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 $n(S, d)$ 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 $n(S, d)$. The Steiner tree problem asks for the Steiner minimum tree, that is, for the least cost Steiner tree. However, finding such a tree is $\text{NP}$-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 $S \subseteq V$ 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. Performance ratios of these heuristics are $\frac{11}{6} \approx 1.8333$ and $\frac{15}{8} \approx 1.875$ and their runtimes are $O(|S| |E| + |V| \log |V|)$ and $O(|S| + |V| |S|^2) + O(\alpha + |V|^2 |S|^3.5)$, respectively. Here $\alpha$ 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 $\frac{61}{32} \approx 1.90625$, 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 preliminary phase to Berman/Ramaiyer’s heuristic. This heuristic decreases the known performance ratios by $\frac{1}{\sqrt{2}} \approx 0.707$ for NSP and achieves $\frac{12}{11} \approx 1.091$ for RSP. Moreover, this improvement can be achieved in the same order of runtime.

*Department of Computer Science, Universität Bonn, Rümerstraße 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 673/14, by the ESPRIT CR Grant 7007 and by ECU9030
†Institute of Mathematics, Academiai 5, Budapest, 1117, Hungary, e-mail: 17arz@mathem.ualmohov.hu. 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 heuristic 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, $\text{MST}(D)$ is the minimum spanning tree of the graph $< S, D >$, we denote this tree with $\text{MST}(D)$, and its cost with $\text{cost}(D)$. If we increase the set of edges $D$ by some extra edges, say forming a set $E$, the shortest-path distance may decrease; $\text{MST}(D \cup E)$ 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., $\text{MST}(D \cup Z(z))$ contains $Z(z)$.

At first, assume that $\bar{Z}(z) = Z(\emptyset)$ 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 $\text{MST}(D \cup Z(\emptyset))$, from which we remove zero-cost edges of $\bar{Z}(\emptyset)$. 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 $g(z, D) = \text{MST}(D) - \text{MST}(D \cup Z(\emptyset)) - d(z)$.

We denote by $t_r = \max\{\text{msf}(D \cup E) : g(z, D \cup E) \leq 0 \text{ for any } z \subseteq S, |z| \leq r\}$. In other words, $t_r$ denotes the the maximum possible MST-cost if any $k$-tuple, $k \leq r$, has a non-positive gain. Let $t_2$ be the length of $\text{MST}(D)$ and $s = t_\infty$ be the length of optimal Steiner tree. It was proved that $t_2 \leq \frac{2}{3}s [16]$, $t_3 \leq \frac{3}{8}s [1]$ and $t_r \to s$ while $r \to \infty [6]$ for arbitrary metrics. For the rectilinear metric, $t_2 \leq \frac{1}{8}\sqrt{s}$. Moreover, $t_2 + t_3 \leq \frac{1}{3}s$ 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 $\text{MST}(D \cup Z(z))$ from $M = \text{MST}(D)$. Say that $\bar{Z}(z) = \{e_1, \ldots, e_i\}$. When $e_1$ is inserted, the longest edge $e_i'$ in the path joining the ends of $e_i$ with cost $c'$ 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, $M = \text{MST}(D)$ and $b_2$ denotes its cost. For every triple $z$ considered, find $g = g(z, M)$. If $g < 0$, $z$ is simply discarded. Otherwise we do the following for every edge $e$ of some spanning tree $Z(z)$: find $e'$ and $\tilde{d}$, make the cost of $e$ equal to $\tilde{d}$, replace in $M$ edge $e'$ with $e$, put $e$ in a set $B_{new}$ and $e'$ in $B_{del}$. Once this spanning tree of $z$ is processed, we place the tuple $< z, B_{new}, B_{del} >$ 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 $b_2$ to be the cost of $M$ at this point, continue the process with quadruples and get $b_3$ as the cost of $M$, and so on till all $k$-tuples being processed.

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

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

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

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

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

1
3 Combined algorithm

Berman/Ramayjer’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 \( \tau \) 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 \( \tau' \) connecting \( v \) with \( S \). The cost of the minimum spanning forest in \( \tau \) is called a loss of \( \tau \) and denoted by \( l(\tau) \). The value \( g'(\tau) = g(\tau) - l(\tau) \) will be called a relative gain of \( \tau \). A relative gain of a \( k \)-tuple \( \tau \) is the maximum relative gain of a \( k \)-tree on terminals of \( \tau \).

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

At first we apply the algorithm BRI but for the relative gain function instead of the usual gain function. (We denote this algorithm BRI*). Actually, we use only the evaluation and selection phases of BRI. 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 BRI 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 CA\((l,k)\) can be implemented in the same order of runtime as BRI, where \( m = \max\{l, k\} \).

