This paper appeared in: Networks 47:2 (2006), 111-115

Lower Bounds for the Relative Greedy Algorithm for Approximating Steiner Trees

Stefan Hougardy Stefan Kirchner

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

Abstract. The Steiner tree problem is to find a shortest subgraph that spans a given set of vertices in a graph. This problem is known to be NP-hard and it is well known that a polynomial time 2-approximation algorithm exists. In 1996 Zelikovsky [11] suggested an approximation algorithm for the Steiner tree problem that is called the relative greedy algorithm. Till today the performance ratio of this algorithm is not known. Zelikovsky provided 1.694 as an upper bound and Gröpl, Hougardy, Nierhoff and Prömel [6] proved that 1.333 is a lower bound. In this paper we improve the lower bound for the performance ratio of the relative greedy algorithm to 1.385.

Keywords. approximation algorithms, analysis of algorithms, graph algorithms, Steiner tree problem, lower bounds

1 Introduction

The Steiner tree problem is to find a shortest subgraph that spans a given set of vertices in a graph. Karp [8] has shown that this problem is NP-hard. By a result of Bern and Plassmann [2] it follows from [1] that the Steiner tree problem in graphs is even APX-complete. This means that unless P=NP there exists some constant $c > 1$ such that no polynomial time approximation algorithm for the Steiner tree problem can exist that has a performance ratio smaller than c.

This paper appeared in: Networks 47:2 (2006), 111-115

It is well known that via the computation of a minimum spanning tree one can get an approximate solution for the Steiner tree problem in graphs that has at most twice the length of an optimal solution [5]. For a long time no better approximation algorithm for the Steiner tree problem in graphs was found until in 1990 Zelikovsky [10] suggested an algorithm with a performance ratio of 11/6. Nowadays several approximation algorithms for the Steiner tree problem are known that have a performance ratio better than two. The currently best one is due to Robins and Zelikovsky [9] and achieves a performance ratio less than 1.550.

In 1996 Zelikovsky [11] proposed an approximation algorithm for the Steiner tree problem that is called the relative greedy algorithm. He proved that this algorithm has a performance ratio better than 1.694. Till today it is not known what the precise performance ratio of Zelikovsky's algorithm is. Better knowledge of this value is of great interest as it might turn out that the relative greedy algorithm has a performance ratio below 1.550. Gröpl, Hougardy, Nierhoff and Prömel [6] provided 1.333 as a lower bound for this value. In this paper we will improve this lower bound to 1.385.

The paper is organized as follows. In Section 2 we introduce some general notations and in Section 3 we present details of the relative greedy algorithm. Section 4 contains our proof for the new lower bound. We close in Section 5 with some concluding remarks.

2 Preliminaries

Given a graph $G = (V, E)$ with edge lengths $w : E \to \mathbb{R}_+$ and a subset $R \subseteq V$ of *terminals* a *Steiner tree* is a tree in G that contains all vertices of R. The *length* of a Steiner tree is the sum of the lengths of the edges contained in it. A shortest possible Steiner tree is called a *Steiner minimal tree*. We denote it by SMT and its length by smt.

An approximation algorithm with *performance ratio* c is an algorithm that for all possible instances computes a solution that is at most by a factor c larger than the optimal solution. All known approximation algorithms for the Steiner tree problem that have a performance ratio better than 2 use the concept of *k-Steiner trees*. This idea was introduced by Zelikovsky [10]. A tree is *full*, if all leaves are terminals and all inner vertices are non-terminals. If a tree is not full it can be decomposed into several full subtrees called *full components* which are edge disjoint. If each of the full components has at most k leaves, these components form a k-Steiner tree for this set of terminals. A shortest possible k-Steiner tree is called a *minimal k-Steiner*

This paper appeared in: Networks 47:2 (2006), 111-115

tree and is denoted by SMT_k and its length by smt_k. Borchers and Du [3] have shown that for large enough k the ratio between smt and smt_k tends to 1:

Theorem 1 (Borchers and Du [3]) *For all instances of the Steiner tree problem holds:* $smt_k \leq \left(1 + \frac{1}{\log k}\right) \cdot smt$.

For a given graph $G = (V, E)$ with edge lengths $w : E \to \mathbb{R}_+$ and terminals $R \subseteq V$ we define the *terminal distance graph* as a complete graph with vertex set R. The length of an edge in the terminal distance graph is defined as the length of a shortest path between these two terminals in G. A minimal spanning tree in the terminal distance graph is denoted by MST and its length by mst. It is well known that $mst \leq 2 \cdot smt$ holds [5]. By replacing the edges of the MST in the terminal distance graph by the paths in G that defined the edge lengths one gets a Steiner tree in G which is a 2-approximation of the Steiner minimal tree in G.

