On Approximation of the Power-p and Bottleneck Steiner Trees

Piotr Berman* Alexander Zelikovsky†

Abstract

Many VLSI routing applications, as well as the facility location problem involve computation of Steiner trees with non-linear cost measures. We consider two most frequent versions of this problem. In the power-p Steiner problem the the cost is defined as the sum of the edge length raised to power p, while in the bottleneck Steiner problem the cost is the maximum of the edge lengths.

In contrast to the classical Steiner problem, the objective of the power-p and bottleneck Steiner tree problems is to minimize the sum of the edge lengths raised to the p power or the maximum edge length, respectively. We show that the both problems are NP-hard and, moreover, the bottleneck Steiner trees cannot be approximated with the factor less than 2, unless P = NP. We prove that the minimum spanning tree only constant times worse than the power-p Steiner tree in any metric space. In particular, for $p = 2$, we show that the minimum spanning tree is at most 23.3 times worse than the optimum and construct an instance when it is 17.2 times worse. We also present a better approximation algorithm for the bottleneck Steiner problem with performance guarantee $\log_2 n$, where n is the number of terminals (the minimum spanning tree can be $2\log_2 n$ times worse than the optimum).

1 Introduction

Many VLSI routing applications, such as routing to minimize estimates of electrical delay [3, 13, 8, 2] as well as the facility location [14, 4, 9] problem involve computation of Steiner trees in which the weight of an edge is a nonlinear function of its length. Formally, Steiner trees can be defined as follows. Given a complete graph $G = (V, E, d)$ with nonnegative edge lengths d, and a set of *terminals* S, a tree T within G is called a *Steiner tree* if T spans S and any non-terminal vertex of T (called a *Steiner point*) has degree at least 3. The *power-p Steiner tree problem* (p-STP) seeks for a Steiner tree that minimizes the sum of the edge lengths each raised to the p power. We also consider the *bottleneck Steiner tree problem* (BSTP) which asks for a Steiner tree that minimizes the maximum length of an edge.

Although the both problems have been extensively considered in literature, no complexity results have been established. In the next section, we show that the both problems are hard even to approximate. We prove that p-STP is MAX SNP-hard and the bottleneck Steiner trees cannot be approximated with the factor less than 2, unless P = NP.

*Department of Computer Science and Engineering, Penn State University, University Park, PA 16802. Email: berman@cse.psu.edu

†Computer Science Department, UCLA, 3713 Boelter Hall, Los Angeles, CA 90095-1596. Email: alexz@cs.ucla.edu. Research partially supported by Volkswagen Stiftung and Packard Foundation.

It is not difficult to see that, similarly to the classical Steiner tree problem, if Steiner points are not allowed, then the minimum spanning tree is the best Steiner tree for the both power-p and bottleneck Steiner problems (see [5]). It is well known that the minimum spanning tree cannot be longer than twice the length of the shortest Steiner tree ([15]), in other words, the *Steiner ratio* (the supremum of the minimum spanning tree cost over optimal cost) is at most 2. In [5], it was conjectured that the *power-p Steiner ratio* (i.e. the Steiner ratio for the p-STP) is finite. In Section 3 we prove that this conjecture is true, i.e. that the power-p Steiner ratio ρ_p is constant-bounded for any p. In particular, for $p = 2$, we show that the minimum spanning tree is at most 23.3 times worse than the optimum. But we do not think that this bound is tight, since the worst instance of 2-STP that we construct has the Steiner ratio 17.2.

In [6], it was proved that the minimum spanning tree heuristic for the bottleneck Steiner tree problem is $2\log_2|S|$. In the last section, we present a better approximation algorithm Its performance ratio is two times better, i.e. $\log_2|S|$.

2 Complexity

Theorem 1 *The power-p Steiner Problem is MAX SNP hard.*

Proof. We will construct an approximation preserving reduction from the problem of finding a maximum independent set in 3-regular graphs (see [12]). Let $G = (V, E)$ be a 3-regular graph. We create an instance of the power-p Steiner problem consisting of a graph $G' = (V', E', d)$ and a terminal set $S \subseteq V$ as follows:

(1) For each edge $e \in E$, we put into correspondence a terminal $e \in S$. Also S contains an auxiliary node θ, i.e. $S = E \cup \{\theta\}$.

