1.757 and 1.267-Approximation Algorithms for the Network and Rectilinear Steiner Tree Problems

Marek Karpinski* Alexander Zelikovsky†

Abstract

The Steiner tree problem requires to find a shortest tree connecting a given set of terminal points in a metric space. We suggest better and fast heuristics for the Steiner problem in graphs and in rectilinear plane with the record worst-case performance ratios 1.648 and 1.267, respectively.

1 Introduction

Consider a metric space with a distance function d. For any set of *terminal* points S one can efficiently find MST(S), a minimum spanning tree of S. Let mst(S, d) be the cost of this tree in metric d. A Steiner tree is a spanning tree of a superset of the terminal points (the extra points are called Steiner points). It was already observed by Pierre Fermat that the cost of a Steiner tree of S may be smaller than mst(S, d). The Steiner tree problem asks for the Steiner minimum tree, that is, for the least cost Steiner tree. However, finding such a tree is NP-hard for almost all interesting metrics, like Euclidean, rectilinear, Hamming distance, shortest-path distance in a graph etc. Because these problems have many applications, they were subject of extensive research [12].

In the last two decades many approximation algorithms for finding Steiner minimum trees appeared. The quality of an approximation algorithm is measured by its performance ratio: an upper bound of the ratio between the achieved length and the optimal length.

The Network Steiner tree problem (NSP) asks for the Steiner minimum tree for a vertex subset $S \subset V$ of a graph $G(V, E, d)$ with cost function d on edges E.

In the *rectilinear* metric, the distance between two points is the sum of the differences of their $x-$ and $y-$coordinates. The rectilinear Steiner tree problem (RSP) got recently new importance in the development of techniques for VLSI routing [13].

The most obvious heuristic for the Steiner tree problem approximates a Steiner minimum tree of S with MST(S). While in all metric spaces the performance ratio of this heuristic is at most 2 [15] (it can be implemented for NSP in time $O(|E| + |V| \log |V|)$ [14]), Hwang [10, 11] proved that this heuristic in the rectilinear plane has the performance ratio exactly 1.5 and can be implemented in time $O(|S| \log |S|)$.

Zelikovsky [16, 18] and Berman/Ramaiyer [2] gave two better heuristics for NSP. Perfomance ratios of these heuristics are $\frac{11}{6} \approx 1.84$ and $\frac{16}{9} \approx 1.78$ and their runtimes are $O(|S|(|E|+|V| \log |V|)+|V||S|^2)$ and $O(\alpha + |V|^2|S|^{3.5})$, respectively. Here α means time complexity of finding of all pairs shortest paths.

In the recent paper Berman *et al* [3] gave a more precise (than in the first papers [17, 2]) analysis of the performance ratio of these heuristics for RSP. They proved that their performance ratios are at most 1.3125 and $\frac{61}{48} \approx 1.271$, respectively. The parametrized versions of these heuristics have a runtime $O(n \log^2 n)$ [3, 7].

Here we present a new heuristic which adds a prelimenary phase to Berman/Ramaiyer's heuristic. This heuristic decreases the known performance ratios by $\frac{1}{48} \approx 2\%$ for NSP and achieves $\frac{19}{15} \approx 1.266$ for RSP. Moreover, this improvement can be achieved in the same order of runtime.

*Department of Computer Science, Universität Bonn, Römerstraße 164, 53117 Bonn, Germany, e-mail: marek@cs.uni-bonn.de. Research partially supported by the International Computer Science Institute, Berkeley, California, by the DFG Grant KA 67314, by the ESPRIT BR Grant 7097 and by ECUS030

†Institute of Mathematics, Akademiei 5, Kishinev, 277028, Moldova, email: 17azz@mathem.moldova.su. Research partially supported by Volkswagen Stiftung. Parts of this work were done in Max-Planck-Institut für Informatik, Saarbrücken, Germany

In the next section we provide a synopsis of Berman/Ramaiyer's approach In Sections 3 we describe our new heristic and derive some estimates for its performance ratios. Sections 4 and 5 deals with the applications of this heuristic to NSP and RSP, respectively.

2 Berman/ Ramaiyer's Heuristic

A Steiner tree T of a set of terminals S is *full* if every internal node of T is a Steiner point, i.e., not a terminal. If T is not full, it can be decomposed into full Steiner trees for subsets of terminals that overlap only at leaves. Such subtrees are called *full Steiner components* of T [9]. A full Steiner tree with k terminals is named *k-tree*.

The method described here can be applied with an arbitrary metric d. Without loss of generality, we may assume that the metric d on the set of terminals S is the shortest-path distance for the weighted edges D connecting S. This way, MST(D) is the minimum spanning tree of the graph $< S, D >$, we denote this tree with MST(D), and its cost with mst(D). If we increase the set of edges D by some extra edges, say forming a set E, the shortest-path distance may decrease; MST($D \cup E$) is the minimum spanning tree for the modified metric.

Let z be a set of k terminals (*k-tuple*). Let $T(z)$ be the minimum k-tree with the terminal set z, $d(z)$ is the cost of $T(z)$ and $Z(z)$ is a spanning tree of z consisting of some sufficiently short edges, i.e. MST($D \cup Z(z)$) contains $Z(z)$.

At first, assume that $Z(z) = Z_0(z)$ consists of zero-cost edges. If we decide to use $T(z)$ as a part of that tree, the remaining part can be computed optimally as MST($D \cup Z_0(z)$), from which we remove zero-cost edges of $Z_0(z)$. The improvement of the tree cost due to this decision is the *gain* of z, denoted $g(z, D)$. It is easy to see that $g(z, D)$ = mst(D)-mst($D \cup Z_0(z)$)-$d(z)$.

