Information Processing Letters 110 (2010) 672–678

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

On exact solutions to the Euclidean bottleneck Steiner tree problem ☆

Sang Won Bae [a,*], Chunseok Lee [b], Sunghee Choi [b]

[a] Dept. Computer Science, Kyonggi University, Suwon, Republic of Korea
[b] Dept. Computer Science, KAIST, Daejeon, Republic of Korea

ARTICLE INFO

Article history:
Received 30 March 2009
Received in revised form 7 April 2010
Accepted 14 May 2010
Available online 24 May 2010
Communicated by M. Yamashita

Keywords:
Computational geometry
Bottleneck Steiner tree
Exact algorithm
Farthest-color Voronoi diagram

ABSTRACT

We study the Euclidean bottleneck Steiner tree problem: given a set P of n points in the Euclidean plane and a positive integer k, find a Steiner tree with at most k Steiner points such that the length of the longest edge in the tree is minimized. This problem is known to be NP-hard even to approximate within ratio $\sqrt{2}$ and there was no known exact algorithm even for $k = 1$ prior to this work. In this paper, we focus on finding exact solutions to the problem for a small constant k. Based on geometric properties of optimal location of Steiner points, we present an optimal $\Theta(n \log n)$-time exact algorithm for $k = 1$ and an $O(n^2)$-time algorithm for $k = 2$. Also, we present an optimal $\Theta(n \log n)$-time exact algorithm for any constant k for a special case where there is no edge between Steiner points.

1. Introduction

A *bottleneck Steiner tree* (also known as a *min–max Steiner tree*) is a Steiner tree such that the length of the longest edge is minimized. We consider the *Euclidean bottleneck Steiner tree problem* with unknown k Steiner points in $\mathbb{R}^2$, described as follows:

Problem 1 *(EuclidBST).* Given n points, called *terminals*, in the Euclidean plane and a positive integer k, find a Steiner tree spanning all terminals and at most k Steiner points that minimizes the length of the longest edge.

Unlike the classical Steiner tree problem where the total length of the Steiner tree is minimized, this problem asks a Steiner tree where the maximum of the edge lengths is minimized and the Steiner points in the resulting tree can be chosen in the whole plane $\mathbb{R}^2$. The classical and the bottleneck Steiner tree problems, and their variations, have some known applications in VLSI layout [5], multi-facility location, and wireless communication network design [15].

The EuclidBST problem is known to be NP-hard to approximate within ratio $\sqrt{2}$ [15]. The best known upper bound on approximation ratio is 1.866 by Wang and Li [16]. For the special case of this problem where there must be no edge connecting any two Steiner points in the optimal solution, Li et al. [10] present a $(\sqrt{2} + \epsilon)$-factor approximation algorithm with inapproximability within ratio $\sqrt{2}$.

There has been some effort on devising an exact algorithm for finding the locations of k Steiner points. Of course, it is impossible to get a polynomial time algorithm unless $P = NP$ since the problem is NP-hard. Thus, many researchers considered the following problem [14,8].

Problem 2 *(EuclidBST-FT).* Given n terminals in the Euclidean plane and a topology tree T of n terminals and k Steiner points, find a bottleneck Steiner tree with topology T spanning all terminals.

☆ A preliminary version of this work was presented at WALCOM 2009 (Bae et al. (2009) [3]). Work by S.W. Bae was supported by the Contents Convergence Software Research Center funded by the GRRC Program of Gyeonggi Province, South Korea. Work by C. Lee and S. Choi is supported by Ministry of Culture, Sports and Tourism (MCST) and Korea Culture Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program 2009.

* Corresponding author.
E-mail addresses: swbae@kgu.ac.kr (S.W. Bae), stonecold@tclab.kaist.ac.kr (C. Lee), sunghee@tclab.kaist.ac.kr (S. Choi).

0020-0190/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.ipl.2010.05.014

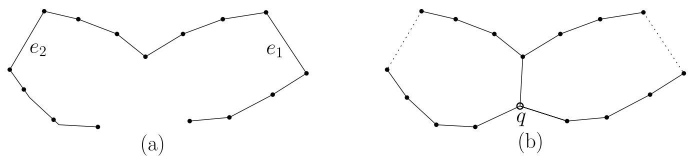

Fig. 1. (a) $MST(P)$ and (b) $MST(P \cup \{q\})$. By a single Steiner point q, the longest edge e_1 and the second longest edge e_2 are removed from $MST(P)$.