(2) For each node $v \in V$ and three edges $e_1, e_2, e_3 \in E$ adjacent to v (see Figure 1(a)), we put into correspondence two nodes $v_1, v_2 \in V' \setminus S$ such that

$$d(e_i, v_j) = d(e_2, v_1) = d(v_1, v_2) = d(e_l, v_2) = d(v_2, \theta) = 1$$

(3) For each pair of edges $e_i, e_j \in E$ (see Figure 1(b)), we put into correspondence a node $e_{ij} \in V' \setminus S$ such that

$$d(e_i, e_{ij}) = d(e_j, e_{ij}) = 1, \quad d(e_{ij}, \theta) = 2$$

(4) All other edges have cost 2.

Note that any 5-tuple of edges (2) allows to attach e_i, e_j and e_l to θ with cost 5. Any triple of edges (3) allows to attach e_i and e_j to θ with cost $2 + 2^p = 2(1 + 2^{p-1})$.

Let a Steiner tree uses two 5-tuples corresponding to two nodes, say $u, v \in V$. If u and v are adjacent, i.e. $e = (u, v) \in E$, then we would get a 6-cycle with the nodes θ, u_2, u_1, e, v_1, and v_2. This allows to translate a Steiner tree into an independent set of G: we select the nodes that define 5-tuples used in the tree.

Now suppose that $|E| = 3m$ and that we have a Steiner tree that contains k such 5-tuples. These 5-tuples attach $3k$ elements of E to θ, and remaining edges attach the remaining $3(m - k)$ elements. It is easy to see that there is no better way of attaching the remaining elements than using the triples described above, with the cost of $3(m - k)(1 + 2^{p-1})$. If we ignore the fact that $3(m - k)$ may be odd, the total cost will be

$$3m(1 + 2^{p-1}) - k(3(1 + 2^{p-1}) - 5) \;=\; m(3 \times 2^{p-1} + 3) - k(3 \times 2^{p-1} - 2) \qquad (1)$$

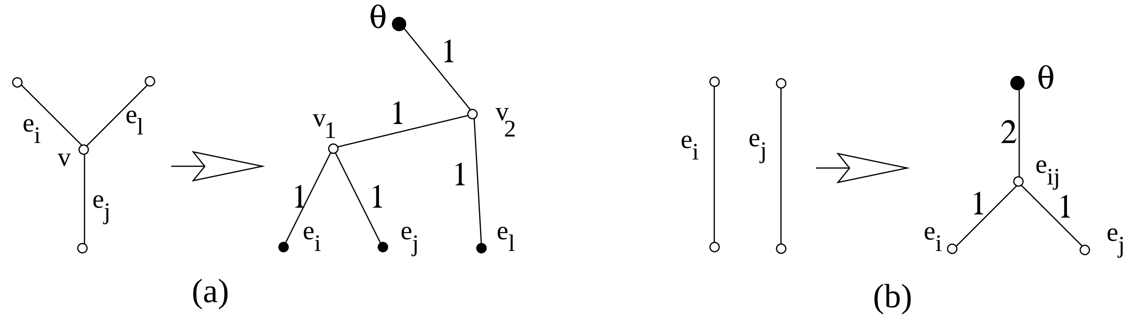

Figure 1: Reduction from the independent set problem for 3-regular graphs to pSTP. The terminals are denoted by filled circles. A node with 3 adjacent edges is transformed into a 5-tuple (a). Any pair of edges is transformed into a 3-star (b)

Let k^* be the maximal size of an independent set in the original graph. Note that because this graph is 3-regular and has $2k$ nodes, $k/2 \leq k^* \leq k$. Suppose that we have a Steiner tree, with k complete 5-tuples, that costs $1+\epsilon$ times the optimum. This translates into an independent set with k nodes. Equation (1) implies that

$$\begin{aligned}
& m(3 \times 2^{p-1}+3) - k(3 \times 2^{p-1}-2) \ \leq [m(3 \times 2^{p-1}+3) - k^*(3 \times 2^{p-1}-2)](1+\epsilon) \\
\iff \quad & k^*(3 \times 2^{p-1}-2)(1+\epsilon) - m(3 \times 2^{p-1}+3)\epsilon \ \leq \ k(3 \times 2^{p-1}-2) \\
\iff \quad & k \ \geq \ k^* \left(1+\epsilon - \frac{m}{k^*}\frac{3\times 2^{p-1}+3}{3\times 2^{p-1}-2}\epsilon\right)
\end{aligned}$$