We denote by $t_r = \max\{mst(D \cup E) : g(z, D \cup E) \leq 0$ for any $z \subset S$, $|z| \leq r\}$. In other words, t_r denote the the maximum possible MST-cost if any k-tuple, $k \leq r$ has a nonpositive gain. Let t_2 be the length of MST(D) and $s = t_\infty$ be the length of optimal Steiner tree. It was proved that $t_3 \leq \frac{5}{3}s$ [16], $t_4 \leq \frac{3}{2}s$ [1] and $t_r \to s$ while $r \to \infty$ [6] for arbitrary metrics. For the rectilinear metric, $t_r \leq \frac{2k}{2k-1}$ for $r \geq 3$, moreover, $t_2 + t_4 \leq \frac{5}{2}s$ and $3t_2 + 4t_3 \leq 9s$ [3].

Before we describe Berman/Ramaiyer's heuristic (BRk) [2], we have to look closer at the way how to obtain MST($D \cup Z(z)$) from M = MST(D). Say that $Z(z) = \{e_1, ..., e_i\}$. When e_1 is inserted, the longest edge e'_1 in the path joining the ends of e_1 with cost c'_1 is removed from M. Then we do the same with e_2 and so on.

The idea of BR is to make the initial choices (performed in the *Evaluation Phase*) tentative, and to check later (in the *Selection Phase*) for better alternatives.

Evaluation Phase. Initially, M = MST(D)and b_2 denotes its cost. For every triple z considered, find $g = g(z, M)$. If $g \leq 0$, z is simply discarded. Otherwise we do the following for every edge e of some spanning tree $Z(z)$: find e' and c', make the cost of e equal to $c - g$, replace in M edge e' with e, put e in a set B_{new} and e' in B_{old}. Once this spanning tree of z is processed, we place the tuple $< z, B_{new}, B_{old} >$ on a Stack (for the future inspection in the second phase). Repeat this while there are triples with positive gain. For later analysis, we define b_3 to be the cost of M at this point, continue the process with quadruples and get b_4 as the cost of M, and so on till all k-tuples being processed.

Selection Phase. We initialize $D = M$. Then we repeatedly pop $< z, B_{new}, B_{old} >$ from the *Stack*, and insert B_{old} to D. If $B_{new} \subseteq$MST(D), then the correspondig minimum i-tree $T(z)$ is placed in a *List*, otherwise we remove all edges of B_{new} from D.

All i-trees, $i = 3, ..., k$, from *List* with the rest of MST-edges form the output Steiner tree of BRk. Its length is at most

$$b_2 - \sum_{i=3}^{k} \frac{b_{i-1} - b_i}{i-1} = \sum_{i=2}^{k-1} \frac{b_i}{i(i-1)} + \frac{b_k}{k-1}.$$

It is easy to see that $b_i \leq t_i$, $i = 2, 3, ...$. Therefore, BRk has the following upper bound on the output cost:

$$t_2 - \sum_{i=3}^{k} \frac{t_{i-1} - t_i}{i-1} = \sum_{i=2}^{k-1} \frac{t_i}{i(i-1)} + \frac{t_k}{k-1}. \tag{1}$$

3 Combined algorithm

Berman/Ramaiyer's heuristic tries to find tuples of terminals with the largest possible total gain. But every time it accepts a k-tree, it also accepts all its Steiner points. This may increase the cost of the cheapest solution achievable at the current step. The main idea of our heuristic is to minimize this possible increase.

Let τ be a k-tree and $V(\tau)$ be its Steiner point set. A forest $\tau' \subset \tau$ is called *spanning* if for any $v \in V(\tau)$, there is a path in τ' connecting v with S. The cost of the minimum spanning forest in τ is called a *loss* of τ and denoted by $l(\tau)$. The value $g'(\tau) = g(\tau) - l(\tau)$ will be called a *relative gain* of τ. A relative gain of a k-tuple z is the maximum relative gain of a k-tree on terminals of z.

Below we describe a *combined algorithm* CA(l,k), which uses the notion introduced. It consists of two applications of Berman/Ramayer algorithm with papameters l and k.

At first we apply the algorithm BRl but for the relative gain function instead of the usual gain function. (We denote this algorithm BRl*). Actually, we use only the evaluation and selection phases of BRl. As an output we obtain a *List* of selected i-trees, $i = 3, ..., l$. Then we extend the initial terminal set S adding all Steiner points of i-trees from *List*. Now we apply usual BRk to the modified terminal set S'.

It is easy to see that the minimum spanning forest for any k-tree can be found exactly by the greedy algorithm. So finding the k-trees of maximum gain or maximum relative gain for a k-tuple has the same time complexity. Moreover, any k-tuple with positive relative gain has a positive usual gain. This implies

Remark 1 *The combined algorithm C(l,k) can be implemented in the same order of runtime as BRm, where* $m = \max\{l, k\}$.

In the rest of the paper we derive performance ratios claimed for the combined algorithm.

