Approximation Algorithm for Bottleneck Steiner Tree Problem in the Euclidean Plane

Zi-Mao Li, Da-Ming Zhu, and Shao-Han Ma

School of Computer Science and Technology, Shandong University, Jinan 250061, P.R. China

E-mail: lizm@sdu.edu.cn; dmzhu@sdu.edu.cn; msh@sdu.edu.cn

Received August 15, 2003; revised April 20, 2004.

Abstract A special case of the *bottleneck Steiner tree problem* in the Euclidean plane was considered in this paper. The problem has applications in the design of wireless communication networks, multifacility location, VLSI routing and network routing. For the special case which requires that there should be no edge connecting any two Steiner points in the optimal solution, a 3-*restricted Steiner tree* can be found indicating the existence of the performance ratio $\sqrt{2}$. In this paper, the special case of the problem is proved to be *NP*-hard and cannot be approximated within ratio $\sqrt{2}$. First a simple polynomial time approximation algorithm with performance ratio $\sqrt{3}$ is presented. Then based on this algorithm and the existence of the 3-restricted Steiner tree, a polynomial time approximation algorithm with performance ratio—$\sqrt{2}+\epsilon$ is proposed, for any $\epsilon > 0$.

Keywords bottleneck Steiner tree, approximation algorithm, performance ratio, algorithm design and analysis

1 Introduction

In recent years, the *Steiner tree* problem attracts considerable attention from theoretical point of view and due to its applicability. It occupies a central place in the emerging theory of approximation algorithms.

Given a weighted graph $G = (V, E, W)$ and a subset $S \subset V$ of required vertices, the classical Steiner tree problem asks for a shortest tree spanning S. The tree may use additional points (called Steiner points) in $V - S$. We call such a tree a Steiner tree. The problem is MAX-SNP hard even when the edge weights are only 1 or 2[1].

In this paper, we consider a related variation of the classical Steiner tree problem, the *bottleneck Steiner tree* problem, which is defined as follows: 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 edge in the tree is minimized.

The problem has applications in the design of wireless communication networks, multifacility location, VLSI routing and network routing. For example, in the design of wireless communication networks, due to budget limit, suppose we could put totally $n+k$ stations in the plane, n of which must be located at given points, then we would like to choose locations for the other k stations interconnecting the n fixed locations such that the distance between stations is as small as possible.

The problem is *NP*-hard. In [2], it is shown that unless $P = NP$, the problem cannot be approximated in polynomial time within performance ratios 2 and $\sqrt{2}$ in the rectilinear plane and the Euclidean plane, respectively. Moreover, they gave an approximation algorithm with performance ratio 2 for both the rectilinear plane and the Euclidean plane. For the rectilinear plane, the performance ratio is the best possible, that is, the performance ratio is tight. For the Euclidean plane, however, the gap between the lower bound $\sqrt{2}$ and the upper bound 2 is still big. Based on the existence of a 3-restricted Steiner tree, we presented a randomized approximation algorithm with performance ratio $1.866+\epsilon$ for the Euclidean plane[3]. Later Du, Xu and Wang improved the performance ratio to $\sqrt{3}+\epsilon$[4].

In this paper, we consider a special case of the bottleneck Steiner tree problem in the Euclidean plane, which requires that there should be no edge connecting any two Steiner points in the optimal solution. We denote the problem as *special-BST* for short. In Section 2, we show the special-BST problem is *NP*-hard and cannot be approximated within ratio $\sqrt{2}$ in the Euclidean plane unless $P = NP$, and we also give a simple polynomial time $\sqrt{3}$-approximation algorithm for the problem. In Section 3, for any $\epsilon > 0$, an approximation algorithm

*Note

Supported partially by Shandong Province Excellent Middle-Aged and Young Scientists Encouragement Fund (Grant No.03BS004) and the Ministry of Education Study Abroad Returnees Research Start-up Fund, and the National Natural Science Foundation of China (Grant No.60273032).

with performance ratio $\sqrt{2}+\epsilon$ is presented. The concluding remarks appear in Section 4.

2 Ratio-$\sqrt{3}$ Approximation Algorithm

In this section, we first show the hardness result for the special-BST problem, then we propose a ratio-$\sqrt{3}$ polynomial time approximation algorithm.

Directly following the proof of Theorem 1 in [2], we have:

Theorem 1. *The special-BST problem in the Euclidean plane cannot be approximated within $\sqrt{2}$ in polynomial time unless $P = NP$.*

Now we design a polynomial time approximation algorithm with performance ratio $\sqrt{3}$ for the problem. The basic idea of the algorithm is from [2]. First we construct a minimum spanning tree for the set of n terminals P, then we add 1 Steiner point to each of the k largest edges in the minimum spanning tree. We call the resulting tree a *Steinerized spanning tree.*

The following two lemmas (Lemma 1 and Lemma 2) show that the length of the longest edge in an optimal Steinerized spanning tree is at most $\sqrt{3}$ times of the optimum, and the optimal Steinerized spanning tree can be found among the Steinerized minimum spanning trees.