Because $m/k^* \leq 2$ and $(3 \times 2^{p-1}+3)/(3 \times 2^{p-1}-2) \leq 6$

$$k \ \geq \ k^*(1+\epsilon-12\epsilon) \ = \ k^*(1-11\epsilon)$$

This shows that our reduction preserves the approximability. □

Theorem 2 *The bottleneck Steiner problem cannot be solved with the performance ratio less than 2 unless P unequal NP.*

Proof. It suffices to translate X3C (exact cover by 3-sets, see [10]). An instance to this problem is a set with some $3m$ elements, and a collection of 3-element subsets. The task is to find a collection of m disjoint subsets, i.e. to cover all the elements with a disjoint collection of subsets. Given an instance with set of elements V and set of triples E, we create an instance of the bottleneck Steiner problem consisting of a graph $G' = (V', E', d)$ and a terminal set $S \subseteq V$ similarly to the previous reduction (see Figure 1(a)).

(1) The set of terminals will be $V \cup \{\theta\}$, where θ is an auxiliary terminal.

(2) For each triple $t = \{u_1, u_2, u_3\}$ we create nodes t_1 and t_2 with a 5-tuple of edges, each having cost 1:

$$d(u_1,t_1) = d(u_2,t_1) = d(t_1,t_2) = d(u_3,t_2) = d(t_2,\theta) = 1.$$

(3) All other edges have cost 2.

It is easy to see that the bottleneck edge cost is 1 if and only if we can create a Steiner tree that consists of complete 5-tuples only, i.e. if and only if there exists an exact 3-cover. Otherwise, the bottleneck cost is at least 2. □

3 The Power-p Steiner Ratio

Theorem 3 *The power-p Steiner ratio ρ_p is finite for any p.*

Proof. We may assume that an instance I of a power-p Steiner problem is defined by a graph (V, E), a length function of edges d and a set of terminals $S \subset V$. In such a case we assume that the distance between two nodes is the length of the respective shortest path. Let $Mst(I)$ and $St(I)$ be the power-p cost of the minimum spanning of S and the minimum Steiner tree respectively. For each instance I we also define $ratio(I) = Mst(I)/St(I)$. Our goal is to show how to find the power-p Steiner ratio, i.e. a value ρ_p such that $ratio(I) \leq \rho_p$ for every I. Our strategy is to show that for every instance I there exists an instance I' such that $ratio(I') \geq ratio(I)$, while the form of I' is so special that we can find succinct expressions for $St(I')$ and $Mst(I')$ and compare them using elementary calculus.

Consider now an instance I with minimum Steiner tree T. Let I' be the instance obtained by replacing (V, E) with T. Obviously, $St(I') = St(I)$ while $Mst(I') \geq Mst(I)$, hence $ratio(I') \geq ratio(I)$. Therefore it suffices to consider only the instances that are defined by their Steiner trees. We will use $I(T)$ to denote the instance defined by tree T.

Similarly, if in a Steiner tree T one of the terminals u is an internal node, we may divide T into two parts, T_1 and T_2, that share node u. Clearly, $St(I(T_1)) + St(I(T_2)) = St(I(T))$, while $Mst(I(T_1)) + Mst(I(T_2)) \geq Mst(I(T))$. Therefore we may consider only the trees where the set of terminals and leaves are equal.

We may put further restrictions on the form of T. Replacing a node of degree higher than 3 with a group of nodes connected by 0-length edges changes neither $Mst(I(T))$, nor $St(I(T))$. Similarly, we can freely introduce and eliminate nodes of degree 2. Thus we may assume that all internal nodes of T have degree 3, with the exception of the "root node" r that has degree 2, i.e. that T is a rooted binary tree. We can also replace a leaf with a group of nodes connected by 0-length edges: $Mst(I(T))$ clearly does not change, while $St(I(T))$ can only decrease. Thus we may assume that T is a full binary tree. The final simplification of T is done by the following lemma.

Lemma 1 *For every full binary tree T, there exists a full binary tree T' such that on every level all edges have the same length, while $St(I(T')) = St(I(T))$ and $Mst(I(T')) \geq Mst(I(T))$.*

Proof. We use the following strategy. We will define a spanning tree of T, $M(T)$. In our transformations, $St(T)$ will be fixed, the cost of $M(T)$ nondecreasing, and at the end $M(T)$ will be a minimum spanning tree.

