Improved Approximation Algorithms for the Quality of Service Steiner Tree Problem*

Marek Karpinski **, Ion I. Măndoiu * * *, Alexander Olshevsky †, and Alexander Zelikovsky ‡

Abstract. The Quality of Service Steiner Tree Problem is a generalization of the Steiner problem which appears in the context of multimedia multicast and network design. In this generalization, each node possesses a rate and the cost of an edge with length l in a Steiner tree T connecting the non-zero rate nodes is $l \cdot r_e$, where r_e is the maximum rate in the component of $T - \{e\}$ that does not contain the source. The best previously known approximation ratios for this problem (based on the best known approximation factor of 1.549 for the Steiner tree problem in networks) are 2.066 for the case of two non-zero rates and 4.211 for the case of unbounded number of rates. We give better approximation algorithms with ratios of 1.960 and 3.802, respectively. When the minimum spanning tree heuristic is used for finding approximate Steiner trees, then the previously best known approximation ratios of 2.667 for two non-zero rates and 5.542 for unbounded number of rates are reduced to 2.414 and 4.311, respectively.

Keywords: Approximation algorithm, interconnection network, quality of service, Steiner tree

1 Introduction

The Quality of Service Steiner Tree (QoSST) problem appears in two different contexts: multimedia distribution for users with different bitrate requests [7] and the general design of interconnection networks with different grade of service requests [6]. The problem was formulated as a natural generalization of the Steiner problem under the names "Multi-Tier Steiner Tree Problem" [8] and "Grade of Service Steiner Tree Problem" [13]. More recently, the problem has

* A preliminary version of this article has appeared in the Proceedings of the Workshop on Algorithms and Data Structures, 2003

** Department of Computer Science, University of Bonn, Bonn 53117, Germany, marek@theory.cs.uni-bonn.de

* * * Electrical and Computer Engineering Department, University of California at San Diego, La Jolla, CA 92093-0114, mandoiu@cs.ucsd.edu

† Department of Electrical Engineering, Georgia Institute of Technology, Atlanta, GA 30332,gte864k@prism.gatech.edu

‡ Computer Science Department, Georgia State University, Atlanta, GA 30303,alexz@cs.gsu.edu

been considered by [5, 7] in the context of multimedia distribution. This problem generalizes the Steiner tree problem in that each node possesses a rate and the cost of a link is not constant but depends both on the cost per unit of transmission bandwidth and the maximum rate routed through the link.

Formally, the QoSST problem can be stated as follows (see [5]). Let $G = (V, E, l, r)$ be a graph with two functions, $l : E \to R^+$ representing the length of each edge, and $r : V \to R^+$ representing the rate of each node. Let $\{r_0 = 0, r_1, r_2, \dots r_N\}$ be the range of r and S_i be the set of all nodes with rate r_i. The Quality of Service Steiner Tree Problem asks for a minimum cost subtree T of G spanning a given source node s and nodes in $\bigcup_{i\geq 1} S_i$, all of which are referred to as *terminals*. The cost of an edge e in T is $\text{cost}(e) = l(e)r_e$, where r_e, called the *rate of edge* e, is the maximum rate in the component of $T - \{e\}$ that does not contain the source. Note that the nodes in S_0, i.e., zero rate nodes, do not require to be connected to the source s but may serve as Steiner points for the output tree T.

The QoSST problem is equivalent to the Grade of Service Steiner Tree Problem (GOSST) [13], which has a slightly different formulation. In GOSST there is no source node and edge rates r_e should be assigned such that the minimum edge rate on the tree path from a terminal with rate r_i to a terminal with rate r_j is at least $min(r_i, r_j)$. It is not difficult to see that these two formulations are equivalent. Indeed, an instance of QoSST can be transformed into an instance of GOSST by assigning the highest rate to the source. The cost of an edge will remain the same, since each edge e in a tree T will be on the path from the source to the node of the highest rate in the component of $T - \{e\}$ that does not contain the source. Conversely, an instance of GOSST can be transformed into a QoSST by giving source status to any node with the highest rate.

