Uma estratégia para selecionar vértices como candidatos a roteadores em uma árvore de Steiner

João Guilherme Martinez[1], Rosiane de Freitas[1], Altigran da Silva[1], Fábio Protti[2]

[1]Instituto de Computação – Universidade Federal do Amazonas (UFAM)
Manaus – AM – Brasil

[2]Instituto de Computação – Universidade Federal Fluminense (UFF)
Niterói – RJ – Brasil

`{joaogam,rosiane,alti}@icomp.ufam.edu.br, fabio@ic.uff.br`

Abstract. *The Steiner tree optimization problem in graphs is NP-hard, however, algorithms that use heuristics obtain results close to optimal solution in polynomial time. In this paper we present a new algorithm based on an exact enumerative algorithm from literature. The proposed heuristic selects vertices as candidates to be routers in the Steiner tree.*

Resumo. *O problema da árvore de Steiner em grafos é NP-difícil, no entanto, algoritmos que fazem uso de heurísticas conseguem obter resultados próximos da solução ótima em tempo polinomial. Neste trabalho apresentamos um novo algoritmo baseado em um algoritmo exato enumerativo da literatura. A heurística proposta seleciona vértices como candidatos a serem roteadores na árvore de Steiner.*

1. Introdução

Considere um grafo $G = (V(G), E(G))$ onde $V(G)$ representa o conjunto de vértices do grafo e $E(G)$ o conjunto de suas arestas. Dado um subconjunto W de vértices de $V(G)$, uma árvore de Steiner consiste num subgrafo conexo T de G que contêm W e cuja soma dos pesos das arestas de $E(T)$ seja mínimo. Os vértices do conjunto W são chamados de terminais e os demais vértices de $V(G)$ utilizados para construir T são chamados de vértices de Steiner. A versão de decisão do problema da árvore de Steiner em grafos é NP-Completo [Karp 1972], portanto não possui algoritmos polinomiais que encontrem a solução ótima, entretanto existem algoritmos na literatura que fazem uso de heurísticas e alcançam soluções próximas da ótima. O problema da árvore de Steiner possui aplicações práticas como o desenho de layouts de circuitos e o projeto de redes, além de outras variações do problema que surgem no campo prático.

Considere n como o número de vértices em $V(G)$, m como o número de arestas em $E(G)$ e k como o número de vértices terminais do conjunto W. A Tabela 1 apresenta alguns trabalhos relacionados publicados na literatura com a complexidade do algoritmo no pior caso e a razão de aproximação das soluções obtidas em relação à solução ótima, sendo alguns destes aproximativos ([Takahashi and Matsuyama 1980],[Wu et al. 1986],[Kou et al. 1981],[Zelikovsky 1993], [Robins and Zelikovsky 2000]) e outros exatos ([Dreyfus and Wagner 1971], [Vygen 2011],[Dourado et al. 2014],[Byrka et al. 2013],[Pajor et al. 2014]).

Título	Autores	Ano	Complexidade	Razão de Aproximação
The Steiner problem in graphs	Dreyfus e Wagner	1971	$O(n^3 + n^2 2^{k-1} + n3^{k-1})$	1
An approximate solution for the steiner problem in graphs	Takahashi e Matsuyama	1980	$O(kn^2)$	2
A fast algorithm for Steiner trees	Kou, Markowsky e Berman	1981	$O(kn^2)$	$2(1 - 1/l)$[a]
A faster approximation algorithm for the Steiner problem in graphs	Wu, Widmayer e Wong	1986	$O(m \log n)$	$2(1 - 1/l)$[a]
An 11/6-approximation algorithm for the network steiner problem	Zelikovsky	1993	$O(nm + k^4)$	1.833
Improved Steiner Tree Approximation in Graphs	Robins e Zelikovsky	2000	$O(kn^2)$	1.55
Faster algorithm for optimum Steiner trees	Vygen	2011	$O(nk2^{k+\log_2 k \log_2 n})$	1
Steiner Tree Approximation via Iterative Randomized Rounding	Byrka, Grandoni, Rothvoss e Sanità	2013	indefinido	1.38
Algorithmic aspects of Steiner convexity and enumeration of Steiner trees	Dourado, Oliveira e Protti	2014	$O(n^2(n + m) + n^{k-2} + n\alpha)$[b]	1
A Robust and Scalable Algorithm for the Steiner Problem in Graphs	Pajor, Uchoa e Werneck	2018	$O(m(min\{nk, m\}))$	indefinido

Tabela 1. Algoritmos conhecidos na literatura para árvore de Steiner.

Na seção 2 são mostrados os conceitos preliminares para o funcionamento do algoritmo, o pseudo-código e a complexidade no pior caso. Na seção 3 são apresentados os resultados dos testes computacionais e na seção 4 são feitas as considerações finais.

2. Descrição do Algoritmo

O trabalho de Dourado et al. (2014) propõe um algoritmo exato enumerativo para obter as árvores de Steiner de um dado grafo em tempo exponencial para o número de terminais k. Nos conceitos preliminares, os autores explicam que os vértices de Steiner podem ser de dois tipos: *linkers* e roteadores. Os *linkers* são vértices de grau igual a 2 e os roteadores devem possuir sempre grau maior que 2. A partir disso, provam que em uma árvore de Steiner, sempre existe no máximo somente $k - 2$ roteadores e que todo roteador ou terminal possui um caminho que é mínimo em G até outro roteador ou terminal. Os vértices que formam esses caminhos são os *linkers*, por isso sempre possuem grau igual a 2.

No algoritmo proposto pelos autores, são explorados todos os possíveis subconjuntos de roteadores que fazem parte da solução ótima, portanto consideram todos os subconjuntos com 0 a $k - 2$ vértices como roteadores. Neste ponto o algoritmo torna-se exponencial na ordem de $O(n^{k-2})$.