Let r be the root of T. We say that *level i* consists of nodes that are i edges away from the leaves, and edges that join nodes on levels i and $i-1$. Let $\pi(u)$ be a shortest path from u to a leaf of $T(u)$. Assuming that u_1 and u_2 are the children of u, we define the edge $e(u)$ as a path from a leaf of $T(u_1)$ to a leaf of $T(u_2)$ obtained by concatenating the paths $\pi(u_1)$, (u_1, u, u_2) and $\pi(u_2)$. The tree $M(T)$ is defined by collecting all edges of this form.

After $k-1$ transformations, the edges on the same level are the same, provided that they are in the same $T(u)$ for some u on level $k-1$. This claim is trivially true after 0 transformations.

Now we will describe our k-th transformation (see Figure 2). We perform it separately for every node on level k. Let v_k be such a node, with children v_{k-1} and v'_{k-1}, and let

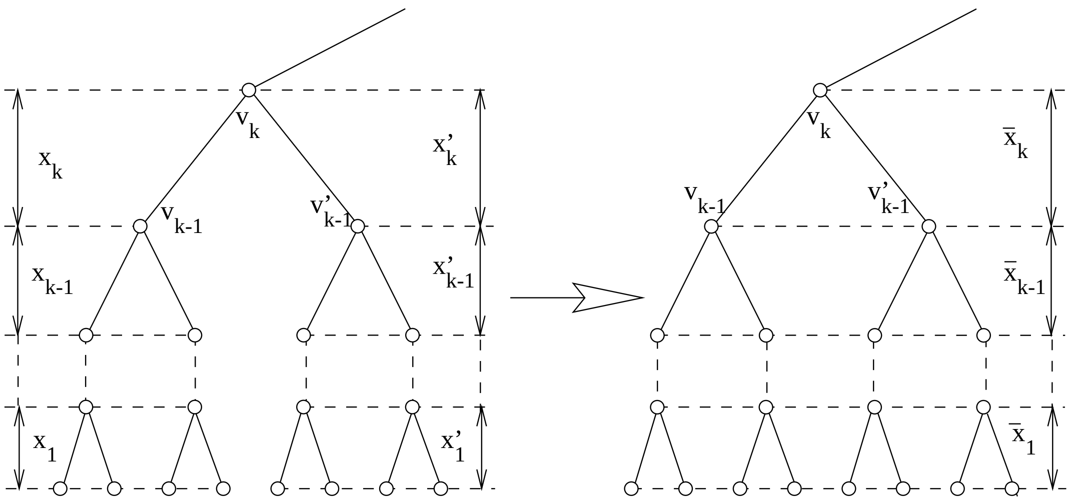

Figure 2: After the k-th transformation, all edges on the same level i, $i \leq k$, have the same length.

$x_k = d(v_k, v_{k-1})$, $x'_k = d(v_k, v'_{k-1})$. Because of the previous $k-1$ transformations, for $1 \leq i < k$, all edges on i-th level of $T(v_{k-1})$ (and $T(v'_{k-1})$) have the same length, x_i (respectively, x'_i).

The transformation replaces the lengths of edges on level i of $T(v_k)$ with

$$\bar{x}_i = \left(\frac{(x_i)^p + (x'_i)^p}{2}\right)^{\frac{1}{p}}$$

It is obvious that this change of edge length does not change $St(T)$. We only need to show the cost of $M(T)$ cannot decrease. Let l be the the length of $\pi(v_{k-1})$ plus x_k, and l' be the the length of $\pi(v'_{k-1})$ plus x'_k. After the transformation, both these lengths become equal to $\bar{l}$. Suppose $l \leq l'$. For an ancestor u of v_k, the length of edge $e(u)$ cannot decrease, because in the respective path we replace l with the greater value $\bar{l}$:

$$l \;\leq\; \frac{1}{2}\left(\sum_{i=1}^{k} x_i + \sum_{i=1}^{k} x'_i\right) \;=\; \sum_{i=1}^{k} \frac{x_i + x'_i}{2} \;\leq\; \sum_{i=1}^{k} \left(\frac{(x_i)^p + (x'_i)^p}{2}\right)^{\frac{1}{p}} \;=\; \sum_{i=1}^{k} \bar{x}_i \;\leq\; l'$$

The central inequality is a well-known inequality on power-p means (see e.g. [7], page 26, inequality 2.9.1).