The problem was studied before in several contexts. Current et al. [6] gave an integer programming formulation for the problem and proposed a heuristic algorithm for its solution. Some results for the case of few rates were obtained in [1] and [2]. Specifically, [2] (see also [13]) suggested an algorithm for the case of two non-zero rates with approximation ratio of $\frac{4}{3}\alpha \approx 2.065$, where $\alpha \approx 1.549$ is the best approximation ratio of an algorithm for the Steiner tree problem. Recently, [5] gave the first constant-factor approximation algorithm for an unbounded number of rates. They achieved an approximation ratio of $e\alpha \approx 4.211$.

In this paper we give algorithms with improved approximation factors. Our algorithms have an approximation ratio of 1.960 when there are two non-zero rates and an approximation ratio of 3.802 when there is an unbounded number of rates. The improvement comes from the reuse of higher rate edges in establishing connectivity for lower rate nodes. We give the first analysis of the gain resulting from such reuse, critically relying on approximation algorithms for computing k-restricted Steiner trees. To improve solution quality, we use different Steiner tree algorithms at different stages of the computation. In particular, we use both the Steiner tree algorithm from [11] which has the currently best approximation ratio and the algorithm from [10] which has the currently best approximation ratio among Steiner tree algorithms producing 3-restricted trees.

Algorithm	LCA [11]		RNS[10]		BR [14, 3]		MST [12]	
runtime	polynomial		polynomial		$O(n^3)$ [15]		$O(n \log n + m)$ [9]	
#rates	2	any	2	any	2	any	2	any
previous ratio	$2.066 + \epsilon$	$4.211 + \epsilon$	$2.222 + \epsilon$	$4.531 + \epsilon$	2.444	4.934	2.667	5.44
our ratio	$1.960{+}\epsilon$	$3.802 + \epsilon$	$2.059 + \epsilon$	$3.802 + \epsilon$	2.237	4.059	2.414	4.311

Table 1. Runtime and approximation ratios of previously known algorithms and of the algorithms given in this paper. In the runtime, n and m denote the number of nodes and edges in the original graph $G = (V, E)$, respectively.

Table 1 summarizes the results of this paper. It presents previously known approximation ratios using various Steiner tree algorithms and the approximation ratios produced by our method utilizing the same algorithms. Note that along with the best approximation ratios resulting from the use of the loss-contracting algorithm from [11], Table 1 also gives approximation ratios resulting from the use of the algorithm in [10] and the more practical algorithms in [3, 12, 14].

The rest of the paper is organized as follows. In next section, we tighten the analysis given in [4] for the k-restricted Steiner ratio. In Section 3, we introduce the so called β-convex Steiner tree approximation algorithms and tighten their performance bounds. We give approximation algorithms for QoSST problem with two non-zero rates and unbounded number of rates in Sections 4 and 5, respectively, and conclude in Section 6.

2 A Tighter Analysis of the k-restricted Steiner Ratio

In this section, we tighten the analysis given in [4] for the k-restricted Steiner ratio. The tightened results will be used later to prove the approximation ratio of our algorithms. The exposition begins with a claim from [4] which encapsulates several of the proofs provided in that paper. This claim is then used in a manner slightly different from [4] to arrive at a stronger result.

We begin by introducing some definitions. A Steiner tree is called *full* if every terminal is a leaf. A Steiner tree can be decomposed into components which are full by breaking the tree up at the non-leaf terminals. A Steiner tree is called k-restricted if every full component has at most k terminals. Let us denote the length of the optimum k-restricted Steiner tree as opt_k and length of the optimum unrestricted Steiner tree as opt. By duplicating nodes and introducing zero length edges, it can be assumed that a Steiner tree T is a complete binary tree (see Figure 1). Furthermore, we may assume that the leftmost and rightmost terminals form a diametrical pair of terminals. The leftmost and rightmost terminals will be called *extreme terminals*, and the edges on the path between them will be called *extreme edges*.

