Algorithmica (2002) 32: 554–561
DOI: 10.1007/s00453-001-0089-4

Algorithmica

Approximations for a Bottleneck Steiner Tree Problem

L. Wang[1] and D.-Z. Du[1,2]

Abstract. In the design of wireless communication networks, due to a budget limit, suppose we could put totally $n + k$ stations in the plane. However, n of them must be located at given points. Of course, one would like to have the distance between stations as small as possible. The problem is how to choose locations for other k stations to minimize the longest distance between stations. This problem is NP-hard. We show that if $NP \neq P$, no polynomial-time approximation for the problem in the rectilinear plane has a performance ratio less than 2 and no polynomial-time approximation for the problem in the Euclidean plane has a performance ratio less than $\sqrt{2}$ and that there exists a polynomial-time approximation with performance ratio 2 for the problem in both the rectilinear plane and the Euclidean plane.

Key Words. Steiner tree, Wireless communication.

1. Introduction. Given a set of terminals in a plane, a *Steiner tree* is an acyclic network interconnecting the terminals. Every vertex on a Steiner tree other than terminals is called a *Steiner point*. Usually, every leaf in a Steiner tree is a terminal. However, a terminal may not be a leaf. A Steiner tree is *full* if all terminals are also leaves. Thus, if a Steiner tree is not full, there must exist a terminal which is not a leaf. So, we can decompose the tree at this terminal into several small trees. In this way we can decompose any Steiner tree into the union of several small trees, in each of them a vertex is a leaf if and only if it is a terminal. These small trees are called *full Steiner components*.

In the design of wireless communication networks, due to a budget limit, suppose we could put totally $n + k$ stations in the plane. However, n of them must be located at given points. Of course, one would like to have the distance between stations as small as possible. The problem is how to choose locations for other k stations to minimize the longest distance between stations. This problem can be formulated as the following *bottleneck Steiner tree problem*: Given a set $P = \{p_1, p_2, \ldots, p_n\}$ of n terminals and a positive integer k, we want to find a Steiner tree with at most k Steiner points such that the length of the longest edges in the tree is minimized. This problem is NP-hard. In this paper we show that (a) if $NP \neq P$, then the performance ratio of any polynomial-time approximation for the problem in the Euclidean plane is at least $\sqrt{2}$; (b) if $NP \neq P$, then the performance ratio of any polynomial-time approximation for the problem in the rectilinear plane is at least 2; (c) there exists a polynomial-time approximation with performance ratio 2 for the problem in both rectilinear and Euclidean planes.

[1] Department of Computer Science, City University of Hong Kong, Kowloon Tong, Hong Kong. {dzd, wangl}@cs.cityu.edu.hk.
[2] Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455, USA. dzd@cs.umn.edu. Supported in part by the National Science Foundation under Grant CCR-9530306.

Received September 11, 2000. Communicated by M.-Y. Kao.
Online publication December 17, 2001.

2. Lower Bounds. In this section we show lower bounds for the performance ratio of polynomial-time approximations for the bottleneck Steiner tree problem.

THEOREM 1. *The bottleneck Steiner tree problem in the rectilinear plane cannot be polynomial-time approximated with a performance ratio smaller than* 2 *provided* $P \neq NP$. *The bottleneck Steiner tree problem in the Euclidean plane cannot be polynomial-time approximated with a performance ratio smaller than* $\sqrt{2}$ *provided* $P \neq NP$.

PROOF. Suppose there exists a polynomial-time approximation A for the bottleneck Steiner tree problem in the rectilinear plane, with performance ratio $2-\varepsilon$ for some positive number ε. Then we show a polynomial-time algorithm for the following restricted version of the planar vertex cover problem that is still NP-hard [8], [9].

Instance: A planar graph $G = (V, E)$ with all vertices of degree at most 4, and a positive integer $k \leq |V|$.
Question: Is there a *connected vertex cover* of size k, i.e., a subset $V' \subseteq V$ with $|V'| = k$ such that for each edge $\{u, v\} \in E$ at least one of u and v belongs to V' and the subgraph induced by V' is connected?