Now it remains to show that the sum of costs of $e(u)$ increases for the descendants of v_k. These descendants can be paired, a node u on i-th level from $T(v_{k-1})$ with a node u' on the same level from $T(v'_{k-1})$. Before the transformation, the sum of costs of $e(u)$ and $e(u')$ equals

$$\left(\sum_{j=1}^{i} 2x_j\right)^p + \left(\sum_{j=1}^{i} 2x'_j\right)^p \;=\; 2^p\left(\left(\sum_{j=1}^{i} x_j\right)^p + \left(\sum_{j=1}^{i} x'_j\right)^p\right)$$

After the transformation, the corresponding sum equals

$$2\left(\sum_{j=1}^{i} 2\left(\frac{(x_j)^p + (x'_j)^p}{2}\right)^{\frac{1}{p}}\right)^p \quad = \quad 2^p\left(\sum_{j=1}^{i}\left((x_j)^p + (x'_j)^p\right)^{\frac{1}{p}}\right)^p$$

The necessary inequality

$$\left(\sum_{j=1}^{i} x_j\right)^p + \left(\sum_{j=1}^{i} x'_j\right)^p \quad \leq \quad \left(\sum_{j=1}^{i}\left((x_j)^p + (x'_j)^p\right)^{\frac{1}{p}}\right)^p$$

follows from Minkowski's inequality (see e.g. [7], page 32, inequality 2.12.1) after substitution: $(x_j)^p = a_j$, $(x'_j)^p = b_j$ and $\frac{1}{p} = r$. □

According to the last lemma, a tree of the kind that we need to consider has all edges on level i equal to some x_i. Thus a k-level tree T is fully defined by the vector $\vec{x} = (x_1, \ldots, x_k)$. Then $St(T) \leq 2\sum_{i=1}^{k} 2^{k-i}x_i^p = 2S(\vec{x})$, while $Mst(T) = \sum_{i=1}^{k} 2^{k-i}\left(2\sum_j = 1^i x_j\right)^p = 2^p \sum_i = 1^k 2^{k-i}\left(\sum_j = 1^i x_j\right)^p = 2^p M(\vec{x})$. Thus $ratio(T) \leq 2^{p-1}M(\vec{x})/S(\vec{x})$. Our remaining task is to find an upper bound for $M(\vec{x})/S(\vec{x})$.

To finish the proof, we will show that for suitably chosen α and β the ratio $M(\vec{x})/S(\vec{x})$ is bounded by $\alpha/(1-\beta)$. This appears to be difficult, because the quantity in question is a function of k variables. However, the next lemma reduces this inequality to another one that involves two variables only.

Lemma 2 *Assume that $\alpha > 1,\ 1 > \beta > 0$, and*

$$\alpha x^p + 2\beta y^p \geq (x+y)^p \text{ for every } x, y \geq 0 \tag{2}$$

Then $M(\vec{x}) \leq \alpha/(1-\beta)S(\vec{x})$ for every vector $\vec{x}$ with nonnegative entries.

Proof. We can rewrite $M(\vec{x})$ as $\sum_{i=1}^{k}\tau_i$, where $\tau_i = 2^{k-i}\left(\sum_{j=1}^{i} x_j\right)^p$; similarly $S(\vec{x}) = \sum_{i=1}^{k} y_i$, where $y_i = 2^{k-i}x_i^p$. Then we can show by induction on i that

$$\tau_i \ \leq\ \alpha\sum_{j=1}^{i}\beta^{i-j}y_j \ =\ \sigma_i.$$

For $i = 1$, this claim states that $y_i = 2^{k-1}x_1^p < \alpha 2^{k-1}x_1^p$. Now, inductively assume that $\tau_{i-1} \leq \sigma_{i-1}$. Then

$$\begin{aligned}
\tau_i &= 2^{k-i}\left(y_i + \sum_{j=1}^{i-1} x_j\right)^p \\
&\leq \alpha 2^{k-i}y_i^p + 2\beta 2^{k-i}\left(\sum_{j=1}^{i-1} x_j\right)^p \\
&= \alpha y_i + \beta\tau_{i-1} \\
&\leq \alpha y_i + \beta\sigma i - 1 \\
&\leq \alpha y_i + \alpha\sum_{j=1}^{i-1}\beta^{1+i-1-j}y_j \\
&= \sigma_i
\end{aligned}$$

