

# On Exact Solutions to the Euclidean Bottleneck Steiner Tree Problem\*

Sang Won Bae, Chunseok Lee, and Sunghee Choi

Division of Computer Science, KAIST, Korea  
{swbae, stonecold, sunghee}@tclab.kaist.ac.kr

**Abstract.** We study the Euclidean bottleneck Steiner tree problem: given a set  $P$  of  $n$  points in the Euclidean plane, called terminals, 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  $\sqrt{2}$ . 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  $O(n \log n)$  time exact algorithm for  $k = 1$  and an  $O(n^2)$  time algorithm for  $k = 2$ . Also, we present an  $O(n \log n)$  time exact algorithm to the problem for a special case where there is no edge between Steiner points.

## 1 Introduction

A *bottleneck Steiner tree* (BST) (also known as a *min-max Steiner tree*) is a Steiner tree with the length of the longest edge 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  $\mathbb{R}^2$ . These problems, and their variations, have some known applications in VLSI layout [3], multifacility location, and wireless communication network design [11].

The EUCLIDBST problem is known to be NP-hard to approximate within ratio  $\sqrt{2}$  [11]. The best known upper bound on approximation ratio is 1.866 by Wang and Li [12]. For the special case of this problem where there should be no edge connecting any two Steiner points in the optimal solution, Li et al. [7] present a  $(\sqrt{2} + \epsilon)$ -factor approximation algorithm with inapproximability within  $\sqrt{2}$ .

---

\* Work by S.W. Bae was supported by the Brain Korea 21 Project. Work by C. Lee and S. Choi was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MOST) (No. R01-2007-000-20865-0).

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 [5, 10].

*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.

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  $(n+k)!$  [5]. 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. [4] and Love et al. [8], which are based on non-linear 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 [10] 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 [5].

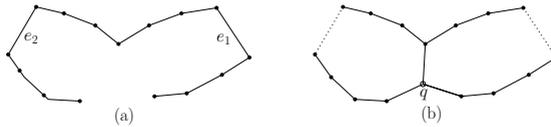
In the rectilinear case where each edge of Steiner trees should be rectilinear, a quadratic-time exact algorithm for finding a bottleneck Steiner tree with a given topology is known by Ganley and Salowe [5]; they also described the difficulty of the Euclidean case.

In this paper, we present an  $O(n \log n)$  time algorithm to 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 using geometric knowledge. 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 any constant  $k$  to 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 hardly to yield an exact algorithm running in time polynomial in  $n$  due to the number of possible topologies [5]. Moreover, our observations can be naturally extended to any metric on  $\mathbb{R}^2$ ; for example, 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 the optimal location  $q \in \mathbb{R}^2$  of a Steiner point such that the bottleneck  $b(P \cup \{q\})$  of any bottleneck spanning tree of  $P \cup \{q\}$  is minimized. We start with a (*Euclidean*) *minimum spanning tree*  $MST(P)$  of given points  $P$ . Obviously,  $MST(P)$  is a bottleneck spanning tree of  $P$ . Since computing a minimum or bottleneck spanning tree of a given set can be done easily, our problem can be viewed as finding an optimal location of  $q$  to minimize the bottleneck of  $P \cup \{q\}$ .



**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  $MST(P)$ .

Let  $e_1, \dots, e_{n-1}$  be the edges of  $MST(P)$  in the order that their lengths are not increasing. Obviously,  $|e_1| = b(P) \geq b(P \cup \{q\})$ , 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  $e_1$ . 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  $|e_{i-1}| \geq |e_i|$ . Thus, in  $MST(P \cup \{q\})$ , we lose all of  $e_1, \dots, e_c$  from  $MST(P)$ , where some positive integer  $c < n$ . Also, note that  $MST(P)$  might have many longest edges, and thus all of them must be removed to have the strictly better bottleneck. The number does not exceed  $n - 1$  but we could really have that many number of longest edges in  $MST(P)$ . Fortunately, the following observations allow us to focus on a constant number of edges to be removed.