Given $G = (V, E)$ and k, an instance of this problem, we can embed G into the plane so that all edges consist of horizontal and vertical segments of lengths at least $2k + 2$, so that every two edges meet at an angle of 90° or 180°. Place terminals on the interior of each edge in G such that each edge $e \in E$ becomes a path $p(e)$ of many edges, each of length at most 1 and the first and the last edges of length exactly 1. Denote by $P(G)$ the set of all terminals. (See Figure 1. Terminals are open circles)

We claim that G has a connected vertex cover of size k if and only if the approximation algorithm A on input $P(G)$ produces a Steiner tree with at most k Steiner points such that the rectilinear length of each edge in the tree is at most $2 - \varepsilon$.

Clearly, if G has a connected vertex cover of size k, then putting k Steiner points at the k vertices in the vertex cover, we can construct a Steiner tree on $P(G)$ with k Steiner points such that the rectilinear length of each edge in the tree is at most 1. This means that the rectilinear length of each edge in any optimal solution on the input $P(G)$ is at most 1. Therefore, the approximation algorithm A on input $P(G)$ produces a Steiner

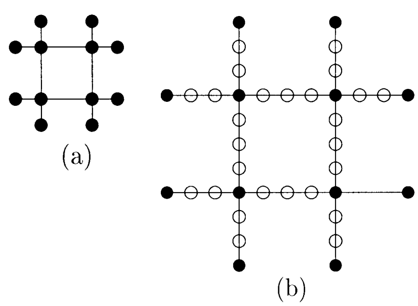

Fig. 1. (a) A planar graph. (b) The constructed graph. The solid circles are candidates of Steiner points and the open circles are terminals.

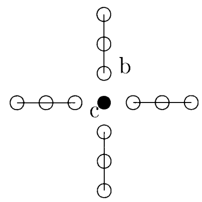

Fig. 2. The distance between c and b is 2 for rectilinear metric and $\sqrt{2}$ for Euclidean metric.

tree with k Steiner points such that the rectilinear length of each edge in the tree is at most $2 - \varepsilon$.

Conversely, assume that the approximation algorithm A on input $P(G)$ produces a Steiner tree T with k Steiner points such that the rectilinear length of each edge in the tree is at most $2 - \varepsilon$. Note that $P(G)$ has the following properties:

(a) Any two terminals at two different edges of G have a distance at least 2.
(b) Any two terminals at two non-adjacent edges of G have a distance at least $2k + 2$. Hence, they cannot be connected through the k Steiner points.

From (b), we know that in any full Steiner component of T, every two terminals lie on either the same edge or two adjacent edges. From (a), we know that if a full Steiner component of T contains two terminals lying on two different edges, then it must contain at least one Steiner point. Thus, we may move a Steiner to the location of the vertex which covers all edges containing terminals in the full Steiner component and remove other Steiner points in the full Steiner component. (See Figure 1.) The result is a Steiner tree T' with at most k Steiner points such that the rectilinear length of each edge in the tree is at most 1. In addition, all Steiner points in T' lie at vertices of G. They form a connected vertex cover V' of size at most k for G. Add more vertices to V' to increase its size equal to k if necessary. This completes the proof of our claim.

The claim shows that the existence of polynomial-time approximation A implies the existence of a polynomial-time algorithm for an NP-complete problem. Thus, if $NP \neq P$, then such a polynomial-time approximation A does not exist.

With a similar argument, we are able to show the second half of the theorem for Euclidean plane, where the distance between b and c in Figure 2 is $\sqrt{2}$. □

3. A 2-Approximation Algorithm. In this section we design a polynomial-time approximation with performance ratio 2 that works for both rectilinear and Euclidean planes. The basic idea of our algorithm is to construct a minimum spanning tree for the set of n terminals, P, and add k Steiner points to the edges in the minimum spanning tree, called a *steinerized* minimum spanning tree. (Similarly, adding some degree-2 Steiner points to a spanning tree results in a *steinerized* spanning tree.)

For simplicity of the proof, we allow a spanning tree to have cross edges so that for any topology, each edge of a spanning tree is a straight line segment between two vertices. Thus, its length is the shortest distance between the two vertices.

Please note that although we allow a spanning tree to have cross edges, the minimum spanning tree would not have cross edges.

LEMMA 2. *Let P be a set of n terminals. There exists a spanning tree T for P such that by adding k Steiner points to T, the longest edge in the resulting steinerized spanning tree is at most twice that of the optimum for the bottleneck Steiner tree problem.*

PROOF. Let T^* be a full Steiner tree with k Steiner points (i.e., all terminals are leaves in T^*.) We arbitrarily select a Steiner point as the root of a tree. Based on T^*, we construct a Steiner tree T' with at most k degree-2 Steiner points such that the length of the longest edge in T' is at most twice the length of the longest edge in T^*.