A heurística proposta neste trabalho consiste em analisar todos os vértices não-terminais e usá-los como roteadores, um por vez, e analisar aquele que melhora a qualidade da última melhor solução obtida. No entanto esse procedimento só é executado até no máximo $k - 2$ vezes, pois esta é a quantidade máxima de roteadores em uma árvore de Steiner mínima ótima. O Algoritmo 1 mostra o pseudo-código do algoritmo heurístico proposto.

[a] l representa o número de folhas em uma árvore de Steiner ótima.
[b] α representa o número de árvores de Steiner ótimas enumeradas.

Assim como no algoritmo exato [Dourado et al. 2014], a ideia geral consiste em utilizar uma matriz d_G com as distâncias dos caminhos mínimos entre todos os vértices do grafo G e construir templates, que são árvores geradoras mínimas de grafos completos G_{RW} formados com os terminais e os roteadores, utilizando as distâncias em d_G para os pesos das arestas.

A complexidade geral do algoritmo fica definida a seguir: para encontrar as distâncias d_G é $O(n^3)$. A iteração principal do algoritmo é da ordem de $O(nk^3)$ devido ao algoritmo para gerar uma árvore geradora mínima no grafo completo G_{RW}. Portanto a complexidade total do algoritmo é $O(n^3 + nk^3)$.

Algoritmo 1: Heurística para árvore de Steiner

```
Entrada: Grafo G, conjunto W
Saída: Árvore de Steiner T
início
    melhor ← ∞;
    d_G ← todas as menores distâncias dos caminhos mínimos entre todos os
    vértices;
    G_RW ← grafo completo somente com os vértices de W usando as
    distâncias de d_G;
    se |AGM(G_RW)| < melhor então
        melhor ← |AGM(G_RW)|;
        T ← AGM(G_RW);
    fim
    r ← 0;
    W' ← W;
    changed ← true;
    enquanto changed = true E r < k − 2 faça
        changed ← false;
        para cada vértice v de G que não esteja em W' faça
            G_RW ← grafo completo somente com os vértices de W' ∪ v
            usando as distâncias de d_G;
            se |AGM(G_RW)| < melhor então
                melhor ← |AGM(G_RW)|;
                T ← AGM(G_RW);
                changed ← true;
                v' ← v;
            fim
        fim
        se changed = true então
            r = r + 1;
            W' ← W' ∪ v';
        fim
    fim
    expande os caminhos mínimos em T;
    retorna T;
fim
```

3. Experimentos Computacionais

A Tabela 2 apresenta os resultados dos testes computacionais realizados com as bases de teste de um *benchmark* conhecido[c]. As árvores encontradas pelo algoritmo foram comparadas com as soluções ótimas esperadas das instâncias em relação ao grau de

[c]Bases de testes disponíveis em: http://steinlib.zib.de/testset.php

aproximação. Para os experimentos foram utilizadas somente instâncias em que a solução ótima é conhecida.

	Esparsos com pesos aleatórios	**Completos com pesos aleatórios**	**Esparsos com pesos euclidianos**	**Completos com pesos euclidianos**	**Esparsos com pesos incidentes**	**TOTAL**
Instâncias testadas	60	27	14	14	395	514
Soluções ótimas encontradas	31	23	14	12	130	212
Porcentagem de ótimas encontradas	51,67%	85,19%	100%	85,71%	32,91%	41,25%
Média de aproximação	1,01	1,00	1,00	1,00	1,01	1,01
Pior aproximação encontrada	1,09	1,04	1,00	1,00	1,12	1,12

Tabela 2. Qualidade das soluções encontradas utilizando a heurística proposta.

4. Considerações Finais

A heurística proposta apresenta bons resultados em termos de qualidade da solução obtida em tempo $O(n^3 + nk^3)$ e é de fácil compreensão e implementação. Melhorar os critérios de escolha dos vértices roteadores, obter a razão de aproximação do algoritmo e fazer um estudo comparativo com outras heurísticas são atividades para trabalhos futuros.

Referências

Byrka, J., Grandoni, F., Rothvoss, T., and Sanità, L. (2013). Steiner tree approximation via iterative randomized rounding. *J. ACM*, 60(1):6:1–6:33.

Dourado, M., Oliveira, R., and Protti, F. (2014). Algorithmic aspects of steiner convexity and enumeration of steiner trees. *Annals of Operations Research*, 223(1):155–171.

Dreyfus, S. E. and Wagner, R. A. (1971). The steiner problem in graphs. *Networks*, 1(3):195–207.

Karp, R. M. (1972). *Reducibility among Combinatorial Problems*, pages 85–103. Springer US, Boston, MA.

Kou, L., Markowsky, G., and Berman, L. (1981). A fast algorithm for steiner trees. *Acta Informatica*, 15(2):141–145.

Pajor, T., Uchoa, E., and Werneck, R. F. (2014). A robust and scalable algorithm for the steiner problem in graphs. *CoRR*, abs/1412.2787.

Robins, G. and Zelikovsky, A. (2000). Improved steiner tree approximation in graphs. In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '00, pages 770–779, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.

Takahashi, H. and Matsuyama, A. (1980). An approximate solution for the steiner problem in graphs. In *Math. Jap*, pages 24:573–577.

Vygen, J. (2011). Faster algorithm for optimum steiner trees. *Information Processing Letters*, 111(21):1075 – 1079.

Wu, Y. F., Widmayer, P., and Wong, C. K. (1986). A faster approximation algorithm for the steiner problem in graphs. *Acta Inf.*, 23(2):223–229.

Zelikovsky, A. Z. (1993). An 11/6-approximation algorithm for the network steiner problem. *Algorithmica*, 9(5):463–470.