Once we have an exact algorithm to this problem, we can find an exact solution to EuclidBST by enumerating all valid topology trees; this number is roughly bounded by $(n+k)!$ [8]. To the best of our knowledge, there is no known exact algorithm to EuclidBST even for a single Steiner point $k = 1$, or even to the EuclidBST-FT problem. The first algorithms for EuclidBST-FT can be found in Elzinga et al. [7] and Love et al. [11], which are based on nonlinear optimization but do not give an exact solution. The decision version of EuclidBST-FT asks whether there exists a Steiner tree with a given topology T such that its maximum edge length does not exceed a given parameter $\lambda > 0$. Sarrafzadeh and Wong [14] presented an $O((n+k)\log(n+k))$ time algorithm for this decision problem. Indeed, one can get a simple $(1+\epsilon)$-approximation to EuclidBST-FT using the decision algorithm in a binary-search fashion [8].

In the rectilinear case where each edge of Steiner trees should be rectilinear, a quadratic-time exact algorithm for finding a bottleneck Steiner tree with a given topology is known by Ganley and Salowe [8]; they also described the difficulty of the Euclidean case.

In this paper, we present an $O(n\log n)$ time algorithm for the EuclidBST problem when $k = 1$ and an $O(n^2)$ time algorithm when $k = 2$. Our approach is rather from a geometric point of view: we reveal several useful properties of Euclidean bottleneck Steiner trees and optimal locations for Steiner points based on basic Euclidean geometry. Among them is an interesting connection between the optimal location of Steiner points and the *smallest color spanning disk*, a smallest disk containing at least one point of each color when we are given a set of colored points [1]. Also, we present an $O(n\log n)$ time algorithm for a special case of EuclidBST where there is no edge between Steiner points; the hidden constant is essentially exponential in k. We remark that the running times of our algorithms are all polynomial in n; any exact solution to EuclidBST-FT seems hard to yield an exact algorithm running in time polynomial in n due to the number of possible topologies [8]. Moreover, our observations can be naturally extended to other metrics on $\mathbb{R}^2$, such as the rectilinear case.

The paper is organized as follows: Section 2 presents an exact algorithm for a single Steiner point. Next, we show several helpful properties of optimal solutions for any k, and make use of them to devise the algorithms for the case without edges between Steiner points and for the case of $k = 2$ in Section 3. Finally, Section 4 concludes the paper.

2. Exact algorithm for a single Steiner point

Let $P \subset \mathbb{R}^2$ be a set of n points; we call each point in P a *terminal*. A (*Euclidean*) *bottleneck spanning tree* of P is a spanning tree of P such that the length of a longest edge is minimized. We call the length of a longest edge in a bottleneck spanning tree of P the *bottleneck of the set* P, denoted by $b(P)$. Note that $b(P)$ is dependent only on the set P, not on how to connect the points in P. Our problem is to find an optimal location $q \in \mathbb{R}^2$ of a Steiner point such that the bottleneck $b(P \cup \{q\})$ is minimized. We start with a (*Euclidean*) *minimum spanning tree* $MST(P)$ of given points P. Throughout the paper, we shall use the following common fact without proof; it can be seen as an exercise in the textbook by Cormen et al. [6, Problems 23-3].

Lemma 1. *Any minimum spanning tree of P is a bottleneck spanning tree of P.*

From now on, we denote by $MST(P)$ any fixed minimum spanning tree of P. Let $e_1, \ldots, e_{n-1}$ be the edges of $MST(P)$ in the order that their lengths are not increasing. Obviously, $|e_1| = b(P) \geqslant b(P \cup \{q\})$, where $|e|$ denotes the length of an edge (or a segment) e. In order to have the strict inequality $b(P) > b(P \cup \{q\})$, we must be able to remove the longest edge e_1 of $MST(P)$ after adding q. Then, q would connect two points in P as a substitution of e_1. Sometimes, q with new edges incident to it can replace the second longest edge e_2 simultaneously with e_1, as shown in Fig. 1. But it is obvious that effort for removing e_3 without removing e_2, or in general removing e_i without removing e_{i-1} is useless in reducing the bottleneck since the length of the longest edge is lower bounded by $|e_{i-1}| \geqslant |e_i|$. Thus, it is sufficient to search such $q \in \mathbb{R}^2$ that we lose in the resulting BST all of $e_1, \ldots, e_c$ from $MST(P)$ for some positive integer $c < n$. Fortunately, the following observations allow us to focus on a constant number of edges to be removed.

Lemma 2. *For any minimum spanning tree $MST(P)$ of P, there exists a bottleneck Steiner tree T^* of P with a single Steiner point q such that T^* is a minimum spanning tree of $P \cup \{q\}$ and each edge of T^* either belongs to $MST(P)$ or is incident to the Steiner point q.*

Proof. Let q be an optimal location for the Steiner point and $T^{(1)}$ be a minimum spanning tree of $P \cup \{q\}$. By Lemma 1, $T^{(1)}$ is a bottleneck Steiner tree for P with a single Steiner point.