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

Let \( t_2 \) and \( t_k \) denote the output Mst-cost of the evaluation phase of BRI applied to the terminal set \( S \) and \( S' \), respectively. Note that the bound (1) for BRI can be represented in the following way:

\[
t_2 - \sum_{i=3}^{k} \frac{t_i - t_{i-1}}{i-1} = \frac{t_2}{2} + \sum_{i=3}^{k} \frac{t_i}{i-1} + \frac{t_k}{k-1} - \frac{2 - t_2 - t_1}{2} - \frac{t_k}{k-1}. (2)
\]

Denote by \( G \) and \( L \) the total gain and loss of all trees of List, respectively. Also, \( G' = G - L \). Note, that \( t_2' = t_2 - G \), \( t_k' \leq t_k + L \) and, therefore, \( t_2' + t_k' \leq t_2 + t_k - G' \). Let \( t_2' = t_2 - G' \). Thus, (2) implies the following performance ratio for the combined algorithm:

\[
\frac{t_2}{t_2'} = \frac{t_2 - \sum_{i=3}^{k} \frac{t_i - t_{i-1}}{i-1}}{\frac{t_2}{2} + \sum_{i=3}^{k} \frac{t_i}{i-1} + \frac{t_k}{k-1} - \frac{2 - t_2 - t_1}{2} - \frac{t_k}{k-1}}. (3)
\]

Note, that the bound (3) for the combined algorithm beats the bound (2) for usual BRI by the value \( G'/2 \). Since \( G' \) might be zero, we will estimate the value \( t_2' \) directly.

Denote by \( t_2^* \) the output Mst-cost of the evaluation phase of BRI*, e.g. \( t_2^* = t_2 \). Then, similarly to the usual BRI, we obtain

\[
t_2' \leq t_2^* - \sum_{i=3}^{k} \frac{t_i - t_{i-1}}{i-1}
\]

The last inequality shows that we need to bound \( t_2^* \). Note that a relative gain of any triple cannot be positive, i.e. \( \frac{t_i^*}{t_i} = \frac{t_i}{t_i} \). Moreover,

\[
t_2^* \leq t_2^* - \frac{t_2^* - t_4}{3} = \frac{2}{3} t_2^* + \frac{1}{3} t_i,
\]

since \( 3G' = t_2^* - t_i^* \) for this case.

To bound the values of \( t_i^* \), \( i \geq 4 \), we use the following property of the output MST of the evaluation phase of BRI*:

(i) for any \( i \)-tuple \( \tau \), \( g(\tau) \leq l(\tau) \).
**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 \( \tau \). Then

\[
\frac{3}{2} t_k \leq t^k
\]

**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, \( t^k = mst(S) \leq \sum mst(S_i) \leq \sum \frac{3}{2} d(T_i) = \frac{3}{2} d(T) = \frac{3}{2} t_k \).

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

Of course, tight bounds for \( t^k \) 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 \( t^k \) 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 \( \tau \), \( g(\tau) \leq l(\tau) \), 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 \( B^* = \{ a \in B^* : |a| \leq k \} \), where \( B = \{0,1\} \). Let \( \rho \) be the root and \( \alpha \) have children \( \alpha_0, \alpha_1 \). The set of terminals with the common ancestor \( \alpha \) is denoted by \( \alpha \).

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, \( s_i(\alpha) = \sum_{d\beta(\alpha), \beta(b) \in B} d(\alpha \beta, \alpha \beta b) \), \( H(\rho) = s_0(\rho) + s_1(\rho) \), \( P(\alpha) \) denote the cost of the cheapest path from \( \alpha \) to \( S \).

An average path cost is defined to be

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

This cost has the following two obvious properties:

\[
\bar{P}(\alpha) \geq P(\alpha) \tag{5}
\]

\[
2\bar{P}(\alpha) = s_0(\alpha) + \bar{P}(\alpha 0) + \bar{P}(\alpha 1).
\]

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:

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

4
where $\alpha \in A = \{000, 001, 01, 1\}$ and $D = s_\emptyset(\rho) + s_\emptyset(0) + s_\emptyset(00)$ (thick lines on Fig. 1).

These five parts correspond to some spanning tree:

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

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 $\alpha \in A$:

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

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

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

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 terminals 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

$$t' \leq D + \sum_{\alpha \in A} P(\alpha) + d(0, 00) + P(00).$$
Now Claim can be proved straightforward using the properties (5) and (6) of the average path cost:

$$2D + 2 \sum_{\alpha \in A} \bar{P}(\alpha) - (t' + 2\bar{P}) \geq 0$$

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

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

The LHS of (10) equals to

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

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

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

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

**Theorem 3**  The output cost of $CA(4,k)$ is bounded with the value which is smaller than the bound \(^{(2)}\) for $BR(k$ by

$$\frac{T_3 - T^4}{6} = \frac{1}{48}s,$$

where $T_3$ and $T^4$ are the upper bounds for $t_3$ and $t^4$, $s$ is the cost of the optimal Steiner tree.  \hfill \diamondsuit

The bounds for $t_3$ and $t^4$ imply

**Corollary 1**  The performance ratio of $CA(4,k)$ is at most $\frac{263}{156} \approx 1.757$.  \hfill \diamondsuit

Figure 2: Two types of a full component

(i) \hspace{2cm} (ii)
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 briefly 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 \( b_0, b_1, b_2, \ldots, b_{p+1}, \ldots \) and \( b_0, b_2, \ldots, b_{2i}, \ldots \). Let

\[
b_{h(i)} = b_0, b_{h(1)}, \ldots, b_{h(p+1)} = b_k
\]