**Lemma 1.** *There exists a bottleneck Steiner tree of  $P$  with a single Steiner point such that each edge belongs to  $MST(P)$  or is incident to the Steiner point  $q$ .*

*Proof.* Suppose that a bottleneck Steiner tree  $MST(P \cup \{q\})$  has an edge connecting two terminals  $p_1, p_2 \in P$  which does not belong to  $MST(P)$ . Then, there exists a path  $\pi$  from  $p_1$  to  $p_2$  in  $MST(P)$  which is not a single edge connecting  $p_1$  and  $p_2$ . By optimality of  $MST(P)$ , the length of each edge on  $\pi$  is at most  $|p_1 p_2|$ . This implies that we can replace the edge between  $p_1$  and  $p_2$  in

$MST(P \cup \{q\})$  by the original path  $\pi$  connecting  $p_1$  and  $p_2$  in  $MST(P)$  without increasing the length of the longest edge contained in the resulting Steiner tree.  $\square$

Due to Lemma 1, we assume that every newly added edge in our optimal solution  $MST(P \cup \{q\})$  from  $MST(P)$  is incident to  $q$ .

**Lemma 2.** *For any  $q \in \mathbb{R}^2$ ,  $b(P \cup \{q\}) \geq |e_5|$ . Therefore, there exists a bottleneck Steiner tree of  $P$  with a single Steiner point such that it contains the edges  $e_5, e_6, \dots, e_{n-1}$  and thus the degree of the Steiner point is at most 5.*

*Proof.* Assume to the contrary that  $b(P \cup \{q\}) < |e_5|$ . This means that  $MST(P \cup \{q\})$  does not contain  $e_1, \dots, e_5$  any more. This also implies that  $q$  has at least 6 incident edges in  $MST(P \cup \{q\})$  by Lemma 1.

Now, we construct another tree  $T$  on  $P \cup \{q\}$ : We add  $e_j$  again to  $MST(P \cup \{q\})$  for  $j \geq 6$ , if it has been removed from  $MST(P)$ , and delete the same number of edges incident to  $q$  to make the resulting graph a tree. Then,  $q$  has exactly 6 incident edges in  $T$ . Let  $N = \{p_1, \dots, p_6\}$  be the set of the 6 points in  $P$ , which are adjacent to  $q$  in  $T$ , in clockwise order at  $q$ . Observe that there exists an integer  $a$  with  $1 \leq a \leq 6$  such that the angle  $\angle p_a q p_{a+1}$  is at most  $60^\circ$ . By simple trigonometry,  $|p_a p_{a+1}|$  is at most  $|qp_a|$  or  $|qp_{a+1}|$ , that is,  $|p_a p_{a+1}| \leq \max\{|qp_a|, |qp_{a+1}|\}$ . Also, note that we have  $|p_i p_{i+1}| \geq |e_5|$  for any  $i$ , since each  $p_i$  belongs to a different subtree after removing the five longest edges  $e_1, \dots, e_5$  from  $MST(P)$ , and that  $|qp_i| \leq b(P \cup \{q\}) < |e_5|$  by the assumption. Hence, we have

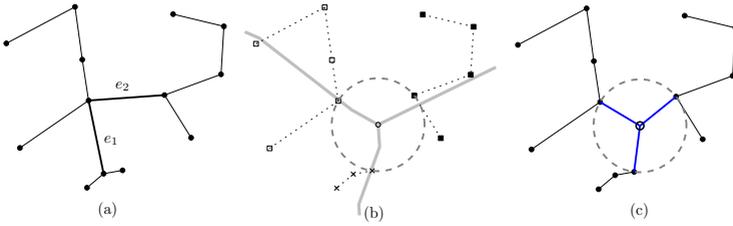
$$|e_5| \leq |p_a p_{a+1}| \leq \max\{|qp_a|, |qp_{a+1}|\} \leq b(P \cup \{q\}) < |e_5|,$$

a contradiction.