The construction is bottom-up, by induction on the *height* of subtrees of T^*, which is defined as the number of edges in the longest path from the root to a leaf in a rooted tree. If the height of a subtree of T^* is 1, i.e., there is one Steiner point in the subtree, we can directly connect the terminals without any Steiner points. From the triangular inequality, the lengths of edges in the tree are at most twice that of the length of the longest edge in the subtree. Moreover, there exists a path from the root to a leaf in the subtree such that (all) the Steiner point(s) in the path is unused. Next, assume that for any subtree T of T^* with height less than or equal to h, if T has s Steiner points, then (1) there exists a path (called the *unused* path) from the root of T to a leaf containing ℓ Steiner points, and (2) there exists a Steiner tree T' with s' degree-2 Steiner points such that the length of the longest edge is at most twice the length of the longest edge in T' and $\ell + s' \leq s$. Now, we consider a subtree T with height $h+1$. Suppose that the degree of the root of T is r. Eliminating the root of T results in r subtrees $T_1, T_2, \ldots, T_r$, all of heights at most h. Let k_i be the number of Steiner points in T_i. By the induction assumption, (1) for each T_i, there exists an unused path from the root of T_i to a leaf of it with ℓ_i Steiner points in T_i, and (2) there exists a Steiner tree T_i' with at most $(k_i - \ell_i)$ degree-2 Steiner points such that the length of the longest edges in T_i' is at most twice the length of the longest edge in T_i. Without loss of generality, we assume that $\ell_i \geq \ell_{i+1}$ for every i. Now we connect the r trees $T_1', T_2', \ldots, T_r'$ with $r - 1$ edges and add the $\ell_1, \ell_2, \ldots, \ell_{r-1}$ Steiner points to the $r - 1$ edges, respectively (Figure 3). In this way the unused paths for $T_r^{\min}$ are not used. However, the unused paths for $T_1, T_2, \ldots, T_{r-1}$ are already used. Adding the root (also a Steiner point) for T, we obtain $\ell_r + 1$ Steiner points which are the Steiner points on the unused path for T.

Finally, we note that every Steiner tree can be decomposed into an edge-disjoint union of full Steiner subtrees. □

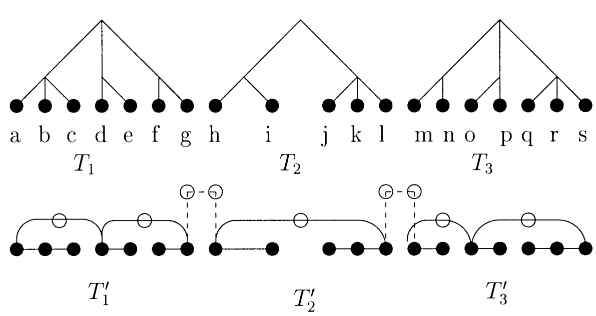

Fig. 3. Connecting the r trees with $r - 1$ edges (paths). The open circled nodes are the added degree-2 Steiner points and the solid circled nodes are terminals.

In the following we study the problem that given a set of terminals and a positive integer k, find a steinerized spanning tree T with k degree-2 Steiner points such that the length of the longest edge in T is minimized. This problem is a bottleneck steinerized spanning tree problem. For simplicity, the optimal solution of this problem is called the (*bottleneck*) *optimal steinerized spanning tree*. We will prove that the optimal steinerized spanning tree can be computed in polynomial time, which therefore is a polynomial-time approximation with performance ratio 2 by Lemma 2.

First we show that the optimal steinerized spanning tree can be found among steinerized minimum spanning trees. To do so, we note the following important fact on the minimum spanning tree.

LEMMA 3. *Let $e_1, e_2, \ldots, e_{n-1}$ be all edges in a spanning tree T and let $e_1^*, e_2^*, \ldots, e_{n-1}^*$ be all edges in a minimum spanning tree T^* for the same terminal set P. Suppose $c(e_i) \le c(e_{i+1})$ and $c(e_i^*) \le c(e_{i+1}^*)$ for all $1 \le i \le n-2$ where $c(e)$ denotes the length of edge e. Then $c(e_i^*) \le c(e_i)$ for all $1 \le i \le n-1$.*

PROOF. We prove it by induction on $|T^*\backslash T|$, the number of edges in T^* but not in T. For $|T^*\backslash T| = 0$, we have $c(e_i) = c(e_i^*)$ for all $1 \le i \le n-1$. Next, consider $|T^*\backslash T| > 0$. Then there exists k, $1 \le k \le n-1$, such that $e_i = e_i^*$ for $1 \le i \le k-1$ and $e_k \ne e_k^*$.

