas	Tempo Médio
SR	5	979.50	7	924.58
Proposta	10	768.46	7	939.92

Na Tabela acima exibimos o tempo médio (em segundos) e quantidade de instâncias solucionadas pela formulação (SR) e pela formulação proposta para os dois conjuntos de instâncias. Observe que a formulação proposta solucionou um número maior ou igual de instâncias que a formulação da literatura em ambos os casos. Além disso, observamos que seu tempo médio de resolução também foi inferior ou comparável. Ainda assim, 31 instâncias não foram resolvidas otimamente no tempo limite. Vale destacar que muitas dessas instâncias não possuem solução viável.

### 4. Conclusão

Neste trabalho, estudamos o problema da árvore  $t$ -spanner de custo mínimo, usando uma abordagem de programação matemática. Propomos uma nova formulação, semelhante a uma formulação da literatura. Apresentamos e comparamos por meio dos testes computacionais os dois modelos e observamos que a formulação proposta se apresentou mais eficiente que a formulação da literatura para o conjunto de instâncias considerado. Como trabalho futuro, podemos realizar um estudo poliédrico das formulações e analisar os conjuntos viáveis de suas respectivas relaxações lineares. Uma outra direção de pesquisa é a utilização de métodos de decomposição para problemas de programação inteira, de modo à solucionar instâncias cada vez maiores.

## Referências

- 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.
- Campelo, M. e Sousa, G. H. (2021). An enumerative algorithm for the minimum weight t-spanner tree problem. *LIII Simpósio Brasileiro de Pesquisa Operacional*, pages 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*, pages 1 – 12.
- Martin, R. K. (1991). Using separation algorithms to generate mixed integer model reformulations. *Operations Research Letters*, 10:119 – 128.
- Singh, K. e Sundar, S. (2018). Artificial 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**. 2021. Dissertação (Mestrado em Ciência da Computação), Universidade Federal do Ceará.