Furthermore, the equality holds only when the six points  $N$  form a regular hexagon centered at  $q$ . Hence, if  $q$  has six or more incident edges, we can just remove some while keeping the longest length in the resulting tree. Thus, the lemma is shown.  $\square$

By the above lemma, we remove at most four longest edges from  $MST(P)$ , and also newly added edges are all incident to  $q$ . Thus, we have only four possibilities:  $e_1, \dots, e_c$  are removed from  $MST(P)$ , where  $c = 1, \dots, 4$ . Our algorithm is summarized as follows: (1) Remove  $e_1, \dots, e_c$  from  $MST(P)$ . Then, we have a forest containing  $c + 1$  subtrees  $T_1, \dots, T_{c+1}$ . (2) Find a smallest disk containing at least one point 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  $C := \{1, 2, \dots, c + 1\}$  be a color set, we color each point of  $T_i$  by color  $i \in C$ . This problem can be solved in  $O(cn \log n)$  time by computing the *farthest color Voronoi diagram* [1, 6]: given a collection  $\mathcal{C} = \{P_1, \dots, P_c\}$  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



**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 FCVD (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.

```

1: procedure 1-EUCLIDBST( $P$ )                                ▷  $P$ : a set of points in  $\mathbb{R}^2$ 
2:   Compute  $MST(P)$                                        ▷ minimum spanning tree
3:   Sort the edges of  $MST(P)$  by their lengths
4:   Let  $e_1, \dots, e_{n-1}$  be the edges in the order
5:   for  $1 \leq c \leq 4$  do
6:     Remove  $e_1, \dots, e_c$  from  $MST(P)$  to get  $c + 1$  subtrees  $T_1, \dots, T_{c+1}$ 
7:     Compute  $FCVD(\mathcal{C})$  where  $\mathcal{C} := \{T_1, \dots, T_{c+1}\}$ 
8:     Traverse  $FCVD(\mathcal{C})$  to find the center of a smallest color-spanning disk  $D$ 
9:     if If the radius of  $D$  is smaller than  $|e_c|$  then
10:        $q \leftarrow$  the center of  $D$ 
11:     else
12:       return  $q$ 
13:     end if
14:   end for
15:   return  $q$ 
16: end procedure

```

**Fig. 3.** Exact algorithm to EUCLIDBST for  $k = 1$

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  $\mathcal{C} := \{T_i \mid 1 \leq i \leq c + 1\}$ , and build  $FCVD(\mathcal{C})$  to compute the center of a smallest color spanning disk, which is the candidate of an optimal location of the Steiner point. Repeating this procedure for  $c = 1, \dots, 4$  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 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.* The time and space complexity of the algorithm is easily checked. To prove the lower bound of the problem, consider the *maximum gap* problem:

*Problem 3 (MAXGAP).* 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 [2]; a proof can be found in the book by Preparata and Shamos [9]. 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 to 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.  $\square$

### 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. First, Lemma 2 extends to the following.

**Lemma 3.** *Let  $k$  be the number of allowed Steiner points in an instance of the Euclidean bottleneck Steiner tree problem. Then,  $b(P \cup \{q_1, \dots, q_k\}) \geq |e_{4k+1}|$  for any  $k$  points  $q_1, \dots, q_k \in \mathbb{R}^2$  and there exists a Euclidean bottleneck Steiner tree in which each Steiner point has degree at most 5.*