Suppose that $T^{(1)}$ violates the condition of the lemma. Then, there are two terminals $p_1, p_2 \in P$ such that the edge p_1p_2 belongs to $T^{(1)}$ but not to $MST(P)$. Let π be the unique path in $MST(P)$ between p_1 and p_2, and P_1 and P_2 be the bipartition of P obtained by cutting p_1p_2 from $T^{(1)}$. Note that π excludes the edge p_1p_2. Then, there must exist an edge e on π such that $e \neq p_1p_2$ and e

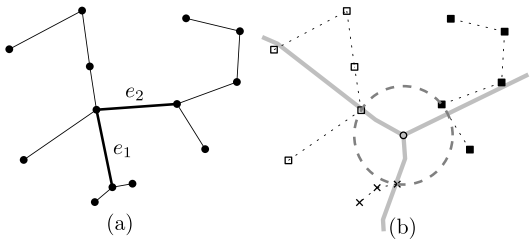

Fig. 2. Illustration of the algorithm for $c = 2$. (a) For a given set P of points, compute $MST(P)$ and remove e_1 and e_2. (b) Color T_i using different colors, and compute the farthest-color Voronoi diagram (gray thick segments) to find the smallest color spanning disk (gray dashed circle). (c) Finally, locate the Steiner point at the center of the disk and add necessary edges to complete the resulting tree.

connects P_1 and P_2; that is, e is between one terminal in P_1 and another in P_2. We perform the following operation: From $T^{(1)}$, remove the edge $p_1 p_2$ and add e. Denote by $T^{(2)}$ the resulting spanning tree on $P \cup \{q\}$.

Observe by optimality of $MST(P)$ that the length of each edge on π is at most $|p_1 p_2|$; we thus have $|e| \leqslant |p_1 p_2|$. Therefore, the longest edge length and the total edge length of $T^{(2)}$ do not exceed those of $T^{(1)}$, and hence $T^{(2)}$ is another bottleneck Steiner tree for P with Steiner point q. We repeat the above operation to have $T^{(3)}, T^{(4)}, \ldots$. Since each execution of the operation decreases the number of such edges $p_1 p_2$ by one, we obtain a bottleneck Steiner tree T^* with the claimed property after performing the operation a finite number of times. □

Lemma 3. *Let $MST(P)$ be any fixed minimum spanning tree of P and $e_1, \ldots, e_{n-1}$ be the edges of $MST(P)$ in the edge-length non-increasing order. Then, there exists a bottleneck Steiner tree T^* of P with a single Steiner point q such that T^* is a minimum spanning tree of $P \cup \{q\}$ and the degree of the Steiner point q in T^* is at most 5. Therefore, $b(P \cup \{q'\}) \geqslant |e_5|$ holds for any $q' \in \mathbb{R}^2$ if $n > 5$.*

Proof. Let q be an optimal location for the Steiner point and T be a bottleneck Steiner tree for P with the Steiner point q satisfying the conditions of Lemma 2. Let m be the degree of q in T, and $p_1, \ldots, p_m \in P$ be the m neighbors of q in clockwise order around q.

Now, suppose that $m \geqslant 6$. Then, there exists an integer a with $1 \leqslant a \leqslant m$ such that the angle $\angle p_a q p_{a+1}$ is at most 60°. (Assume that the index is taken by modulo m.) By simple trigonometry, $|p_a p_{a+1}|$ is at most $|qp_a|$ or $|qp_{a+1}|$, that is, $|p_a p_{a+1}| \leqslant \max\{|qp_a|, |qp_{a+1}|\}$. Thus, we can replace the longer of qp_a and qp_{a+1} by $p_a p_{a+1}$ without increasing the longest edge length and the total edge length in the resulting Steiner tree. We perform the following operation: We first remove the longer of qp_a and qp_{a+1} and add $p_a p_{a+1}$, and then if $p_a p_{a+1}$ does not belong to $MST(P)$, we execute the operation described in the proof of Lemma 2 to replace $p_a p_{a+1}$ by an edge in $MST(P)$. (Observe that the operation does not increase the degree of the Steiner point q.) We repeat the above operation until we have at most 5 neighbors of q in the resulting Steiner tree T^*.

Observe that T^* is still a minimum spanning tree of $P \cup \{q\}$ since the above operation does not increase the total edge length. This shows the existence of a bottleneck Steiner tree T^* satisfying the following properties: (i) T^* is a minimum spanning tree of $P \cup \{q\}$, (ii) the degree of q is at most 5, and (iii) each edge of T^* either belongs to $MST(P)$ or is incident to q. By properties (ii) and (iii), T^* consists of at least $n - 5$ edges from $MST(P)$ plus at most 5 edges incident to q. In the best case, q removes the four longest edges $e_1, \ldots, e_4$ from $MST(P)$. Thus, if $n = |P| > 5$, we have $|e_5| \leqslant b(P \cup \{q\}) \leqslant b(P \cup \{q'\})$ for any $q' \in \mathbb{R}^2$, as claimed. □

By the above lemmas, we can remove at most four longest edges from any fixed $MST(P)$, and further newly added edges are all incident to q. Thus, we have only four possibilities; $e_1, \ldots, e_c$ are removed from $MST(P)$, where $c = 1, \ldots, 4$. Our algorithm is summarized as follows (see Fig. 2):

(1) Remove $e_1, \ldots, e_c$ from $MST(P)$. Then, we have a forest containing $c + 1$ subtrees $T_1, \ldots, T_{c+1}$.
(2) Find a smallest disk containing at least one terminal from each subtree T_i. (This also minimizes the maximum length of edges incident to q.)