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 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 which a vertex is a leaf if and only if it is a terminal. These small trees are called *full Steiner components.*

Let a and b be two points in the plane, we denote ab an edge and $|ab|$ the length of ab. Without loss of generality, we assume the length of the longest edge in the optimal Steiner tree is 1.

Lemma 1. *Let P be a set of n terminals in the Euclidean plane. There exists a Steinerized spanning tree T for P containing k Steiner points such that the length of the longest edges in T is at most $\sqrt{3}$.*

Proof. Let $T^{\min}$ be an optimal Steiner tree with k Steiner points for the special-BST problem. For there is no edge connecting every two Steiner points in the optimal Steiner tree, $T^{\min}$ can be decomposed into its full components, each of which is either a star with a Steiner point as the center or just an edge connecting two terminals.

For a star with at least 3 terminals, we can always decrease the degree of the Steiner point step by step to 2 and the length of the longest edges in the modified tree is at most $\sqrt{3}$. The procedure is as follows.

Suppose the Steiner point is labeled as v, then there must exist two terminals a and b satisfying $\angle avb \leqslant 120°$. By directly connecting a and b and removing the longer edge of va and vb, the degree of v is decreased by 1, and it is easily seen that $|ab| \leqslant \sqrt{3}$. Repeat the procedure until the degree of v becomes 2. Fig.1 gives an example.

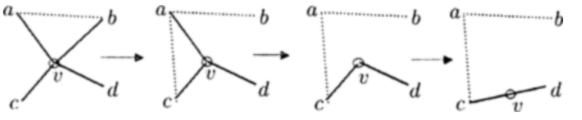

Fig.1. Transformation of T_s.

Thus we transform the star T_s into a Steiner subtree in which the length of the longest edge is at most $\sqrt{3}$. The proof completes by combining all such Steiner subtrees. □

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

Lemma 2 indicates that an optimal Steinerized spanning tree can be found among Steinerized minimum spanning trees. Since there is no edge connecting any two Steiner points, we only have to add 1 Steiner point to each of the first k longest edges in the minimum spanning tree. The complete algorithm is given in Fig.2.

ratio-$\sqrt{3}$ for special-BST
Input: A set P of n terminals in the Euclidean plane and an integer k.
Output: A Steiner tree T with at most k Steiner points.
1. Compute a minimum spanning tree.
   Suppose $e_1, e_2, \ldots, e_{n-1}$ are all edges in it.
2. Sort the edges in a non-increasing order of $|e|$.
3. Add degree-2 Steiner points to the first k longest edges.

Fig.2. $\sqrt{3}$-algorithm bottleneck problem.

Theorem 2. *The algorithm in Fig.2 is an $O(n^2 \log n)$ approximation algorithm with performance ratio $\sqrt{3}$.*

3 Ratio-$\sqrt{2}$ Approximation Algorithm

A Steiner tree for n terminals is a *k-restricted Steiner tree* if each full component spans at most k terminals. In this section, we first show there exists a 3-restricted Steiner tree containing the same number of Steiner points as in an optimal solution such that the length of the longest edge is at most $\sqrt{2}$ times of the optimum.

Without loss of generality, we assume that the length of the longest edge is 1 in an optimal Steiner tree.

Lemma 3. *For the special-BST problem, there exists a 3-restricted Steiner tree of k Steiner points such that* 1) *the length of the longest edge in the tree directly connecting two terminals is at most $\sqrt{2}$ and* 2) *the length of edges connecting a terminal and a Steiner point is at most* 1.

Proof. Similar to the proof of Lemma 1, each full component is either a star with a Steiner point as the center or just an edge connecting two terminals.

For a star with at least 4 terminals, we can always decrease the degree of the Steiner point step by step to 3 and the length of the longest edge in the modified tree is at most $\sqrt{2}$. We can use the following method.

Suppose the Steiner point is labeled as v, there must exist two terminals a and b satisfying $\angle avb \leqslant 90°$. By directly connecting a and b and removing the longer edge of va and vb, the degree of v is decreased by 1. It is easily seen that $|ab| \leqslant \sqrt{2}$. Repeat the procedure until the degree of v becomes 3. Thus we transform the star into a 3-restricted Steiner subtree in which the length of the longest edge is at most $\sqrt{2}$. By union of all the resulting Steiner subtrees, we obtain a 3-restricted Steiner tree satisfying (1) and (2). □

Now, we will transform the computation of an optimal 3-restricted Steiner tree into the *minimum spanning tree problem for 3-hypergraphs.* We need to introduce the following notions.

A *hypergraph* $H = (V, F)$ is a generalization of a graph where the edge set F is an arbitrary family of subsets of vertex set V. A *3-hypergraph* $H_3 = (V, F)$ is a hypergraph, each of whose edges has a cardinality at most 3. An *unweighted 3-hypergraph* is a 3-hypergraph with each edge weight of 1. A *minimum spanning tree* for an unweighted 3-hypergraph $H_3 = (V, F)$ is a subgraph T of H_3 that is a tree containing every node in V with the least number of edges. In other words, a minimum spanning tree for an unweighted 3-hypergraph contains as many as possible edges with cardinality 3.