First we show that $c(e_k^*) \le c(e_k)$. In fact, otherwise, suppose $c(e_k^*) > c(e_k)$. Then $c(e_i^*) > c(e_k)$ for all $i, k \le i \le n-1$. Adding e_k to T^* results in a cycle containing e_k and an edge e_h^* for some $h, k \le h \le n-1$. Deleting this edge e_h^* yields a spanning tree with the total length smaller than $c(T^*)$, the total length of T^*, contradicting the minimality of T^*. Therefore $c(e_k^*) \le c(e_k)$.

Now we add e_k^* to T. The resulting graph $T \cup e_k^*$ contains a cycle which contains e_k^* and e_h for some $h, k \le h \le n-1$. Denote $T' = (T \cup e_k^*)\backslash e_h$. Then $|T^*\backslash T'| = |T^*\backslash T| - 1$. Note that all edges in T' in the ordering $c(e_1) \le \cdots \le c(e_{k-1}) \le c(e_k^*) \le c(e_k) \le \cdots \le c(e_{h-1}) \le c(e_{h+1}) \le \cdots \le c(e_{n-1})$. By the induction hypothesis, we have $c(e_{k+1}^*) \le c(e_k), \ldots, c(e_h^*) \le c(e_{h-1}), c(e_{h+1}^*) \le c(e_{h+1}), \ldots, c(e_{n-1}^*) \le c(e_{n-1})$. Therefore, $c(e_i^*) \le c(e_i)$ for all $1 \le i \le n-1$. □

It follows immediately from Lemma 3 that when we use the same number of Steiner points to steinerize a spanning tree and a minimum spanning tree, the result from the latter has a longest edge of length not exceeding that from the former. That is, an optimal steinerized spanning tree can be found among steinerized minimum spanning trees.

Next, we study how to add k Steiner points to a minimum spanning tree in order to obtain an optimal steinerized spanning tree. An algorithm is given in Figure 4, which is explained as follows: For each edge $e_i = (u, v)$ in the minimum spanning tree, if we use l_i Steiner points to steinerize it, then the length of the longest edge in the resulting path from u to v has the minimum value $c(e_i)/(l_i+1)$. This minimum value is achieved when the l_i Steiner points divide e_i evenly. Denote $l(e_i) = c(e_i)/(l_i+1)$. At the beginning of the algorithm, $l(e_i) = c(e_i)$. The basic idea is to add a degree-2 Steiner point to the edge e_i with the largest $l(\cdot)$ value at a time. After e_i receives one more degree-2 Steiner point, $l(e_i)$ is updated by $l_i \leftarrow l_i + 1$. The process is repeated until k degree-2 Steiner points points are added.

1. Input: a set of n terminals.
2. Compute a minimum spanning tree. Suppose $e_1, e_2, \ldots, e_{n-1}$ are all edges in it.
3. Compute $l(e_i)$ for each edge e_i.
4. Sort the edges in a non-increasing order of $l(\cdot)$.
5. Add a degree-2 Steiner point to $e_i = (u, v)$ with the largest $l(\cdot)$ value.
6. Update $l(e_i)$ for the selected edge in Step 3.
7. Re-organize the Steiner points in the path from u to v
8. Re-set e_i's position in the ordering.
9. Repeat Steps 5-8 until k degree-2 Steiner points are added.

Fig. 4. The algorithm for finding an optimal steinerized spanning tree.

The following lemma shows the correctness of the algorithm.

LEMMA 4. *The algorithm in Figure* 4 *returns an optimal steinerized spanning tree with* k *Steiner points*.