3 The Relative Greedy Algorithm

The main idea of the relative greedy algorithm (see Figure 1) is to start with a Steiner tree that is obtained via the minimum spanning tree in the terminal distance graph. This solution is successively improved by adding certain Steiner minimal trees on at most k terminals of R and destroying the resulting cycles. Note that for a constant number of terminals a Steiner minimal tree can be computed in polynomial time [4].

Relative Greedy Algorithm $G = (V, E), R \subseteq V$
$i := 0$
while $mst(R/t_1, \ldots, t_i) \neq 0$
    choose $t_{i+1} \subseteq R, |t_{i+1}| \leq k$, so that
        $f_i(t_{i+1}) := \frac{smt(t_{i+1})}{mst(R/t_1,\ldots,t_i) - mst(R/t_1,\ldots,t_{i+1})}$ is minimal
    $i := i + 1$
$i_{max} := i$
$return = \bigcup_{i=1}^{i_{max}} SMT(t_i)$

Figure 1: The relative greedy algorithm

For a subset t_i of the terminals R we denote by $SMT(t_i)$ a Steiner minimal tree for these terminals and by $smt(t_i)$ its length. For subsets

This paper appeared in: Networks 47:2 (2006), 111-115

$t_1, t_2, \ldots, t_i$ of terminals we define $mst(R/t_1, \ldots, t_i)$ as the length of a minimal spanning tree in the terminal distance graph to which for each set $t_j, 1 \le j \le i$ one has added edges of length 0 completely connecting the terminals in t_j. For a tuple t_{i+1} of terminals the difference $mst(R/t_1, \ldots, t_i) - mst(R/t_1, \ldots, t_i, t_{i+1})$ is the change in length that is obtained by adding a Steiner minimal tree on terminals t_{i+1}. The relative greedy algorithm chooses in each iteration a tuple t_{i+1} that maximizes this difference in relation to the length $smt(t_{i+1})$, i.e. it chooses a tuple t that minimizes the function

$$f_i(t) := \frac{smt(t)}{mst(R/t_1, \ldots, t_i) - mst(R/t_1, \ldots, t_i, t)}. \tag{1}$$

The relative greedy algorithm outputs the union of the Steiner minimal trees of the chosen tuples t_i as the full components of the k-Steiner tree. We denote by RGA_k a solution returned by the relative greedy algorithm that is obtained by considering tuples of size at most k and by rga_k its length. Zelikovsky [11] proved that the relative greedy algorithm has a performance ratio of less than 1.694 for sufficiently large k.

4 Lower Bounds

In [6] a lower bound for the performance ratio of the relative greedy algorithm arbitrary close to $\frac{4}{3}$ was presented. First, we present a simpler example yielding a lower bound of $\frac{41}{30} - \epsilon$ which is extended in a second step to get a larger lower bound.

The fist instance G_k is shown in Figure 2. It contains $4k$ terminals $w_1, \ldots, w_{4k}$, three special vertices T_a, T_b, and T_c, $2k$ vertices $y_1, \ldots, y_{2k}$ and k vertices $z_1, \ldots, z_k$. Each terminal w_{4i-3} is connected to T_a and each terminal w_{4i} and the terminal w_1 are connected to T_b by an edge of length x (the value of x will be specified later). Moreover each vertex y_i is connected to the terminals w_{2i-1} and w_{2i} by edges of length 1, the vertices z_i are connected to y_{2i-1} and y_{2i} by edges of length 1/2 and the vertices z_i are connected to T_c by edges of length 1 for $1 \le i \le k$.

Lemma 1 *For every $\epsilon > 0$ and $x = \frac{21}{10} - \epsilon$ it holds* $\lim_{k\to\infty} \frac{rga_{4k}(G_k)}{smt_{4k}(G_k)} = \frac{41}{30} - \epsilon$.