*Proof.* Our proof is by induction on  $k$ . The case when  $k = 1$  is proven by Lemma 2. Suppose that there exists a smallest positive integer  $k' > 1$  such that  $b(P \cup Q) < |e_{4k'+1}|$  where  $Q = \{q_1, \dots, q_{k'}\}$  is a set of optimal locations of  $k'$  Steiner points. Let  $T$  be the resulting Steiner tree of vertices  $P \cup Q$ . By the assumption,  $T$  does not contain  $e_1, \dots, e_{4k'+1}$  of  $MST(P)$ . Also, we can assume that each  $q_i$  is of degree at most 5 in  $T$  by Lemma 2: consider the problem instance where we are given  $P \cup Q \setminus \{q_i\}$  as terminals and want to locate one Steiner point. On the other hand, consider the problem instance where we are given  $P \cup \{q_i\}$  as terminals and we want to locate  $k' - 1$  Steiner points. By the assumption, an optimal location of  $k' - 1$  Steiner points removes at most  $4k' - 4$  edges of  $\{e_1, \dots, e_{4k'+1}\}$ . This means that  $q_i$  removes at least 5 edges among them and  $T - q_i$  is a forest consisting of at least 6 subtrees, implying that the degree of  $q_i$  in  $T$  should be at least 6, a contradiction. Hence, there does not exist such  $k'$  and the lemma is true.  $\square$

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

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

Let  $V := \{v_1, \dots, v_{c+1}, s_1, \dots, s_k\}$  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 [10]. Together with Lemma 3, 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  $k \leq c \leq 4k$ , we remove  $e_1, \dots, e_c$  from  $MST(P)$  to get  $c + 1$  subtrees  $T_1, \dots, T_{c+1}$ . 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  $k \leq c \leq 4k$ , 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.

**Lemma 4.** *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  $k \leq c \leq 4k$ .*

*Proof.* Since each Steiner point has degree at most 5 and  $c \leq 4k$  by Lemma 3, 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

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

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

Note that there was no known exact algorithm even for this special case of the Euclidean bottleneck Steiner tree problem. Thus, it is worth noting the following theorem.

**Theorem 2.** *There is an  $O((4k)!k!k^2/6^k \cdot n \log n)$  time algorithm to compute an optimal Euclidean bottleneck Steiner tree with no edge between Steiner points when we are given  $n$  terminals in the plane and allow  $k$  Steiner points.*

*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 4 and it is easy to see that each enumeration can be done in  $O(1)$  time. A rough calculation results in  $\sum_{c=k}^{4k} O((4k)!k!/6^k \cdot cn \log n) = O((4k)!k!k^2/6^k \cdot n \log n)$ .  $\square$

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  $k$  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

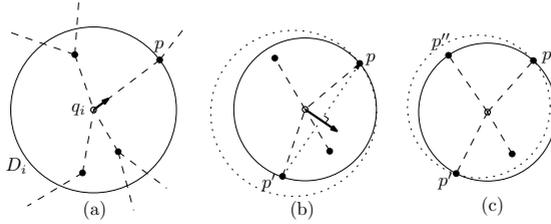
Consider an optimal location of  $k$  Steiner points and its resulting Euclidean Steiner tree; the optimal location of  $s_i \in V$  is denoted by  $q_i \in \mathbb{R}^2$  and the resulting Steiner tree is denoted by  $T^*$ , where  $Q := \{q_1, \dots, q_k\}$ . Let  $N_i$  be the set of points in  $P \cup Q$  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 5.** *There exists an optimal location of  $k$  Steiner points,  $q_1, \dots, q_k$  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.* We first prove that there exists an optimal solution satisfying (1). See Figure 4. Suppose that  $q_1, \dots, q_k$  are an optimal location of  $k$  Steiner points and that there is  $D_i$  which does not satisfy (1). Then, we have two possibilities: there are no three nor two points in  $N_i$  determining  $D_i$ , or two in  $P \cap N_i$  determining  $D_i$  belong to the same  $T_j$  for some  $j$ . We can simply discard the latter case since it makes a cycle in the resulting Steiner tree; thus, this is the former case. If there is only one point  $p \in N_i$  on the boundary of  $D_i$ , we can continuously move  $q_i$  in direction towards  $p$  while maintaining  $N_i$  to lie inside of the corresponding disk  $D_i$ , until  $D_i$  hits another point  $p' \in N_i$ . Observe that the radius of  $D_i$  decreases during this process, and two points  $p$  and  $p'$  lie on the boundary of  $D_i$  in the end.