Let t_k and t'_k denote the output Mst-cost of the evaluation phase of BRk applied to the terminal set S and S', respectively. Note that the bound (1) for BRk can be represented in the following way:

$$t_2 - \sum_{i=3}^{k} \frac{t_{i-1} - t_i}{i-1} = \frac{t_2}{2} + \sum_{i=3}^{k-1} \frac{t_i}{(i-1)i} + \frac{t_k}{k-1} = \sum_{i=3}^{k-1} \frac{t_2 + t_i}{(i-1)i} + \frac{t_2 + t_k}{k-1} \tag{2}$$

Denote by G and L the total gain and loss of all trees of *List*, respectively. Also, $G' = G - L$. Note, that $t'_2 = t_2 - G$, $t'_i \le t_i + L$ and, therefore, $t'_2 + t'_i \le t_2 + t_i - G'$. Let $t^l_2 = t_2 - G'$. Thus, (2) implies the following performance ratio for the combined algorithm:

$$\sum_{i=3}^{k-1} \frac{t^l_2 + t_i}{(i-1)i} + \frac{t^l_2 + t_k}{k-1} = \frac{t^l_2}{2} + \sum_{i=3}^{k-1} \frac{t_i}{(i-1)i} + \frac{t_k}{k-1}. \tag{3}$$

Note, that the bound (3) for the combined algorithm beats the bound (2) for usual BRk by the value $G'/2$. Since G' might be zero, we will estimate the value t^l_2 directly.

Denote by t^i the output Mst-cost of the evaluation phase of BRi*, e.g. $t^2 = t_2$. Then, similarly to the usual BRl, we obtain

$$t^l_2 \le t^2 - \sum_{i=3}^{l} \frac{t^{i-1} - t^i}{i-1}$$

The last inequality shows that we need to bound t^i. Note that a relative gain of any triple cannot be positive, i.e. $t^3 = t^2 = t_2$. Moreover,

$$t^4_2 \le t^2 - \frac{t^2 - t^4}{3} = \frac{2}{3}t^2 + \frac{1}{3}t^4, \tag{4}$$

since $3G' = t^2 - t^4$ for this case.

To bound the values of t^i, $i \ge 4$, we use the following property of the output MST of the evaluation phase of BRi*:

(i) for any i-tuple τ, $\quad g(\tau) \le l(\tau)$.

Theorem 1 *Let t^k be the MST-length for an instance of the Steiner tree problem such that $g(\tau) \leq l(\tau)$ for any k-tree τ. Then*

$$t^k \leq \frac{3}{2} t_k$$

Proof. Let T_i be a full component of an optimal k-restricted Steiner tree T and T_i span a subset S_i of the whole terminal set S. We transform such a component to the form of the complete binary tree by replicating certain vertices, so that copies of the same vertex are connected with zero-cost edges.

The loss of T_i can be bounded in the following way. For any inner vertex of T_i, choose the shorter edge among two edges going to its two children. It is easy to see, that the forest F obtained spans all inner vertices of T_i. $d(F)$ is at most half of $d(T_i)$, since F contains exactly half of all edges of T_i and $T_i - F$ contains longer edges. This means, that $l(T_i) \leq \frac{1}{2} d(T_i)$.

Now, $mst(S_i) - d(T_i) = g(T_i) \leq l(T_i) \leq \frac{1}{2} d(T_i)$ and $mst(S_i) \leq \frac{3}{2} d(T_i)$. Therefore, $t^k = mst(S) \leq \sum mst(S_i) \leq \sum \frac{3}{2} d(T_i) = \frac{3}{2} d(T) = \frac{3}{2} t_k$. $\diamondsuit$.

The next section shows how to use the last bounds to obtain $1.648 + \epsilon$-approximation algorithm for STP in graphs. Unfortunately, this algorithm has an impractical runtime for $\epsilon < 0.2$.

Of course, tight bounds for t^i depend on metric space. The sections 4 and 5 deal with the cases of the Steiner tree problem in graphs and rectilinear metric. We will prove that the tight bounds for t^4 are $\frac{15}{8}$ and $\frac{7}{5}$ for NSP and RSP, respectively. These bounds lead to the practical approximation algorithms with the performance guarantee 1.757 and 1.267 for NSP and RSP, respectively.

4 The Steiner Trees in Graphs

Theorem 2 *Given an instance of the Steiner tree problem in graphs, if for any 4-tree τ, $g(\tau) \leq l(\tau)$, then the minimum spanning tree cost is at most $15/8$ of the minimal Steiner tree cost.*

Proof. We may prove Theorem for each full Steiner component separately. We transform such a component to the form of the complete binary tree by replicating certain vertices, so that copies of the same vertex are connected with zero-cost edges. Note that all terminals are leaves of this tree.

Let k be the depth of this tree. We label its vertices with words from $B^* = \{\alpha \in B^* : |\alpha| \leq k\}$, where $B = \{0, 1\}$. Let ρ be the root and α have children $\alpha 0$, $\alpha 1$. The set of terminals with the common anchestor α is denoted by α also.

Some more denotations: Let $s = s(\rho)$ denote the cost of the Steiner minimal tree, $t = t(\rho)$ be the cost of MST for the whole terminal set, $s_i(\alpha) = \sum_{|\beta|=i, b \in B} d(\alpha\beta, \alpha\beta b)$, $H = H(\rho) = s_0(\rho) + s_1(\rho)$, $P(\alpha)$ denote the cost of the cheapest path from α to S.

An *average path cost* is defined to be

$$\bar{P} = \bar{P}(\rho) = \frac{\sum_{i=1}^{k-1} 2^{k-i} s_i(\rho)}{2^k} = \sum_{i=1}^{k-1} 2^{-i} s_i(\rho)$$

This cost has the following two obvious properties:

$$\bar{P}(\alpha) \geq P(\alpha) \tag{5}$$

$$2\bar{P}(\alpha) = s_0(\alpha) + \bar{P}(\alpha 0) + \bar{P}(\alpha 1). \tag{6}$$

Since $\bar{P} \geq \frac{H}{4}$, the following inequality is slightly stronger than Theorem.

$$t \leq 2s - 2\bar{P} - \frac{s - H}{8} \tag{7}$$

We will prove (7) by induction on k. Indeed, for $k \leq 2$, (7) is trivially true. Let (7) be true for all trees of depth at most k. We will prove it for a tree of depth $k+1$ (Fig. 1).

