This paper appeared in: Information Processing Letters 89 (2004), 15–18

On Approximation Algorithms for the Terminal Steiner Tree Problem

Doratha E. Drake Stefan Hougardy

Humboldt-Universität zu Berlin
Institut für Informatik
10099 Berlin, GERMANY
{drake,hougardy}@informatik.hu-berlin.de

revised version

Abstract. The terminal Steiner tree problem is a special version of the Steiner tree problem, where a Steiner minimum tree has to be found in which all terminals are leaves. We prove that no polynomial time approximation algorithm for the terminal Steiner tree problem can achieve an approximation ratio less than $(1 - o(1)) \ln n$ unless NP has slightly superpolynomial time algorithms. Moreover, we present a polynomial time approximation algorithm for the metric version of this problem with a performance ratio of 2ρ, where ρ denotes the best known approximation ratio for the Steiner tree problem. This improves the previously best known approximation ratio for the metric terminal Steiner tree problem of $\rho + 2$.

Keywords. approximation algorithms, Steiner tree

1 Introduction

Given a graph $G = (V, E)$ with a length function $w : E \to \mathbb{R}^+$ and a subset $R \subseteq V$ of *terminals* the *Steiner tree problem* is to find a minimum length tree in G that contains all vertices of R. Such a tree is called a *Steiner minimum tree* (SMT). The computation of SMTs is an important problem in many applications and therefore much effort has been spent over the last three decades to design algorithms for solving this problem. As the Steiner tree problem is known to be APX-hard [1, 2], unless P=NP, only approximate solutions to the Steiner tree problem can be found in polynomial time. The best polynomial time approximation algorithm for the Steiner tree problem that is currently known is due to Robins and Zelikovsky [10]. They proved that their algorithm has an approximation ratio of at most 1.55, i.e. it always finds a solution to the Steiner tree problem that is larger than the optimum solution by at most

This paper appeared in: Information Processing Letters 89 (2004), 15–18

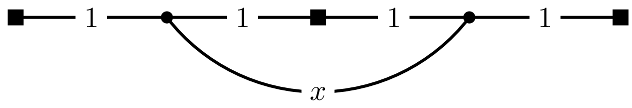

Figure 1: A non-metric instance for the terminal Steiner tree problem. Terminals are indicated by squares, non-terminals by circles.

a factor of 1.55. See the book of Hwang, Richards and Winter [8] for more background on the Steiner tree problem and the survey paper [7] for more on the development of approximation algorithms for this problem.

Clearly, all leaves in an SMT are vertices of R, but not all vertices of R need to be leaves in an SMT. However, in some applications, as for example the global routing in VLSI-Design, all vertices of R must be leaves in the Steiner tree. This motivates the study of the *terminal Steiner tree problem* [9]. This is a special version of the Steiner tree problem, where a shortest length Steiner tree has to be found in which all terminals are leaves.

For the Steiner tree problem one may assume that the given graph is metric, i.e. it is a complete graph and the edge lengths satisfy the triangle inequality. The reason for this is that in an SMT no two vertices can be connected by an edge that is longer than a shortest path in the graph connecting these two vertices. Lin and Xue [9] remark that similar to the Steiner tree problem one may also always assume that a given instance for the terminal Steiner tree problem is metric. However this assumption is not true. A simple counterexample is shown in Figure 1. For this graph the optimum solution has length $x + 3$ while the optimum solution in the metric case has length at most 5 because x must have a value of at most 2 to satisfy the metric. As x can be chosen arbitrarily large in the non-metric case the difference between the metric and non-metric solution can be arbitrarily large. To avoid these problems one may consider the restriction of the terminal Steiner tree problem to instances where for each terminal there is a non-terminal vertex in the instance that has exactly the same neighbors. In this case the non-metric version can be transformed into an equivalent metric one.

We show in Section 2 that unless $NP = DTIME(n^{O(\log\log n)})$ the non-metric version of the terminal Steiner tree problem cannot be approximated to any constant. This will be proved by showing that a constant factor approximation algorithm for the non-metric terminal Steiner tree problem would give a constant factor approximation algorithm for the set cover problem. The latter problem is known to be not approximable better than $\ln n$ [5], unless $NP = DTIME(n^{O(\log\log n)})$.

Lin and Xue [9] prove that the metric version of the terminal Steiner tree problem is APX-hard and they present a factor $(2+\rho)$-approximation algorithm for this problem. Here ρ denotes the best approximation ratio for the Steiner tree problem that can be achieved by a polynomial time algorithm. According to the result of Robins and Zelikovsky [10] it follows $\rho < 1.55$. Chlebík and Chlebíková proved that $\rho > 1.01$ must hold if $P \neq NP$.

In Section 3 we improve the factor $2 + \rho$ of Lin and Xue [9] by presenting a

This paper appeared in: Information Processing Letters 89 (2004), 15–18

factor 2ρ approximation algorithm for the metric terminal Steiner tree problem. Note that $2\rho < 2 + \rho$ for $\rho < 2$.