Proof. The value of x will be chosen in such a way that the relative greedy algorithm is forced to choose the stars with centers T_a and T_b in the first two steps. In all further steps it will select full components of size 2 only. The Steiner minimal tree consists of the binary tree with root T_c and

This paper appeared in: Networks 47:2 (2006), 111-115

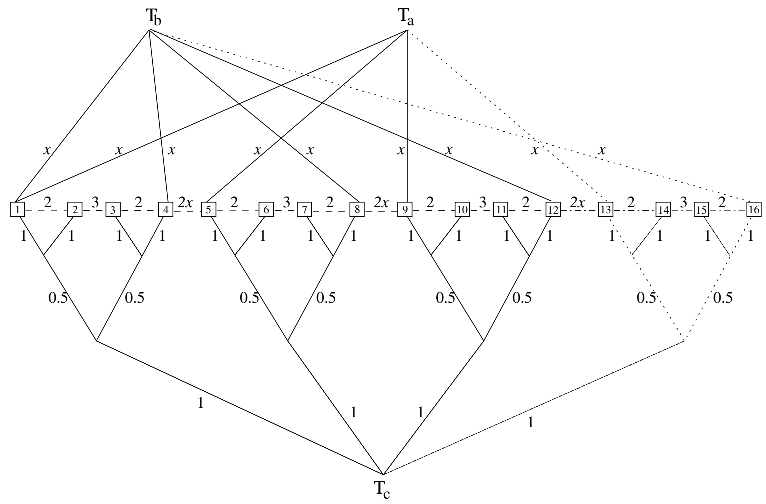

Figure 2: Lower bound ($G_k, k = 16$)

has length $6k$. In the following we will assume $x > 2$. This assumption is justified later.

The dashed edges in Figure 2 form a minimum spanning tree in the terminal distance graph and have length $7k + (k-1) \cdot 2x$.

Let t_a and t_b denote the set of terminals that are connected to T_a and T_b respectively. By t_c we denote the set of all terminals in G_k. We want the relative greedy algorithm to select the star with center T_a as the first full component. This requires that the inequality

$$f_0(t_a) = \frac{kx}{(k-1) \cdot 2x} \leq \frac{(k+1)x}{(k-1) \cdot 2x + 3} = f_0(t_b) \tag{2}$$

holds, which is true for $k \geq \frac{2x}{2x-3}$. In addition it must be the case that

$$f_0(t_a) = \frac{kx}{(k-1) \cdot 2x} \leq \frac{6k}{7k + (k-1) \cdot 2x} = f_0(t_c) \tag{3}$$

which is true for $k \geq \frac{12-2x}{5-2x}$. Note that for no other subset of terminals the function f_0 can be smaller than $f_0(t_a)$. This shows that the greedy algorithm chooses the star with center T_a in the first step. For the second step, using symmetry and monotonicity of the function f_i, it is enough to consider the subsets $t_1 = \{w_1, w_2, w_3, w_4\}, t_2 = t_1 \cup \{w_{4i-1} | 1 \leq i \leq k\} \cup \{w_{4i} | 1 \leq i \leq k\}, t_3 = t_2 \cup \{w_{4i-2} | 1 \leq i \leq k\}$ and t_b.

This paper appeared in: Networks 47:2 (2006), 111-115

The relative greedy algorithm chooses t_b in the second step if

$$\begin{aligned} f_1(t_b) = \frac{(k+1)x}{3k} &\leq \min\left(f_1(t_1), f_1(t_2), f_1(t_3)\right) \\ &= \min\left(\frac{5}{7}, \frac{3.5\cdot k+2.5}{5\cdot k+2}, \frac{5\cdot k+1}{7\cdot k}\right) \\ &= \frac{7k+5}{10k+4} \qquad \text{for } k \geq 15. \end{aligned} \tag{4}$$

If $\epsilon > 0$ and x is chosen as $\frac{21}{10} - \epsilon$ then for sufficiently large k the relative greedy algorithm will choose t_a and t_b in the first two steps. In all remaining steps the relative greedy algorithm chooses a full component of size two. Therefore the total length of the Steiner tree returned by the relative greedy algorithm is

$$k \cdot x + (k+1) \cdot x + 2k \cdot 2 = 4k + x + 2kx. \tag{5}$$

Setting $x = \frac{21}{10} - \epsilon$ we get as the ratio between the solution found by the relative greedy algorithm and the optimal solution:

$$\frac{4k + \frac{21}{10} - \epsilon + 2k(\frac{21}{10} - \epsilon)}{6k} = \frac{41}{30} - \frac{\epsilon}{3} + \frac{21 - 10\epsilon}{6k} \tag{6}$$

which tends to $\frac{41}{30} - \frac{\epsilon}{3}$ for $k \to \infty$. $\square$