In the second step, we indeed ask a solution to the *smallest color spanning disk*: letting $C := \{1, 2, \ldots, c+1\}$ be a color set, we color each terminal of T_i by color $i \in C$. A brute force approach is to evaluate every possible combination by selecting a terminal from each subtree T_i and constructing the smallest enclosing disk, taking $O(n^{c+1})$ time for fixed c. More efficiently, the smallest color spanning disk can be computed in $O(cn \log n)$ time by computing the *farthest color Voronoi diagram* [9,1]: given a collection $\mathcal{C} = \{P_1, \ldots, P_c\}$ of c sets of colored points, define the *distance to a color* $i \in C$ as $d(x, i) := \min_{p \in P_i} |xp|$. Then, the farthest color Voronoi diagram $FCVD(\mathcal{C})$ is the farthest Voronoi diagram of the colors under the distance to colors. Like the standard (Euclidean) farthest Voronoi diagram, $FCVD(\mathcal{C})$ has an application to finding a *center* minimizing the maximum distance to all colors, which is exactly the center of a smallest color spanning disk. Also, such a center is located on a vertex or an edge of $FCVD(\mathcal{C})$. The combinatorial complexity of $FCVD(\mathcal{C})$ is known to be $O(cn)$, where n is the total number of points contained in $\mathcal{C}$.

More precisely, we regard each T_i as a set of points with color i: let $\mathcal{C} := \{T_i \mid 1 \leqslant i \leqslant c+1\}$, and build $FCVD(\mathcal{C})$ to compute the center of a smallest color spanning disk, which is a candidate of an optimal location of the Steiner point. Repeating this procedure for $c = 1, \ldots, 4$ gives us four candidates of an optimal location of the Steiner point.

Theorem 1. *Given a set P of n points in the plane, a Euclidean bottleneck Steiner tree with a single Steiner point can be exactly computed in $O(n \log n)$ time with $O(n)$ space, and this time bound is worst-case tight in the algebraic decision tree model.*

Proof. It takes $O(n \log n)$ time to compute the minimum spanning tree of P. For each iteration, it is required to compute $FCVD(\mathcal{C})$ and traverse it, spending $O(cn \log n)$ time and $O(cn)$ space for $c = 1, \ldots, 4$. Hence, the algorithm works in $O(n \log n)$ time.

To prove the lower bound of the problem, consider the *maximum gap* problem: Given n real numbers in $\mathbb{R}$, find the maximum difference between two consecutive numbers when they are sorted. This problem has an $\Omega(n \log n)$ lower bound in the algebraic decision tree model by Ben-Or [4]; another proof can be found in the book by Preparata and Shamos [13]. For any instance of the maximum gap problem, we transform the given numbers into the points in $\mathbb{R}^2$ along the x-axis. Any algorithm for the Euclidean bottleneck Steiner tree problem locates a Steiner point between two points with the maximum gap. Thus, two points adjacent to the Steiner point are from the two consecutive numbers defining the maximum gap. Hence, finding the optimal location of a single Steiner point needs at least $\Omega(n \log n)$ steps in the algebraic decision tree model. □

3. Towards optimal location of multiple Steiner points

In this section, we extend the above discussion for a single Steiner point to multiple $k > 1$ Steiner points. We again denote by $MST(P)$ any fixed minimum spanning tree of P and by $e_1, \ldots, e_{n-1}$ its edges sorted by edge length. Following is a key lemma throughout this section.

Lemma 4. *Let k be the number of allowed Steiner points. Then, there exists a Euclidean bottleneck Steiner tree T^* for P with k Steiner points $Q = \{q_1, \ldots, q_k\}$ such that the following properties hold*:

(i) *T^* is a minimum spanning tree of $P \cup Q$,*
(ii) *the degree of q_i is at most 5 for each $i = 1, \ldots, k$, and*
(iii) *each edge of T^* either belongs to $MST(P)$ or is incident to q.*

Therefore, $b(P \cup \{q'_1, \ldots, q'_k\}) \geqslant |e_{4k+1}|$ holds for any k points $q'_1, \ldots, q'_k \in \mathbb{R}^2$ if $n > 4k + 1$.

Proof. In the proof, we make use of the following well-known fact on the Euclidean minimum spanning tree:

There always exists a Euclidean minimum spanning tree of any given finite set of points in the plane whose degree is at most five.

The above fact is regarded as a folklore in computational geometry[1]; for example, see Papadimitriou and Vazirani [12].