2 The hardness result

Given a set S with elements $1, \ldots, n$ and a family $\mathcal{F} = \{S_1, S_2, \ldots, S_m\}$ of subsets of S, the *set cover problem* is to find a minimum size subfamily of $\mathcal{F}$ that covers all elements of S. In this section we prove that approximating the terminal Steiner tree problem is at least as hard as approximating the set cover problem. For the latter problem Feige [5] has shown that unless $NP = DTIME(n^{O(\log \log n)})$ no polynomial time approximation algorithm can have a performance guarantee better than $(1 - o(1)) \ln n$.

Theorem 1 *For any constant α a polynomial time α-approximation algorithm for the terminal Steiner tree problem yields a polynomial time α-approximation algorithm for the set cover problem.*

Proof. We use a reduction from set cover similar to the one used for showing the hardness of the group Steiner tree problem [6]. Consider an instance of the (unweighted) set cover problem with elements $1, \ldots, n$ and sets $S_1, \ldots, S_m$. We construct from this instance an instance of the terminal Steiner tree problem as follows. Take a star with the vertex x as its center and m rays ending in the vertices $S_1, \ldots, S_m$. All these rays get a weight of 1. Now add n terminals $T_1, \ldots, T_n$ and an extra terminal T_0. The terminal T_0 is connected to vertex x by an edge of length 0. For each j connect terminal T_j by edges of length 0 with all vertices S_i for which j is an element of the set S_i. Now a solution to the terminal Steiner tree problem contains the vertex x (as terminal T_0 must be connected with the other terminals) and some of the rays emenating from x. The weight of the solution of the terminal Steiner tree problem is exactly the number of these rays. All the sets S_i that are connected to x by these rays form a set cover because for each j terminal T_j is connected to other terminals in the solution of the terminal Steiner tree problem only via sets which contain element j. Therefore the weight of the solution to the terminal Steiner tree problem is exactly as large as the solution to the set cover problem that is induced by this solution. In particular this implies that an α-approximation algorithm for the terminal Steiner tree problem yields an α-approximation algorithm for the set cover problem. $\square$

Note that the instance of the terminal Steiner tree problem produced by this reduction is not metric. Theorem 1 easily extends to the case where α is a function depending on the input size. Let N denote the number of vertices in an instance of the terminal Steiner tree problem and let n and m denote the number of elements and sets in a set cover instance. Then an $\alpha(N)$-approximation algorithm for the terminal Steiner tree problem yields an $\alpha(n+m+2)$-approximation algorithm for the set cover problem. As the hardness result of Feige [5] for set cover also applies to the case where $m = o(n)$ we get the following corollary.

This paper appeared in: Information Processing Letters 89 (2004), 15–18

Corollary 1 *Unless $NP = DTIME(n^{O(\log\log n)})$ no polynomial time approximation algorithm for the terminal Steiner tree problem has a performance ratio better than $(1 - o(1)) \ln n$.* □

3 The approximation algorithm

In this section we present a polynomial time approximation algorithm for the metric version of the terminal Steiner tree problem which has a performance ratio of 2ρ, where ρ denotes the best possible performance ratio that can be obtained in polynomial time for the Steiner tree problem. (As we learned from an anonymous referee this result has been proved independently in [3].)

In the following we will always assume that a feasable solution to the terminal Steiner tree problem exists. Note that in the metric version of the terminal Steiner tree problem this is precisely the case if at least one non-terminal vertex exists or the graph has at most two vertices. If the given instance contains only two vertices the terminal Steiner tree problem is simply a shortest path problem. Therefore we may assume in the following that the given instance of the terminal Steiner tree problem contains at least three terminals.

Our algorithm for computing a factor 2ρ approximation for the terminal Steiner tree problem is based on two crucial operations. The first is a preprocessing step in which all edges in the given graph are removed if they connect two vertices from R. The second operation is the *star-replacement*. This operation reduces the degree of a vertex in R to one. It goes as follows (see Figure 2). Assume r is a vertex in a Steiner tree T for R that has degree $k > 1$. Denote by $a_1, \ldots, a_k$ the neighbors of r. By the preprocessing step all these neighbors must be vertices in $V \setminus R$. Without loss of generality assume that $w(\{r, a_1\}) \leq w(\{r, a_i\})$ for $1 \leq i \leq k$. Then a new Steiner tree T' for R is obtained by replacing the edges $\{r, a_2\}, \ldots, \{r, a_k\}$ by the edges $\{a_1, a_2\}, \ldots, \{a_1, a_k\}$ (see Figure 2). In T' vertex r is a leaf while the degree of all other vertices in R has not changed.

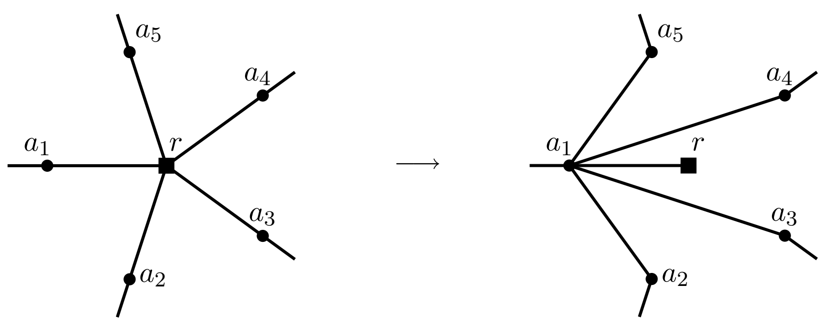

Figure 2: The star-replacement.