Further assume that $s_1(0) \geq s_1(1)$.

Now we partition $s(\rho)$ into five subtrees:

$$s(\rho) = \sum_{\alpha \in A} s(\alpha) + D,$$

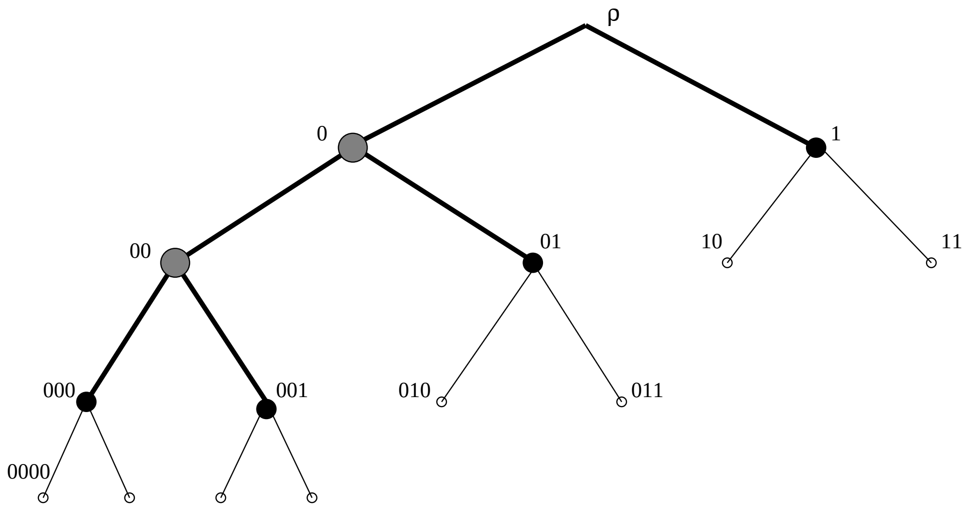

Figure 1: A full component

where $\alpha \in A = \{000, 001, 01, 1\}$ and $D = s_0(\rho) + s_0(0) + s_0(00)$ (thick lines on Fig. 1).

These five parts correspond to some spanning tree:

$$t(\rho) \leq \sum_{\alpha \in A} t(\alpha) + t', \tag{8}$$

where t' is the cost of three cheapest edges connecting four MST for the sets $\alpha \in A$. By induction, inequality (7) holds for every $\alpha \in A$:

$$t(\alpha) \leq 2s(\alpha) - 2\bar{P}(\alpha) - \frac{s(\alpha) - H(\alpha)}{8} \tag{9}$$

Substituting (9) into (8) we obtain

$$t(\rho) \leq 2(s - D) - 2\sum_{\alpha \in A} \bar{P}(\alpha) - \sum_{\alpha \in A} \frac{s(\alpha) - H(\alpha)}{8} + t'$$

and, therefore,

$$t(\rho) - (2s - 2\bar{P} - \frac{s - H}{8}) \leq t' + 2\bar{P} + \frac{s - H}{8} - 2D - 2\sum_{\alpha \in A} \bar{P}(\alpha) - \sum_{\alpha \in A} \frac{s(\alpha) - H(\alpha)}{8}.$$

To prove (7) it is sufficient to show that the RHS of the last inequality is nonpositive, which is equivalent to the following inequality

$$\frac{1}{8}\left(s - H - \sum_{\alpha \in A}(s(\alpha) - H(\alpha))\right) \leq 2D + 2\sum_{\alpha \in A} \bar{P}(\alpha) - (t' + 2\bar{P}) \tag{10}$$

Claim 1 *The RHS of (10) is at least* $\bar{P}(0) - d(0, 00)$.

Proof. Consider an arbitrary 4-tree q with Steiner points 0 and 00 and four terminals achievable from 000, 001, 01 and 1, respectively. Note, that $t' \leq t(q)$, where $t(q) = d(q) + g(q)$ is the cost of three corresponding longest edges on paths connecting treminals of q. Let terminals of q be the nearest to the corresponding vertices of A. Since $g(q) \leq l(q) \leq d(0, 00) + P(00)$, we obtain

$$t' \leq D + \sum_{\alpha \in A} P(\alpha) + d(0, 00) + P(00)$$

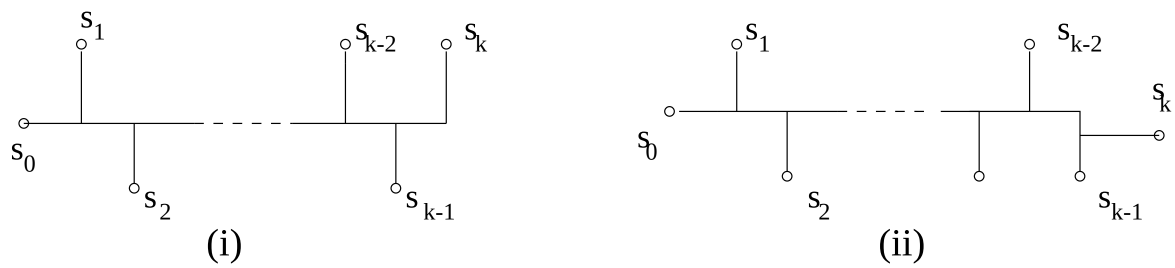

Figure 2: Two types of a full component

Now Claim can be proved straitforward using the properties (5) and (6) of the average path cost:

$$2D + 2\sum_{\alpha\in A} \bar{P}(\alpha) - (t' + 2\bar{P}) \geq$$

$$2D + 2\sum_{\alpha\in A} \bar{P}(\alpha) - (D + \sum_{\alpha\in A} P(\alpha) + d(0,00) + P(00) + s_0(\rho) + \bar{P}(0) + \bar{P}(1)) \geq$$

$$s_0(0) + s_0(00) + \bar{P}(000) + \bar{P}(001) + \bar{P}(01) - P(00) - \bar{P}(0) - d(0,00) \geq \bar{P}(0) - d(0,00) \quad \diamondsuit$$