Let the k-restricted Steiner ratio ρ_k be $\rho_k = sup\frac{opt_k}{opt}$, where the supremum is taken over all instances of the Steiner tree problem. It has been shown in [4] that $\rho_k = \frac{(r+1)2^r+s}{r2^r+s}$ where r and s are obtained from the decomposition $k = 2^r + s$, $0 \le s < 2^r$.

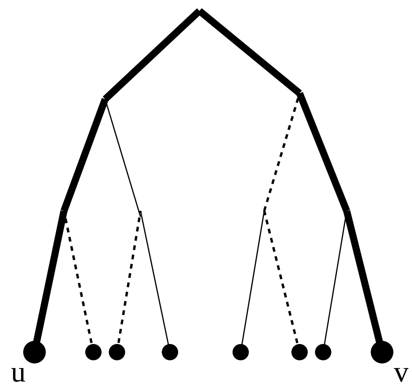

Fig. 1. Optimal Steiner tree T represented as a complete binary tree. Extreme terminals u and v form a diametrical pair of terminals, extreme edges (the path between u and v) are shown thicker. Each L_i represents the total length of a collection of paths (e.g., dashed paths) connecting internal nodes of T to non-extreme terminals via non-extreme edges.

Lemma 1. *[4] Given a Steiner tree T, there exist k-restricted Steiner trees T_i, $i = 1, 2, \ldots, r2^r + s$ such that $l(T_i) = l(T) + L_i$, where each L_i represents the total length of a collection of paths connecting internal nodes of T to non-extreme terminals via non-extreme edges in such a way that each non-extreme edge of T is counted at most 2^r times in the sum $L_1 + L_2 + \cdots + L_{r2^r+s}$.*

This claim is stated in [4] for an optimum Steiner tree T, but optimality is not needed in the proof.

We now use Lemma 1 to produce a tighter bound on the length of the optimal k-restricted Steiner tree.

Theorem 1. *For every full Steiner tree T, $opt_k \le \rho_k(l-D)+D$, where $l = l(T)$ is the length of T and $D = D(T)$ is the length of the longest path in T.*

Proof. Lemma 1 implies that $L_1 + L_2 + \cdots + L_{r2^r+s} \le 2^r(l - D)$. From this it follows that there exists L_m such that $L_m \le \frac{2^r}{r2^r+s}(l-D)$. Since $l(T_m) = l + L_m$, it follows that $l(T_m) \le l + \frac{2^r}{r2^r+s}(l - D)$. Therefore,

$$\begin{aligned} opt_k &\le l(T_m) \\ &\le l + \frac{2^r}{r2^r + s}(l - D) \\ &= \left(1 + \frac{2^r}{r2^r + s}\right)(l - D) + D \\ &= \rho_k(l - D) + D \end{aligned}$$

We now strengthen this theorem to the case of partitioned trees.

Corollary 1. *For every Steiner tree T partitioned into edge-disjoint full components T^i,*

$$opt_k \leq \sum_i \left(\rho_k(l(T^i) - D(T^i)) + D(T^i)\right)$$

Proof. Let opt_k^i be the length of the optimal k-restricted tree for the full component T^i. Then,

$$opt_k \leq \sum_i opt_k^i \leq \sum_i \left(\rho_k(l(T^i) - D(T^i)) + D(T^i)\right)$$

3 β-Convex Steiner Tree Approximation Algorithms

In this section we introduce β-convex Steiner tree approximation algorithms and show tighter upper bounds on their output when applied to the QoSST problem.