The algorithm TerminalSteiner uses these two operations as a main ingredient and is shown in Figure 3.

Theorem 2 *Algorithm TerminalSteiner has performance ratio 2ρ.*

This paper appeared in: Information Processing Letters 89 (2004), 15–18

Algorithm TerminalSteiner $(\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E}), \boldsymbol{w} : \boldsymbol{E} \rightarrow \mathbb{R}^+, \boldsymbol{R} \subseteq \boldsymbol{V})$

1 remove all edges from G that connect two terminals in R
2 compute a ρ-approximate Steiner tree T in G for R
3 **for** $r \in R$ **do**
4   **if** r is not a leaf in T
5   **then** make a star-replacement for r

Figure 3: A 2ρ-approximation algorithm for the terminal Steiner tree problem.

Proof. In line 1 of the algorithm all edges are removed from G that connect two terminals in R. As an optimal solution to the terminal Steiner tree problem must not contain such edges, this preprocessing step does not change the value of an optimal solution. In line 2 of the algorithm a ρ-approximation of a Steiner minimal tree for the preprocessed graph G is computed. Its length is a lower bound for the length of an optimal terminal Steiner tree in G. In lines 3–5 of the algorithm TerminalSteiner the tree T may be modified by star-replacement operations. Consider one such star-replacement at a terminal r with neighbors $a_1, \ldots, a_k$ in T (see Figure 2). The star-replacement replaces the edges $\{r, a_2\}, \ldots, \{r, a_k\}$ with the edges $\{a_1, a_2\}, \ldots, \{a_1, a_k\}$. Because the instance of the terminal Steiner tree problem is metric and edge $\{a_1, r\}$ is shortest from among the edges $\{r, a_i\}$ we have

$$\sum_{i=2}^{k} w(\{a_1, a_i\}) \;\leq\; \sum_{i=2}^{k} w(\{a_1, r\}) + w(\{r, a_i\}) \;\leq\; 2\sum_{i=2}^{k} w(\{r, a_i\}) \ .$$

As no two star-replacements involve the same edges, this proves that the solution returned by the algorithm TerminalSteiner has length at most 2ρ times the length of an optimum solution. □

Note that the running time of lines 2, 4, 5, and 6 of algorithm TerminalSteiner is linear. Therefore the total running time of this algorithm is dominated by the ρ-approximation algorithm used for the Steiner tree problem.

Acknowledgement

We are grateful to the anonymous referees for useful comments.

References

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, M. Szegedy, *Proof verification and hardness of approximation problems,* 33rd FOCS 1992, pp. 14–23.

[2] M. Bern, P. Plassmann, *The Steiner problem with edge lengths 1 and 2,* Information Processing Letters 32 (1989), pp. 171–176.

This paper appeared in: Information Processing Letters 89 (2004), 15–18

[3] Y.H. Chen, C.L. Lu, C. Y. Tang, *On the Full and Bottleneck Full Steiner Tree Problems,* In: Proceedings of the 9th Annual International Conference (COCOON 2003), Big Sky, MT, USA, July 25-28, 2003, LNCS 2697, T. Warnow, B. Zhu (Eds.), Springer, pp. 122–129

[4] M. Chlebík, J. Chlebíková, *Approximation Hardness of the Steiner Tree Problem on Graphs,* In: Proceedings of 8th Scandinavian Workshop on Algorithm Theory (SWAT'02) Turku, Finland, July 3-5, 2002, LNCS 2368, M. Penttonen, E. M. Schmidt (eds.), Springer Verlag, pp. 170–179.

[5] U. Feige, *A threshold of* $\ln n$ *for approximating set cover,* Journal of the Association for Computing Machinery 45 (1998), pp. 634–652.

[6] N. Garg, G. Konjevod, R. Ravi, *A polylogarithmic approximation algorithm for the group steiner tree problem,* Journal of Algorithms 37 (2000), pp. 66–84.

[7] C. Gröpl, S. Hougardy, T. Nierhoff, H.J. Prömel, *Approximation algorithms for the Steiner tree problem in graphs,* In: Steiner Trees in Industry, X. Cheng and D.-Z. Du (Eds.), Kluwer Academic Publishers 2001, pp. 235–279.

[8] F.K. Hwang, D.S. Richards, P. Winter, *The Steiner Tree Problem,* NorthHolland, New York 1992.

[9] G.-H. Lin, G. Xue, *On the Terminal Steiner Tree Problem,* Information Processing Letters 84 (2002), pp. 103-107.

[10] G. Robins, A. Zelikovsky, *Improved Steiner tree approximation in graphs,* In: Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms 2000, pp. 770–779.