It remains to estimate $\sum_{i=1}^{k} \sigma_i$. Note that this sum is a linear combination of y_js. In particular, y_j appears in σ_j, σ_{j+1}, σ_{j+2}, etc., with coefficients that are $\alpha\beta^0$, $\alpha\beta^1$, $\alpha\beta^2$ etc. Obviously, these coefficients add to less than $\sum_{l=0}^{\infty} \alpha\beta^l = \alpha/(1-\beta)$. Therefore

$$M(\vec{x}) \quad \leq \quad \frac{\alpha}{1-\beta} \cdot \sum_{i=1}^{k} y_i \quad = \quad \frac{\alpha}{1-\beta} S(\vec{x})$$

□

Now it remains to show how to find α and β that satisfy inequality (2). As a preliminary step, we eliminate one of the variables: we can divide both sides by $(x+y)^p$, and substitute $x \leftarrow x/(x+y)$. The range of the new variable is between 0 and 1. Then the existence of the desired α and β follows quickly from the following

Lemma 3 *Assume that δ and x are between 0 and 1 and $p > 1$. Then*

$$\frac{x^p}{\delta^{p-1}} + \frac{(1-x)^p}{(1-\delta)^{p-1}} \geq 1$$

Proof. We will find a necessary and sufficient condition for γ so that

$$\frac{x^p}{\delta^{(p-1)}} + \frac{(1-x)^p}{\gamma^{(p-1)}} \geq 1 \text{ for every } x \in (0,1) \tag{3}$$

Let $f(x)$ be the left hand side of (3), then the derivative of f is

$$f'(x) = p\left(\frac{x^{p-1}}{\delta^{(p-1)}} - \frac{(1-x)^{p-1}}{\gamma^{(p-1)}}\right) = p\left(\left(\frac{x}{\delta}\right)^{p-1} - \left(\frac{1-x}{\gamma}\right)^{p-1}\right).$$

It is easy to see that the minimum of f satisfies

$$f'(x) = 0 \iff \left(\frac{x}{\delta}\right)^{p-1} = \left(\frac{1-x}{\gamma}\right)^{p-1} \iff \frac{x}{\delta} = \frac{1-x}{\gamma}.$$

The last equality has a unique solution $x_0 = \delta/(\delta+\gamma)$ (consequently $1 - x_0 = \gamma/(\delta+\gamma)$). Because x_0 is the minimum, it is sufficient and necessary that $f(x_0) \geq 1$:

$$\begin{aligned}
f(x_0) &= \frac{x^p}{\delta^{p-1}} + \frac{(1-x)^p}{\gamma^{p-1}} \\
&= \frac{1}{\delta^{p-1}} \frac{\delta^p}{(\delta+\gamma)^p} + \frac{1}{\delta^{p-1}} \frac{\gamma^p}{(\delta+\gamma)^p} \\
&= \frac{\delta}{(\delta+\gamma)^p} + \frac{\gamma}{(\delta+\gamma)^p} \\
&= \frac{1}{(\delta+\gamma)^{p-1}} \geq 1 \\
\iff & (\delta+\gamma)^{p-1} \leq 1 \iff \delta + \gamma \leq 1
\end{aligned}$$

The last inequality shows that $1-\delta$ is the maximum value of γ such that condition 3 holds.

□

This concludes the proof of our theorem. □

To illustrate how this technique allows to estimate the power-p Steiner ratio with a concrete constant, we will review the case of $p = 2$. We know that this ratio is bounded by $2^{p-1}M(\vec{x})/S(\vec{x})$, which in turn, by lemma 2 is bounded by $2\alpha/(1-\beta)$. By Lemma 3, we can pick α and β by choosing some δ between 0 and 1 and setting $\alpha = (1-\delta)^{-p+1}$ and $\beta = \delta^{-p+1}/2$; in our case $\alpha = 1/(1-\delta)$ and $\beta = 1/2\delta$. The resulting ratio is

$$f(\delta) \quad = \quad 2\frac{1}{1-\delta} \cdot \frac{1}{1-1/2\delta}$$

It is easy to check (0 of the derivative etc.) that $f(\sqrt{1/2}) = 8\sqrt{2} + 12 \approx 23.314$ yields the minimum value. Thus, we proved the following

Corollary 1 *The power-2 Steiner ratio is at most* $8\sqrt{2} + 12 \approx 23.314$.

This upper bound is not exact. In the worst case that we have found the edge length x_i form an arithmetical progression with the remainder 1 and the power-2 Steiner ratio is ≈ 17.2.