Definition 1. *An α-approximation Steiner tree algorithm A is called* β-convex *if the length of the tree it produces, $l(A)$, is upper bounded by a linear combination of optimal k-restricted Steiner trees, i.e.,*

$$l(A) \leq \sum_{i=2}^{m} \lambda_i opt_i$$

and the approximation ratio is equal to

$$\alpha = \sum_{i=2}^{m} \lambda_i \rho_i$$

where $\lambda_i \geq 0$, $i = 2, \ldots, m$ and

$$\beta = \sum_{i=2}^{m} \lambda_i$$

The algorithms from [12, 3, 10] are β-convex, while the currently best approximation algorithm from [11] is not known to be β-convex.

Given a β-convex α-approximation algorithm A, it follows from Theorem 1 that

$$l(A) \leq \sum_i \lambda_i opt_i \leq \sum_i \lambda_i \rho_i (opt - D) + \beta D = \alpha(opt - D) + \beta D \qquad (1)$$

Let OPT be the optimum cost QoSST tree T, and let t_i be the length of rate r_i edges in T. Then,

$$cost(OPT) = \sum_{i=1}^{N} r_i t_i$$

Below we formulate the main property that makes β-convex Steiner tree approximation algorithms useful for QoSST approximation.

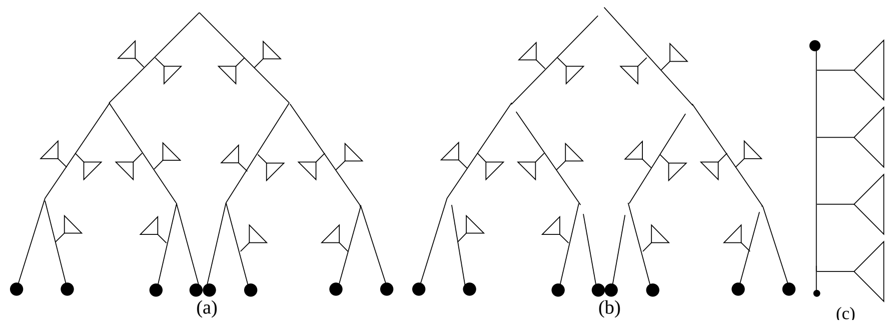

Fig. 2. (a) The subtree OPT_k of the optimal QoS Steiner tree OPT induced by edges of rate r_i, $i \geq k$. Edges of rate greater than r_k (shown as solid lines) form a Steiner tree for $s \cup S_{k+1} \cup \ldots S_N$ (filled circles); attached triangles represent edges of rate r_k. (b) Partition of OPT_k into edge-disjoint connected components OPT_k^i each containing a single terminal of rate r_i, $i > k$. (c) A connected component OPT_k^i which consists of a path D_k^i containing all edges of rate r_i, $i > k$, and attached Steiner trees containing edges of rate r_k.

Lemma 2. *Given an instance of the QoSST problem, let T_k be the Steiner tree computed for s and all nodes of rate r_k by a β-convex α-approximation Steiner tree algorithm after collapsing all nodes of rate strictly higher than r_k into the source s and treating all nodes of rate lower than r_k as Steiner points. Then,*

$$cost(T_k) \leq \alpha r_k t_k + \beta(r_k t_{k+1} + r_k t_{k+2} + \cdots + r_k t_N)$$

Proof. We can visualize the subtree OPT_k of the optimal QoS Steiner tree OPT induced by edges of rate r_i, $i \geq k$ as in Figure 2(a), with nodes of rate r_k forming subtrees attached to the tree OPT_{k+1} that spans all nodes with rate higher than r_k and the source. We break OPT_{k+1} into edge disjoint paths connecting each terminal with an appropriately chosen non-extreme node as illustrated in Figure 2(b). A proof that this kind of decomposition is always possible can be found in [14]. We then consider each such path along with all nodes of rate r_k that are attached to it. This results in a decomposition of OPT_k into edge-disjoint connected components OPT_k^i, where each component consists of a path $D_k^i = OPT_k^i \cap OPT_{k+1}$ and attached Steiner trees with edges of rate r_k (see Figure 2(c)). Furthermore, note that the total length of D_k^i's is $l(OPT_{k+1}) = t_{k+1} + t_{k+2} + \cdots + t_N$.