[1] Personal communication with Otfried Cheong.

Let $Q = \{q_1, \ldots, q_k\}$ be a set of optimal locations of k Steiner points. Then, the above fact implies the existence of a (Euclidean) minimum spanning tree $T^{(1)}$ of $P \cup Q$ whose degree is at most 5. By Lemma 1, $T^{(1)}$ is indeed a Euclidean bottleneck Steiner tree for the set P of terminals with k Steiner points Q. Observe that $T^{(1)}$ satisfies properties (i) and (ii).

Initially from $T^{(1)}$, we repeatedly apply the operation described in the proof of Lemma 2 to have $T^{(2)}, T^{(3)}, \ldots$. Let T^* be the final output of the sequence of operations. As discussed in the proof of Lemma 2, we know that T^* satisfies properties (i) and (iii). On the other hand, observe that the operation does not affect the degree of each Steiner point q_i; at each execution of the operation, we remove and add an edge between two terminals. Thus, the operation preserves property (ii). This shows the existence of a bottleneck Steiner tree T^* with properties (i)–(iii).

By properties (ii) and (iii), $T^* - Q$ is a forest consisting of at most $4k + 1$ subtrees of $MST(P)$. Thus, when the number n of terminals is more than $4k + 1$, we have a lower bound as follows:

$$|e_{4k+1}| \leqslant b(P \cup Q) = \min_{q'_1, \ldots, q'_k \in \mathbb{R}^2} b\big(P \cup \{q'_1, \ldots, q'_k\}\big).$$

This proves the second part of the lemma. □

In this section, we introduce another variation of the problem, namely the *Euclidean bottleneck Steiner tree with fixed topology on subtrees*.

Problem 3 *(EuclidBST-FT-ST).* Given a set P of n points (terminals) in the plane, positive integers k and c with $c \leqslant 4k$, and a topology tree $\mathcal{T}$ on the $c + 1$ subtrees T_i and k Steiner points, find a set of optimal locations of k Steiner points to obtain a Euclidean bottleneck Steiner tree with the given topology $\mathcal{T}$.

Let $V := \{v_1, \ldots, v_{c+1}, s_1, \ldots, s_k\}$ be the vertex set of $\mathcal{T}$, where v_i represents T_i and s_j represents a Steiner point. Each Steiner point does not have degree 1 but can have degree 2 in the bottleneck Steiner tree [14]. Together with Lemma 4, we can restrict $\mathcal{T}$ so that each s_j has degree between 2 and 5.

3.1. A special case without edges between two Steiner points

The idea of the single Steiner point location can be used to solve a special case of the EuclidBST, where the resulting Steiner tree should have no edge between two Steiner points; for fixed c with $k \leqslant c \leqslant 4k$, we remove $e_1, \ldots, e_c$ from $MST(P)$ to get $c+1$ subtrees $T_1, \ldots, T_{c+1}$. Then, for a fixed topology $\mathcal{T}$ on all T_i and k Steiner points as vertices, each Steiner point s_j is located at the center of the smallest color spanning disk of $\{T_i \mid T_i \text{ adjacent to } s_j \text{ in } \mathcal{T}\}$. Hence, for a given c and a topology tree $\mathcal{T}$ as input of EuclidBST-FT-ST, we can find an optimal location of k Steiner points Q in $O(cn \log n)$, when we do not allow edges between Steiner points in $\mathcal{T}$.

For each $k \leqslant c \leqslant 4k$, enumerating all such topologies gives us an exact solution to EuclidBST. We now count the number of possible topologies $\mathcal{T}$ without edges between two Steiner points.

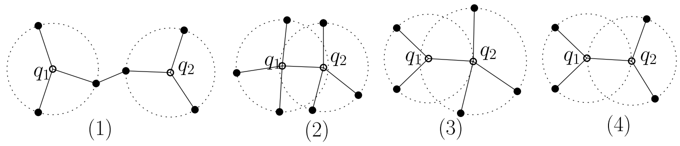

Fig. 3. Illustration to the four cases of optimal locations for two Steiner points. Small black circles are two Steiner points q_1 and q_2, and black dots represent terminals that are determinators of q_1 or q_2. D_1 and D_2 are depicted as dotted circles centered at q_1 and q_2.