4 A Better Heuristic for the Bottleneck Steiner Problem

It was proved in [6] that the Bottleneck Steiner ratio is at most $2\log_2 n$, where $n = |S|$ is the number of terminals. Moreover they showed that this bound is tight, i.e. the minimum spanning tree can have an edge as long as $2\log_2 n$ times than length of the longest edge in the optimal bottleneck tree. In this section we suggest a better approximation algorithm with the performance ratio half of that of the minimum spanning tree heuristic.

Let consider a weighted hypergraph $H = (S, U, w)$ in which the vertex set coincides with the terminal set S and the edge set U contains all pairs and triples of terminals. The weight of an edge (s, s') is its length and the weight of a triple τ is the maximum edge length in the minimum bottleneck steiner tree for τ. Let $H_{\leq\alpha}$ be subhypergraph of H which contains only hyperedges of weight at most α. Note that in hypergraphs with the edge size at most 3, we can find efficiently using Lovasz' algorithm [11] the spanning hypertree B (i.e. hypergraph B such that replacing all 3-hyperedges of B with two usual edges makes a usual spanning tree), if it exists. It is easy to see that replacing each 3-hyperedge of B with the corresponding minimum bottleneck Steiner tree converts B to a Steiner tree for S.

Algorithm 1:

(1) For each triple of terminals find the optimal bottleneck tree

(2) Find the minimum α, such that the hypergraph $H_{\leq\alpha}$ contains a spanning hypertree

(3) Output the Steiner tree corresponding to the spanning hypertree of the hypergraph $H_{\leq\alpha}$

Theorem 4 *The performance ratio of Algorithm 1 is at most* $\log_2 n$, *where* n *is the number of terminals.*

Proof. Let OPT be an optimal bottleneck Steiner tree. We may assume that all internal nodes has degree 3 and all terminals are leaves in OPT. Indeed, if an internal node v has degree at least 4, then we add an auxiliary copy of v connected to v with a zero-cost edge

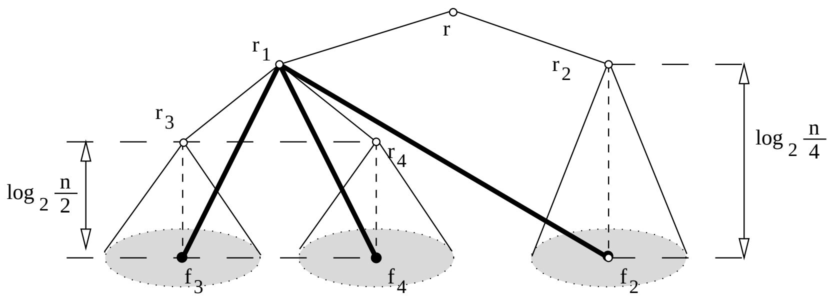

Figure 3: The Steiner tree for terminals f_3, f_4 and f_2 with the additional node r_1 (thick edges) has bottleneck cost at most $\log_2 n$.

and evenly split vertices adjacent to v between v and and its copy. Similarly, if a terminal s is not a leaf in OPT, then we add a copy of s, say s', which is not a terminal. Then we substitute s with s' in OPT while joining s to s' with a zero-cost edge.

Moreover, we may assume that all edges in OPT have the same length 1. Indeed, if there is an edge of length less than the maximum length, then we may increase its length up to the maximum without increasing the bottleneck cost.

We will use the following two lemmas which are well-known graph-theoretical facts (e.g. Lemma 5 can be found in [1]).

Lemma 4 *Given a tree with n leaves and all internal nodes of degree 3, except the root r that may have degree 2 or 3. There exists a leaf at most $\log_2 n$ edges away from r.*

Lemma 5 *Any tree with n leaves contains a node c (called center) such that removing c splits the tree into connecting components each containing at most $n/2$ nodes.*

In order to proof Lemma, we will span the set of terminals S with a union of Steiner trees for triples (and pairs) of terminals each having the bottleneck cost at most $\log_2 n$.