PROOF. We prove it by induction on k. For $k = 0$, it is trivial. In general, let T_k be obtained from a minimum spanning tree T through adding k Steiner points by the algorithm in Figure 4. Assume T_k is an optimal steinerized spanning tree with k Steiner points. We next show that T_{k+1} is an optimal steinerized spanning tree with $k+1$ Steiner points. Denote by $C(T_k)$ the length of a longest edge in T_k, i.e., $C(T_k) = \max_{1\le i\le n-1} c(e_i)/(l_i^{(k)}+1)$ where $l_i^{(k)}$ is the number of Steiner points of T_k on the edge e_i of T. Let L_{k+1}^* denote the length of a longest edge in an optimal steinerized spanning tree with $k + 1$ Steiner points. Clearly, $L_{k+1}^* \le C(T_{k+1}) \le C(T_k)$. If $C(T_k) = L_{k+1}^*$, then $C(T_{k+1}) = L_{k+1}^*$, that is, T_{k+1} is an optimal steinerized spanning tree with $k + 1$ Steiner points. Thus, we may assume that $C(T_k) > L_{k+1}^*$. Without loss of generality, suppose e_1 is an edge of T satisfying $c(e_1)/(l_1^{(k)} + 1) = C(T_k)$. Consider an optimal steinerized spanning tree T_{k+1}^* with $k+1$ Steiner points. Let l_i^* denote the number of Steiner points of T_{k+1}^* on the edge e_i of T. We claim that $l_1^* = l_1^{(k)} + 1$.

Clearly, $l_1^* \ge l_1^{(k)} + 1$. In fact, if $l_1^* \le l_1^{(k)}$, then $L_{k+1}^* = C(T_{k+1}^*) \ge c(e_1)/(l_1^{(k)} + 1) = C(T_k)$, a contradiction. Now, suppose to the contrary that $l_1^* > l_1^{(k)} + 1$. Then, removing a Steiner point from e_1, we obtain a steinerized spanning tree with k Steiner points which has the length of the longest edge not exceeding $\max(L_{k+1}^*, c(e_1)/(l_1^{(k)} + 2)) < C(T_k)$, contradicting the induction hypothesis.

Next, we want to show that for every $2 \le i \le n-1$, $l_i^{(k)} \le l_i^*$. In fact, for contradiction, suppose $l_i^{(k)} > l_i^*$ for some $2 \le i \le n-1$. Then we must have $l_i^{(k)} > 0$, that is, e_i receives some Steiner points at some step in the algorithm. This means that $c(e_i)/l_i^{(k)}$ achieved the largest $l(\cdot)$-value at the step. Since the $l(\cdot)$-value for each edge of T decreases during the computation of the algorithm, we have $c(e_i)/l_i^{(k)} \ge c(e_1)/(l_1^{(k)} + 1) = C(T_k)$. $L_{k+1}^* = C(T_{k+1}^*) \ge c(e_i)/(l_i^* + 1) \ge c(e_i)/l_i^{(k)} \ge C(T_k)$, contradicting our assumption that $L_{k+1}^* < C(T_k)$.

We now have $l_i^{(k)} \le l_i^*$ for every $2 \le i \le n-1$. Moreover, $\sum_{i=2}^{n-1} l_i^{(k)} = \sum_{i=2}^{n-1} l_i^*$. Therefore, $l_i^{(k)} = l_i^*$ for every $2 \le i \le n-1$. It follows that $l_i^{(k+1)} = l_i^*$ for every $1 \le i \le n-1$. Hence, $T_{k+1}^* = T_{k+1}$. □

Computing a minimum spanning tree takes $O(n \log n)$ time. Sorting also takes $O(n \log_2 n)$ time. Moreover, re-setting e_i's position in the ordering takes $O(\log_2 n)$ time. Thus, the algorithm in Figure 4 takes $O(n \log n + k \log_2 n)$ time, where n is the number of terminals in P.

We conclude this section by the following theorem.

THEOREM 5. *For both rectilinear and Euclidean metrics, there exists a ratio-*2 *approximation algorithm for the bottleneck Steiner tree problem with k Steiner points that runs in* $O(n \log_2 n + k \log_2 n)$ *time, where n is the number of input terminals.*

4. Discussion. We already showed that for the bottleneck Euclidean Steiner tree problem, the optimal steinerized spanning tree is a polynomial-time approximation with performance ratio 2. This ratio may be improvable, but cannot be smaller than $\sqrt{3}$. In fact, consider $k = 1$ and three terminals that form an equilateral triangle. Then the ratio of lengths between the longest edges respectively in the optimal steinerized spanning tree and the optimal bottleneck Euclidean Steiner tree is exactly $\sqrt{3}$. We conjecture that if $NP \neq P$, the least possible performance ratio of a polynomial-time approximation for the bottleneck Euclidean Steiner tree problem is exactly $\sqrt{3}$.