Lemma 5. *There are at most $O((4k)!k!/6^k)$ possible topology trees as input of* EuclidBST-FT-ST, *where there is no edge between two Steiner points, for any positive integers k and c with $k \leqslant c \leqslant 4k$.*

Proof. Since each Steiner point has degree at most 5 and $c \leqslant 4k$ by Lemma 4, we can roughly count the number of possible topologies by choosing 5 subtrees for each Steiner point. For the first Steiner point s_1, it has at most $\binom{4k+1}{5}$ number of possibilities to choose five subtrees to connect. For the second one s_2, one of the five subtrees chosen by s_1 should be connected to s_2 without loss of generality; thus the number of its possibilities is at most $5 \cdot \binom{4k-4}{4}$. In this way, s_i has $(4i-3) \cdot \binom{4(k-i)+4}{4}$ possibilities to choose five subtrees. Then, we have at most

$$\binom{4k+1}{5} \cdot 5\binom{4k-4}{4} \cdots (4i-3) \times \binom{4(k-i)+4}{4} \cdots (4k-3)\binom{4}{4} = O\left((4k)!k!/6^k\right)$$

possible topology trees for any k and $c \leqslant 4k$, where there is no edge between Steiner points. □

Note that there was no known exact algorithm even for this special case of the Euclidean bottleneck Steiner tree problem; only a $(\sqrt{2}+\epsilon)$-approximation algorithm with inapproximability within ratio $\sqrt{2}$ is known for this special case [10]. Thus, it is worth noting the following theorem.

Theorem 2. *Given n terminals and the number k of Steiner points, a Euclidean bottleneck Steiner tree with no edge between Steiner points can be computed in $O((4k)!k!k^2/6^k \cdot n\log n)$ time.*

Proof. As discussed above, EuclidBST-FT-ST can be solved in $O(cn\log n)$ time. And for fixed c, we have $O((4k)!k!/6^k)$ possible topology trees by Lemma 5 and it is easy to see that each enumeration can be done in $O(1)$ time. A rough calculation results in $\sum_{c=k}^{4k} O((4k)!k!/6^k \cdot cn\log n) = O((4k)!k!k^2/6^k \cdot n\log n)$. □

The running time appeared in Theorem 2 is exponential in k but polynomial in n. Thus, our algorithm runs in $O(n\log n)$ time for any constant k. This is remarkable since there was no known exact algorithm polynomial in n for the bottleneck Steiner tree problem for any fixed k even for the rectilinear case.

In order to allow edges between Steiner points, we need more geometric observations. In the following subsection, we show some properties of optimal locations of Steiner points.

3.2. Properties of the optimal solutions

Consider a set of optimal locations, $Q = \{q_1, \ldots, q_k\}$, of Steiner points and its resulting Euclidean Steiner tree T^*; the optimal location of $s_i \in V$ is denoted by $q_i \in \mathbb{R}^2$. Let N_i be the set of points in $P \cup Q$ that are adjacent to q_i in T^*, r_i be the length of the longest edge incident to q_i in T^*, and D_i be the disk centered at q_i with radius r_i. By local optimization, we can force each D_i to have two or three points in N_i on its boundary. Note that a disk is said to be *determined by three points* or *by two diametral points* if the three points lie on the boundary of the disk or if the two points define the diameter of the disk, respectively.

Lemma 6. *There exists a set of optimal locations of k Steiner points, $q_1, \ldots, q_k$, such that* (1) *D_i is determined by two diametral points or by three points in N_i, each of which belongs to different components T_j and that* (2) *if there is a terminal $p \in T_j \cap N_i$ on the boundary of D_i, then there is no other terminal $p' \in T_j$ contained in the interior of D_i.*

Proof. As explained above, it is easy to see that we can move each D_i to hold condition (1) without increasing its radius r_i. Now, suppose that $q_1, \ldots, q_k$ is an optimal solution satisfying condition (1) but not condition (2), so that there is D_i such that $p, p' \in D_i \cap T_j$ and p lies on the boundary of D_i. Then, we can discard the farther one p from q_i and connect q_i to p', instead. Therefore, the lemma is shown. □

Thus, we can assume that our optimal location Q of Steiner points satisfies the above properties. We call the two or the three points in N_i determining D_i the *determinators* of D_i. If the determinators of D_i are all terminals, the position of q_i is rather fixed by Lemma 6; we call such q_i *solid*. Otherwise, if at least one of the determinators of D_i is a Steiner point, then q_i is called *flexible*.

3.3. Exact algorithm for $k = 2$

Based on the properties observed above, we present an exact algorithm for computing the Euclidean bottleneck Steiner tree with two Steiner points.

We classify possible optimal locations of two Steiner points as follows (see Fig. 3):

(1) There is no edge between q_1 and q_2, that is, $q_1 \notin N_2$ and $q_2 \notin N_1$.
(2) $q_1 \in N_2$, and both q_1 and q_2 are solid.
(3) $q_1 \in N_2$, and one of q_1 and q_2 is solid and the other is flexible.
(4) $q_1 \in N_2$, and both q_1 and q_2 are flexible.

Given $c \leqslant 8$ and a topology tree $\mathcal{T}$ as above, if $\mathcal{T}$ has no edge between s_1 and s_2, then we can find an optimal solution as in Theorem 2. Thus, here we focus on cases (2)–(4).