Now, we can assume that we have at least two points  $p, p'$  on the boundary of  $D_i$ . If  $p$  and  $p'$  define the diameter of  $D_i$ , we are done. Hence, we are left with the case when the smaller angle  $\angle pq_i p'$  at  $q_i$  is strictly less than  $180^\circ$ . We move  $q_i$  along the *bisecting line*<sup>1</sup> between  $p$  and  $p'$  in the direction that the smaller angle at  $q_i$  increases. Then, the smaller angle  $\angle pq_i p'$  increases continuously and the radius of  $D_i$  decreases continuously, locally. We stop moving  $q_i$  when the boundary of  $D_i$  touches a third point  $p'' \in N_i$  or when the angle  $\angle pq_i p'$  reaches exactly  $180^\circ$ . Also, observe that the radius of  $D_i$  decreases during this second movement of  $q_i$ .

<sup>1</sup> The bisecting line between two points  $p$  and  $p'$  is the set of equidistant points in  $\mathbb{R}^2$  from the two points, that is, the line  $\{x \in \mathbb{R}^2 \mid |xp| = |xp'|\}$ .



**Fig. 4.** Illustration for the proof of Lemma 5. Black dots are those in  $N_i$  around  $q_i$ . (a) When there is only one point  $p \in N_i$  on the boundary of  $D_i$ , (b) we can move  $q_i$  in direction towards  $p$  so that the boundary of  $D_i$  touches another point  $p'$  and further (c) in direction along the bisecting line between  $p$  and  $p'$  so that three points  $p, p'$ , and  $p''$  determines  $D_i$ . During this process, the radius of  $D_i$  is subsequently decreasing.

From now, suppose that  $q_1, \dots, q_k$  is an optimal solution satisfying (1) but not (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 we connect  $q_i$  to  $p'$ , instead, and perform the above process to make  $D_i$  satisfy (1). Therefore, the lemma is shown.  $\square$

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 5; 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*. The following is an immediate observation.

**Lemma 6.** *Suppose that there is an edge in  $T^*$  between two Steiner points  $q_i, q_j \in Q$ . If  $q_i$  is flexible and  $q_j$  is a determinator of  $D_i$ , then  $r_j \geq r_i = |q_i q_j|$ .*

### 3.3 Exact Algorithm for $k = 2$

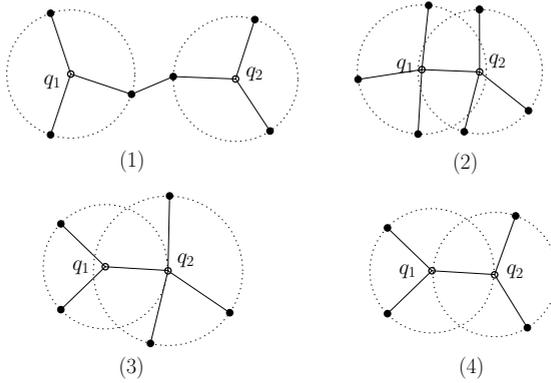
Based on the properties observed above, we present an exact algorithm for computing the Euclidean bottleneck Steiner tree with two Steiner points.

**Lemma 7.** *An optimal location  $\{q_1, q_2\}$  of two Steiner points fall into one of the following cases:*

- (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  $c \leq 8$  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) of Lemma 7.

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  $C := \{1, 2, \dots, c + 1\}$  and  $C_j \subset C$  be defined as



**Fig. 5.** Illustration to four cases in Lemma 7. Small black circles and black dots represent Steiner points and terminals, respectively.  $D_1$  and  $D_2$  are depicted as dotted circles centered at  $q_1$  and  $q_2$ .