Theorem 3[5]. *There is a polynomial time algorithm to compute a minimum spanning tree for an unweighted 3-hypergraph if it exists.*

To construct an unweighted 3-hypergraph, we need to know B, the length of the longest edge in an optimal solution. It is hard to find the exact value of B in an efficient way because of the hardness of the special-BST problem. However, we can guess the length of the longest edge in an optimal solution. The following procedure finds a value B' that is at most $(1+\epsilon)B$ for any $\epsilon > 0$.

Step 1. Run the ratio-$\sqrt{3}$ algorithm in Section 2 to get an upper bound X of B.

Step 2. Try to use one of $\frac{X}{\sqrt{3}}$, $\frac{X}{\sqrt{3}}\left(1+\frac{1}{p}\right)$, $\frac{X}{\sqrt{3}}\left(1+\frac{2}{p}\right), \ldots, X$ as B', where p is an integer such that $\frac{1}{p} \leqslant \epsilon$.

Thus, we can assume that $B' = (1+\epsilon)B$ is the approximation of the longest edge in an optimal solution. Now we can construct an unweighted 3-hypergraph $H_3 = (V, F)$ from the set P of terminals. The construction is as follows.

1) Connect terminals with a distance at most $\sqrt{2}B'$ and treat each of the connected components as a supernode (a terminal may also become a supernode). The distance between two supernodes is the smallest distance between two terminals, one in each supernode. These supernodes form V.

2) If there is a Steiner point v connecting 3 supernodes a, b and c such that $|va| \leqslant B'$, $|vb| \leqslant B'$ and $|vc| \leqslant B'$, then we have the edge (a, b, c) in F.

3) If the distance between two supernodes a and b is at most $2B'$, then we have the edge (a, b) in F.

4) The weight of each edge in F is 1. In fact, the weight of an edge is the number of Steiner points that connect the endpoints of the edge.

Obviously, the unweighted 3-hypergraph $H_3 = (V, F)$ can be constructed in polynomial time. According to Theorem 3, we have a polynomial time algorithm to compute an optimal 3-restricted Steiner tree. The complete algorithm is shown in Fig.3.

Theorem 4. *For any given ϵ, there exists a polynomial time approximation algorithm for the special-BST problem that computes a Steiner tree with n terminals and k Steiner points such that the length of the longest edge in the tree is at most $\sqrt{2}+\epsilon$ times of the optimum.*

ratio-$\sqrt{2}$ for special-BST

Input: A set P of n terminals in the Euclidean plane, an integer k and a positive number ϵ.

Output: A 3-restricted Steiner tree T with at most k Steiner points.

1. Call the ratio-$\sqrt{3}$ approximation algorithm in Fig.2 for the special-BST problem and obtain an upper bound X of B (B is the length of the longest edge in an optimal solution).
2. **for** $B' = \frac{X}{\sqrt{3}}, \frac{X}{\sqrt{3}}(1+\epsilon), \frac{X}{\sqrt{3}}(1+2\epsilon), \ldots, \frac{X}{\sqrt{3}}\left(1+(\sqrt{3}-1)\epsilon \times \lceil \frac{1}{\epsilon} \rceil\right)$ **do**

2.1 Construct an unweighted 3-hypergraph $H_3 = (V, F)$ as above description;

2.2 Call the algorithm in [5] to compute a minimum spanning tree T for $H_3 = (V, F)$.

3. Consider the solution T' of the smallest B' such that $w(T') \leqslant k$.
4. Restore T' to a 3-restricted Steiner tree T according to the construction of the unweighted 3-hypergraph.
5. Output the obtained tree.

Fig.3. Algorithm for restricted bottleneck Steiner tree problem.

4 Conclusion

We mainly considered a special case of the bottleneck Steiner tree problem in the Euclidean plane. We first showed that the special case is NP-hard and cannot be approximated with ratio $\sqrt{2}$ unless $P = NP$, then we presented a simple $O(n^2 \log n)$ algorithm with performance ration $\sqrt{3}$ and based on this algorithm we finally give a $\sqrt{2}+\epsilon$-approximation algorithm running in polynomial of n and ϵ. The second algorithm almost closes the special case of bottleneck Steiner tree problem in the Euclidean plane.

References

[1] Bern M, Plassmann P. The Steiner problem with edge lengths 1 and 2. *Information Processing Letters*, 1989, 32: 171–176.

[2] Wang L, Du D Z. Approximations for a bottleneck Steiner tree problem. *Algorithmica*, 2002, 32: 554–561.

[3] Wang L, Li Z. An approximation algorithm for a bottleneck k-Steiner tree problem in the Euclidean plane. *Information Processing Letters*, 2002, 81: 151–156.

[4] Du D Z, Wang L, Xu B, The Euclidean bottleneck Steiner tree and Steiner tree with minimum number of Steiner points. In *Proceedings of the 7th Annual International Conference on Computing and Combinatorics*, Guilin, China, August 2001, *LNCS 2108*, pp.509–518.

[5] Gabow H N, Stallmann M. An augmenting path algorithm for linear matriod parity. *Combinatorica*, 1986, 6: 123–150.