In the case when $\mathcal{T}$ has the edge between s_1 and s_2, other edges incident to s_1 or s_2 cover all the v_i in $\mathcal{T}$. Let $C := \{1, 2, \ldots, c+1\}$ and $C_j \subset C$ be defined as $C_j := \{i \in C \mid v_i \text{ adjacent to } s_j\}$, for $j = 1, 2$. Obviously, C_1 and C_2 are disjoint and $C_1 \cup C_2 = C$. Let $\mathcal{C}_j := \{T_i \mid i \in C_j\}$ be the two sets of colored points induced from the subtrees T_i.

Consider an optimal location $Q = \{q_1, q_2\}$. If q_1 is solid, then D_1 is a color spanning disk of $\{T_i \mid i \in C_1\}$ and is determined by two or three points from different T_i by Lemma 6; that is, q_1 lies on a vertex or an edge of the farthest color Voronoi diagram *FCVD*($\mathcal{C}_1$) of $\mathcal{C}_1$. (This process is almost the same as done for the case of $k = 1$.) Hence, the number of possible positions of q_1, provided that q_1 is solid, is $O(|C_1|n)$.

Note that finding the location q_1 of a solid Steiner point is not independent from finding q_2 even if q_2 is solid, since the distance between q_1 and q_2 does matter in minimizing the bottleneck. Nonetheless, since q_1 is the center of a color spanning disk of $\mathcal{C}_1$ if q_1 is solid, we are done by checking all the possible $O(|C_1|n)$ locations for q_1. On the other hand, possible determinators of q_2 can be found from *FCVD*($\mathcal{C}_2$), and the number of such possible sets of determinators of q_2 is at most $O(|C_2|n)$. Hence, checking all pairs of each possible location for q_1 and each possible set of determinators for q_2, we can find an optimal solution minimizing $b(P \cup \{q_1, q_2\})$ in $O(c^2n^2)$ time.[2]

What remains is case (4) where both q_1 and q_2 are flexible. In this case, we have $r_1 = r_2$ since q_1 is a determinator of D_2 and q_2 is a determinator of D_1. The other determinators of D_1 except q_2 are of course terminals in P. There are two cases; D_1 has three determinators or two. In the former case, two terminals p_1 and p_2 are determinators of D_1 and q_1 lies on an edge of *FCVD*($\mathcal{C}_1$) determined by p_1 and p_2 since $|p_1q_1| = |p_2q_1| = r_1$. In the latter case where one terminal p lies on the boundary of D_1, q_1 is the midpoint on segment p_1q_2.

Thus, we can find an optimal location in case (4) in $O(c^2n^2)$ time as follows:

1. Choose $p_1 \in T_i \in \mathcal{C}_1$ and $p_2 \in T_j \in \mathcal{C}_2$. Let x_1 and x_2 be two points on segment p_1p_2 such that $|p_1x_1| = |x_1x_2| = |x_2p_2| = \frac{1}{3}|p_1p_2|$. Test whether x_1 lies in the cell of T_i in *FCVD*($\mathcal{C}_1$) and x_2 lies in the cell of T_j in *FCVD*($\mathcal{C}_2$). If this test is passed, $\{x_1, x_2\}$ is a candidate of an optimal location $\{q_1, q_2\}$ of two Steiner points where both D_1 and D_2 have only one terminal on each of their boundaries.
2. Choose $p_1 \in T_i \in \mathcal{C}_1$ and an edge e from *FCVD*($\mathcal{C}_2$). Note that e is a portion of the bisecting line of two points $p \in T_j$ and $p' \in T_{j'}$ for some $j, j' \in C_2$. Then, we find a point $x_2 \in e$ such that $2|x_2p| = |x_2p_1|$, if any, and let x_1 be the midpoint on segment x_2p_1. Test whether x_1 lies in the cell of T_i in *FCVD*($\mathcal{C}_1$). If the test is passed, $\{x_1, x_2\}$ is a candidate of an optimal location $\{q_1, q_2\}$ of two Steiner points where D_1 has one terminal and D_2 has two on its boundary, respectively. The case where two terminals lie on the boundary of D_1 and one lies on the boundary of D_2 can be handled in a symmetric way.
3. Choose an edge e_1 from *FCVD*($\mathcal{C}_1$) and e_2 from *FCVD*($\mathcal{C}_2$). Let $p_1 \in T_i$ and $p'_1 \in T_{i'}$ for $i, i' \in C_1$ be the terminals such that e_1 is from the bisecting line between p_1 and p'_1. And let $p_2 \in T_j$ and $p'_2 \in T_{j'}$ for $j, j' \in C_2$ be the terminals such that e_2 is from the bisecting line between p_2 and p'_2. Then, find two points $x_1 \in e_1$ and $x_2 \in e_2$ such that $|p_1x_1| = |x_1x_2| = |x_2p_2|$, if any. Then, $\{x_1, x_2\}$ is a candidate of an optimal location $\{q_1, q_2\}$ of two Steiner points where two terminals lie on the boundary of each of D_1 and D_2.