Let c be a center of the tree OPT and let c_1, c_2, and c_3 be the three nodes adjacent to c. Removing c splits the tree OPT into three connected components, say OPT_1, OPT_2, and OPT_3 such that $c_i \in OPT_i$. Note that each node c_i has degree 2 inside the tree OPT_i, and Lemma 4 implies that there is a path from c_i to a leaf of OPT_i, say l_i, of length at most $\log_2 \frac{n}{2} = \log_2 n - 1$. Thus, $d(c, l_i) = d(c, c_i) + d(c_i, l_i) \leq \log_2 n$ and the Steiner tree B_0 with the three edges (c, l_i), $i = 1, 2, 3$, has the bottleneck cost at most $\log_2 n$. The tree $B(OPT)$ connects terminals from three different terminal sets $OPT_i \cap S$ and therefore, reduces the problem to the trees OPT_i, $i = 1, 2, 3$, with at most $n/2$ terminals and exactly one internal node of degree 2.

Now we consider a tree T of such form, i.e. T has at most $\frac{n}{2}$ terminals-leaves, the root r of T has degree 2 and all other internal nodes have degree 3 (see Figure 3). Removing r splits T into two subtrees T_1 and T_2, with roots r_1 and r_2 which are neighbors of r in T. For certainty, we assume that T_1 has as many leaves as T_2, i.e. either $|S \cap T_1| = |S \cap T_2| = 1$ or $|S \cap T_1| \geq 2$ and $|S \cap T_2| \leq \frac{n}{4}$. In the former case the cost of edge $B(T)$ between two terminals is 2. Otherwise, removing r_1 from T_1 splits it into two trees, say T_3 and T_4, with the roots r_3 and r_4, respectively. Let f_i be the leaf closest to r_i in each tree T_i, $i = 2, 3, 4$.

Note that $d(r_1, f_2) \leq d(r_2, f_2) + 2 \leq \log_2 \frac{n}{4} + 2 = \log_2 n$. Similarly, $d(r_1, f_3)$ and $d(r_1, f_4)$ are at most $\log_2 n$. The bottleneck cost of the Steiner tree $B(T)$ with edges (r_1, f_i), $i = 2, 3, 4$, is at most $\log_2$, and therefore, $B(T)$ reduces the problem to the trees T_i, $i = 2, 3, 4$ each having the same form but less terminals. $\square$

References

[1] D. Bateman, C. Helvig, G. Robins and A.Zelikovsky. *Provably good routing tree construction with multi-port terminals.* Proc. of IEEE-ACM Int. Symp. on Physical Design, 96–102, 1997.

[2] K.D. Boese, A. Kahng, B.A. McCoy and G. Robins. *Near-optimal critical sink routing tree constructions.* IEEE Trans. Comput.-Aided Des. **14**, 1417–1436, 1995.

[3] C.Chiang, M. Sarrafzadeh and C.K. Wong. *Global routing based on Steiner min-max trees.* IEEE Trans. Comput.-Aided Des. **9**, 1318–1325, 1990.

[4] J. Elzigna, D. Hearn and W.D. Randolph. *Minimax multifacility location with Euclidean distances.* Transport. Sci. **10**, 321–336, 1976.

[5] J.L. Ganley. *Geometric interconnection and placement algorithms.* Ph.D. Thesis, Department of Computer Science, University of Virginia, Charlottesville, VA, 1995.

[6] J.L. Ganley and J.S. Salowe. *Optimal and approximate bottleneck Steiner trees.* Oper. Res. Lett. **19**, 217–224, 1996.

[7] G. Hardy, E. Littlewood and G. Polya, *Inequalities*, Cambridge University Press, 1934.

[8] N.D. Holmes, N.A. Sherwani and M. Sarrafzadeh. *Utilization of vacant terminals for improved over-the-cell channel routing.* IEEE Trans. Comput.-Aided Des. **12**, 780–782, 1993.

[9] J.N. Hooker. *Solving nonlinear multiple-facility network, location problem.* Networks **19**, 117–133, 1989.

[10] R.M. Karp. *Reducibility among combinatorial problems.* In R.E. Miller and J.W. Thatcher (eds.), Complexity of Computer Computations, Plenum Press, New York, 85–103, 1972.

[11] L. Lovasz and M.D. Plummer. *Matching Theory.* Elsvier Science, Amsterdam, 1986.

[12] C.H. Papadimitriou, M. Yannakakis, *Optimization, approximation, and complexity classes*, Proc. 20th STOC, 229-234, 1988.

[13] M. Sarrafzadeh and C.K. Wong. *Bottleneck Steiner trees in the plane.* IEEE Trans. Comput. **41**, 370–374, 1992.

[14] J. Soukup. *On minimum cost networks with nonlinear costs.* SIAM Journal on Appl. Math., **29**, 571–581, 1975.

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