Now we decompose the tree T_k along these full components OPT_k^i and by Corollary 1 we get:

$$\begin{aligned} l(T_k) &\leq \sum_i \left[\alpha(l(OPT_k^i) - D_k^i) + \beta D_k^i\right] \\ &= \alpha t_k + \beta(t_{k+1} + t_{k+2} + \cdots + t_N) \end{aligned}$$

Input: Graph $G = (V, E, l)$ with two nonzero rates $r_1 < r_2$, source s, terminal sets S_1 of rate r_1 and S_2 of rate r_2, Steiner tree α_1-approximation algorithm A_1 and a β-convex α_2-approximation algorithm A_2
Output: Low cost QoSST spanning all terminals

---

1. Compute an approximate Steiner tree ST1 for $s \bigcup S_1 \bigcup S_2$ using algorithm A_1
2. Compute an approximate Steiner tree T_2 for $s \bigcup S_2$ (treating all other points as Steiner points) using algorithm A_1. Next, contract T_2 into the source s and compute the approximate Steiner tree T_1 for s and remaining rate r_1 points using algorithm A_2. Let ST2 be $T_1 \bigcup T_2$
3. Output the minimum cost tree among ST1 and ST2

Fig. 3. QoSST approximation algorithm for two non-zero rates

The lemma follows by multiplying the last inequality by r_k.

4 QoSST Approximation Algorithm for Two Non-Zero Rates

In this section we give a generic approximation algorithm for the QoSST Steiner tree problem with two non-zero rates (see Figure 3) and analyze its approximation ratio.

Recall that an edge e has rate r_i if the largest rate of a node in the component of $T - \{e\}$ that does not contain the source is r_i. Let the optimal Steiner tree in G have cost $opt = r_1 t_1 + r_2 t_2$, with t_1 being the total length of the edges of rate r_1 and t_2 being the total length of the edges of rate r_2. Let α_1 be the approximation ratio of algorithm A_1 and let α_2 be the approximation ratio of the β-convex algorithm A_2. Then, the following theorem holds:

Theorem 2. *The approximation ratio of the algorithm from Figure 3 is*

$$\max\left\{\alpha_2, \quad \max_r \ \alpha_1 \frac{\alpha_1 - (\alpha_2 - \beta) r}{\beta r^2 + \alpha_1 - \alpha_2 r}\right\}$$

Proof. We can bound the cost of ST1 by $cost(ST1) \le \alpha_1 r_2 (t_1 + t_2)$. To obtain a bound on the cost of ST2 note that $cost(T_2) \le \alpha_1 r_2 t_2$, and that, by Lemma 2, $cost(T_1) \le \alpha_2 r_1 t_1 + \beta r_1 t_2$.

Thus, the following two bounds for the costs of $ST1$ and $ST2$ follow:

$$cost(ST1) \le \alpha_1 r_2 t_1 + \alpha_1 r_2 t_2$$
$$cost(ST2) \le \alpha_1 r_2 t_2 + \alpha_2 r_1 t_1 + \beta r_1 t_2$$

We distinguish between the following two cases:
Case 1: If $\beta r_1 - (\alpha_2 - \alpha_1) r_2 \le 0$, then $cost(ST2) \le \alpha_2 (r_2 t_2 + r_1 t_1) = \alpha_2 opt$.