Consequently, we can find an optimal location q_1, q_2 by examining all pairs of terminals and edges of *FCVD*($\mathcal{C}_1$) and *FCVD*($\mathcal{C}_2$). Since the complexity of these diagrams is bounded by $O(cn)$, $O(c^2n^2)$ time is sufficient to compute an optimal location of two Steiner points in case (4). To find an optimal bottleneck Steiner tree with two Steiner points, we enumerate all possible topologies with two Steiner points and repeat the above process for each $1 \leqslant c \leqslant 8$. Finally, we conclude the following theorem.

Theorem 3. *Given a set P of n points in the plane, a Euclidean bottleneck Steiner tree with two Steiner points can be exactly computed in $O(n^2)$ time with $O(n)$ space.*

4. Concluding remarks

We presented exact algorithms for the Euclidean bottleneck Steiner tree problem when the number of allowed Steiner points is one or two. In doing so, we revealed an interesting relation between the optimal location of Steiner points and the farthest color Voronoi diagram.

One remarkable fact is that our approach can be naturally extended to other metric spaces, such as the rectilinear case (the L_1 metric); our proofs are not very dependent on the Euclidean geometry. For example, in the rectilinear case, one could show that there exists a rectilinear bottleneck Steiner tree where each Steiner point is of degree at most 7 and the farthest color Voronoi diagram with respect to the L_1 metric, which has been also well understood [9], could be used to build an algorithmic block as we did for the Euclidean case.

We should report that the above idea of generalization has been successfully developed by the authors and Tanigawa during preparation of this article and recently presented at the *20th International Symposium on Algorithms and Computation* (ISAAC 2009) [2].

Acknowledgement

The authors thank anonymous referees for valuable comments to improve our paper.

References

[1] M. Abellanas, F. Hurtado, C. Icking, R. Klein, E. Langetepe, L. Ma, B. Palop, V. Sacristán, The farthest color Voronoi diagram and related

[2] In a preliminary version [3] of the present paper, the explanation for cases (2) and (3) was not correct. This fixes the incorrectness.

problems, Technical Report 002, Institut für Informatik I, Rheinische Fridrich-Wilhelms-Universität, Bonn, 2006.
[2] S.W. Bae, S. Choi, C. Lee, S. Tanigawa, Exact algorithms for the bottleneck Steiner tree problem, in: Proc. 20th Annu. Internat. Sympos. Alg. Comput. (ISAAC), in: LNCS, vol. 5878, 2009, pp. 24–33.
[3] S.W. Bae, C. Lee, S. Choi, On exact solutions to the Euclidean bottleneck Steiner tree problem, in: Proc. 3rd Workshop Alg. Comput. (WALCOM), 2009, pp. 105–116.
[4] M. Ben-Or, Lower bounds for algebraic computation trees, in: Proc. 15th Annu. ACM Sympos. Theory Comput. (STOC), ACM, New York, NY, USA, 1983, pp. 80–86.
[5] C. Chiang, M. Sarrafzadeh, C. Wong, A powerful global router: Based on Steiner min–max trees, in: Proc. IEEE Int. Conf. CAD, 1989, pp. 2–5.
[6] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, 2nd edition, MIT Press, 2001.
[7] J. Elzinga, D. Hearn, W. Randolph, Minimax multifacility location with Euclidean distances, Transport. Sci. 10 (1976) 321–336.
[8] J.L. Ganley, J.S. Salowe, Optimal and approximate bottleneck Steiner trees, Oper. Res. Lett. 19 (1996) 217–224.
[9] D.P. Huttenlocher, K. Kedem, M. Shrir, The upper envelope of Voronoi surfaces and its applications, Discrete Comput. Geom. 9 (1993) 267–291.
[10] Z.-M. Li, D.-M. Zhu, S.-H. Ma, Approximation algorithm for bottleneck Steiner tree problem in the Euclidean plane, J. Comput. Sci. Tech. 19 (6) (2004) 791–794.
[11] R. Love, G. Wesolowsky, S. Kraemer, A multifacility minimax location problem with Euclidean distances, J. Prod. Res. 11 (1973) 37–45.
[12] C.H. Papadimitriou, U.V. Vazirani, On two geometric problems related to the travelling salesman problem, J. Algorithms 5 (1984) 231–246.
[13] F.P. Preparata, M.I. Shamos, Computational Geometry, Springer-Verlag, 1985.
[14] M. Sarrafzadeh, C. Wong, Bottleneck Steiner trees in the plane, IEEE Trans. Comput. 41 (3) (1992) 370–374.
[15] L. Wang, D.-Z. Du, Approximations for a bottleneck Steiner tree problem, Algorithmica 32 (2002) 554–561.
[16] L. Wang, Z. Li, An approximation algorithm for a bottleneck k-Steiner tree problem in the Euclidean plane, Inform. Process. Lett. 81 (2002) 151–156.