The LHS of (10) equals to

$$\frac{1}{8}(D + \sum_{\alpha\in A} H(\alpha) - H) = \frac{1}{8}(s_1(1) + s_0(01) + s_1(01) + s_0(00) + s_1(00) + s_2(00))$$

By Claim and our assumption of $s_0(00) + s_0(01) = s_1(0) \geq s_1(1)$, (10) follows from the following inequality

$$\frac{1}{8}(2s_0(01) + s_1(01) + 2s_0(00) + s_1(00) + s_2(00)) \leq \bar{P}(0) - d(0,00) \tag{11}$$

Similarly, the corresponding partition of the Steiner minimal tree induced by the 4-tree with Steiner points 0 and 01 implies that it is sufficient to prove

$$\frac{1}{8}(2s_0(00) + s_1(00) + 2s_0(01) + s_1(01) + s_2(01)) \leq \bar{P}(0) - d(0,01) \tag{12}$$

Thus to prove (7) we may show that one of the inequalities (11) or (12) is true. This follows from the fact that their sum is true. Indeed, summing (11) and (12) we obtain

$$\frac{1}{8}(4s_0(00) + 2s_1(00) + s_2(00) + 4s_0(01) + 2s_1(01) + s_2(01)) \leq 2\bar{P}(0) - s_0(0) = \bar{P}(00) + \bar{P}(01),$$

which trivially follows from the definition of the average path cost. $\diamondsuit$

Theorem 1, bounds (3) and (4) imply

Theorem 3 *The output cost of CA(4,k) is bounded with the value which is smaller than the bound (2) for BRk by*

$$\frac{T_2 - T^4}{6} = \frac{1}{48}s,$$

where T_2 and T^4 are the upper bounds for t_2 and t^4, s is the cost of the optimal Steiner tree.$\diamondsuit$

The bounds for t_3 and t_4 imply

Corollary 1 *The performance ratio of CA(4,4) is at most $\frac{253}{144} \approx 1.757$.* $\diamondsuit$

5 Approximating Rectilinear Steiner Trees

Hwang [10] proved that there is a Steiner minimum tree where every full component has one of the shapes shown in Fig. 2. It was suggested in [3] some partition of a full component into so called Steiner segments. Below we breifly describe this useful technique.

Let $a_1, \ldots, a_k$ and $b_0 = 0, b_1, \ldots, b_k$ be the lengths of horizontal and vertical lines of a full Steiner component F with terminals $s_0, \ldots, s_k$. The horizontal lines form its *spine*. Moreover, in case (i) $b_k < b_{k-2}$ holds. In case (ii) assume that $b_k = 0$. Consider the sequences $b_0, b_1, b_3, \ldots, b_{2i+1}, \ldots$ and $b_0, b_2, \ldots, b_{2i}, \ldots$. Let

$$b_{h(0)} = b_0, b_{h(1)}, \ldots, b_{h(p+1)} = b_k \tag{13}$$

be the sequence of local minima of these sequences, i.e. $b_{h(j)-2} \geq b_{h(j)} < b_{h(j)+2}$. If $h(p) = k-1$, we exclude the member $b_{h(p)}$ from (13). For the case of $h(j+1) = h(j)+1$, $(j = 1, \ldots, p-1)$, we exclude arbitrarily either $b_{h(j+1)}$ or $b_{h(j)}$. So, we get $h(j+1) - h(j) \geq 3$. The elements of the refined sequence (13) are called *hooks*. Further we assume that a full Steiner tree nontrivially contains at least 4 terminals ($k \geq 4$). A *Steiner segment* K is a part of a full Steiner component bounded by two sequential hook terminals. So two neighbouring Steiner segments have a common hook. K contains the two furthest terminals below and above the spine called *top* and *bottom*, respectively.

Now we present the main result of this section.

Theorem 4 *Given an instance of the Steiner tree problem in rectilinear plane, if for any 4-tree τ, $g(\tau) \leq l(\tau)$, then the minimum spanning tree cost is at most $7/5$ of the minimal Steiner tree cost.*

Proof. Further assume that some terminals are connected with short edges such that $g(\tau) \leq l(\tau)$ for any 4-tree τ. It is sufficient to prove Theorem for a full Steiner component F with a terminal set Set. Let $F = \cup_{i=0}^{k} K_i$ be a partition of F into Steiner segments. Then $d(F) = \sum_{i=0}^{k} d(K_i) - \sum_{i=1}^{k-1} h_i$, where h_i are hooks. Consider some Steiner segment $K = K_i$ of F with terminal set $S = S_i$, hooks $hl = h_i$ and $hr = h_{i+1}$ and the length $s = d(K)$. Similarly to Section 4, denote the MST-length for a terminal set X by $t(X)$. We intend to prove that

$$t(S) - s \leq \frac{2}{5}s - \frac{7}{10}(hl + hr) \tag{14}$$

This inequality yields Theorem, since then

$$t(Set) \leq \sum_{i=0}^{k} t(S_i) \leq \frac{7}{5}\sum_{i=0}^{k} d(K_i) - \frac{7}{10}\sum_{i=0}^{k-1}(h_i + h_{i+1}) \leq$$

$$\frac{7}{5}\left(\sum_{i=0}^{k} d(K_i) - \sum_{i=1}^{k-1} h_i\right) = \frac{7}{5}d(F)$$