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 \( h(j) - h(j) \geq 3 \). 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 \( \tau \), \( g(\tau) \leq l(\tau) \), 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 \( \tau \). It is sufficient to prove Theorem for a full Steiner component \( F \) with a terminal set \( S \). Let \( F = \bigcup_{i=0}^{k} K_i \) be a partition of \( F \) into Steiner segments. Then \( d(F) = \sum_{i=0}^{k} d(K_i) \). As the length \( s = d(K) \) is defined to be the MST-length for a terminal set \( X \) by \( t(X) \). We intend to prove that

\[
t(S) - s \leq \frac{2}{5} s - \frac{7}{10} (hl + hr)
\]

This inequality yields Theorem, since then

\[
t(S) \leq \frac{2}{5} \sum_{i=0}^{k} d(S_i) \leq \frac{7}{10} \sum_{i=0}^{k} d(K_i) - \frac{7}{10} \sum_{i=0}^{k-1} (h_i + h_{i+1}) \leq
\]

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

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

\[
Rest \leq \frac{s}{5} - \frac{3}{5}(hl + hr). \quad (15)
\]

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

\[
t(L) + t(R) \leq left + right + Rest - x, \quad (16)
\]

where \( x \) is the horizontal 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,

\[
t(C) - center \leq l(q) \leq x + center - (2vl + 2vr + x) \leq x + \frac{s - Rest - (hl + hr)}{4} \quad (17)
\]

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

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

Figure 3: The partition of the Steiner segment

Figure 4: \( top \) besides \( bottom \): the whole segment (i) and its \( center \) (ii)
\[ \frac{s}{4} + \frac{3}{4} \text{Rest} - \frac{hl + hr}{4} \leq \frac{s}{4} + \frac{3}{5} \left( \frac{3}{5} - \frac{hl + hr}{5} \right) - \frac{hl + hr}{4} = \frac{7}{10} (hl + hr) \]

**Case 2.** Let two terminals lie between **top** and **bottom**. Now **center** contains two quadruples \( q_1 \) and \( q_2 \) with central edges \( x_1 \) and \( x_2 \) (Fig. 5). We construct 5 spanning trees for the set \( C \). Three trees contain some connection of the quadruple \( q_1 \) and pairs of edges spanning the last two terminals: thick dotted, dashed, and solid lines, respectively. Theorem assumes that the connection of the quadruple \( q_1 \) cannot be longer the length of \( q_1 \) (Steiner edges in the dark region) plus the loss of \( q_1 \). Denote by **light** the length of Steiner edges out of the dark region. Then

\[
T_1 - \text{center} \leq d(q_1) + l(q_1) + \text{light} + a + h_3 - \text{center} = l(q_1) + a + h_3 \leq x_1 + c + a + h_3
\]

\[
T_2 - \text{center} \leq l(q_1) + h_2 + d \leq h_1 + b + h_2 + d
\]

\[
T_3 - \text{center} \leq l(q_1) + 2a + x_2 \leq x_1 + b + 2a + x_2
\]

The last pair of trees is symmetric to \( T_1 \) and \( T_2 \)

\[
T_4 - \text{center} \leq l(q_2) + b + h_1 \leq x_2 + d + b + h_1
\]

\[
T_5 - \text{center} \leq l(q_2) + h_2 + c \leq h_3 + a + h_2 + c
\]

Summing all inequalities we obtain

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

If there are more terminals between **top** and **bottom** then **center** contains several quadruples \( q_i \). Three necessary spanning trees contain connections of odd quadruples 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

\[
5(t(L) + l(R)) - 5(\text{left} + \text{right}) \leq 2(\text{left} + \text{right}) - 4(hl + hr) + 6(\text{vl} + \text{vr}),
\]

which means for the right side of the Steiner segment

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

If \( vr \) is the right hook (\( vr = hr \)), then (19) is trivial, since \( t(R) = \text{right} = 0 \).

If the hook is the next after \( vr \) (Fig. 6(i)), then use the solid line five times and twice times replace the edge of \( T_1 \) and \( T_2 \) (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

\[
5t(R) - 5\text{right} \leq 5vr + 2f - 2hr \leq 2\text{right} - 4hr + 6vr.
\]
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. $\diamond$

Theorem 4, bounds (3) and (4), inequalities $3l_2+4l_3 \leq 9s$, $t_2+t_4 \leq \frac{3}{2}s$ imply that the performance guarantee of the algorithm CA(4,4) can be bounded with the following value

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

$$\frac{3}{24}t_2 + \frac{4}{3}t_3 + \frac{3}{24}t_2 + \frac{3}{24}t_4 + \frac{3}{24} \leq \frac{3}{2}s + \frac{5}{6}s + \frac{7}{120}s = \frac{19}{15}s$$

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

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 NSP is also SNP-complete, and therefore it could have a polynomial time approximation scheme.

References