The classical Steiner tree problems have been studied extensively [5], [13], [11], [14], [7], [19]. Their best known approximations can be found in [2] and [15]. Due to the importance of applications in optical networks, VLSI designs, and transportations, some variations of classical Steiner tree problems have also received more and more attention recently [4], [12], [16], [18], [20], [21], [10], [6]. Among them, a closely related problem is the Steiner tree with the minimum number of Steiner points and bounded edge-length [3], [17]. In this problem we are given a set of terminals and an upper bound for the edge-length, and we want to find a Steiner tree with the minimum number of Steiner points such that all edges meet the required upper bound on their lengths. Our approach in the previous section could give a polynomial-time approximation with the number of Steiner points up bounded by $n + 2 \cdot OPT$ where n is the number of terminals.

Acknowledgment. The authors thank two referees for their insightful comments.

References

[1] H. M. F. AboElFotoh, Algorithms for computing message delay for wireless networks, *Networks* **29** (1997), 117–124.

[2] S. Arora, Polynomial time approximation schemes for Euclidean TSP and other geometric problems, *Proc*. 37*th FOCS*, 1996.

[3] D. Chen, D.-Z. Du, X. Hu, G. Lin, L. Wang, and G. Xue, Approximations for Steiner trees with minimum number of Steiner points, submitted for publication.

[4] C. Chiang, M. Sarrafzadeh, and C. K. Wong, A powerful global router: based on Steiner min-max tree, *Proc*. *ICCAD*-89; also in *IEEE Transactions on Computer-Aided Design*, **19** (1990), 1318–1325.

[5] D.-Z. Du and F. K. Hwang, A proof of Gilbert–Pollak's conjecture on the Steiner ratio, *Algorithmica*, **7** (1992), 121–135.

[6] J. L. Ganley and J. S. Salowe, Optimal and approximate bottleneck Steiner trees, *Operations Research Letters* **19** (1996), 217–224.

[7] M. R. Garey, R. L. Graham, and D. S. Johnson, The complexity of computing Steiner minimal trees, *SIAM Journal on Applied Mathematics*, **32** (1977), 835–859.

[8] M. Gary and D. Johnson, The rectilinear Steiner tree problem is NP-complete, *SIAM Journal on Applied Mathematics* **32** (1977), 826–834.

[9] M. Gary and D. Johnson, *Computers and Intractability*, Freeman, San Francisco, CA, 1979.

[10] E. N. Gilbert, Minimum cost communication networks, *Bell System Technical Journal*, **9** (1967), 2209–2227.

[11] E. N. Gilbert and H. O. Pollak, Steiner minimal trees, *SIAM Journal on Applied Mathematics*, **16** (1968), 1–29.

[12] D. S. Hochbaum and W. Maass, Approximation schemes for covering and packing problems in image processing and VLSI, *Journal of the Association for Computing Machinery* **32** (1985), 130–136.

[13] F. K. Hwang and D. S. Richards, Steiner tree problems, *Networks* **22** (1992), 55–89.

[14] F. K. Hwang, D. Richard, and P. Winter, *The Steiner Minimum Tree Problems*, Annals of Discrete Mathematics, Vol. 53, North-Holland, Amsterdam, 1992.

[15] M. Karpinski and A. Zelikovsky, New approximation algorithms for the Steiner tree problems, *Journal of Combinatorial Optimization* **1**(1) (1997), 47–65.

[16] C.-S. Li, F. F. Tong, C. J. Georgiou, and M. Chen, Gain equalization in metropolitan and wide area optical networks using optical amplifiers, *Proc*. *IEEE INFOCOM* '94, pp. 130–137, June 1994.

[17] G.-H. Lin and G. L. Xue, Steiner tree problem with minimum number of Steiner points and bounded edge-length, *Information Processing Letters*, **69** (1999), 53–57.

[18] B. Ramamurthy, J. Iness, and B. Mukherjee, Minimizing the number of optical amplifiers needed to support a multi-wavelength optical LAN/MAN, *Proc*. *IEEE INFOCOM* '97, pp. 261–268, April 1997.

[19] J. L. Rubinstein and J. F. Weng, Compression theorems and Steiner ratios on spheres, *Journal of Combinatorial Optimization* **1**(1) (1997), 67–78.

[20] M. Sarrafzadeh and C. K. Wong, Bottleneck Steiner trees in the plane, *IEEE Transactions on Computers*, **41** (1992), 370–374.

[21] J. Soukup, On minimum cost network with nonlinear costs, *SIAM Journal on Computing*, **29** (1975), 571–581.