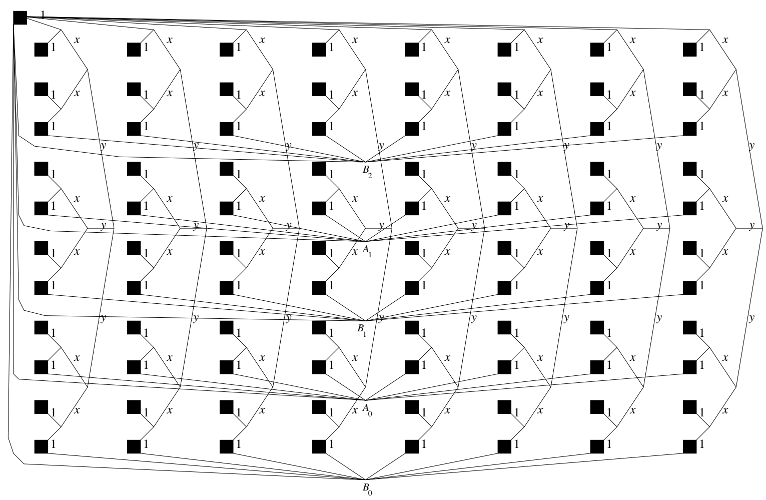

Figure 3: Improved lower bound ($G_{k,l}, k = 3, l = 8$)

This paper appeared in: Networks 47:2 (2006), 111-115

The second lower bound is obtained by constructing an instance $G_{k,l}$ which places the instance G_k into a grid. The instance $G_{k,l}$ consists of an $4k \times l$ grid of terminals where the last terminals of each column have been identified as one terminal. For each column of terminals the graph $G_k - T_b - T_c$ is added, while the terminals of each odd row (except for row $4k-1$) and the identified terminal are connected alternately by stars with center B_i and A_i (see Figure 3).

The edges which are incident to the vertices with degree k have weight y. The identified edges with weight 0.5 in G_k have weight x. The weights $a_i = a_i(k,l,x,y)$ and $b_i = b_i(k,l,x,y)$ of the edges incident to A_i resp. B_i will be specified later but assume here that $a_i > 1+x+y$, $\max\{a_i\} < \min\{b_i\}$ and the sequences a_i and b_i are monotone increasing. Every other edge has weight 1. Let $w_{i,j}$ denote the terminal in the ith column and jth row and $N(A_i)$ the leaves of the star a_i.

The idea is that in the instance $G_{k,l}$ the solution of the relative greedy algorithm consists of all stars with center A_i and B_i and all full components of size two. The optimal solution consists of the vertical trees. This will be proved in the following theorem.

Theorem 2 *The performance ratio of the relative greedy algorithm is not better than 1.385.*

Proof.

Since $b_j > a_i > 1 + x + y$ the minimum spanning tree has length

$$mst = ((6+2x)k + 2(1+x+y)(k-1))\, l. \tag{7}$$

The change in length of a minimun spanning tree after selection of a star $A_i \cup N(A_i)$ is $2(1+x+y)l$. Now we have to find the minimizing vertical component after the stars $(A_0 \cup N(A_0)) \cup \ldots \cup (A_{i-1} \cup N(A_{i-1}))$ have been selected. For a fixed h with $1 \le h \le l$ let $S_h = \{w_{i,h} | 1 \le i \le 4k\}$. For the ith step, using symmetry and monotonicity of the function f_i, it is enough to consider the subsets $S_h^{i,1} = S_h \backslash (\{w_{4j,h} | 1 \le j \le i\} \cup \{w_{4j-1,h} | 1 \le j \le i\})$ and $S_h^{i,2} = \{w_{j,h} | 4i+1 \le j \le 4k\}$.

For the selection function of $N(A_i)$ we get

$$\begin{aligned} f_i(N(A_i)) &= \frac{l+1}{l} \cdot \frac{a_i(k,l,x,y)}{2(1+x+y)} \le \min(f_i(S_h^{i,1}), f_i(S_h^{i,2})) \\ &= \min\left(\frac{(4+2x+y)k - (2+x)i}{\frac{mst}{l} - (4+2x)i}, \right. \end{aligned}$$

This paper appeared in: Networks 47:2 (2006), 111-115

$$\left.\frac{(k-i)(4+2x+y)}{(k-i)(6+2x)+(k-i-1)(2+2x+2y)}\right)$$
$$= \quad f_i(S_h^{i,1}). \tag{8}$$

Therefore we can specify the value of

$$a_i(k,l,x,y) = \frac{l}{l+1} \cdot \frac{2(1+x+y)\left((4+2x+y)k-(2+x)i\right)}{\frac{mst}{l}-(4+2x)i}. \tag{9}$$

Notice that a_i is monotone increasing. After all A-stars are selected the argument for selecting the stars with center B_i is similar. Only the change in length of a minimun spanning tree after selection of a star $B_i \cup N(B_i)$ has changed and is $2+2x$.