Let *top* of K be to the left of its *bottom*. We partition S into three parts $S = L \cup C \cup R$, where L is the set of terminals from the left hook till the first before top, C contains all terminals from the the first before top till the next after bottom and R contains ones from the next after bottom till the right hook. Similarly, we partition F into three corresponding parts

$$s = left + center + right,$$

where $center$ contains all edges spanning C, and $left$ and $right$ consists of the rest of the Steiner segment to the left and right of $center$ (Fig. 3). Denote by vl and vr the lengths of two vertical lines which bound $center$ from the left and the right. Note that K should contain $center$, but $left$ and $right$ might be empty.

We have two cases depending on the size of $center$.

Case 1. Let *bottom* be the next to *top* (Fig. 4). It was noticed in [3] that

Lemma 1 *There are two trees (Fig. 4(i)) Top (dashed lines) and Bot (dotteded lines) spanning terminals of K with a total length*

$$d(Top) + d(Bot) = 3s - 2(hl + hr) - Rest;$$

$Rest$ sums the lengths of the thin drawn Steiner tree lines.

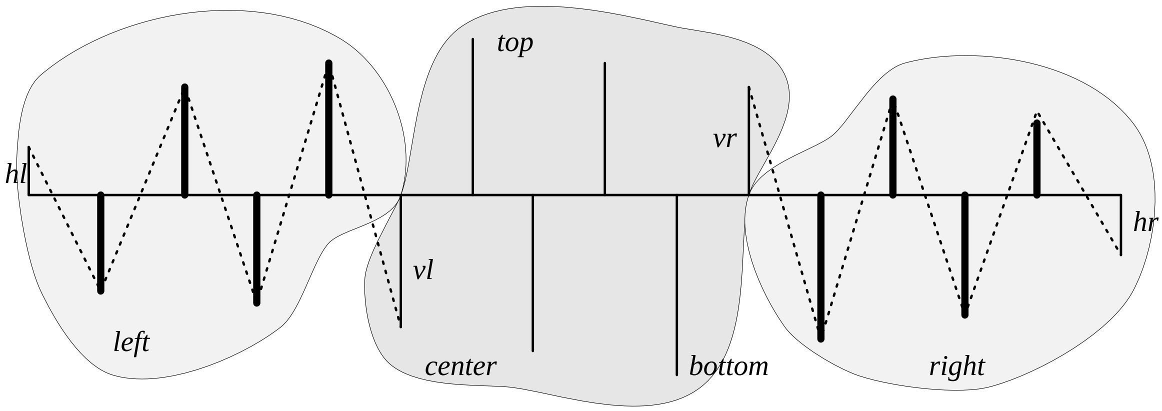

Figure 3: The partition of the Steiner segment

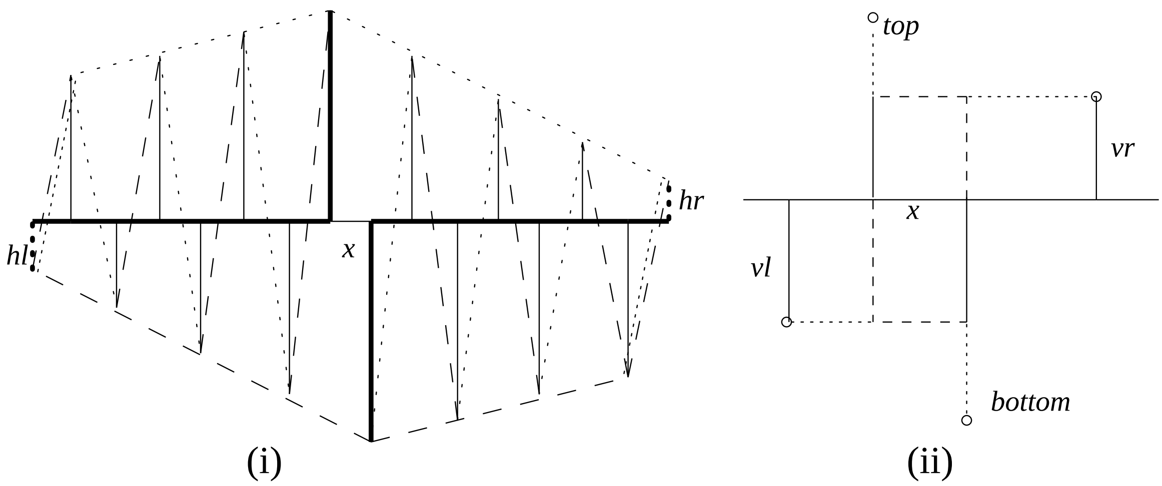

Figure 4: *top* besides *bottom*: the whole segment (i) and its *center* (ii)

Lemma 1 says that $t \le \frac{3}{2}s - \frac{Rest}{2} - (hl + hr)$. It is easy to see that (14) holds if $Rest$ is big enough, i.e. $Rest \ge \frac{s}{5} - \frac{3}{5}(hl + hr)$. So further assume that

$$Rest \le \frac{s}{5} - \frac{3}{5}(hl + hr). \tag{15}$$

We may span R and L with the alternative chains (Fig. 3), therefore,

$$t(L) + t(R) \le left + right + Rest - x, \tag{16}$$

where x is the horisontal edge length of $Rest$.