Case 2: If $\beta r_1 - (\alpha_2 - \alpha_1) r_2 \geq 0$, then let

$$x_1 = \frac{\beta r_1^2 + (\alpha_1 - \alpha_2) r_1 r_2}{\alpha_1 r_2 (\alpha_1 r_2 - \alpha_2 r_1 + \beta r_1)}$$
$$x_2 = \frac{r_2 - r_1}{\alpha_1 r_2 - \alpha_2 r_1 + \beta r_1}$$

It is easy to check that

$$x_1 cost(ST1) + x_2 cost(ST2) \leq opt$$

which implies that

$$Approx \leq \frac{1}{x_1 + x_2} opt$$

In turn, this simplifies to

$$Approx \leq \alpha_1 \frac{\alpha_1 - (\alpha_2 - \beta) r}{\beta r^2 - \alpha_2 r + \alpha_1} opt$$

where $r = \frac{r_1}{r_2}$.

We can use Theorem 2 to obtain numerical bounds on the approximation ratios of our solution. Using $\alpha_1 = 1 + \ln 3/2$ for the algorithm from [11], $\alpha_2 = 5/3$ for the algorithm from [10], $\alpha_1 = \alpha_2 = 11/6$ for the algorithm from [3], and $\alpha_1 = \alpha_2 = 2$ for the MST heuristic, and $\beta \to 1$ for all of the above algorithms, we maximize the expression in Theorem 2 to obtain the following theorem.

Theorem 3. *If the algorithm from [11] is used as A_1 and the algorithm from [10] is used as A_2, then the approximation ratio of the QoSST algorithm in Figure 3 is* 1.960*. If the algorithm from [3] is used in place of both A_1 and A_2, then the ratio is 2.237. If the MST heuristic is used in place of both A_1 and A_2, then the ratio is 2.414.*

5 Approximation Algorithm for QoSST with Unbounded Number of Rates

In this section, we propose an algorithm for the case of a graph with arbitrarily many non-zero rates $r_1 < r_2 < \cdots < r_N$. Our algorithm is a modification of the algorithm in [5]. A description of the algorithm is given in Figure 4. As in [5], node rates are rounded up to the closest power of some number a starting with a^y, where y is picked uniformly at random between 0 and 1. In other words, we round up node rates to numbers in the set $\{a^y, a^{y+1}, a^{y+2}, \ldots\}$. The only difference is that we *contract* each approximate Steiner tree, $Approx_k$, constructed over nodes of rounded rate a^{y+k}, instead of simply taking their union as in [5]. This allows contracted edges to be reused at zero cost by Steiner trees connecting lower rate

Input: Graph $G = (V, E, l)$, source s, sets S_i of terminals with rate r_i, positive number a, and α-approximation β-convex Steiner tree algorithm
Output: Low cost QoSST spanning all terminals

1. Pick y uniformly at random between 0 and 1. Round up each rate to the closest power of some number a starting with a^y, i.e. round up to numbers in the set $\{a^y, a^{y+1}, a^{y+2}, \ldots\}$. Form new terminal sets S'_i which are unions of terminal sets with rates rounded to the same number r'_i
2. $Approx \leftarrow \emptyset$
3. Repeat until all terminals are contracted into the source s:
   Find an α-approximate Steiner tree $Approx_i$ spanning $s \bigcup S'_i$
   $Approx \leftarrow Approx \cup Approx_i$
   Contract $Approx_i$ into source s
4. Output $Approx$

Fig. 4. Approximation algorithm for multirate QoSST

nodes. The following analysis of this improvement shows that it decreases the approximation ratio from 4.211 to 3.802.

Let T_{opt} be the optimal QoS Steiner tree, and let t_i be the total length of the edges of T_{opt} with rates rounded to a^{y+i}. First, we prove the following technical lemma:

Lemma 3. *Let S be the cost of T_{opt} after rounding node rates as in Figure 4, i.e., $S = \sum_{i=0}^{n} t_i a^{y+i}$. Then,*

$$S \leq \frac{a-1}{\ln(a)} cost(T_{opt})$$

Proof. First, note that an edge e used at rate r in T_{opt} will be used at the rate a^{y+m}, where m is the smallest integer i such that a^{y+i} is no less than r. Indeed, e is used at rate r in T_{opt} if and only if the maximum rate of a node connecting to the source via e is r, and every such node will be rounded to a^{y+m}. Next, let $r = a^{x+m}$ (Our proof follows the proof of Lemma 4 in [5]). If $x \leq y$ then the rounded up cost is a^{y-x} times the original cost; otherwise, if $x > y$, is a^{y+1-x} times the original cost. Hence, the expected factor by which the cost of each edge increases is

$$\int_0^x a^{y+x-1} dy + \int_x^1 a^{y-x} dy = \frac{a-1}{\ln a}$$

By linearity of expectation, the expected cost after rounding of T_{opt} is

$$S \leq \frac{a-1}{\ln a} cost(T_{opt})$$

Theorem 4. *The approximation ratio of the algorithm given in Figure 4 is*

$$\min_a \left((\alpha - \beta) \frac{a-1}{\ln a} + \beta \frac{a}{\ln a} \right)$$

Proof. Let $Approx_k$ be the tree added when considering rate r_k. Then, by Lemma 2,

$$cost(Approx_k) \leq \alpha a^{y+k} t_k + \beta a^{y+k+1} t_{k+1} + \beta a^{y+k+2} t_{k+2} + \cdots + \beta a^{y+n} t_n$$

where n is the total number of rates after rounding. Thus, we obtain the following upper bound on the total cost of our approximate solution.

$$
\begin{array}{llllll}
cost(Approx) \leq \alpha t_1 a^y & + \beta t_2 a^y & + \beta t_3 a^y & + \cdots + \beta t_{n-1} a^y & + \beta t_n a^y \\
 & + \alpha t_2 a^{y+1} & + \beta t_3 a^{y+1} & + \cdots + \beta t_{n-1} a^{y+1} & + \beta t_n a^{y+1} \\
 & & \ddots & & \\
 & & & + \alpha t_{n-1} a^{y+n-1} & + \beta t_n a^{y+n-1} \\
 & & & & + \alpha t_n a^{y+n}
\end{array}
$$

$$
= (\alpha-\beta) S + \beta \times \left(
\begin{array}{llllll}
t_1 a^y & + t_2 a^y & + t_3 a^y & + \cdots + t_{n-1} a^y & + t_n a^y \\
 & + t_2 a^{y+1} & + t_3 a^{y+1} & + \cdots + t_{n-1} a^{y+1} & + t_n a^{y+1} \\
 & & & \ddots & \\
 & & & + t_{n-1} a^{y+n-1} & + t_n a^{y+n-1} \\
 & & & & + t_n a^{y+n}
\end{array}
\right)
$$

$$
\leq (\alpha-\beta) S + \beta \times \left(
\begin{array}{llllll}
\vdots & & & & \\
t_1 a^{y-n+1} & \vdots & & & \\
t_1 a^{y-n+2} & + t_2 a^{y-n+2} & \vdots & & \\
t_1 a^{y-n+3} & + t_2 a^{y-n+3} & + t_3 a^{y-n+3} & & \\
\vdots & \vdots & \vdots & \ddots \vdots & \\
t_1 a^{y-1} & + t_2 a^{y-1} & + t_3 a^{y-1} & + \cdots + t_{n-1} a^{y-1} & \vdots \\
t_1 a^y & + t_2 a^y & + t_3 a^y & + \cdots + t_{n-1} a^y & + t_n a^y \\
 & + t_2 a^{y+1} & + t_3 a^{y+1} & + \cdots + t_{n-1} a^{y+1} & + t_n a^{y+1} \\
 & & \ddots & & \\
 & & & + t_{n-1} a^{y+n-1} & + t_n a^{y+n-1} \\
 & & & & + t_n a^{y+n}
\end{array}
\right)
$$

$$\leq (\alpha-\beta) S + \beta S \left(1 + \frac{1}{a} + \frac{1}{a^2} + \cdots\right)$$

$$\leq (\alpha-\beta) \frac{a-1}{\ln a} cost(T_{opt}) + \beta \frac{a}{\ln a} cost(T_{opt})$$

where the last inequality follows from Lemma 3.