Again using monotonicity, the minimizing vertical component is

$$S_h^{i,3} = \{w_{4j+\kappa,h} | j \leq i < k-1, 1 \leq \kappa \leq 2\} \cup \{w_{4k,h}, w_{4k-1,h}\}$$

after every A-star and the stars $B_j \cup N(B_j)$ with $j < i < k-1$ have been contracted. For the selection function of $N(B_i)$ we get

$$\begin{aligned} f_{k-i+1}(N(B_i)) &= \frac{l+1}{l} \cdot \frac{b_i(k,l,x,y)}{2+2x} \\ &\leq f_{k-i+1}(S_h^{i,3}) \\ &= \frac{(k-i)(2+x+y)+2+x}{(k-i)(4+2x)+2}. \end{aligned} \tag{10}$$

Therefore we can specify the value of

$$b_i(k,l,x,y) = \frac{l}{l+1} \cdot 2(1+x) \cdot \frac{(k-i)\cdot(2+x+y)+2+x}{(k-i)\cdot(4+2x)+2}. \tag{11}$$

Notice for the last star $B_{k-1} \cup N(B_{k-1})$ the component $S_h^{k-1,3}$ consists of only four terminals and there is no edge with costs y in $SMT(S_h^{k-1,3})$. So we get

$$b_{k-1}(k,l,x,y) = \frac{l}{l+1} \cdot 2(1+x) \cdot \frac{2+x}{3+x}. \tag{12}$$

Notice that b_i is monotone increasing.

After every star $A_i \cup N(A_i)$ and $B_i \cup N(B_i)$ is selected the algorithm has to choose full components of size two. This can only be satisfied if $4k < \min((2+2x+y)k+1, (3+2x)k)$, because otherwise the algorithm

This paper appeared in: Networks 47:2 (2006), 111-115

chooses the components $\{w_{2i,h} | 1 \leq i \leq 2k\} \cup \{w_{4k,h}\}$ resp. $\{w_{4i+j,h} | 0 \leq i < k, 2 \leq j \leq 4\}$.

If every contraint is satisfied we get a lower bound for the performance ratio of

$$\frac{4kl + (l+1)\sum_{i=0}^{k-2} a_i + (l+1)\sum_{i=0}^{k-1} b_i}{(4+2x+y)kl}. \tag{13}$$

This is the case for the values $k = 5, l \geq 15, x = 0.8688$ and $y = 0.063$ and so we get a lower bound of 1.385383. □

5 Concluding Remarks

Further improvement on the performance ratio for the relative greedy algorithm is still an open problem and the gap between our new lower bound 1.385 and 1.694 remains large. Especially it would be very interesting to know whether the relative greedy algorithm has a performance ratio better than 1.550.

References

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, M. Szegedy, *Proof verification and hardness of approximation problems*, J. ACM 45 (1998), 501–555.

[2] M. Bern, P. Plassmann, *The Steiner problem with edge lengths 1 and 2*, Inform. Process. Lett. 32 (1989), 171–176.

[3] A. Borchers, D.-Z. Du, *The k-Steiner ratio in graphs*, SIAM J. Comput. 26 (1997), 857–869.

[4] S. Dreyfus, R. Wagner, *The Steiner problem in graphs*, Networks 1 (1972), 195–207.

[5] E.N. Gilbert, H.O. Pollak, *Steiner minimal trees*, SIAM J. Appl. Math. 16 (1968), 1–29

[6] C. Gröpl, S. Hougardy, T. Nierhoff, H.J. Prömel, *Lower bounds for approximation algorithms for the Steiner tree problem*, In: Proceedings of 27th International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2001, Springer LNCS, 2204, 217–228.

This paper appeared in: Networks 47:2 (2006), 111-115

[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.), 235–279, Kluwer Academic Publishers, 2001

[8] R.M. Karp, *Reducibility among combinatorial problems*, In: Complexity of Computer Computations, (Proc. Sympos. IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., 1972). New York: Plenum 1972, 85–103.

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

[10] A. Zelikovsky, *An 11/6-approximation algorithm for the network Steiner problem*, Algorithmica 9 (1993), 463–470.

[11] A. Zelikovsky, *Better approximation bounds for the network and Euclidean Steiner tree problems*, technical report CS-96-06, University of Virginia, 1996.