Let q be the quadruple with terminals from C (Fig. 4 (ii)). Theorem assumes that $g(q) = t(C) - center$ is at most $l(q)$. But the loss of q is at most x plus the length of the shortest among four dotted lines (we may shift the central edge up or down till dashed lines). Therefore,

$$t(C) - center \le l(q) \le x + \frac{center - (2vl + 2vr + x)}{4} \le x + \frac{s - Rest - (hl + hr)}{4} \tag{17}$$

Thus, we can prove (14) using (15), (16), (17):

$$t(S) - s = (t(C) - center) + (t(L) - left + t(R) - right) \le x + \frac{s - Rest - (hl + hr)}{4} + Rest - x \le$$

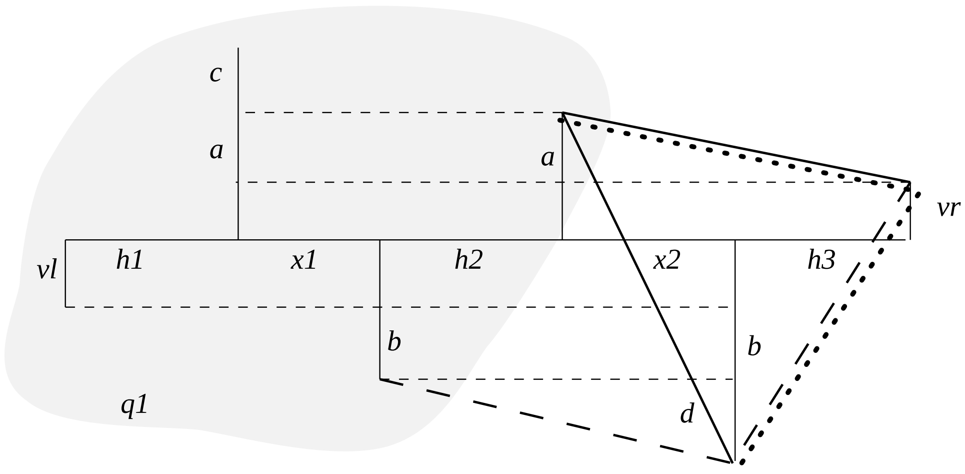

Figure 5: 2 terminals between top and $bottom$

$$\frac{s}{4}+\frac{3}{4}Rest-\frac{hl+hr}{4}\leq\frac{s}{4}+\frac{3}{4}(\frac{s}{5}-3\frac{hl+hr}{5})-\frac{hl+hr}{4}=\frac{2}{5}s-\frac{7}{10}(hl+hr)$$

Case 2. Let two terminals lie between top and $bottom$. Now $center$ contains two quadruples $q1$ and $q2$ with central edges $x1$ and $x2$ (Fig. 5). We construct 5 spanning trees for the set C. Three trees contain some connection of the quadruple $q1$ and pairs of edges spanning the last two terminals: thick dotted, dashed, and solid lines, respectively. Theorem assumes that the connection of the quadruple $q1$ cannot be longer the length of $q1$ (Steiner edges in the dark region) plus the loss of $q1$. Denote by $light$ the length of Steiner edges out of the dark region. Then

$$T1-center\leq d(q1)+l(q1)+light+a+h3-center=l(q1)+a+h3\leq x1+c+a+h3$$

$$T2-center\leq l(q1)+h2+d\leq h1+b+h2+d$$

$$T3-center\leq l(q1)+2a+x2\leq x1+b+2a+x2$$

The last pair of trees is symmetric to $T1$ and $T2$

$$T4-center\leq l(q2)+b+h1\leq x2+d+b+h1$$

$$T5-center\leq l(q2)+h2+c\leq h3+a+h2+c$$

Summing all inequalities we obtain

$$5t(C)-5center\leq 2center-6(vl+vr) \tag{18}$$

If there are more terminals between top and $bottom$ then $center$ contains several quadruples q_i. Three necessary spanning trees contain connections of odd qudruples and two contain connections of even quadruples. Similarly, we obtain (18) using the Theorem assumption that such connections are no longer than $d(q_i)+l(q_i)$.

To prove (14), we will show that

$$5(t(L)+t(R))-5(left+right)\leq 2(left+right)-4(hl+hr)+6(vl+vr),$$

which means for the right side of the Steiner segment

$$5t(R)-5right\leq 2right-4hr+6vr \tag{19}$$

If vr is the right hook ($vr=hr$), then (19) is trivial, since $t(R)=right=0$.

If the hook is the next after vr (Fig. 6(i)), then we use the solid line five times and two times replace the edge of $T1$ and $T2$ (the thick dashed line) with the dotted line. In the latter case we replace vr and hr with f, the horizontal edge length. Thus, we obtain $5t(R)-5right\leq 5vr+2f-2hr\leq 2right-4hr+6vr$.