$C_j := \{i \in C \mid v_i \text{ adjacent to } s_j\}$ , 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 5; 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 independent from finding  $q_2$  even if  $q_2$  is flexible. Thus, even if  $q_1$  is solid and  $q_2$  is flexible, then we can just compute the exact location of  $q_1$  as above, and next compute  $q_2$  by adding one more colored point  $\{q_1\}$  into  $\mathcal{C}_2$  and performing the above procedure as if  $q_2$  were solid. Hence, solutions in Cases (2) and (3) can be checked in  $O(cn \log n)$  time.

What remains is the case when both  $q_1$  and  $q_2$  are flexible. In this case, we have  $r_1 = r_2$  by Lemma 6 since  $q_1$  is a determinant of  $D_2$  and vice versa. 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 case of two terminals  $p_1$  and  $p_2$ ,  $q_1$  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  $j, j' \in \mathcal{C}_2$ . Then, we find a point  $x_2 \in e$  such that  $2|x_2p| = |x_2p'|$ , 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  $i, i' \in \mathcal{C}_1$  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  $j, j' \in \mathcal{C}_2$  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) of Lemma 7. 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  $1 \leq c \leq 8$ . 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. Surprisingly, any exact algorithm to EUCLIDBST or EUCLIDBST-FT is still unknown for three or more Steiner points, even that with time super-polynomial in  $n$  or  $k$ . However, it seems not so simple to compute an optimal location of Steiner points when there are many flexible Steiner points. It might need more geometric or combinatorial observations on situations, which have not been discovered yet.

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, observe 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 [6], could be used to build an algorithmic block as we did for the Euclidean case.

## References

1. Abellanas, M., Hurtado, F., Icking, C., Klein, R., Langetepe, E., Ma, L., Palop, B., Sacristán, V.: The farthest color Voronoi diagram and related problems. Technical Report 002, Institut für Informatik I, Rheinische Friedrich-Wilhelms-Universität Bonn (2006)
2. Ben-Or, M.: Lower bounds for algebraic computation trees. In: Proc. 15th Annu. ACM Sympos. Theory Comput (STOC), pp. 80–86. ACM, New York (1983)
3. Chiang, C., Sarrafzadeh, M., Wong, C.: A powerful global router: based on Steiner min-max trees. In: Proc. IEEE Int. Conf. CAD, pp. 2–5 (1989)
4. Elzinga, J., Hearn, D., Randolph, W.: Minimax multifacility location with Euclidean distances. *Transport. Sci.* 10, 321–336 (1976)
5. Ganlet, J.L., Salowe, J.S.: Optimal and approximate bottleneck Steiner trees. *Oper. Res. Lett.* 19, 217–224 (1996)
6. Huttenlocher, D.P., Kadem, K., Shrir, M.: The upper envelope of Voronoi surfaces and its applications. *Discrete Comput. Geom.* 9, 267–291 (1993)
7. Li, Z.-M., Zhu, D.-M., Ma, S.-H.: Approximation algorithm for bottleneck Steiner tree problem in the Euclidean plane. *J. Comput. Sci. Tech.* 19(6), 791–794 (2004)
8. Love, R., Wesolowsky, G., Kraemer, S.: A multifacility minimax location problem with Euclidean distances. *J. Prod. Res.* 11, 37–45 (1973)
9. Preparata, F.P., Shamos, M.I.: *Computational Geometry*. Springer, Heidelberg (1985)
10. Sarrafzadeh, M., Wong, C.: Bottleneck Steiner trees in the plane. *IEEE Trans. Comput.* 41(3), 370–374 (1992)
11. Wang, L., Du, D.-Z.: Approximations for a bottleneck Steiner tree problem. *Algorithmica* 32, 554–561 (2002)
12. Wang, L., Li, Z.: An approximation algorithm for a bottleneck  $k$ -Steiner tree problem in the Euclidean plane. *Inform. Process. Lett.* 81, 151–156 (2002)