Numerically, we obtain approximation ratios of 3.802, 4.059, respectively 4.311, when the α-approximation β-convex Steiner tree algorithm used in Figure 4 is the algorithm in [10], [3], respectively the MST heuristic.

Remark. The algorithm in Figure 4 can be easily derandomized using the same techniques as in [5]

6 Conclusions and Open Problems

In this paper we have considered a generalization of the Steiner problem in which each node possesses a rate and the cost of an edge with length l in a Steiner tree T connecting the terminals is $l \cdot r_e$, where r_e is the maximum rate in the component of $T - \{e\}$ that does not contain the source. We have given improved approximation algorithms finding trees with a cost at most 1.960 (respectively 3.802) times the minimum cost for the case of two (respectively unbounded number of) non-zero rates. Our improvement is based on the analysis of the gain resulting from the reuse of higher rate edges in the connectivity of the lower rate edges. An interesting open question is to extend this analysis to the case of three non-zero rates. The best known approximation factor for this case, is $\alpha(5+4\sqrt{2})/7 \approx 2.358$ [2, 13].

References

1. A. Balakrishnan, T.L. Magnanti, P. Mirchandani, Modeling and Heuristic Worst-Case Performance Analysis of the Two-Level Network Design Problem, *Management Science*, **40**: (1994), 846-867
2. A. Balakrishnan, T.L. Magnanti, P. Mirchandani, Heuristics, LPs, and Trees on Trees: Network Design Analyses, *Operations Research*, **44**: (1996), 478-496
3. P. Berman and V. Ramaiyer, Improved Approximations for the Steiner Tree Problem, *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, (1992), 325-334.
4. A. Borchers and D.Z. Du, The k-Steiner Ratio in Graphs, *SIAM Journal on Computing* ,**26**: (1997), 857-869
5. M. Charikar, J. Naor, and B. Schieber, Resource Optimization in QoS Multicast Routing of Real-Time Multimedia, *Proceedings of the 19th Annual IEEE INFOCOM*, (2000)
6. J.R. Current, C.S. Revelle, and J.L.Cohon, The Hierarchical Network Design Problem, *European Journal of Operations Research*, **27**: (1986), 57-66
7. N. Maxemchuk, Video Distribution on Multicast Networks, *IEEE Journal on Selected Issues in Communications*, **15**: (1997), 357-372
8. P. Mirchandani, The Multi-Tier Tree Problem, *INFORMS Journal on Computing*, **8**: (1996), 202-218
9. K. Mehlhorn, A faster approximation algorithm for the Steiner problem in graphs, *Information Processing Letters*, **27**:(1988), 125-128
10. H. Promel and A. Steger, A New Approximation Algorithm for the Steiner Tree Problem with Performance Ratio $\frac{5}{3}$, *Journal of Algorithms*, **36**: (2000), 89-101
11. G. Robins and A. Zelikovsky, Improved Steiner Tree Approximation in Graphs, *Proc. of ACM/SIAM Symposium on Discrete Algorithms*, (2000), 770-779
12. H. Takahashi and A. Matsuyama, An Approximate Solution for the Steinepwdr Problem in Graphs, *Math. Japonica*, **6:** (1980), 573-577
13. G. Xue, G.-H. Lin, D.-Z. Du, Grade of Service Steiner Minimum Trees in the Euclidean Plane, *Algorithmica*, **31**: (2001), 479-500
14. A. Zelikovsky, An 11/6-approximation algorithm for the network Steiner problem, *Algorithmica*, **9**: (1993), 463-470
15. A. Zelikovsky, A faster approximation algorithm for the Steiner tree problem in graphs, *Information Processing Letters*, **46**: (1993), 79-83