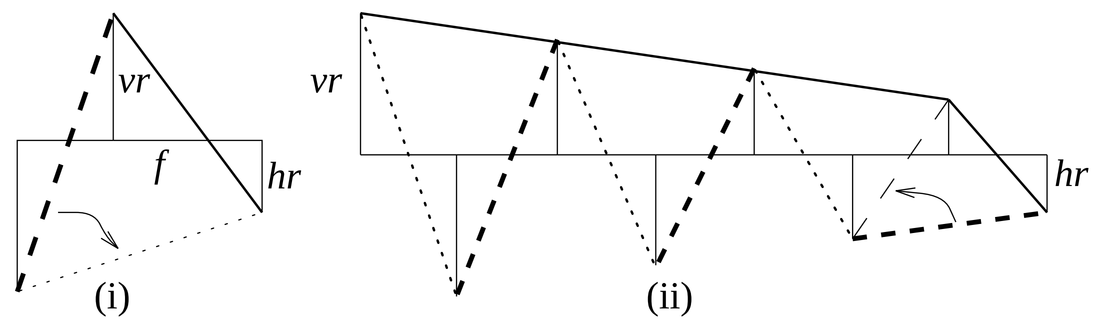

Figure 6: The short (i) and the long (ii) *right*

For a nontrivial R we use the following 5 trees (Fig. 6(ii)) which contain:
(1) thick solid and dotted lines. It doubles vr and Steiner tree lines crossed by its dotted lines.
(2-3) thick solid and dashed lines or the thin dashed line if the hook is above the spine (2 times). It doubles the Steiner tree lines crossed by its edges and saves the hook hr.
(4-5) the alternative chain (Fig. 3) (2 times). It doubles all vertical lines except vr and hr.

Thus, these trees double $right - hr$ at most two times, vr only once, and save hr two times. $\diamondsuit$

Theorem 4, bounds (3) and (4), inequalities $3t_2+4t_3 \leq 9s$, $t_2+t_4 \leq \frac{5}{2}s$ imply that the performance guarantee of the algorithm CA(4,4) can be bounded with the following value

$$\frac{\frac{2}{3}t_2 + \frac{1}{3}t^4 + t_3}{6} + \frac{t_2 + t_4}{3} \leq \frac{\frac{3}{4}t_2 + \frac{1}{4}t^4 + t_3}{6} + \frac{t_2 + t_4}{3} =$$

$$\frac{3t_2 + 4t_3}{24} + \frac{t_2 + t_4}{3} + \frac{t^4}{24} \leq \frac{3}{8}s + \frac{5}{6}s + \frac{7}{120}s = \frac{19}{15}s$$

Theorem 5 *The performance guarantee of CA(4,4) is at most* $\frac{19}{15} \approx 1.2667$. $\diamondsuit$

6 Open problems

The main open question remaining for the Network Steiner Tree Problem is to compute the exact value of a constant c which separates polynomial approximability from nonapproximability (NP-hardness) of this problem. Such a constant c must exist since NSP is SNP-complete [4]. We conjecture that c lies somewhere below 1.7 for that problem. Note that we do not know at the moment whether RSP is also SNP-complete, and therefore it could have a polynomial time approximation scheme.

References

[1] P. Berman, V. Ramaiyer. *An approximation algorithm for the Steiner tree problem* . TR #CS-91-05, The Pennsylvania State University, 1991.

[2] P. Berman, V. Ramaiyer. *Improved approximations for the Steiner tree problem.* In Proc. of 3d ACM-SIAM Symp. on Discrete Algorithms, 325–334, 1992.

[3] P. Berman, U. Fößmeier, M. Karpinski, M. Kaufmann, A. Zelikovsky. *Approaching the 5/4–Approximations for Rectilinear Steiner Trees.* LNCS 855, 60–71, 1994.

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

[5] A. Borchers and D. Du. *The k-Steiner Ratio in Graphs* To appear in STOC'1995.

[6] D. Du, Y. Zhang, Q. Feng. *On better heuristic for Euclidean Steiner minimum trees.* 32nd IEEE Symp. on Found. of Comp. Science, 431–439, 1991.

[7] U. Fößmeier, M. Kaufmann, A. Zelikovsky. *Faster Approximation Algorithms for the Rectilinear Steiner Tree Problem.* LNCS 762, 533-542, 1993.

[8] M. R. Garey, D. S. Johnson. *The Rectilinear Steiner Problem is NP-Complete.* SIAM J. Appl. Math., 32, 826-834, 1977.

[9] E. N. Gilbert, H. O. Pollak. *Steiner Minimal Trees.* SIAM Appl. Math., 16, 1-29, 1968.

[10] F. K. Hwang. *On Steiner Minimal Trees with Rectilinear Distance.* SIAM J. Appl. Math., 30, 104-114, 1976.

[11] F. K. Hwang. *An $o(n \log n)$ algorithm for rectilinear minimum spanning trees.* J. ACM 26, 177–182, 1979.

[12] F. K. Hwang, D. Richards, P. Winter. *The Steiner Tree Problem.* Annals of Disc. Math. 53, 1992.

[13] B.Korte, H.J.Prömel, A.Steger. *Steiner Trees in VLSI-Layouts.* In Korte et al.: Paths, Flows and VLSI-Layout, Springer, 1990.

[14] K.Mehlhorn. *A faster approximation algorithm for the Steiner problem in graphs.* Inf. Process. Lett. 27: 125–128, 1988.

[15] H. Takahashi, A. Matsuyama. *An approximate solution for the Steiner problem in graphs.* Math. Japonica, 24: 573–577, 1980.

[16] A. Z. Zelikovsky. *An 11/6-approximation Algorithm for the network Steiner Problem* Algorithmica, 9: 463–470, 1993.

[17] A. Z. Zelikovsky. *An 11/8-approximation Algorithm for the Steiner Problem on Networks with Rectilinear Distance.* In Coll. Math. Soc. J. Bolyai 60: 733-745, 1992.

[18] A. Z. Zelikovsky. *A faster approximation algorithm for the Steiner Tree Problem in Graphs.* Inf. Process. Lett. 46: 79-83, 1993.

[19] A. Z. Zelikovsky. *Better approximation bounds for the network and Euclidean Steiner tree problems* 1995.